

Wavelet-Texture Method: Appearance Compression by Polarization, Parametric Reflection Model,
and Daubechies Wavelet
Daisuke Miyazaki, Takushi Shibata, Katsushi Ikeuchi,
International Journal of Computer Vision,
vol. 86, no. 2-3, pp. 171-191, 2010.01

Wavelet-Texture Method: Appearance Compression by Polarization, Parametric Reflection Model, and Daubechies Wavelet

Daisuke Miyazaki, Takushi Shibata, and Katsushi Ikeuchi
Institute of Industrial Science, The University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo, Japan 153-8505.

Abstract

In order to create a photorealistic Virtual Reality model, we have to record the appearance of the object from different directions under different illuminations. In this paper, we propose a method that renders photorealistic images from a small amount of data. First, we separate the images of the object into a diffuse reflection component and a specular reflection component by using linear polarizers. Then, we estimate the parameters of the reflection model for each component. Finally, we compress the difference between the input images and the rendered images by using wavelet transform. At the rendering stage, we first calculate the diffuse and specular reflection images from the reflection parameters, then add the difference decompressed by inverse wavelet transform into the calculated reflection images, and finally obtain the photorealistic image of the object.

1 Introduction

Rendering of photorealistic 3D images is widely used today in medical, educational, entertainment, arts, and digital archive fields. Under the circumstances, we propose a compression technique for brightness information that reconstructs the appearance of the object and renders photorealistic 3D images with improved clarity. Image-based rendering is a powerful tool for representing the appearance of an object [5, 14, 25, 32, 64, 65]. Most of the methods cited do not use geometrical information about the object, but we prefer to use this information since there is a wide application field for this method. By using the geometrical model, we can detect collision between multiple objects, and we can calculate shadows cast between them. A virtual object image in an arbitrary environment can be obtained from geometrical information and the mathematical reflection model. We categorize the methods for doing this as model-based appearance methods. Model-based appearance methods have a problem in that there is a limitation on the object types to which the parametric reflection model can be applied. Another approach is to render a scene by using geometrical information based on real images, and we categorize these methods as image-based texture methods. Image-based texture methods have an advantage in that they can be applied to any type of object, regardless of the object's reflection property. However, image-based methods need a huge database of real images. Our approach is a hybrid of model-based appearance and image-based texture methods. We first represent the appearance of the object by using

the mathematical reflection model. We also store the difference between the input images and the rendered images computed by the mathematical reflection model. We compress such difference by using a wavelet transform.

1.1 Related Work

1.1.1 Model-Based Appearance Method

Given the 3D model of the object and the illumination condition, we can estimate the parameters of the reflection model. Ikeuchi and Sato [17] are the pioneers who estimated the reflection parameters of the object by using 3D geometry obtained by range sensor. Baribeau *et al.* [3] estimated the reflection parameters and the 3D geometry using 3 types of lasers: red, green, and blue. Kay and Caelli [19] estimated the 3D shape of the object using a photometric stereo method, and also estimated the reflection parameters. Sato *et al.* [59] developed a system to model the reflection parameters and 3D geometry by using a range sensor. Shibata *et al.* [63] improved Sato *et al.*'s method by separating the diffuse reflection and the specular reflection with polarizers. Marschner *et al.* [36] captured the reflectance of an object with a known shape by rotating the light source. Debevec *et al.* [8] acquired the reflectance field of the human face by moving the light source around the face. Also, they [9] acquired the reflectance field of cultural artifacts and showed the usefulness of their method for digital archiving. Machida *et al.* [31] analyzed the interreflection using the radiosity algorithm in order to estimate the correct reflection parameters.

Some researchers also estimated the illumination condition at the same time. In addition to the reflection parameters, Ramamoorthi and Hanrahan [54] estimated the illumination distribution by representing the low frequency component of the illumination with spherical harmonics. Nishino *et al.* [48] estimated both the direction of the light sources and the reflection parameters from a small number of photographs. Hara *et al.* [15] estimated both the reflection parameters and the position of the light source from a single image. They [16] also proposed a method to estimate the direction of the light sources when the number of the light sources is unknown. Sato *et al.* [56] estimated both the illumination distribution and the reflection parameters from the shadow of the object. Kim and Hong [21] extended Sato *et al.*'s method so that it could be applied even if the shadowed region has a complex texture.

Recently, interesting research projects have been proposed to estimate the reflection parameters when the 3D shape of the object is unknown or unreliable. In addition to estimating the reflection parameters, Lensch *et al.* [24] optimized the surface normal in order to represent the fine detail of the object. Nayar *et al.* [44] estimated the reflection parameters of shiny objects by using a setup they called "photometric sampler." Fuchs *et al.* [10] fitted the deformable face model to human face images and estimated the reflection parameters that can easily transfer them to other face models. Georghiades [12] proposed a method to estimate the reflection property and the 3D shape at the same time from a small number of images. Lu and Little [29] estimated both the reflection parameters and the shape of the object by rotating the object. Goldman *et al.* [13] estimated both the reflection parameters and the surface normal by using the photometric stereo method. The specular reflection component is removed by the analysis of dichromatic reflection model in the research presented by Shen *et al.* [62], and they estimated both the reflection parameters and the surface normal by using the photometric stereo method. Sato and Ikeuchi [58] observed the object by changing the direction of the light source, and they separated the diffuse reflection component and the specular reflection component as well as estimating the surface

normal of the object. Tominaga and Tanaka [69] estimated the reflection parameters and the surface normal from a single image by using color analysis and the shape-from-shading method. Zheng and Chellapa [78] estimated the surface normal and the light source direction from a single image by using the shape-from-shading method. Nayar *et al.* [45] separated the diffuse reflection component and the specular reflection component from color and polarization analysis, and they also estimated the surface normal by using the photometric stereo method. Kim *et al.* [20] estimated both the surface normal and the direction of the light source from a bump on the surface of the object.

Polarization is a powerful tool for estimating the shape of the object. Miyazaki *et al.* [39] estimated the shape, the reflection parameters, and the direction of the light sources from color and polarization information. Atkinson and Hancock [1, 2] estimated the surface normal from polarization images by matching local patches of the surface in two views.

Capturing the temporal variance of the reflectance is also interesting. Sun *et al.* [67] captured the transition of the appearance while the object is sprayed, dried, or dusted.

Table 1 and Table 2 list the features of some of the above-mentioned methods. Our method can express arbitrary reflection as is also done in other state-of-the-art techniques.

1.1.2 Image-Based Texture Method

Image-based texture methods represent the appearance of the object with the captured images themselves. For example, Marschner *et al.* [37] rendered the images of an object by retrieving the appearance from a database that is constructed from real images of the object taken under different illumination and from different viewpoints.

Since image-based texture methods need a huge image database, many researchers tried to compress the data. Wood *et al.* [76] compressed the appearance obtained by multiple images taken from multiple views by using principal function analysis. The eigen-texture method proposed by Nishino *et al.* [47] reduces the data by using principal component analysis (PCA) for each face on a 3D geometric model of input images. Furukawa *et al.* [11] compressed the image database for each face of the geometric model with tensor product expansion. The tensor-texture method proposed by Vasilescu *et al.* [73] renders the image by applying N-mode singular value decomposition to the image database. Wang *et al.* [74] used out-of-core tensor approximation instead of N-mode singular value decomposition. Magnor *et al.* [33] compressed the textures with wavelet transform. Ma *et al.* [30] expressed the image database by a Laplacian pyramid for each face. Ju *et al.* [18] used PCA in order to compress the components that cannot be expressed by the reflection model. Table 3 lists the features of each method.

PCA, tensor product expansion, N-mode singular value decomposition, and out-of-core tensor approximation also preserve the basis function. More data size is needed to preserve the basis function as well as the coefficients of the basis function, which results in a larger data size than the data size of the method that only preserves the coefficients. The basis functions of Laplacian transform and wavelet transform are exponential function and wavelet function, respectively. Since the basis functions are known for these techniques, we only have to preserve the coefficients. Wavelet transform is more useful than Laplacian transform for image compression and is commonly used in recent research, such as image-based lighting [38, 46] or image-based rendering [23, 26, 52]. Wavelet basis (used in JPEG 2000) produces better results than the Fourier basis (or discrete cosine transform used in JPEG) [66]. JPEG 2000 uses both the Le Gall wavelet and the Daubechies wavelet, and the Daubechies wavelet can more effectively compress the images than the Le Gall wavelet [66]. Wavelet-like basis produced by sparse

Table 1: Model-based appearance methods. Here, “active 3D sensing” represents the light-stripe laser range sensor or the structured-light range sensor that estimates the shape by triangulation, “time-of-flight” represents the time-of-flight laser range sensor, “CT” represents the computed tomography, and “photometric method” represents shape-from-shading, photometric stereo, or the extension of these methods. For reflection component separation, “color” means that the components are separated from the analysis of dichromatic reflection model, and “do not separate” means that separating the components is not needed, or the method separated them numerically without the necessity of having different color vectors between the diffuse reflection and the specular reflection.

	Geometry	Specular reflection	Separation
Shibata <i>et al.</i> [63]	Active 3D sensing	Torrance-Sparrow	2 polarizers
Debevec <i>et al.</i> [8]	Active 3D sensing	Torrance-Sparrow	2 polarizers
Marschner <i>et al.</i> [36]	Time-of-flight	Lafortune	2 polarizers
Sato <i>et al.</i> [59]	Active 3D sensing	Torrance-Sparrow	Color & image sequence
Machida <i>et al.</i> [31]	Active 3D sensing	Torrance-Sparrow	Image sequence
Ikeuchi & Sato [17]	Active 3D sensing	Torrance-Sparrow	Do not separate
Baribeau <i>et al.</i> [3]	Active 3D sensing	Torrance-Sparrow	Do not separate
Lensch <i>et al.</i> [24]	Active 3D sensing / CT	Lafortune	Do not separate
Fuchs <i>et al.</i> [10]	3D face model	Ward / Cook-Torrance	Do not separate
Nishino <i>et al.</i> [48]	Active 3D sensing	Torrance-Sparrow	Image sequence
Hara <i>et al.</i> [15]	Active 3D sensing	Torrance-Sparrow	2 polarizers
Sato <i>et al.</i> [56]	Photomodeling tool	Torrance-Sparrow	Do not separate
Kim & Hong [21]	Given	Diffuse only	Diffuse only
Ramamoorthi & Hanrahan [54]	Active 3D sensing	Phong	Do not separate
Sato & Ikeuchi [58]	Photometric method	Specular spike	Color & image sequence
Nayar <i>et al.</i> [44]	Photometric method	Specular spike	Do not separate
Kay & Caelli [19]	Photometric method	Torrance-Sparrow	Do not separate
Shen <i>et al.</i> [62]	Photometric method	Ward	Color
Goldman <i>et al.</i> [13]	Photometric method	Arbitrary	Do not separate
Lu & Little [29]	Photometric method	Arbitrary	Do not separate
Zheng & Chellapa [78]	Photometric method	Diffuse only	Diffuse only
Kim <i>et al.</i> [20]	Photometric method	Diffuse only	Diffuse only
Tominaga & Tanaka [69]	Photometric method	Phong	Color
Georghiadis [12]	Photometric method	Torrance-Sparrow	Do not separate
Nayar <i>et al.</i> [45]	Photometric method	Arbitrary	Color & 1 polarizer
Miyazaki <i>et al.</i> [39]	Polarization	Torrance-Sparrow	Color
Our method	Active 3D sensing	Arbitrary	2 polarizers

Table 2: Model-based appearance methods continued from Table 1. In “Light source” column, most methods assume the infinite far light source. “Metal sphere” means that the light source direction is given by the observation of shiny spherical object.

	Geometric configuration			Light source	Illumination estimation
	Camera	Object	Light		
Shibata <i>et al.</i> [63]	Fixed	Moving	Fixed	Point light	Given
Debevec <i>et al.</i> [8]	Moving	Fixed	Moving	Point light	Given
Marschner <i>et al.</i> [36]	Fixed	Moving	Moving	Point light	Given
Sato <i>et al.</i> [59]	Fixed	Moving	Fixed	Point light	Given
Machida <i>et al.</i> [31]	Moving	Fixed	Moving	Point light	Given
Ikeuchi & Sato [17]	Fixed	Fixed	Fixed	Point light	Given
Baribeau <i>et al.</i> [3]	Fixed	Fixed	Fixed	Laser	Given
Lensch <i>et al.</i> [24]	Moving	Fixed	Moving	Point light	Metal sphere
Fuchs <i>et al.</i> [10]	Fixed	Moving	Moving	Point light	Metal sphere
Nishino <i>et al.</i> [48]	Moving	Fixed	Fixed	Point light	Estimate
Hara <i>et al.</i> [15]	Fixed	Fixed	Fixed	Point light	Estimate
Sato <i>et al.</i> [56]	Fixed	Fixed	Fixed	Arbitrary	Estimate
Kim & Hong [21]	Fixed	Fixed	Fixed	Arbitrary	Estimate
Ramamoorthi & Hanrahan [54]	Fixed	Fixed	Moving	Arbitrary	Estimate
Sato & Ikeuchi [58]	Fixed	Fixed	Moving	Extended light	Given
Nayar <i>et al.</i> [44]	Fixed	Fixed	Moving	Extended light	Given
Kay & Caelli [19]	Fixed	Fixed	Moving	Point light	Given
Shen <i>et al.</i> [62]	Fixed	Fixed	Moving	Point light	Given
Goldman <i>et al.</i> [13]	Fixed	Fixed	Moving	Point light	Given
Lu & Little [29]	Fixed	Moving	Fixed	Point light	Given
Zheng & Chellapa [78]	Fixed	Fixed	Fixed	Point light	Estimate
Kim <i>et al.</i> [20]	Fixed	Fixed	Fixed	Point light	Estimate
Tominaga & Tanaka [69]	Fixed	Fixed	Fixed	Point light	Estimate
Georghiades [12]	Fixed	Fixed	Moving	Point light	Estimate
Nayar <i>et al.</i> [45]	Fixed	Fixed	Moving	Point light	Estimate
Miyazaki <i>et al.</i> [39]	Fixed	Fixed	Moving	Point light	Estimate
Our method	Fixed	Moving	Fixed	Point light	Given

Table 3: Image-based texture methods.

