



# Data compression for measured heterogeneous subsurface scattering via scattering profile blending

Tatsuya Yatagawa<sup>1</sup> · Hideki Todo<sup>2</sup> · Yasushi Yamaguchi<sup>3</sup> · Shigeo Morishima<sup>4</sup>

Published online: 20 February 2019  
© The Author(s) 2019

## Abstract

Subsurface scattering involves the complicated behavior of light beneath the surfaces of translucent objects that includes scattering and absorption inside the object's volume. Physically accurate numerical representation of subsurface scattering requires a large number of parameters because of the complex nature of this phenomenon. The large amount of data restricts the use of the data on memory-limited devices such as video game consoles and mobile phones. To address this problem, this paper proposes an efficient data compression method for heterogeneous subsurface scattering. The key insight of this study is that heterogeneous materials often comprise a limited number of base materials, and the size of the subsurface scattering data can be significantly reduced by parameterizing only a few base materials. In the proposed compression method, we represent the scattering property of a base material using a function referred to as the base scattering profile. A small subset of the base materials is assigned to each surface position, and the local scattering property near the position is described using a linear combination of the base scattering profiles in the log scale. The proposed method reduces the data by a factor of approximately 30 compared to a state-of-the-art method, without significant loss of visual quality in the rendered graphics. In addition, the compressed data can also be used as bidirectional scattering surface reflectance distribution functions (BSSRDF) without incurring much computational overhead. These practical aspects of the proposed method also facilitate the use of higher-resolution BSSRDFs in devices with large memory capacity.

**Keywords** Subsurface scattering · Data compression · BSSRDF · Mobiles

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s00371-018-01626-x>) contains supplementary material, which is available to authorized users.

---

✉ Tatsuya Yatagawa  
tatsy@acm.org

Hideki Todo  
todo@fla.cgu.ac.jp

Yasushi Yamaguchi  
yama@graco.c.u-tokyo.ac.jp

Shigeo Morishima  
shigeo@waseda.jp

<sup>1</sup> Graduate School of Advanced Science and Engineering, Waseda University, 3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan

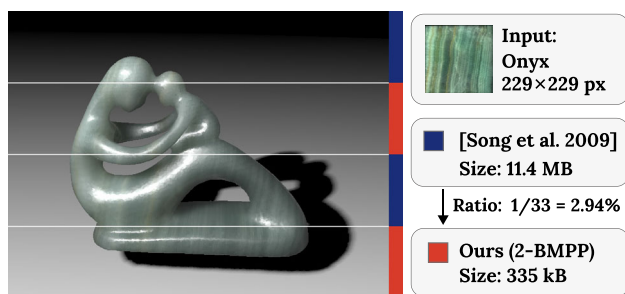
<sup>2</sup> Faculty of Liberal Arts, Chuo Gakuin University, 451 Kujike, Abiko, Chiba 270-1196, Japan

<sup>3</sup> Graduate School of Arts and Sciences, The University of Tokyo, 3-8-1 Komaba, Meguro, Tokyo 153-0041, Japan

## 1 Introduction

One of the important goals in the field of computer graphics is the realistic representation of the appearance of real-world objects. Among such objects, the representation of translucent objects such as human skin, leaves, and biological tissue is challenging because of *subsurface scattering*, which is a complex light interaction which occurs beneath the surface of a sample. Subsurface scattering is represented numerically by a vast number of scattering parameters, especially for heterogeneous translucent materials. Although these parameters can be measured by special setups which involve camera(s) and projector(s) [1–3], the large data sizes of the obtained representations often complicate their practical application, particularly in the case of real-time rendering on devices with

<sup>4</sup> Waseda Research Institute for Science and Engineering, Waseda University, 3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan



**Fig. 1** Qualitative and quantitative comparison between the proposed method and a representative previous method by Song et al. [5]. The rendered image is divided into four regions. The regions highlighted in blue represent the rendered image with the compressed BSSRDF of the previous method. Those highlighted in red represent the image with the compressed BSSRDF of the proposed method. As can be seen, the proposed compression method achieves a data size that is 2.94% smaller compared to the previous method even though the visual qualities of all the regions are nearly identical

limited random access and graphics memory, such as video game consoles, tablets, and mobile phones.

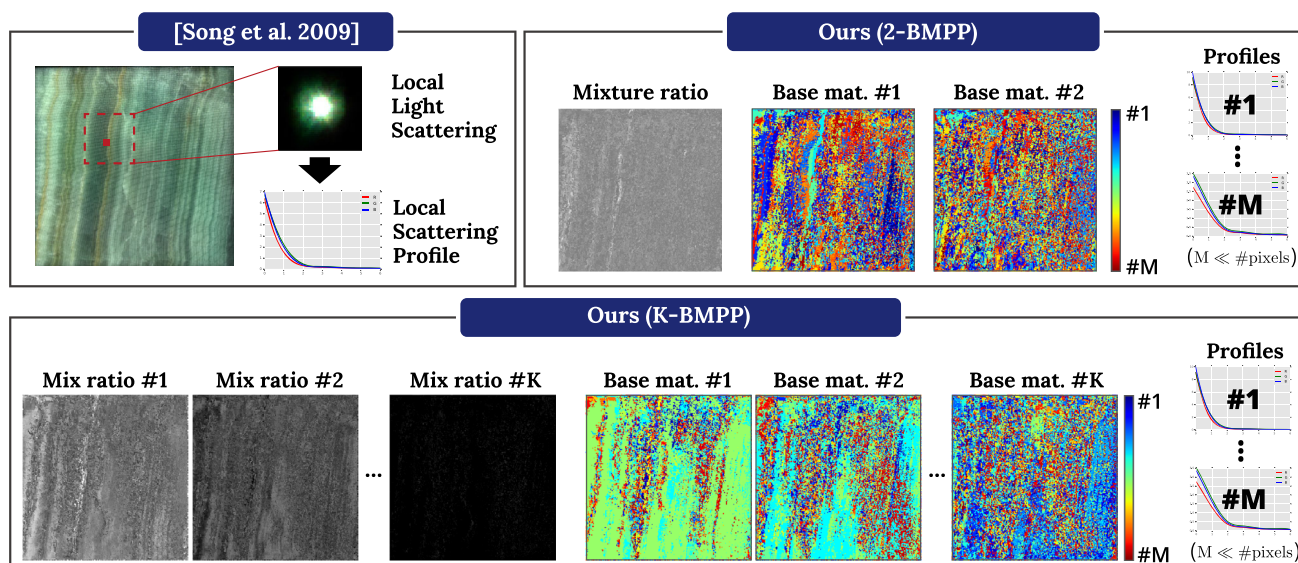
Subsurface scattering in heterogeneous materials is often represented by the bidirectional scattering surface reflectance distribution function (BSSRDF) [4], in which explicit material distribution inside translucent objects does not need to be considered. Because the measured BSSRDF of a heterogeneous material also has a significant amount of data, compression methods for the BSSRDF have been investigated to improve its range of applications. In previous studies, Peers et al. [2] and Song et al. [5] applied such approximation techniques using nonnegative matrix factorization, and local scattering profiles, respectively. Unfortunately, the resulting data sizes based on these methods are of the order of megabytes for small pixel areas. Such a compression ratio is insufficient for memory-limited devices or for large pixel areas. While the size of random access and graphics memory has rapidly increased over the last decade, the large data size of measured BSSRDFs is still a challenge from a data storage and processing perspective. Firstly, the size of the memory capacity on mobile devices such as tablets and mobile phones is still limited, even though many graphics applications are currently being developed on these devices. Secondly, the aforementioned studies examined only measured BSSRDFs for small pixel areas. As such, when these parameters are measured for much larger pixel areas, their compression ratios are insufficient in practice.

To address these problems, we propose a data compression method for BSSRDFs of heterogeneous translucent materials. The key insight of this study is that many non-biological materials such as waxes, soaps, candles, and minerals are often comprised of a limited number of base materials. For example, the cross-section surface of the onyx shown in Fig. 1 is comprised of approximately 20–30 base materials. Based on this observation, we represent the scattering properties of

heterogeneous translucent materials by parameterizing only a few base materials. Figure 1 represents a comparison between the proposed compression method and a representative previous method by Song et al. [5] in terms of the visual qualities of rendered graphics. As shown in this figure, the proposed compression method achieves nearly identical visual quality although the size of the output data using the proposed method is only 2.94% compared to that of the previous study.

Figure 2 compares the proposed compression model to that of Song et al. [5]. In their representation, the surfaces of the object are discretized with pixels, and a local scattering profile is assigned to each pixel. The local scattering profile is a one-dimensional function, which is discretized as a piecewise linear function in the previous method. Directionally varying local light scattering is approximated by processing the values of local scattering profiles in the log scale. Instead of assigning a local scattering profile to each pixel, the proposed method assigns several scattering profiles of the base materials to each pixel. We refer to the scattering profile of the base material as the *base scattering profile*. We approximate the local scattering profile of a given pixel by using a linear combination of base scattering profiles in the log scale. In this case, we assume that a target material is comprised of  $M$  base materials and approximate a local scattering profile of a pixel using  $K$  out of  $M$  base scattering profiles ( $K < M$ ). The optimization problem to obtain the proposed compression model is formulated in a similar manner to that of Song et al. [5]. However, the formulation of the proposed approach requires the assignment of  $K$  different base materials to each pixel, which results in an NP-hard combinatorial optimization problem. In addition, the task of solving for the linear combination of weights and the base scattering profiles is formulated as a nonlinear constrained problem, although a simple linear optimization approach was used in the previous study. To solve this complex problem, we proposed an optimization method by alternating between two different solvers for the combinatorial and nonlinear constrained problems. In the case of combinatorial optimization, we also propose a randomized algorithm based on PatchMatch [6], which was originally a method for texture synthesis.

In the proposed method, the base materials are indexed by 8-bit unsigned integers, and the linear combination weights for the base scattering profiles are represented by 32-bit floating point numbers. Therefore, the data size assigned to each pixel is  $(8 + 32) \times K = 40K$ . We empirically determined that  $K = 2$  is typically sufficient to compress the BSSRDF data without significant loss of visual quality in rendered graphics. In this case, it is possible to further reduce the data size of a single pixel to 48 bits, which is significantly less than the local scattering profile that was assigned to each pixel in the previous method. The resulting data size for the proposed method is discussed in further detail in Sect. 3.



**Fig. 2** A measured BSSRDF is represented by light attenuation factors for arbitrary pairs of surface positions. Local scattering properties of the BSSRDF can be represented in the image of local light scattering. Song et al. [5] compressed this form with local scattering profiles by approximating the directionally varying local light scattering with multiple isotropic local scattering profiles. The proposed method further approximates the local scattering profiles with the profiles of base materials. The number of base materials  $M$  is much smaller than the number of pixels. From  $M$  base materials, a different set of  $K$  base materials is assigned to each surface positions. Then, the scattering profiles of the  $K$  base materials are blended using the mixture ratio. This figure illustrates the magnitude of the mixture ratios with a grayscale image, and the distribution of the base material indices using different colors (the colors themselves do not make any sense)

In summary, the proposed data compression method for heterogeneous translucent materials has the following characteristics:

- it achieves an efficient representation for heterogeneous subsurface scattering that requires only  $40K$  bits for each pixel when  $K$  base materials are assigned to the pixel,
- the resulting data size can be determined flexibly by changing the value of  $K$ ,
- for example, with  $K = 2$ , approximately 30 times smaller data size than that of the state-of-the-art method is achieved,
- resulting compressed data can be used in standard real-time rendering algorithms without incurring computational overhead or the loss of visual quality.

