

PHOTOMETRIC OBJECT MODELING  
–RENDERING FROM A DENSE/SPARSE SET OF IMAGES–

BY  
KO NISHINO

A DOCTORAL DISSERTATION

SUBMITTED TO  
THE GRADUATE SCHOOL OF  
THE UNIVERSITY OF TOKYO  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF SCIENCE  
IN INFORMATION SCIENCE

DECEMBER 2001



## ABSTRACT

For some time, rendering photorealistic synthetic images from observations of real objects has been a major research topic in the computer vision and computer graphics communities. An extensive amount of work in this research area has led to a few representative schools: image-based rendering, inverse rendering and 3D photography. However, each of these methods still suffers from several drawbacks, such as massive data storage, constraints on applicable objects, restrictive scenarios on application, etc. To overcome these problems, we propose two methods. The first method, *Eigen-Texture Rendering*, handles the appearance variation of the target object on its own 2D surface, enabling effective compression and interpolation with PCA thereby resulting in a compact representation of objects with arbitrary reflectance properties. The second method, *Rendering from a Sparse Set of Images*, recovers both the illumination distribution and reflection parameters simultaneously from input images, providing even more compact representation for photorealistic image synthesis. With the leverage of assuming a specific reflection model and restricting the treatable objects, the latter method provides more flexibility in application, i.e., fewer input images. In this dissertation, we present the theory of these methods, and report on the results obtained by applying the methods to real world objects.



## 論文要旨

実物体の観察にもとづく仮想物体画像の合成は、コンピュータビジョン、コンピュータグラフィックスの分野にまたがり活発に研究がおこなわれている研究課題である。現在までに、Image-based Rendering, Inverse Rendering, 3D Photography などの手法が提案されているが、非常に大きなデータ量を保存する必要があったり、適用できる物体が限られていたりとそれぞれに制限が存在する。本論文では、これらの問題に配慮し、新たに2つの手法を提案する。ひとつめの手法である *Eigen-Texture Rendering* は、物体の見えを物体表面上で扱うことにより主成分解析による効率の良い圧縮と補間を可能し、あらゆる反射特性の物体に対して効率の良い見えの表現を実現する。ふたつめの手法である *Rendering from a Sparse Set of Images* は、入力画像から光源状況と物体の反射パラメタを同時に推定する手法であり、より効率の良い見えの表現を実現する。また、前者の手法に比べ、特定の反射モデルを仮定することにより適用できる物体に制限を設けるものの、非常に少ない入力画像から任意視点画像の合成を実現するという利便性を兼ね備える。本論文では、これらの手法の提案と実物体を用いた実験による有効性の評価をおこなう。



# Acknowledgements

I would first like to thank Katsushi Ikeuchi, my advisor and my mentor. Had I not met him, I would never have dreamed of pursuing my career as a researcher. Five years of interaction with him have brought me tremendous experience that I could never have gained elsewhere. Being his first student after he came back to Japan was a prestigious honor for me. Starting from only a handful people, our lab has now grown faster than Moore's law. If I am responsible for even a small bit of this prosperity, I am more than delighted. I gratefully thank him for giving me the freedom to explore whatever I was interested in and for having an open door whenever I needed it.

Yoichi Sato, my former co-advisor and co-worker, deserves special thanks for helping me make a good start. His practical advices have always helped me.

I would also like to express my deepest gratitude to Zhengyou Zhang, my mentor at Microsoft Research Redmond. I was very fortunate to have him as my mentor; he gave me the freedom to explore my ideas and to conduct research as an independent researcher. I have learnt a lot from his passion and self-disciplined attitude toward research. My good friends at Microsoft Research Redmond, Mike Sinclair, Zicheng Liu and Ying Shan, also deserve special thanks for inspiring me with many good ideas and insights. Doing hardware work with Mike was a lot of fun.

Being the first student of Katsu in Japan was not always easy. Whenever I was experiencing a difficult time, daily communication with my colleagues in the lab encouraged me and made me relax. Especially, conversations, including really silly ones, with George V. Paul, Kentaro Kawamura and Taku Nishikawa have always relaxed me and fully recharged my energy to enable me to overcome the hard times. I would like to express my deepest gratitude to them.

For taking the time to edit and proofread most of my publications, including this thesis, I am very grateful to Marie Elm. She always has been kind to spare her time for correcting my writing and improving my writing skills.

I was lucky to have many great people to work with in the same laboratory. Special thanks go to Ryusuke Sagawa, who has collaborated with me on the range image integration described in this thesis. I am also very proud of, and was fortunate to work with, the talented people in the photometry group. Although I cannot name all because of the limited space, I am very grateful to the people I have met and interacted with in this lab.

My friends outside the lab, have helped me a lot in many aspects. Especially, I thank from the bottom of my heart, Shin-ichi Fukushima and Masaki Hilaga, who helped me to overcome the most challenging ordeal throughout my graduate school life. I would also like to express my deepest gratitude to Emiko Unno, who has always encouraged me with her delightfulness.

Memories of Ryoko Mizukami, who lived a short but condensed life with passion and vitality, have always straightened me up and pushed me forward. I wish that I could show her this thesis and ask her whether I'm living up to my potential.

Lastly, I would like to thank my family – my Father, Mother and Sister – for their support and patience during these many years and in the years to come.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Related Work . . . . .	3
1.2.1	Rendering from a Dense Set of Images . . . . .	3
1.2.2	Rendering from a Sparse Set of Images . . . . .	6
1.3	Overview . . . . .	9
<b>2</b>	<b>Geometric Modeling</b>	<b>13</b>
2.1	Introduction . . . . .	13
2.2	Robust Simultaneous Registration of Multiple Range Images . . .	16
2.2.1	Related Work . . . . .	16
2.2.2	Approach . . . . .	18
2.2.3	Point Mate Search . . . . .	20
2.2.4	Least-square Minimization Strategy . . . . .	25
2.2.5	Results . . . . .	30
2.3	Robust and Adaptive Integration of Multiple Range Images . . .	38
2.3.1	Related Work and Overview . . . . .	38
2.3.2	Volumetric Range Image Integration . . . . .	40
2.3.3	Consensus Surface with Photometric Attributes . . . . .	43
2.3.4	Integrating in Adaptive Resolution . . . . .	45
2.3.5	Results . . . . .	49
2.4	Summary . . . . .	51
<b>3</b>	<b>Eigen-Texture Rendering</b>	<b>55</b>
3.1	Approach . . . . .	55

3.2	Eigen-Texture . . . . .	56
3.3	Implementation . . . . .	61
3.3.1	System Setup . . . . .	61
3.3.2	Cell-adaptive Dimensional Eigenspace . . . . .	62
3.3.3	Compression Ratio . . . . .	68
3.3.4	Interpolation in eigenspace . . . . .	69
3.4	Integrating into real scenes . . . . .	71
3.5	Summary . . . . .	73
<b>4</b>	<b>Rendering from a Sparse Set of Images</b>	<b>77</b>
4.1	Approach . . . . .	77
4.2	Reflection Separation . . . . .	80
4.2.1	Reflection Model . . . . .	80
4.2.2	Diffuse Texture Map . . . . .	82
4.2.3	Illumination Hemisphere . . . . .	83
4.3	Illumination and Reflectance Parameter Estimation . . . . .	84
4.3.1	Problem Formulation . . . . .	84
4.3.2	Specular Reflection as Convolution . . . . .	86
4.3.3	Alternating Minimization . . . . .	87
4.3.4	Reflectance Parameter and Illumination Hemisphere Esti- mation through Alternating Minimization . . . . .	89
4.4	View-dependent Rendering . . . . .	91
4.5	Relighting . . . . .	91
4.6	Results . . . . .	93
4.6.1	Inputs . . . . .	93
4.6.2	Estimated parameters . . . . .	94
4.6.3	View-dependent Rendering . . . . .	98
4.6.4	Changing Reflection Parameters . . . . .	98
4.6.5	Relighting . . . . .	98
4.7	Summary . . . . .	100
<b>5</b>	<b>Conclusions</b>	<b>105</b>
5.1	Summary . . . . .	105
5.2	Contributions . . . . .	107

5.3 Future Work and Discussion . . . . .	108
<b>References</b>	<b>111</b>

# List of Figures

2.1	Geometric modeling pipeline. . . . .	14
2.2	K-d tree subdivision of 2D points. The number indicates the level of the tree at which the split occurs. . . . .	21
2.3	Point correspondences using point-point and point-plane distance metric. . . . .	22
2.4	Images using laser reflectance strength as pixel values. . . . .	23
2.5	Point mates using point-point distance metric with reflectance strength values. Different shape marks indicate different reflectance strength values. . . . .	24
2.6	Weight values corresponding to several M-estimation functions. . . . .	28
2.7	Initial positions of the cupnoodle sequence. . . . .	31
2.8	Left column: After registration without intensity attribute. Right column: After registration with intensity attribute. Each row shows a different range image. Grids are depicted for better visualization. . . . .	33
2.9	Initial positions of the Noisy Cat sequence. . . . .	34
2.10	Left: Registered with [84] Right: Registered with our method.(Viewing from the top left of the cat.) . . . . .	34
2.11	Histogram of errors. . . . .	35
2.12	M-estimator error v.s. iteration number . . . . .	36
2.13	Registered Great Buddha. Different color indicates different range image. . . . .	36
2.14	Registered Great Buddha. . . . .	37
2.15	Zero-crossing interpolation from the grid sampling of an implicit surface . . . . .	40

2.16	Naive algorithm: An example of inferring the incorrect sign of a voxel's value, $f(x)$ , due to a single noisy triangle. . . . .	41
2.17	Consensus surface algorithm: The signed distance is chosen from the consensus surfaces inside the circle. . . . .	42
2.18	A 2D slice of the octree splitting volume. The resolution is high around the surface and low elsewhere. . . . .	43
2.19	An example of the histogram of the intensity values of consensus points. Some outliers due to specular reflection are observed. In this case, the median value is 0.04. . . . .	46
2.20	Our method approximates neighboring range images points to a plane and computes its normal vector $\bar{n}$ by principal component analysis (PCA) for the cloud of range image points. The approximate normal is then used to determine further subdivision. . . . .	47
2.21	Edges connecting adjacent voxels in an adaptive octree. . . . .	49
2.22	Range and intensity images of the box is acquired from various viewpoints using a turn table. These are two intensity images of sixty input images. Some images include specular reflection. . . . .	51
2.23	Merging results of a box with texture: the model in the left column is created by adaptive subdivision based on the curvature of the surface, and the one in the right column is created by adaptive subdivision based on both the surface curvature and the variation of intensity. . . . .	52
2.24	Merging results of the Kamakura Buddha: Column: (A) without adaptive subdivision, (B) with adaptive subdivision based on surface curvature, (C) with adaptive subdivision based on both surface curvature and LRS variation. Row: (1) wireframe model, (2) polygonal model, (3) images rendered with LRS values. The far upper and far lower rows are zoom-ups of the forehead of the Buddha. . . . .	53
3.1	Outline of the <i>Eigen-Texture Rendering</i> method. . . . .	59
3.2	A sequence of cell images. . . . .	60
3.3	The system setup we use. . . . .	61

3.4	Virtual object image synthesized by using three dimensional eigenspaces (an example of a highly reflective object). . . . .	63
3.5	Virtual object image synthesized by using three dimensional eigenspace (an example of an object with rough surface). . . . .	64
3.6	Graph of the eigen ratio and the error per pixel versus the number of dimensions of eigenspace. . . . .	66
3.7	Left: Input color images, Right: Synthesized images (by using cell-adaptive dimensional eigenspaces). . . . .	67
3.8	Interpolation results in the eigenspace. . . . .	70
3.9	Synthetic images reconstructed by interpolating input images in eigenspace. . . . .	71
3.10	Side by side comparison of input images and rendered images with interpolation in eigenspace. The original images are in the first column and the rendered images are in the second column. The first and last images in the original images were used as the input; the intermediate images in the rendered sequence are the results of interpolation. . . . .	72
3.11	Zoom ups of the images in Figure 3.10. . . . .	73
3.12	A comparison with MPEG1 compression accomplished on the image. . . . .	73
3.13	Linear combination of light sources. . . . .	74
3.14	Integrating virtual object into real scene. . . . .	75
4.1	The mechanism of dichromatic reflection model on a object surface.	81
4.2	Six input HDR images of the Bali mask. . . . .	94
4.3	Diffuse-texture-mapped images. . . . .	95
4.4	Separation of reflection components. Left: original input image, Middle: diffuse reflection image (diffuse-texture-mapped image), Right: specular reflection image. . . . .	96
4.5	Left and Middle: partial illumination hemispheres corresponding to two viewpoints in the original input image sequence. Right: the combined initial illumination hemisphere. . . . .	97

4.6	Left: Estimated illumination hemisphere, Right: An photograph of the true illumination distribution captured through a fish-eye lens and flipped horizontally. . . . .	97
4.7	Comparison on input images. Left: input real images, Middle: synthesized images, Right: difference . . . . .	99
4.8	Left: real photograph, Middle: synthetic image, Right: difference.	100
4.9	Left Top: $\sigma = 0.1$ $K_s = 1.0$ , Right Top: $\sigma = 0.04$ $K_s = 1.0$ , Left Bottom: $\sigma = 0.075$ $K_s = 2.0$ , Right Bottom: $\sigma = 0.075$ $K_s = 0.3$ . . .	101
4.10	Left Column: real photographs, Right Column: synthetic images.	102
4.11	Synthetic images rendered under a moving light source. . . . .	103

# List of Tables

2.1	Comparison of errors in mm. . . . .	32
2.2	Statistics of Models of the Buddha: Our method reduces the amount of data and computation time. However, the mean errors are quite small compared with the Buddha size. . . . .	50
3.1	Some statistics of the compression results. . . . .	66

# Chapter 1

## Introduction

### 1.1 Background

With the rapid growth of the ubiquitous computational environment that surrounds us, we now have greater needs as well as the ability to construct a hybrid environment where reality and virtuality are seamlessly integrated. Many applications, including digital museums, tele-conferencing, e-commerce, medical surgical assistance, etc. can be built into the framework of these hybrid environments. Many exciting research topics and applications arise from this area. Among these topics are: how do we construct a realistic virtual object/scene, how do we superimpose that scene onto a real world environment, and how can we immerse ourselves to that integrated environment.

We recently started a project to construct a digital archive of cultural heritage objects [47, 74] as an application of these hybrid environments. The goal of our project is to develop a scheme to digitally archive cultural heritage objects as so called multimedia contents. We Japanese, are in a unique position in this respect, because most of the Japanese cultural assets are made of wood and paper, while most foreign assets are made of stone. Thus, at any moment, our heritage objects can be destroyed due to natural disasters such as floods, fires, typhoons. Therefore, it is very meaningful to digitally archive those cultural assets in order to permanently preserve the great heritage objects bequeathed to us by our ancestors, so that they can be safely passed down to future generations. Also, this digital archive will enable people to see and feel the great historical heritages

without spending time and money to visit their actual locations. Furthermore, it will let us Japanese, introduce our outstanding cultural heritage to the world through the digital network. While the project itself has not yet reached the stage of appearance preservation, the demands are clear: an accurate and efficient representation of both geometry and photometry of real objects.

In this dissertation, we focus on one of the fundamental research topics to construct these real and virtual hybrid environments: how do we construct a photorealistic virtual object from the observation of a real world object? As we aim to construct a “photometrically real looking” object, we will refer to this research topic as *photometric object modeling*.

The goal of photometric object modeling is to digitally represent a real world object so that its appearance, when displayed on a computer, looks realistic. As a measurement of the reality we find in the displayed virtual object, we commonly use the term “photorealistic”. Although the precise definition of “photorealism” remains quite fuzzy, an extensive amount of research has been conducted in mainly the computer vision and computer graphics communities to make the virtual objects look as if they were a photograph of real objects. As we will review in the next section, the current trend of photometric object modeling is based on the observation of real objects. Since we try to make the synthetic virtual objects photorealistic, it is quite natural to start from photographs of real objects. In addition to the images (photographs), the leverage of knowing the geometry of the target object brings about a lot of information on analyzing the appearance of objects. This is also natural, since people see real world objects as a mixture of geometry and photometry, not as 2D images. Thus, while we propose two new methods to render synthetic virtual object images from the observation of real objects, the basic underlying philosophy is to take full advantage of the geometry we assume to have been obtained *a priori*. Also, for geometric modeling, where we reconstruct a geometric model to be used as the basis of our appearance analysis and synthesis, we propose two new methods to register and integrate range images robustly and adaptively, while assuming that we can obtain range images using existing techniques such as laser range finders, stereo systems etc.

## 1.2 Related Work

The ultimate goal of computer vision and computer graphics research is to visually bring the whole real world into the digital world as it is; with “realism”. As a natural result, although the approaches are from opposite directions, where computer vision studies are concerned with how we see, while computer graphics studies are concerned with how to display those images, both literatures converge at a middle point on several research topics. Photometric object modeling, whose aim is to construct a photorealistic virtual object inside the computer, is one of the typical research topics that lie on the overlapping boundaries of vision and graphics. We will overview past work on photometric object modeling with regard to the methods we will present in Chapter 3 and Chapter 4. Related work of the geometrical modeling part of our work will be presented in Chapter 2.

### 1.2.1 Rendering from a Dense Set of Images

An extensive amount of research has been conducted in both the computer vision and the computer graphics communities with the goal of efficiently representing a real object on the computer to synthesize photorealistic images from arbitrary viewpoints and novel lighting environments (relighting). This research constitutes three schools: image-based rendering, inverse rendering and 3D photography. Most of these methods rely on a dense sampling of the appearance variation, requiring a large number of images of the target object.

Image-based rendering is a rendering method that basically does not assume any geometry. Taking only 2D images as the input, image-based methods set their basis on the fact that light rays can be parameterized as a 7D function called the *plenoptic function* [1]. Considering each pixel in real images as a sample of this plenoptic function, image-based methods synthesize virtual images by selecting the most appropriate sample of rays, or by interpolating between the sampled rays [78]. Levoy and Hanrahan [63] represent the light rays in free space (space free of occluders) in 4D by using two 2D planes, thereby enabling the plenoptic function to be described in a compact manner. Gortler et al. [39] adopt a similar two 2D plane representation, but additionally they use rough geometric information, in a volumetric representation, derived from images, to correct the

basis function of the rays. Shum et al. [112] proposed a 3D representation of the plenoptic function. The method is especially suited to rendering scenes with the viewing direction inside-out, which is the case when modeling real world scenes rather than objects. Since these plenoptic function-based methods require only real images as the input, they provide high generality, i.e., they can be applied to a wide variety of objects and scenes. Because of the principle of interpolation, these approaches tend to require a large amount of input images: a very dense sampling of the appearance variation of objects under a static illumination environment. Even though compression techniques such as vector quantization- or discrete cosine transformation (DCT)-based approaches can significantly reduce the total amount of information that has to be stored on the computer, they still require the user to take hundreds of images as the input.

To estimate the lower bound of the number of images necessary for image-based rendering, Chai et al. [10] have proposed a framework to analytically compute the minimum sampling rate from spectral analysis with the use of sampling theorem. However, since they assume diffuse surfaces, view-dependent variance such as specularities that plays an important role in photorealistic synthesis of images is not taken into consideration. These view-dependent components occur in a high frequency domain compared with diffuse components, so that it excessively increases the sampling rate. Recently, several propositions have been made to incorporate rough geometrical structure in the framework of image-based rendering. The underlying geometry can be constant depth, layers, sprites etc. [9, 40, 51], and the geometry is basically used to provide a better strategy to pick up the closest ray sample [8].

“Inverse rendering” is another major approach in this area. Taking a dense set of images and a 3D model of the target object model reconstructed *a priori*, inverse rendering methods estimate an approximation of the full bidirectional reflectance distribution function (BRDF) [85] of the object surface by fitting a particular reflection model to the pixel values observed in input images [105] or by solving the inverse radiosity problem [131]. As these methods estimate the BRDF parameters of the object surface, view-dependent rendering and relighting can be accomplished with a very compact representation of the object. In these methods the radiance and positions of the light sources need to be known to compute the BRDF, and direct information of lighting environment has to be

provided in some way, e.g., with high dynamic range images [24, 73] of the light sources, making the method less attractive for practical use. A detailed review of inverse rendering methods will be described in the next section, since it will be closely related to our work presented in Chapter 4.

Recent research in the so-called “3D photography” domain have proposed methods that lie somewhere between these two major research streams of image-based rendering and inverse rendering. By taking advantage of the latest advances in 3D sensing equipments, such as laser range scanners [20, 19, 72] and structured light scanners [102], these approaches try to make full use of the 3D geometry as well as the images. The method presented in Chapter 3, Wood et al. [127] and W-C. Chen et al. [14] (both after the first publication of Chapter 3 [90]) can be considered as setting one of the 2D planes in the light field approaches attached to the object surface, in concrete, on the coarse triangular patches or dense surface points, respectively. By deriving information from the geometry in this way, these approaches succeed in achieving higher compression ratio without losing smooth view-dependent variation such as the movement of highlights.

Several other methods, such as “Voxel Coloring”, take intermediate approaches between purely image-based and model-based approaches. Voxel Coloring/Space Carving methods [107, 17, 58] represent the scene with a colored voxel model. By using the fact that surface points in a scene and voxels that represent them project to consistent colors in the input images, the volume containing the scene is iteratively carved out generating a volumetric representation of the scene with its texture color attached on each surface voxel. However, since these methods rely on the color consistency assumption that holds only on object surfaces that have Lambertian reflection property, it has difficulty in handling objects that have reflection properties that are not just diffusive.

The method we present in Chapter 3 for rendering from a dense set of images can be considered as a variant of view-dependent texture mapping. View-dependent texture mapping is a technique to combine multiple textures corresponding to each triangular patch of the geometric model, so that it can be rendered from viewpoints that do not exist in the sampled images. On combining textures, weights are multiplied to the textures, e.g., weights depending on the angle between the triangular patch’s surface normal and viewing an-

gle [95, 26, 25], and blending is accomplished between the weighted textures. Our approach in Chapter 3 is different from these approaches in the respect that we derive the principle components of the textures and synthesize new textures by combining those principle components, thereby providing an intuitive interpolation method to render textures from new viewpoints.

Analyzing real-world textures to derive their essential characteristics is an on-going research topic in the computer vision community. Especially for recognition tasks, several methods have been proposed to derive the basic components of real-world textures sampled under different illumination conditions and viewing directions. For instance, Leung et al. [62] represent textures with *3D textons* extracted by applying filter banks, and Suen et al. [115] derive the basis textures by applying orthonormal basis expansion and define a *dimensionality surface* which represents the number of basis textures necessary to represent the texture. We apply principle component analysis on sampled textures to represent the appearance variation of textures under varying illumination and viewing conditions. While our objective is image synthesis, recognition of real-world texture can be a straightforward application of our approach.

### 1.2.2 Rendering from a Sparse Set of Images

Although the method we present in Chapter 3 provides an efficient representation of real objects for view-dependent rendering and relighting, and enables the user to drastically reduce the number of input images through interpolation in eigenspace, the method still requires a relatively dense set of images; for instance, thirty images for changing viewpoints on a circle surrounding the object. To further cut down on the necessary number of images for another order, we will have to compensate for the information that will be lost from the input images. Hence, we will take an approach similar to inverse rendering.

Forward rendering, or simply rendering, is a process for creating photorealistic synthetic images given the geometry of the scene, including the viewing conditions, the spatial distribution of the light sources, the radiance of the light sources and the 3D model of the target object, and a specific reflection model that approximates the reflection mechanism on the target object surface. Inverse rendering is the exact opposite of this procedure. Given a set of images of a

real world object and a 3D model of the object, inverse rendering extracts the information necessary to synthesize images of the target object from arbitrary viewpoints and under novel lighting environments. The information, in addition to the geometry of the objects and the viewpoints necessary to accomplish the forward rendering, is about the lighting environment, including their spatial and radiance distribution, the texture and the BRDF (Bidirectional Reflectance Distribution Function) [85] of the object. Several approaches have been taken to accomplish inverse rendering under the circumstances where one or two of these three components are unknown.

A number of approaches have been demonstrated for recovering one of the three components when the other two are unknown. When the illumination distribution and the BRDF of the object surface is fully provided, the texture of the object surface can be estimated at each surface point by computing the total irradiance and then dividing the diffuse reflection intensity by that value [67, 64]. (Our method in Chapter 4 does the same thing for relighting diffuse reflection; however, the lighting environment and the BRDF approximation is estimated from images.)

Methods that directly measure the unknown lighting and synthesis of images with known texture and BRDF have been proposed, including methods that use mirror balls [22] and fish-eye stereo systems [100]. As an indirect method, Tomimaga and Wandell [117] recover the color of a single point light source by finding the intersection of the specular reflection direction in RGB space.

Recovering the BRDF or its approximation has been a very active research topic. The full BRDF can be represented by dense samples of it. Image-based measurements of the BRDF [69, 66] and the 2D maps of BRDFs called Bidirectional Texture Function (BTF) [21] have been proposed to tabulate the BRDF under controlled lighting. An alternative approach to sampling the BRDF and using their values, is to approximate the BRDF as a parametric reflection model. From the forward rendering point of view, low-parameter reflection models that capture the reflection mechanism of real object surfaces become convenient. Hence, reflection models such as that of Ward [122] and Torrance and Sparrow [118] have been widely accepted. Several approaches have been proposed to recover the parameters of these reflection models by estimating the texture and the BRDF simultaneously, under a known lighting environment: typically a

single point light source. Nayar et al. [79] estimate the parameters of a hybrid reflection model, consisting of diffuse and specular reflection, and the geometry of the target object from an image sequence captured under a controlled extended light source. Sato et al. [105] take a dense set of images under a single point light source as the input with its position and radiance calibrated. Then they decouple the diffuse reflection and specular reflection for each object surface point via SVD and fit the Lambertian and Torrance-Sparrow reflection model to the diffuse and specular reflection values. Lensch et al. [61] also estimate the spatially varying BRDF by estimating the parameters of the Lafortune reflection model [59], taking a sparser set of images, however still on the order of tens, of the target object under a known lighting condition. Yu et al. [131] take an inverse radiosity approach to recover the diffuse texture and specular reflection model parameters successfully, taking interreflection into account. Sato [103] and Yu and Malik [132] recover reflection model parameters that account for both diffuse and specular reflection of buildings in outdoor scenes, assuming a parametric model for a natural lighting illumination condition.

Ikeuchi and Sato [46] recover the reflectance parameters of a reflection model consisting of the Lambertian model for diffuse reflection and the Torrance-Sparrow model for the specular reflection (the same reflection model we use in Chapter 4) from a single brightness (intensity) image and its corresponding range image taken under a single point light source. They directly fit a linearized version of the two term reflection model by iteratively solving a least square minimization problem and segmenting the image into regions: diffuse, specular, interreflection and shadow regions. Boivin and Gagalowicz [7] take a similar approach to recover the texture and parameters of the Ward reflection model from a single color image captured under a known complex lighting environment. Debevec et al. [23] recover the reflectance parameters of a reflection model specific to human skin from a dense sampling of its appearance variation under controlled lighting captured with a polarizer to separate diffuse and specular reflections.

Marschner and Greenberg [68] solve an inverse lighting problem: given the 3D model and a set of images with their corresponding viewpoints, estimate the lighting environment. They consider the illumination distribution to be a linear combination of basis lightings that corresponds to regions on an imaginary sphere covering the object, and solve a linear system with a regularization

factor imposed on the basis lights, without explicitly estimating the BRDF. With the assumption that the object surface has a Lambertian reflectance property, to relight the target object, the ratio of the forward rendered images under the original lighting and the new lighting is used. In the same spirit, except that we explicitly factor out the illumination distribution and the reflectance property, we use ratios of the incoming irradiance between the original and new illumination distributions to obtain relit diffuse reflection images. In a similar manner, except for using shadows as the cue, Sato et al. [101] recover the illumination distribution of the scene and also the BRDF of the surface on which shadows are cast.

Recently, most closely related to our work in Chapter 4, Ramamoorthi and Hanrahan [96] have proposed a signal processing framework to provide a sound mathematical background to solve many of the problems in the field of inverse rendering. By assuming that the light sources are far enough apart so that they can be parameterized by angles, and that the object surface has a homogeneous reflectance property, they formulate the reflection mechanism as a convolution in the angular space and accomplish inverse rendering through a spherical harmonic representation. While we also assume that the object surface has a homogeneous reflectance property for specular reflection, in terms of the applicable object surfaces, their framework is more general than ours, since they can handle both matte surfaces that have Lambertian reflection [97] and also surfaces with additional specular reflection of micro facet reflection models [96]; we, however, focus only on shiny objects that exhibit specular reflection which can be approximated with a micro facet reflection model. They assume that one out of three components is known while the other two are unknown, specifically the two cases where the texture is known and the lighting and BRDF are unknown and the case where the lighting is known and the texture and BRDF are unknown; we, on the other hand, consider the situation where all three of them are unknown.

### **1.3 Overview**

In this dissertation, we propose two methods to render photorealistic images of an virtual object from observation of real objects. Both methods assume that the

geometry of the target object is known.

Before going into the photometric part of photometric object modeling, we present two new methods for the geometric modeling pipeline in Chapter 2. First, we describe a robust simultaneous registration method to align multiple range images. The proposed method minimizes an error function that is constructed to be global against all range images, providing the ability to diffusively distribute errors instead of accumulating them. The minimization strategy is designed to be efficient and robust against outliers by using a conjugate gradient search that utilizes M-estimator. Also, for “better” point correspondence search, additional attributes of 3D points, such as laser reflectance strength and intensity, are used. For robustness against data noise, the framework is designed not to use secondary information, i.e., surface normals, in its error metric. We describe the details of the proposed method, and present experimental results obtained by applying the proposed method to real data. Also, as an application for digital preservation of cultural heritage objects, we show the results of applying our framework to register a large ancestral object.

Next, we present a robust and adaptive integration method to merge multiple pre-aligned range images. Unlike other volume-based integration methods, our method adaptively subdivides voxels depending on the curvature of the surface to be reconstructed, thereby providing efficient representation of the underlying geometry and efficient use of computational resources. In our range image merging framework, additional attributes, e.g., color, laser reflectance strength, etc., can be taken into account as well as 3D geometric information. This ability allows us to generate 3D models preserving sharp edges around texture boundaries, thereby providing a good basis for efficient rendering and texture mapping. The overall framework is designed to be robust against noise, taking consensus carefully in both geometry and color, which could be suitable for 3D model reconstruction from noisy range images. After describing the system, we present several results of applying our framework to real data.

With regard to the core part of photometric object modeling, we propose two new methods to render photorealistic synthetic images from a set of images and a 3D model. In Chapter 3, we present a new rendering framework to render photorealistic view-dependent and relighted images from a dense set of images with the pre-acquired geometric model constructed through the geomet-

ric pipeline described in Chapter 2. The method aligns and pastes color images of the object onto the 3D surface of the object model, and then compresses those appearances in the 2D coordinate system defined on the 3D model surface. This compression is accomplished in an eigenstructure decomposition framework. The synthesis process is achieved by using the inverse transformation of the eigenstructure decomposition. Thanks to the linearity of image brightness, virtual images under a complicated illumination condition can be generated by summation of component virtual images sampled under single illuminations. When superimposing virtual objects onto real scenes, the geometric consistency between the virtual object and the lighting distribution has to be maintained precisely. As we employ a geometric model to represent the virtual object, if we know the real illumination condition, we can precisely render the cast shadows and also the appearance by decomposing the real illumination distribution into a set of point light sources. As well as the framework, experimental results will be presented.