	Representation	Diffuse	Specular
Wood <i>et al.</i> [76]	Principal function analysis		
Nishino <i>et al.</i> [47] (Eigen-texture)	PCA		
Furukawa <i>et al.</i> [11]	Tensor product expansion		
Vasilescu <i>et al.</i> [73] (Tensor-texture)	N-mode SVD		
Wang <i>et al.</i> [74]	Out-of-core tensor approximation		
Magnor <i>et al.</i> [33]	Wavelet transform		
Ju <i>et al.</i> [18]	PCA	Lambertian	Phong
Ma <i>et al.</i> [30]	Laplace transform	Lambertian	Phong
Our method (Wavelet-texture)	Wavelet transform	Lambertian	Torrance-Sparrow

coding [50] more effectively compresses a natural image than the Fourier-like basis produced by PCA. Liu [27] demonstrated that the Gabor wavelet can represent the detailed features of images more effectively than PCA. The basis images obtained by PCA or tensor-based approach can represent the image well only if the input images are similar. Our approach compresses the difference between the input images and the rendered images computed by the mathematical reflection model. Such differences are difficult to represent by a small number of basis images. Wavelet can effectively compress the edges and the brightness value, which is spatially incoherent; thus, we favor wavelet rather than PCA.

1.1.3 Reflection Component Separation by Using Polarizers

Before we compress the appearance of the object, we separate the diffuse reflection component and the specular reflection component in order to model each component with the parametric reflection function. Though we can fit the parametric reflection model to the input images without separating each component, we take this approach for increased flexibility of the system: if we want to change the specular reflection model, the system can be easily upgraded by just swapping the source code of the specular reflection model. Also, separating each component is useful for analyzing the appearance of the object in order to upgrade the system to enhance data compression.

Many previous researchers separate each component by analyzing the color of the object. The basic idea of such separation is proposed by Klinker *et al.* [22], who analyzed the T-shaped distribution of the dichromatic reflection in color space. Recently, Tan *et al.* [68] separated the diffuse reflection component and the specular reflection component by using the color space called the maximum chromaticity intensity space. Mallick *et al.* [35] represented the input image with SUV color space, and applied photometric stereo to the image without specular reflection, which is calculated from U and V components of the SUV color space. In order to separate the reflection components, these methods use an image whose specular reflection component is suppressed though the color is changed. This image is called a specular-free image [68] or a UV image [35], and is useful for shape estimation [35, 39].

The reflection components can also be separated by using polarizers. Lin and Lee [28] separated the diffuse reflection component and the specular reflection component by using a special camera that has a 0° linear polarizer in front of the R channel, a 60° linear polarizer in front of the G channel, and a 120° linear polarizer in front of the B channel. They took two images from two different views, and they separated the components by analyzing the distribution of the components projected onto the 3D space, which consisted of an R- 0° axis, a G- 60° axis, and a B- 120° axis. Umeyama and Godin [72] separated the diffuse reflection component and the specular reflection component by using independent component analysis (ICA). Cula *et al.* [6] lit the object with multiple linear polarized lights in order to obtain a specular image with a high signal-to-noise ratio. They decomposed the specular image of the object illuminated by multiple lights into individual specular images of the object, each illuminated by a single light.

If we use a color-based method, we cannot estimate the diffuse reflection component if the object is white. Therefore, we separate the specular reflection component and the diffuse reflection component using linear polarizers [8, 15, 36, 63]. The polarization-based separation method can be applied to plastics, paints, papers, metals, woods, clothes, glasses, or liquids. However, it cannot be applied to polarizers, liquid crystals, or calcite; it can only be applied to the materials whose molecules are isotropically arranged and cannot be applied to the materials whose molecules are anisotropically arranged. In practice, the polarizers cannot perfectly block the light; thus, the separation will be slightly imperfect

for the objects whose specular reflection is strong, such as metals, glasses, or liquids. The appearance of such separation error cannot be expressed by a reflection model while our method can express such appearance; however, the size of the compressed data increases since such separation error cannot be represented by reflection models.

1.2 Overview

The goal of our method is to compress the data in order to render photorealistic images. Fig. 1 describes the flow of our proposed algorithm. First, we obtain the geometrical data of the target object by using a laser range sensor. Second, we rotate the object and observe the object from many directions. Next, we obtain the correspondence between the image and the geometrical data. Then, we separate the specular reflection component and the diffuse reflection component using a linear polarizer. After that, we estimate the parameters of the reflection model for each reflection component. Our proposed method uses the Torrance-Sparrow model for specular reflection, which is much more photorealistic than the Phong model. However, a reflection model is a simplified expression of a real reflection; thus, it cannot always express the exact reflection. Therefore, we enhance the rendering precision by saving the component that cannot be expressed by reflection models. We compress this component by discrete wavelet transform to reduce the data size. We call our proposed method the “wavelet-texture method.” The Eigen-Texture method can only compress in one dimension; however, our method can compress the data in three or more dimensions. The texture is two-dimensional data, and if we take many images by moving the camera two-dimensionally and moving the light two-dimensionally, we obtain six-dimensional data, and our method can be useful for such high-dimensional data. Our proposed method is based on the research done by Shibata *et al.* [63]. We use the software developed in the previous research in order to estimate the parameters of the reflection model. Both the previous research and the current research use polarizers in order to separate the specular reflection component and the diffuse component. The main contribution of the current research is to use wavelet transform to compress the component that cannot be expressed by reflection models.

We describe the proposed method based on wavelet compression in Section 2, and we provide some experimental results in Section 3. We summarize this paper in Section 4, and also discuss a drawback of this method to be overcome in the future.

2 Wavelet-Texture Method

2.1 Modeling Stage

In our experiment, we used the acquisition system shown in Fig. 2. We set the target object on a rotary table and obtained the range images and the color images while rotating the rotary table in a constant interval. In order to express the appearance of the object as realistically as possible when viewed from any directions and illuminated from any direction, we sampled the data in a constant interval. Each range image obtained by the laser range sensor was registered by alignment software [49] and integrated into a unified mesh model by merging software [55]. Through the merging stage, we also reduced many meshes so that the data size would be small. The proposed method only requires a small set of meshes for representing the appearance of the object. By using the camera calibration method [71], we obtained the correspondence between the 3D mesh model and the 2D color image. To differentiate diffuse reflection

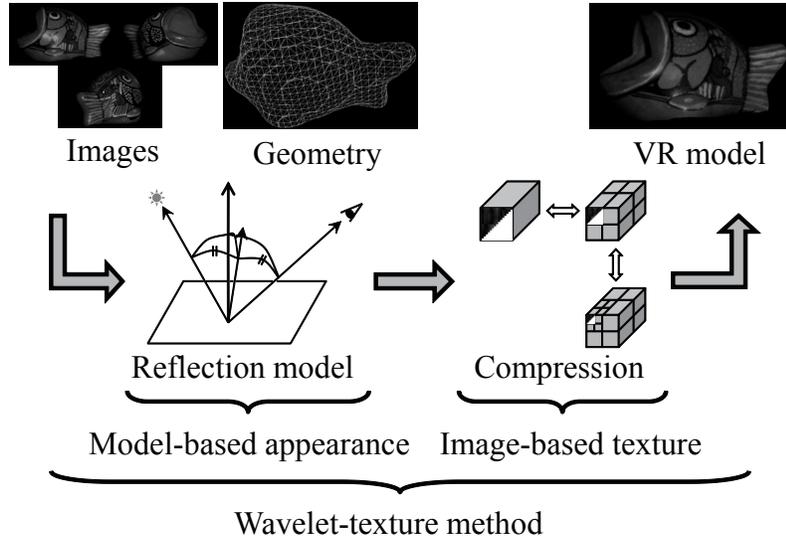


Figure 1: Wavelet-texture method.

from specular reflection, we set linear polarizers in front of the camera and the light source (Fig. 2). In order to separate the reflection components, we have to rotate the polarizer in at least 3 angles. For robust calculation, 4 images [41] or 36 images [40] are preferable, though we only set 2 angles [8] manually in order to reduce the time spent for the measurement. We set a vertical polarizer in front of the light source. First, we take the images with the camera where the polarizer is placed horizontally to block specular reflection, and next we take the images with the vertical polarizer to accentuate it. Fig. 3 represents the diffuse reflection component and the specular reflection component separated by polarizers.

By taking the images of the object from many viewpoints and under many illuminations, we can sample the BRDF (bidirectional reflectance distribution function) of the object's surface. The geometric data and the photometric data were calibrated in the same position by the camera calibration method [71]. We represented the geometric data with 3D triangular meshes. For the time being, suppose that we

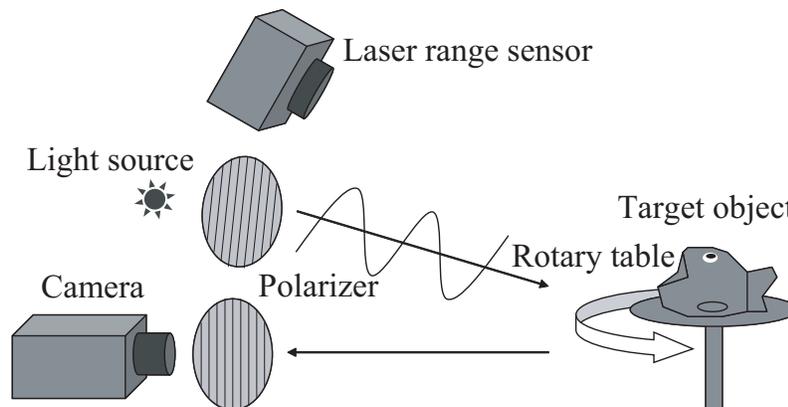


Figure 2: Data-capturing system.

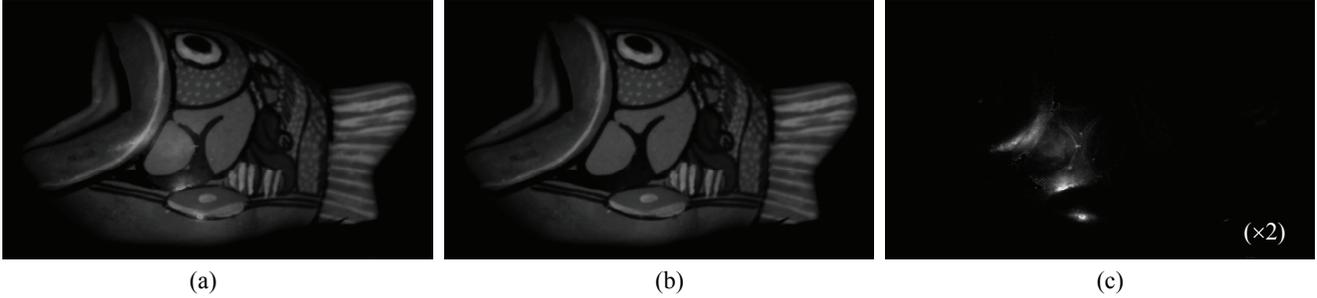


Figure 3: Result of reflection component separation: (a) Ordinary image, (b) diffuse component image, (c) specular component image.

rotated the object in one dimension, t -axis; thus, the whole input image data would be three-dimensional data, $I(x, y, t)$.

$$I(x, y, t) = I_{\text{reflect}}(x, y, t) + I_{\text{residual}}(x, y, t), \quad (1)$$

where I represents the input image and I_{reflect} represents the mathematical reflection model. Here, $I_{\text{residual}}(x, y, t)$ represents the difference image between I and I_{reflect} , and represents the residual which cannot be expressed by the parametric reflection model. First, we extracted the input image for each triangular mesh (Fig. 4 (a)). Next, we estimated the parameters of the reflection models to approximate the obtained BRDF. Then, we rendered the image using the estimated reflection parameters, under the same conditions as the input image. This rendered image was also extracted for each triangular mesh (Fig. 4 (b)). The difference image, I_{residual} , is the difference between these images (Fig. 4 (c)). Finally, we compressed I_{residual} by using the wavelet [34].

We used the Lambertian model for the diffuse reflection model, and the Torrance-Sparrow model [70] for the specular reflection model in order to express the mathematical reflection model, I_{residual} (Eq. (1)). We estimated the parameters of the Lambertian model and the Torrance-Sparrow model from the component images. We used the same specular reflection parameters for all the surface points to reduce the data size for storage, and to robustly estimate the parameters. This paper proposes the basic idea for BRDF compression, and does not deal with the computation speed of the rendering.