To demonstrate the practical applicability of the proposed method with regard to memory-limited devices, we implemented a rendering system on several mobiles and tablets. It was determined that the small data size of the model is preferred to reduce the time to load the data onto such devices.

## 2 Related work

### 2.1 Subsurface scattering models

Light transport behavior in translucent materials is theoretically described according to the radiative transfer theory [7],

and most of the practical approximations used in the computer graphics community are derived based on this theory. For homogeneous materials, Jensen et al. [8] proposed a practical dipole model derived from an analytical dipole diffusion approximation, and Donner and Jensen [9] extended the dipole model to multilayered materials, such as human skin. d'Eon and Irving [10] performed an elaborate analysis of subsurface light transport and introduced a computation model referred to as the quantized diffusion model. Frisvad et al. [11] proposed a directional dipole model to appropriately handle the directional effect of incident light. Recently, Frederickx and Dutré [12] introduced a variant of the dipole model that considers the relationship between the incident and outgoing directions induced by strong forward scattering.

For heterogeneous materials, most surface-based representations are based on the BSSRDF [4]. The main advantage of the BSSRDF is that it does not explicitly represent material distribution in object volumes. A BSSRDF  $S(x_i, \omega_i; x_o, \omega_o)$  describes the ratio of exiting radiance at position  $x_o$  to direction  $\omega_o$  to incident radiant flux at  $x_i$  from direction  $\omega_i$ . When BSSRDFs are used for heterogeneous materials, they are often discretized and compressed to provide greater flexibility in practical applications. For example, nonnegative matrix factorization [2] and local scattering profiles [5,13] have been used to compress the discretized form of the functions. However, these representations focus mainly on physical accuracy rather than data size. Thus, the data size

aspect of BSSRDF representations has not attracted much attention despite the long-standing research on BSSRDF representations.

The proposed data compression method also targets BSSRDFs just as the aforementioned previous studies. However, compared to the previous methods, we focus more on the final data sizes of the compressed BSSRDFs. The proposed model achieves nearly the same visual quality in rendering results with a data size approximately 30 times smaller compared to the previous methods.

## 2.2 Translucency rendering

Here, we briefly introduce real-time rendering algorithms for translucent materials. We focus primarily on how BSSRDFs or other numerical models for subsurface scattering are used in rendering. Three classes of rendering algorithms are introduced, i.e., volumetric, image-space, and surface-based methods.

*Volumetric methods* simulate the light diffusion process in material volumes. This simulation can be performed efficiently on graphics hardware. Stam et al. [14] simulated light transport in an optically thick material as a diffusion process in the volume of materials. Wang et al. [3] extended their procedure to heterogeneous translucent materials by solving the diffusion equation for tetrahedralized object volumes. A similar method with a data structure, which is known as the light propagation volume (LPV) [15], is often used in real-time applications. Although the LPV is typically used to represent weak scattering in aerosols, it has also been applied to render optically dense translucent materials [16,17]. The main drawback of these volumetric approaches is that they often produce blurry results because of the utilization of volumetric parameter mappings on low-resolution discretized grids. To obtain accurate rendering results, the object volumes must be precisely divided into small grids. However, storing different scattering parameters for such small grids requires significant memory resources. Consequently, volumetric methods are not suitable for the use in memory-limited devices.

*Image-space methods* compute the BSSRDF convolution of surface irradiance on texture space or screen space. Lensch et al. [18] divided subsurface light transport into local and global components and computed the local component for close point pairs using texture-space image filtering. Shah et al. [19] proposed an interesting approach based on image-space splatting for homogeneous translucent materials. Chen et al. [20] later extended their approach to heterogeneous materials. Currently, methods that employ screen-space filtering are commonly used in real-time graphics applications. In these methods, diffuse reflectance components are initially calculated and stored in a G-buffer. Then, the diffuse

reflection components are blurred in the screen space [21,22]. Although these methods of screen-space filtering are computationally efficient, and the resulting visual quality is reasonable, the filtering operation is usually based on empirically determined parameters. Therefore, these methods cannot be directly used to physically measured BSSRDFs. *Surface-based methods* rely on BSSRDFs to represent light attenuation factors between pairs of surface positions. Dachsbacher and Stamminger [23] sampled virtual point lights (VPL) from an irradiance buffer of shadow maps and evaluated the appearance of objects by summing the contributions from VPLs. Rather than computing vertex-to-vertex light transport, Mertens et al. [24] computed area-to-area light transport by clustering triangular meshes of the target object. Recently, more sophisticated approaches using screen-space splatting [19] and multiple irradiance buffers [25] have been proposed. However, the aforementioned methods are all based on dipole approximation [8], which is only applicable to homogeneous materials. For heterogeneous materials, BSSRDFs are typically approximated and compressed to improve the computational efficiency of graphics hardware. In previous studies, compressed BSSRDFs were rendered by implementing an accelerated BSSRDF evaluation method known as hierarchical integration [26] with graphics hardware [2,5]. However, these studies reported that they achieved at most 2 frames per second (fps) using computer systems of 10 years ago. Therefore, this performance is considered to be far from real-time frame rates, even if the latest computers are used. Recently, Chen et al. [20] introduced a real-time multiresolution splatting algorithm using BSSRDFs and achieved more than 30 fps using computer systems of 5 years ago. Considering the improvement in computation power of graphics hardware, their method can achieve real-time frame rates of more than 60 fps using current technology.

We obtain memory-efficient BSSRDFs by further reducing the data sizes of the previous compression method using local scattering profiles [5]. The BSSRDF representation of our compression method is compatible with other existing BSSRDF representations, and it does not incur any computational overhead. As demonstrated in the later sections, the proposed method achieves real-time frame rates when used, for example, with a real-time rendering algorithm based on multiresolution splatting [20].

## 3 BSSRDF compression

In the following explanation, we used the symbols listed in Table 1. We do not explicitly provide the definitions of these symbols for simplicity of explanation.

**Table 1** The notations used in this paper

Symbol	Meaning
$M$	# of Base materials that comprise a target object
$K$	# of Base scattering profile assigned to each pixel
$S(x_i, \omega_i; x_o, \omega_o)$	BSSRDF
$F_{dr}(x)$	Fresnel diffuse reflectance at $x$
$F_t(x, \omega_i)$	Fresnel transmission for $x$ and $\omega$
$R_d(x_i, x_o)$	Diffuse reflectance between $x_i$ and $x_o$
$P_x(r)$	Scattering profile at $x$ in distance $r$
$\hat{P}_x(r)$	Log-scale scattering profile at $x$ , ( $= \log P_x(r)$ )
$\hat{P}_b(r)$	Log-scale base scattering profile of base material $b$
$\hat{p}_b^h$	$h$ th coefficient of the log-scale base scattering profile $P_b$
$b_{x,k}$	$k$ th base material assigned to $x$
$t_{x,k}$	$k$ th linear combination weight at $x$
$\Theta_x$	Set of parameters for $x$ , i.e., $\{b_{x,k}\}_k$ and $\{t_{x,k}\}_k$
$\mathcal{B}$	Set of base materials
$a$	Acceptance probability for updating base material assignments
$p$	Progression ratio to increase the acceptance probability $a$
$N_{\text{iter}}$	# of Iteration for the proposed alternating optimization method

### 3.1 Background

The subsurface scattering behavior of light can be described generally by BSSRDFs [4]. A BSSRDF  $S(x_i, \omega_i; x_o, \omega_o)$  can be separated into two components, i.e., a local component and a global component. As in the previous studies [1,2,5], we restrict our focus to the global components of BSSRDFs, which we denote by  $S_d$ . For optically dense materials, approximation based on only  $S_d$  can sufficiently reproduce the scattering properties of the original BSSRDFs. The function  $S_d$ , which is also known as the diffuse BSSRDF [8], can be further decomposed as:

$$S_d(x_i, \omega_i; x_o, \omega_o) \approx \frac{1}{\pi} F_{dr}(x_o) R_d(x_i, x_o) F_t(x_i, \omega_i),$$

where  $F_{dr}$  denotes the Fresnel diffuse reflectance at  $x_o$ ,  $F_t$  denotes Fresnel transmittance at  $x_i$ , and  $R_d$  denotes the diffuse reflectance function. In this form, a standard eight-dimensional BSSRDF is simplified into a product of lower-dimensional functions. Among these functions, only  $R_d$  describes the properties of subsurface light transport, i.e., a light attenuation factor between each pair of surface positions  $x_i$  and  $x_o$ . For heterogeneous materials, light attenuation factors differ for different position pairs. By denoting the  $N$  discretized surface positions as  $x_1, x_2, \dots, x_N$ , we can represent the attenuation factors in a matrix form  $\mathbf{R}$  where  $R_{ij} = R_d(x_i, x_j)$ .

The proposed BSSRDF compression is inspired by the previous method by Song et al. [5], which uses local scattering profiles to compress BSSRDFs. Here, we first introduce

their compression model; then, we describe the proposed compression model and how its modeling parameters are obtained. Both the previous and the proposed methods are based on physically measured BSSRDFs by Peers et al. [2]. Their BSSRDFs are measured for several flat materials and formulated using the aforementioned matrix form with  $\mathbf{R}$ . Song et al. assigned a local scattering profile to each surface position and approximated the diffuse reflectance function  $R_d(x_i, x_o)$  using the profiles  $P_{x_i}$  and  $P_{x_o}$  at two surface positions  $x_i$  and  $x_o$ :

$$R_d(x_i, x_o) \approx \sqrt{P_{x_i}(r) P_{x_o}(r)}, \tag{1}$$

where  $r$  denotes the Euclidean distance between  $x_i$  and  $x_o$  on the flat surface of a material. By taking the logarithms of both sides of the equation, Eq. 1 can be rewritten as follows:

$$2 \log R_d(x_i, x_o) = \log P_{x_i}(r) + \log P_{x_o}(r). \tag{2}$$

For simplicity, we use the hat symbol to indicate that the function is in the log scale, e.g.,  $\log P_x$  is denoted as  $\hat{P}_x$ . In the representation by Song et al., a log-scale scattering profile is further approximated as a piecewise linear function:

$$\hat{P}_x(r) \approx (1 - w(r)) \hat{P}_x^h + w(r) \hat{P}_x^{h+1}, \tag{3}$$

when distance  $r$  satisfies:

$$\frac{h}{n} r_{\max} \leq r \leq \frac{h+1}{n} r_{\max}.$$

Here,  $r_{\max}$  denotes the maximum scattering radius, and the range between 0 and  $r_{\max}$  is divided into  $n$  intervals. The coefficient  $\hat{P}_x^h$  is the value of the local scattering profile for  $r_h = hr_{\max}/n$ . The linear interpolation weight  $w(r)$  is commonly defined as  $w_x^h(r) = nr/r_{\max} - h$  for every  $x$ . The profile coefficients  $\{\hat{P}_x^h\}$  for all pixel  $x$  and all interval indices  $h = 0, \dots, n$  are obtained by minimizing the least square errors over the original diffuse reflectance function. Song et al. minimized this function by simply solving a linear system, which was obtained by taking the derivative of the least square error function.