In Chapter 4, we present a framework to accomplish view-dependent rendering and relighting from a sparse set of images, with the leverage of geometric information. Our goal is to accomplish photorealistic rendering, with an efficient representation of the data, from a casually acquired image sequence as the input. By casual acquisition we mean a relatively sparse set of images. In this case, the input images are not enough to enable us to simply apply either image-based rendering methods or the framework we present in Chapter 3. As well as the geometric model of the target object, we assume that the 3D model and the camera parameters are known, while we do not rely on any direct information of the light sources, i.e., the radiance and the positions. To solve this problem, we first separate the view-dependent (specular reflection) and view-independent components (diffuse reflection) of the surface reflectance. The result of diffuse reflection on the object surface is stored as one texture map for the whole object. Then we use the specular reflection images to derive an efficient representation of the lighting environment. We accomplish this by initially shooting back the pixels in the specular reflection images along the perfect mirror direction to form an *illumination hemisphere*. Using this illumination hemisphere as the initial estimation of the lighting environment, we estimate both the lighting environment and the surface properties using a reflection model to approximate the

specular reflectance mechanism. The estimation procedure can be considered as a blind deconvolution problem, factoring out the lighting and the reflectance from observation. As a result, we can represent a virtual object efficiently with reflectance parameters, a 3D model and an illumination hemisphere, providing the ability to accomplish rendering from arbitrary viewpoints and under different lighting environment from the original scene. As well as the framework, we present experimental results of applying the method to real world objects.

Finally, in Chapter 5, we summarize our contributions and discuss several future works.

# Chapter 2

## Geometric Modeling

### 2.1 Introduction

Geometric modeling of real world objects is fundamental for many applications that rely on accurate shape data, e.g., object recognition, localization, tracking, metamorphism and so on. Also, when analyzing the appearance of real world objects, knowledge of the geometry of the target object provides us a great deal of information. Although pure image-based rendering techniques that do not assume any *a priori* information of the scene geometry enables feasible view-dependent rendering of a real world scene, it is still advantageous to have the geometric information on hand. For instance, integrating synthetic objects into real world scenes with correct geometrical visuality, e.g., attaching cast shadows, can be achieved much easier with the help of geometric information. More importantly, the geometric information brings about a compact representation of the target object for view-dependent rendering and relighting, as we will describe in the next chapter.

The geometric modeling pipeline of real world objects can be depicted as in Figure 2.1. The pipeline consists of three steps, namely range image acquisition, registration (alignment) and integration (merging). We will not consider other post-processing such as hole-filling that may follow the pipeline or may be embedded in one of those components.

There are many methods to capture the three-dimensional information of a real object; these may be roughly categorized by their approaches: active or passive. Active methods directly interact with the target object or scene to obtain

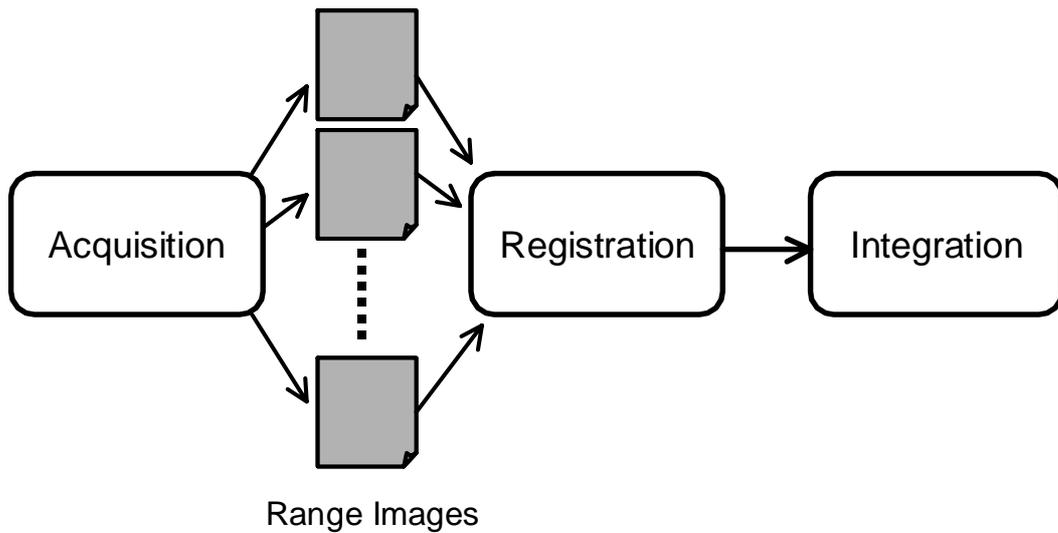


Figure 2.1: Geometric modeling pipeline.

its geometric information. The interaction can be based on casting laser ray, laser stripes, light stripe patterns, multiple lights, etc. The actual distance measurement can be accomplished by the time-of-flight of laser, triangulation between the casted laser stripes or the spatially coded scene (time-line based patterns [102], moire patterns and so on), and its observation through an ordinary camera, shading due to multiple light sources, etc. On the other hand, passive methods recover the three-dimensional information from raw observation of the scene. Shape-from- $X$ , where  $X$  can be shading, texture, highlights, etc., stereo including binocular, trinocular and multi-baseline stereos and depth-from-focus are representative passive range scanning methods. Whichever method we use, unless the target object is geometrically simple enough to be captured from one viewpoint, we need to range scan from multiple viewpoints. This creates the necessity for range image registration and integration.

Registration of multiple range images is a procedure for gathering range images scattered in different coordinate systems, each defined by the viewpoint of the range finder when scanned, to sit in one common coordinate system. Many approaches to conduct this have been proposed in the computer vision and computer graphics literature; most of them are designed to solve a least-square minimization problem set up to minimize the distance of correspondences in different range images. The correspondences can be between raw 3D points in point

cloud range images or feature points with characteristics of the local surface assigned to them. The most important problem in registration is how to deal with the noise embedded in the range images. Whichever range scanner we use, the range image obtained from it includes some noise due to, for instance, specular reflection with laser range scanners, folding effects on occluding boundaries with light stripe range scanner. Thus, to obtain accurate geometric models, it becomes crucial to make the registration procedure as robust as possible against noise.

Integration of multiple range images refers to the process of merging pre-registered range images to compose one complete geometric model, usually a polygonal mesh model. The procedure can be considered as extracting one surface from multiple overlapped surfaces. As in the registration procedure, it is important to make the integration framework robust against noise, which can be in the scanned range images and can also be inherited from the registration procedure. Also, depending on the use of the resulting mesh model, additional constraints such as mesh simplification and de-noising can be imposed when outputting the final complete mesh model, rather than just stitching together pre-aligned multiple range images.

In the next two sections, we propose a robust simultaneous registration method to accurately align multiple range images obtained through a range scanner, and a robust and adaptive range image integration method that can produce a complete mesh model in adaptive resolution depending on its local geometric or/and photometric characteristics.

With regard to registration, we propose a framework to register multiple range images with emphasis placed on robustness against noise. Taking the point cloud images obtained through a range sensor, e.g., a laser range scanner [20, 19, 72], or a light-stripe range finder [102], as the input, we simultaneously register all range images to sit in one common coordinate system. We highly prioritize our efforts to make the resulting registered geometric model accurate as opposed to making the whole procedure computationally fast. For this reason, we design our registration procedure to be a simultaneous registration method based on an error metric computed from point-point distance including additional attributes in its metric. Also for robustness and efficiency, we adopt a conjugate gradient framework utilizing M-estimator to solve the least-square

problem of minimizing the total errors through registration. To accomplish the point-point correspondence search efficiently, we employ a k-d tree data structure to store the range images. The robustness and effectiveness are demonstrated by experimental results. Also, as an application for digital preservation of cultural heritage objects, we show results of applying our framework to register a large ancestral object.

For integration of multiple pre-registered range images, we propose a new method based on volumetric representation. Unlike other volume-based integration methods, ours adaptively subdivides voxels, depending on the curvature of the surface to be reconstructed, thereby providing efficient representation of the underlying geometry as well as efficient use of computational resources. In our range image merging framework, additional attributes such as color or laser reflectance strength can be taken into account as well as 3D geometric information. This ability allows us to generate 3D models while preserving sharp edges around texture boundaries, thereby providing a good basis for efficient rendering and texture mapping. The overall framework is designed to be robust against noise, taking consensus carefully in both geometry and color; it is suitable for 3D model reconstruction from noisy range image sequences. As well as describing the system, we present several results of applying our framework to real data.

## 2.2 Robust Simultaneous Registration of Multiple Range Images

### 2.2.1 Related Work

Past work on range image registration can be roughly classified with respect to the following three aspects.

**Strategy: simultaneous<sup>1</sup> or sequential** The basic strategy of registering multiple range images can be classified into two different approaches. The straightforward strategy is to focus on only two range images at a time and to register one range image to another [5, 120, 133]. After one range image pair is regis-

---

<sup>1</sup>Commonly referred to as “global registration” and “multi-view registration”, especially in the graphics community

tered, a new pair including either range image in the former pair, positioned in the resulting coordinate, is registered. This process is repeated until all range images are used. Since this sequential strategy requires only two range images for each registration stage, it can be implemented with less memory and the overall computational cost tends to be cheap. Also, the computational cost for each registration stage is not affected by the total number of range images to be registered consequently.

However, this straightforward strategy is well known to be less accurate. In each range image pair registration stage, some error is introduced due to data noise, etc. Since each range image is fixed in the resulting position for each registration stage, this unavoidable error will be propagated to later registration stages and will result in an intolerable error accumulation in the last range image position. Although the “gap” developed by this error accumulation can be small enough depending on the use of the resulting geometric model, it is much more preferable to avoid this theoretically, especially when the geometric model will be used as a basis for texture-mapping or appearance analysis and so on.

Simultaneous registration solves this error accumulation problem by aligning all range images at once [2, 3, 6, 15, 28, 52, 70, 84, 94]. This can be accomplished by defining an error minimization problem using an error metric common among all range images. This approach can diffusively distribute the registration error over all overlaps of each range image. The drawback is its large computational cost as compared with that of sequential approaches.

**Matching unit: features or points** On registering range images, usually the problem is redesigned as an error (distance) minimization problem. The basis of the error to be measured may be features derived from the range images or points consisting of the range data. Feature-based methods extract some signatures around 3D points, invariant to Euclidean transformation, in each target range image and make correspondences among those features [15, 53, 54]. Based on the assumption that all correspondences are matched correctly, the transformation for registration can be computed in a closed form manner. On the other hand, if the signatures computed from the range images do not provide enough information and the matching of them cannot be done correctly, the registration stage can fail miserably. Point-based methods directly use the 3D points in an iterative manner. The point mates, the point correspondences to compute

the error metric, are dynamically updated and several iterative steps are used to minimize the total error. One drawback of this point-based approach is that it requires an initial estimation of the rough transformation between the target range images, which is generally provided by human hand or interaction, while most feature-based approaches do not have this requirement.

**Error metric: point-point distance or point-plane distance** Originally, point-based approaches such as the ICP algorithm [5, 133] set the error metric basis on the Euclidean distance between two points that correspond to each other [29, 70]. However, since this error metric does not take the surface information into account, the point-based approaches based on point-point distance suffer from the inability to “slide” overlapping range images. An alternative to this distance metric is to use point-plane Euclidean distance, which can be computed by evaluating the distance between the point and its mate’s tangent plane [15, 84]. Embedding the surface information into the error metric in this way results in point-based approaches which utilize point-plane distance metric, and tend to be robust against local minima and converge rapidly. However, computing the point-plane distance is computationally expensive compared with point-point distance computation, since it requires a few additional computational steps after establishing the point-point correspondences. As work-arounds, methods using viewing direction, z-buffer, etc. to find the correspondence are proposed for efficiency [2, 6, 84].

### 2.2.2 Approach

Based on the consideration described above, we have designed a registration algorithm which i) is based on the simultaneous strategy, ii) uses points as matching units, iii) utilizes the point-point distance metric. The framework is inspired by the work of Wheeler et al. [125, 123], that applied similar techniques for 2D-3D and 3D-3D object recognition and localization.

We want to construct the geometric model as accurately as possible. For this reason, as a preliminary step, we attach more importance to robustness and accuracy than to computational expense in the registration method. This causes us to choose a simultaneous strategy, which is accurate in principle.

We employ points as matching units. Although the laser range scanner we

use is quite accurate, still the distance to the object is large and the measurement condition is poor in many cases. Because the scanned range images include noise, the information computed from them will be even more corrupted by that noise. Thus, we avoid using any secondary features derived from raw range data; instead, we directly use data points as matching units.

We use the point-point distance metric. Due to the noise problem, as mentioned above, we have to avoid obtaining secondary features (surface normals in this case) and thus, cannot use the point-plane metric that requires us to calculate surface normals. It is also true that point-point metric is less expensive in computational cost than the point-plane metric, and is preferable when the data set is very large.

The overall simultaneous registration framework can be described as an iteration of the following procedure until it converges:

#### Procedure *OneStepOfSimultaneousRegistration*

input Array RangeImages

Array KDTrees, Scenes, PointMates, Transforms

for each r in RangeImages

    KDTrees[r] ← BuildKDTree(RangeImages[r])

    for each s in RangeImages except r

        Scenes[s] ← RangeImages[s]

    end for

    for each i in Pointsof(RangeImages[r])

        for each s in Scenes

            PointMates[i] + ← CorrespondenceSearch(i, KDTree[s])

        end for

    end for

    Transforms[r] ← Transformation(PointMates)

end for

for each r in RangeImages

    Transform(r, Transform[r])

end for

We basically extend the framework of the pairwise ICP algorithm [5, 133] to

handle multiple range images simultaneously. This is achieved by setting up a global objective function to minimize with respect to each range images. Defining *model* as the particular range image of interest and *scene* as one of the range images in the rest of range images set, in one simultaneous registration loop, each range image becomes a *model* once. Point mate search (search for nearest neighbor point) for each point in the *model* will be done against all *scene* range images ( $M - 1$  if we have  $M$  range images), and they will be stored in an array. Rigid transformation for the current *model* will be computed in a conjugate gradient search framework utilizing M-estimator, and will be stored in an array. After each range image has become a *model* once, all range images are transformed using the transformation stored in the array. Note that each range image is not transformed immediately after its transformation is computed. Considering that each step transformation evaluated inside one step of simultaneous registration procedure will not be so large, this latency of transformation will not cause a problem. Furthermore, this timing of transformation saves us a large amount of computational time, since construction of k-d trees is required only once per range image in one simultaneous registration procedure. Furthermore, it allows parallel computation of transformation steps. Details will be discussed in the following sections.

### 2.2.3 Point Mate Search

#### K-D Tree

As we try to register range images that consist of a large amount of 3D points, finding correspondences for each point in each range image can easily dominate a critical portion of the overall computational time. To obtain point correspondences efficiently, we employ k-d tree structure to store the range images [33]. K-d tree's k-d abbreviates *k-dimensional* and it is a generalization of binary-search tree for efficient search in high dimension space. The k-d tree is created by recursively splitting a data set down the middle of its dimension of greatest variance. The splitting continues until the leaf nodes contain a small enough number of data points. For instance, Figure 2.2 shows how 2D data point set is split and separated into leaf nodes.

The k-d tree constructed will become a tree of depth  $O(\log N)$  where  $N$  is

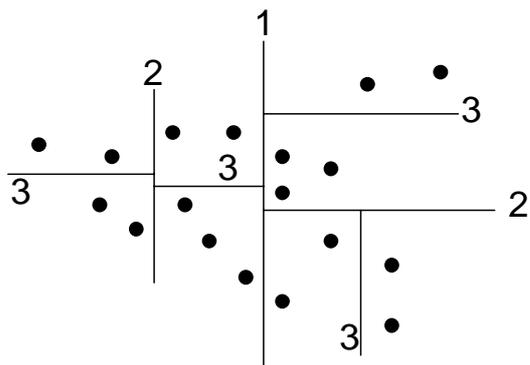


Figure 2.2: K-d tree subdivision of 2D points. The number indicates the level of the tree at which the split occurs.

the number of points stored. Nearest-neighbor search can be accomplished by following the appropriate branches of the tree until a leaf node is reached. A hyper-sphere centered at the key point with radius of the distance to the current closest point can be used to determine which, if any, neighboring leaf nodes in the k-d tree must be checked for closer points. Once we have tested all the data in leaf nodes which could possibly be closer, we are guaranteed to have found the closest point in the tree. Though its worst case complexity is  $O(N)$ , the expected number of operations for nearest-neighbor search is  $O(\log N)$ , which will be the case if the data is evenly distributed. For the cases of storing surfaces in 3D space in k-d tree, this even distribution assumption usually holds. The largest overhead involved in using k-d trees is that of construction: the k-d tree of range-image points must be built prior to the search. This operation costs  $O(N \log N)$ . To avoid making this computational expense critical, we update each range image position only once in one step of simultaneous registration procedure as listed in the pseudo code in Section 2.2.2, requiring only  $M$  times of k-d tree rebuilds in one global iteration where  $M$  is the number of range images.

### Distance Metric

To utilize nearest-neighbor search based on k-d tree structure, we need a measure of dissimilarity between a pair of points. The dissimilarity,  $\Delta$ , between k-d

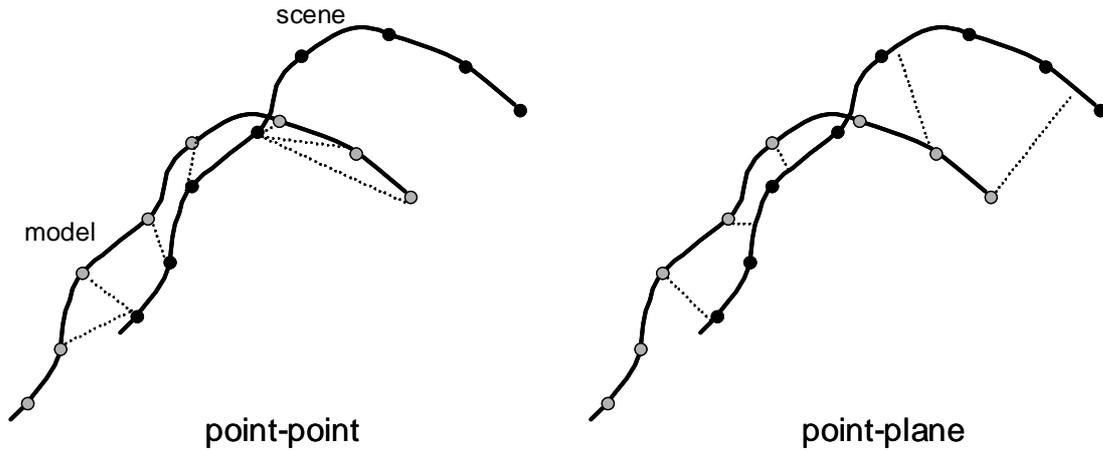


Figure 2.3: Point correspondences using point-point and point-plane distance metric.

points  $\mathbf{x}$  and  $\mathbf{y}$  must have the form

$$\Delta(\mathbf{x}, \mathbf{y}) = \mathbf{F}\left(\sum_{i=1}^k f_i(\mathbf{x}_i, \mathbf{y}_i)\right) \quad (2.1)$$

where the functions  $f_i$  are symmetrical functions over a single dimension and functions  $f_i$  and  $F$  are monotonic. All distances satisfy these conditions, including the Euclidean distance  $\|\mathbf{x} - \mathbf{y}\|$ . As mentioned in Section 2.2.1, using point-plane distance as the error metric provides faster convergence. However, the point-plane distance, which can be computed by

$$\Delta(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y}) \cdot \mathbf{N}_y \quad (2.2)$$

does not satisfy the monotonic condition. To take advantage of the efficiency of the k-d tree structure, we use the point-point Euclidean distance as the dissimilarity measure. Also, we prefer point-point distance for the sake of robustness; it enables us to avoid the usage of secondary information derived from raw data, such as surface normals in point-plane, which can be sensitive to noise in the raw data points.

Figure 2.3 depicts an example of point correspondences in the case of using point-point distance metric and point-plane distance metric. While the point-point distance metric searches for the nearest neighboring point, meaning establishing a discrete mapping of one surface to another, the point-plane distance



Figure 2.4: Images using laser reflectance strength as pixel values.

metric can be considered to be a way to find the continuous mapping of one surface to another. In cases like Figure 2.3, where the *model* surface has to be “slid” to fit the *scene* surface, the point-plane approach successfully finds the correspondences that enables us to compute the rigid transformation close to the sliding direction, while the point-point approach tends to get stuck in a local minima because of the inability to find point mates in the sliding direction. This sliding ability of point-plane approaches provides faster convergence compared with using point-point distance metric.

To compensate for the inability to slide with the point-point based distance measurement, we need some information to be attached to the 3D points; this information should suggest better matches. For this purpose, we use the laser reflectance strength (LRS) value when we use laser range finders; or we use the intensity value when we use a light-stripe range finder as a photometric attribute of each 3D point. Most laser range finders return the strength of the laser reflected at each surface point that is measured as an additional output value. Figure 2.4 shows two images with LRS values used as the pixel values. For better visualization, the images are histogram-equalized. As can be seen, the LRS values are mostly invariant against Euclidean transformation, since the dominant factor of the power of laser reflected at an object surface is its surface material. The same thing can be said about the intensity values when using the

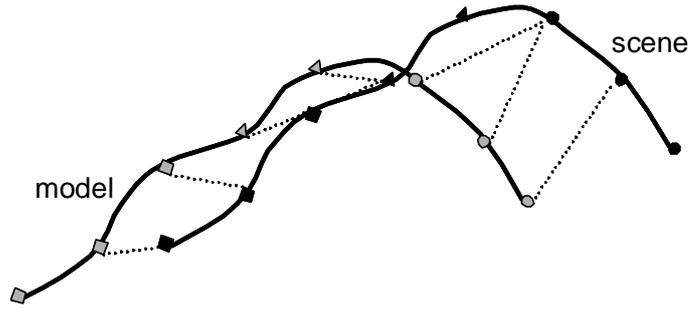


Figure 2.5: Point mates using point-point distance metric with reflectance strength values. Different shape marks indicate different reflectance strength values.

light stripe range finder under a fixed lighting environment, and when the object is mostly Lambertian. One common method to utilize two different sources of information in distance measurement, in this case the position distance and LRS distance, is to set up a combined metric, such as

$$\Delta(\mathbf{x}, \mathbf{y}) = [(\mathbf{x}_x - \mathbf{x}_y)^2 + (\mathbf{y}_x - \mathbf{y}_y)^2 + (\mathbf{z}_x - \mathbf{z}_y)^2 + \lambda(\mathbf{r}_x - \mathbf{r}_y)^2]^{\frac{1}{2}} \quad (2.3)$$

where  $r$  is the photometric attribute value and  $\lambda$  is a constant scalar. However, this scalar introduces a tedious and ad hoc effort in finding the “best”  $\lambda$ . Instead, we use the photometric attribute to determine the best pair among candidates of closest points. Namely, we first search for multiple ( $m$ ) closest points in the k-d tree, and then evaluate the photometric distance for each of them to obtain the closest point with respect to the photometric attribute. We gradually reduce the number of the candidates  $m$  along the iteration so as to make it inversely proportional to the number of iterations. This utilization of the photometric attribute is similar to [55], which uses color attribute to narrow down the closest point candidates. Figure 2.5 depicts how the point-point distance metric utilizing LRS values as additional attribute works in the example case depicted in Figure 2.3 ( $m = 4$  in this example).

### Speeding Up

Even though we employ k-d tree structure for efficient point correspondence search, when the number of points in the target range images gets large, the

computational cost becomes massive. In early stages of the simultaneous registration, when the range images are distributed apart, it is more important to get them close to each other than to accurately compute the rigid transformation for each registration step. To provide a way to speed up the registration, we subsample each range image to reduce the number of points used in the registration process. The points in each range image are given a sequential identification number  $m = 0, \dots, M - 1$  and a uniformly distributed random number within the interval  $[0, M - 1]$  is generated to pick up the points to be used. The seed number to generate the random numbers is common for all range images in one simultaneous registration procedure and updated once per each global registration step. In the current implementation, we let the user determine the percentile of points to be used in each range image interactively. In future implementations, this could be done automatically, first using a small percentage and gradually increasing it to reach one hundred percent.

As the range images are set to be still in one iteration of simultaneous registration, it is very easy to design the whole framework to run in a parallel manner. In our current implementation, constructing k-d trees and search points mates and computing transformation steps are done in threads, providing high scalability.

## 2.2.4 Least-square Minimization Strategy

### Representing Transformation

Given a set of corresponding points  $(x_i, y_i)$  where  $i = 0, \dots, N - 1$ , the registration problem is to compute the rigid transformation which registers the *model* points  $x_i$  with their corresponding *scene* points  $y_i$ . The rigid transformation can be specified by a pair of a  $3 \times 3$  rotation matrix  $\mathbf{R}$  and a 3D translation vector  $\mathbf{t}$ . When the corresponding points are aligned with one another,  $y_i$  can be written as

$$y_i = \mathbf{R}x_i + \mathbf{t}. \quad (2.4)$$

Since range data points are contaminated by noise, the range image registration problem can be described as a error minimization problem with the error

function as,

$$f(\mathbf{R}, \mathbf{t}) = \sum_i \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2 \quad (2.5)$$

to minimize with regard to  $(\mathbf{R}, \mathbf{t})$ . As mentioned in Section 2.2.2,  $i$  will stand for all point mates established from all pairs of range images (if there are  $M$  range images,  $i$  will include all point mates from  $M \times (M - 1)$  range image pairs). Although it is convenient for vector computation to represent the rotation as a  $3 \times 3$  matrix  $\mathbf{R}$ ,  $\mathbf{R}$  will be constrained in a non-linear way as follows ( $T$  stands for transpose).

$$\begin{aligned} \mathbf{R}\mathbf{R}^T &= \mathbf{I} \\ |\mathbf{R}| &= 1 \end{aligned}$$

It is difficult to take advantage of the linear matrix representation of rotation while satisfying these constraints. For this reason, we will use the quaternion representation for rotation, a well known solution to this rotation problem. The benefits of using quaternion will be described later. Thus, the position parameters of each range image and the rigid transformation to register all of them will be represented with seven element vectors as follows.

$$\begin{aligned} \mathbf{p} &= [\mathbf{t}^T \ \mathbf{q}^T]^T \\ \text{where } \mathbf{q} &= [u \ v \ w \ s]^T \end{aligned} \quad (2.6)$$

### M-Estimator

As seen in Section 2.2.4, the registration problem can be described as a least-square minimization problem with the objective function Equation (2.5). Point correspondences are acquired using the techniques described in Section 2.2.3. In solving this error minimization problem, there are several points to take special care of. First, the early stages of registration will contain a lot of mismatches in correspondences, since range images will be positioned far apart. Second, even after several registration iterations, many mismatches can still be contained in the correspondences, due to noise contaminated data points, etc. The underlying problem here is how to robustly reject outliers. The following three representative classes of solutions can be found in the field of robust statistics.

The first class of solutions, outlier thresholding, is the simplest and most computationally cheap technique and thus the most common technique used in

vision applications. The basic idea is to estimate the standard deviation  $\sigma$  of the errors in the data and to eliminate data points which have errors larger than  $|k\sigma|$  where  $k$  is typically greater than, or equal to, 3. The problem with outlier thresholding is that a hard threshold is determined to eliminate the outliers. This means that, regardless of where the threshold is chosen, some number of valid data points will be classified as outliers and some number of true outliers will be classified as valid. In this sense, it is unlikely that a perfect method for selecting the threshold exists unless the outliers are all known a priori.

The second class of robust estimators is the median/rank estimation method. The basic idea is to select the median or  $k$ th value (for some percentile  $k$ ) with respect to the errors for each observation and to use that value as our error estimate. The logic behind this is that the median is almost guaranteed not to be an outlier as long as half of the data is valid. An example of median estimators is the least-median-of-squares method (LMedS) [37, 121]. LMedS computes the parameters of interest which minimize the median of the squared error computed from all data pairs using that parameter. Essentially, this requires an exhaustive search of possible values of the parameters by testing least-squares estimates using that parameter for all possible combinations of point correspondences. While these median-based techniques can be very robust, this exhaustive search remains a large drawback.

The third class of robust techniques is M-estimation [93, 37, 121], the one we use. The general form of M-estimators allows us to define a probability distribution which can be maximized by minimizing a function of the form,

$$E(z) = \sum_i \rho(z_i) \quad (2.7)$$

where  $\rho(z)$  is an arbitrary function of the errors  $z_i$  in the data set. The M-estimate is the maximum-likelihood estimate of the probability distribution  $P$  equivalent to  $E(z)$ . Least-squares estimation, such as minimizing Equation (2.5), corresponds to M-estimation with  $\rho(z) = z^2$ .

$$P(z) = e^{-E(z)} = e^{-\sum_i z_i^2} \quad (2.8)$$

We can find the parameters  $\mathbf{p}$  that minimize  $E$  by taking the derivative of  $E$  with

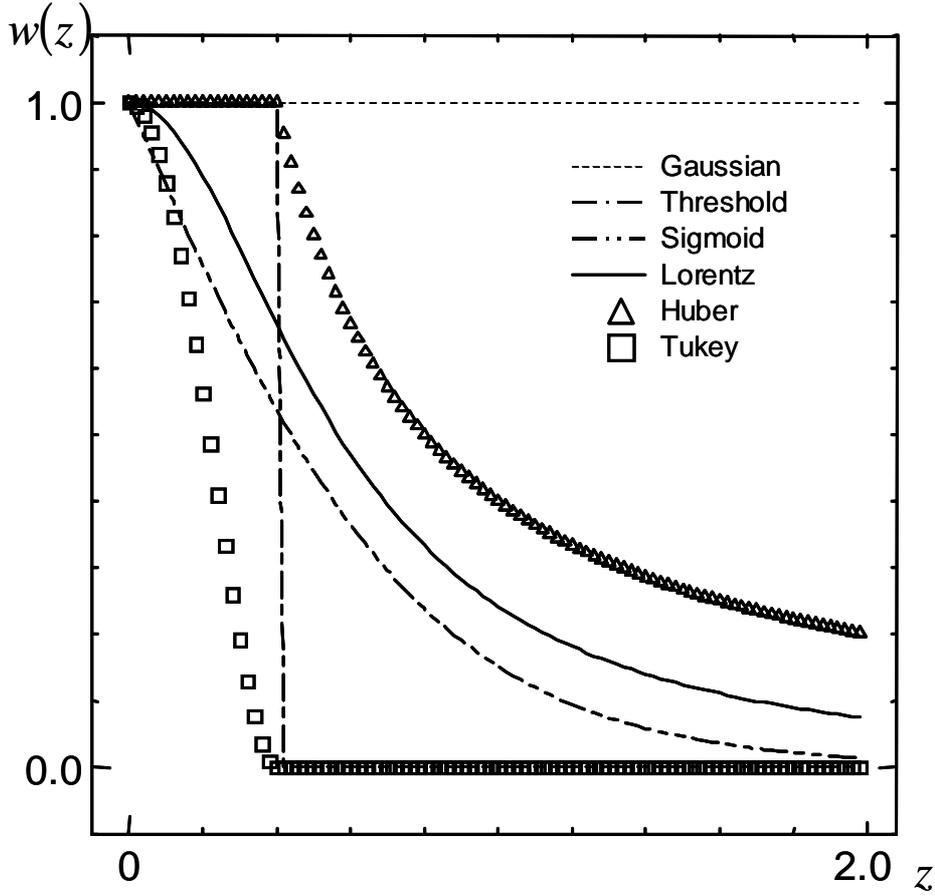


Figure 2.6: Weight values corresponding to several M-estimation functions.

respect to  $\mathbf{p}$  and setting it to 0.

$$\frac{\partial E}{\partial \mathbf{p}} = \sum_i \frac{\partial \rho}{\partial z_i} \cdot \frac{\partial z_i}{\partial \mathbf{p}} = \sum_i w(z_i) z_i \frac{\partial z_i}{\partial \mathbf{p}} = 0 \quad (2.9)$$

where  $w(z) = \frac{1}{z} \frac{\partial \rho}{\partial z}$ .

As can be seen in Equation (2.9), M-estimation can be interpreted as weighted-least square minimization with the weight function  $w$  being a function of data points  $z_i$ . Figure 2.6 shows the plots of the respective weight functions corresponding to several M-estimation functions. In our current implementation, we use Lorentz's function as the M-estimator; we found that it works best with our range image data.