The diffuse reflection parameters were estimated for each pixel of the upper left half of the image whose size is $m \times m$, which can be mapped onto the face of each triangle of the 3D mesh model (Fig. 4). The size m should be appropriately determined by considering the resolution of the input images, the number of 3D meshes, and the resolution of the output image the user desires. In order to enhance the quality of the output image, we have to use as much data as possible for input images and 3D mesh models. On the other hand, in order to reduce the data size to store, we have to use as little data as possible. Calculating the 3D geometry is usually more time-consuming than calculating the 2D texture; thus, we limited the number of the triangle's faces to be less than 4096. Also, in our experiment, the size of each input image was 640×480 . In order to preserve all the data, we had to use an albedo map whose size was larger than 640×480 ; in fact, it was 1024×1024 . Consequently, in our experiment, we used size 16×16 for the albedo map for each triangle's face, where the size of the whole albedo map was 1024×1024 , and the number of the faces was less than 4096.

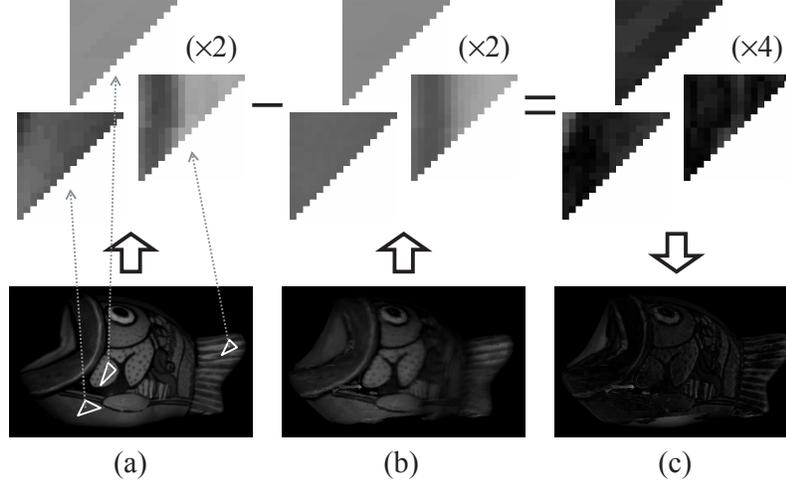


Figure 4: Calculating the difference for the faces of each triangle: (a) Input diffuse image, (b) calculated diffuse image by reflection model, (c) difference image of diffuse component. In (c), the negative value is expressed as its absolute value for visibility. “ $(\times n)$ ” indicates that the intensity is multiplied by n to improve visualization for the reader.

2.2 Rendering Stage

Any discretized illumination distribution can be expressed by a set of point light sources L_j (which include both its intensity and its size). We assume that the point light source L (which also includes both the intensity and the size) that is used for obtaining input images is known.

The final rendering image \hat{I} can be calculated from the formula shown below, if we render the image from the same viewpoint and with the same light source as the input image.

$$\hat{I} = \sum_j \frac{L_j}{L} (\hat{I}_{\text{reflect}} + \hat{I}_{\text{residual}}). \quad (2)$$

\hat{I}_{reflect} is rendered by the reflection model, and $\hat{I}_{\text{residual}}$ is the difference image (Eq. (1)).

As for rendering an arbitrary scene, we used the linear interpolation in the experiment for $\hat{I}_{\text{residual}}$, and we calculated \hat{I}_{reflect} exactly for the required scene. The problem of linear interpolation is inconspicuous, thanks to the small size of the difference information; however, it is better to use more efficient interpolation for rendering an arbitrary scene. Another solution is to densely sample the data, and both are still remaining as possible solutions in our next implementation.

2.3 Data Compression

Image-based texture methods render the images by using the sampled BRDF or the compressed data representing it. Model-based appearance methods approximate the sampled BRDF by a parameterized BRDF. Since the reflection model used for model-based appearance methods is just an approximation, there is a difference between the real image and the image rendered by the reflection model. In addition to such photometric reasons, geometric factors such as the precision of the geometric data and the precision

of the camera calibration also cause some types of noise. Therefore, we rendered the image by also saving the information that cannot be expressed by the reflection model, which is the difference image between the input image and the image rendered by the reflection model I_{residual} (Eq. (1)).

However, multiple difference images have redundant information; thus, we compressed the data. Since the difference images have less information about the appearance than the sampled BRDF, it is more effective to compress the difference images than the sampled BRDF itself, in order to reduce the data size while still preserving the photorealistic appearance. We compressed the sequence of difference images with n -D discrete wavelet transform [7]. There are many kinds of wavelets, such as the Haar wavelet, the Gabor wavelet, and the Daubechies wavelet. The Daubechies wavelet has a higher performance in image compression than the Haar wavelet [38]. The Gabor wavelet is not orthogonal [7] and it cannot be applied for multi-resolution representation, namely image compression. Therefore we used the Daubechies N=2 wavelet rather than Haar or Gabor wavelets. We calculated the sequence of the difference images, I_{residual} (Eq. (1)), and compressed them for each component with the Daubechies wavelet. We stored only the wavelet coefficients that have large values in order to reduce the size of the data.

In our experiment, we fixed the light source, and changed the viewpoint in one dimension. We want to capture images by rotating the light source and the viewpoint in all directions, and this is important future work. Note that it is easy to extend the 3D wavelet to a 4D wavelet, a 5D wavelet, or a 6D wavelet.

3 Evaluation

In this paper, we represented the quality of the image by PSNR (peak signal-to-noise ratio). The unit of PSNR is dB (decibel), and it has a large value if the image quality is high; it varies from 0 to ∞ . PSNR is an objective measure, and is widely used in order to evaluate the quality of a compressed image; for example, ISO uses both PSNR and human perception in order to evaluate the quality of CODEC (coder/decoder). The calculation of PSNR is as follows:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \quad (3)$$

$$\text{MSE} = \frac{\sum_{p \in \mathcal{P}} (I - \hat{I})^2}{|\mathcal{P}|}, \quad (4)$$

where \mathcal{P} represents the whole pixels. Empirically, people found that if the PSNR changes 0.2 dB, humans can detect the difference between two images. In addition, it is said that the image is indistinguishable from the original if the PSNR is more than 50 dB, the quality of the image is excellent if the PSNR is around 40 dB, the image has a fair quality around 30 dB, and the image is totally dissimilar to the original if the PSNR is less than 20 dB [4, 60].

Through our experiment, we intend to show that preserving the residual is important; thus, we compare our method to the model-based appearance method (Section 3.2) and the image-based texture method (Section 3.3). We do not compare it with the image-based rendering method since the purpose of our research is to compress the appearance mapped as a texture onto a 3D mesh model. Since we use the same 3D mesh model when comparing our method to other methods, we only calculate the compression ratio of the image data, and we do not include the 3D data in our calculations.

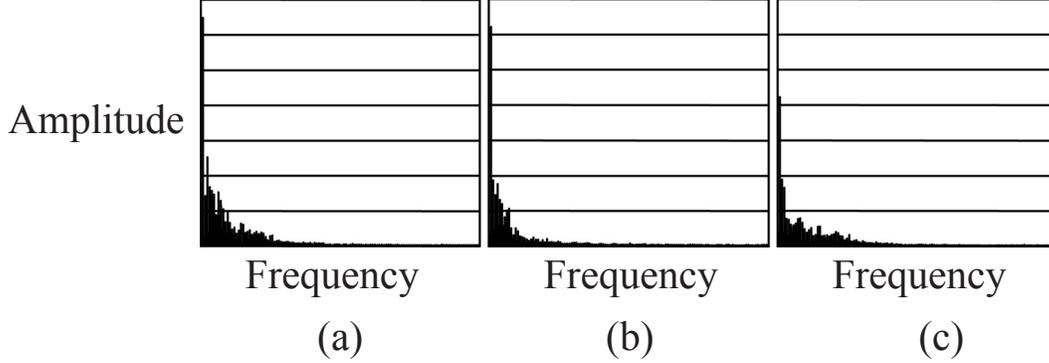


Figure 5: Frequency analysis: (horizontal axis) frequency, (vertical axis) amplitude; (a) input diffuse reflection image, (b) rendered image by diffuse reflection model, (c) difference diffuse reflection image.

We chose three different objects in our experiment in order to evaluate the performance of the proposed method. In Section 3.1, 3.2, 3.3, and 3.4, we applied our method to fish-like pottery in order to corroborate the usefulness of our method to real objects. Here, we show three results (Fig. 6, Fig. 7, and Fig. 9) with different compression ratios (28:1, 42:1, and 48:1, respectively) in order to show the variation of image quality when the compression ratio changes. In order to verify that our method can apply to a real object that cannot be assumed as a Lambertian object, we applied our method to a cloth in Section 3.5, and we show the detailed results when the compression ratio changes. Our method can be also applied to an object surface that cannot be represented by the Torrance-Sparrow reflection model, and we show the result when applied to a specular surface generated by simulation in Section 3.6. Our method can also deal with an object surface that cannot be represented by Lambertian reflection model; thus, we show the result in Section 3.7 when applied to a diffuse surface generated by simulation.

3.1 Applying Fourier Transform

Fig. 5 (c) shows the spectral power of the difference images, I_{residual} , calculated by Fourier transform. The spectrals of input images I_{input} and rendered images by reflection parameters I_{reflect} are also shown in Fig. 5 (a) and Fig. 5 (b), respectively. The low frequency component of the difference image (Fig. 5 (c)) is smaller than that in Fig. 5 (a) (b); however, this low frequency component is higher than the high frequency component. The difference image (Fig. 5 (c)) has a similar amount of high frequency component as the input image (Fig. 5 (a)). Fourier transform is not adequate because it eliminates the high frequency component for compression. However, compression by wavelet transform can preserve both the high frequency component and the low frequency component.

3.2 Comparison to the Model-Based Appearance Method

We rotated the object shown in Fig. 6 (a) in 36 directions with a 10° interval. Fig. 6 (b) represents the rendered result using the estimated diffuse reflection parameters, specular reflection parameters, and geometrical model. The PSNR of the image produced by the model-based appearance method was less than 30 dB. On the other hand, the result of our wavelet-texture method can render the information that

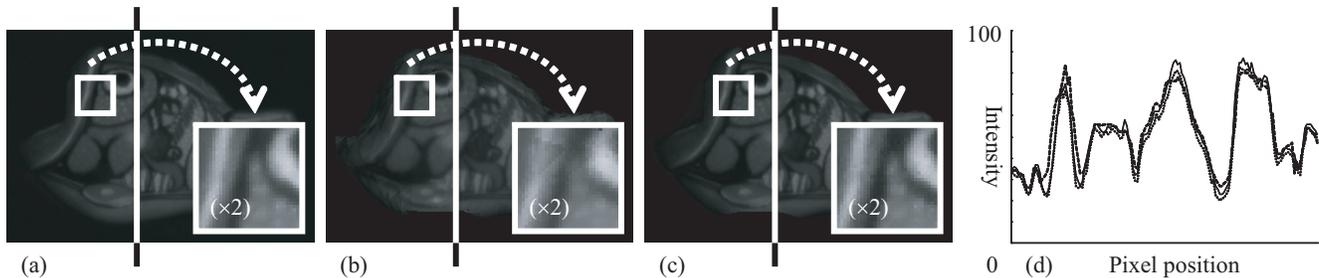


Figure 6: Comparison between model-based appearance method and wavelet-texture method: (a) Input image, (b) result of model-based appearance method, (c) result of wavelet-texture method (compression ratio 28:1), (d) (horizontal axis) pixel position, (vertical axis) intensity; (solid line)=(a), (dashed line)=(b), (dotted line)=(c).

cannot be reconstructed from the reflection model only (Fig. 6 (c)). Fig. 6 (d) is the plot of the intensity of the points indicated by the horizontal line in Fig. 6 (a)–(c). The intensity of specular reflection varies dramatically; thus, the estimation of the specular reflection parameters tends to be affected by some noises caused by the geometrical model, camera calibration, object surface, and so on. However, our method succeeds in producing an image that is close to the input image.

3.3 Comparison to the Image-Based Texture Method

Fig. 7 (b) (c) shows the result of the image-based texture method, and Fig. 7 (d) (e) shows the result of the wavelet-texture method. Fig. 7 (a) is an input image for comparison. The image-based texture method we used here does not separate the reflection components, does not use reflection models, and compresses only by wavelet. Fig. 7 (b) (d) shows the rendering results, and Fig. 7 (c) (e) shows the errors between the rendered image and the input image. The PSNR of the wavelet-texture method was 48 dB, and the PSNR of the image-based texture method was 47 dB; thus, the image quality of the proposed method was higher than that of the image-based texture method. Note that increasing 0.1 dB is considered to be quite important for research in the field of image compression, and our method increased the PSNR 0.95 dB. Also, there is a noise at the boundary of each face of geometrical data when processed by the image-based texture method compared to the proposed method.

In this experiment, the compression ratio was 42:1 for both of these methods. However, we only compared the data size of wavelet coefficients, and we have not included the data size of reflection parameters. In our current implementation, the reflection parameters are not compressed. We represented the diffuse reflection parameters with a 1-byte non-negative integer value. We represented the reflectance of specular reflection parameters with a 1-byte non-negative integer value and the surface roughness of specular reflection model with a 4-byte real number. The wavelet coefficients are represented by 4-byte integer value. The fish data we used have 4,076 faces. As we have already stated, each face is represented by the texture whose size is 16×16 . We took 36 images from different viewpoint. In this case, the actual compression ratio and the PSNR are shown in Table 4 and Fig. 8. As expected, the wavelet-texture method is effective than the image-based texture method when the data size is small.