### 3.2 Compression by scattering profile blending

As described in the previous subsection, the compression method proposed by Song et al. [5] assigns different local scattering profiles to different surface positions. However, this expression is obviously redundant when the target material is composed of a few base materials. For example, the cross-section surface of the onyx shown in Fig. 2 contains at most 20 base materials, whereas Song's representation assigns different scattering profiles to its target area of  $229 \times 229$  pixels. Rather than the per-pixel assignment of the local scattering profiles, the proposed compression method represents each local scattering profile using the scattering profiles of the base materials. The scattering profiles of the base materials are referred to as the *base scattering profiles*.

Figure 2 shows the compressed BSSRDF representation for the proposed method. According to the number of base materials assigned to a single pixel, we refer to the representation of  $K$ -BMPP as an abbreviation of “ $K$  base materials per pixel.” The simplest representation with base scattering profiles is obtained by assigning a single base material profile to a single location (1-BMPP). This can be extended by assigning two or more base scattering profiles to each location. To represent the local scattering profile at a given location, the weighted averages of the assigned base scattering profiles are taken in the log scale. As shown in Fig. 2, each pixel generally has data for  $M$  mixture ratios and  $M$  indices for the base materials. The 2-BMPP model, which is a special case of the  $K$ -BMPP model, can further reduce the resulting data size because two mixture ratios, i.e.,  $t_1$  and  $t_2$ , can be represented with a single  $t$  as  $t_1 = t$  and  $t_2 = 1 - t$ . Thus, only one illustration is shown for the mixture ratio of the 2-BMPP model in Fig. 2.

We begin with a representation of the local scattering profiles in the proposed compression method. Let  $\mathcal{B} = \{b_1, \dots, b_M\}$  be a set of  $M$  base materials and  $P_b$  be the base scattering profile for a base material  $b$ . As represented in Eq. 3, we approximate a log-scale scattering profile  $\hat{P}_x$  at position  $x$  with a convex combination of the log-scale base scattering profiles  $\hat{P}_b$ :

$$\hat{P}_x(r) \approx t_x^1 \hat{P}_{b_1}(r) + \dots + t_x^M \hat{P}_{b_M}(r),$$

where  $\sum_i t_x^i = 1$  and  $t_x^i \geq 0$  for  $\forall i$ . However, assigning all base materials to each pixel is redundant because a single position is often associated with only a few base materials. Thus, the proposed method assigns up to  $K$  of  $M$  base materials  $b_{x,1}, \dots, b_{x,K}$  ( $K < M$ ) to a single position  $x$ . Then, the local scattering profile at  $x$  is rewritten as follows:

$$\hat{P}_x(r) \approx \sum_{k=1}^K t_{x,k} \hat{P}_{b_{x,k}}(r), \quad \sum_{k=1}^K t_{x,k} = 1, \quad (4)$$

where  $t_{x,k} \geq 0$  is the mixture weight at position  $x$ . Eventually, the compressed BSSRDF obtained using the proposed method is described using the indices of the base materials, i.e.,  $\{b_{x,1}, \dots, b_{x,K}\}$ , the mixture ratios for these base materials, i.e.,  $\{t_{x,1}, \dots, t_{x,K}\}$ , and the profile coefficients  $\hat{P}_b^h$  for  $b \in \mathcal{B}$  at a sampling distance  $r_h$ .

### 3.3 Parameter optimization

A compressed BSSRDF of the proposed method is obtained by minimizing the least square error of the input diffuse reflectance function  $R_d$ . However, compared to the previous method, the minimization problem for the proposed method involves two difficulties. Firstly, the optimization problem is a constrained nonlinear problem, while the previous methods are formulated as a simple nonnegative matrix factorization [2] or linear problem [5]. Secondly, the assignment of the optimal base materials represented with indices to each pixel is an NP-hard combinatorial problem. Therefore, it requires an inordinately long computational time. To address these problems, the nonlinear optimization and combinatorial optimization problems are solved alternatively as opposed to the simultaneous optimization of all the parameters. In addition, we propose a randomized algorithm based on PatchMatch [6] to solve the combinatorial optimization problem. Pseudocodes for the proposed optimization method are provided in Algorithms 1 and 2.

In the first step, the indices for the base material for each pixel are fixed. The profile coefficients and mixture ratios are updated by solving a nonlinear minimization problem. Using Eqs. 2, 3, and 4, the cost function can be defined in the least squares sense:

$$\begin{aligned} E_{\text{LS}} = & \frac{1}{2} \sum_{x,y} \varphi(\|x - y\|) \left[ \left( (1 - w(\|x - y\|)) \sum_{k=1}^K t_{x,k} \hat{P}_{b_{x,k}}^h \right. \right. \\ & + w(\|x - y\|) \sum_{k=1}^K t_{x,k} \hat{P}_{b_{x,k}}^{h+1} + (1 - w(\|x - y\|)) \sum_{k=1}^K t_{y,k} \hat{P}_{b_{y,k}}^h \\ & \left. \left. + w(\|x - y\|) \sum_{k=1}^K t_{y,k} \hat{P}_{b_{y,k}}^{h+1} \right) - 2 \log R_d(x, y) \right]^2. \quad (5) \end{aligned}$$

**Algorithm 1** Pseudocode for parameter optimization

**Input:** Initial acceptance ratio  $a_0$  and # of iteration  $N_{iter}$   
 1: Initialize  $a = a_0, p = (1/a_0)^{2/N_{iter}}$   
 2: **for**  $i = 0, 1, 2, \dots, N_{iter} - 1$  **do**  
 3:   **Step 1:** Optimize  $\{\hat{P}_b^h\}$  and  $\{t_{x,k}\}$ .  
 4:   Solve the problem in Equation 6 with L-BFGS-B ( $K = 1, 2$ ) or augmented Lagrangian ( $K \leq 3$ ).  
 5:   **Step2:** Optimize base material assignments  $\{b_{x,k}\}$ .  
 6:   **for** each pixel  $x = (\xi, \eta)$  **do**  
 7:     **if**  $i$  is odd **then**  
 8:        $x' \leftarrow (\xi - 1, \eta)$                      $\triangleright$  Left neighbor  
 9:       UPDATEWITH( $x, x', a$ )  
 10:        $x'' \leftarrow (\xi, \eta - 1)$                 $\triangleright$  Upper neighbor  
 11:       UPDATEWITH( $x, x'', a$ )  
 12:     **else**  
 13:        $x' \leftarrow (\xi + 1, \eta)$                 $\triangleright$  Right neighbor  
 14:       UPDATEWITH( $x, x', a$ )  
 15:        $x'' \leftarrow (\xi, \eta + 1)$               $\triangleright$  Lower neighbor  
 16:       UPDATEWITH( $x, x'', a$ )  
 17:     **end if**  
 18:     Sample random pixel  $\tilde{x}$  from the entire image area  
 19:     UPDATEWITH( $x, \tilde{x}, a$ )  
 20:   **end for**  
 21:    $a \leftarrow \min(ap, 1)$                      $\triangleright$  Increase acceptance ratio  
 22: **end for**

**Algorithm 2** Pseudocode for UPDATEWITH

1: **function** UPDATEWITH( $x, x', a$ )  
 2:    $u \sim U[0, 1]$                              $\triangleright$  Sample uniform random number  
 3:   **if**  $u < a$  **then**  
 4:     Evaluate  $E_{LS}$  terms involving  $x$  and  $x'$ . Store the value to  $E$   
 5:      $t'_k \leftarrow t_{x,k}, b'_k \leftarrow b_{x,k}$  for  $\forall k$                      $\triangleright$  Backup  
 6:      $t_{x,k} \leftarrow t_{x',k}, b_{x,k} \leftarrow b_{x',k}$  for  $\forall k$               $\triangleright$  Update  
 7:     Evaluate  $E_{LS}$  terms involving  $x$  and  $x'$ . Store the value to  $E'$   
 8:     **if**  $E \leq E'$  **then**     $\triangleright$  Not decrease cost  
 9:        $t_{x,k} \leftarrow t'_k, b_{x,k} \leftarrow b'_k$  for  $\forall k$               $\triangleright$  Restore  
 10:    **end if**  
 11:   **end if**  
 12: **end function**

In this equation,  $\hat{P}_b^h$  denotes the  $h$ th coefficient of  $\hat{P}_b$ . Each least square term is multiplied by a weighting factor  $\varphi(r)$ , which resolves the imbalance in the number of least square terms for position pairs. The number of pairs over a short distance is smaller than the number of pairs over a large distance. We use the inverse quadratic function  $\varphi(r) = 1/(1 + r^2)$  for this weighting factor. The domain of the cost function in Eq. 5 is limited by the constraints for the mixture ratios  $t_{x,k}$  because they are in  $[0, 1]$  and their sum is equal to 1.

Based on several experiments, it was determined that the preceding optimization process is numerically unstable because it often falls into a local minima or goes beyond the feasible region. To avoid such problems, we first add a regularization term to ensure the  $C^2$  continuity of the profiles following the previous method [5]:

$$E_{reg} = \frac{1}{2} \sum_{b,h} \left( \hat{P}_b^h - 2\hat{P}_b^{h+1} + \hat{P}_b^{h+2} \right)^2,$$

where  $b \in \mathcal{B}$  and  $h = 0, \dots, n - 2$ . Secondly, we added barrier functions to keep the optimization variables within the feasible region. Since the mixture ratios should be between 0 and 1, we added the following barrier functions to the cost function.

$$E_{barr} = \sum_{x,k} B_0(t_{x,k}) + B_1(t_{x,k}),$$

$$B_0(t) = \begin{cases} -\log(\gamma t) & t > 0, \\ \infty & \text{otherwise,} \end{cases}$$

$$B_1(t) = \begin{cases} -\log(1 - \gamma t) & t < 1, \\ \infty & \text{otherwise,} \end{cases}$$

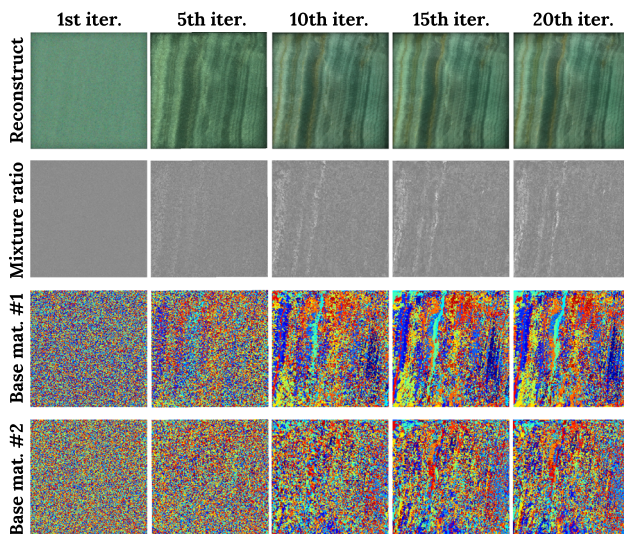
where we typically use  $\gamma = 10^{-3}$  as the default value. Consequently, the optimization problem for the proposed compression method is defined as follows:

$$\left. \begin{aligned} &\text{minimize: } E_{LS} + \lambda_{reg} E_{reg} + \lambda_{barr} E_{barr}, \\ &\text{subject to: } \sum_{k=1}^K t_{x,k} = 1 \text{ for } \forall x, \\ &\qquad\qquad t_{x,k} \geq 0 \text{ for } \forall x, k. \end{aligned} \right\} \quad (6)$$

The weighting factors in Eq. 6 are empirically determined as  $\lambda_{reg} = \lambda_{barr} = 10^{-3}$ .