## Putting It Together

Now, we can redefine our registration problem as follows. Given a set of corresponding points  $(\mathbf{x}_i, \mathbf{y}_i)$  ( $i=0, \dots, N-1$ ), we will minimize

$$E(\mathbf{p}) = \frac{1}{N} \sum_i^N \rho(z_i(\mathbf{p})) \quad (2.10)$$

$$\text{where } z_i(\mathbf{p}) = \|\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\| \quad (2.11)$$

$$\text{and } \rho(z_i) = \log\left(1 + \frac{1}{2}z_i^2\right). \quad (2.12)$$

The minimization of function  $E$  can be accomplished in a conjugate gradient search framework [37, 93]. Conjugate gradient search is a variation of gradient descent search, which constrains each gradient step to be conjugated to the former gradient step. This constraint avoids much of the zig-zagging that pure gradient descent will often suffer from, and consequently provides faster convergence. In applying conjugate gradient search to our minimization problem, we need to compute the gradient of function  $E$  with respect to pose parameter  $\mathbf{p}$  which can be described as Equation (2.9). For the following derivations, we redefine  $z_i$  to be

$$z_i(\mathbf{p}) = \|\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i\|^2. \quad (2.13)$$

A priori to the computation of the gradient, we pre-rotate the model points, so that the current quaternion is  $\mathbf{q}_I = [0 \ 0 \ 0 \ 1]^T$  which has the property that  $\mathbf{R}(\mathbf{q}_I) = \mathbf{I}$ . This allows us to take advantage of the fact that the gradient of  $\mathbf{R}(\mathbf{q})\mathbf{x}$  can easily be evaluated at  $\mathbf{q} = \mathbf{q}_I$ :

$$\frac{\partial(\mathbf{R}\mathbf{x})}{\partial\mathbf{q}}\mathbf{x} = 2\mathbf{C}(\mathbf{x})^T \quad (2.14)$$

where  $\mathbf{C}(\mathbf{x})$  is the  $3 \times 3$  skew-symmetric matrix of the vector  $\mathbf{x}$  which has a useful characteristic as follows.

$$\mathbf{C}(\mathbf{x})\mathbf{y} = \mathbf{x} \times \mathbf{y} \quad (2.15)$$

where  $\times$  is the cross product. With these facts,  $\frac{\partial z_i}{\partial \mathbf{p}}$  in Equation (2.9) can be de-

rived as

$$\begin{aligned}
\frac{\partial z_i}{\partial \mathbf{p}} &= 2(\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i) \frac{\partial(\mathbf{R}(\mathbf{q})\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i)}{\partial \mathbf{p}} \\
&= \begin{bmatrix} 2(\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i) \\ 4\mathbf{C}(\mathbf{x})^T(\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i) \end{bmatrix} \\
&= \begin{bmatrix} 2(\mathbf{x}_i + \mathbf{t} - \mathbf{y}_i) \\ 4\mathbf{x}_i \times (\mathbf{t} - \mathbf{y}_i). \end{bmatrix} \tag{2.16}
\end{aligned}$$

With the gradient computed in the above manner, line minimization is accomplished with golden section search. Line minimization methods using interpolation are not adopted, since it is easy to imagine the base function to be highly non-linear.

## 2.2.5 Results

### Photometric Attribute

To demonstrate that we can accomplish precise registration when an additional attribute which is almost Euclidean invariant is available, we prepared a range image sequence of a rotationally symmetric object that contains intensity values for each 3D point. By using a light stripe range finder [102], we obtained three sets of the range and color image pair of a CupNoodle, and assigned the intensity values ( $Y$  of  $YC_bC_r$  computed from  $RGB$  values of corresponding color images) to each 3D point in each range image. The lighting environment was static, and the object was roughly Lambertian. Note that, even if the object was not Lambertian and contained specularities, these could be successfully eliminated with the robust M-estimation framework. Range images were located as in Figure 2.7 by human interaction, and our registration method was applied by first setting  $m$  in Section 2.2.3 to 100 and gradually decreasing it to 1 as the procedure converged for each  $m$ . Figure 2.8 depicts the registration results. As can be seen in Figure 2.8, without the additional attribute, range images get stuck in a local minima as soon as they overlap and they do not slide around the symmetry axes; however, when using the photometric attribute, the images do slide.



Figure 2.7: Initial positions of the cupnoodle sequence.

### Noisy Range Images

To examine the robustness against noise, we applied the proposed method to a noisy range image sequence. By setting the threshold of the aforementioned light-stripe range finder to include quite an amount of background and not to eliminate ill triangle patches (triangle patches that have large aspect ratios), we obtained three range images of a ceramic cat, including a lot of noise. To compare the proposed method with the registration method proposed in [84], the range images were initially aligned with each other via human interaction as depicted in Figure 2.9. <sup>2</sup> After iterating both methods until convergence, we eliminated all 3D points and triangle patches that did not belong to the ceramic cat and measured the errors by using a point-plane distance metric. Table 2.1 shows the results and Figure 2.11 depicts the histograms of errors for both methods. Our method converged robustly, while the method of [84] converged into a local minima, leaving a gap as can be observed in Figure 2.10.

---

<sup>2</sup>With more rough initial hand alignment, the other registration method did not converge.

	Average Error	Max. Error	Min. Error
Our method	0.84	2.55	$5.35 \times 10^{-7}$
[84]	1.29	2.57	$3.21 \times 10^{-5}$

Table 2.1: Comparison of errors in mm.

### Cultural Heritage Preservation

We have applied the proposed method to register real data, the Great Buddha in Kamakura. The Great Buddha was scanned from fourteen different directions using Cyrax 2400<sup>3</sup>, a time-of-flight laser range scanner that can scan up to 100m with  $\pm 6$ mm error at 50m distance. Each point cloud image consists of approximately three to four million vertices. Since registering all range images with full resolution requires massive computational resources and time, we registered those range images in 1/25 resolution as a preliminary experiment.

First, the input range images were registered in a pairwise manner with occasional human interaction for initial alignment, and then registered simultaneously. The variance of Lorentz’s function was set to large in the beginning and then gradually decreased each time the registration procedure converged with a particular variance value. Rough initial pairwise alignment was accomplished with approximately five to ten iterations, and the final simultaneous registration was done with 25 iterations. Figure 2.12 depicts the M-estimator error for each iteration for the last 25 iterations. Since all range images are treated to be static inside each iteration, the M-estimator error does not always become smaller after each iteration. However, because the error is guaranteed to decrease inside each iteration, it is clear that the algorithm converges to a certain minimum, which is shown in the graph.

Figure 2.13 shows the resulting registered Great Buddha using different colors for each range image and Figure 2.14 shows the resulting Great Buddha using the same color for all range images.

---

<sup>3</sup>Cyra Technologies Inc.

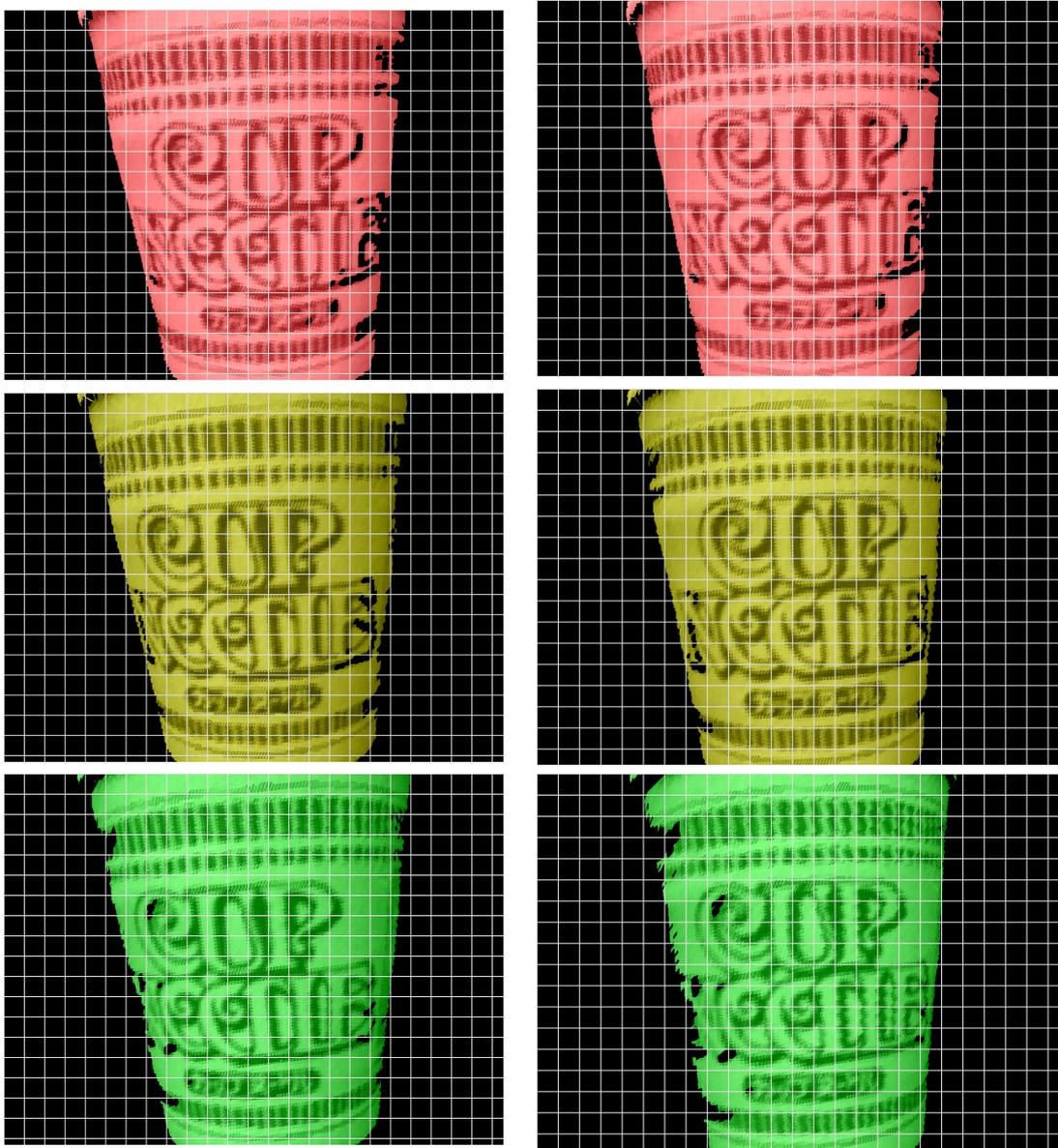


Figure 2.8: Left column: After registration without intensity attribute. Right column: After registration with intensity attribute. Each row shows a different range image. Grids are depicted for better visualization.

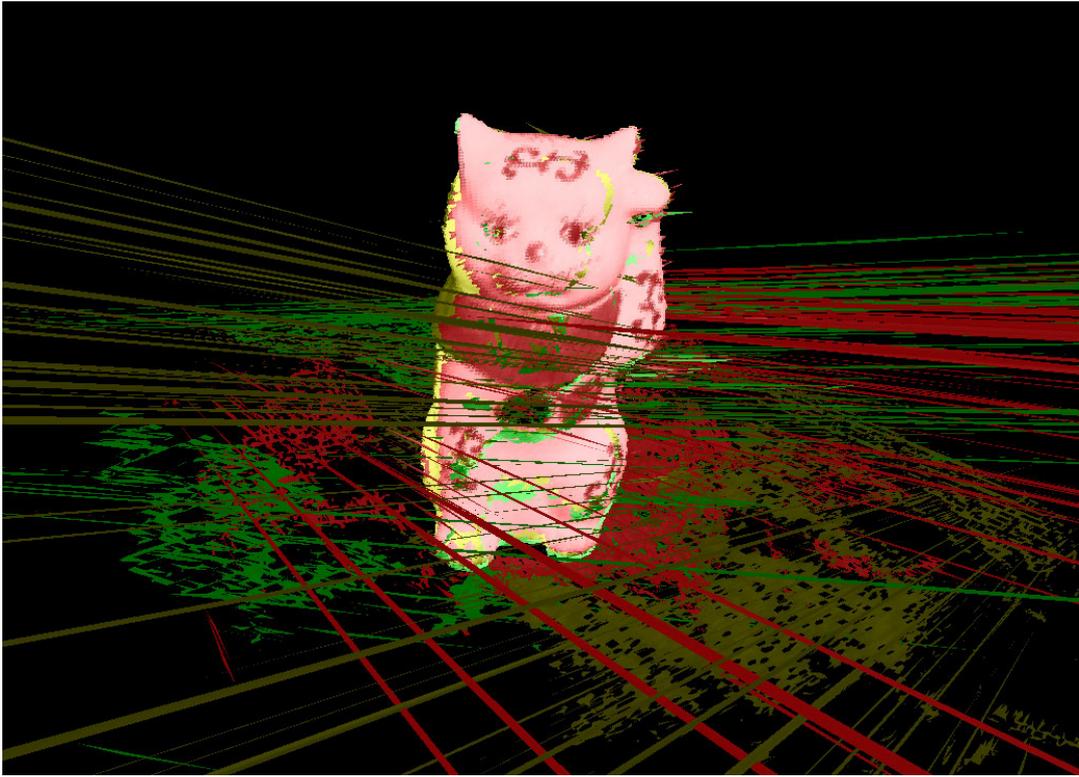


Figure 2.9: Initial positions of the Noisy Cat sequence.

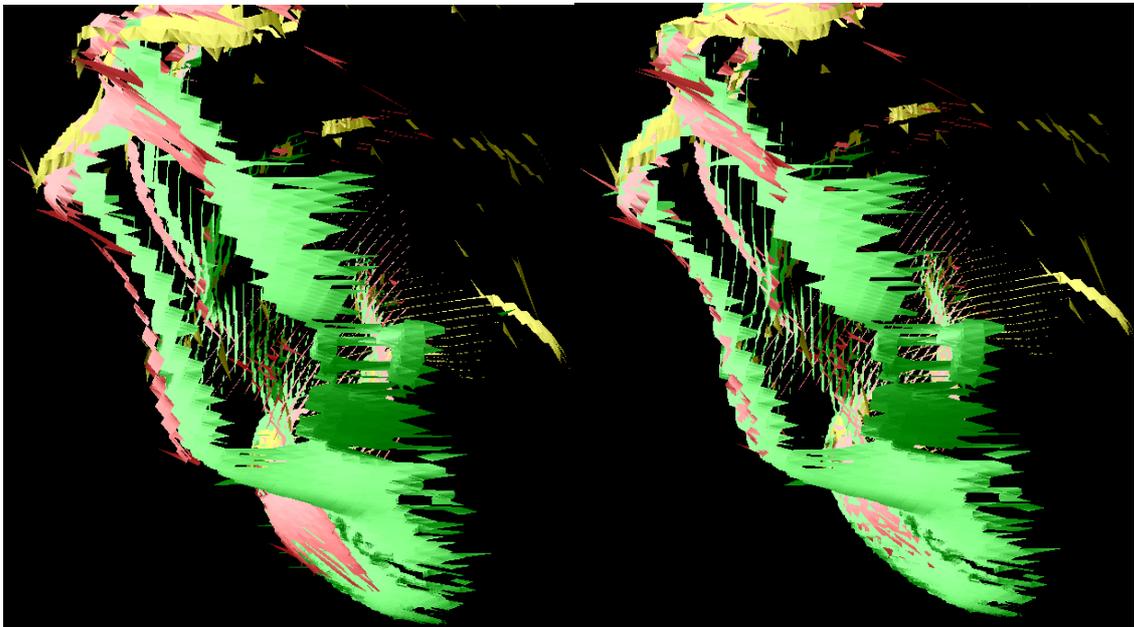


Figure 2.10: Left: Registered with [84] Right: Registered with our method.(Viewing from the top left of the cat.)

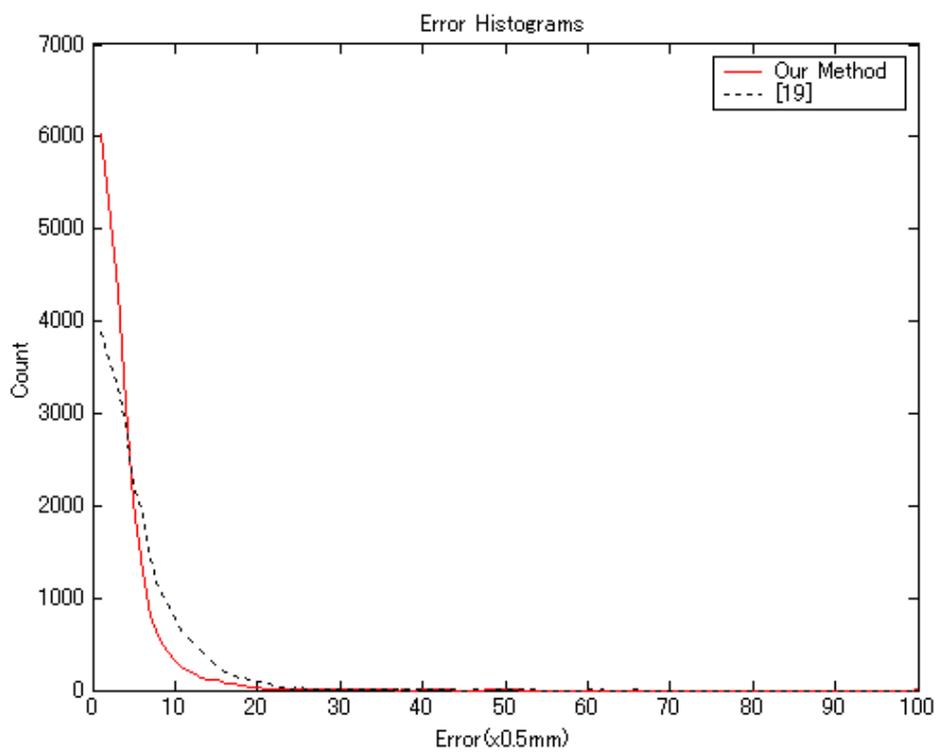


Figure 2.11: Histogram of errors.

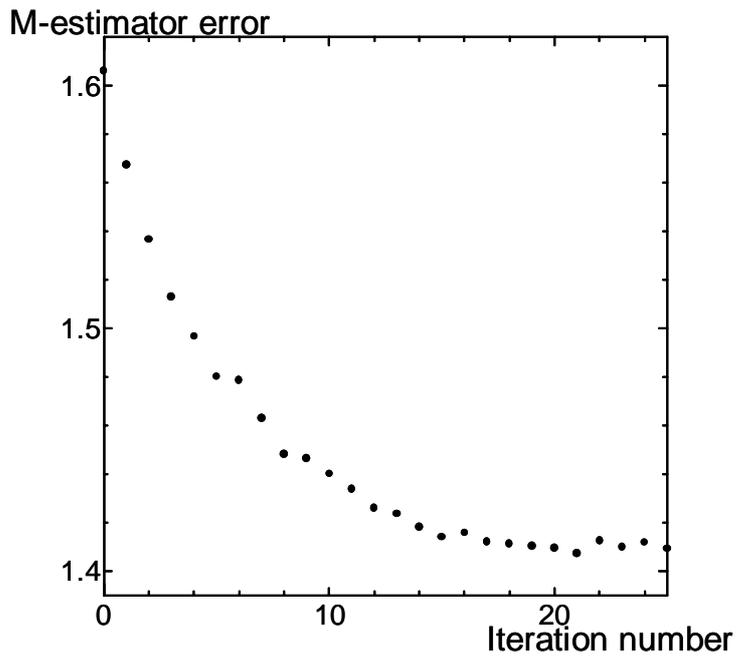


Figure 2.12: M-estimator error v.s. iteration number



Figure 2.13: Registered Great Buddha. Different color indicates different range image.

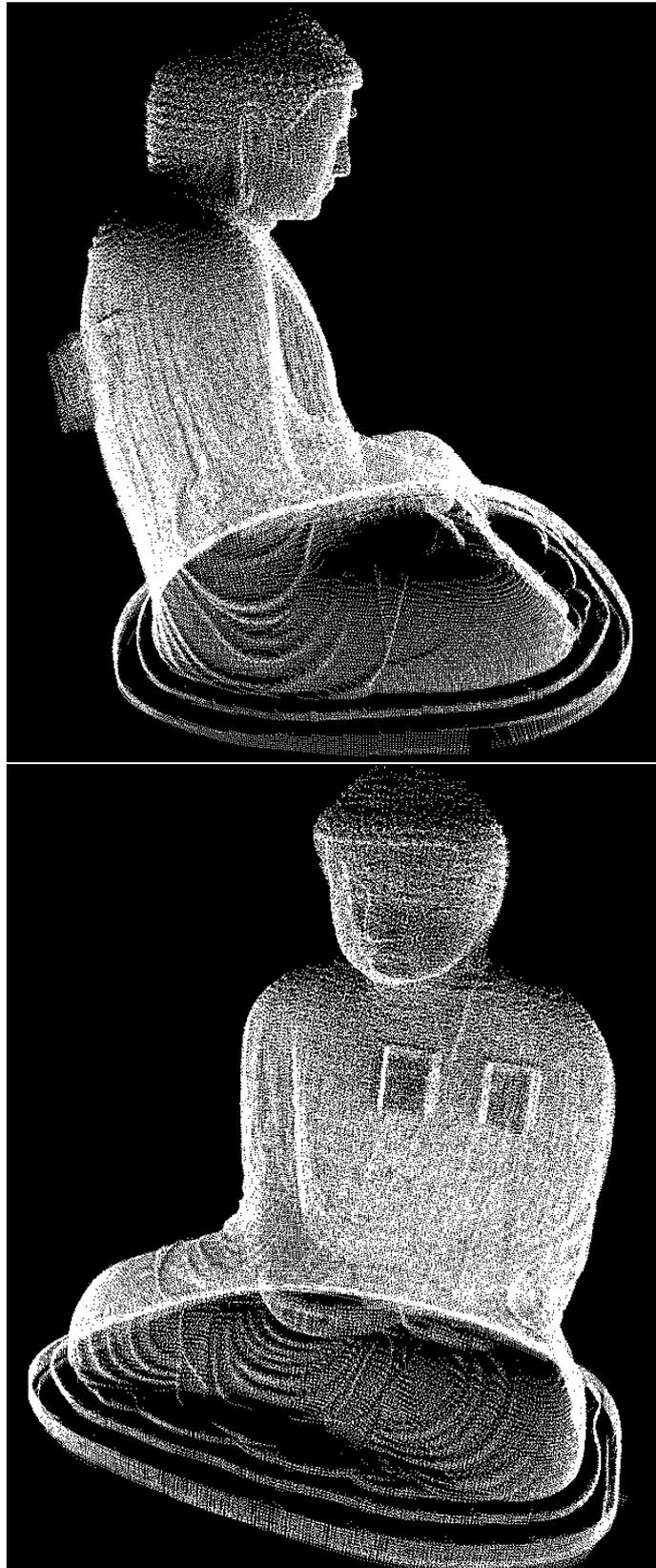


Figure 2.14: Registered Great Buddha.

## 2.3 Robust and Adaptive Integration of Multiple Range Images

### 2.3.1 Related Work and Overview

Several approaches have been proposed to integrate multiple pre-registered range images to obtain one complete geometric model from them. Turk and Levoy [120] proposed a method to “zipper” two range images at a time, by first removing overlapping portions of the meshes, next clipping one mesh against another, and then re-triangulating the mesh on the boundary. Although merging two range images is an intuitive process, pairwise merging does not work well when integrating multiple range images. Given a number of range images overlapping each other, a merging procedure which extracts the isosurface is necessary. Merging methods that make use of volumetric, implicit-surface representation and the marching-cubes algorithm [65] are suitable for this purpose. Hoppe et al. [44] construct 3D surface models by applying the marching-cubes algorithm to a discrete, implicit-surface function generated from a set of range images. After inferring local surface approximations from clouds of points based on tangent plane estimations, local search is accomplished to compute the signed distance from each voxel to the surface of the point set. Curless and Levoy [18] enhances Hoppe’s algorithm in a few significant points. However, none of these methods, including several others [41], compensate for noise or extraneous point data; the data is assumed to be part of the object, and noise is assumed to be negligible. Each of these methods suffers from inaccuracy due to their integration strategy, e.g., integrating unrelated observations, and these accuracy problems will affect the result even when the data is noise-free. Wheeler et al. [125, 124] addressed these important problems by designing a consensus surface algorithm. The consensus surface algorithm attempts to justify the selection of observations used to produce the average by finding a quorum or consensus of locally coherent observations. This process successfully eliminates many troublesome effects of noise and extraneous surface observations, and also provides desirable results with noise-free data.

Based on Wheeler’s algorithm to robustly integrate multiple range images, we propose a range image integration method which can construct 3D models with photometric attributes. Considering applications that utilize geometric

models, for instance, 3D object recognition and localization tasks, it is desirable to construct 3D models with additional attributes such as color and intensity. With the additional information provided by photometric attributes, higher accuracy and robustness can be expected from those applications. By taking a consensus of appearance changes of the target object from multiple range images, we reconstruct the 3D model with appearance values attached per vertex, successfully discarding outliers due to noise produced in the image-capturing process. This algorithm, taking consensus of photometric attributes, can also be used to derive the rigid part of the appearance change on the object surface, providing a 3D model with Lambertian reflected light values under a static illumination environment.

In some sense, our method is analogous to “Voxel Coloring” [17, 58, 107], where photometric consistency is used to carve a volume to reconstruct a geometric model with texture. Our framework goes the opposite way; we already have 3D information; from it, we pick out the consistent photometric attribute values per voxel to be attached to one geometric model.

We also propose an algorithm to construct the 3D model in an efficient representation. By taking the surface curvature into account when splitting the voxels recursively in an octree manner, the resulting 3D surface will be subdivided more in high curvature areas and less in surface areas that are near planar. Thus, the resulting geometric model will require fewer triangular patches to represent the object surface. Furthermore, by taking the photometric attributes into account, we can construct 3D models that have higher detail in surface areas that contain significant variation of appearance, providing an efficient basis for texture-mapping and shading. This is similar to research on mesh model simplification algorithms based on surface [36, 42, 43], while we reconstruct a simplified 3D model through a range image merging procedure based on implicit surface representation. The simplification is done when splitting voxels recursively, enabling better preservation of topology and mass of the object compared with results of other volume-based simplification methods [110, 111]. Frisken et al.[35] proposed adaptive sampling of distance field; however, their method does not produce a triangular mesh model. For converting the volumetric representation of the 3D model to a triangular-based mesh model, we propose an extended version of the marching-cube algorithm; this version handles voxels

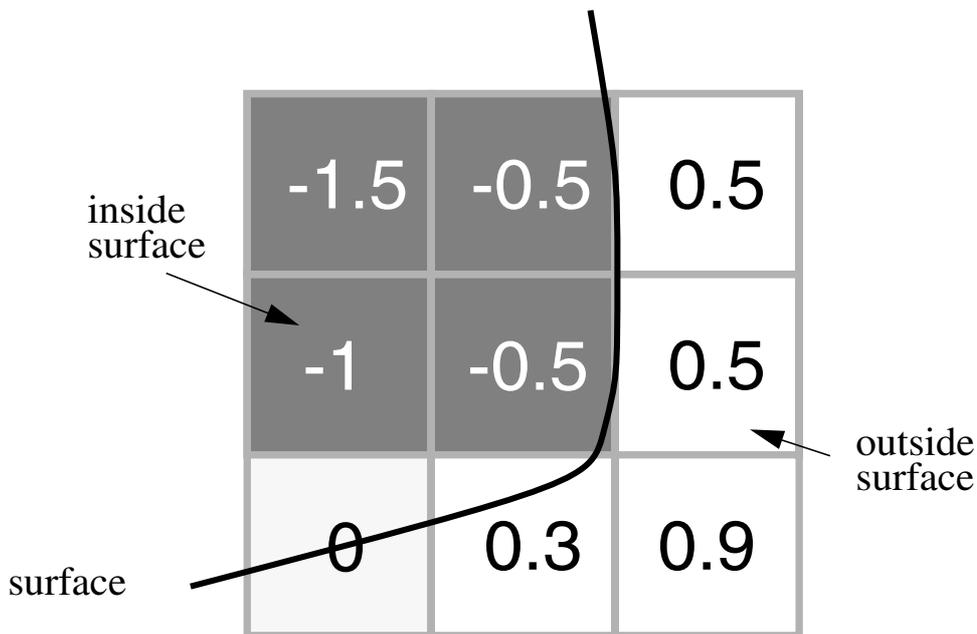


Figure 2.15: Zero-crossing interpolation from the grid sampling of an implicit surface

in different resolutions.

### 2.3.2 Volumetric Range Image Integration

In this section, we review the range image integration algorithm proposed by Wheeler [125, 124], which we will use as our basis.

In Wheeler's algorithm, all range images are first stored in a volumetric representation. The volume is split into grids, with each grid containing samples of an implicit surface. Namely, in each voxel, the signed distance  $f(\mathbf{x})$  from the center point of the voxel  $\mathbf{x}$  to the closest point of the object's surface. Positive values of  $f(\mathbf{x})$  indicate that the voxel lies outside the surface, while negative values indicate that it lies inside. The range image integration (view merging) problem can be interpreted as: given multiple samplings of this implicit function through multiple range images, extract the isosurface that is the zero crossing of this function (Figure 2.15). To solve this problem, we first have to consider how to compute  $f(\mathbf{x})$ , without knowing which, of many possible surfaces, is the exact surface.

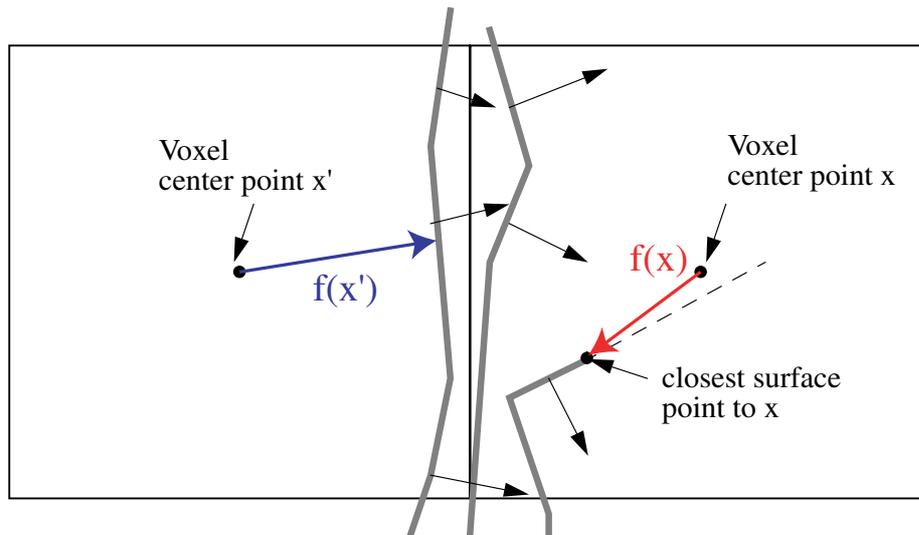


Figure 2.16: Naive algorithm: An example of inferring the incorrect sign of a voxel's value,  $f(x)$ , due to a single noisy triangle.

### Consensus Surface Algorithm

Wheeler et al. proposed the *consensus surface algorithm*, to compute the signed distance function  $f(x)$  for arbitrary points  $x$  when given  $N$  triangulated surface patches from various views of the object surface.

Previous naive algorithms define the magnitude of the implicit function  $|f(x)|$  of each voxel as the distance from the voxel center to the nearest triangle in all views (range images). However, this definition returns false values when range images contain errors. Figure 2.16 depicts this situation; the point chosen as the closest point from  $x$  does not belong to the real surface, and the naive algorithm incorrectly considers that  $x$  lies inside the object surface from noisy surface normal information.