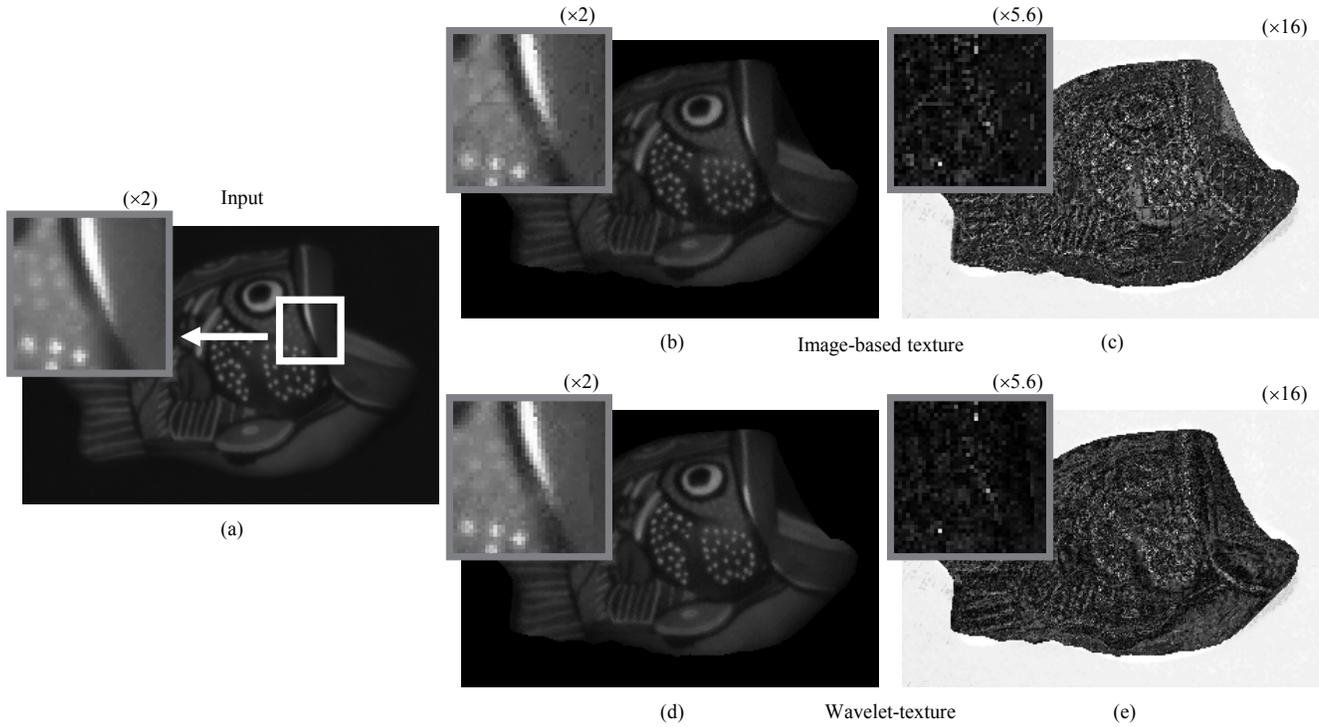


Figure 7: Comparison between image-based texture method and wavelet-texture method: (a) Input image, (b)(c) result of image-based texture method (compression ratio 42:1), (d)(e) result of wavelet-texture method (compression ratio 42:1), (b)(d) rendered result, (c)(e) difference between rendered image and input image.

Table 4: The value of PSNR of image-based texture method and wavelet-texture method. The actual compression ratio is used for experiment.

Compression ratio		35.8:1	30.5:1	28.7:1	10.8:1	5.2:1	4.7:1	4.3:1	2.9:1	2.0:1	1.6:1	1.5:1	1.4:1
PSNR [dB]	Image-based texture	22.9	23.6	23.8	29.4	32.6	32.9	33.2	34.1	34.5	34.6	34.6	34.6
	Wavelet-texture	23.7	24.6	25.0	30.6	33.0	33.3	33.5	34.1	34.5	34.6	34.6	34.6

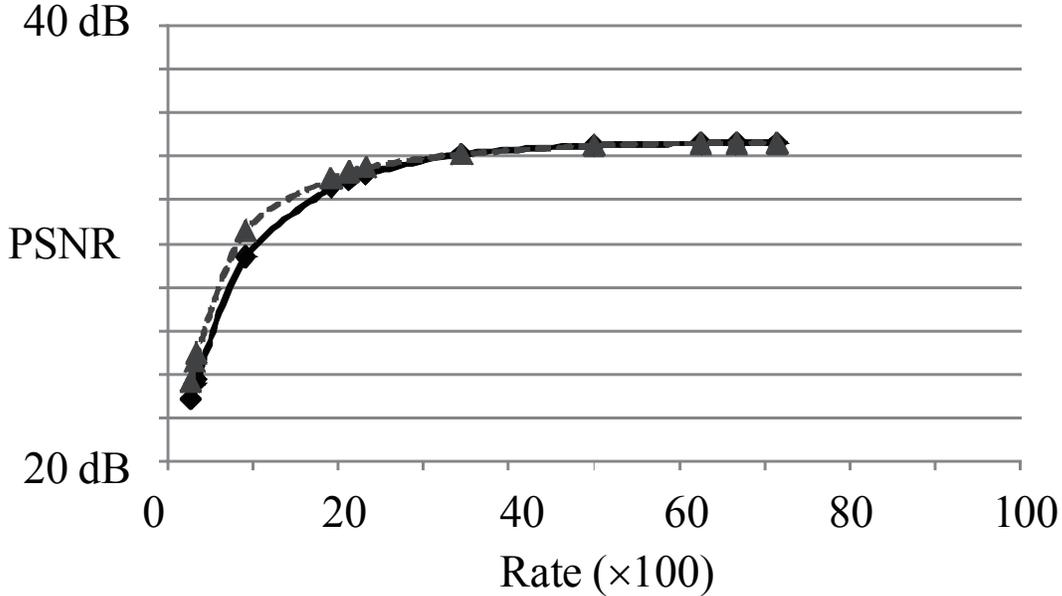


Figure 8: The relationship between the PSNR of the rendering image and the rate. The result of image-based texture method is represented by diamond mark, and the result of wavelet-texture method is represented by triangle mark. The actual compression ratio is used for experiment.

3.4 Comparison to the Motion JPEG 2000 Method

The sequence of 2D images is treated as 3D data in this experiment, and our method compresses them by a 3D Daubechies wavelet. Another way to compress the image sequence is to compress all 2D images one by one with a 2D Daubechies wavelet. In this section, we temporarily call this the “motion JPEG 2000” method, and compare it with our method. This motion JPEG 2000 method is implemented not to use reflection models; thus, for fair comparison, we do not compare it with the proposed method but compare it with the image-based texture method described in Section 3.3.

The result is shown in Fig. 9. Fig. 9 (a) is an input image. Fig. 9 (b) is the result of the motion JPEG 2000 method, and Fig. 9 (c) is the result of the image-based texture method. The image quality of the image-based texture method, whose PSNR is 43 dB, is higher than that of the motion JPEG 2000 method, whose PSNR is 27 dB. Section 3.3 indicates that the image quality of the proposed method is higher than the image-based texture method; thus, the proposed method is superior to the motion JPEG 2000 method. If the image sequence is compressed as 3D data, the compression effectiveness is good because of the information from the neighboring image.

3.5 Comparison between Different Compression Ratios

In this section, we describe results of compressing 12 images downloaded from the Photex Photometric Image Database [53]. The target object has an anisotropic surface, which cannot be represented by the Torrance-Sparrow model (Fig. 10 (a) (f)). In order to analyze the effectiveness of the wavelet transform to the anisotropic surface, we set the geometrical shape of the target object as a completely flat plane.

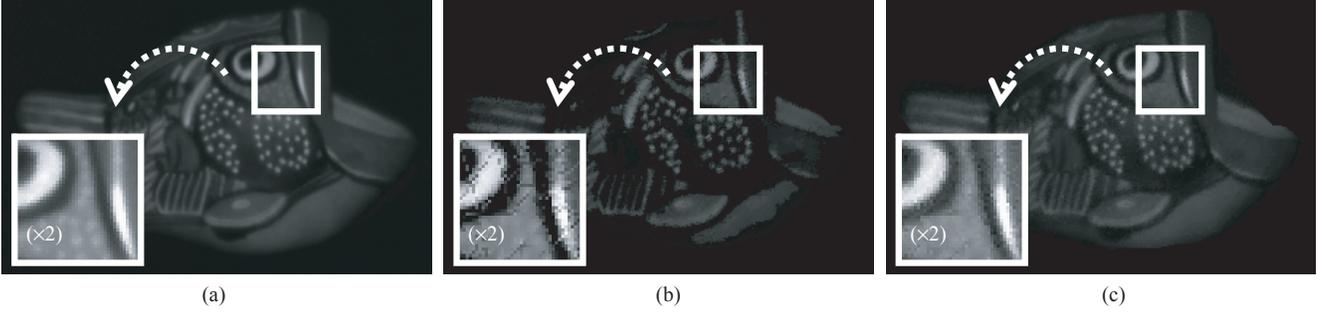


Figure 9: Comparison between motion JPEG 2000 method and image-based texture method: (a) Input image, (b) result of motion JPEG 2000 method (compression ratio 48:1), (c) result of image-based texture method (compression ratio 48:1).

Fig. 10 (b) (g), Fig. 10 (c) (h), and Fig. 10 (d) (i) are the results of our method where the compression ratio is 73:1, 5.8:1, and 1.2:1, respectively.

Fig. 11 and Table 5 show the relationship between the PSNR of the rendering image and the compression ratio. The image quality becomes higher than 40 dB when the compression ratio is smaller than 4:1. On the other hand, the image quality is less than 20 dB for the model-based appearance method, which uses the Torrance-Sparrow model. Though the Torrance-Sparrow model can only represent an isotropic surface, some reflection models such as the Ward model [75] can represent an anisotropic surface. Since our framework is flexible, we can improve our system if we substitute the reflection model used in the wavelet-texture method.

Table 5: The value of PSNR with different compression ratios.

	Model-based appearance	Wavelet-texture									Input data
Compression ratio	∞	474.0:1	197.8:1	73.2:1	24.3:1	8.5:1	5.8:1	3.8:1	2.3:1	1.2:1	1
PSNR [dB]	18.2	20.7	22.3	24.7	28.3	35.4	38.8	41.7	45.2	51.0	∞

3.6 Comparison between Different Number of Input Images

In this section, we describe results of compressing simulational images calculated by the Ward model, which can represent an anisotropic reflection. We set 0.1 for the surface roughness in the x -axis, and 0.2 for that in the y -axis. The input images are shown in Fig. 12 (a). In this section, we discuss the results when we change the number of input images; thus, we only rendered the specular reflection, since it is well-known that the diffuse reflection can be expressed from only three images [61]. First, we estimated the parameters of the Torrance-Sparrow model, and rendered the images from the estimated parameters (Fig. 12 (b)). Next, we calculated the difference between the input image and the rendered image, and compressed it by wavelet transform. We rendered the images by the wavelet-texture method (Fig. 12 (c) (d)). For comparison, we also show the results of the image-based texture method in Fig. 12 (e).

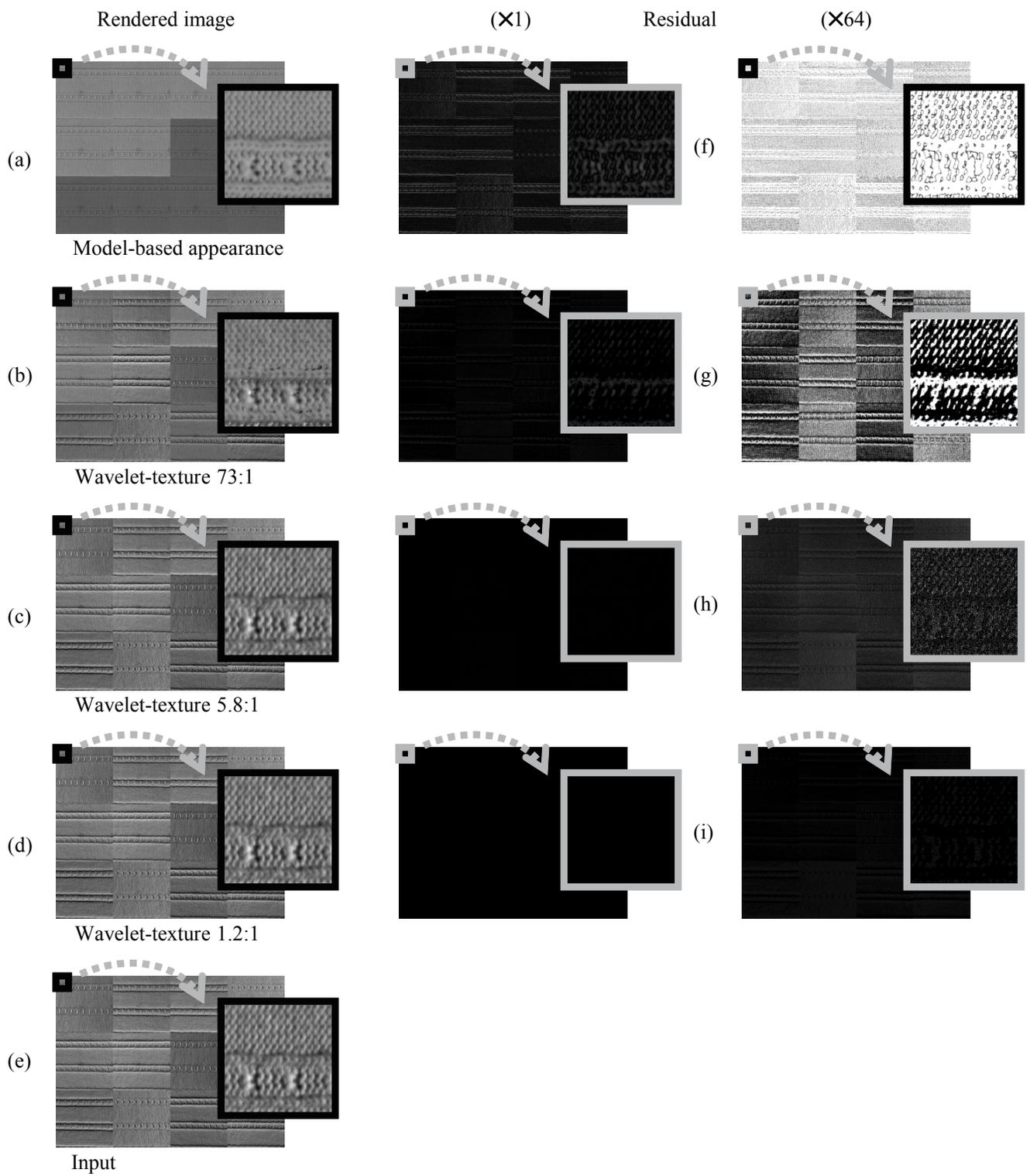


Figure 10: Rendered images for different compression ratios.