The optimization problem in Eq. 6 is a nonlinear constrained problem with both inequality and equality constraints. Despite this complexity, this class of optimization has been extensively studied in applied mathematics, and it can be efficiently solved using the augmented Lagrangian method [27]. In addition, in cases where  $K = 1$  and  $K = 2$ , the preceding optimization can be simplified. For  $K = 1$ , we do not need to consider the weighting factors for the base scattering profiles; therefore, the optimization problem in Eq. 6 can be simplified as a box-constrained nonlinear problem. For  $K = 2$ , we only need to assign a single weight  $t_x$  to pixel  $x$  because we can represent  $t_{x,1} = t_x$  and  $t_{x,2} = 1 - t_x$ . Therefore, only the box constraints  $t_x \in [0, 1]$  are required. These nonlinear optimizations with box constraints can be solved more efficiently with the L-BFGS-B method [28], which is a variant of the quasi-Newton method for box-constrained nonlinear problems.

The second partial optimization step is inserted after every iteration for these iterative nonlinear solvers, i.e., the augmented Lagrangian and L-BFGS-B methods. In the second step, we fix the profile coefficients  $\hat{P}_{b_{x,k}}^h$  and update the base material indices  $b_{x,k}$  and mixture ratios  $t_{x,k}$ . To find optimal indices and mixture ratios, we need to solve an NP-hard combinatorial optimization problem. Rather than solving



**Fig. 3** Parameter updates in the 2-BMPP model. The iterative optimization process for the proposed compression method gradually improves the detailed appearance of the onyx. Similarly, the mixture ratios and indices of the base materials are updated to form pixel clusters. Here, we observe that the clusters are grown along the vertical layers of the onyx

this complex problem directly, we update the base material indices and mixture ratios approximately with a randomized process inspired by PatchMatch [6]. Let  $\Theta_x$  be a set of parameters at pixel  $x$  that includes the base material indices  $\{b_{x,k}\}$  and  $\{t_{x,k}\}$ . The parameter set  $\Theta_x$  is temporarily substituted by the parameters  $\Theta_{x'}$  of candidate pixel  $x'$ . The parameters  $\Theta_{x'}$  are accepted as the parameters of  $x$  provided that this substitution reduces the least square error in Eq. 5. In this updating process, the immediate neighbors of  $x$  to the left and above are used as candidate pixels in even-numbered iterations, and those to the right and bottom are used in odd-numbered iterations. We also use the random pixels chosen from the entire pixels as candidates. Unlike the original PatchMatch, we leverage a probabilistic update strategy to avoid unexpectedly updating parameters using the scattering profile coefficients that are not sufficiently converged. In the  $i$ th iteration, the parameters are updated based on the probability of  $\min(a_0 p^i, 1.0)$  even if the error values are reduced. Typically, we use  $a_0 = 0.1$  and set the value  $p = (1/a_0)^{2/N_{\text{iter}}}$  such that  $a_0 p^i = 1$  when  $i = N_{\text{iter}}/2$ . Figure 3 shows the update processes of the base material indices, mixture ratios, and reconstructed appearances in each iteration of the 2-BMPP model. In this experiment, the number of total iterations  $N_{\text{iter}} = 20$  and the indices and mixture ratios are only partially updated when the number of iteration was less than  $N_{\text{iter}}/2 = 10$ .

In our implementation, we initialized the parameters with random values before performing the above alternating optimization process. The profile indices were chosen randomly for each pixel from  $\{1, \dots, M\}$  such that any two of the base material indices did not overlap. Mixture ratios were all initialized with  $1/K$ , and the profile coefficients in the log scale

were initialized with random real numbers between  $-1.0$  and  $1.0$ .

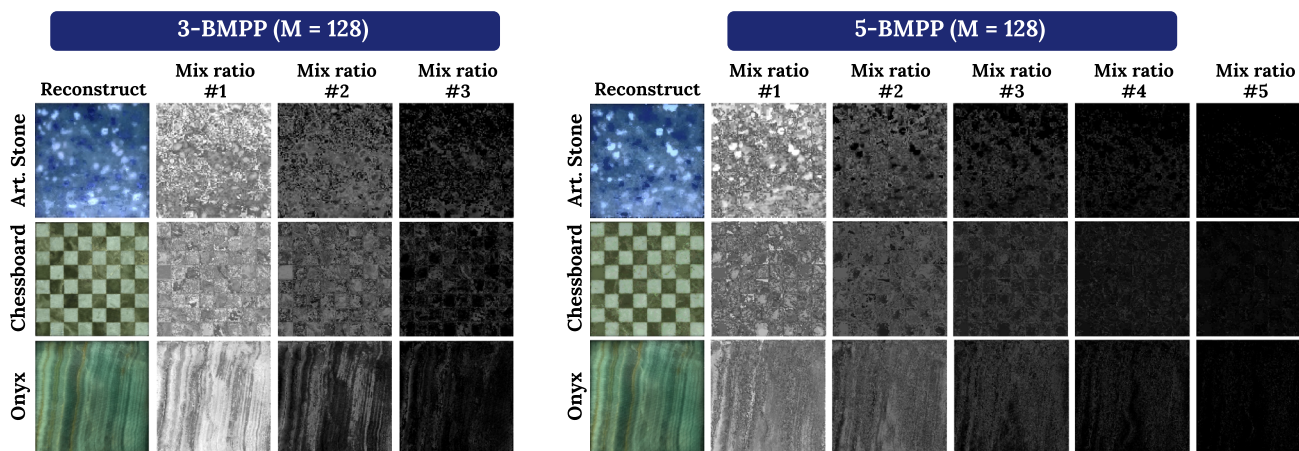
### 3.4 Real-time rendering

Since BSSRDFs compressed by the proposed method are compatible with BSSRDFs in the general forms, they can be used in ordinary rendering algorithms based on BSSRDFs. The rendering results presented in this paper were computed using a multiresolution splatting algorithm proposed by Chen et al. [20]. Their algorithm first renders multiresolution G-buffers from the light sources; then, irradiance samples are positioned on these multiresolution buffers. Subsequently, the splatting-based algorithm [19] is performed for multiresolution render buffers, and the buffers are accumulated to obtain the final rendering result. In our implementation, log-scale values of the profile coefficients are sent to shader programs as uniform buffer objects, and other parameters, i.e., profile indices and mixture ratios, are sent as textures. In our experiment, the compression was typically performed with  $M \leq 128$  base materials. Therefore, we can represent the profile indices with 8-bit unsigned integers, which can represent integers between 0 and 255. The mixture ratios are represented with 32-bit floating point numbers. The BSSRDF values are evaluated in a pixel shader using Eqs. 2, 3, and 4. This evaluation of BSSRDFs in the shader program is slightly more complex than the evaluation that is used in the previous study by Song et al. [5]. However, the computational overhead using our compressed BSSRDFs is nearly negligible considering the complexity of the other computations in rendering algorithms.

### 3.5 Estimated data size

Let  $N$  be the number of pixels in the target area, then the estimated data size of an uncompressed BSSRDF is  $(N^2 \text{ pixel pairs}) \times (4 \text{ bytes}) \times (3 \text{ color channels}) = 12N^2$  bytes when each attenuation factor is represented with three-channel 32-bit floating point numbers. If only the neighboring pixels are considered as the counterparts of the pairs, it is reduced to  $12NW^2$ , where  $W$  is the window size for neighboring pixels. In the compression method by Song et al., a local scattering profile is assigned to each pixel in the target area. When each local scattering profile is defined by a piecewise linear function with  $S$  discrete coefficients ( $S \ll N$ ), the data size is estimated as  $N \times S \times 4 \times 3 = 12NS$  bytes. Typically, the value of  $S$  is determined as  $S = (W - 1)/2$ . In contrast, the proposed model only considers the profile functions of  $M$  base materials ( $M \ll N$ ). Therefore, the data size for a base scattering profile is as large as  $12MS$  bytes. In addition, we assign  $K$  pairs of 8-bit unsigned integers and 32-bit floating point number to a pixel, which corresponds to the base material indices and mixture ratios, respectively.





**Fig. 4** Magnitudes of the mixture ratios in the 3-BMPP and 5-BMPP models. For all the three materials and both the 3-BMPP and 5-BMPP models, the mixture ratios #1 and #2 assume large values, and the other three ratios assume values close to zero. These results imply that two or three materials are enough for our compression model to appropriately approximate the original BSSRDFs

The data size for these indices and mixture ratios is estimated as  $(8 + 32) \times K \times N = 40KN$  bits, i.e.,  $5KN$  bytes. Overall, the estimated data size of the proposed compressed BSSRDF is  $12MS + 5KN$  bytes. In the case of  $K = 5$ , the proposed model is more efficient than Song's method if  $12MS + 25N < 12NS$ , which can be simplified to  $N < 0.48(N - M)S$ . Because Song et al. reported that they typically used  $S = 25$  in their implementation, this condition can be rewritten as  $M < 0.92N$ . Consequently, the data size of the proposed compression method is smaller than that of the previous method in most cases. The advantage of the proposed method in terms of the data size is further improved when a large number of pixels  $N$  are included in the target area. In addition, for  $K = 2$ , it is not necessary to use two floating point numbers for the mixture ratios of each pixel because two mixture ratios can be represented by either of them only. Therefore, the data size for the 2-BMPP model can be as small as  $12MS + (8 \times 2 + 32)N/8 = 12MS + 6N$  bytes.

### 3.6 Relationship to other data compression methods

Although our idea of using basis functions to compress the BSSRDF data is common with many other compression methods, the problem formulation introduced in this paper is much more complicated than that encountered in common approaches. The base scattering profiles are conceptually similar to clusters with common clustering algorithms such as the  $k$ -means and the fuzzy  $c$ -means methods. In these methods, the assignment of a cluster to a single pixel does not affect neighboring pixels. Therefore, the problem of assigning the clusters to each pixel can be independently solved. However, in the BSSRDF compression problem that we solve as a part of the proposed solution, the assignment of the base scattering profile can affect neighboring pixels as well. This makes the optimization problem much more difficult to solve. Other compression methods such as principal compo-

nent analysis for dimensionality reduction cannot be applied straightforwardly to our problem for the same reason.

