Daisuke Miyazaki, Takushi Shibata, Katsushi Ikeuchi,
"Wavelet-Texture Method: Appearance Compression by Daubechies Wavelet, Reflection Model, and Polarization,"
Proceedings of International Workshop on Photometric Analysis for Computer Vision,
Rio de Janeiro, Brazil, 2007.10

http://www.cvl.iis.u-tokyo.ac.jp/~miyazaki/

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

Daisuke Miyazaki, Takushi Shibata, and Katsushi Ikeuchi
The University of Tokyo
Institute of Industrial Science, Komaba 4–6–1, Meguro-ku, Tokyo, 153–8505 Japan
http://www.cvl.iis.u-tokyo.ac.jp/

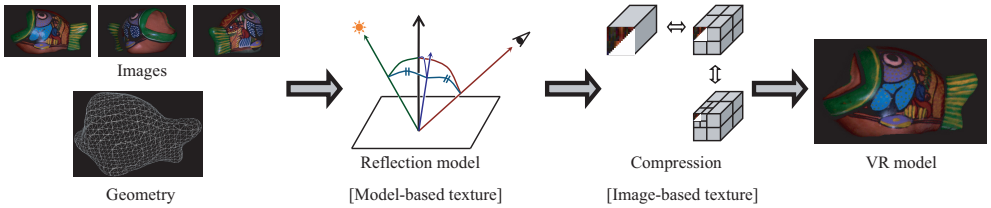

Figure 1. Wavelet-Texture method.

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. In these 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 the object [1–6]. Most of the methods cited do not use geometrical information of the object, but we prefer to use geometrical information of the object since there is a wide application field for this method. By using the geometrical model, we can detect the 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 texture methods [7–10, 12–18]. However, model-based texture methods have a problem in that there is a limitation on the object types that can apply the parametric reflection model.

Another approach to render a scene by using geometrical information is based on real images, and we categorize these methods as image-based texture methods. For example, Marschner et al. [19] 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. 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. Wood et al. [20] 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. [21] reduces the data by using principal component analysis (PCA) for each face on a 3D geometric model of input images. Furukawa et al. [22] compressed the image database for each face of the geometric model with tensor product expansion. The tensor-texture method proposed by Vasilescu et al. [23] ren-

Table 1. Comparison between other methods.

	Representation	Diffuse	Specular
Wood et al. [20]	Principal function analysis		
Nishino et al. [21] (Eigen Texture)	PCA		
Furukawa et al. [22]	Tensor product expansion		
Vasilescu & Terzopoulos [23] (Tensor Texture)	N-mode SVD		
Wang et al. [24]	Out-of-core tensor approximation		
Magnor et al. [26]	Wavelet transform		
Ju et al. [11]	PCA	Lambertian	Phong
Ma et al. [25]	Laplace transform	Lambertian	Phong
Our method (Wavelet Texture)	Wavelet transform	Lambertian	Torrance-Sparrow

ders the image by applying N-mode singular value decomposition to the image database. Wang et al. [24] used out-of-core tensor approximation instead of N-mode singular value decomposition. Ma et al. [25] expressed the image database by a Laplacian pyramid for each face. Magnor et al. [26] compressed the textures with wavelet transform. Ju et al. [11] compressed the residuals between the input image and the rendered image with PCA.

Table 1 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 which only preserve 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 [27, 28] or image-based rendering [29–31]. Wavelet basis (used in JPEG 2000) produces better results than the Fourier basis (used in JPEG[1]) [33]. Wavelet-like basis produced by sparse coding [32] more effectively compresses a natural image than the Fourier-like basis produced by PCA. 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 [33].

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 [7, 9, 13]. 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." Eigen-texture method can only compress in one dimension; however, can compress the data in three or more dimensions.

We describe the key idea of our proposed method based on wavelet compression in Section 2 and explain the detail framework in Section 3. We provide some experimental results in Section 4 and conclude this paper in Section 5.