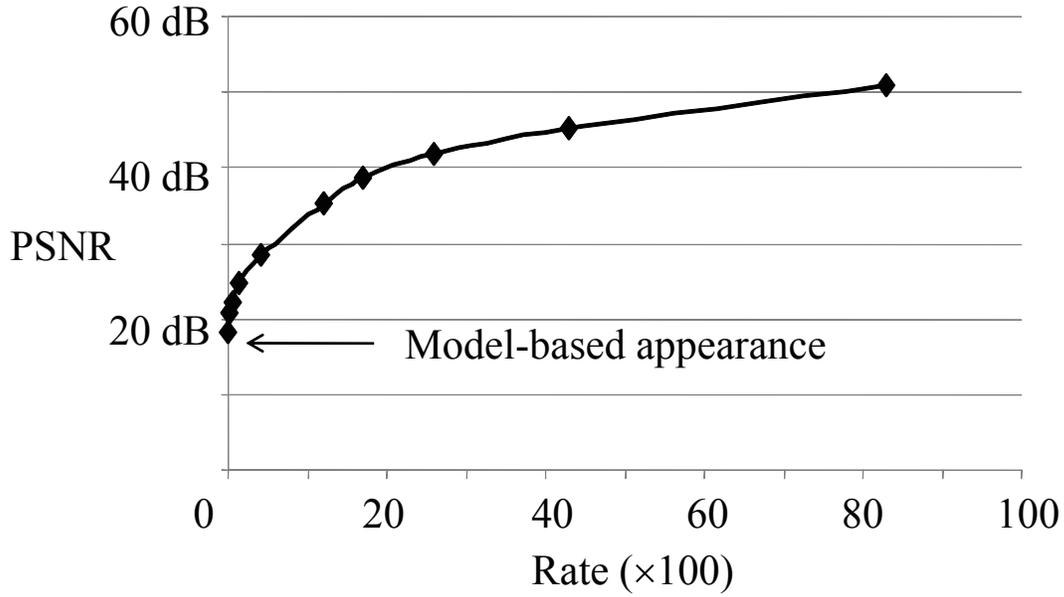


Figure 11: The relationship between the PSNR of the rendering image and the rate. In this experiment, we use only the difference images to evaluate the rate, and the reflection parameters are not used to calculate it. The rate is calculated by the number of wavelet coefficients divided by the number of pixels.

Table 6: The PSNR of anisotropic sphere. The PSNR of model-based appearance was 31.0. The actual compression ratio is used for experiment.

Compression ratio	9.9:1	9.6:1	9.4:1	5.9:1	4.4:1	3.1:1	2.3:1	2.2:1	2.1:1
PSNR [dB] Image-based texture	35.9	36.0	36.1	36.9	36.9	36.9	36.9	36.9	36.9
Wavelet-texture	38.6	38.7	38.7	39.0	39.0	39.0	39.0	39.0	39.0

Table 7: The PSNR of diffuse texture. The PSNR of model-based appearance was 30.4. The actual compression ratio is used for experiment.

Compression ratio	1.2:1	1.5:1	2.0:1	2.4:1	2.6:1	4.6:1	5.7:1	6.0:1	8.7:1	8.8:1	10.4:1	31.3:1	42.6:1
PSNR [dB] Image-based texture	59.3	58.9	58.9	58.9	58.9	58.9	58.9	58.9	58.7	58.7	57.6	42.5	38.7
Wavelet-texture	58.9	58.9	58.9	58.9	58.9	58.9	58.9	58.9	58.9	58.9	58.9	56.3	54.2

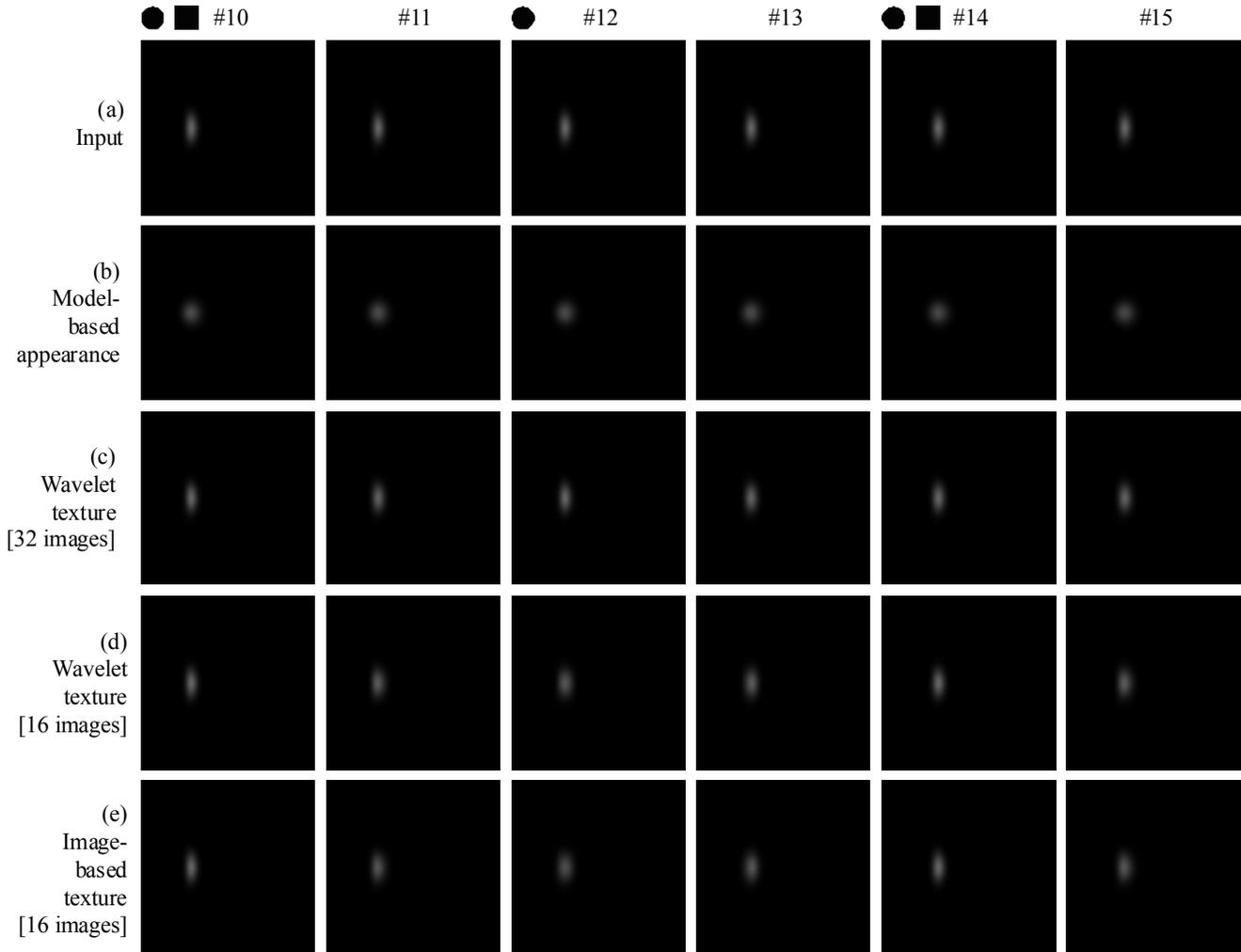


Figure 12: Rendered images for different number of input images: (a) Input images, (b) rendered images by model-based appearance method (PSNR 31.0), (c) rendered images by wavelet-texture method using 32 input images (PSNR 51.2), (d) rendered images by wavelet-texture method using 16 input images (PSNR 39.0), and (e) rendered images by image-based texture method using 16 input images (PSNR 36.9). The images marked with a circle (No. 10, No. 12, and No. 14) were used for (c), and the images marked with a rectangle (No. 10 and No. 14) were used for (d) and (e).

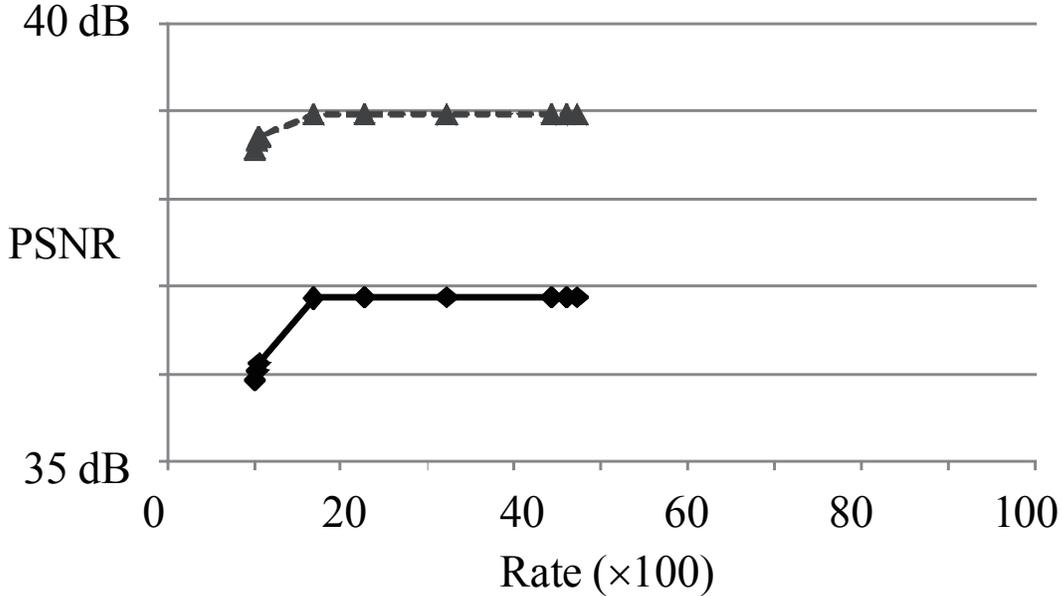


Figure 13: The relationship between the PSNR of the rendering image and the rate for the specular sphere. The result of image-based texture method is represented by diamond mark, and the result of wavelet-texture method is represented by triangle mark. The actual compression ratio is used for experiment.

For the results shown in Fig. 12 (d) and Fig. 12 (e), we used 16 images marked with a rectangle in Fig. 12. Note that the images are well rendered by the wavelet-texture method, while they are not well rendered by the image-based texture method. For the result shown in Fig. 12 (c), we used 32 images marked with a circle in Fig. 12. The larger the number of input images is, the higher the quality of output images become. Note that the images without the circle are also well rendered though these are not used as the inputs. Fig. 12 shows the results with different data size, and the detailed results with same data size is shown in Table 6 and Fig. 13. In this experiment, the following data size is used. We used 16 images under different light sources. The wavelet transform is applied to the image whose size is 1024×1024 . The wavelet coefficients are represented by 4-byte integer value. We represented the reflectance of specular reflection parameters with a 1-byte non-negative integer value and the surface roughness of specular reflection model with a 4-byte real number.

Diffuse reflection can be expressed only from three images [61]; thus, any techniques such as PCA, Fourier transform, or wavelet transform can compress the input data to the data size that is same as that of three images. However, we need all of the input images in order to express specular reflections [57]. Since specular reflection moves spatially, a photometric approach such as PCA-based methods break down, while a geometric approach such as morphing [42] is effective. Wavelet transform can also express the spatial variation of specular reflection. Suppose that the number of pixels is P and the number of images is N , so the whole size of the input data is NP . PCA compresses the input data to $kP + k$ by only using k eigenvectors and k eigenvalues. If we want to compress specular reflection, we need $NP + N$ data if we use PCA. The specular reflection occurs in a small area in each image. If we denote the number of pixels of specular reflection as s , the total data to store will be sN at most, if we

use wavelet transform. For example, when $N = 32$, $P = 1024 \times 1024$, $s = 16 \times 16$, the compression ratios of PCA and wavelet transform are 1.0:1 and 4096:1, respectively.