Our problem formulation is also similar to that of the Gaussian mixture model in the sense that a target function is approximated with a mixture of multiple basis functions. The Gaussian mixture model approximates the original function using a basis function defined in the same domain. Therefore, to approximate the 8D BSSRDFs or 4D diffuse reflectance functions using the Gaussian mixture model, we need to use 8D or 4D basis functions. However, an accurate approximation of such higher-dimensional function requires dense sampling of the function's domain. Therefore, an approximation using the Gaussian mixture model is not reasonable to approximate high-dimensional BSSRDFs.

Another option is clustering local scattering profiles that are obtained by Song et al.'s approach [5]. When the local scattering profiles are clustered using the ordinary  $k$ -means method, the resulting data size is the same as our 1-BMPP model. We refer to this option as the "LSP-clusters" (local scattering profile clusters) and compare with the proposed method in the following section.

## 4 Results and discussion

We experimented with the compression and rendering system described in this paper using a computer with an Intel Core i7-6700K 4.0 GHz CPU, an NVIDIA GeForce GTX 1070 graphics card, and 32 GB RAM. The compression program was implemented using C++, and the real-time rendering method was implemented using C++ with OpenGL and GLSL as the graphics API. The BSSRDFs used in our experiments were physically measured using the previous method proposed by Peers et al. [2]. The experiments described in this report were performed using  $M = 128$  base materials unless otherwise specified. Owing to space limitations, all

**Table 2** Performance of our compression method and the comparison to the previous study by Song et al. [5]

Material Pixels	Art. stone 108 × 108	Chessboard 222 × 222	Onyx 229 × 229	Blue wax 228 × 88	Yellow wax 110 × 110	Jade 260 × 260
Size (MB)						
Original	163	857	912	281	210	947
Song et al. [5]	2.5	10.7	11.4	4.4	2.6	14.6
Size (kB)						
1-BMPP	39	80	83	48	43	95
2-BMPP	98	326	345	148	103	433
3-BMPP	203	770	817	329	212	1041
5-BMPP	319	1263	1341	529	333	1717
Time (s)						
1-BMPP	3 m 45 s	23 m 10 s	23 m 12 s	6 m 54 s	5 m 17 s	24 m 23 s
2-BMPP	4 m 07 s	24 m 04 s	21 m 48 s	7 m 25 s	4 m 57 s	22 m 56 s
3-BMPP	5 h 45 m	33 h 21 m	35 h 19 m	9 h 50 m	7 h 14 m	38 h 08 m
5-BMPP	8 h 15 m	46 h 41 m	20 h 01 m	13 h 49 m	10 h 21 m	53 h 48 m

results for mixture ratios, base material indices, and reconstructed appearances for  $K = 1, 2, 3,$  and  $5$  are provided in the supplementary materials.

Table 2 summarizes the compression performance for the proposed method and its comparison to a state-of-the-art method [5]. As shown in this table, the proposed method with  $K = 2, M = 128$  resulted in BSSRDF data that are approximately 2,000 times smaller than the original. Compared to the state-of-the-art method [5], these data are approximately 30 times smaller. The size of the resulting data is of the order of a hundred kilobytes which is sufficiently small for use in video game consoles or mobile phones, which typically have relatively limited memory resources. The computation times of the proposed method with  $K = 2, M = 128$  are 4–25 min for input BSSRDFs in Table 2. These compression processes of several minutes should be performed only once for each BSSRDF data.

#### 4.1 Selecting the number $K$

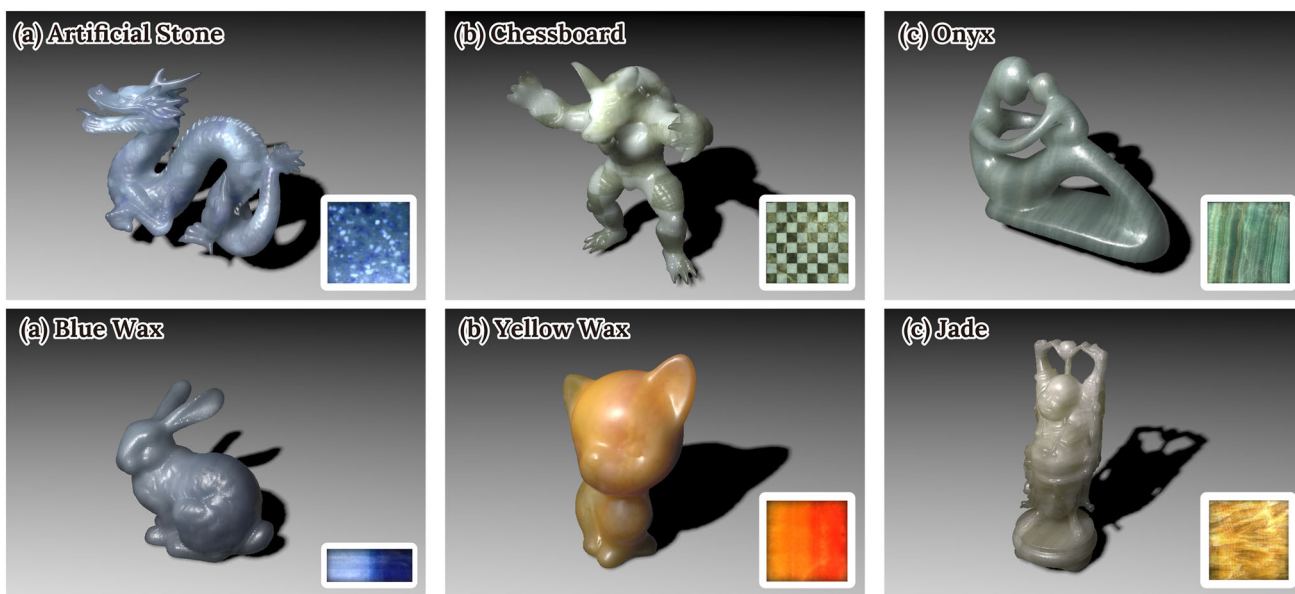
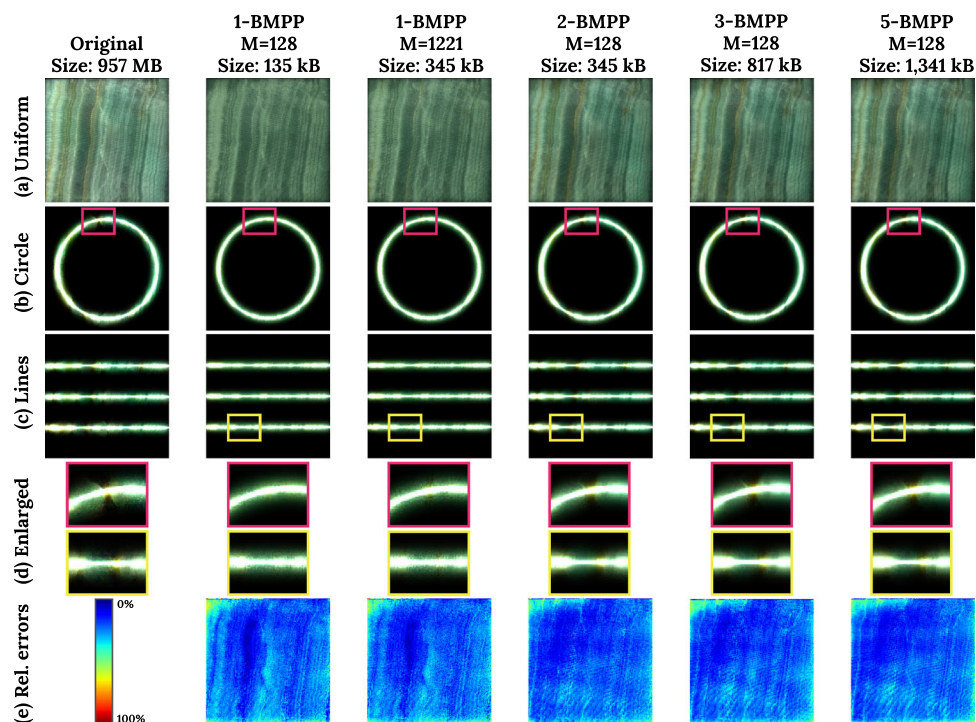
In the proposed compression method, the number  $K$  is an important factor in determining the final data size, computation time for compression, and visual quality of the rendered images. The final data sizes and computation times for different  $K$  values are provided in Table 2. In addition, the appearance of the reconstructed material under different illumination and the relative errors of the diffuse albedo are compared in Fig. 5. Here, a diffuse albedo value  $\rho_d(x)$  for pixel  $x$  is computed by integrating  $R_d$  over the local scattering range  $A$  as  $\rho_d(x) = \int_A R_d(x, y) dy$ , and the relative error is computed as  $\|\rho_d(x) - \rho'_d(x)\| / \|\rho_d(x)\|$  where  $\rho'_d(x)$  is a diffuse albedo value of the compressed diffuse reflectance function  $R_d$ .

As shown in Table 2, a smaller value of  $K$  results in smaller BSSRDF data, whereas fewer relative errors occur for a larger

value of  $K$ , as can be seen in Fig. 5. To provide a more fair comparison between the cases of  $K = 1$  and  $K = 2$ , we also provide the results for  $K = 1$ , which has data sizes that are matched to those of  $K = 2$  by changing the value of  $M$ . By comparing these results in Fig. 5, it is determined that both the reconstructed appearance and relative errors of  $K = 2$  are superior to those of  $K = 1$ , even if the size of the resulting data is equal. Specifically, anisotropic light scattering is appropriately reproduced by the compressed BSSRDF of  $K = 2$ , as shown in Fig. 5d. The improved reproducibility of  $K = 2$  compared to  $K = 1$  is considered because the number of base scattering profiles used to represent a value of  $R_d(x_i, x_o)$  is larger for  $K = 2$  compared to  $K = 1$ . Although only two base scattering profiles are used in the case of  $K = 1$ , four base scattering profiles are used in the case of  $K = 2$ . By comparing the relative errors of  $K = 1, M = 128$  and  $K = 1, M = 1221$ , it was observed that the error and scattering heterogeneity are not improved by increasing the value of  $M$ . According to these observations,  $K = 2$  is more preferable than  $K = 1$  for the efficient reproduction of the input BSSRDF with a specific amount of data.