This sensitivity to noise in range data can be solved by estimating the surface locally by averaging the observations of the same surface. Nearby observations are compared using their locations and surface normals, and if those values are within a predefined error tolerance, they can be considered to be observations of the same surface. The search for "nearby" observations can be accomplished using k-d trees [34] containing each range image separately. Given multiple surfaces that can be considered to be observations of the same surface, the con-

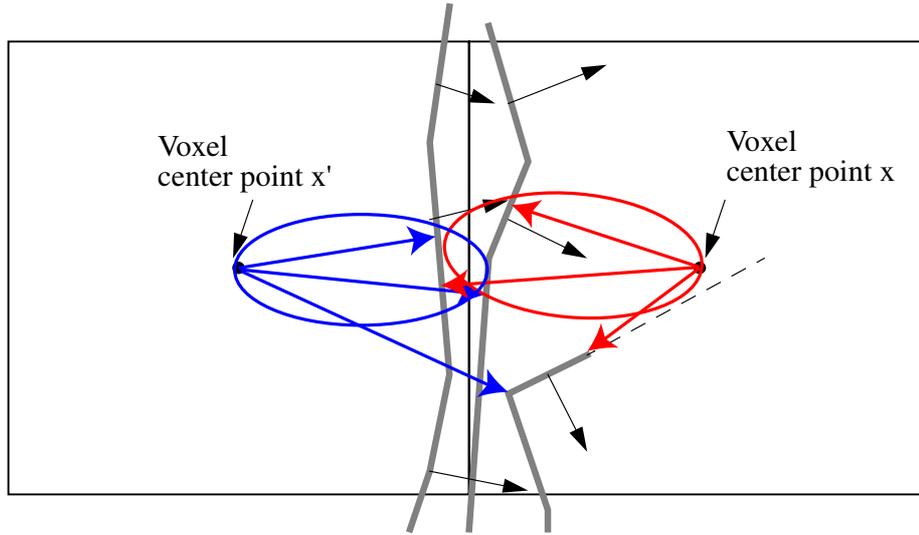


Figure 2.17: Consensus surface algorithm: The signed distance is chosen from the consensus surfaces inside the circle.

sensus surface algorithm examines the closest point in each image's triangle set to determine whether it can contribute to the consensus, by checking whether it is sufficiently close in terms of location and normal direction. A simple example when two range image surfaces are present is as follows.

$$\text{SameSurface}(\langle \mathbf{p}_0, \mathbf{n}_0 \rangle, \langle \mathbf{p}_1, \mathbf{n}_1 \rangle) = \begin{cases} \text{True} & (\| \mathbf{p}_0 - \mathbf{p}_1 \| \leq \delta_d) \wedge (\mathbf{n}_0 \cdot \mathbf{n}_1 \geq \cos \theta_n) \\ \text{False} & \text{otherwise} \end{cases} \quad (2.17)$$

where  $\delta_d$  is the maximum allowed distance and  $\theta_n$  is the maximum allowed difference in normal directions.

Range image surfaces that pass this check are considered to be the consensus surfaces, and the distance to the closest one of them is used as the signed distance (see Figure 2.17).

### Octree-based Voxel Splitting

To represent surface in volume, the volume has to be fine only around where the surface lies, instead of having fixed resolution [18]. In Wheeler's algorithm, this is accomplished by recursively splitting the voxels in an octree manner. Voxels containing zero crossing or neighboring zero crossing implicit function values

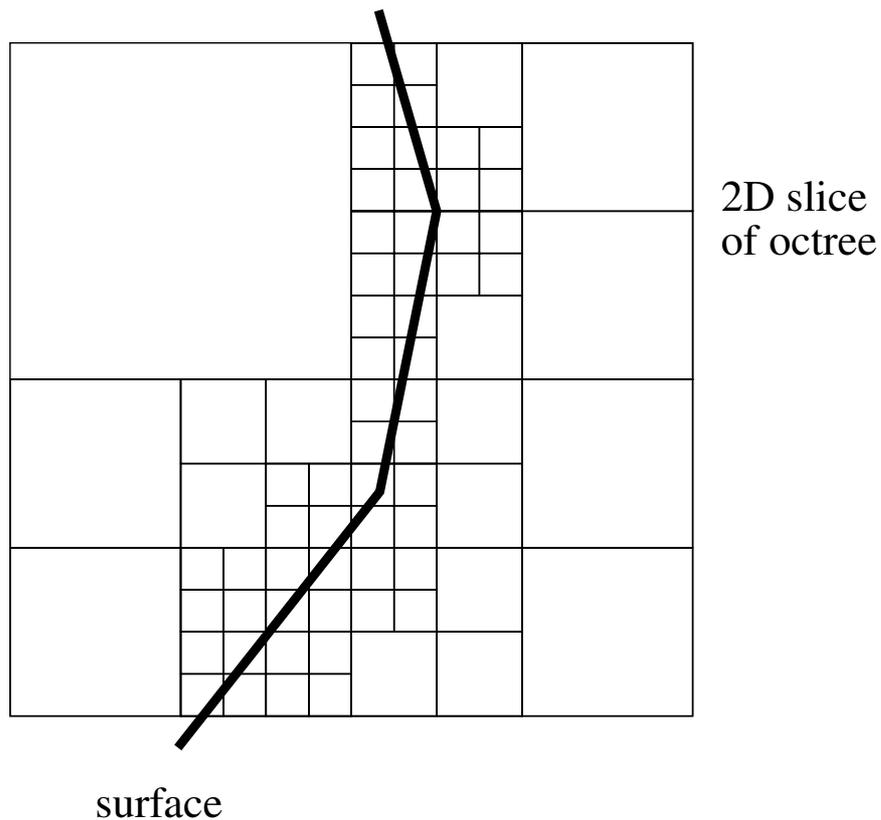


Figure 2.18: A 2D slice of the octree splitting volume. The resolution is high around the surface and low elsewhere.

are subdivided into the finest level by checking the magnitude of the signed distance<sup>4</sup>. As the resolution of voxels containing the resulting surface are the same (Figure 2.18), the marching-cube algorithm [65] can be applied straightforwardly to extract the final triangular-based mesh surface.

### 2.3.3 Consensus Surface with Photometric Attributes

In applications utilizing geometric models, e.g., 3D object recognition/localization, non-rigid appearance variation plays a crucial role in the accuracy and robustness. In particular, specular reflection causes the appearance to change non-rigidly, consequently making the whole process difficult. Thus, to date, most object recognition and object localization algorithms simply neglect specular reflection as outliers and assume Lambertian surfaces for target scenes and mod-

---

<sup>4</sup>Refer to [125, 124] for details.

els. To cooperate with this basic assumption on photometric properties, it is highly desirable to construct the geometric model to be used in such applications with Lambertian reflection property. If the 3D model is represented with photometric attributes that are rigid against changes in illumination and viewing directions, higher accuracy and robustness can be expected. Furthermore, if the illumination directions and viewing directions can be pre-estimated when processing recognition algorithms, non-rigid appearance variation such as specular reflection can be predicted and added to the 3D model appearance to further elevate the accuracy.

We accomplish this photometrically rigid 3D model construction in our range image integration framework. As examples of photometric attributes attached to range images, we consider two different attributes: laser reflectance strength and intensity/color.

As mentioned in Section 2.2.3, when using laser range finders to obtain the geometric information, laser reflectance strength (LRS) values can be obtained as additional attributes of each 3D point. As the laser can be considered as light with very narrow wavelength distribution, almost a single value, the behavior of the reflected laser on the target surface can be considered as same as the general light reflection. Namely, almost isotropic reflection analogous to diffuse reflection and sharp reflection distributed around the perfect mirror direction analogous to specular reflection occurs. Since the laser reflected in the perfect mirror direction will not be observed from the range finder direction, the portion of the laser that is reflected back can be considered to be caused by this diffusive reflection. As can be seen in Figure 2.4, although the LRS values are almost view-independent, they do vary slightly depending on the scanned direction [81], besides the occasional specular reflection.

To construct a 3D model with these LRS values attached to each vertex, we take the consensus of LRS values from different range images as well as the 3D information when merging multiple range images. In this LRS attribute case, the consensus can be obtained simply by taking the median of the LRS values from multiple range images inside each voxel and assigning it to each voxel center. As an example of applications utilizing these 3D models with LRS values attached, robust 2D-3D registration can be considered [57].

The color or intensity of target objects can be handled in the same manner.

Range images with color information attached per vertex can be obtained by using range finders that can acquire color images aligned to range images, e.g., light-stripe range finders, stereo systems, or by aligning 2D images taken separately to range images obtained by laser range finders [57, 114]. As is well-known, the color variation on the object surface is composed of two reflection components: the diffuse reflection and the specular reflection. While the diffuse reflection is almost independent of the viewing direction and its strength varies depending on the illumination direction, the specular reflection changes its strength drastically depending on the viewing direction and illumination direction. For simplicity, we consider a situation where the color images of the target object are taken under a static illumination environment with only the viewing direction varying while the object remains static. This assumption can be made naturally when scanning objects with laser range finders, especially when the target object is large, and also when scanning small objects to build 3D models for use in object recognition tasks in indoor scenes. As the illumination direction can be considered to be static for all color images, the intensity variation of each 3D point on the target object surface should have a DC component because of the invariant diffuse reflection with a sharp peak caused by specular reflection added to it, which can be observed from a narrow viewing direction. Thus, if each 3D point is observed from enough viewing directions, the histogram of the intensity values should have a sharp peak at the diffuse reflection value with some distribution around it due to image-capturing noise. Figure 2.19 depicts an example of this from real data. Based on this consideration, by taking the the median from multiple range images inside each voxel and then assigning it to each voxel, we can determine the color or intensity values to be attached to the resulting 3D model.

### 2.3.4 Integrating in Adaptive Resolution

The original consensus surface algorithm efficiently computed signed distances by utilizing an octree. However, it generated a mesh model in finest resolution everywhere. To reduce the amount of data required to represent the object and to use computational resources efficiently, we propose a method which generates a mesh model in adaptive resolution, with appropriate resolution according to



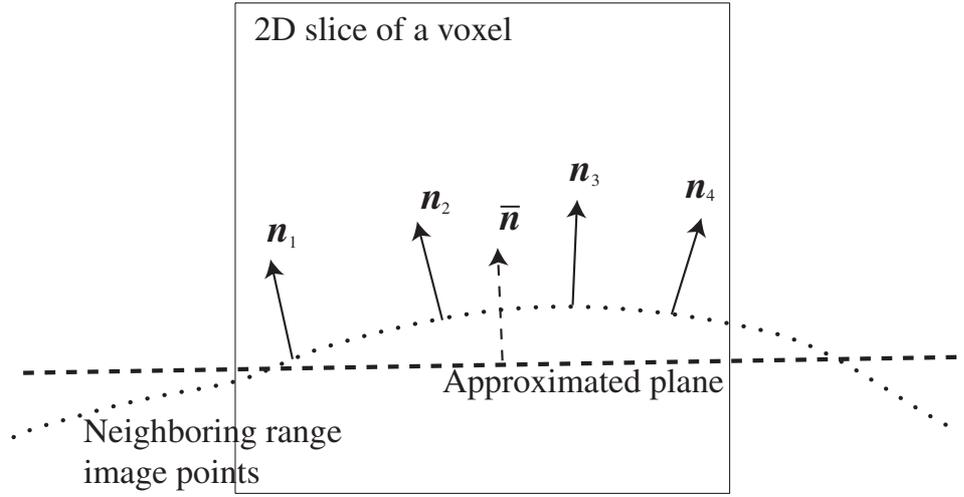


Figure 2.20: Our method approximates neighboring range images points to a plane and computes its normal vector  $\bar{n}$  by principal component analysis (PCA) for the cloud of range image points. The approximate normal is then used to determine further subdivision.

where  $\delta_n$  is the threshold of angle, the sampling interval is fine enough and no further voxel splitting is required.

To avoid erroneous subdivisions of voxels by the influence of noise included in each range image, our method takes a consensus between range images on decision of voxel subdivision. Now,  $N_n$  is the number of range images which satisfies Equation (2.18). Our method does not subdivide the voxel if

$$N_n > T_n, \quad (2.19)$$

where  $T_n$  is the threshold of consensus for normal vectors.

### Voxel Subdivision based on Photometric Attribute

Voxel subdivision based on variation of photometric attributes can be accomplished in a similar manner. When the voxel is subdivided depending on geometric attributes only or without consideration of any attributes, the appearance of the resulting object will be significantly smoothed out. Since ordinary shaders such as smooth shading and phong shading will simply interpolate the intensity values attached to each vertex, this smoothing is unavoidable. However, if we can triangulate the 3D mesh model with regard to the appearance variation, i.e.,

fine around appearance boundaries and having each triangular patch contain almost the same texture color, simple shading will work dramatically well. Also, 3D models tessellated with regard to their texture variation are useful to accomplish further texture analysis and synthesis. For instance, view-dependent texture mapping such as the one we explain in the next chapter can achieve higher compression, since global texture compression stacking triangular patches with similar texture can be applied.

In a similar manner to subdividing by the curvature of the surface, our method computes the variation of photometric attributes of 3D points inside the voxel of interest. Now,  $c_i, c_j$  are the photometric attributes of neighbor points included in a range image. If the maximum difference satisfies

$$\max_{i,j}(\text{Distance}(c_i, c_j)) < \delta_c, \quad (2.20)$$

where  $\delta_c$  is the threshold and  $\text{Distance}(c_i, c_j)$  is the function which computes the difference of two photometric attributes, the sampling interval is fine enough for the range image.

Our method also takes a consensus while considering photometric attributes. Similar to Equation (2.19), our method does not subdivide the voxel if

$$N_c > T_c, \quad (2.21)$$

where  $N_c$  is the number of range images which satisfies Equation (2.20) and  $T_n$  is the threshold of consensus for photometric attributes.

### Marching Cubes for Adaptive Octree

The original marching cubes algorithm can be applied only to voxels that have the same resolution (size). We extend the algorithm for triangulation of voxels in adaptive resolution generated from our method.

For voxels that are surrounded by voxels with the same resolution, the vertices of a cube to march are the central points of 8 adjacent voxels. In a similar manner, voxels surrounded by different sized voxels will have a set of connected voxels in the form of a quadratic pyramid or other special forms to march. Figure 2.21 shows the edges connecting adjacent voxels in an adaptive octree. Since these forms can be considered to be degenerated and transformed cubes, the original marching cube algorithm can be applied without modification.

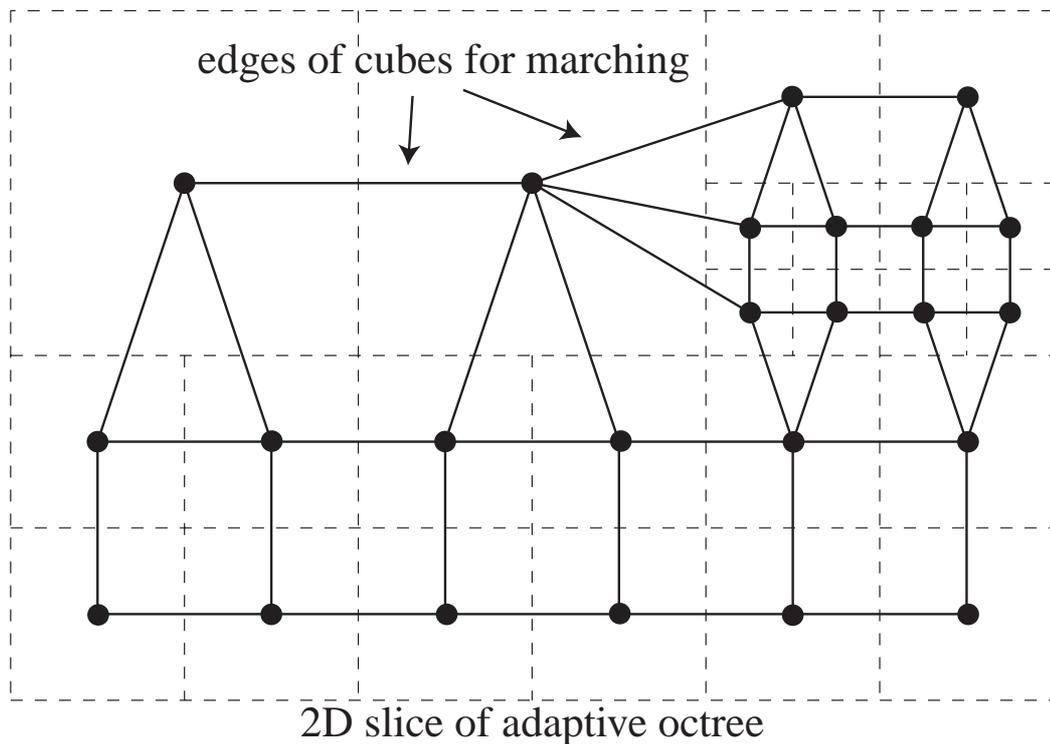


Figure 2.21: Edges connecting adjacent voxels in an adaptive octree.

### 2.3.5 Results

We have built a PC cluster that consists of eight PCs, each equipped with dual PentiumIII 800MHz processors with 1GB memory, connected by 100BASE-TX Ethernet. As consensus surfaces can be computed independently requiring only adjacent voxels, we parallelize the whole process by splitting the whole volume into pieces. With this parallel implementation, we are able to handle a huge amount of range image data.

First, we observe the shape and intensity of a box using the light-stripe range finder [102]. We acquire 60 range and intensity images from various viewpoints by putting the box on a turntable and rotating it six degrees for each step. Since range and intensity images are already aligned, we can attach intensity values to corresponding 3D points as a photometric attribute. Specular reflection is observed in some images (see Figure 2.22).

Figure 2.23 shows the merging result of the box. We experimented with our algorithm in two cases: merging with adaptive subdivision based only on sur-

	Number of points	Time for Integration	Mean Error
(A)	3.0 million	252 min.	N/A
(B)	1.2 million	82 min.	0.99 mm
(C)	1.4 million	88 min.	0.44 mm

Table 2.2: Statistics of Models of the Buddha: Our method reduces the amount of data and computation time. However, the mean errors are quite small compared with the Buddha size.

face curvature and based on both surface curvature and intensity. As can be seen in Figure 2.23, although specular reflections appear in some input images, they are not observed in the resulting model.

When subdividing voxels only by the curvature of the surface, the sampling becomes coarse in the planar area. However, since the intensity varies drastically around the character edges, the characters are smoothed out and deformed when rendered with simple smooth shading. On the other hand, when voxel subdivision is accomplished based on both curvature and intensity, the consensus surface sampling becomes fine around the character edges. Because of this, the sharpness of intensity edges is well preserved and the shapes of the characters are well rendered.

Next, we applied our algorithm to the Great Buddha in Kamakura, whose height is about 13 meters. Sixteen range images with LRS values attached to each 3D point were acquired by Cyrax 2400 [20]. Figure 2.24 shows 3 different results of our method. Their statistics are described in Table 2.2. We compared the size and accuracy of the models between A and B, A and C, using Metro [16].

In a comparison of the images rendered with LRS values, the appearance of (B3) is smoothed out compared with (A3). On the other hand, the sharp edges due to variation in LRS values of (C3) are well preserved while the amount of the data is successfully reduced.

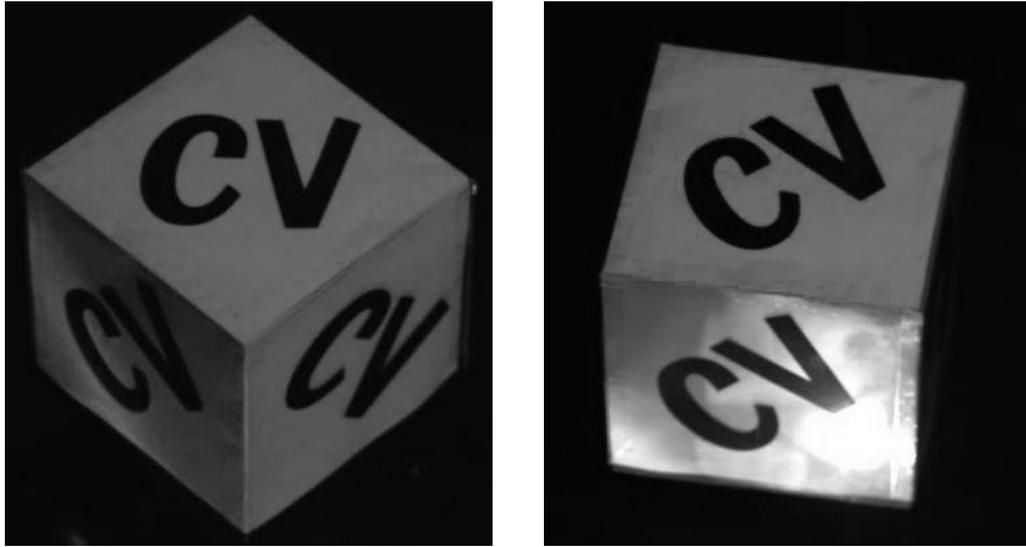


Figure 2.22: Range and intensity images of the box is acquired from various viewpoints using a turn table. These are two intensity images of sixty input images. Some images include specular reflection.

## 2.4 Summary

We have presented two methods for the geometric modeling pipeline: a robust simultaneous registration method and a robust and adaptive integration method taking photometric and geometric attributes into account.

For registration, we proposed a framework to simultaneously register multiple range images. The simultaneous registration problem is redefined as a least-square problem with an objective function globally constructed with respect to each range image. For efficiency, we employ k-d tree structure for fast point correspondence search and apply conjugate gradient search in minimizing the least-square problem for faster convergence. For robustness, we employ photometric attributes of raw 3D points, and search for “better” point mates based on their distance. Also, M-estimator is used for robust outlier rejection.

For integration, we proposed a range image integration framework which can construct 3D models with photometric attributes. By taking a consensus of appearance changes of the target object from multiple range images, we reconstruct the 3D model with an appearance which successfully discards outliers caused by noise. Also, we can provide a model with Lambertian reflected light

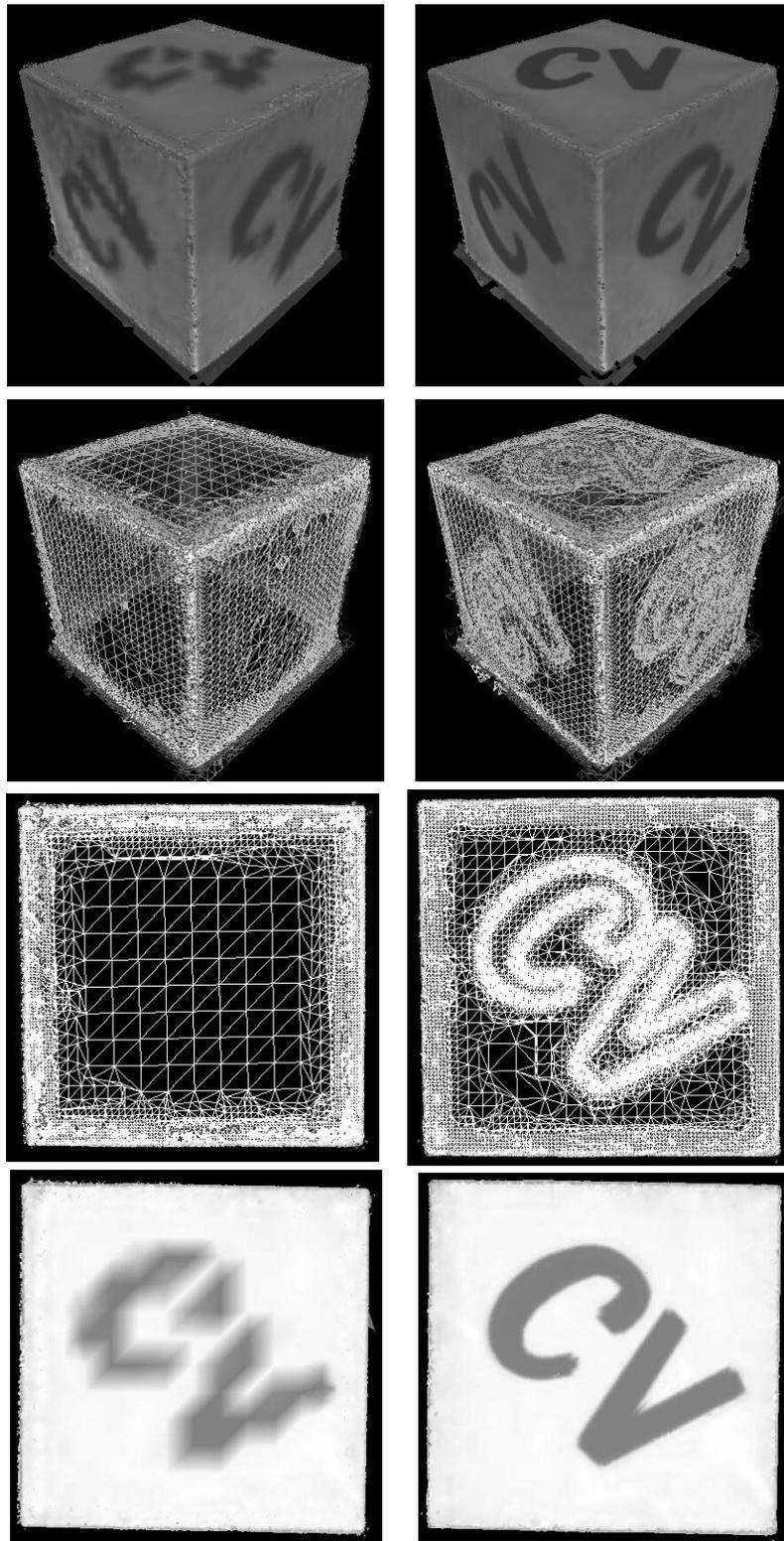


Figure 2.23: Merging results of a box with texture: the model in the left column is created by adaptive subdivision based on the curvature of the surface, and the one in the right column is created by adaptive subdivision based on both the surface curvature and the variation of intensity.

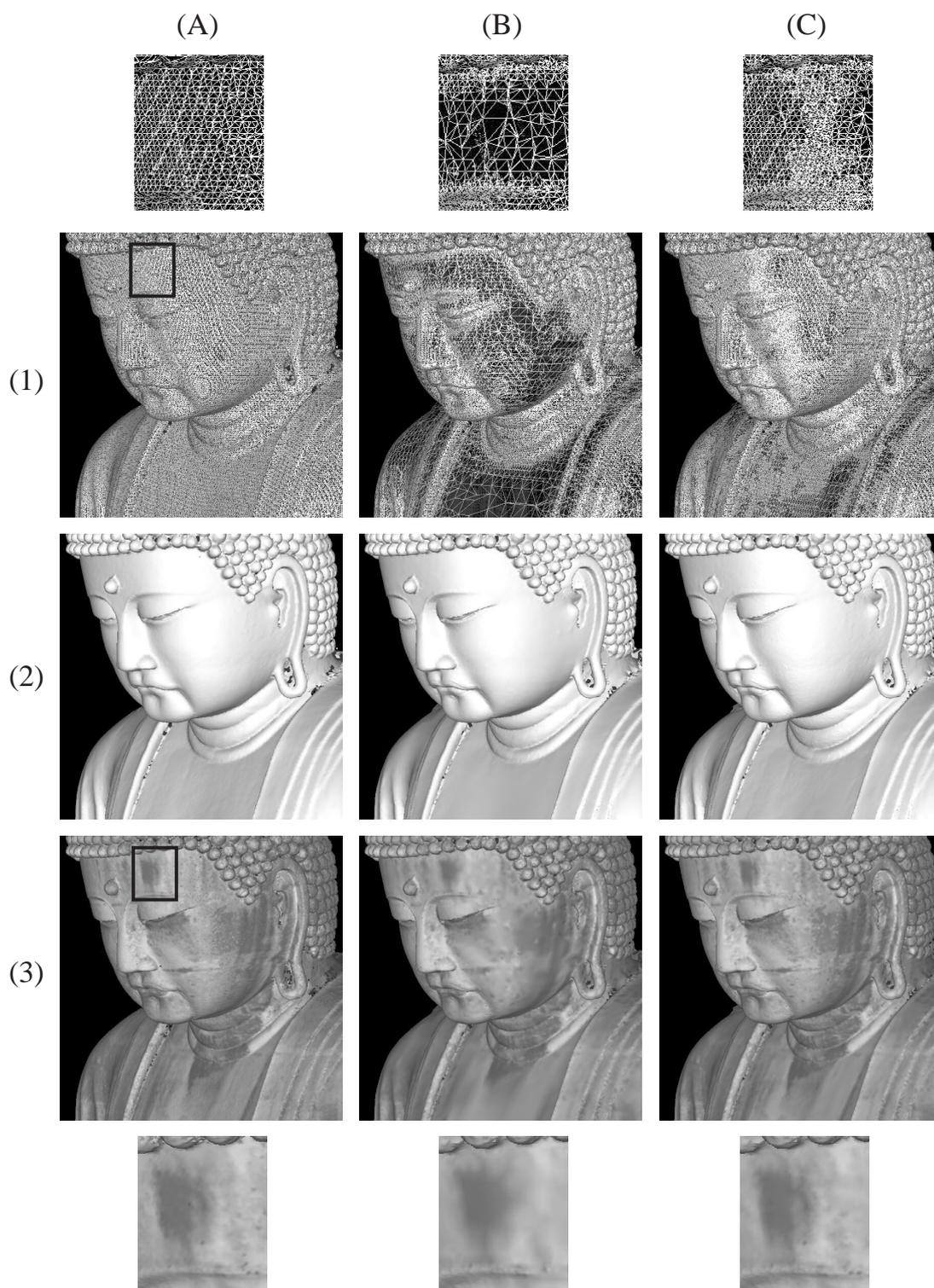


Figure 2.24: Merging results of the Kamakura Buddha: Column: (A) without adaptive subdivision, (B) with adaptive subdivision based on surface curvature, (C) with adaptive subdivision based on both surface curvature and LRS variation. Row: (1) wireframe model, (2) polygonal model, (3) images rendered with LRS values. The far upper and far lower rows are zoom-ups of the forehead of the Buddha.

values by discarding specular reflection as outliers. We also proposed an algorithm for constructing a 3D model in an efficient representation. Considering the surface curvature and the photometric attributes, we constructed 3D models that have higher detail in surface areas that contain either high curvature or significant variation of appearance. Thus, we could efficiently use computational resources.

Both methods take special care of their robustness against noise, by means of M-estimation in registration and consensus surface extraction in integration. Experimental results, including application for digital preservation of cultural heritage objects, prove the robustness and effectiveness of these methods.

Given an accurate geometric model through the geometric pipeline, we are ready to analyze the appearance of real world objects based on it. In the next two chapters, we present two frameworks for appearance analysis and synthesis, both of which take full advantage of the geometric model we have built.

# Chapter 3

## Eigen-Texture Rendering

### 3.1 Approach

In this chapter, we tackle the problem of rendering from a dense set of images. Given a 3D model of the target object and an image sequence of the object with varying viewing directions and light source directions, how can we efficiently represent the object so that it can be rendered from arbitrary viewpoints? Furthermore, how can we change the lighting conditions based on that representation?

As mentioned in Section 1.2, unlike pure image-based rendering methods, ours assumes that we know the geometry of the scene, i.e., the shape of the object and the viewpoint. Knowledge of the geometry of the object contributes much information toward making the representation compact. To take full advantage of the geometry of the object, we handle the appearance variation, caused by view direction and light source direction changes on the surface of the object. Note that, although it is the same 2D as the images in image-based rendering methods, it significantly reduces the redundancy to represent the appearance. In an image-based rendering point of view; this can be considered as pasting one of the two 2D slices to parameterize light rays on the object surface, more concretely, on the triangular patches of the mesh model of the object. The term “Surface Lightfields” introduced by Miller et al. [71] captures this idea well, although they use it in a different context.

Figure 3.1 displays an overview of the proposed method, which we will refer to as *Eigen-Texture Rendering* method. First, we create a 3D model of the target object from a sequence of range images. We use a light stripe range finder [102] to acquire the range images, and then the registration and integration of range images are done with the methods presented in Chapter 2. Next, we take a number of color images and align and paste them onto the surface of the object model. Then, the color images are converted into a new representation, *cell images*, and compressed by extracting the principle components of each cell image sequence via principle component analysis (PCA). The principle components of the cell images are stored with their coefficients for each viewpoint. View-dependent image synthesis can be accomplished by taking a linear combination of basis cell images per triangular patch. Virtual images under a complicated illumination condition can be generated by a summation of component virtual images sampled under single illuminations thanks to the linearity of image brightness.