2. Wavelet-Texture Method

By taking the images of the object from many viewpoints and under many lightings, we can sample the BRDF (bidirectional reflectance distribution function) of the object's surface. Image-based texture methods render the images by using the sampled BRDF or the compressed data representing it. Model-based texture methods approximate the sampled BRDF by a parameterized BRDF. Since the reflection model used for model-based texture 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. Our approach is to render 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.

However, such multiple difference images have redundant information; thus, we compress the data. Since the difference images have less information about the appearance than the sampled BRDF, it is more effective to com-

[1] Actually, JPEG uses the basis of discrete cosine transform.

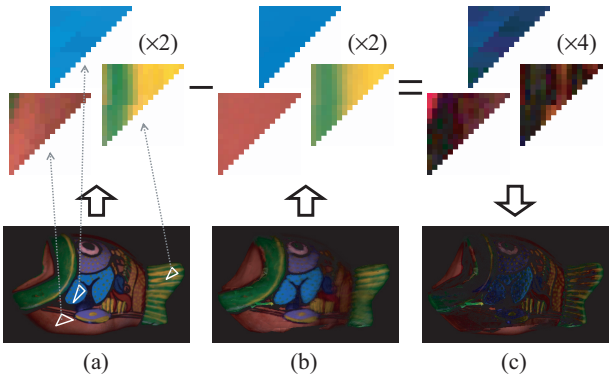

Figure 2. Calculating the difference for the faces of each triangle: (a) Input diffuse image, (b) calculated diffuse image by reflection model, (c) difference image of diffuse component. In (c), the negative value is expressed as its absolute value for visibility. "$(\times n)$" indicates that the intensity is multiplied by n to improve visualization for the reader.

press the difference images than the sampled BRDF itself, in order to reduce the data size while still preserving the photorealistic appearance. We compress the sequence of difference images with n-D discrete wavelet transform [34]. There are many kinds of wavelets, such as the Haar wavelet, the Gabor wavelet, and the Daubechies wavelet. However, the Daubechies wavelet has a higher performance in image compression than the Haar wavelet [27], and the Daubechies wavelet is more adequate for image compression than the Gabor wavelet[2]; thus, we use the Daubechies N=2 wavelet in this paper.

Suppose that the geometric data and the photometric data are already calibrated in the same position. We represent the geometric data with 3D triangular meshes. First, we extract the input image for each triangular mesh (Fig. 2(a)). Next, we estimate the parameters of the reflection models we used to approximate the obtained BRDF. The parameterized BRDF we use are described in Section 3. Then, we render the image using the estimated reflection parameters, under the same condition as the input image. This rendered image is also extracted for each triangular mesh (Fig. 2(b)). The difference image is the difference between these images (Fig. 2(c)). For the time being, suppose that we rotate the object in one dimension, t-axis; thus, the whole difference image data are three-dimensional data, $I_{\mathrm{residual}}(x, y, t)$.

$$I_{\mathrm{residual}}(x, y, t) = I(x, y, t) - BRDF(x, y, t)\,, \qquad (1)$$

where I represents the input image and $BRDF$ represents the mathematical reflection model. Finally, we compress I_{residual} by using the wavelet [35].

[2]The Gabor wavelet is not orthogonal [34]; thus, it cannot be applied for multi-resolution representation.

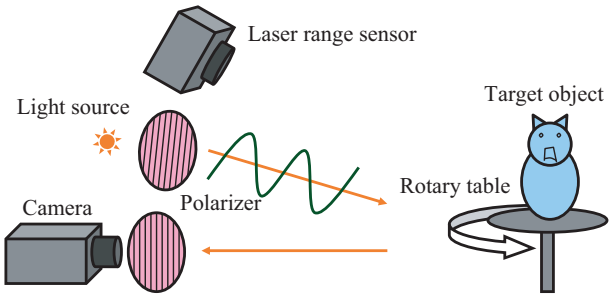

Figure 3. Data-capturing system.

Figure 4. Result of reflection component separation: (a) Ordinary image, (b) diffuse component image, (c) specular component image.