By comparing the results of  $K = 2$  and  $K = 3$ , the relative errors of for the diffuse albedo and the reconstructed appearances in Fig. 5e are nearly identical, whereas the compression time for  $K = 3$  is significantly longer than that for  $K = 2$ . This is because nonlinear optimization with equality constraints requires considerably more computational processing compared to box-constrained nonlinear optimization. Even if the value of  $K$  is increased to 5, the relative errors and reconstructed appearance are still nearly identical to the case of  $K = 2$ . In addition, the magnitude of the mixture ratios in Fig. 4 implies that using two or three materials at most for each pixel is sufficient to approximate the input BSSRDFs. Figure 4 shows the mixture ratios of the 3-BMPP and 5-BMPP models after they are sorted in descending order for

**Fig. 5** Comparison of reconstructed appearances under different lighting conditions using the proposed compression method with different values of  $K$ . For  $K = 1, 2, 3,$  and  $5,$  the compressed BSSRDFs are computed with  $M = 128$  base materials. For  $K = 1,$  we also provide another result with  $M = 1221$  base materials. The resulting data size for  $K = 1, M = 1221$  is nearly the same as that for  $K = 2, M = 128$



**Fig. 6** Real-time rendering results using the 2-BMPP models computed with the multiresolution splatting algorithm [20]. The performance data of the rendering processes are provided in Table 3

each pixel. As shown in this figure, the mixture ratios #1 and #2 assume particularly large values, whereas the other mixture ratios assume values close to zero. According to the result of these experiments, we consider that the 2-BMPP model is more practical than the  $K$ -BMPP models for  $K \geq 3$ . Eventually, the 2-BMPP model was determined to be the best choice by considering its small data size, reduced computation time, and low relative error. Comparisons are shown in Fig. 5 for the other material samples in the supplementary materials.

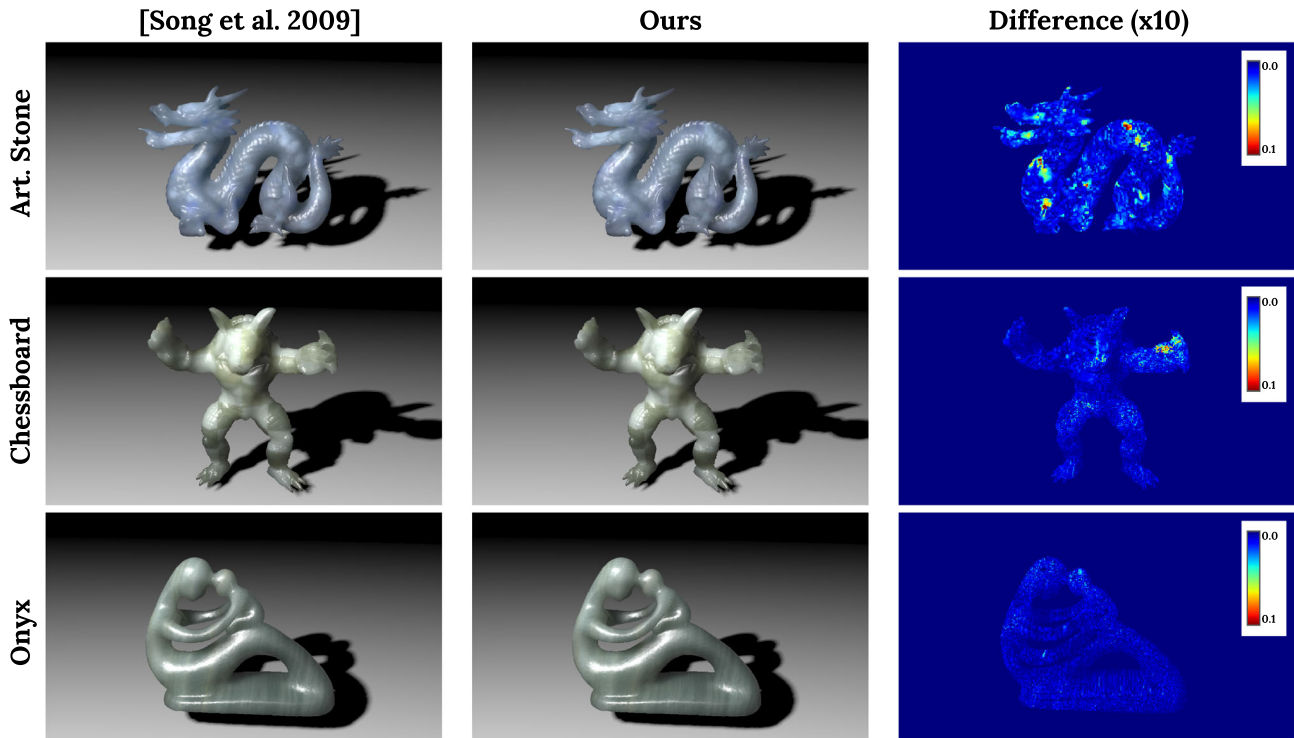
Following the preceding discussion, the case of  $K = 2,$  i.e., the 2-BMPP model, is primarily considered in the following experiments.

### 4.2 Real-time rendering

Figure 6 shows real-time rendering results obtained using the 2-BMPP model. For these images, we compressed the measured BSSRDFs of the materials shown in the insets. These

**Table 3** Rendering performance of the multiresolution splatting algorithm [20] used with BSSRDFs of the 2-BMPP model

	Figure	Model	Meshes	Time (ms)	fps
Art. stone	Figure 6a	Dragon	871,414	9.7	103
Chessboard	Figure 6b	Armadillo	345,944	7.9	127
Onyx	Figure 6c	Fertility	81,782	9.6	104
Blue wax	Figure 6d	Bunny	69,664	6.7	149
Yellow wax	Figure 6e	Kitten	25,246	5.4	185
Jade	Figure 6f	Budda	870,256	12.0	83



**Fig. 7** Comparisons of rendered images with compressed BSSRDFs obtained with the proposed method and the state-of-the-art method [5]. As shown in this figure, our results are nearly identical to those obtained for the state-of-the-art method in terms of visual quality, even though the size of our compressed BSSRDFs is significantly smaller as shown in Table 2. The differences of colors in the rendered images are magnified by a factor of 10 for ease of identification. It should be noted that only a negligible computational overhead is required while using the proposed compression models

images were rendered for the frame buffers with  $1280 \times 720$  pixels. The rendering performance is summarized in Table 3. As shown in this table, the rendering speed was approximately 103 fps for the dragon model with 871 K meshes. The speed is sufficiently fast considering that real-time rendering at 60 fps is insufficient for real-time applications such as computer games, due to the time requirements for non-rendering computations. The rendering results obtained with the 2-BMPP models are compared to that of the previous study in Fig. 7. Even though the size of the data is much less than that of the previous methods, as shown in Table 2, the rendered images are nearly identical. The difference values in this figure represent the Euclidean distance of RGB pixel colors in  $[0.0, 1.0]^3$ , and the distances are magnified by a

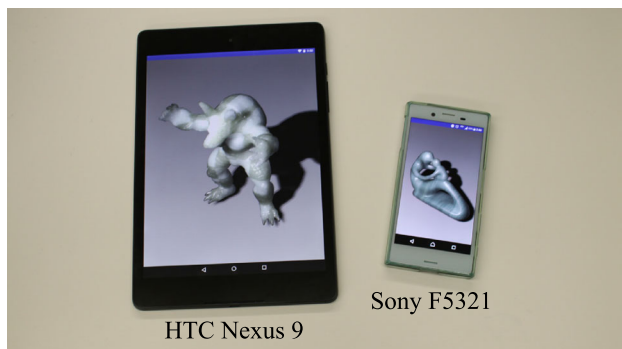
factor of 10 such that they can be readily identified. The supplementary materials include the complete video for these rendering results and comparisons.

We implemented our rendering system on a tablet and a mobile phone to demonstrate the applicability of the proposed method in memory-limited hardware. The hardware configurations are summarized in Table 4. The system was implemented using Android SDK 7.0 and coded with Kotlin 1.12.10<sup>1</sup> [29]. We demonstrate the rendering results in Fig. 8 and the demo video in the supplementary materials.

<sup>1</sup> Kotlin is a modern multiplatform programming language. It primarily focuses on accelerating the development of Android applications, and its compiled binary is executed on Java VM.

**Table 4** Hardware configurations of the tablet and mobile used in the experiment

	HTC Nexus 9 (tablet)	Sony F5321 (mobile)
CPU	Dual core 2.3 GHz	Hexa core 1.8 GHz
GPU	NVIDIA Kepler DX1	Qualcomm Adreno 510
RAM	1.8 GB	2.8 GB

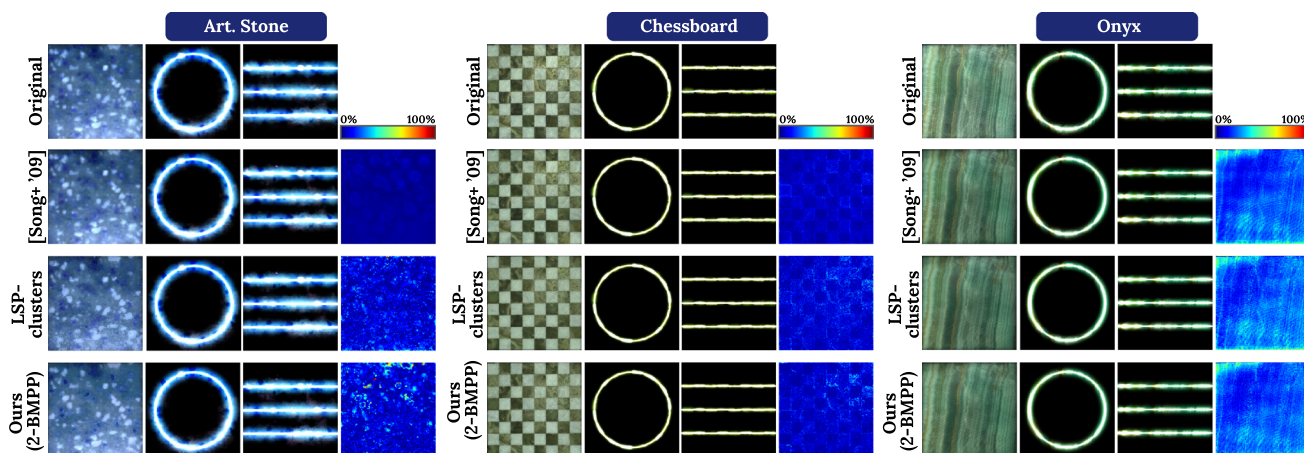


**Fig. 8** Rendering demo on a tablet and a mobile phone

Compared to the rendering performance on computers, the duration of the data loading time can be an issue on such hardware. For the “Onyx” example, 335 kB of the BSSRDF data obtained using the proposed method was loaded in 0.28 s on the HTC Nexus 9, while 11.4 MB of the data obtained using Song et al.’s method was loaded in 6.60 s. This result reveals that the high compression ratio of the proposed method can reduce both the loading time and graphics memory usage.