The linear interpolation between two views is implemented for the wavelet-texture method (Fig. 12 (c) (d)) and the image-based texture method (Fig. 12 (e)). Therefore, the ghosting effect, the blending effect of specular reflection, appears in Fig. 12 (e). In our method (Fig. 12 (c) (d)), the ghosting effect is unrecognizable thanks to the hybrid approach of image-based texture method and model-based appearance method. In this experiment, we sampled the images in one direction; thus, we interpolated the image from the nearest two images. When we sample 2D BRDF, 3D BRDF, or 4D BRDF, we can interpolate the image from the nearest 4-point square, 8-point cube, or 16-point hypercube. Mathematically, we only need to search the nearest 3-point triangle, 4-point tetrahedron, or 5-point pentachoron for 2D, 3D, or 4D BRDF. However, we sample the data with even intervals; thus, interpolating using hypercubes is useful for implementation. But if we do not sample the data with constant intervals, we have to search the nearest points of the simplex.

3.7 Performance for 4D BRDF

In this section, we corroborate that our method can be also applied to 4D BRDF, since the wavelet can be applied to a tensor of any rank. Since our experimental setup shown in Fig. 2 cannot capture the whole 4D BRDF, we generated input images by computer. The target object has a reflection which obeys the Oren-Nayar model [51], which cannot be represented by the Lambertian model, since the appearance of Lambert surface does not change when the viewpoint changes though the appearance of Oren-Nayar surface changes when the viewpoint changes. We set the surface roughness parameter of Oren-Nayar model as 1. The azimuth angle of the light source, the zenith angle of the light source, the azimuth angle of the viewpoint, and the zenith angle of the viewpoint vary in 32 steps, 16 steps, 32 step, and 16 steps, respectively. We applied the algorithms to one triangle of the surface mesh whose texture looks like a heart and whose size is 16×16 . The wavelet-texture method is applied to rank 6 tensor whose dimension is (32, 16, 32, 16, 16, 16). Fig. 14 (e) is the sequence of input images and Fig. 14 (a) (f) is the result of model-based appearance method. Fig. 14 (d) (i), Fig. 14 (c) (h), and Fig. 14 (b) (g) are the results of our method whose compression ratio are 180:1, 2800:1, and 200000:1, respectively. Though the input data size is large, the appearance under different light and different viewpoint is correlated, thus, the image quality does not degrade so much even if the compression ratio is large. The detailed results with considering the size of the reflection parameters are shown in Table 7 and Fig. 15. In this experiment, the following data size is used. We used $32 \times 16 \times 32 \times 16$ images under different light sources and different viewpoints. The wavelet transform is applied to the image whose size is 16×16 . The wavelet coefficients are represented by 4-byte integer value. We represented the diffuse reflection parameters with a 1-byte non-negative integer value.

4 Conclusion

We proposed a novel framework for rendering photorealistic images of real objects with a small amount of data. By using the Daubechies wavelet, we compressed the difference image between the input image and the image rendered by the Lambertian model and the Torrance-Sparrow model. The proposed method is able to represent the correct surface reflection, which is important for photorealism, and is able

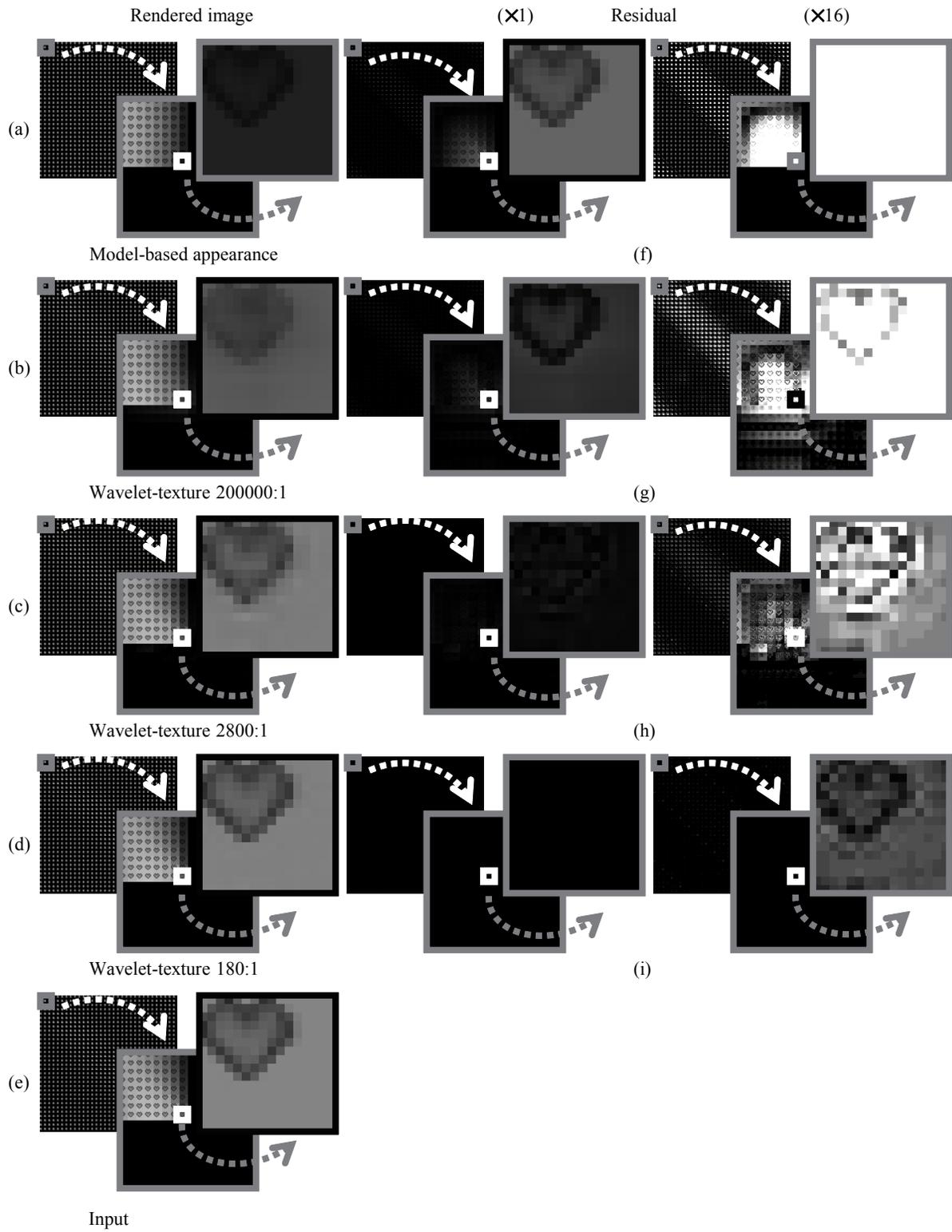


Figure 14: Rendered images for different compression ratios.

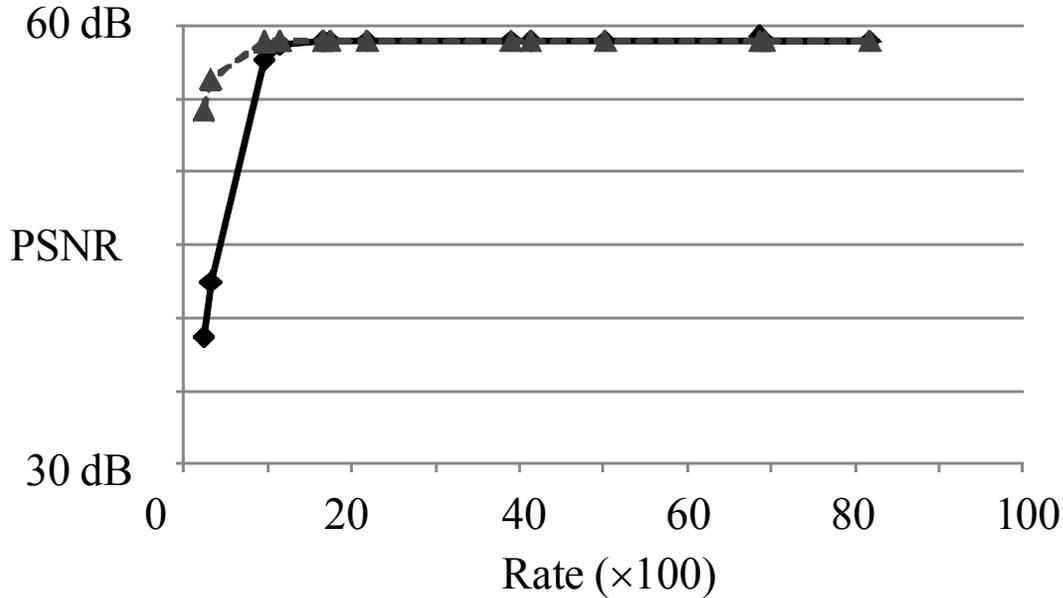


Figure 15: The relationship between the PSNR of the rendering image and the rate for the diffuse texture. The triangle mark represents the result of wavelet-texture and the diamond mark represents the result of image-based texture. The actual compression ratio is used for experiment.

to effectively compress large amounts of data. We are now planning to sample the images by rotating the viewpoint and the light source in all directions in order to generate an arbitrary scene. Also, we are planning to increase the image quality by either densely sampling the data or effectively interpolating the data. The algorithm can represent the interreflection and self-shadow of the object by difference image, and we are planning to verify this prospect.

Our framework measures the target object by laser range sensor in order to obtain its 3D geometry. The laser range sensor can obtain more precise data than the photometric stereo method. We also decomposed the reflection components by polarizers, and fit the reflection model. Estimating the specular reflection parameter is usually less stable than estimating the diffuse reflection parameter; thus, we estimate each reflection parameter separately in order to avoid the interference from the specular reflection component, when estimating the diffuse reflection parameter. Consequently, our framework can express high-quality images with a small amount of data. Many compression techniques for image-based rendering methods [23, 26, 52] that do not use 3D geometry are already proposed; however, we believe that our framework is one of the most useful techniques for compressing the object's appearance when the 3D geometry of the object is given.

Our method, which used the Lambertian model, the Torrance-Sparrow model, and the Daubechies wavelet, produced better results than previous methods. Recently, more effective reflection models, such as the Lafortune model or the He-Torrance-Sillion-Greenberg model, and more effective wavelets, such as the Coiflet or the Symmlet, have been proposed year by year. The proposed framework has the flexibility to easily upgrade the method by only substituting the compression software.

In this paper, we concentrated on the photometric problem; namely, the component that cannot be represented by reflection models. However, the geometric problem still remains; namely, we should

also store the information that cannot be represented by laser range sensor. In order to estimate a better geometric model, we are planning to acquire the detailed surface normal for each triangle of the mesh model as is proposed in the paper presented by Lensch *et al.* [24]. In addition, we are planning to express the geometric data with a small number of parameters of implicit polynomial surface using the technique proposed by Zheng *et al.* [77].

Acknowledgement

This research was supported in part by the Ministry of Education, Culture, Sports, Science and Technology under the Leading Project, “Development of High Fidelity Digitization Software for Large-Scale and Intangible Cultural Assets.” The alignment software developed by Oishi *et al.* [49] is used in this research. The merging software developed by Sagawa *et al.* [55] is used in this research. The camera calibration [71] software developed by Ryo Kurazume, Hiroki Unten, Ryo Ohkubo, Fujitsu Ltd., and Atsuhiko Banno, is used in this research. The software to compute the surface normal developed by Ryo Kurazume, Ryo Ohkubo, Tomoaki Higo, and one of the authors, Daisuke Miyazaki, is used in this research. The software to estimate the reflection parameters [63] developed by Toru Takahashi, Yoichi Sato, and one of the authors, Daisuke Miyazaki, is used in this research. The software partially used the programming code developed by Mark D. Wheeler. The source code shown in the book written by Nakano *et al.* [43] is used for implementing the wavelet transform. Atsuhiko Banno and Rei Kawakami helped the authors to calibrate the camera. Many useful comments were obtained from Akifumi Ikari and Rei Kawakami. The authors thank Joan Knapp for proofreading and editing this manuscript. They also thank anonymous reviewers for their careful reviews of the paper.

References

- [1] G.A. Atkinson and E.R. Hancock, “Multi-view surface reconstruction using polarization,” *Proceedings of IEEE International Conference on Computer Vision*, pp. 309–316, 2005.
- [2] G.A. Atkinson and E.R. Hancock, “Polarization-based surface reconstruction via patch matching,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 495–503, 2006.
- [3] R. Baribeau, M. Rioux, and G. Godin, “Color reflectance modeling using a polychromatic laser range sensor,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 263–269, 1992.
- [4] L. Cheng, S. Bossi, S. Mohapatra, M.E. Zarki, N. Venkatasubramanian, and N. Dutt, “Quality adapted backlight scaling (QABS) for video streaming to mobile handheld devices,” *Lecture Notes in Computer Science*, vol. 3420, pp. 662–671, 2005.
- [5] S.E. Chen, “QuickTime VR: an image-based approach to virtual environment navigation,” *Proceedings of International Conference on Computer Graphics and Interactive Techniques*, pp. 29–38, 1995.