3. Proposed System

3.1. Modeling Stage

In our experiment, we use the acquisition system shown in Fig. 3. 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. Each range image obtained by the laser range sensor is registered by alignment software [36], and integrated into a unified mesh model by merging software [37]. Also, by using the camera calibration method [38], we obtain the correspondence between the 3D mesh model and the 2D color image. To differentiate diffuse reflection from specular reflection, we set linear polarizers in front of the camera and the light source (Fig. 3). Fig. 4 represents the diffuse reflection component and the specular reflection component separated by polarizers.

We use the Lambertian model for the diffuse reflection model and the Torrance-Sparrow model [39] for the specular reflection model.

$$I_{\mathrm{reflect}} = I_{\mathrm{d,reflect}} + I_{\mathrm{s,reflect}} \qquad (2)$$

$$I_{\mathrm{d,reflect}} = BRDF_{\mathrm{L}} \qquad (3)$$

$$I_{\mathrm{s,reflect}} = BRDF_{\mathrm{TS}}\,. \qquad (4)$$

We estimate the parameters of the Lambertian model $BRDF_{\mathrm{L}}$ and the Torrance-Sparrow model $BRDF_{\mathrm{TS}}$ from the component images. The diffuse reflection parameters are estimated for each pixel of the upper left half of the image whose size is $m \times m$, which can be mapped onto each triangle's faces of the 3D mesh model (Fig. 2). In the experiments shown in this paper, we use the size 16×16 for this albedo map for each triangle's faces. We use the same specular reflection parameters for all the surface points to reduce the data size for storing them, and to robustly esti-

mate the parameters. If we use the same specular reflection parameters, the image can be easily calculated by using the environmental mapping technique with a Gaussian-blurred environment; however, we have not implemented such real-time rendering software. This paper proposes the basic idea for BRDF compression, and does not deal with the computation speed of the rendering.

Next, we calculate the sequence of the difference images, I_{residual}.

$$I_{\{\mathrm{d,s}\},\mathrm{residual}}(x,y,t) = I_{\{\mathrm{d,s}\},\mathrm{input}}(x,y,t) - BRDF_{\{\mathrm{L,TS}\}}(x,y,t)\,. \quad (5)$$

Finally, we compress the sequence of difference images for each component with the Daubechies wavelet.

3.2. 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 includes both the intensity and the size) that is used for obtaining input images is known. As for the rendering by reflection model, we can generate the appearance under arbitrary illumination distribution by using the following calculation:

$$\hat{I}_{\mathrm{reflect}} = \hat{I}_{\mathrm{d,reflect}} + \hat{I}_{\mathrm{s,reflect}} \quad (6)$$

$$\hat{I}_{\mathrm{d,reflect}} = \sum_j \frac{L_j}{L} BRDF_{\mathrm{L}}(j) \quad (7)$$

$$\hat{I}_{\mathrm{s,reflect}} = \sum_j \frac{L_j}{L} BRDF_{\mathrm{TS}}(j)\,. \quad (8)$$

We obtain the difference images, $\hat{I}_{\mathrm{d,residual}}$ and $\hat{I}_{\mathrm{s,residual}}$, computed from the compressed difference images by using the inverse wavelet transform. The difference image $\hat{I}_{\mathrm{residual}}$ needed for the rendering will be the following:

$$\hat{I}_{\mathrm{residual}} = \sum_j \frac{L_j}{L} \left(\hat{I}_{\mathrm{d,residual},j} + \hat{I}_{\mathrm{s,residual},j} \right)\,. \quad (9)$$

The final rendering image $\hat{I}$ can be calculated from the formula shown below, if the input image is obtained from the viewpoint and with the light source that is the same as the viewpoint and illumination used in the rendering.

$$\hat{I} = \hat{I}_{\mathrm{reflect}} + \hat{I}_{\mathrm{residual}}\,. \quad (10)$$

As for rendering an arbitrary scene, we use the nearest data in this paper for $\hat{I}_{\mathrm{residual}}$, and we calculate $\hat{I}_{\mathrm{reflect}}$ exactly for the required scene. The problem of choosing the nearest data is inconspicuous, thanks to the smallness 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.