### 4.3 Experiments

Figure 9 compares the appearances of light scattering among the original data, previous method [5], the LSP-clusters, and the proposed method. We used  $M = 128$  base materials for the proposed method and chose the number of clusters for the LSP-clusters such that the resulting data size was the same as that of the proposed method using  $(K, M) = (2, 128)$ . In this figure, the reconstructed appearances under uniform, circle-shaped, and line-shaped illuminations are shown from left to right, and the relative errors of the diffuse albedo are shown in the rightmost column. By comparing the reconstructed appearances of the 2-BPMM model, previous method, and original data, it was determined that all their appearances are nearly identical. According to the relative errors, the 2-BMPP model performs comparably to the previous method [5] even though the data size of the proposed method is significantly smaller. These results demonstrate that the proposed method with two base materials effectively reduces redundancy in Song et al.’s previous method. Comparing the proposed method with the LSP-clusters, it was determined that the reconstructed appearances and relative errors are almost the same. We further performed a quantitative



**Fig. 9** Appearances of light scattering compared among the original data, approximation with local scattering profiles [5], the LSP-clusters, and our compression method. In this experiment, uniform, circle-shaped, and line-shaped illuminations were exposed to three different materials. For the LSP-clusters, we used the number of clusters such that the resulting data size was the same as that used for the 2-BMPP method with  $M = 128$ . The images in the rightmost column of each group represent the relative errors of the diffuse albedo over the original data. The similarities of the light scattering and relative error values reveal that the proposed compression method effectively reduces the redundancy of the per-pixel scattering profile assignment

**Table 5** Comparison of the average relative errors of compressed BSSRDFs among the proposed method, the previous method by Song et al. [5], and the LSP-clusters

Material	Art. stone	Chessboard	Onyx	Blue wax	Yellow wax	Jade
Avg. relative error						
Song et al. [5]	0.0223	0.0376	0.0413	0.0237	0.0222	0.0610
LSP-clusters	0.0429	0.0438	0.0435	0.0402	0.0619	0.0807
1-BMPP	0.0303	0.0425	0.0434	0.0441	0.0351	0.0728
2-BMPP	0.0372	0.0438	0.0428	0.0276	0.0319	0.0753
3-BMPP	0.0413	0.0452	0.0430	0.0273	0.0307	0.0789
5-BMPP	0.0491	0.0444	0.0427	0.0345	0.0370	0.0773

The average relative errors are calculated by averaging the error values in the error maps as in the bottom row of Fig. 5. The number  $K$  used for the results of the LSP-clusters and 1-BMPP are controlled such that the data size is the same as those of 2-BMPP. The relative error distributions for each pixel are depicted in Fig. 9 and in the supplementary materials

**Table 6** Comparison of PSNR and SSIM among our compression method, the previous study by Song et al. [5], and the LSP-clusters. Since the PSNR and SSIM are metrics for comparing image qualities, we compared the appearance of the reconstructed material under uniform illumination as in Fig. 5. In addition, the number of base materials  $K$  used for the results of the LSP-clusters and 1-BMPP are controlled such that the data sizes are the same as those of 2-BMPP

Material	Art. stone	Chessboard	Onyx	Blue wax	Yellow wax	Jade
PSNR (dB)						
Song et al. [5]	39.7	28.8	18.8	40.5	36.8	26.6
LSP-clusters	25.3	25.2	18.0	21.3	21.4	19.6
1-BMPP	31.3	26.4	18.0	20.7	28.8	20.5
2-BMPP	30.7	26.5	18.6	27.3	31.9	20.9
3-BMPP	26.7	24.8	18.5	27.1	32.9	20.4
5-BMPP	24.6	26.8	18.3	25.3	31.7	20.9
SSIM						
Song et al. [5]	0.998	0.960	0.888	0.999	0.995	0.977
LSP-clusters	0.875	0.855	0.835	0.949	0.949	0.852
1-BMPP	0.970	0.940	0.853	0.843	0.942	0.821
2-BMPP	0.937	0.905	0.870	0.964	0.962	0.823
3-BMPP	0.901	0.873	0.870	0.974	0.962	0.793
5-BMPP	0.854	0.901	0.873	0.959	0.952	0.811

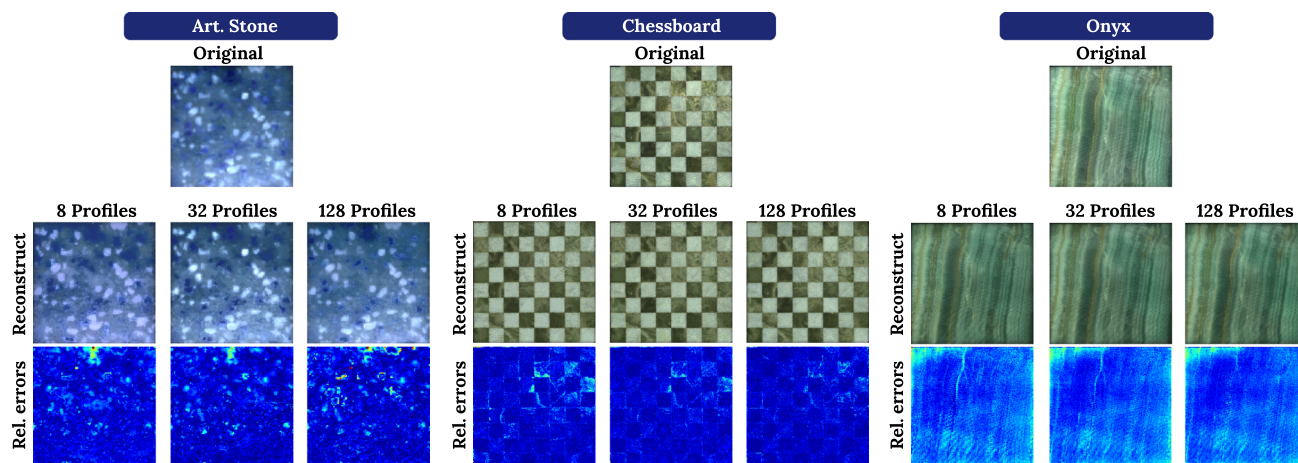
comparison of the proposed method with the LSP-clusters as discussed in the following paragraphs.

Table 5 quantitatively compares the average relative errors of the scattering profiles among the proposed method, previous method [5], and the LSP-clusters. The values are computed by averaging the relative error distribution shown in Fig. 5. In addition, Table 6 shows the results of a quantitative comparison of the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [30] among the above methods. Since these quantitative metrics are both for comparing images, we compared the reconstructed appearances under uniform illumination as shown in the first row of Fig. 5. It is worth recalling that the relative error values in Table 5 compare the qualities of BSSRDFs, while the PSNR and SSIM values compare the qualities of the reconstructed images.

The values in Tables 5 and 6 are computed by comparing the reference BSSRDF data and the compressed data obtained using the proposed method or the previous method. The PSNR and SSIM values are only slightly lower than those of the previous method even though the data size obtained

using the proposed method is significantly smaller than those obtained by the previous method. In addition, compared to the results for the LSP-clusters, the proposed method with 2-BMPP outperforms the LSP-clusters for all values except in the case of the SSIM for “Jade.” For the “Jade” example, the SSIM values of the proposed method are smaller than that of the previous method. The resulting base material assignments can sometimes converge insufficiently for such precise and complex mixtures, even for an increase in the number of base materials  $K$ . The development of a more efficient optimization for material assignment will be addressed in future work.

Figure 10 shows the reproduced appearances and relative errors of the diffuse albedo for different numbers of scattering profiles. As can be seen in this figure, the reproduced appearances obtained with 8 and 32 base materials are sufficiently comparable with those obtained using the original data. As the number of profiles increases, the details of the reproduced appearance become clearer. For example, the appearance of the small white circles in the upper left of “Art.



**Fig. 10** Comparison of reproduced appearance and relative errors of diffuse albedo obtained for different numbers  $M$  of the total base materials. When the values of  $K$  were relatively small, it was possible to reproduce the appearance obtained using the original data. By increasing the number of base materials used for the compression, the relative errors decrease, and the texture details become more apparent

stone” becomes close to that of the original data, and the vertical lines of the “Onyx” become more apparent. According to these results, users can flexibly select the number of base materials depending on their size requirements for BSSRDF data. Even when  $M = 128$  is selected for high visual quality, the resulting data size is still significantly smaller than that of the previous methods as shown in Table 2.

## 4.4 Discussion

### 4.4.1 Target object shapes

The compression results shown in this paper are all for the BSSRDFs on flat objects, even though such flat shapes are not common in real-world materials. On the other hand, the proposed compression method is independent of the objects shape. Therefore, it can be extended to the case of non-flat objects as far as the distance between two surface positions can be calculated. Such BSSRDFs for general shapes are measured, for example, by the method of Goesele et al. [1], and our compression method can be immediately applied to their measurements. When we use the measured BSSRDFs in real-time rendering, the BSSRDFs measured for flat objects can be applied to non-flat shapes as shown in Fig. 6 based on the assumption that the target shapes are locally flat and the materials are optically thick.

### 4.4.2 Computational time for optimization

To obtain compressed BSSRDFs of K-BMPP models with  $K \geq 3$ , the current implementation is time-consuming as shown in Table 2 because the optimization problem for  $K \geq 3$  needs to be solved using the slow-convergence augmented Lagrangian method. The computational bottleneck

in the optimization is the evaluation of the least squares cost function in Eq. 5 and its derivative. However, evaluations of these cost function and its derivative can be accelerated with GPUs. Therefore, the problem of the computational time can be reasonably addressed. In addition, the 2-BMPP models, which can be yielded in a significantly reduced computational time, are practically sufficient in most cases as shown by the experimental results in the previous subsections.

### 4.4.3 Random versus cluster-based initialization

As described in Sect. 3, the parameters of the compressed BSSRDF are initialized with random numbers before optimization. On the other hand, these parameters can be initialized with an input BSSRDF data. For example, the base scattering profiles can be initialized by clustering the local scattering profiles of the input pixels. However, it was determined in the preliminary experiments that such initialization by clustering can easily be stuck in a local minimum of the cost function in Eq. 5. This is because cluster-based initialization tends to inhibit updating of base material assignment by the preceding optimization steps. As a result, this inappropriate initialization can cause approximation errors at material boundaries. This is also the reason why the probabilistic update strategy was used in the PatchMatch-based material assignment. If material assignment is always updated as in the original PatchMatch algorithm, the solution can be stuck in a local minimum of the cost function.

## 4.5 Limitations

Except for the slight increase in the approximation errors, the proposed method with 2-BMPP does not have any critical

limitations. One possible problem is the many optimization parameters in the proposed optimization method. As far as the proposed method was examined with the parameters indicated in the paper, the optimization problems were robustly solved for the 2-BMPP model.

On the other hand, if the user specifically needs to approximate a measured BSSRDF using more than two base materials, the proposed method has several limitations as we discussed in the previous paragraphs. First of all, the computation time for the case of  $K \geq 3$  is impractically long as shown in Table 2. This long computation time is due to the need for many iterations to allow the modeling parameters to converge. In addition, as indicated in Tables 5 and 6, approximation results given with  $K \geq 3$  can be worse than the case of  $K = 2$ . A possible reason for this problem is the slow convergence of the augmented Lagrangian method, which can complicate the appropriate assignment of the base materials. Another possible reason is the large search space for the combinations of the base materials in the cases of  $K \geq 3$ . Suppose that the typical test case of  $M = 128$  and a BSSRDF is defined on  $256 \times 256 = 65,536$  pixel area. When  $K = 2$  is used, the number of possible base material combinations is  ${}_{128}C_2 = 8128 < 65,536$ . Meanwhile, it increases to  ${}_{128}C_3 = 341,376 > 65,536$  when  $K = 3$  is used. Since the proposed combinatorial optimization does not change the combinations assigned by the random initialization, the increase in the base material combinations can complicate the appropriate assignment of the appropriate combinations.