## 3.2 Eigen-Texture

This section describes the theory of *Eigen-Texture Rendering*. The method samples a sequence of color and range images. Once these two sequences are input to the system, a 3D geometric model of the object is created from the sequence of range images via the geometric pipeline described in Chapter 2. Each color image is aligned with the 3D model of the object. In our system, this alignment is relatively simple because we use the same color charge coupled device (CCD) camera for taking both range and color images. Each color image is divided into small areas that correspond to triangular patches on the 3D model. Then the color images corresponding to triangular patches are warped to have a predetermined normalized triangle shape by bi-linearly interpolating the pixel values. We refer to this normalized triangular shape corresponding to each triangular patch as a *cell* and to its color image as a *cell image*. A sequence of cell images from the same cell is collected as shown in Figure 3.2. Here, this sequence depicts appearance variations on the same physical patch of the object under various viewing conditions. Principle component analysis (PCA) is applied for each cell image sequence to compress each cell image set separately. Note that the compression is done in a sequence of cell images. The appearance

changes in the cell image sequence are mainly due to the change of brightness, when the object surface is well approximated with the triangular patch based 3D model; some cell images, however, may have highlights passing across. Thus, high compression ratio can be expected with PCA. Furthermore, it is possible to interpolate appearances in the eigenspace.

Eigenspace compression on cell images can be achieved by performing the following steps. The color images are represented in  $RGB$  pixels, but the compression is accomplished in  $YC_rC_b$  using  $4 : 1 : 1$  subsampling, based on the accepted theory that human perception is less sensitive to slight color changes than to brightness changes. First, each cell image is converted into a  $1 \times 3N$  vector  $\mathbf{X}_m$  by arranging color values for each color band  $YC_rC_b$  in a raster scan manner (Equation (3.1)). Here,  $M$  is the total number of poses of the real object,  $N$  is the number of pixels in each cell image and  $m$  is the pose number.

$$\mathbf{X}_m = \begin{bmatrix} x_{m,1}^Y & \dots & x_{m,1}^{C_r} & \dots & x_{m,N}^{C_b} \end{bmatrix} \quad (3.1)$$

Then the sequence of cell images can be represented as a  $M \times 3N$  matrix as shown in Equation (3.2).

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1^T & \mathbf{X}_2^T & \dots & \mathbf{X}_M^T \end{bmatrix}^T \quad (3.2)$$

The average of all color values in the cell image set is subtracted from each element of matrix  $\mathbf{X}$ . This ensures that the eigen vector with the largest eigen value represents the dimension in eigenspace in which the variance of images is maximum in the correlation sense.

$$\mathbf{X}_{ave} = \begin{bmatrix} \dots & x_{1,N}^Y - E & \dots & x_{1,N}^{C_r} - E & \dots & x_{1,N}^{C_b} - E \\ \dots & \cdot & \dots & \cdot & \dots & \cdot \end{bmatrix} \quad (3.3)$$

$$\mathbf{X}_a = \mathbf{X} - \begin{bmatrix} E & \dots & E \\ \cdot & \dots & \cdot \\ E & \dots & E \end{bmatrix} \quad (3.4)$$

$$E = \frac{1}{3MN} \sum_{i=1, j=1, c \in \{Y, C_r, C_b\}}^{M N} x_{i,j}^c$$

With this  $M \times 3N$  matrix, we define a  $3N \times 3N$  matrix  $\mathbf{Q}$ , and determine eigen vectors  $\mathbf{e}_i$  and the corresponding eigen values  $\lambda_i$  of  $\mathbf{Q}$  by solving the eigenstructure decomposition problem.

$$\mathbf{Q} = \mathbf{X}_a^T \mathbf{X}_a \quad (3.5)$$

$$\lambda_i \mathbf{e}_i = \mathbf{Q} \mathbf{e}_i \quad (3.6)$$

At this point, the eigenspace of  $\mathbf{Q}$  is a high dimensional space, i.e.,  $3N$  dimensions. Although  $3N$  dimensions are necessary to represent each cell image in an exact manner, a small subset of them is sufficient to describe the principal characteristics as well as to reconstruct each cell image with adequate accuracy. Accordingly, we extract  $k$  ( $k \ll 3N$ ) eigen vectors which represent the original eigenspace adequately; by this process, we can substantially compress the image set. The  $k$  eigen vectors can be chosen by sorting the eigen vectors by the size of their corresponding eigen values, and then computing the eigen ratio (Equation (3.7)).

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^{3N} \lambda_i} \geq T \quad \text{where } T \leq 1 \quad (3.7)$$

Using the  $k$  eigen vectors  $\{\mathbf{e}_i \mid i = 1, 2, \dots, k\}$  (where  $\mathbf{e}_i$  is a  $3N \times 1$  vector) obtained by using the process above; each cell image can be projected onto the eigenspace composed by matrix  $\mathbf{Q}$  by projecting each matrix  $\mathbf{X}_a$ . And the projection of each cell image can be described as a  $M \times k$  matrix  $\mathbf{G}$ .

$$\mathbf{G} = \mathbf{X}_a \times \mathbf{V} \quad \text{where } \mathbf{V} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_k \end{bmatrix}$$

To put it concisely, the input color image sequence is converted to a set of cell image sequences, and each sequence of cell images is stored as the matrix  $\mathbf{V}$ , which is the subset of eigen vectors of  $\mathbf{Q}$ , and the matrix  $\mathbf{G}$ , which is the projection onto the eigenspace. As we described in Equation (3.1), each sequence of cell images corresponds to one  $M \times 3N$  matrix  $\mathbf{X}$ , and is stored as  $3N \times k$  matrix  $\mathbf{V}$  and  $M \times k$  matrix  $\mathbf{G}$ , so that the compression ratio becomes that described in Equation (3.8).

$$\text{compression ratio} = k \frac{M + 3N}{3MN} \quad (3.8)$$

Each synthesized cell image can be computed by Equation (3.9). A virtual object image of one particular pose (pose number  $m$ ) can be synthesized by aligning each corresponding cell appearance ( $R_m$ ) to the 3D model.

$$\mathbf{R}_m = \sum_{i=1}^k g_{m,i} \mathbf{e}_i^T + [E \ E \ \dots \ E] \quad (3.9)$$

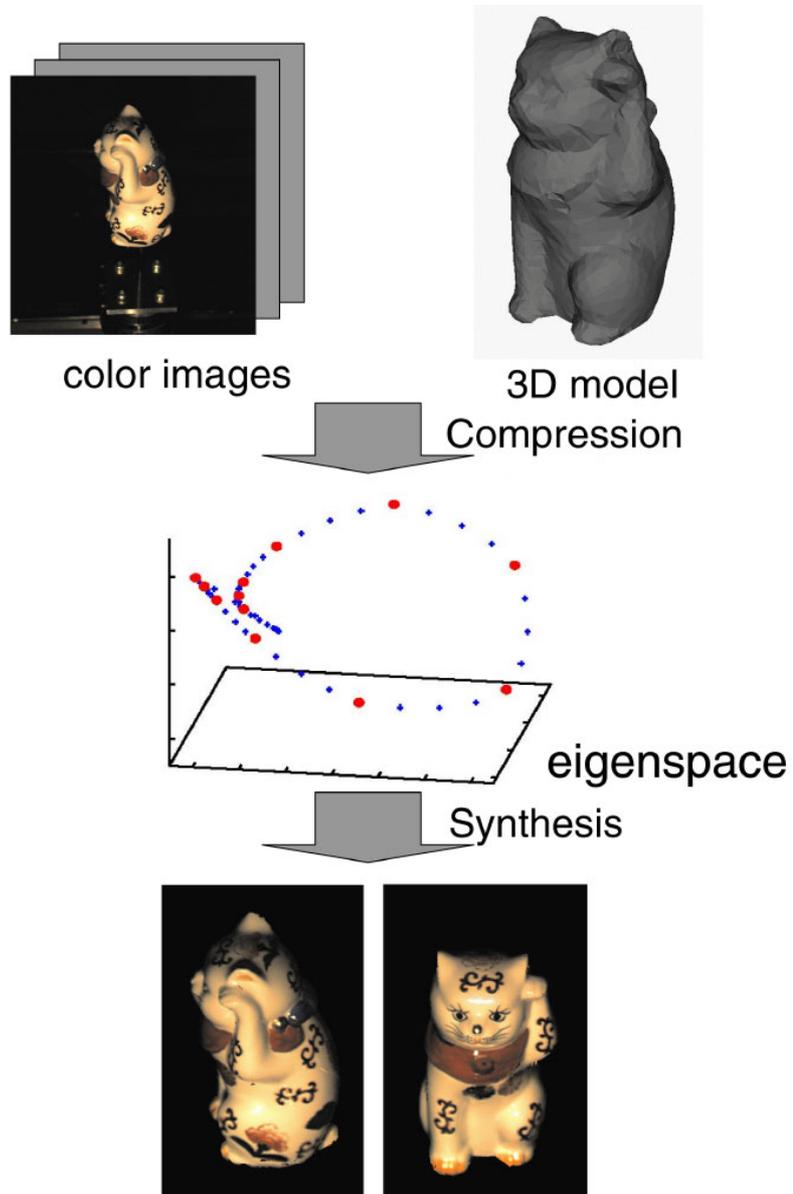


Figure 3.1: Outline of the *Eigen-Texture Rendering* method.

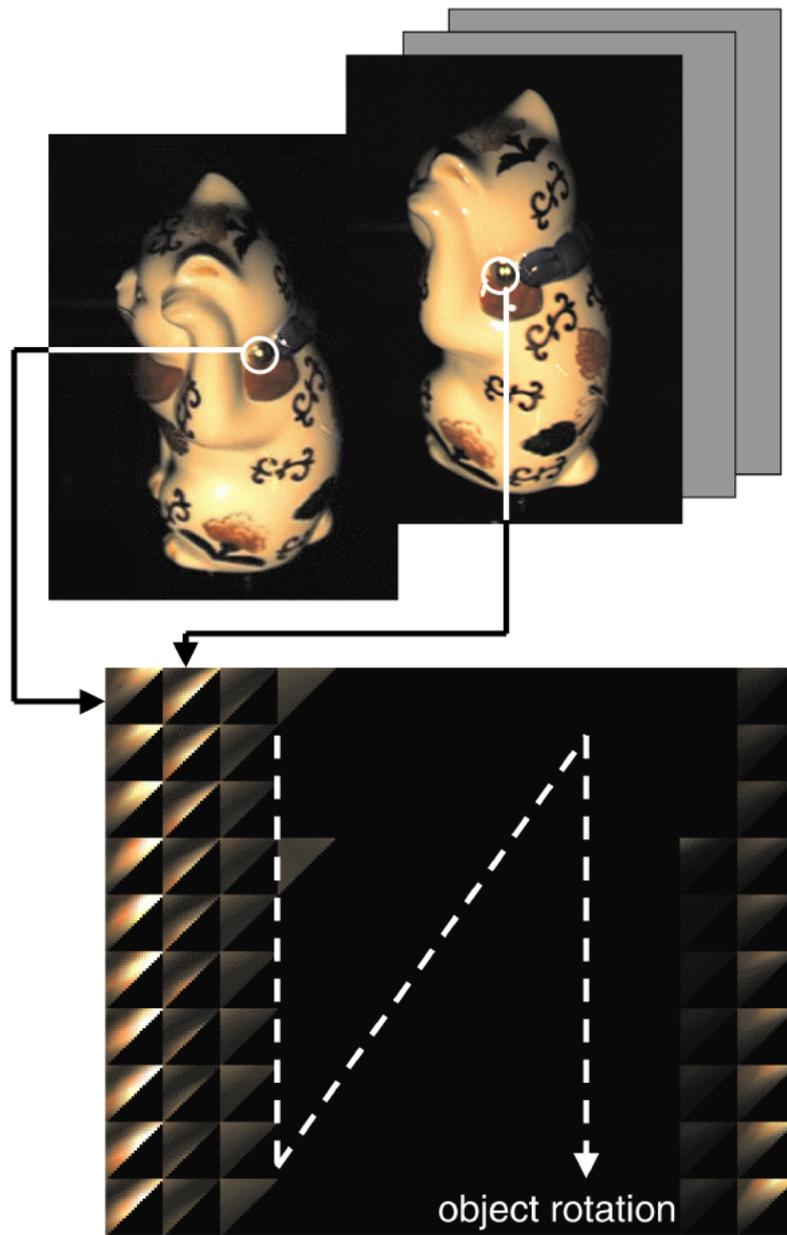


Figure 3.2: A sequence of cell images.

### 3.3 Implementation

We have implemented the system described in the previous section, and have applied the *Eigen-Texture Rendering* method to real objects.

#### 3.3.1 System Setup

We built an experimental system to capture the input color images and range images. In our capturing system setup, the object is attached to a rotary table (see Figure 3.3).

A single point light source fixed in the world coordinate is used to illuminate the object. A range image is taken by the 3 CCD color camera under a nematic liquid crystal mask [102]. A sequence of range images is taken by rotating the object  $30^\circ$  by each step in this experiment. Each range image is converted into

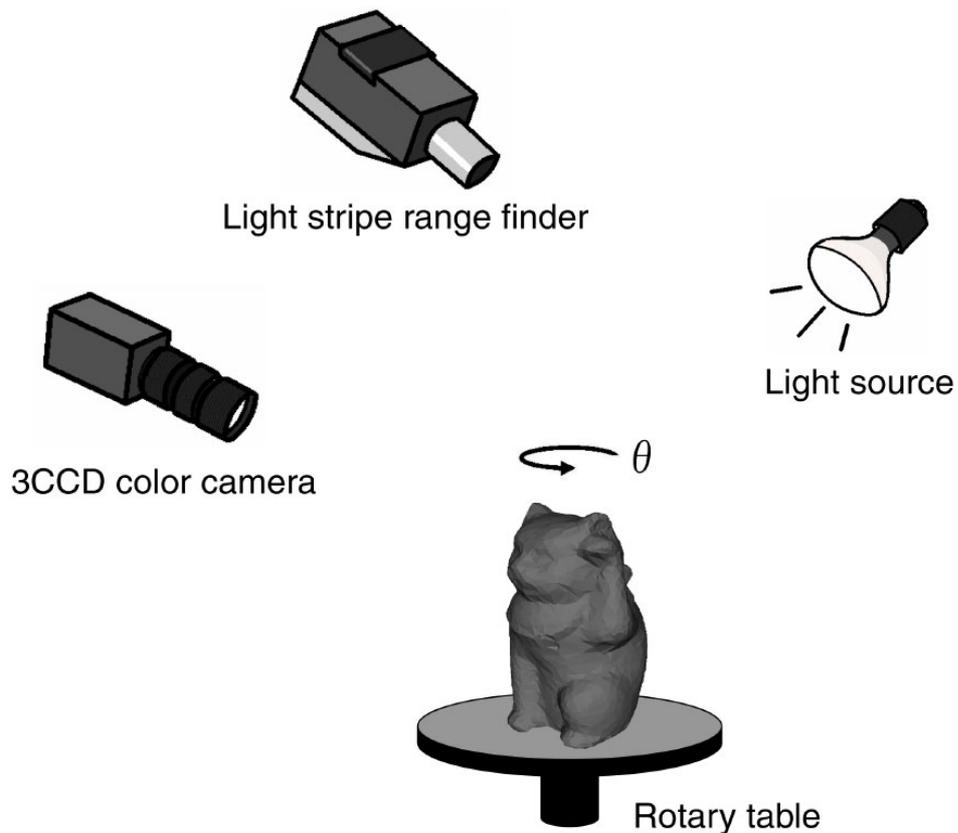


Figure 3.3: The system setup we use.

a triangular mesh model; the models are then registered to one another by the robust simultaneous registration method described in Section 2.2. After registration, to obtain one complete mesh model, all mesh models are integrated by the integration method in Section 2.3. The integration is done in a non-adaptive manner. We will discuss the possible advantages of using an adaptive integration later in this document. A sequence of color images is also taken by the same 3 CCD color camera, rotating the object similarly to the rotation of the range image sequence; however, the object rotation interval is smaller than that of the range image sequence. For instance, a step of  $3^\circ$  was used for the first experiment described in the next section.

Since we use a light stripe range finder, the range images and the color images are taken with the same camera. Therefore, the color images can be easily aligned with the 3D model generated from the range images. Another alternative way to obtain accurate 3D models is to use laser range finders. When using other types of range finders such as laser range finders, since the 3D model and the color images will be taken in different coordinate systems, the systems must be aligned with one another. This can be accomplished by minimizing some error metric of the point correspondence between the color images and the 3D model [57], or by maximizing the correlation between them.

### 3.3.2 Cell-adaptive Dimensional Eigenspace

Determining the number of dimensions of the eigenspace in which the sequence of cell images are stored is a non-trivial issue, as it has significant influence on the quality of the synthesized images. According to the theory of photometric stereo [128] and Shashua's linear theory [109], in an ideal case where the 3D model is perfectly aligned with the input images and where the model approximates the real object surface well, i.e., when the object surface approximated by the triangular patch is a plane, or when the geometric model is sufficiently tessellated, three dimensions are enough for compressing and synthesizing the appearance of an object with a Lambertian surface. However, as shown in Figure 3.4, the images synthesized by using the data stored in only 3 dimensional eigenspace for every cell have an intolerable blurred effect around the highlights, and the textures are matted. As the reflection of most general real objects

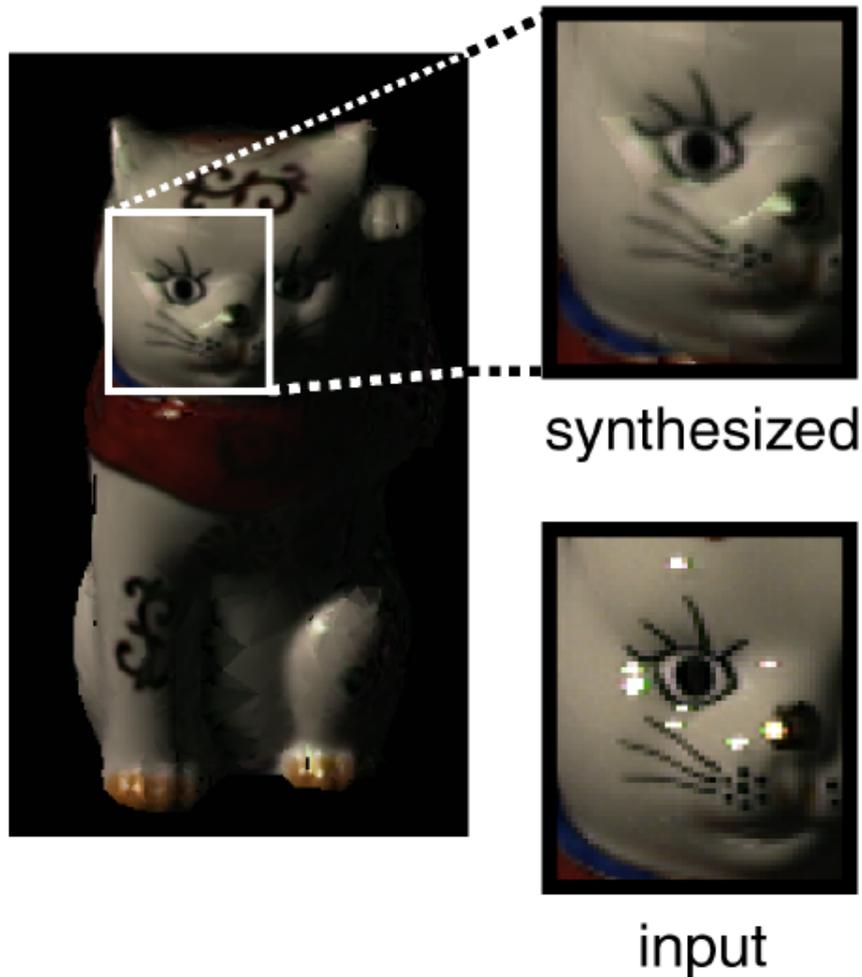


Figure 3.4: Virtual object image synthesized by using three dimensional eigenspaces (an example of a highly reflective object).

cannot be approximated by a simple Lambertian reflection model due to non-linear factors such as specular reflection and self shadowing, three dimensional eigenspace is not appropriate to store all the appearance changes of each cell.

The quality of the geometric model has a serious influence on the number of dimensions of eigenspace required to synthesize the image precisely. The simpler the construction of the geometric model, the higher the eigenspace dimensions that are needed, since the triangular patches get far from approximating the object's real surface, and the correlation of each cell image becomes low. Figure 3.5 shows a rendered image of a wool plush duck; only three eigen vectors were used to render the image.. Although the material has a Lambertian re-

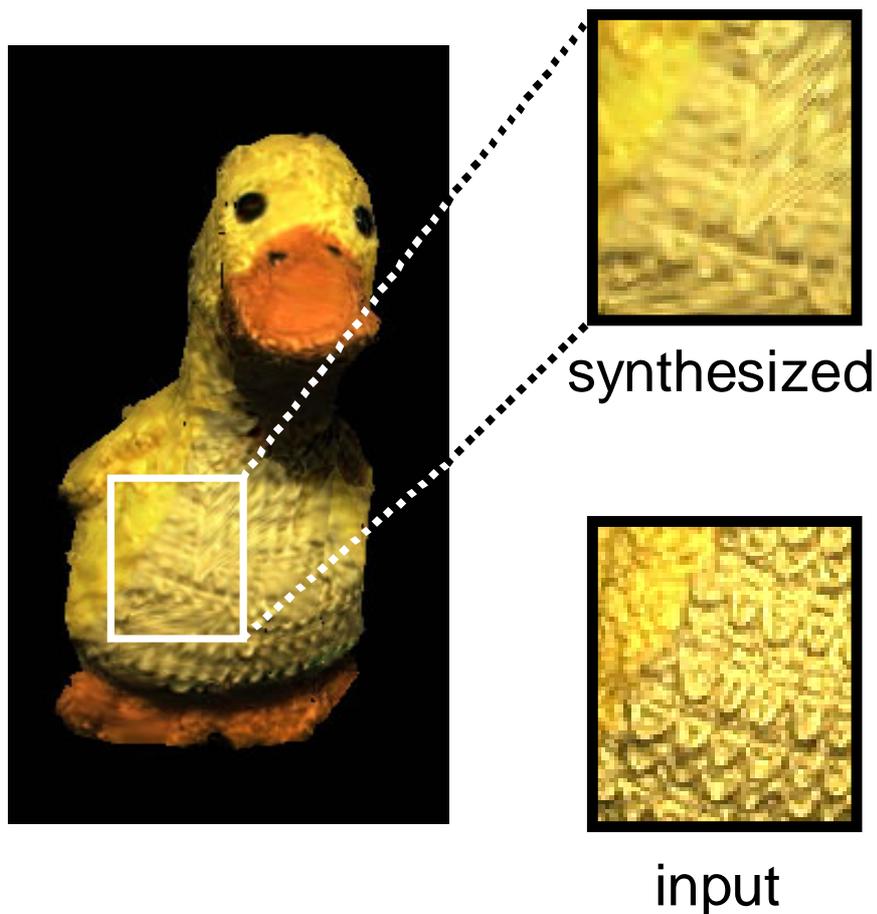


Figure 3.5: Virtual object image synthesized by using three dimensional eigenspace (an example of an object with rough surface).

flectance property because it has fairly detailed structure that is difficult to be approximated well by the 3D model surface, three dimensions do not provide an accurate rendition; in the final rendering, self shadows generated by the wool that sticks out from the duck's belly are smoothed out.

These facts indicate that the number of dimensions of eigenspace should differ for each of the cells, according to whether they have highlights or self shadows in their sequences, and also to the size of their triangle patch.

Taking these points into account, we determined the number of dimensions of the eigenspace independently for each cell so that each could be synthesized precisely. We used eigen ratio to determine the number of dimensions for each cell. Figure 3.6 describes the relationship between the eigen ratio, the error per

pixel versus the number of dimensions of the eigenspace. Here, the eigen ratio is the ratio of the eigen values in their whole summation; the error per pixel describes the average difference of the pixel values in 256 gradation between the synthesized and real images. The eigen ratio is in inverse proportion to the error per pixel, so that it is reasonable to threshold the eigen ratio to determine the number of dimensions of the eigenspace. For each sequence of cell images, we computed the eigen ratio with Equation (3.7), and used the first  $k$  eigen vectors whose corresponding eigen values satisfied a predetermined threshold of the eigen ratio. The number of dimensions for each cell required to compress the sequence of input images and to reconstruct the synthesized images can be optimized by using these cell-adaptive dimensions. Yet, on the whole, the size of the database can be reduced. This cell-adaptive dimension method can be implemented because our method deals with the sequence of input images as small segmented cell image sequences.

Figure 3.7 shows the images synthesized by using 0.999 as the threshold of the eigen ratio for each cell. As can be seen in Figure 3.7, the results described on the right side are indistinguishable from the input images shown on the left side. The number of dimensions of eigenspace used, the compression ratio, and the average error per pixel are summarized in Table 3.1. Due to the small tufts of protruding wool around the duck's stomach, tiny self shadows appear throughout the input sequence of color images. This causes the increase of the necessary number of dimensions for the duck compared with those necessary for the other objects. Because our method does not assume any reflectance model and does not require any analysis of the surface properties, it can be applied to objects with rough surfaces such as the bear with fine fur and the wool plush duck, an application which is difficult to accomplish with model-based methods.

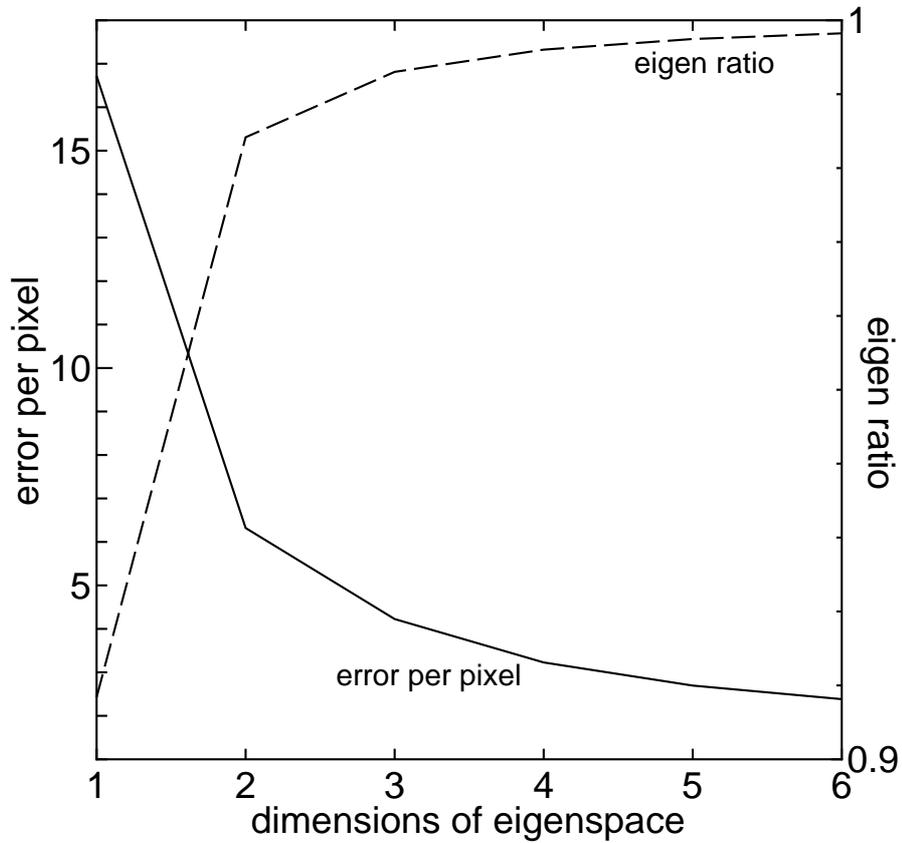


Figure 3.6: Graph of the eigen ratio and the error per pixel versus the number of dimensions of eigenspace.

	Cat	Bear	Duck
Average number of dimensions	6.56	6.96	17.6
Compression ratio	15.3:1	14.5:1	5.72:1
Essential compression ratio	8.34:1	6.02:1	3.15:1
(MB)	357:42	702:117	290:92
Average error per pixel	1.46	1.79	1.79

Table 3.1: Some statistics of the compression results.



Figure 3.7: Left: Input color images, Right: Synthesized images (by using cell-adaptive dimensional eigenspaces).

### 3.3.3 Compression Ratio

The compression ratio defined in Equation (3.8) is the compression ratio of the cell image sequences; this ratio is computed from the number of dimensions, the size of each cell image and the number of input color images. The essential compression ratio described in Table 3.1 is also the compression ratio between the cell image sequence and the data size to be stored in eigenspaces, but its value is computed from the real data size on the computer. The values of this essential compression ratio are relatively lower than the compression ratio computed by Equation (3.8). This disparity is caused by the difference in data types: the cell image sequences derived from the input color image sequence are represented by *unsigned char* type, while the data to be stored in eigenspaces – the projections and eigen vectors – are represented with *float* type. In future work, standard compression techniques such as vector quantization can be applied to avoid this loss in compression due to data types. As our main goal for the proposed method is to show the effectiveness of treating the appearance variation of real objects on the its surface, we applied PCA as a straightforward compression technique that allows intuitive interpolation as will be discussed in Section 3.3.4.

In this thesis, we do not address computing the compression ratio between the input image sequence and the data size stored in eigenspaces because an accurate comparison cannot be accomplished due to the resolution problem of cell images. When the *Eigen-Texture Rendering* method converts the input image sequence into a set of cell image sequences, the resolution of the cell images is determined to a fixed number with regard to the largest triangular patch on the 3D model be surface. Although this enables us to synthesize virtual images in higher resolution as compared with that of the input images, it causes loss in compression ratio. We are now investigating a method to determine the resolution of cell images adaptively to their corresponding triangular patch size; in addition, we are considering a method to derive higher compression ratio, mesh generation of the 3D model with regard to the color attributes on its surface.

### 3.3.4 Interpolation in eigenspace

Once the input color images are decomposed into a set of sequences of cell images, and projected onto their own eigenspaces, interpolation between each input image can be accomplished in these eigenspaces.

As an experiment, we took thirty images of a real object as the input image sequence by rotating the object at  $12^\circ$  by step. By interpolating the projections of these input images in the eigenspace, we obtained interpolated projections for  $3^\circ$  degrees rotation by each step. By synthesizing images using these interpolated projections, images of the object whose poses were not captured in the input color images could be synthesized. Figure 3.8 depicts the curve composed of the coefficients of the first three eigenvectors for each viewpoint corresponding to one particular cell. The red points represent the views included in the input images sequence and the blue points are the interpolated virtual views.

Figure 3.9 shows the synthesized images with interpolation in eigenspace. The results prove that we can reconstruct synthetic images by interpolation in eigenspace with adequate accuracy. The average dimensions of eigenspace for all cells used to fill 99.9% in eigen ratio was 5.1, and the compression ratio was about 20:1. But as we used only 30 images as the input image sequence and synthesized 120 images, the substantial compression ratio became almost 80:1. The average error per pixel at this time was about 7.8.

On the other hand, when we attempt to interpolate the appearance of highly reflective objects, the highlights will not be interpolated smoothly. As can be seen in Figure 3.10 and Figure 3.11, the highlights will fade out and fade in, rather than extend across the objects. This is an unavoidable effect as far as we interpolate the specular reflection linearly, in eigenspace in our case. However, from our experience, this highlight jumping effect can be tolerated when the sequence is shown as a movie in which the input images are provided not too sparse, for instance  $12^\circ$  by step for  $360^\circ$  reconstruction.