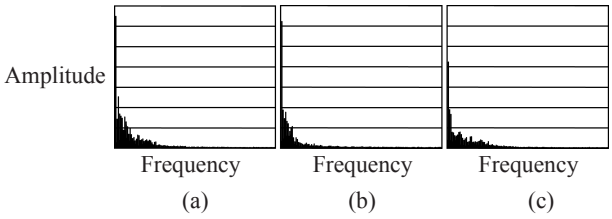

Figure 5. Frequency analysis: (horizontal axis) frequency, (vertical axis) amplitude; (a) input diffuse reflection image, (b) rendered image by diffuse reflection model, (c) difference diffuse reflection image.

In our experiment, we fix the light source, and change the viewpoint in one dimension. We have to capture the 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.

4. Evaluation

In this paper, we represent 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 is said that the image is indistinguishable from the original if the PSNR is more than 40dB, and the image is totally dissimilar to the original if the PSNR is less than 20dB.

4.1. Comparison to the 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, the compression by wavelet transform can preserve both the high frequency component and the low frequency component.

4.2. Comparison to the Model-Based Texture Method

We rotated the object shown in Fig. 6(a) in 36 directions with 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 texture method was less than 30dB. On the other hand, the result of our wavelet-

texture method can render the information that 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. Therefore, the model-based texture method struggles to reproduce specular reflection. However, our method succeeds in producing an image that is close to the input image.

4.3. Comparison to the Image-Based Texture Method

Image-based texture methods do not use the reflection model, and they directly compress the image data. Fig. 7(b)(c) are the result of the image-based texture method (compression ratio 42:1), and Fig. 7(d)(e) are the result of the wavelet-texture method (compression ratio 42:1). 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) are the rendering results, and Fig. 7(c)(e) are the errors between the rendered image and the input image. The PSNR of the wavelet-texture method was 48dB, and the PSNR of the image-based texture method was 47dB; thus, the image quality of the proposed method was higher than that of the image-based texture method. 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.

4.4. 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 each 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 4.3.

The result is shown in Fig. 8. Fig. 8(a) is an input image. Fig. 8(b) is the result of the motion JPEG 2000 method (compression ratio 48:1), and Fig. 8(c) is the result of the image-based texture method (compression ratio 48:1). The image quality of the image-based texture method, whose PSNR is 43dB, is higher than that of the motion JPEG 2000 method, whose PSNR is 27dB. Section 4.3 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 of the neighboring image. However, if the images are compressed one by one, the compression effectiveness is degraded.

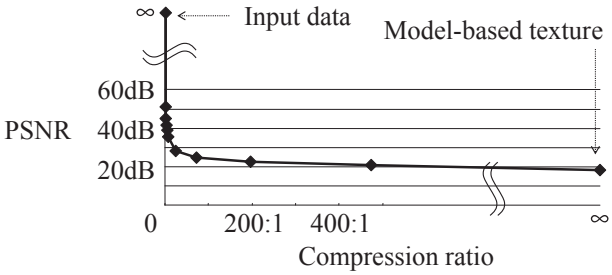

Figure 10. The relationship between the PSNR of the rendering image and the compression ratio. In this paper, we use only the difference images to evaluate the compression ratio, and the reflection parameters are not used to calculate it.

4.5. Comparison between Different Compression Ratios

In this section, we compressed twelve images downloaded from Photex Photometric Image Database [40]. The target object has an anisotropic surface, which cannot be represented by the model-based texture method (Fig. 9(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. Fig. 9(b)(g), Fig. 9(c)(h), and Fig. 9(d)(i) are the results of our method where the compression ratio is 73:1, 5.8:1, and 1.2:1, respectively.

Fig. 10 and Table 2 show the relationship between the PSNR of the rendering image and the compression ratio. The image quality becomes higher than 40dB when the compression ratio is smaller than 4:1. On the other hand, the image quality is less than 20dB for the model-based texture method.