In conclusion, we recommend the use of the 2-BMPP model in most practical cases given its numerical instability, relatively low computation time, and good approximation for many heterogeneous translucent materials.

## 5 Conclusion

In this paper, we proposed a memory-efficient BSSRDF model for the measurement of heterogeneous translucent materials. The high compression ratio of the proposed method is typically preferred for memory-limited devices such as video game consoles and mobile phones. The proposed method compresses physically measured BSSRDFs into significantly smaller data sizes compared to the state-of-the-art compression method by Song et al. [5]. The compression is based on the observation that most translucent materials are comprised of only a limited number of base materials. The compression method represents the local scattering profile of each pixel with a linear combination of the scattering profiles of base materials in the log scale. In the proposed method, we assign multiple indices of the base materials and mixture ratios to each pixel. However, optimizing these parameters is a complicated problem in which constrained nonlinear optimization and NP-hard

combinatorial optimization must be simultaneously solved. To solve this complex problem, we proposed an alternating optimization method using the L-BFGS-B method [28] or an augmented Lagrangian method [27] for the constrained nonlinear optimization and a randomized approach inspired by PatchMatch [6] for the combinatorial optimization. The resulting compressed data are approximately 30 times smaller than the data obtained using the state-of-the-art method. In addition, the visual quality of the rendered images compared to the compressed BSSRDFs is nearly identical to that obtained with the previous method. Several experiments have confirmed that the small data size obtained using the proposed method is preferable in practice for memory-limited devices. Apart from the direct utilization of the proposed compression method in memory-limited devices, its high compression ratio is expected to be more important when high-resolution BSSRDFs are used in practical applications.

**Acknowledgements** The authors would like to thank Pieter Peers, Xin Tong, and Yue Dong for providing their heterogeneous BSSRDF datasets, which made this study possible. The authors would also like to thank the anonymous reviewers for their constructive comments and helpful suggestions which assisted in significantly improving the quality of this paper.

**Funding** Tatsuya Yatagawa was supported by a Research Fellowship for Young Researchers from Japan's Society for the Promotion of Science (Grant-in-Aid for JSPS Fellows JP16J02280). Hideki Todo was supported by JSPS KAKENHI Grant No. JP15H05924. Yasushi Yamaguchi was supported by JSPS KAKENHI Grant No. JP16H02818. Shigeo Morishima was supported by JST ACCEL Grant No. JPM-JAC1602, and a Grant-in-Aid from the Waseda Institute of Advanced Science and Engineering.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Goesele, M., Lensch, H.P.A., Lang, J., Fuchs, C., Seidel, H.P.: Disco: acquisition of translucent objects. *ACM Trans. Graph.* **23**(3), 835 (2004). <https://doi.org/10.1145/1186562.1015807>
2. Peers, P., vom Berge, K., Matusik, W., Ramamoorthi, R., Lawrence, J., Rusinkiewicz, S., Dutré, P.: A compact factored representation of heterogeneous subsurface scattering. *ACM Trans. Graph.* **25**(3), 746 (2006). <https://doi.org/10.1145/1179352.1141950>
3. Wang, J., Zhao, S., Tong, X., Lin, S., Lin, Z., Dong, Y., Guo, B., Shum, H.Y.: Modeling and rendering of heterogeneous translucent



- materials using the diffusion equation. *ACM Trans. Graph.* **27**(1), 9:1 (2008). <https://doi.org/10.1145/1330511.1330520>
4. Nicodemus, F.E., Richmond, J.C., Hsia, J.J., Ginsberg, I.W., Limperis, T.: Geometrical Considerations and Nomenclature for Reflectance. National Bureau of Standards, Gaithersburg (1977)
  5. Song, Y., Tong, X., Pellacini, F., Peers, P.: Subedit: a representation for editing measured heterogeneous subsurface scattering. *ACM Trans. Graph.* **28**(3), 31–1 (2009). <https://doi.org/10.1145/1576246.1531337>
  6. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **28**(3), 24:1 (2009). <https://doi.org/10.1145/1531326.1531330>
  7. Chandrasekhar, S.: Radiative Transfer. Oxford University Press, Oxford (1960)
  8. Jensen, H.W., Marschner, S.R., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01), pp. 511–518 (2001). <https://doi.org/10.1145/383259.383319>
  9. Donner, C., Jensen, H.W.: Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.* **24**(3), 1032 (2005). <https://doi.org/10.1145/1073204.1073308>
  10. d'Eon, E., Irving, G.: A quantized-diffusion model for rendering translucent materials. *ACM Trans. Graph.* **30**(4), 56 (2011). <https://doi.org/10.1145/2010324.1964951>
  11. Frisvad, J.R., Hachisuka, T., Kjeldsen, T.K.: Directional dipole model for subsurface scattering. *ACM Trans. Graph.* **34**(1), 5:1 (2014). <https://doi.org/10.1145/2682629>
  12. Frederickx, R., Dutré, P.: A forward scattering dipole model from a functional integral approximation. *ACM Trans. Graph.* **36**(4), 109:1 (2017). <https://doi.org/10.1145/3072959.3073681>
  13. Song, Y., Wang, W.: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 1–7 (2013). <https://doi.org/10.1109/APSIPA.2013.6694333>
  14. Stam, J.: Multiple Scattering as a Diffusion Process, pp. 41–50. Springer, Vienna (1995). [https://doi.org/10.1007/978-3-7091-9430-0\\_5](https://doi.org/10.1007/978-3-7091-9430-0_5)
  15. Kaplanyan, A., Dachsbacher, C.: Cascaded light propagation volumes for real-time indirect illumination. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 99–107 (2010). <https://doi.org/10.1145/1730804.1730821>
  16. Børlum, J., Christensen, B.B., Kjeldsen, T.K., Mikkelsen, P.T., Noe, K.Ø., Rimstad, J., Mosegaard, J.: In: Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics (ACM, 2011), HPG '11, pp. 7–14. <https://doi.org/10.1145/2018323.2018325>
  17. Koa, M.D., Johan, H.: ESLPV: enhanced subsurface light propagation volumes. *Vis. Comput.* **30**(6), 821 (2014). <https://doi.org/10.1007/s00371-014-0952-3>
  18. Lensch, H.P.A., Goesele, M., Bekaert, P., Kautz, J., Magnor, M.A., Lang, J., Seidel, H.P.: Interactive rendering of translucent objects. In: 10th Pacific Conference on Computer Graphics and Applications, pp. 214–224 (2002). <https://doi.org/10.1109/PCCGA.2002.1167862>
  19. Shah, M.A., Kontinen, J., Pattanaik, S.: Image-space subsurface scattering for interactive rendering of deformable translucent objects. *IEEE Comput. Graph. Appl.* **29**(1), 66 (2009). <https://doi.org/10.1109/MCG.2009.11>
  20. Chen, G., Peers, P., Zhang, J., Tong, X.: Real-time rendering of deformable heterogeneous translucent objects using multiresolution splatting. *Vis. Comput.* **28**(6), 701 (2012). <https://doi.org/10.1007/s00371-012-0704-1>
  21. Jimenez, J., Gutierrez, D.: Screen-space subsurface scattering. In: GPU Pro: Advanced Rendering Techniques, pp. 335–351 (2010). <https://doi.org/10.1201/b10648-26>
  22. Jimenez, J., Zsolnai, K., Jarabo, A., Freude, C., Auzinger, T., Wu, X.C., der Pahlen, J., Wimmer, M., Gutierrez, D.: Separable subsurface scattering. *Comput. Graph. Forum* **34**(6), 188 (2015). <https://doi.org/10.1111/cgf.12529>
  23. Dachsbacher, C., Stamminger, M.: Translucent shadow maps. In: Proceedings of the 14th Eurographics Workshop on Rendering, pp. 197–201 (2003). <https://doi.org/10.2312/EGWR/EGWR03/197-201>
  24. Mertens, T., Kautz, J., Bekaert, P., Seidel, H.P., Van Reeth, F.: Interactive rendering of translucent deformable objects. In: Proceedings of the 14th Eurographics Workshop on Rendering, pp. 130–140 (2003). <https://doi.org/10.2312/EGWR/EGWR03/130-140>
  25. Dal Corso, A., Frisvad, J.R., Mosegaard, J., Bærentzen, J.A.: Interactive directional subsurface scattering and transport of emergent light. *Vis. Comput.* **33**(3), 371 (2017). <https://doi.org/10.1007/s00371-016-1207-2>
  26. Jensen, H.W., Buhler, J.: A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.* **21**(3), 576 (2002). <https://doi.org/10.1145/566654.566619>
  27. Conn, A.R., Gould, N.I., Toint, P.: A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.* **28**(2), 545 (1991). <https://doi.org/10.1137/0728030>
  28. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **16**(5), 1190 (1995). <https://doi.org/10.1137/0916069>
  29. Kotlin Programming Language. <https://kotlinlang.org/>. Accessed 20 Dec 2017
  30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600 (2004). <https://doi.org/10.1109/TIP.2003.819861>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Tatsuya Yatagawa** is a postdoctoral researcher at Waseda University in Japan. He received his bachelor degree from Kyoto University in 2010 and received Ph.D. of Science from the University of Tokyo in 2015. His research interests center on computer graphics and computer vision, including inverse rendering, efficient rendering, and image editing methods. He currently receives the postdoctoral fellowship from Japan Society for the Promotion of Science (JSPS).



**Hideki Todo** is an assistant professor at Chuo Gakuin University. His work experience includes working as a researcher in animation industry at OLM Digital, Inc. from 2011 to 2013. He received his Ph.D. degree in Information Science and Technology from the University of Tokyo in 2013. His research interests lie in computer graphics, image processing, and computer vision, including stylized rendering, image stylization, inverse rendering, and image/video editing.



**Shigeo Morishima** is a professor of Graduate School of Advanced Science and Engineering, Waseda University in Japan. He received the B.S., M.S., and Ph.D. degrees, all in Electrical Engineering from the University of Tokyo, Japan, in 1982, 1984, and 1987, respectively. From 1987 to 2001, he was an associate professor and from 2001 to 2004, a professor of Seikei University. He is a member of the IEEE, ACM SIGGRAPH, and is a trustee of Japanese Academy of Facial Studies. He is a vice

president of the Institute of Image Electronics Engineers of Japan.



**Yasushi Yamaguchi** received his Dr. Eng. degree in information engineering from the University of Tokyo in 1988. He has been a professor at the University of Tokyo since 2002. His research interests lie in image processing, computer graphics, and visual illusion, including image/video editing, computer-aided geometric design, visual cryptography, and hybrid image. He was a former president of Japan Society for Graphic Science and is a president of International Society for

Geometry and Graphics.