- [6] O.G. Cula, K.J. Dana, D.K. Pai, and D. Wang, "Polarization multiplexing for bidirectional imaging," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1116–1123, 2005.
- [7] I. Daubechies, *Ten lectures on wavelets*, p. 357, Philadelphia, PA : Society for Industrial and Applied Mathematics, 1992.
- [8] P. Debevec, T. Hawkins, C. Tchou, H.P. Duiker, and W. Sarokin, "Acquiring the reflectance field of a human face," *Proceedings of SIGGRAPH 2000*, pp. 145–156, 2000.
- [9] T. Hawkins, J. Cohen, and P. Debevec, "A photometric approach to digitizing cultural artifacts," *Proceedings of the Conference on Virtual Reality, Archaeology, and Cultural Heritage*, pp. 333–342, 2001.
- [10] M. Fuchs, V. Blanz, H.P.A. Lensch, and H.P. Seidel, "Reflectance from images: a model-based approach for human faces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 3, pp. 296–305, 2005.
- [11] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance based object modeling using texture database: acquisition compression and rendering," *Proceedings of Eurographics Workshop on Rendering*, pp. 257–266, 2002.
- [12] A.S. Georghiadis, "Recovering 3-D shape and reflectance from a small number of photographs," *Proceedings of Eurographics Workshop on Rendering*, pp. 230–240, 2003.
- [13] D. Goldman, B. Curless, A. Hertzmann, and S.M. Seitz, "Shape and spatially-varying BRDFs from photometric stereo," *Proceedings of IEEE International Conference on Computer Vision*, pp. 341–348, 2005.
- [14] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, "The lumigraph," *Proceedings of SIGGRAPH 96*, pp. 43–54, 1996.
- [15] K. Hara, K. Nishino, and K. Ikeuchi, "Light source position and reflectance estimation from a single view without the distant illumination assumption," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 493–505, 2005.
- [16] K. Hara, K. Nishino, and K. Ikeuchi, "Multiple light sources and reflectance property estimation based on a mixture of spherical distributions," *Proceedings of IEEE International Conference on Computer Vision*, pp. 1627–1634, 2005.
- [17] K. Ikeuchi and K. Sato, "Determining reflectance properties of an object using range and brightness images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1139–1153, 1991.
- [18] D.Y. Ju, J.-H. Yoo, K.C. Seo, G. Sharp, and S.W. Lee, "Image-based illumination for electronic display of artistic paintings," *SIGGRAPH 2002 Sketches*, p. 199, 2002.
- [19] G. Kay and T. Caelli, "Inverting an illumination model from range and intensity maps," *CVGIP: Image Understanding*, vol. 59, no. 2, pp. 183–201, 1994.

- [20] C.Y. Kim, A.P. Petrov, H.K. Choh, Y.S. Seo, and I.S. Kweon, "Illuminant direction and shape of a bump," *Journal of the Optical Society of America A*, vol. 15, no. 9, pp. 2341–2350, 1998.
- [21] T. Kim and K.S. Hong, "A practical single image based approach for estimating illumination distribution from shadows," *Proceedings of IEEE International Conference on Computer Vision*, pp. 266–271, 2005
- [22] G. J. Klinker, S. A. Shafer, and T. Kanade, "The measurement of highlights in color images," *International Journal of Computer Vision*, vol. 2, no. 1, pp. 7–32, 1988.
- [23] P. Lalonde and A. Fournier, "Interactive rendering of wavelet projected light fields," *Proceedings of Graphics Interface 99*, pp. 107–114, 1999.
- [24] H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.P. Seidel, "Image-based reconstruction of spatial appearance and geometric detail," *ACM Transactions on Graphics*, vol. 22, no. 2, pp. 234–257, 2003.
- [25] M. Levoy and P. Hanrahan, "Light field rendering," *Proceedings of SIGGRAPH 96*, pp. 31–42, 1996.
- [26] J. Li, H.Y. Shum, and Y.Q. Zhang, "On the compression of image based rendering scene: a comparison among block, reference and wavelet coders," *International Journal of Image and Graphics*, vol. 1, no. 1, pp. 45–61, 2001.
- [27] C. Liu, "Gabor-based kernel PCA with fractional power polynomial models for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 572–581, 2004.
- [28] S. Lin and S.W. Lee, "Detection of specularities using stereo in color and polarization space," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 336–346, 1997.
- [29] J. Lu and J.J. Little, "Reflectance function estimation and shape recovery from image sequence of a rotating object," *Proceedings of IEEE International Conference on Computer Vision*, pp. 80–87, 1995.
- [30] W.C. Ma, S.H. Chao, Y.T. Tseng, Y.Y. Chuang, C.F. Chang, B.Y. Chen, and M. Ouhyoung, "Level-of-detail representation of bidirectional texture functions for real-time rendering," *Proceedings of ACM Symposium on Interactive 3D Graphics and Games*, pp. 187–194, 2005.
- [31] T. Machida, N. Yokoya, and H. Takemura, "Surface reflectance modeling of real objects with interreflections," *Proceedings of IEEE International Conference on Computer Vision*, pp. 170–177, 2003.
- [32] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 338–343, 2000.
- [33] M. Magnor, P. Ramanathan, and B. Girod, "Multi-view coding for image-based rendering using 3-D scene geometry," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1092–1106, 2003.

- [34] S. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [35] S.P. Mallick, T.E. Zickler, D.J. Kriegman, and P.N. Belhumeur, “Beyond Lambert: reconstructing specular surfaces using color,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 619–626, 2005.
- [36] S.R. Marschner, B. Guenter, and S. Raghupathy, “Modeling and rendering for realistic facial animation,” *Proceedings of Eurographics Workshop on Rendering*, pp. 231–242, 2000.
- [37] S.R. Marschner, S.H. Westin, E.P.F. Lafortune, and K.E. Torrance, “Image-based bidirectional reflectance distribution function measurement,” *Applied Optics*, vol. 39, no. 16, pp. 2592–2600, 2000.
- [38] V. Masselus, P. Peers, P. Dutré, and Y.D. Willems, “Smooth reconstruction and compact representation of reflectance functions for image-based relighting,” *Proceedings of Eurographics Symposium on Rendering*, pp. 287–298, 2004.
- [39] D. Miyazaki, R.T. Tan, K. Hara, and K. Ikeuchi, “Polarization-based inverse rendering from a single view,” *Proceedings of IEEE International Conference on Computer Vision*, pp. 982–987, 2003.
- [40] D. Miyazaki, M. Kagesawa, and K. Ikeuchi, “Transparent surface modeling from a pair of polarization images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 73–82, 2004.
- [41] D. Miyazaki and K. Ikeuchi, “Shape estimation of transparent objects by using inverse polarization raytracing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 2018–2030, 2007.
- [42] Y. Mukaigawa, H. Miyaki, S. Mihashi, and T. Shakunaga, “Photometric image-based rendering for image generation in arbitrary illumination,” *Proceedings of IEEE International Conference on Computer Vision*, pp. 652–659, 2001.
- [43] H. Nakano, S. Yamamoto, and Y. Yoshida, *ISBN4-320-08549-3*, p. 174, Tokyo : Kyoritsu Shuppan Co., Ltd., 1999 (in Japanese).
- [44] S.K. Nayar, K. Ikeuchi, and T. Kanade, “Extracting shape and reflectance of hybrid surfaces by photometric sampling,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 418–431, 1990.
- [45] S.K. Nayar, X.S. Fang, and T. Boult, “Separation of reflection components using color and polarization,” *International Journal of Computer Vision*, vol. 21, no. 3, pp. 163–186, 1996.
- [46] R. Ng, R. Ramamoorthi, and P. Hanrahan, “All-frequency shadows using non-linear wavelet lighting approximation,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 376–381, 2003.

- [47] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-texture method: appearance compression and synthesis based on a 3D model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 23, pp. 1257–1265, 2001.
- [48] K. Nishino, Z. Zhang, and K. Ikeuchi, "Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis," *Proceedings of IEEE International Conference on Computer Vision*, pp. 599–606, 2001.
- [49] T. Oishi, A. Nakazawa, R. Kurazume, and K. Ikeuchi, "Fast simultaneous alignment of multiple range images using index images," *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, pp. 476–483, 2005.
- [50] B.A. Olshausen and D.J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.
- [51] M. Oren and S.K. Nayar, "Generalization of the Lambertian model and implications for machine vision," *International Journal of Computer Vision*, vol. 14, no. 3, pp. 227–251, 1995.
- [52] I. Peter and W. Staßer, "The wavelet stream: interactive multi resolution light field rendering," *Proceedings of Eurographics Workshop on Rendering Techniques*, pp. 127–138, 2001.
- [53] Photex Photometric Image Database,
<http://www.macs.hw.ac.uk/texturelab/resources/databases/Photex/index.htm>.
- [54] R. Ramamoorthi and P. Hanrahan, "A signal processing framework for inverse rendering," *Proceedings of SIGGRAPH 2001*, pp. 379–387, 2001.
- [55] R. Sagawa, K. Nishino, and K. Ikeuchi, "Adaptively merging large-scale range data with reflectance properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 392–405, 2005.
- [56] I. Sato, Y. Sato, and K. Ikeuchi, "Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions," *Proceedings of IEEE International Conference on Computer Vision*, pp. 875–882, 1999.
- [57] I. Sato, T. Okabe, and Y. Sato, "Appearance sampling of real objects for variable illumination," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 29–48, 2007.
- [58] Y. Sato and K. Ikeuchi, "Temporal-color space analysis of reflection," *Journal of the Optical Society of America A*, vol. 11, no. 11, pp. 2990–3002, 1994.
- [59] Y. Sato, M.D. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," *Proceedings of SIGGRAPH 97*, pp. 379–387, 1997.
- [60] P. Seeling, M. Reisslein, and F.H.P. Fitzek, "Offset distortion traces for trace-based evaluation of video quality after network transport," *Proceedings of 14th International Conference on Computer Communications and Networks*, pp. 375–380, 2005.

- [61] A. Shashua, “On photometric issues in 3D visual recognition from a single 2D image,” *International Journal of Computer Vision*, vol. 21, no. 1/2, pp. 99–122, 1997.
- [62] L. Shen, T. Machida, and H. Takemura, “Efficient photometric stereo technique for three-dimensional surfaces with unknown BRDF,” *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, pp. 326–333, 2005.
- [63] T. Shibata, T. Takahashi, D. Miyazaki, Y. Sato, and K. Ikeuchi, “Creating photorealistic virtual model with polarization based vision system,” *Proceedings of SPIE*, vol. 5888, pp. 25–35, 2005.
- [64] H.Y. Shum, S.B. Kang, and S.C. Chan, “Survey of image-based representations and compression techniques,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, 2003.
- [65] H.Y. Shum, K.T. Ng, and S.C. Chan, “A virtual reality system using the concentric mosaic: construction, rendering, and data compression,” *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 85–95, 2005.
- [66] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 still image compression standard,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.
- [67] B. Sun, K. Sunkavalli, R. Ramamoorthi, P. Belhumeur, and S. Nayar, “Time-varying BRDFs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 595–609, 2007.
- [68] R.T. Tan and K. Ikeuchi, “Separating reflection components of textured surfaces using a single image,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 179–193, 2005.
- [69] S. Tominaga and N. Tanaka, “Estimating reflection parameters from a single color image,” *IEEE Computer Graphics and Applications*, vol. 20, no. 5, pp. 58–66, 2000.
- [70] K.E. Torrance and E.M. Sparrow, “Theory for off-specular reflection from roughened surfaces,” *Journal of the Optical Society of America*, vol. 57, no. 9, pp. 1105–1114, 1967.
- [71] R.Y. Tsai, “An efficient and accurate camera calibration technique for 3D machine vision,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 364–374, 1986.
- [72] S. Umeyama and G. Godin, “Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5 pp. 639–647, 2004.
- [73] M.A.O. Vasilescu and D. Terzopoulos, “Tensortextures: multilinear image-based rendering,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 336–342, 2004.
- [74] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja, “Out-of-core tensor approximation of multi-dimensional matrices of visual data,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 527–535, 2005.

- [75] G.J. Ward, “Measuring and modeling anisotropic reflection,” *Computer Graphics*, vol. 26, no. 2, pp. 265–272, 1992.
- [76] D.N. Wood, D.I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D.H. Salesin, and W. Stuetzle, “Surface light fields for 3D photography,” *Proceedings of SIGGRAPH 2000*, pp. 287–296, 2000.
- [77] B. Zheng, J. Takamatsu, and K. Ikeuchi, “Adaptively determining degrees of implicit polynomial curves and surfaces,” *Computer Vision – ACCV 2007*, LNCS 4844, pp. 289–300, 2007.
- [78] Q. Zheng and R. Chellapa, “Estimation of illuminant direction, albedo, and shape from shading,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 680–702, 1991.