5. 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 to effectively compress large amounts of data. We have to improve the detail of the implementation and provide deeper evaluation in the future. 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

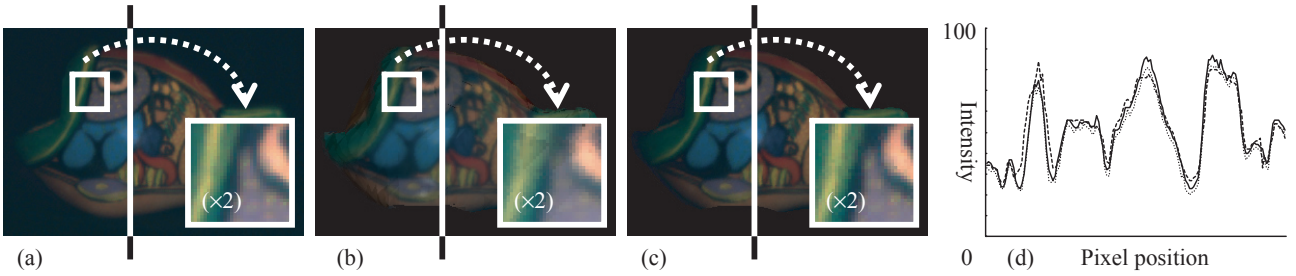

Figure 6. Comparison between Model-Based Texture method and Wavelet Texture method: (a) Input image, (b) result of Model-Based Texture method, (c) result of Wavelet Texture method (compression ratio 28:1), (d) (horizontal axis) pixel position, (vertical axis) intensity; (solid line)=(a), (dashed line)=(b),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

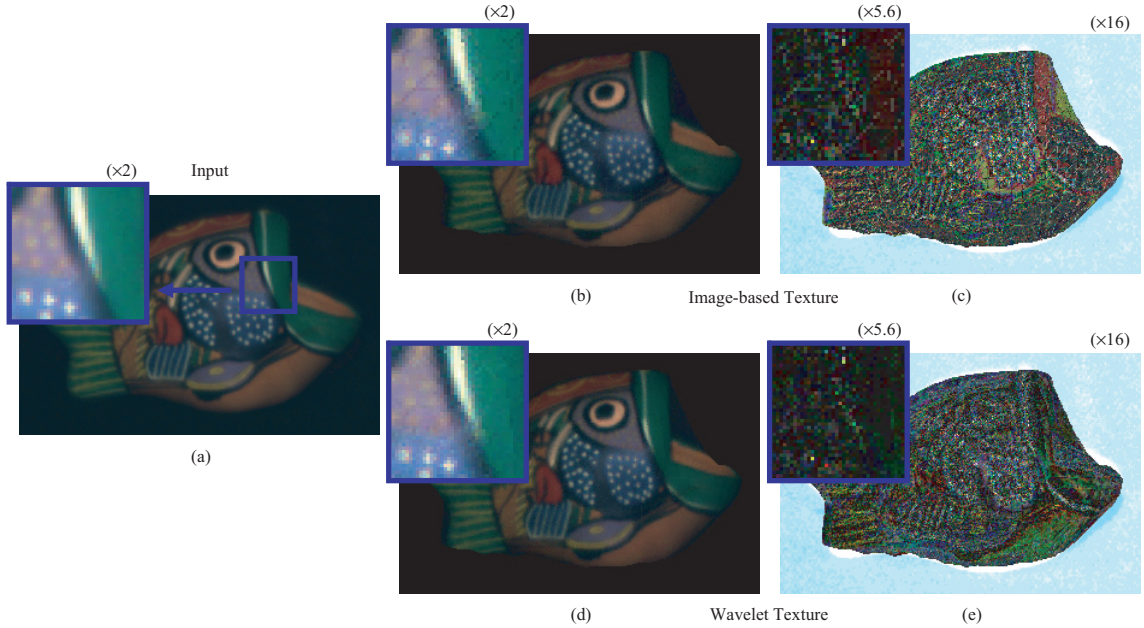

Figure 7. Comparison between image-based texture method and wavelet-texture method: (a) Input image, (b)(c) result of image-based texture method (compression ratio 42:1), (d)(e) result of wavelet-texture method (compression ratio 42:1), (b)(d) rendered result, (c)(e) difference between rendered image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

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 method, which used the Lambertian model, the Torrance-Sparrow model, and the Daubechies wavelet, produced better results than previous methods in this experiment. 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, are proposed year by year. The proposed framework has a flexibility to easily upgrade the method by only substituting the compression software.