When using Moving Picture Experts Group 1 (MPEG1) to compress all the input color images into a single image sequence, the compression ratio becomes around 127 : 1. Despite this high compression ratio achieved by MPEG1, the average error per pixel becomes around 7 to 8. Since image sequence compression methods such as MPEG1 compress the images in their 2D coordinates without

any geometric information of the target object, the errors tend to appear around the nonlinear changes of the pixel values in the images (Figure 3.12); i.e., edges of the occluding boundaries, edges of the texture, highlights, etc. These errors cannot be seen in the images synthesized with *Eigen-Texture Rendering*. Even when compressing the input color images putting the priority on the quality by using MPEG1, it is hard to avoid these errors while keeping the compression ratio lower than *Eigen-Texture Rendering*. This is because *Eigen-Texture Rendering* accomplishes compression on the object surface. Other compression techniques than principle component analysis, e.g., discrete cosine transformation, may work as well, as long as the appearance change is treated on the object

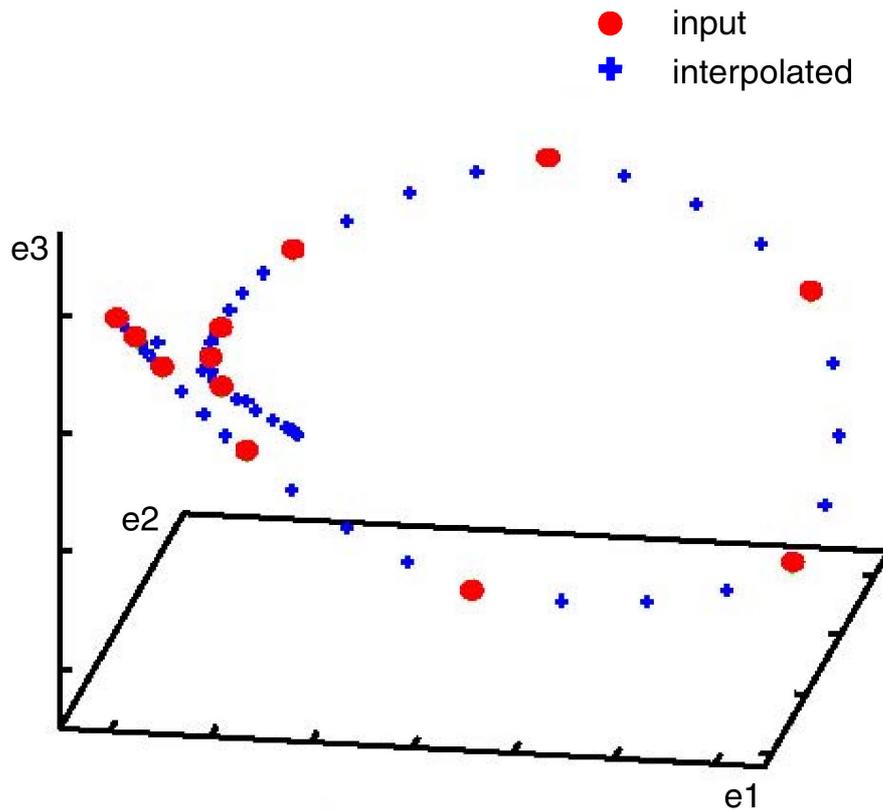


Figure 3.8: Interpolation results in the eigenspace.



interpolated synthetic images

Figure 3.9: Synthetic images reconstructed by interpolating input images in eigenspace.

surface.

### 3.4 Integrating into real scenes

Since our method holds an accurate 3D object model constructed from a range image sequence, the synthesized virtual images can be seamlessly integrated into real scene images, taking the real illumination environment into account. As shown in Equation (3.10), the irradiance at one point on the surface of an object from the whole illumination environment is a linear combination of the irradiance due to all light sources that compose the illumination environment. For this reason, a virtual object image under the real illumination environment can be synthesized by decomposing the real illumination environment into several component light sources and then sampling the color images under each component light source separately.

$$\begin{aligned}
 I &= \int \int \left\{ \sum_{j=1}^n L_j(\theta_i, \phi_i) \right\} R(\theta_i, \phi_i, \theta_e, \phi_e) \cos \theta_i d\theta_i d\phi_i \\
 &= \sum_{j=1}^n \int \int L_j(\theta_i, \phi_i) R(\theta_i, \phi_i, \theta_e, \phi_e) \cos \theta_i d\theta_i d\phi_i
 \end{aligned} \tag{3.10}$$

where  $I$ : irradiance at one point on the object surface

$L_j$ : radiance of one component light source

$R$ : BRDF of the object surface

$(\theta_i, \phi_i)$ : illumination direction

$(\theta_e, \phi_e)$ : view direction



Figure 3.10: Side by side comparison of input images and rendered images with interpolation in eigenspace. The original images are in the first column and the rendered images are in the second column. The first and last images in the original images were used as the input; the intermediate images in the rendered sequence are the results of interpolation.

As a preliminary experiment, we took input color images under 3 point light sources, lighting them separately; we synthesized images of the virtual object with all lights turned on. Under each point light source, we took 40 color images and synthesized 120 virtual object images for each light source with the threshold 0.999 in eigen ratio with interpolation in eigenspace; we then synthesized an image sequence of the virtual object with all lights on by taking a linear combination of image sequences of the virtual object with each point light source on.

Figure 3.13 shows the result of the linear combination of a certain pose. In addition, we integrated the virtual object image sequence into a real scene image, which also had been taken under a condition in which all lights were turned on. The real illumination environment was captured with a stereo setup of cameras with fish-eye lenses [100], and the virtual coordinate system was transformed into the real scene coordinate by calibrating the camera to capture the

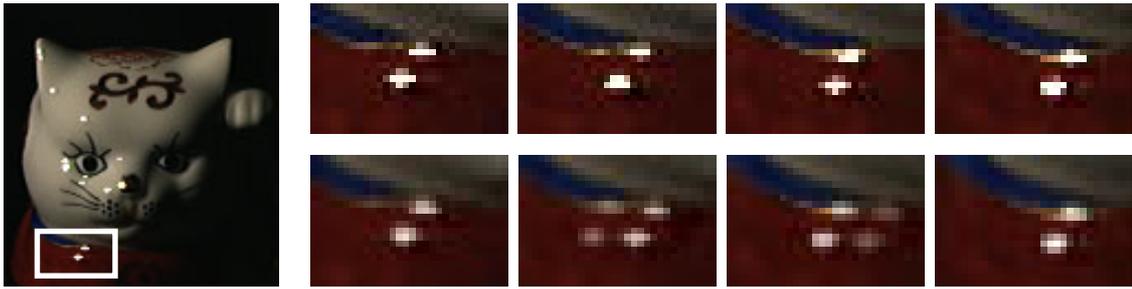


Figure 3.11: Zoom ups of the images in Figure 3.10.



Figure 3.12: A comparison with MPEG1 compression accomplished on the image.

real scene [119]. The result of the integration, shown in Figure 3.14, proves that our method enables the creation of the accurate appearance of the object surface and the precise shadow of the virtual object according to the real illumination environment.

### 3.5 Summary

We have proposed the *Eigen-Texture Rendering* method as a new rendering method for synthesizing virtual images of an object from a sequence of range and color images. The implementation of the method proves its effectiveness, in particular, its high compression ratio. Also, we have demonstrated seamless integration of virtual appearance with real background images by using the method. The

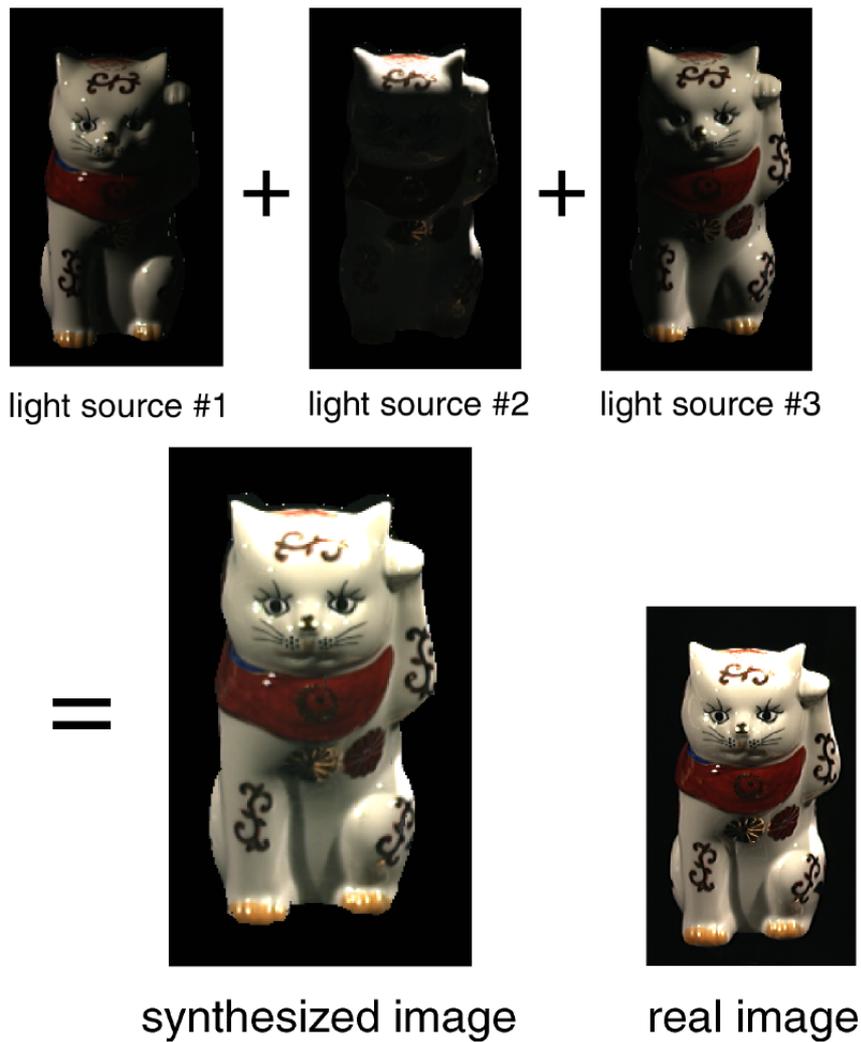


Figure 3.13: Linear combination of light sources.

merits of the proposed method can be summarized as follows.

First, high compression ratio can be achieved because we compress a sequence of cell images corresponding to a physical patch on the object surface. Appearance variation in the sequence is approximately limited to brightness change due to illumination geometry. Second, interpolation in eigenspace can achieve synthesis of a virtual object image when object pose is not included in the input image sequence. Owing to this interpolation in eigenspace, we can reduce the necessary number of sampling color images, and reduce the amount of data to be stored. Third, a wide range in application is possible, because we do not need any detailed reflectance analysis. The method can be applied to such



Figure 3.14: Integrating virtual object into real scene.

objects as those with rough surfaces, i.e., the bear and duck shown in Figure 3.7, or with strong highlights, and whose color values saturate the dynamic range of a CCD camera, e.g., the cat in Figure 3.7. Finally, since an accurate 3D model of the real object is constructed from the input range images, we can generate accurate cast shadows when integrating virtual images with real background images. Thanks to the linearity in image brightness, we can decompose the real illumination distribution into separate component light sources, and sample the object under each component light source separately.

On the other hand, our method has a few drawbacks. In particular, the computational expense for compression using eigenspace could be large. As solving the eigenstructure decomposition problem requires a large number of iterations, the computational cost for storing the input image sequence in eigenspace becomes relatively expensive, although, once the input images are compressed

and stored in eigenspace, the synthesis of virtual images can be computed in real time. With regard to this point, we believe our method has an advantage especially on applications for interactive mixed reality systems, such as virtual museums and electric shopping malls, where the targets can be compressed beforehand off-line.

Currently, we treat each cell image sequence that stacks the appearance change of one triangular patch separately. However, if the cell image sequences can be classified into groups that have the same underlying texture and reflection property, accomplishing compression on the whole set of cell image sequences will achieve better compression ratio. As future work, we plan to adopt the mesh models subdivided (with regard to the base texture) that can be constructed via the integration method presented in Section 2.3; in addition, we plan to analyze the possibility and advantages of constructing a global eigenspace by stacking several cell image sequences and applying principle component analysis (PCA) to it.

# Chapter 4

## Rendering from a Sparse Set of Images

### 4.1 Approach

In this chapter, we will begin by considering an imaginary practical scenario. For instance, consider a situation where a person wants to show one of his objects to a friend remotely, e.g., via the internet, thereby allowing his/her friend to closely examine every detail of the object just as if the friend were actually holding it in his/her own hand. This situation can directly apply to what people might want to do when they are purchasing objects online, i.e., e-commerce. Current techniques require the user to take a large amount of images (image-based rendering) or to assume that the scene structure, including the lighting environment, is known perfectly (inverse rendering). Although, the method we have seen in Chapter 3 can drastically reduce the amount of the number of required input images by achieving interpolation in the eigenspace, when the input image sequence becomes too sparse, users suffer from a fade-in and fade-out problem of the non-rigid appearance variation (see Figure 3.11). Our ultimate goal is to enable people to realize the aforementioned scenario with as little effort as possible with a representation that is as compact as possible. For our immediate goal, however, in this chapter, we will concentrate on a constrained but still realistic scenario. We target a situation where a user takes several snapshots of an object of interest with a digital camera, aiming the camera from various locations around the object. The user wants that information to be extended into some sort of representation, so that the user can see the object from arbitrary

viewpoints and, if possible, change the lighting environment. On tackling this problem, we make several assumptions while trying to keep the generality of the scenario.

### **Static illumination environment, static object and moving camera**

We consider a scenario where the user takes several snapshots of an object while moving around the object. Obviously, the object and the light sources do not move during the image capturing period; only the camera moves.

### **Pre-acquired geometry of the target object**

We assume we know the precise geometry of the target object. By using a laser range scanner, we acquire several range images of the target object and go through the geometric modeling pipeline: 3D-3D registration, integration and simplification. The final result is a 3D mesh model.

### **Known intrinsic camera parameters**

The camera that is used to capture the input images can be pre-calibrated easily, using techniques like [135, 119].

### **Known extrinsic camera parameters**

The motion of the camera can be pre-estimated by applying low-level computer vision techniques: structure from motion, bundle adjustment, etc. In the experiments we conduct, we simultaneously acquire a range image using a light stripe range finder [102] as well as capturing color images for each viewpoint. We then use those range images to estimate the viewpoint via 3D-3D registration against the mesh model we mentioned earlier. Note that these range images are only used to estimate the viewpoints and not used to reconstruct the mesh model of the target object. Other methods, for example, specifying several corresponding points both in 2D and 3D by hand and solving a least square minimization problem to accomplish 2D-3D registration may provide accurate viewpoints, especially if the relative motions between the cameras are known [127].

### **Homogeneous specular reflectance property**

We assume the specular reflectance property of the object surface is the same

for all points on it. Specifically, we assume  $(\mathbf{K}_S, \sigma)$  in the simplified Torrance-Sparrow reflection model that we describe in Section 4.2.1 are the same for the whole object surface.

With these assumptions, we present a framework to accomplish photorealistic rendering, including view-dependent rendering and relighting, from a sparse set of images. These images sparsely sample the appearance variation of a real world object under a static illumination environment. With the aid of the geometric information of the target object, and assuming that the specular reflectance property is homogeneous over the object surface, we estimate three important elements necessary for photorealistic forward rendering: diffuse texture, approximate Bidirectional Reflectance Distribution Function (BRDF) and the illumination distribution.

We assume that the surface reflection can be approximated by a dichromatic reflection model with a Lambertian diffuse reflection and simplified Torrance-Sparrow specular reflection. Given a relatively small number of images (on the order of one) of an object, the geometry of the object (as a 3D mesh model) and the positions of the cameras corresponding to each image, we first separate the diffuse and specular reflection components on each object surface point. By constructing a *diffuse texture map* that represents the diffuse reflection and subtracting the diffuse-texture-mapped images from the input images one by one, we obtain a sequence of specular reflection images. Then, we derive an initial estimation of the illumination distribution by shooting back each pixel value to the perfect mirror direction and mapping those values to a hemisphere, which we call the *illumination hemisphere*. We then formulate the specular reflection mechanism as a 2D convolution on the surface of the illumination hemisphere and estimate the reflection parameters and the true illumination distribution by solving a blind 2D deconvolution problem. The algorithm for this blind deconvolution problem is based on the Alternating Minimization (AM) algorithm with a Total Variation (TV) regularization term imposed on the illumination distribution. A robust technique based on M-estimators is incorporated to combat noise inherited from the diffuse texture extraction procedure.

## 4.2 Reflection Separation

To derive an efficient representation of the appearance of a real world object from a set of sparse sampling of its variations under a static illumination, we begin by separating the reflection components in the input images: the diffuse reflection and the specular reflection.

### 4.2.1 Reflection Model

We consider real world objects whose reflection mechanism at their surfaces can be approximated by a dichromatic reflection model [108, 56], where the reflected light at a surface point is explained as a linear combination of two reflection components: diffuse reflection and specular reflection. Since the image irradiance (pixel value) can be considered to be proportional to scene radiance [45], we will forget about the constant multiplier that relates scene radiance to image irradiance; instead, we will consider that pixel values in the images are directly associated to scene radiance. Under this assumption, the dichromatic reflection model tells us:

$$\mathbf{I} = \mathbf{I}_D + \mathbf{I}_S \quad (4.1)$$

where  $\mathbf{I}$  is the pixel value corresponding to a particular object surface point, and  $\mathbf{I}_D$  and  $\mathbf{I}_S$  denote the diffuse and specular reflection radiance, respectively. The bold characters denote a three-dimensional color vector, such that  $\mathbf{I} = [I_R \ I_G \ I_B]^T$ .

Diffuse reflection can be explained as result of the light that penetrated into the object medium and radiated back to the air after internal scattering due to the small particles in the object surface layer (see Figure 4.1). Because of this internal scattering, diffuse reflection can be approximated with the Lambertian model [60], where the diffuse reflection is described as an isotropic reflected light parameterized by the angle between the light source direction and the surface normal. If we consider a local spherical coordinate system with its origin set to the object surface point in interest, the radiance of diffuse reflection can be described as:

$$\mathbf{I}_D = \max[0, \mathbf{K}_D \int_{\Omega} L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i] \quad (4.2)$$

where  $\mathbf{K}_D$  is a three band color vector which is determined by both the light source color and the surface color. Also,  $\theta_i$  and  $\phi_i$  denote the altitude and az-

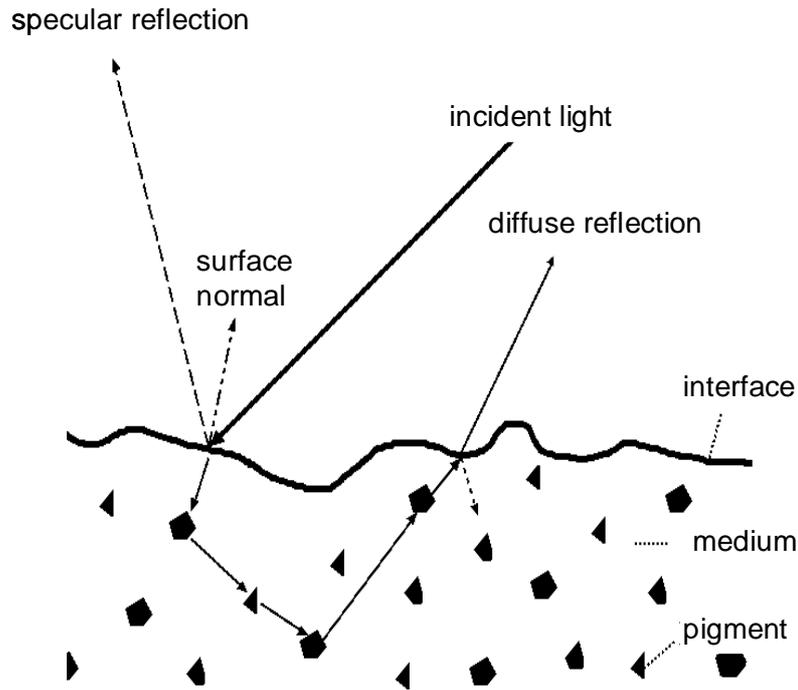


Figure 4.1: The mechanism of dichromatic reflection model on a object surface.

imuth coordinate of the light source with radiance  $L_i$ , respectively, and  $\omega_i$  denotes the solid angle of the light source  $L_i$ . Although recent studies on the diffuse reflection mechanism have revealed that, when the object surface has a high macroscopic roughness, the diffuse reflection becomes view-dependent [82], we consider the objects we handle have a diffuse reflection that can be approximated with a Lambertian reflection model, which is true for most man-made objects we encounter in daily life.

Specular reflection is the light directly reflected at the interface between the air and the object surface (see Figure 4.1). Reflected light spreads around the perfect mirror direction to form a lobe and exhibits a spike in the perfect mirror direction. The spike of the specular reflection can be observed only on object surfaces smooth enough in comparison with the wavelength of the incident light, and when the camera viewing direction is aligned with the perfect mirror direction. Hence, we will ignore the spike and choose to adopt the Torrance-Sparrow reflection model [118] as a numerical approximation model for specular reflection, which is well known to represent the specular lobe successfully with a simple numerical expression [80]. The Torrance-Sparrow reflection model ex-

presses the specular lobe as the light reflected at object surface with microscopic roughness; small V-shaped grooves which are lined with flat mirrors called microfacets. The distribution of the orientations of microfacets is approximated with a Gaussian distribution as follows:

$$\mathbf{I}_S = \int_{\Omega} \frac{\mathbf{K}_S F G}{\cos \theta_r} L_i(\theta_i, \phi_i) \exp\left[-\frac{\alpha^2}{2\sigma^2}\right] d\omega_i \quad (4.3)$$

where  $\mathbf{K}_S$  is the color vector of the reflection which accounts for the normalization factor of the exponential function and the reflectivity of the surface,  $F$  is the Fresnel's coefficient,  $G$  is the geometrical attenuation factor,  $\theta_r$  is the angle between the viewing direction and the surface normal,  $\alpha$  is the angle between the surface normal and the bisector of the viewing direction and the light source direction, and  $\sigma$  represents the surface roughness.

#### 4.2.2 Diffuse Texture Map

With the assumption that the light reflected at the object surface can be approximated with a dichromatic reflection model with Lambertian diffuse reflection and Torrance-Sparrow specular reflection, as seen in the last section, it can be easily observed that the diffuse reflection component at each surface point has a constant value for all images in the input image sequence and that the pixel value is equal to or larger than the diffuse reflection color. Concretely, since we vary the viewpoint while setting the light source directions and object still,  $\theta_i$  in Equation (4.2) does not vary in the input image sequence. Hence, we are able to represent the diffuse reflection with a single RGB color vector for each surface point.

To extract this diffuse reflection vector for each object surface point, we examine the scene radiance (image irradiance) variation of each surface point throughout the image sequence, and use the pixel value with minimum magnitude as the diffuse reflection color. Since we have only a few images, and because the object surface point may be occluded in some of those images, taking the minimum brightness pixel value is the best strategy we can employ to obtain the constant diffuse reflection vector. If we do have more images as the input, more sophisticated strategies may work better, for instance, taking the median value like Wood et. al [127] or fitting a Gaussian distribution to the histogram of pixel values.

Instead of keeping these diffuse reflection color vectors for each object surface point, we extract them in a triangular patch-based manner. By considering 3D grids in each triangular patch of the geometric mesh model, and projecting those grids onto each input image and extracting the minimum brightness pixel value for each grid point, we are able to construct a texture map that represents the constant diffuse reflection. We will refer to this texture map as the diffuse texture map. In our current implementation, we extract the diffuse texture for each triangular patch in the same pre-defined size, e.g.,  $20 \times 20$  pixels.

### 4.2.3 Illumination Hemisphere

By texture mapping the diffuse texture map for each viewpoint in the input image sequence and by subtracting those diffuse-texture-mapped images from each input image one by one, we obtain a set of residual images. These residual images mainly consist of scene radiance resulting from specular reflection at the object surface and some interreflection and noise inherited from the diffuse texture map extraction procedure. In our work, we ignore interreflection and simply consider it as noise. Hence, we will refer to these residual images as specular images.

As described in Section 4.2.1, the specular reflection approximated with the Torrance-Sparrow reflection model has an intensity peak slightly off from the perfect mirror direction: the direction where the surface normal becomes the bisector of the incident light vector and viewing direction vector. To obtain a rough approximation of the illumination distribution, we shoot back each pixel value in each specular image in the perfect mirror direction and map those values to a hemisphere covering the 3D object model. The radius of the hemisphere will be pre-determined. We call this representation of the illumination distribution the *illumination hemisphere*.

The illumination hemisphere generated from each specular image covers only partial regions of the true illumination environment; we need to combine these partial illumination hemispheres to make a full illumination hemisphere that approximates the real lighting environment. To deal with noise, we make a mask hemisphere that counts how many times each point was taken into account while making the partial illumination hemispheres. We then adopt hemi-

sphere points that have counts close to the total number of images. Since some light sources may be occluded in some partial illumination hemispheres, we set the threshold less than the total number of images. Only those points that pass this check will be mapped onto the final illumination hemisphere; in this case, we take the mean of the intensity values from the partial illumination hemispheres as the intensity value. This way of consistency checking also reduces the errors, e.g., those introduced by 3D-3D miss-alignment for viewpoint estimation. These errors are not view-dependent and would not stay in a particular region on the illumination hemisphere. Interreflection will also be faded out, since it can be considered to be the reflected light of moving light sources.

### 4.3 Illumination and Reflectance Parameter Estimation

Next, using the aforementioned illumination hemisphere and a simplified Torrance-Sparrow reflection model, we will consider decoupling the surface reflectance property and the illumination distribution from the specular images.

#### 4.3.1 Problem Formulation

First we assume that the object has a specular reflection property that obeys the Torrance-Sparrow reflection model, except that we assume the geometrical attenuation factor  $G$  in Equation (4.3) to be 1 and the Fresnel reflectance coefficient  $F$  to be constant. These assumptions are valid for most dielectric and metal objects [113]. With this simplified Torrance-Sparrow reflection model, the specular reflection at one object surface point  $v$  can be given as:

$$\mathbf{I}_S(v) = \frac{\mathbf{K}_{S,v}}{\cos \theta_r} \int_{\Omega} L_i(\theta_i, \phi_i) \exp\left[-\frac{\alpha^2}{2\sigma^2}\right] d\omega_i \quad (4.4)$$

We now represent the illumination distribution as a hemisphere that covers the object. By using a geodesic hemisphere, which is a subdivided icosahedron and hence can uniformly distribute nodes on the hemispherical surface, and by considering each geodesic node to be a point light source, we can compute the specular reflection at point  $v$  with a global spherical coordinate system where each point light source's position can be parameterized with its altitude and

azimuth angle  $\theta_l$  and  $\phi_l$  respectively,

$$\mathbf{I}_S(v) = \frac{K_{S,v}}{\cos\theta_r} \sum_l^{N_L} \omega_l L_l(\theta_l, \phi_l) \exp\left[-\frac{\alpha^2}{2\sigma_v^2}\right] \quad (4.5)$$

where  $L_l$  stands for the radiance of each point light source. Also,  $\theta_r$  and  $\alpha$  are still the angle between the viewing direction and the surface normal and the angle between the surface normal and the bisector of the viewing direction and the light source direction, respectively, however in a global coordinate system.

$\omega_l$  stands for the solid angle of the area that each point light source represents; in our case, where each node will be distributed uniformly on a  $2\pi$  hemisphere,  $\omega_l$  becomes  $\frac{2\pi}{N_L}$  where  $N_L$  stands for the number of nodes in the geodesic hemisphere. The color vector direction of the specular reflection is the same as that of the light source. Thus, if we assume that all light sources in the scene have the same color, we can concentrate on the relationship between the radiance of the light sources  $L_l$  and the image irradiance  $I_S(v)$ . Therefore, we use the average color,  $\mathbf{L}$ , of the initial illumination hemisphere as the color of the point light sources. Also, we assume that the target object has a homogeneous specular reflection property, so that  $\sigma_v$  can be represented with one value  $\sigma$ . Similarly  $K_{S,v}$  will be represented with one value  $K_S$ . With these assumptions, we can rewrite Equation (4.5) as,

$$\mathbf{I}_S(v) = I_S(v)\mathbf{L} \quad (4.6)$$

$$I_S(v) = \frac{2\pi}{N_L} \frac{K_S}{\cos\theta_r} \sum_l^{N_L} L_l \exp\left[-\frac{\alpha^2}{2\sigma^2}\right]. \quad (4.7)$$

Note that all values in Equation (4.7) are scalar values, and  $L_l$  is now the magnitude of the color vector of each point light source placed on the illumination hemisphere.

The problem of estimating the reflection parameters and illumination distribution that best explains all specular images can now be formulated as a minimization problem with respect to  $\sigma$ ,  $K_S$  and  $L_l$  as:

$$\arg \min_{\sigma, K_S, L_l(l=0, \dots, N_L)} \sum_{k=0}^{N_K} \sum_{s,t=0,0}^{N_S, N_T} |I_{(s,t,k)} - I_{S,(s,t,k)}|^2 \quad (4.8)$$

where  $I_{(s,t,k)}$  is the pixel value observed at point  $(s, t)$  in the  $k$ th specular image, and  $(N_S, N_T)$  are the width and height of the specular images. Since we know

the camera position for each specular image and the geometry of the target object, the specular reflection can be computed with Equation (4.7) as  $I_{S,(s,t,k)}$ .

### 4.3.2 Specular Reflection as Convolution

As described in the last section, estimating the reflectance parameters and the illumination distribution from the specular images can be formulated as a minimization problem. However, Equation (4.8) is ill-posed and estimating  $\sigma$ ,  $K_s$  and  $L_l (l = 0, \dots, N_L)$  that best explains the specular images becomes solving a deconvolution problem. To see why the specular reflection becomes a convolution, we will explicitly write down the physical coordinate transformations embedded in Equation (4.8).

Since we have the 3D geometric model of the target object, the 3D coordinates of a point in the specular image can easily be obtained. Let us consider that a mapping function  $M$  takes a 2D point and the frame number of the specular image and returns the 3D coordinate of that point. If the 2D point in the specular image does not fall in the target object region,  $M$  simply returns some 3D point that does not have any meaning.

$$M : (s, t, k) \mapsto (x, y, z) \quad (4.9)$$

Also, let us consider a mapping function that returns the surface normal of a 3D point, such that,

$$N : (x, y, z) \mapsto (n_x, n_y, n_z). \quad (4.10)$$

We parameterize the illumination hemisphere with the altitude angle  $\theta$ , azimuth angle  $\phi$  and the radius  $r$ . As we pre-determine the radius of the illumination hemisphere to be a constant value  $R$ , the illumination hemisphere can be parameterized as a two dimensional function  $L(\theta, \phi)$ .

Given a 2D point  $(s, t, k)$  in the specular image  $k$ , we can compute the intersection of its surface normal with the illumination hemisphere as follows.

$$C : (s, t, k) \mapsto (\theta_C, \phi_C) \quad (4.11)$$

$$\begin{aligned} C(s, t, k) &= M(s, t, k) + \zeta(N(M(s, t, k))) \\ &= (x + \zeta n_x, y + \zeta n_y, z + \zeta n_z) \\ &= (\theta_C, \phi_C, R) \end{aligned}$$

where  $\zeta$  is chosen such that  $|C(x, y, z)| = R$  is satisfied. Similarly, given a point on the illumination hemisphere  $(\theta, \phi, R)$ , a 2D point in the specular image  $k$ , we can compute the intersection of the bisector of the angle between the view point  $(v_x^k, v_y^k, v_z^k)$  and object surface point and the light source position and the object surface point as follows.

$$BS : (s, t, k, \theta, \phi) \mapsto (\theta_{BS}, \phi_{BS}) \quad (4.12)$$

With these mapping functions, we can write down the expected pixel value in each specular image as follows.

$$I_S(s, t, k) = \frac{K_s \omega}{\cos \theta_r} \sum_l^{N_L} L_l(\theta_l, \phi_l) g_{\frac{1}{2\sigma^2}}(|BS(s, t, k, \theta_l, \phi_l) - C(s, t, k)|) \quad (4.13)$$

where  $\omega = \frac{2\pi}{N_L}$

and  $g_\gamma$  is a Gaussian function:

$$g_\gamma(\phi) = \exp[-\gamma\phi^2].$$

Recalling that  $(\theta_{BS}, \phi_{BS})$  is a function of  $(s, t, k, \theta, \phi)$  and  $(\theta_C, \phi_C)$  is a function of  $(\theta, \phi)$ , we can see that the specular reflection is a simple 2D convolution of the illumination hemisphere with a Gaussian filter, from Equation (4.13).

### 4.3.3 Alternating Minimization

From Equation (4.13) we can interpret the specular reflection mechanism with a Torrance-Sparrow reflection model as a 2D convolution, such that the 2D surface of the illumination hemisphere is convoluted with a Gaussian filter (Torrance-Sparrow reflection model) and mapped back to the 3D points. To make the following argument simple and consistent with references we cite, we will describe the illumination hemisphere as  $u(x, y)$ , the convolution filter as  $h(x, y)$  and the specular image  $z(x, y)$ . Then, the task of estimating the reflectance parameters and the true illumination distribution becomes finding  $u, h$  that best describes the following equation.

$$z = h * u + \eta \quad (4.14)$$

where,  $\eta$  is the noise mainly added while observing through an imaging system, which can be modeled by white Gaussian noise with zero mean.