A Detailed Implementation

A.1 Calibration between Camera and Laser Range Sensor

We measure a known object to calibrate the 2D image and the 3D range image. By using the camera calibration method [71], we obtain the 4×4 matrix \mathbf{C} , which transforms the coordinate system of the laser range sensor to the coordinate system of the color camera. The alignment process transforms the coordinate system of the laser range sensor to the coordinate system of the merged data of the target object with the 4×4 matrix \mathbf{A}_i , where i denotes each range image of the target object. The matrices \mathbf{C} and \mathbf{A}_i are represented as follows.

$$\left(\begin{array}{ccc|c} & & & \mathbf{t} \\ \hline & \mathbf{R} & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right), \quad (5)$$

where \mathbf{R} represents a rotation matrix and \mathbf{t} represents a translation vector. We use the 4×4 matrix $\mathbf{C}\mathbf{A}_i^{-1}$ to transform the coordinate system of the merged data to the coordinate system of the color camera. This procedure gives the correspondence between the 3D mesh model and the 2D color image.

A.2 Reflection Model

To differentiate diffuse reflection from specular reflection, we set linear polarizers in front of the camera and the light source. We obtain two kinds of color images for each rotating angle: one is the maximum intensity image when rotating the polarizer; the other is the minimum intensity image when rotating the polarizer. We denote the intensity of the diffuse reflection component as I_d , the intensity of the specular reflection component as I_s , the maximum intensity observed by the camera when rotating the polarizer as I_{\max} , and the minimum intensity observed when rotating the polarizer as I_{\min} . Then, the relationship between these intensities can be expressed in the following equations:

$$I_{\min} = \frac{1}{2}I_d, \quad (6)$$

$$I_{\max} = \frac{1}{2}I_d + I_s. \quad (7)$$

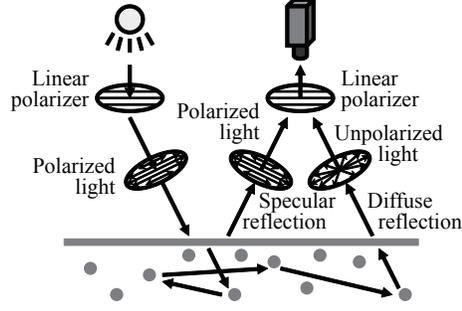


Figure 16: Reflection component separation by linear polarizer.

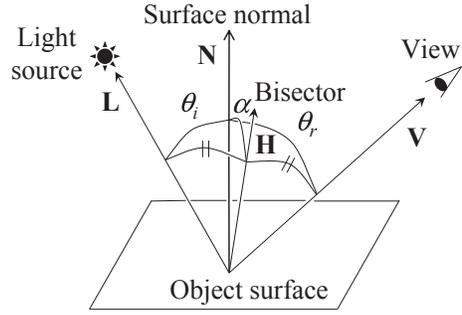


Figure 17: Relationship between incident light, surface normal direction, and view direction.

We use the following Lambertian model for the diffuse reflection model.

$$I_d = K_d (\mathbf{N} \cdot \mathbf{L}) = K_d \cos \theta_i. \quad (8)$$

Here, I_d represents the intensity of the diffuse reflection, K_d represents the reflectance (albedo) of the diffuse reflection, \mathbf{N} is a unit vector of the object surface normal, \mathbf{L} is a unit vector of the light source direction, and θ_i is the angle between the surface normal and the light source direction (Fig. 17). K_d also includes the information about the light source.

One of the mathematical models of specular reflection is the Torrance-Sparrow model [70]. The Torrance-Sparrow model for a point light source can be expressed as Eq. (9).

$$I_s = \frac{FGK_s}{\cos \theta_r} \exp\left(-\frac{\alpha^2}{2\sigma^2}\right). \quad (9)$$

Here, I_s is the intensity of specular reflection, K_s is the reflectance of specular reflection, σ is the surface roughness, θ_r is the angle between the surface normal and the view direction, and α is the angle between the surface normal vector and the bisector of the view vector and light vector (Fig. 17). F is the Fresnel coefficient, and G is the geometric attenuation factor. K_s includes the light source information. θ_r and α are calculated by the following equations.

$$\cos \theta_r = \mathbf{N} \cdot \mathbf{V}, \quad (10)$$

$$\cos \alpha = \mathbf{N} \cdot \mathbf{H}, \quad (11)$$

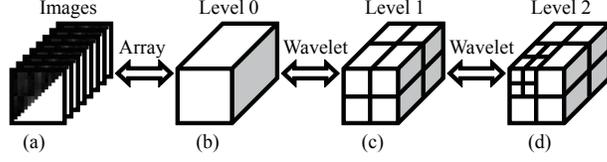


Figure 18: Image compression for each triangle's faces: (a) Sequence of difference images for each triangle's faces, (b) difference image sequence aligned as 3D data, (c) eight elements after decomposition, (d) further decomposition result.

$$\mathbf{H} = \frac{\mathbf{L} + \mathbf{V}}{\|\mathbf{L} + \mathbf{V}\|}. \quad (12)$$

Here, \mathbf{N} , \mathbf{L} , \mathbf{V} , and \mathbf{H} are the unit vector of the surface normal, the unit vector of the light source, the unit vector of the view direction, and the unit vector of the bisector of the view direction and the light source direction, respectively. In this paper, we use the following simple model to speed up the rendering time.

$$I_s = K_s \exp\left(-\frac{\alpha^2}{2\sigma^2}\right). \quad (13)$$

A.3 Wavelet Transform

In this section, we explain a multiresolution analysis by using the wavelet. Here we rewrite the top-most level of the multiresolution representation f as $f^{(0)}$; *i.e.*, $f^{(0)} = f$. The decomposition of $f^{(0)}$ in multiresolution representation up to the three level is expressed as follows.

$$f^{(0)}(x, y, t) = g^{(1)}(x, y, t) + g^{(2)}(x, y, t) + g^{(3)}(x, y, t) + f^{(3)}(x, y, t). \quad (14)$$

A similar process goes to four or more levels. Here, the superscript (j) represents the level of the resolution. $f^{(j)}$ is a linear sum of the so-called scaling function, and $g^{(j)}$ is a linear sum of the wavelet function.

$$f^{(j)}(x, y, t) = \sum_{k_x} \sum_{k_y} \sum_{k_t} s_{k_x, k_y, k_t}^{(j)} \varphi_{k_x, k_y, k_t}^{(j)}(x, y, t) \quad (15)$$

$$g^{(j)}(x, y, t) = \sum_{k_x} \sum_{k_y} \sum_{k_t} w_{k_x, k_y, k_t}^{(j)} \psi_{k_x, k_y, k_t}^{(j)}(x, y, t). \quad (16)$$

Here, s is a scaling coefficient and w is a wavelet coefficient.

$$s_{k_x, k_y, k_t}^{(j)} = \iiint_{-\infty}^{+\infty} f^{(j)}(x, y, t) \overline{\varphi_{k_x, k_y, k_t}^{(j)}(x, y, t)} dx dy dt \quad (17)$$

$$w_{k_x, k_y, k_t}^{(j)} = \iiint_{-\infty}^{+\infty} f^{(j)}(x, y, t) \overline{\psi_{k_x, k_y, k_t}^{(j)}(x, y, t)} dx dy dt \quad (18)$$

where $\bar{\varphi}$ and $\bar{\psi}$ are the complex conjugates of φ and ψ , respectively. However, φ and ψ are real numbers for the Daubechies wavelet; thus, $\bar{\varphi} = \varphi$ and $\bar{\psi} = \psi$. The above example is of three-dimensional data;

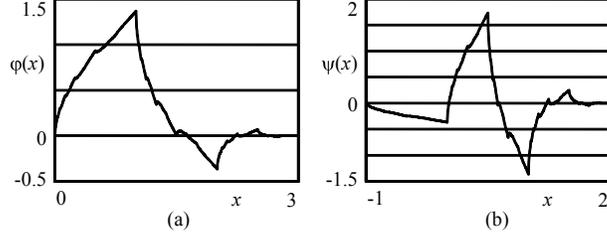


Figure 19: Daubechies (N=2) wavelet: (a) Scaling function, (b) wavelet function.

however, a similar explanation applies to four- or more dimensional data. Though the coefficients can be calculated by Eq. (17) and Eq. (18), we actually compute them by using a so-called Fast Wavelet Transform [34], which we will explain in Section A.3.1 and Section A.3.2. After decomposing $f^{(0)}$, we store only the number of the highest absolute coefficients, in order to reduce the data size. The scaling function $\varphi_0^{(0)}(x)$ and the wavelet function $\psi_0^{(0)}(x)$ for a one-dimensional Daubechies N=2 wavelet are shown in Fig. 19.

A.3.1 Compression by Daubechies Wavelet

In this example, we suppose that we rotated the object in one dimension, which we represent as the t -axis; thus, the whole image data are three-dimensional data, $f(x, y, t)$. Next, we represent the data with multiresolution by using the wavelet. Mallat [34] approximated the scaling coefficient of the topmost level multiresolution representation $s^{(0)}$ as f ; *i.e.*, $s^{(0)} = f$. From the scaling coefficient of the higher level resolution, the scaling coefficient and the wavelet coefficient of the next lower resolution can be calculated by the following formula.

$$s_{k_x, k_y, k_t}^{(j+1)} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{p_{n_x-2k_x}} \overline{p_{n_y-2k_y}} \overline{p_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (19)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{HLL}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{q_{n_x-2k_x}} \overline{p_{n_y-2k_y}} \overline{p_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (20)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{LHL}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{p_{n_x-2k_x}} \overline{q_{n_y-2k_y}} \overline{p_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (21)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{LLH}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{p_{n_x-2k_x}} \overline{p_{n_y-2k_y}} \overline{q_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (22)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{HHL}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{q_{n_x-2k_x}} \overline{q_{n_y-2k_y}} \overline{p_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (23)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{HLH}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{q_{n_x-2k_x}} \overline{p_{n_y-2k_y}} \overline{q_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (24)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{LHH}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{p_{n_x-2k_x}} \overline{q_{n_y-2k_y}} \overline{q_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (25)$$

$$w_{k_x, k_y, k_t}^{(j+1), \text{HHH}} = \sum_{n_x} \sum_{n_y} \sum_{n_t} \overline{q_{n_x-2k_x}} \overline{q_{n_y-2k_y}} \overline{q_{n_t-2k_t}} s_{n_x, n_y, n_t}^{(j)} \quad (26)$$

where p_k and q_k are defined as follows for Daubechies N=2 wavelet.

$$p_0 = 0.4829629131445341 \quad (27)$$

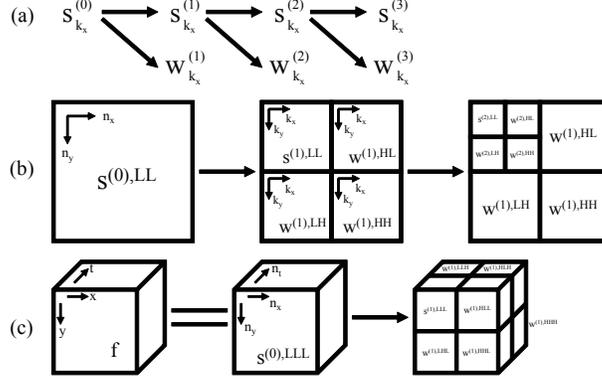


Figure 20: Decomposing the higher resolution image into lower resolution images: (a) 1D wavelet, (b) 2D wavelet, (c) 3D wavelet.

$$p_1 = 0.8365163037378077 \quad (28)$$

$$p_2 = 0.2241438680420134 \quad (29)$$

$$p_3 = -0.1294095225512603 \quad (30)$$

$$q_{-2} = -0.1294095225512603 \quad (31)$$

$$q_{-1} = -0.2241438680420134 \quad (32)$$

$$q_0 = 0.8365163037378077 \quad (33)$$

$$q_1 = -0.4829629131445341. \quad (34)$$

Note that $p_k = 0$ and $q_k = 0$ for other k . p and q are real numbers for Daubechies wavelet; thus, $\bar{p} = p$ and $\bar{q} = q$. The above mathematical expressions are of three-dimensional data; however, similar formulae are used for four- or more dimensional data. The wavelet transform can be applied for the data whose size for each axis is powers of two. If the size is not powers of two, we fill the data with zero values to make the size be powers of two. The above decomposition is illustrated in Fig. 20.

A.3.2 Decompression by Inverse Daubechies Wavelet

Now we calculate the image of the topmost level resolution $s^{(0)} = f$. From the scaling coefficient and the wavelet coefficient of the lower level resolution, the scaling coefficient of the next higher resolution can be calculated by the following formula.

$$\begin{aligned}
s_{n_x, n_y, n_t}^{(j)} &= \sum_{k_x} \sum_{k_y} \sum_{k_t} \left[p_{n_x-2k_x} p_{n_y-2k_y} p_{n_t-2k_t} s_{k_x, k_y, k_t}^{(j+1)} \right. \\
&\quad + q_{n_x-2k_x} p_{n_y-2k_y} p_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), HLL} \\
&\quad + p_{n_x-2k_x} q_{n_y-2k_y} p_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), LHL} \\
&\quad + p_{n_x-2k_x} p_{n_y-2k_y} q_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), LLH} \\
&\quad + q_{n_x-2k_x} q_{n_y-2k_y} p_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), HHL} \\
&\quad \left. + q_{n_x-2k_x} p_{n_y-2k_y} q_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), HLH} \right]
\end{aligned}$$

$$\begin{aligned}
& + p_{n_x-2k_x} q_{n_y-2k_y} q_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), \text{LHH}} \\
& + q_{n_x-2k_x} q_{n_y-2k_y} q_{n_t-2k_t} w_{k_x, k_y, k_t}^{(j+1), \text{HHH}}] \tag{35}
\end{aligned}$$

This procedure is just the opposite of Fig. 20.