References

[1] S.J. Gortler, R. Grzeszczuk, R. Szeliski, M.F. Cohen, "The lumigraph," *Proc. SIGGRAPH*, 43-54, 1996.

[2] M. Levoy, P. Hanrahan, "Light field rendering," *Proc. SIGGRAPH*, 31-42, 1996.

[3] M. Magnor, B. Girod, "Data compression for light-field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, **10**(3), 338-343, 2000.

[4] H.Y. Shum, S.B. Kang, S.C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, **13**(11), 1020-1037, 2003.

[5] H.Y. Shum, K.T. Ng, S.C. Chan, "A virtual reality system using the concentric mosaic: construction, rendering, and data compression," *IEEE Trans. Multimedia*, **7**(1), 85-95, 2005.

[6] S.E. Chen, "QuickTime VR: An image-based approach to virtual environment navigation," *Proc. Comput. Graph. Interactive Tech.*, 29-38, 1995.

[7] P. Debevec, T. Hawkins, C. Tchou, H.P. Duiker, W. Sarokin, "Acquiring the reflectance field of a human face," *Proc. SIGGRAPH*, 145-156, 2000.

[8] P. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, T.

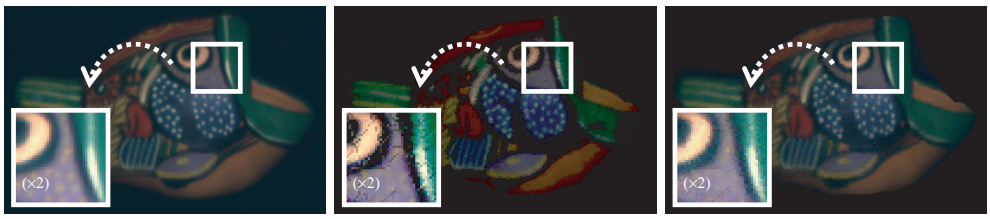

Figure 8. Comparison between motion JPEG 2000 method and image-based texture method: (a) Input image, (b) result of motion JPEG 2000 method (compression ratio 48:1), (c) result of image-based texture method (compression ratio 48:1).