This problem is called a blind image deconvolution problem, restoring both the image  $u$  and the blurring function  $h$  while given only the observed image  $z$  and probably some statistics of the noise  $\eta$ . The blind image deconvolution problem is well studied in the image processing society, and there are many existing algorithms to simultaneously identify  $u$  and  $h$  [11, 129, 130, 76]. The difficulty of solving this blind deconvolution is that it is ill-posed with respect to the image (the illumination hemisphere) and the blurring function (the Gaussian of the Torrance-Sparrow reflection model). Recently, You and Kaveh [129] have proposed a joint regularization technique to regularize both  $u$  and  $h$  to solve this problem:

$$\min_{u,h} f(u, h) = \min_{u,h} \|h * u - z\|_{L^2(\Omega)}^2 + \alpha_1 H^1(u) + \alpha_2 H^1(h) \quad (4.15)$$

where  $\Omega$  is the domain of  $u$  and  $h$ ,  $H^1$  regularization is defined as,

$$H^1(u) \equiv \int_{\Omega} |\nabla u|^2 dx dy \quad (4.16)$$

and  $\alpha_1$  and  $\alpha_2$  are positive parameters which measure the trade off between a good fit and the regularity of the solutions  $u$  and  $h$ .

Objective function  $f(u, h)$  in Equation (4.15) as a two-variables function is not convex and can hence have multiple solutions. You and Kaveh [129] observed that, for a fixed  $h$  (respectively  $u$ ),  $f(\cdot, h)$  (respectively  $f(u, \cdot)$ ) is a convex function of  $u$  (respectively  $h$ ) and they proposed the Alternating Minimization (AM) algorithm. There are two ways to run the AM algorithm with respect to the order of the fixed non-blind deconvolution. The two variants are:

#### **AMHU**

Given  $u^0$ :

1. Find minimizer  $h^k$  of  $f(u^{k-1}, h)$  by solving  $0 = \nabla_h f(u^{k-1}, h)$
2. Find minimizer  $u^k$  of  $f(u, h^k)$  by solving  $0 = \nabla_u f(u, h^k)$
3. If not convergence, go to 1.

#### **AMUH**

Given  $h^0$ :

1. Find minimizer  $u^{k-1}$  of  $f(u, h^{k-1})$  by solving  $0 = \nabla_u f(u, h^{k-1})$
2. Find minimizer  $h^k$  of  $f(u^{k-1}, h)$  by solving  $0 = \nabla_h f(u^{k-1}, h)$
3. If not convergence, go to 1.

More recently, Chan and Wong [11] have proposed using the *Total Variation* (TV) norm as the regularization term instead of  $H^1$  norm.

$$TV(u) \equiv \int_{\Omega} |\nabla u| dx dy \quad (4.17)$$

They observe that the AM algorithm converges much faster when TV regularization is used. Also, Chan and Wong analyzed the convergence of the AM algorithm with the  $H^1$  regularization and proved that the discrete version of the AM algorithm converges to a local minimizer for any given initial guess [12]. Besides the proof of convergence, they also proved that it is not necessary to explicitly impose the numerical conditions that  $u$  and  $h$  has to satisfy, e.g.,  $u \geq 0$ . These conditions can be embedded in the minimization steps of the AM algorithm, for instance, by simply setting  $u(x, y) = 0$  when its estimate is negative. Detailed proofs are beyond our scope in this thesis, and interested readers are referred to [12].

#### 4.3.4 Reflectance Parameter and Illumination Hemisphere Estimation through Alternating Minimization

Now let us return to our own problem, that of estimating the reflectance parameters and the illumination distribution from the specular images, which is formulated in Equation (4.13).

When setting up a joint regularization minimization formula to solve the blind deconvolution problem, we do not have to impose a regularization term for the blurring function, since we already know that the blurring function is a one dimensional Gaussian and because it is smooth with respect to its parameter. Also, as we have assumed that  $K_s$  is unique for all object surface points, we cannot estimate a meaningful value for  $K_s$ . Therefore, we will set  $K_s$  as a constant 1.0, The reason we leave  $K_s$  separate from  $L$  is so that we can have a controllable value to simulate the effect of changing the specular reflection strength. Thus, the minimization problem to estimate the reflectance parameter  $\sigma$  and the illumination distribution represented as a discretized hemisphere can be formulated as follows.

$$\min_{L, \sigma} f(L, \sigma) \equiv \min_{L, \sigma} \sum_k^{N_K} \sum_l^{N_L} \|I_S(s, t, k) - I(s, t, k)\|_{L^2}^2 + \rho \sum_l^{N_L} |\nabla L(\theta_l, \phi_l)| \quad (4.18)$$

To solve Equation (4.18), we apply the AM algorithm mentioned in the last section, with two exceptions described in the following sections.

### **M-estimator for Robustness**

Since we have obtained the observed values of the specular reflection,  $I(s, t, k)$  by subtracting the diffuse-texture-mapped images (estimated diffuse images) from the original input images, we have inherited errors generated in the diffuse reflection estimation. Thus, when estimating both the illumination distribution and the reflectance parameters from the residual specular images, we will have to ensure that the estimation procedure is robust against noise. For this purpose, we adopt the robust estimation technique based on M-estimator [93, 37]. Consequently, we minimize Equation (4.18) for each step in the AM algorithm through conjugate gradient with Lorentzian function [93, 37] as the M-estimator and golden section search for line minimization, similar to the minimization strategy we adopt for 3D-3D registration (Section 2.2).

### **Initial Estimation**

As observed in [11] with numerical results, estimating blurring functions without edges, e.g., Gaussian, with the AM algorithm happens to converge slowly as compared with estimating PSFs with edges, e.g., out-of-focus blur. One way to speed up this convergence rate is to give a good initial estimation of the blurring function. In our case, if we can utilize a good initial estimation of  $\sigma$ , instead of setting  $\sigma_0$  very small to simulate a delta function, and accomplish the AMUH algorithm with the initial illumination hemisphere estimated in Section 4.2.3, we can expect a fast convergence.

Then, the question is what is  $\sigma_{true}$ ? Fortunately, we know that most shiny objects that can be modeled with the microfacet BRDF reflection model have a  $\sigma$  around the order of 0.1. So it is fair to start with  $\sigma_0 = 0.1$ .

Also,  $\rho$  in Equation (4.18) is set to a small value ( $1.0 \times 10^{-6}$ ), not to impose too much smoothness on the illumination distribution.

## 4.4 View-dependent Rendering

After estimating the refined illumination hemisphere and the surface reflectance parameters, we can easily render synthetic images from arbitrary viewpoints by performing the following steps.

1. **Render diffuse image.** This can be easily accomplished by texture mapping the diffuse texture map, derived in Section 4.2.2, to the 3D model and rendering the 3D model from the given viewpoint.
2. **Compute a shadow map.** To correctly handle self occlusion on the object surface and attach self shadows, we need to know whether each point light source positioned on the illumination hemisphere is visible from the surface point we want to compute the specular reflection value. We currently implement this in a lazy shadow z-buffer [126] approach: render the 3D model from a viewpoint set to each point light source and then store the visible triangle patches. Note, however, that this should be done at a pixel level; we do this on a triangular patch base to reduce computational cost.
3. **Render specular image.** Using the shadow map, the estimated illumination distribution and the reflectance parameters, we can render a specular reflection image with Equation (4.7) and Equation (4.6).
4. **Add diffuse image and specular image.** By adding the two rendered images, the diffuse image and the specular image, we obtain the synthetic image rendered from the given viewpoint.

## 4.5 Relighting

Given the estimated illumination hemisphere and the reflection parameters, we can render images of the target object under a different lighting distribution from that of the original, with one exception: we are not able to change the color of the illumination distribution since the diffuse reflection will have a color that is a convolution of the spectral distribution of the illumination and the surface color.

Let us refer to the radiances of estimated original illumination environment represented as point light sources on the illumination hemisphere as  $\{L_l : l =$

$0, \dots, N_L\}$ . Note that these values are the radiances of each point light source scaled by  $K_s$ , and however, that they do not affect the following discussion since  $K_s$  will be canceled out. Under this point light source set, the diffuse reflection color vector at an object surface point  $v$  can be written down as,

$$\mathbf{I}_{D,v} = \mathbf{K}_{D,v} \sum_l^{N_L} M_{l,v} L_l \cos \theta_{l,v} \quad (4.19)$$

where  $\mathbf{K}_{D,v}$  is the diffuse reflection color of the object surface point and  $M_{l,v}$  is the shadow map mentioned in the last section, which accounts for the occlusion of the illuminations, that is, it takes 1 when  $L_l$  is visible from  $v$  and 0 when  $L_l$  is invisible from  $v$ .

Now, given a new illumination distribution as a set of point light sources on the illumination hemisphere,  $\{\widetilde{L}_l : l = 0, \dots, \widetilde{N}_L\}$ , the irradiance at object surface point  $v$  will be:

$$\widetilde{\mathbf{I}}_{D,v} = \mathbf{K}_{D,v} \sum_l^{\widetilde{N}_L} \widetilde{M}_{l,v} \widetilde{L}_l \cos \theta_{l,v}. \quad (4.20)$$

From Equation (4.19) and Equation (4.20), we can see that the ratio of the diffuse reflection can be easily computed by:

$$\frac{\mathbf{I}_{D,v}}{\widetilde{\mathbf{I}}_{D,v}} = \frac{\sum_l^{N_L} M_{l,v} L_l \cos \theta_{l,v}}{\sum_l^{\widetilde{N}_L} \widetilde{M}_{l,v} \widetilde{L}_l \cos \theta_{l,v}}. \quad (4.21)$$

Note that we do not have to know  $\mathbf{K}_{D,v}$ .

Thus, we can render a diffuse reflection image of the target object under a new illumination distribution from arbitrary viewpoints by performing the following steps:

1. **Compute the shadow maps.** We compute  $M$  and  $\widetilde{M}$  with the method described in the last section. The shadow maps need to be computed only once for each illumination distribution.
2. **Render the diffuse reflection image under the original illumination distribution.** This can be simply done by texture mapping the diffuse texture map we have derived in Section 4.2.2. Let us refer to this image as the original image.
3. **Compute the albedo image.** Given the original illumination distribution through the estimated illumination hemisphere and the pre-computed mask

map  $M$ , we can compute the total irradiance at surface point  $v$ ,  $\sum_l^{N_L} M_{l,v} L_l \cos \theta_{l,v}$ ; and by dividing each pixel value in the original image by this value, we can achieve a pseudo albedo image, which represents the spatial distribution of  $\frac{1}{K_S} \mathbf{K}_D$  from the given viewpoint. Pixels that are black because of shadows will be filled with neighboring albedo values.

4. **Render the relit diffuse image.** Given the new illumination hemisphere and the pre-computed shadow map  $\widetilde{M}$ , we can compute  $\sum_l^{N_L} \widetilde{M}_{l,v} \widetilde{L}_l \cos \theta_{l,v}$  and multiply these values to each object surface point; we can then obtain a relit diffuse image of the target object from the given viewpoint.

By rendering a specular reflection image as explained in the last section, and then compositing it with the diffuse reflection image rendered in the above procedure, we can obtain a relit object image from arbitrary viewpoints.

## 4.6 Results

We applied our framework to model the appearance of a real object from a sparse set of images, and render new views from arbitrary viewpoints possibly with changing reflectance parameters and/or under novel illumination conditions.

### 4.6.1 Inputs

The object we used in our experiment is a mask made in Bali. Five range images of the mask were scanned with a Vivid 900 [72]. They were registered and fused to produce a 3D mesh model.

Additionally, we took 6 high dynamic range (HDR) images [24, 73] and used them as the input image sequence. A HDR image is obtained by combining several photographs taken with different shutter speeds. Figure 4.2 depicts the six input images with the shutter speed approximately set to  $\frac{1}{30}$ . The number of input images is determined by roughly considering how much the highlights overlapped between each image, meaning that the diffuse texture map separation determines the necessary number of images (see Section 4.2.2). As can be seen in Figure 4.2, three point light sources were placed above the mask.



Figure 4.2: Six input HDR images of the Bali mask.

To estimate the viewpoint of each input image, we captured a range image at the same viewpoint with a light-stripe range finder, and carried out 3D-3D alignment between the range data and the 3D mesh model with the method we presented in Section 2.2.

#### 4.6.2 Estimated parameters

First the diffuse texture map which represents the diffuse reflection of the object surface was computed by taking the minimum pixel values at each object surface point. Figure 4.3 depicts the diffuse-texture-mapped images viewed from two of the viewpoints in the input image sequence. Note that the specular reflection is successfully removed. By subtracting these diffuse-texture-mapped images from the original images one by one, we obtain a specular reflection image se-



Figure 4.3: Diffuse-texture-mapped images.

quence which we use to estimate the reflection parameters and the illumination distribution. Figure 4.4 depicts one of the input images separated into two images: a diffuse image and a specular image. Unfortunately, slight misalignment in 3D-3D registration for viewpoint estimation cannot be avoided and it results in erroneous estimation of the diffuse texture map, as can be observed in the blurring effect of the mark on the mask's forehead. Although these errors will be inherited by the specular reflection image, as can be seen in Figure 4.4, we are able to estimate the reflection parameters and the illumination distribution robustly since we adopt the robust estimation framework as mentioned in Section 4.3.4.

By shooting each pixel value in the specular image sequence back in the perfect mirror direction and then mapping those values to a hemisphere with a pre-determined radius (in this case  $160cm$ ), we obtain partial illumination hemispheres. The left and middle image of Figure 4.5 show two of these partial illumination hemispheres as viewed from the top. By combining these partial illumination hemispheres, we obtain an initial illumination hemisphere, as shown



Figure 4.4: Separation of reflection components. Left: original input image, Middle: diffuse reflection image (diffuse-texture-mapped image), Right: specular reflection image.

in the right image of Figure 4.5. Note that the errors in the partial illumination hemisphere are mostly removed by the voting strategy described in Section 4.2.3 and that three clusters corresponding to the three point light sources are visible.

Through the Alternating Minimization (AM) algorithm with Total Variation (TV) regularization described in Section 4.3.4, we estimated the lighting distribution as the illumination hemisphere and the reflection parameter  $\sigma$ . Figure 4.6 depicts the estimated final illumination hemisphere. To compare the estimated illumination distribution with the true illumination distribution, we captured the lighting condition with a fish-eye lens mounted CCD camera, as shown in Figure 4.6. Since we did not do any geometric calibration to enable direct comparison between the two images of the illumination distribution, we cannot see whether they match precisely. However, it is obvious that the estimated illumination hemisphere has an illumination distribution very close to the positions of the three point light sources. The intensity of the brightest point light source in the illumination hemisphere was 1370.83 and that of the darkest was 0.65. Note that these values are a combination of  $K_s$  and  $L_i$ . By starting with 0.1 for  $\sigma$ , we obtained 0.075 as its final estimated value.

Figure 4.7 shows real and synthetic images from three of the input viewpoints. The right row visualizes the difference of pixel intensity values, where



Figure 4.5: Left and Middle: partial illumination hemispheres corresponding to two viewpoints in the original input image sequence. Right: the combined initial illumination hemisphere.



Figure 4.6: Left: Estimated illumination hemisphere, Right: An photograph of the true illumination distribution captured through a fish-eye lens and flipped horizontally.

blue ranges approximately from 0 to 50, green ranges from 150 to 200 and red ranges 200 to 255, same to the relation between RGB color and wavelength. While slight difference arises due to the difference of the input 3D model from the true geometry of the target object, the rendered images are consistent to these geometric differences, and consequently it does not effect the visual consistency, which makes the synthetic images appear realistic. The average of RMS errors was 12.5.

### 4.6.3 View-dependent Rendering

With the estimated illumination hemisphere and the reflection parameter, we are able to accomplish photorealistic view-dependent rendering from arbitrary viewpoints. Figure 4.8 shows the result of rendering the target object from a novel viewpoint. The left image in Figure 4.8 is a real photograph taken from the same viewpoint but not used in the input image sequence. As can be seen, the result is almost indistinguishable from the real photograph. The slight difference is mainly due to lack of perfection in the 3D model and consequent difference in computed normals. The RMS error was 16.1.

### 4.6.4 Changing Reflection Parameters

We can also render images of the target object with different reflection parameters. As an example, Figure 4.9 displays four synthetic images rendered with different roughness ( $\sigma$ ) and different  $K_s$  (Note that the estimated roughness is  $\sigma = 0.075$ ). Note that, as we have mentioned in Section 4.3.4,  $K_s$  cannot be decoupled from the radiance of the illumination; consequently, we have estimated the illumination distribution with  $K_s = 1.0$  and we can only relatively change its value.

### 4.6.5 Relighting

Besides rendering from arbitrary viewpoints, we can render images under arbitrary novel illumination conditions.

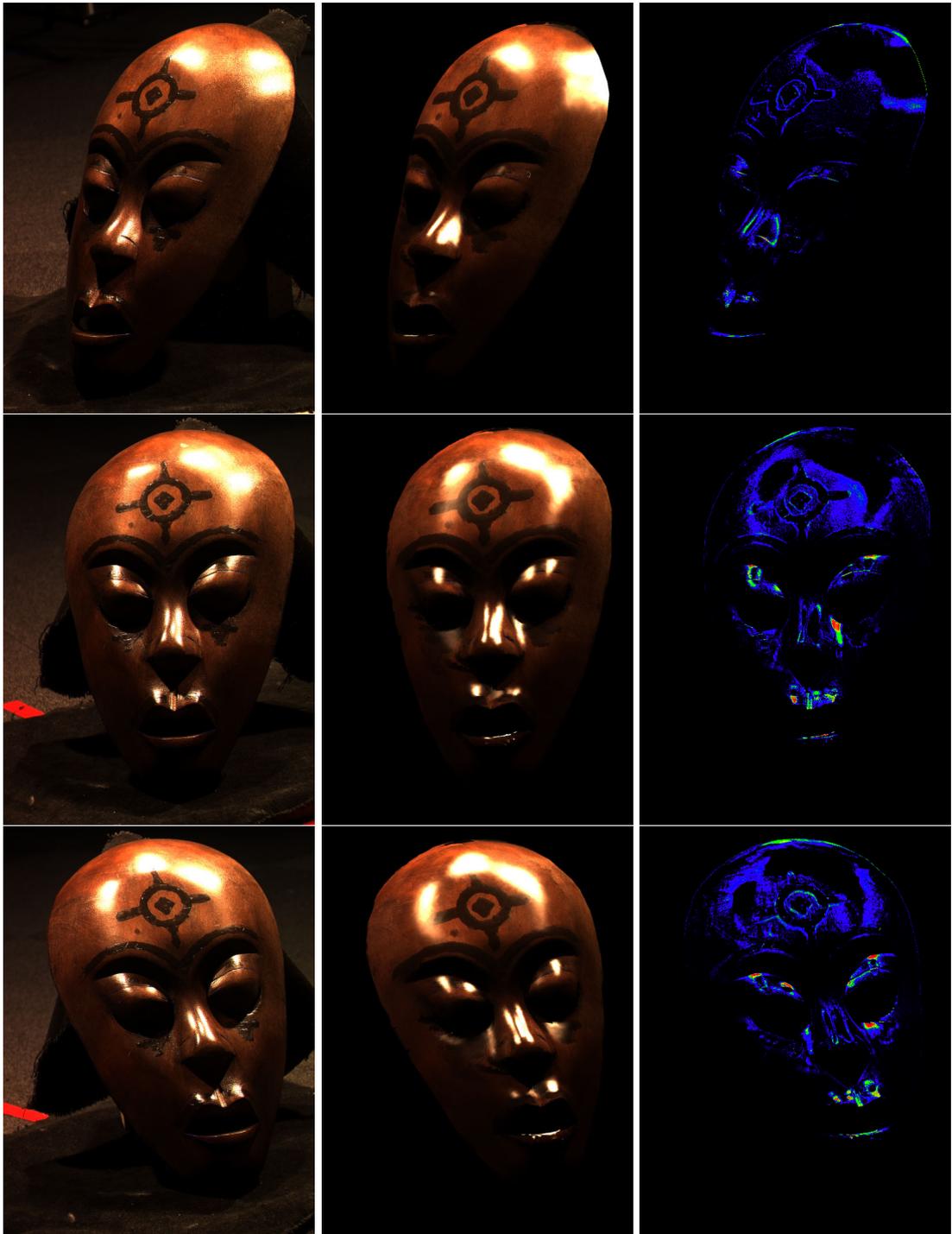


Figure 4.7: Comparison on input images. Left: input real images, Middle: synthesized images, Right: difference



Figure 4.8: Left: real photograph, Middle: synthetic image, Right: difference.

### Turning off lights

By masking the estimated illumination hemisphere, we can easily obtain illumination hemispheres corresponding to each real point light source. Figure 4.10 shows real photographs and rendered synthetic images with each of the three point light sources turned on separately.

### Moving Lights

We can render synthetic images of the target object under arbitrary illumination conditions. Figure 4.11 depicts some frames from a movie we have made with moving point light sources. While the light source on the right hand side remains static, another point light source moves from the left hand side to the right hand side. Note that the diffuse reflection is also rendered correctly depending on the illumination condition.

## 4.7 Summary

We have presented a framework to accomplish photorealistic view-dependent rendering and relighting from a sparse set of images and a pre-acquired geometric model of the target object. We take an approach similar to inverse rendering, except that we estimate all three components necessary for forward rendering:



Figure 4.9: Left Top:  $\sigma = 0.1$   $K_s = 1.0$ , Right Top:  $\sigma = 0.04$   $K_s = 1.0$ , Left Bottom:  $\sigma = 0.075$   $K_s = 2.0$ , Right Bottom:  $\sigma = 0.075$   $K_s = 0.3$

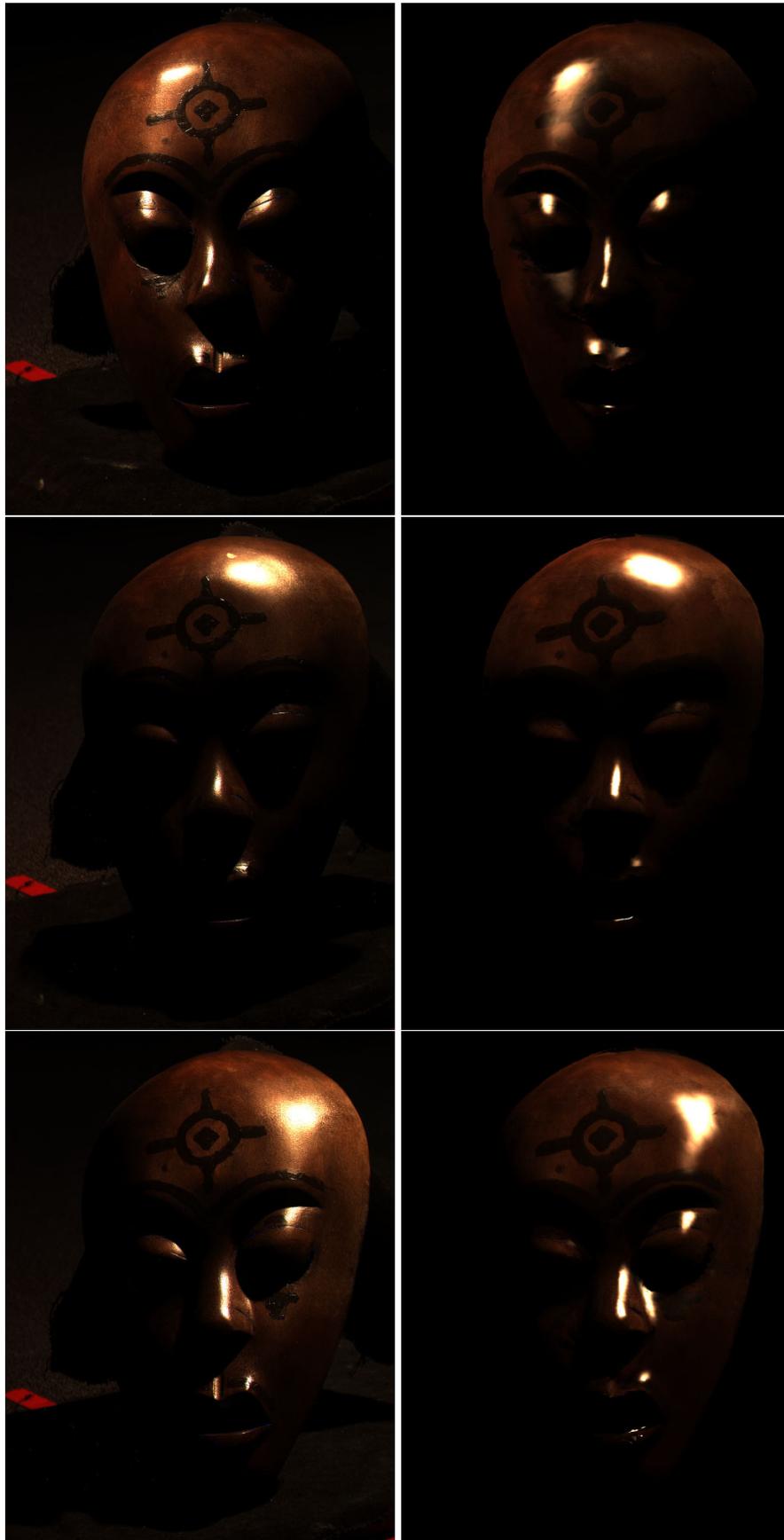


Figure 4.10: Left Column: real photographs, Right Column: synthetic images.

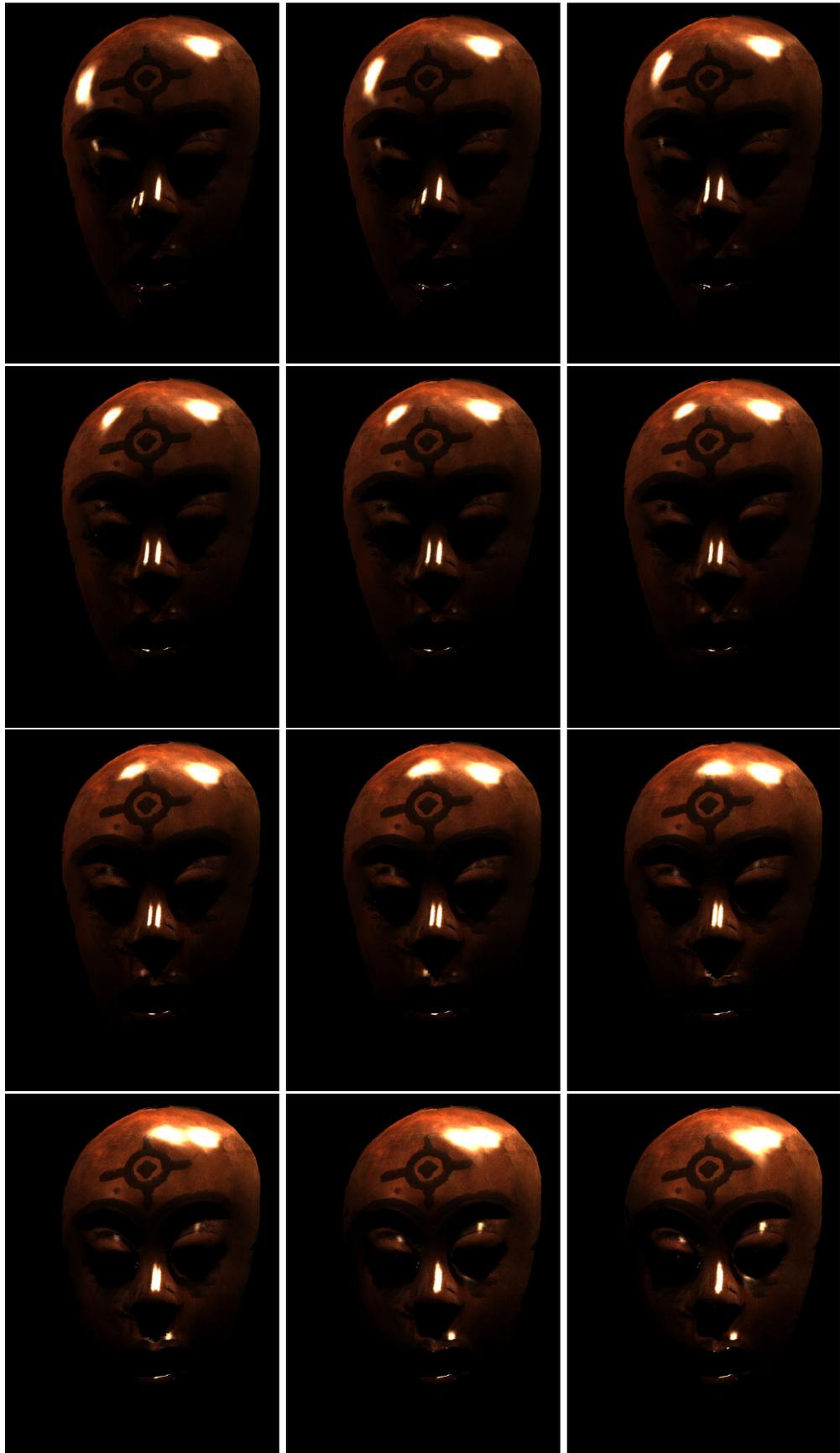


Figure 4.11: Synthetic images rendered under a moving light source.  
103

texture, BRDF and illumination distribution. The key idea is to first separate the reflection components to obtain a texture map that represents the diffuse reflection and images whose pixel values' dominant factor is specular reflection. Then the specular images are used to recover the illumination distribution and reflectance parameters of a simplified Torrance-Sparrow reflection model via 2D blind deconvolution with regularization term imposed on the illumination distribution. We have explained our framework in theory, and have presented results applying the framework to a real world object.

# Chapter 5

## Conclusions

### 5.1 Summary

In this dissertation, we have explored *photometric object modeling*, whose aim is to digitally represent real world objects in the computer for photorealistic display of them. To achieve this goal, we have developed two new methods: *Eigen-Texture Rendering* and *Rendering from a Sparse Set of Images*. Both methods build up an efficient representation of the virtual object for view-dependent rendering and relighting, taking a set of photographs as the input. Also, the underlying philosophy of both methods is to take full advantage of knowledge of the geometry of the target object.

For rendering from a dense set of images, a dense sampling of the target object's appearance variation under changing lighting and viewing conditions, we developed *Eigen-Texture Rendering*. By handling and compressing the appearance variation on the 2D surface of the target object, we can successfully render synthetic images from arbitrary viewpoints and under novel lighting environments while preserving the non-rigid appearance variation. By elevating the correlation of appearance variation by treating them based on their physical location on the object surface and by applying PCA, the resulting representation of the virtual object becomes compact as compared with previous image-based approaches. Furthermore, by interpolation in eigenspaces, we are able to decrease the number of necessary input images, thereby increasing the flexibility of the system. Since the method does not make any assumptions on the surface

reflection property, a wide variety of objects can be handled in the framework, as we have seen demonstrated by the experimental results. Also, usage of the geometric model allows us to easily superimpose virtual objects into real images with correct geometric visuality.

To drastically reduce the necessary number of input images for another order, we developed *Rendering from a Sparse Set of Images*. Taking a sparse set of images and a pre-acquired geometric model of the target object, we recover the illumination distribution as a hemisphere covering the object and the parameters of the simplified Torrance-Sparrow reflection model, thereby enabling view-dependent rendering from arbitrary viewpoints and relighting under novel illumination conditions. The problem of recovering the illumination distribution and reflection parameters simultaneously is formulated as a 2D blind deconvolution problem and solved by Alternating Minimization (AM) algorithm with TV regularization. Unlike previous inverse rendering approaches, ours enables recovery of all three unknowns, namely texture, BRDF and lighting from the observation of real objects, thereby increasing the flexibility of the system. The resulting representation of the virtual object is very compact – a 3D model of the object, the illumination hemisphere and reflection parameters.

With regard to the geometric models we use in both methods, we proposed two new methods in the geometric modeling pipeline. For registration, we proposed a framework to simultaneously register multiple range images. The simultaneous registration problem is redefined as a least-square problem with an objective function globally constructed with respect to each range image. For efficiency, we employ a k-d tree structure for fast point correspondence search and apply a conjugate gradient search in minimizing the least-square problem for faster convergence. For robustness, we employ photometric attributes of raw 3D points, and search for “better” point mates based on their distance.

For integration, we proposed a range image integration framework which can construct 3D models with photometric attributes. By taking a consensus of appearance changes of the target object from multiple range images, we reconstruct the 3D model with an appearance which successfully discards outliers caused by noise. Also, we can provide a model with Lambertian reflected light values by discarding specular reflections as outliers. We also proposed an algorithm for constructing a 3D model in an efficient representation. Considering

the surface curvature and the photometric attributes, we constructed 3D models that have higher detail in surface areas that contain either high curvature or significant variation of appearance. Thus, we could efficiently use computational resources. Both methods take special care of their robustness against noise, by means of M-estimation in registration and consensus surface extraction in integration. Experimental results, including application for digital preservation of cultural heritage objects, prove the robustness and effectiveness of these methods.

## 5.2 Contributions

The specific contributions of this thesis are:

1. A new framework to reconstruct a efficient representation of the appearance of real world objects from a dense sampling of its variation: The key idea is to make full use of the underlying geometry of the target object and handle appearance variation on the object surface to elevate the correlation for efficient compression via PCA.
2. A new framework to reconstruct a compact representation of appearance of real world objects from a sparse sampling of its variation: By simultaneously recovering the texture, BRDF approximation and lighting conditions from the input images and a 3D model, we are able to accomplish rendering from arbitrary viewpoints and relight under novel illumination conditions. The key idea is to first decompose the input images into two sets of images, diffuse images and specular images, and then formulate the factorization of illumination and reflection effects from specular images as a 2D blind deconvolution problem.
3. Robust simultaneous registration of multiple range images: By performing M-estimation over an error function common for all range images, we simultaneously register multiple range images. Also, photometric attributes of raw 3D points, e.g., intensity and laser reflectance strength, can be incorporated in the registration framework for further robustness and faster convergence. The implementation can be done in a parallel manner.

4. Robust and adaptive integration of multiple range images: By adaptively subdividing voxels depending on the curvature or/and the photometric variation of the object surface, we can reconstruct an efficient representation of the target object. Special care is taken for robustness against noise by taking a consensus of the range image points, and it can be executed in a parallel manner.

### 5.3 Future Work and Discussion

We conclude with a discussion of open problems and future improvements which we are interested in pursuing.

#### **Effective sampling in simultaneous range image registration**

In early stages of the simultaneous registration of multiple range images, when the range images are distributed apart, we subsample the range images to reduce the number of points used in the registration framework to achieve significant speed-up. While conducting many experiments using our framework, we found that reducing the number of data points to be used in registration does not necessarily affect the accuracy of the final registration result. According to the principle of statistics, the more data points we use when minimizing the global error function, the higher the accuracy will be. However, as long as we use range finders that will always return data points that have some noise embedded in their values, the minimization of the error function will have a lower bound that we will not be able to cross. Hence, it should be necessary and meaningful to estimate the accuracy limitation that we can obtain, and use as few data points as possible to achieve that bound for efficient registration. Although we have not fully explored this idea, preliminary experiments showed us that using around only 5% of the points used in registering the Great Buddha in Kamakura gives us almost the same accuracy as using all the points we scanned. Our guess for the best sampling is to subsample points based on the derivative of the surface curvature. In this sense, the framework for adaptive subdivision of voxels in our range image integration framework may be useful for building a subsampled range image.

### **On the fly integration of multiple pre-registered range images**

We have developed a fully scalable range image integration method, and the next step would be to aim real-time. One interesting idea might be to embed the integration framework in a geometric model viewer, and accomplish the integration of multiple pre-registered range images gradually while the user is viewing the volumetric representation of the range images. We believe that the subdivision framework based on oct-tree data structure and consensus surface algorithm in our integration framework has the potential to achieve this goal.

### **Further compression in a global eigenspace**

While we currently normalize each *cell image* corresponding to each triangular patch to have the same triangular shape in *Eigen-Texture Rendering*, with the exception that it makes it easy to implement the system, theoretically there is no strong reason not to treat each triangular patch in a different size. As well as making it possible to have different sized *cell image sequences*, it would also be very interesting to construct a global eigenspace for extracting the principle components across different *cell image sequences*. To maximize the correlation inside and across the combined *cell image sequences*, it would be better to use a 3D model that is subdivided with regard to its texture variation and to combine *cells* that have almost the same underlying texture in their appearance variation. Our adaptive integration framework can be used to construct a good basis for this purpose.

### **Geometry refinement in *Rendering from a Sparse Set of Images***

Now that we have developed a scheme to recover the texture, lighting and BRDF from a sparse set of images and a geometric model, we are also interested in recovering the geometry from the set of images simultaneously to the other unknowns. While it would be extremely difficult to reconstruct the geometry from nothing, it would still be possible to start from a rough geometry and refine it gradually. The initial rough geometry can be provided by well studied computer vision techniques, such as stereo and structure from motion. As well as estimating the illumination radiance and reflection parameters, probably the surface normals can be estimated to well explain the input images, and the surface normals may impose constraints on the object geometry for refinement.

### **Recovering the spatio-spectral distribution of the illumination environment**

In our *Rendering from a Sparse Set of Images* framework, we recover the spatial radiance distribution of illumination environment as an *illumination hemisphere*. While we can accomplish relighting with this estimated illumination hemisphere by changing the distribution of the light sources and changing the radiance of each point light sources, we cannot change the color of the light sources; it is fixed to the average color of the original illumination environment. To relight the target object with novel illumination colors, we have to recover the spectral distribution for each point light source placed on the vertices of the illumination hemisphere. Once we have estimated this spatio-spectral distribution of the original illumination environment, we are able to estimate the true RGB albedo of the object surface for diffuse reflection. We think color constancy approaches [30, 31, 32, 116] may provide very powerful means to accomplish this estimation.

## References

- [1] ADELSON, E., AND BERGEN, J. *The Plenoptic Function and the Elements of Early Vision*. Computational Models of Visual Processing, M.Landy and J.A.Movshon(eds),MIT Press, 1991, pp. 3–20.
- [2] BENJEMMA, R., AND SCHMITT, F. Fast global registration of 3d sampled surfaces using a multi-z-buffer technique. In *Proc. Int. Conf. On Recent Advances in 3-D Digital Imaging and Modeling* (May 1997), pp. 113–120.
- [3] BERGEVIN, R., SOUCY, M., GAGNON, H., AND LAURENDEAU, D. Towards a general multi-view registration technique. *IEEE Trans. Patt. Anal. Machine Intell.* 18, 5 (May 1996), 540–547.
- [4] BERNARDINI, R., AND RUSHMEIER, H. The 3d model acquisition pipeline. In *Eurographics 2000 State of the Art Report (STAR)* (Aug. 2000).
- [5] BESL, P., AND MCKAY, N. A method for registration of 3-d shapes. *IEEE Trans. Patt. Anal. Machine Intell.* 14, 2 (Feb 1992), 239–256.
- [6] BLAIS, G., AND LEVINE, M. Registering multiview range data to create 3d computer objects. *IEEE Trans. Patt. Anal. Machine Intell.* 17, 8 (Aug 1995), 820–824.
- [7] BOIVIN, S., AND GAGALOWICZ, A. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *Computer Graphics Proceedings, ACM SIGGRAPH 01* (Aug. 2001), pp. 197–116.
- [8] BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S., AND COHEN, M. Unstructured Lumigraph Rendering. In *Computer Graphics Proceedings, ACM SIGGRAPH 01* (Aug. 2001), pp. 425–432.

- [9] CHAI, J., KANG, S., AND SHUM, H. Rendering with Non-uniform Concentric Mosaics and Invariable Input Sampling. In *Proc. of ECCV 2000 Workshop SMILE 2000* (Jul. 2000), pp. 94–108.
- [10] CHAI, J., TONG, X., CHAN, S.-C., AND SHUM, H. Plenoptic Sampling. In *Computer Graphics Proceedings, ACM SIGGRAPH 00* (Jul. 2000), pp. 307–318.
- [11] CHAN, T., AND WONG, C. Total Variation Blind Deconvolution. *IEEE Transactions on Image Processing* 7 (Mar. 1998), 370–375.
- [12] CHAN, T., AND WONG, C. Convergence of the Alternating Minimization Algorithm for Blind Deconvolution. *Linear Algebra and its Applications* 316, 1–3 (2000), 259–285.
- [13] CHEN, S., AND WILLIAMS, L. View interpolation for image synthesis. In *Computer Graphics Proceedings, ACM SIGGRAPH 93* (Aug. 1993), pp. 279–288.
- [14] CHEN, W.-C., GRZESZCZUK, R., AND BOUGUET, J.-Y. Light Field Mapping: Hardware-Accelerated Visualization of Surface Light Field. In *ACM SIGGRAPH 01 Course Notes, Course 46 Acquisition and Visualization of Surface Light Fields* (Aug. 2001).
- [15] CHEN, Y., AND MEDIONI, G. Object modeling by registration of multiple range images. *Image and Vision Computing* 10, 3 (Apr 1992), 145–155.
- [16] CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. Metro: measuring error on simplified surfaces. *Computer Graphics Forum* 17, 2 (June 1998), 167–174.
- [17] CULBERTSON, W. B., MALZBENDER, T., AND SLABAUGH, G. Generalized voxel coloring. In *Vision Algorithms: Theory and Practice* (sep 1999), B. Triggs, A. Zisserman, and R. Szeliski, Eds., no. 1883 in LNCS, Springer-Verlag, pp. 100–115.
- [18] CURLESS, B., AND LEVOY, M. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH'96* (1996), ACM, pp. 303–312.

- [19] CYBERWARE. <http://www.cyberware.com>.
- [20] CYRA. <http://www.cyra.com>.
- [21] DANA, K., GINNEKEN, B., NAYAR, S., AND KOENDERINK, J. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics* 18, 1 (Jan. 1999), 1–34.
- [22] DEBEVEC, P. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *Computer Graphics Proceedings, ACM SIGGRAPH 98* (Jul. 1998), pp. 189–198.
- [23] DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. Acquiring the Reflectance Field of a Human Face. In *Computer Graphics Proceedings, ACM SIGGRAPH 00* (Jul. 2000), pp. 145–156.
- [24] DEBEVEC, P., AND MALIK, J. Recovering High Dynamic Range Radiance Maps from Photographs. In *Computer Graphics Proceedings, ACM SIGGRAPH 97* (Aug. 1997), pp. 369–378.
- [25] DEBEVEC, P., TAYLOR, C., AND MALIK, J. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Computer Graphics Proceedings, ACM SIGGRAPH 96* (Aug. 1996), pp. 11–20.
- [26] DEBEVEC, P., YU, Y., AND BORSHUKOV, G. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In *Proc. of 9th Eurographics Rendering Workshop* (Jun. 1998), pp. 105–116.
- [27] DORAI, C., WANG, G., JAIN, A., AND MERCER, C. From images to models: Automatic 3d object model construction from multiple views. In *Proc. of the 13th IAPR International Conference on Pattern Recognition* (1996), pp. 770–774.
- [28] EGGERT, D., FITZGIBBON, A., AND FISHER, R. Simultaneous registration of multiple range views for use in reverse engineering. Tech. Rep. 804, Dept. of Artificial Intelligence, University of Edinburgh, 1996.

- [29] FAUGERAS, O., AND HEBERT, M. The representation, recognition, and locating of 3-d objects. *International Journal of Robotic Research* 5, 3 (Fall 1986), 27–52.
- [30] FINLAYSON, G., AND HORDLEY, S. Color constancy at a pixel. *Journal of the Optical Society of America A* 18, 2 (Feb. 2001), 253–264.
- [31] FINLAYSON, G., HORDLEY, S., AND HUBEL, P. Color by Correlation: A Simple, Unifying Framework for Color Constancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (Nov. 2001), 1209–1221.
- [32] FINLAYSON, G., AND SCHAEFER, G. Solving for Colour Constancy using a Constrained Dichromatic Reflection Model. *International Journal of Computer Vision* 42, 3 (2001), 127–144.
- [33] FRIEDMAN, J., BENTLEY, J., AND FINKEL, R. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. On Mathematical Software* 3, 3 (1977), 209–226.
- [34] FRIEDMAN, J. H., BENTLEY, J., AND FINKEL, R. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software* 3, 3 (1977), 209–226.
- [35] FRISKEN, S., PERRY, R., ROCKWOOD, A., AND JONES, T. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Proc. SIGGRAPH2000* (July 2000), ACM, pp. 249–254.
- [36] GARLAND, M., AND HECKBERT, P. Simplifying surfaces with color and texture using quadric error metrics. In *Proc. IEEE Visualization 1998* (1998).
- [37] GILL, P., MURRAY, W., AND WRIGHT, M. *Practical Optimization*. Academic Press, 1981.
- [38] GODIN, G., RIOUX, M., AND BARIBEAU, R. Three-dimensional registration using range and intensity information. In *Proc. SPIE vol.2350: Vision-metrics III* (1994), pp. 279–290.
- [39] GORTLER, S., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. The Lumigraph. In *Computer Graphics Proceedings, ACM SIGGRAPH 96* (Aug. 1996), pp. 43–54.

- [40] HEIGL, B., KOCH, R., POLLEFEYS, M., DENZLER, J., AND GOOL, L. V. Plenoptic modeling and rendering from image sequences taken by hand-held camera. In *Proc. of DAGM '99* (1999), pp. 94–101.
- [41] HILTON, A., STODDART, A., ILLINGWORTH, J., AND WINDEATT, T. Reliable surface reconstruction from multiple range images. In *Proceedings of European Conference on Computer Vision* (Springer-Verlag, 1996), pp. 117–126.
- [42] HOPPE, H. Progressive meshes. In *Computer Graphics (SIGGRAPH 1996 Proceedings)* (1996), pp. 99–108.
- [43] HOPPE, H. New quadric metric for simplifying meshes with appearance attributes. In *Proc. IEEE Visualization 1999* (1999), pp. 59–66.
- [44] HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH'92* (1992), ACM, pp. 71–78.
- [45] HORN, B. *Robot Vision*. McGraw-Hill Book Company, 1986.
- [46] IKEUCHI, K., AND SATO, K. Determining Reflectance Properties of an Object Using Range and Brightness Images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 13, 11 (Nov. 1991), 1139–1153.
- [47] IKEUCHI, K., SATO, Y., NISHINO, K., SAGAWA, R., NISHIKAWA, T., OISHI, T., SATO, I., TAKAMATSU, J., AND MIYAZAKI, D. Modeling Cultural Heritage through Observation. In *Proc. of IEEE First Pacific-Rim Conference on Multimedia* (2000).
- [48] IKEUCHI, K., SATO, Y., NISHINO, K., AND SATO, I. *Mixed Reality -Merging Real and Virtual World-*. Y. Ohta and H. Tamura, Eds., Ohmsha, ISBN 3-540-65623-5, 1999, ch. 8: Photometric Modeling for Mixed Reality, pp. 147–163.
- [49] IKEUCHI, K., SATO, Y., NISHINO, K., AND SATO, I. Modeling from Reality: Photometric Aspect. *Transactions of the Virtual Reality Society of Japan* 4, 4 (Dec. 1999), 623–630.

- [50] IKEUCHI, K., Y.SATO, K.NISHINO, AND I.SATO. Appearance Modeling for Mixed Reality: Photometric Aspect. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics SMC '99* (Oct. 1999), vol. VI, pp. 36–41.
- [51] ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. Dynamically Reparameterized Light Fields. In *Computer Graphics Proceedings, ACM SIGGRAPH 00* (Jul. 2000), pp. 297–306.
- [52] JIN, H., DUCHAMP, T., HOPPE, H., MCDONALD, J., PULLI, K., AND STUETZLE, W. Surface reconstruction from misregistered data. In *Proc. SPIE vol.2573: Vision Geometry IV* (1995), pp. 324–328.
- [53] JOHNSON, A. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Aug 1997.
- [54] JOHNSON, A., AND HEBERT, M. Surface registration by matching oriented points. In *Proc. Int. Conf. On Recent Advances in 3-D Digital Imaging and Modeling* (May 1997), pp. 121–128.
- [55] JOHNSON, A., AND KANG, S. Registration and integration of textured 3-d data. In *Proc. Int. Conf. On Recent Advances in 3-D Digital Imaging and Modeling* (May 1997), pp. 234–241.
- [56] KLINKER, G., SHAFER, S., AND KANADE, T. A Physical Approach to Color Image Understanding. *International Journal of Computer Vision* 4 (1990), 7–38.
- [57] KURAZUME, R., NISHINO, K., ZHANG, Z., AND IKEUCHI, K. Simultaneous 2D Images and 3D Geometric Model Registration for Texture Mapping Utilizing Reflectance Attribute. In *Proc. of Fifth Asian Conference on Computer Vision, ACCV02* (Jan. 2002), pp. 99–106.
- [58] KUTULAKOS, K., AND S.M.SEITZ. What Do N Photographs Tell Us about 3D Shape? Tech. Rep. 680, Computer Science Dept. U. Rochester, Jan. 1998.

- [59] LAFORTUNE, E., FOO, S.-C., TORRANCE, K., AND GREENBERG, D. Non-Linear Approximation of Reflectance Functions. In *Computer Graphics Proceedings, ACM SIGGRAPH 97* (Aug. 1997), pp. 117–126.
- [60] LAMBERT, J. *Photometria sive de mensura de gradibus luminis colorum et umbrae*. Augsburg, 1960.
- [61] LENSCH, H., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. Image-Based Reconstruction of Spatially Varying Materials. In *Proceedings of Eurographics Rendering Workshop* (Jun. 2001), pp. xx–xx.
- [62] LEUNG, T., AND MALIK, J. Recognizing surfaces using three-dimensional textures. In *Proc. of International Conference on Computer Vision '99* (1999), vol. 2, pp. 1010–1017.
- [63] LEVOY, M., AND HANRAHAN, P. Light Field Rendering. In *Computer Graphics Proceedings, ACM SIGGRAPH 96* (Aug. 1996), pp. 31–42.
- [64] LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. The Digital Michelangelo Project: 3D Scanning of Large Statues. In *Computer Graphics Proceedings, ACM SIGGRAPH 00* (Jul. 2000), pp. 131–144.
- [65] LORENSEN, W., AND CLINE, H. Marching cubes: a high resolution 3d surface construction algorithm. In *Proc. SIGGRAPH'87* (1987), ACM, pp. 163–170.
- [66] LU, R., KOENDERINK, J., AND KAPPERS, A. Optical properties (bidirectional reflection distribution functions) of velvet. *Applied Optics* 37, 25 (1998), 5974–5984.
- [67] MARSCHNER, S. *Inverse Rendering for Computer Graphics*. PhD thesis, Cornell University, 1998.
- [68] MARSCHNER, S., AND GREENBERG, D. Inverse Lighting for Photography. In *Proc. of IS&T/SID Fifth Color Imaging Conference* (Nov. 1997), pp. 262–265.

- [69] MARSCHNER, S., WESTIN, S., LAFORTUNE, E., AND TORRANCE, K. Image-Based Measurement of the Bidirectional Reflectance Distribution. *Applied Optics* 39, 16 (2000), 2592–2600.
- [70] MASUDA, T., SAKAUE, K., AND YOKOYA, N. Registration and integration of multiple range images for 3-d models construction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (Jun 1996), pp. 879–883.
- [71] MILLER, G., RUBIN, S., AND PONCELEON, D. Lazy Decompression of Surface Light Fields for Precomputed Global Illumination. In *Eurographics Rendering Workshop* (Jun. 1998), pp. 281–292.
- [72] MINOLTA. *Vivid 900*, 2001. <http://www.minolta-rio.com/vivid/>.
- [73] MITSUNAGA, T., AND NAYAR, S. Radiometric Self Calibration. In *Proc. of Computer Vision and Pattern Recognition '99* (Jun. 1999), pp. 374–380.
- [74] MIYAZAKI, D., OISHI, T., NISHIKAWA, T., SAGAWA, R., NISHINO, K., TOMOMATSU, T., TAKASE, Y., AND IKEUCHI, K. The Great Buddha Project: Modelling Cultural Heritage through Observation. In *Proc. of 6th International Conference on Virtual Systems and MultiMedia VSMM2000* (Oct. 2000), pp. 138–145.
- [75] MIYAZAKI, D., OISHI, T., NISHIKAWA, T., SAGAWA, R., NISHINO, K., TOMOMATSU, T., TAKASE, Y., AND IKEUCHI, K. The Great Buddha Project: Modelling Cultural Heritage through Observation. In *6th International Conference on Virtual Systems and MultiMedia VSMM2000* (Oct. 2000), pp. 138–145.
- [76] MOAYERI, N., AND KONSTANTINIDES, K. An Algorithm for Blind Restoration of Blurred and Noisy Images. Tech. Rep. HPL-96-102, HP Laboratories, Jul. 1996.
- [77] MURASE, H., AND NAYAR, S. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision* 14 (1995), 5–24.

- [78] NAEMURA, T., YANAGISAWA, T., KANEKO, M., AND HARASHIMA, H. Efficient projection of ray space for a ray based description of 3-D real world. In *IEICE IE95-119 (in Japanese)* (Feb. 1996).
- [79] NAYAR, S., IKEUCHI, K., AND KANADE, T. Determining Shape and Reflectance of Hybrid Surfaces by Photometric Sampling. *IEEE Trans. on Robotics and Automation* 6, 4 (1990), 418–431.
- [80] NAYAR, S., IKEUCHI, K., AND KANADE, T. Surface reflection: physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 7 (Jul. 1991), 611 – 634.
- [81] NAYAR, S., AND OREN, M. Generalization of the lambertian model and implications for machine vision. *International Journal of Computer Vision* 14 (1995), 227–251.
- [82] NAYAR, S., AND OREN, M. Generalization of the Lambertian Model and Implications for Machine Vision. *International Journal of Computer Vision* 14 (1995), 227–251.
- [83] NENE, S., NAYAR, S., AND MURASE, H. Columbia Object Image Library (COIL-100). Tech. Rep. CUCS-006-96, Columbia University, Computer Science Dept., Feb. 1996.
- [84] NEUGEBAUER, P. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proc. Int. Conf. on Shape Modeling and Application* (Mar 1997), pp. 130–139.
- [85] NICODEMUS, F., RICHMOND, J., HSIA, J., GINSBERG, I., AND LIMPERIS, T. Geometric Considerations and Nomenclature for Reflectance. National Bureau of Standards (US), 1977.
- [86] NISHINO, K., AND IKEUCHI, K. Robust Simultaneous Registration of Multiple Range Images Comprising A Large Number of Points. In *Proc. of IPSJ Computer Vision and Image Media* (May 2001), no. 36 in 2001, pp. 1–8.
- [87] NISHINO, K., AND IKEUCHI, K. Robust Simultaneous Registration of Multiple Range Images. In *Proc. of Fifth Asian Conference on Computer Vision, ACCV 02* (Jan. 2002), pp. 454–461.

- [88] NISHINO, K., I.SATO, Y.SATO, AND K.IKEUCHI. Eigen-Texture Method: Appearance Compression and Synthesis based on 3D Model for Mixed Reality. *The Transactions of IEICE J82-D-II*, 10 (1999), 1793–1803.
- [89] NISHINO, K., SATO, Y., AND IKEUCHI, K. Appearance Compression and Synthesis based on 3D Model for Mixed Reality. In *Proc. of Seventh International Conference on Computer Vision ICCV '99* (Sep. 1999), vol. 1, pp. 38–45.
- [90] NISHINO, K., SATO, Y., AND IKEUCHI, K. Eigen-Texture Method: Appearance Compression based on 3D Model. In *Proc. of Computer Vision and Pattern Recognition '99* (Jun. 1999), vol. 1, pp. 618–624.
- [91] NISHINO, K., SATO, Y., AND IKEUCHI, K. Eigen-Texture Method: Appearance Compression and Synthesis based on a 3D Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (Nov. 2001), 1257–1265.
- [92] NISHINO, K., ZHANG, Z., AND IKEUCHI, K. Determining Reflectance Parameters and Illumination Distribution from a Sparse Set of Images for View-dependent Image Synthesis. In *Proc. of Eighth IEEE International Conference on Computer Vision ICCV '01* (Jul. 2001), vol. 1, pp. 599–606.
- [93] PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. *Numerical Recipes in C, Second Edition*. Cambridge University Press, 1992.
- [94] PULLI, K. Multiview registration for large data sets. In *Second Int. Conf. on 3D Digital Imaging and Modeling* (Oct 1999), pp. 160–168.
- [95] PULLI, K., COHEN, M., DUCHAMP, T., HOPPE, H., SHAPIRO, L., AND STUETZLE, W. View-based Rendering: Visualizing Real Objects from Scanned Range and Color Data. In *Proceedings of 8th Eurographics Workshop on Rendering* (Jun. 1997).
- [96] RAMAMOORTHY, R., AND HANRAHAN, P. A Signal-Processing Framework for Inverse Rendering. In *Computer Graphics Proceedings, ACM SIGGRAPH 01* (Aug. 2001), pp. 117–128.
- [97] RAMAMOORTHY, R., AND HANRAHAN, P. On the relationship between radiance and irradiance: determining the illumination from images of a

- convex Lambertian object. *Journal of the Optical Society of America A* 18, 10 (Oct. 2001), 2448–2459.
- [98] SAGAWA, R., NISHINO, K., AND IKEUCHI, K. Robust and Adaptive Integration of Multiple Range Images with Photometric Attributes. In *Proc. of IEEE Computer Vision and Pattern Recognition, CVPR '01* (Dec. 2001), pp. 172–179.
- [99] SAGAWA, R., NISHINO, K., WHEELER, M., AND IKEUCHI, K. Parallel Processing of Range Data Merging. In *Proc. of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2001), vol. 1, pp. 577–583.
- [100] SATO, I., SATO, Y., AND IKEUCHI, K. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Trans. on Visualization and Computer Graphics* 5, 1 (Jan.-Mar. 1999), 1–12.
- [101] SATO, I., SATO, Y., AND IKEUCHI, K. Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *Proc. of Seventh International Conference on Computer Vision ICCV '99* (Sep. 1999), vol. 1, pp. 875–882.
- [102] SATO, K., AND INOKUCHI, S. Range-imaging system utilizing nematic liquid crystal mask. In *First International Conference on Computer Vision* (1987), pp. 657–661.
- [103] SATO, Y. *Object shape and reflectance modeling from color image sequence*. PhD thesis, Carnegie Mellon University, 1997.
- [104] SATO, Y., AND IKEUCHI, K. Temporal-color space analysis of reflection. *Journal of the Optical Society of America* 11, 11 (1994), 2990–3002.
- [105] SATO, Y., WHEELER, M., AND IKEUCHI, K. Object shape and reflectance modeling from observation. In *Computer Graphics Proceedings, ACM SIGGRAPH 97* (Aug. 1997), pp. 379–387.
- [106] SEITZ, S., AND DYER, C. View Morphing. In *Computer Graphics Proceedings, ACM SIGGRAPH 96* (Aug. 1996), pp. 21–30.

- [107] SEITZ, S., AND DYER, C. Photorealistic Scene Reconstruction by Voxel Coloring. In *Proc. of Computer Vision and Pattern Recognition '97* (1997), pp. 28–34.
- [108] SHAFER, S. Using Color to Separate Reflection Components. *COLOR Research and Application* 10, 4 (1985), 210–218.
- [109] SHASHUA, A. *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, Dept. Brain and Cognitive Science, MIT, 1992.
- [110] SHEKHAR, R., FAYYAD, E., YAGEL, R., AND CORNHILL, J. Octree-based decimation of marching cubes surfaces. In *Proc. Visualization'96* (1996), pp. 335–342.
- [111] SHU, R., CHEN, Z., AND KANKANHALLI, M. Adaptive marching cubes. *The Visual Computer* 11 (1995), 202–217.
- [112] SHUM, H.-Y., AND L-W.HE. Rendering with Concentric Mosaics. In *Computer Graphics Proceedings, ACM SIGGRAPH 99* (Aug. 1999), pp. 299–306.
- [113] SOLOMON, F., AND IKEUCHI, K. Extracting the shape and roughness of specular lobe objects using four light photometric stereo. In *Proc. of Computer Vision and Pattern Recognition '92* (1992), pp. 466–471.
- [114] STAMOS, I., AND ALLEN, P. Registration of 3d with 2d imagery in urban environments. In *Proc. the Eighth International Conference on Computer Vision, to appear* (Vancouver, Canada, 2001).
- [115] SUEN, P., AND HEALEY, G. The analysis and recognition of real-world textures in three dimensions. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22, 5 (2000), 491–503.
- [116] TOMINAGA, S., AND EBISUI, S. Scene illuminant classification: brighter is better. *Journal of the Optical Society of America A* 18, 1 (Jan. 2001), 55–64.
- [117] TOMINAGA, S., AND WANDELL, B. Standard surface-reflectance model and illuminant estimation. *J.Opt. Soc. Am. A* 6, 4 (1989), 576–584.

- [118] TORRANCE, K., AND SPARROW, E. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57 (1967), 1105–1114.
- [119] TSAI, R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation* 3, 4 (Aug. 1987), 323–344.
- [120] TURK, G., AND LEVOY, M. Zippered polygon meshes from range images. In *SIGGRAPH 94* (Jul 1994), pp. 311–318.
- [121] WALTER, E., AND PRONZATO, L. *Identification of Parametric Models from Experimental Data*. Springer, 1997.
- [122] WARD, G. Measuring and modeling anisotropic reflection. In *Computer Graphics Proceedings, ACM SIGGRAPH 92* (1992), pp. 265–272.
- [123] WHEELER, M., AND IKEUCHI, K. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 17, 3 (1995), 252–265.
- [124] WHEELER, M., SATO, Y., AND IKEUCHI, K. Consensus surfaces for modeling 3d objects from multiple range images. In *Proc. International Conference on Computer Vision* (January 1998).
- [125] WHEELER, M. D. *Automatic Modeling and Localization for Object Recognition*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1996.
- [126] WILLIAMS, L. Casting curved shadows on curved surfaces. In *Computer Graphics Proceedings, ACM SIGGRAPH 78* (1978), pp. 270–274.
- [127] WOOD, D., AZUMA, D., ALDINGER, K., CURLESS, B., DUCHAMP, T., SALESIN, D., AND STUETZLE, W. Surface Light Fields for 3D Photography. In *Computer Graphics Proceedings, ACM SIGGRAPH 00* (Jul. 2000), pp. 287–296.
- [128] WOODHAM, R. Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from a Single View. In *Image Understanding*

- Systems and Industrial Applications* (Aug. 1978), vol. 155, Proc. SPIE 22nd Annual Technical Symp., pp. 136–143.
- [129] YOU, Y., AND KAVEH, M. A regularization approach to joint blur identification and image restoration. *IEEE Transactions on Image Processing* 5 (Mar. 1996), 416–428.
- [130] YOU, Y., AND KAVEH, M. Blind Image Restoration by Anisotropic Regularization. *IEEE Transactions on Image Processing* 8 (Mar. 1999), 396–407.
- [131] YU, Y., DEBEVEC, P., MALIK, J., AND HAWKINS, T. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes From Photographs. In *Computer Graphics Proceedings, ACM SIGGRAPH 99* (Aug. 1999), pp. 215–224.
- [132] YU, Y., AND MALIK, J. Recovering Photometric Properties of Architectural Scenes from Photographs. In *Computer Graphics Proceedings, ACM SIGGRAPH 98* (Jul. 1998), pp. 207–217.
- [133] ZHANG, Z. Iterative point matching for registration of free form curves and surfaces. *International Journal of Computer Vision* 12, 2 (1994), 119–152.
- [134] ZHANG, Z. Modeling Geometric Structure and Illumination Variation of a Scene from Real Images. In *Sixth International Conference on Computer Vision* (Jul. 1998), pp. 1041–1046.
- [135] ZHANG, Z. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Proc. of Seventh International Conference on Computer Vision ICCV '99* (Sep. 1999), pp. 666–673.