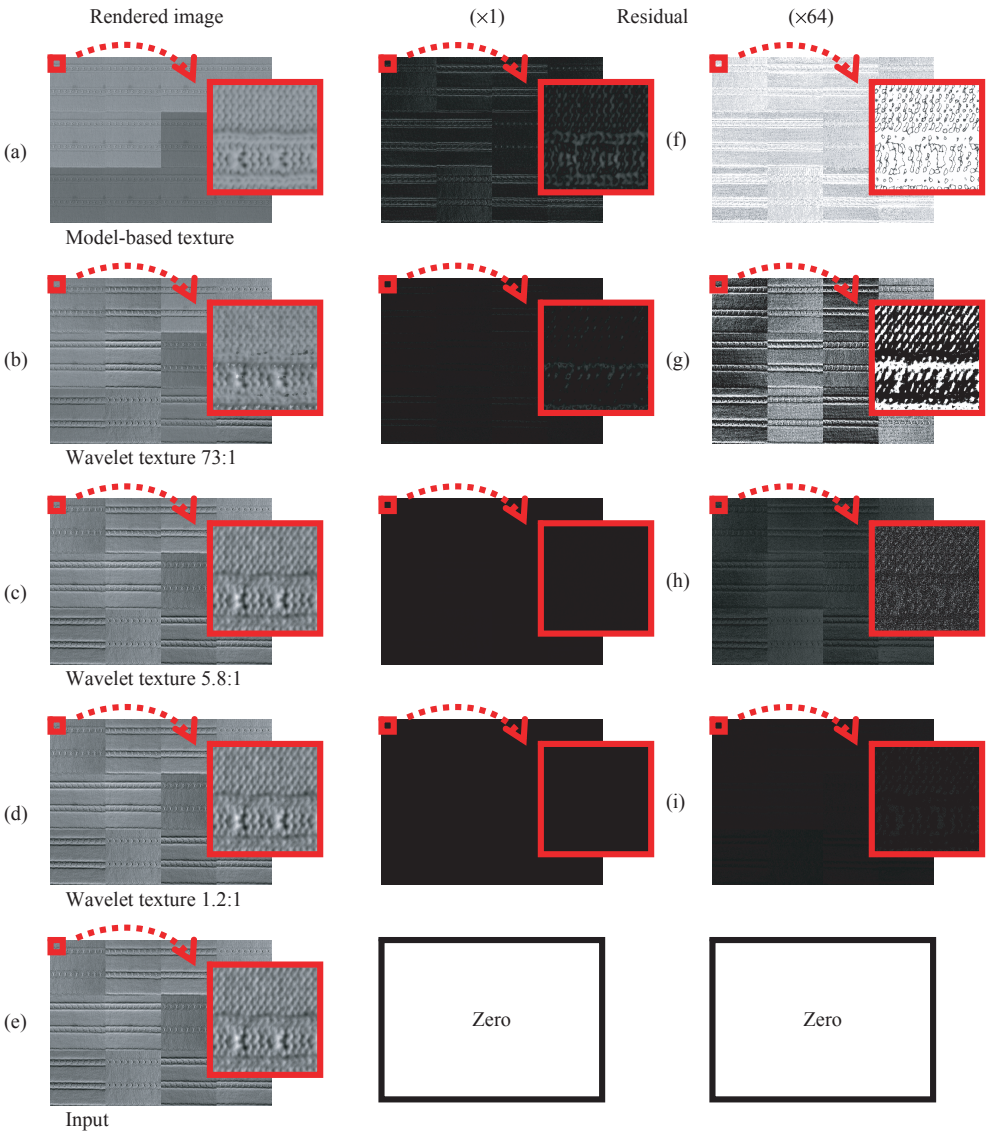

Figure 9. Rendered image for different compression ratios.

Hawkins, "A lighting reproduction approach to live-action compositing," *ACM Trans. Graph.*, **22**(3), 547-556, 2002.

[9] K. Hara, K. Nishino, K. Ikeuchi, "Light source position and reflectance estimation from a single view without the distant illumination assumption," *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**(4), 493-505, 2005.

[10] K. Ikeuchi, K. Sato, "Determining reflectance properties of an object using range and brightness images," *IEEE Trans. Patt. Anal. Mach. Intell.*, **13**(11), 1139-1153, 1991.

[11] D.Y. Ju, J.H. Yoo, K.C. Seo, G. Sharp, S.W. Lee, "Image-

Table 2. The value of PSNR with different compression ratios.

	Model-based texture	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	∞

based illumination for electronic display of artistic paintings," *SIGGRAPH Sketches*, 199-199, 2002.

[12] H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H.P. Seidel, "Image-based reconstruction of spatial appearance and geometric detail," *ACM Trans. Graph.*, **22**(2), 234-257, 2003.

[13] S.R. Marschner, B. Guenter, S. Raghupathy, "Modeling and rendering for realistic facial animation," *Proc. Eurographics Workshop Rendering*, 231-242, 2000.

[14] S.K. Nayar, K. Ikeuchi, T. Kanade, "Extracting shape and reflectance of hybrid surfaces by photometric sampling," *IEEE Trans. Rob. Autom.*, **6**(4), 418-431, 1990.

[15] R. Ramamoorthi, P. Hanrahan, "A signal processing framework for inverse rendering," *Proc. SIGGRAPH*, 379-387, 2001.

[16] M. Fuchs, V. Blanz, H.P.A. Lensch, H.P. Seidel, "Reflectance from images: A model-based approach for human faces," *IEEE Trans. Visualization Comput. Graph.*, **11**(3), 296-305, 2005.

[17] B. Sun, K. Sunkavalli, R. Ramamoorthi, P. Belhumeur, S. Nayar, "Time-varying BRDFs," *IEEE Trans. Visualization Comput. Graph.*, 2007.

[18] Y. Sato, M.D. Wheeler, K. Ikeuchi, "Object shape and reflectance modeling from observation," *Proc. SIGGRAPH*, 379-387, 1997.

[19] S.R. Marschner, S.H. Westin, E.P.F. Lafortune, K.E. Torrance, "Image-based bidirectional reflectance distribution function measurement," *Appl. Opt.*, **39**(16), 2592-2600, 2000.

[20] D.N. Wood, D.I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D.H. Salesin, W. Stuetzle, "Surface light fields for 3D photography," *Proc. SIGGRAPH*, 287-296, 2000.

[21] K. Nishino, Y. Sato, K. Ikeuchi, "Eigen-texture method: appearance compression and synthesis based on a 3D model," *IEEE Trans. Patt. Anal. Mach. Intell.*, **11**(23), 1257-1265, 2001.

[22] R. Furukawa, H. Kawasaki, K. Ikeuchi, M. Sakauchi, "Appearance based object modeling using texture database: acquisition compression and rendering," *Proc. Eurographics Workshop Rendering*, 257-266, 2002.

[23] M.A.O. Vasilescu, D. Terzopoulos, "Tensortextures: multilinear image-based rendering," *ACM Trans. Graph.*, **23**(3), 336-342, 2004.

[24] H. Wang, Q. Wu, L. Shi, Y. Yu, N. Ahuja, "Out-of-core tensor approximation of multi-dimensional matrices of visual data," *ACM Trans. Graph.*, **24**(3), 527-535, 2005.

[25] W.C. Ma, S.H. Chao, Y.T. Tseng, Y.Y. Chuang, C.F. Chang, B.Y. Chen, M. Ouhyoung, "Level-of-detail representation of bidirectional texture functions for real-time rendering," *Proc. Symp. Interactive 3D Graph. Games*, 187-194, 2005.

[26] M. Magnor, P. Ramanathan, B. Girod, "Multi-view coding for image-based rendering using 3-D scene geometry," *IEEE Trans. Circuits Syst. Video Technol.*, **13**(11), 1092-1106, 2003.

[27] V. Masselus, P. Peers, P. Dutré, Y.D. Willems, "Smooth reconstruction and compact representation of reflectance functions for image-based relighting," *Proc. Eurographics Symp. Rendering*, 287-298, 2004.

[28] R. Ng, R. Ramamoorthi, P. Hanrahan, "All-frequency shadows using non-linear wavelet lighting approximation," *ACM Trans. Graph.*, **22**(3), 376-381, 2003.

[29] P. Lalonde, A. Fournier, "Interactive rendering of wavelet projected light fields," *Proc. Graph. Interface*, 107-114, 1999.

[30] J. Li, H.Y. Shum, Y.Q. Zhang, "On the compression of image based rendering scene: a comparison among block, reference and wavelet coders," *Int'l J. Image Graph.*, **1**(1), 45-61, 2001.

[31] I. Peter, W. Staßer, "The wavelet stream: interactive multi resolution light field rendering," *Proc. Eurographics Workshop Rendering Tech.*, 127-138, 2001.

[32] B.A. Olshausen, D.J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, **381**, 607-609, 1996.

[33] A. Skodras, C. Christopoulos, T. Ebrahimi, "The JPEG 2000 still image compression standard," **18**(5), 36-58, 2001.

[34] I. Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, 1992.

[35] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Patt. Anal. Mach. Intell.*, **11**(7), 674-693, 1989.

[36] T. Oishi, A. Nakazawa, R. Kurazume, K. Ikeuchi, "Fast simultaneous alignment of multiple range images using index images," *Proc. Int'l Conf. 3-D Digit. Imaging Model.*, 476-483, 2005.

[37] R. Sagawa, K. Nishino, K. Ikeuchi, "Adaptively merging large-scale range data with reflectance properties," *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**(3), 392-405, 2005.

[38] R.Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 364-374, 1986.

[39] K.E. Torrance, E.M. Sparrow, "Theory for off-specular reflection from roughened surfaces," *J. Opt. Soc. Am.*, **57**(9), 1105-1114, 1967.

[40] Photex Photometric Image Database, `http://www.macs.hw.ac.uk/texturelab/resources/databases/Photex/index.htm`.