# An Efficient Plugin for Representing Heterogeneous Translucent Materials

Sermet Önel
Department of Computer Engineering
Yaşar University
Email: sermet.onel@yasar.edu.tr

Murat Kurt
International Computer Institute
Ege University
Email: murat.kurt@ege.edu.tr

Aydın Öztürk
Department of Computer Engineering
İzmir University
Email: aydin.ozturk@izmir.edu.tr

***Abstract*—This paper presents a plugin that adds an efficient representation of heterogeneous translucent materials to the Blender 3D modeling tool. Algorithm of the plugin is based on Singular Value Decomposition (SVD) method and Mitsuba renderer is the default rendering software used by the proposed plugin. We validate the efficiency of the proposed plugin by using a set of measured heterogeneous subsurface scattering data sets.**

***Keywords*—BSSRDF, Subsurface Scattering Model, Factorization, Heterogeneous Subsurface Scattering, Mitsuba Renderer, Blender 3D Modeling Tool**

## I. Introduction

Efficient representation of heterogeneous translucent materials in computer graphics is a common problem. A number of efficient methods have been proposed to represent the Bidirectional Scattering Surface Reflectance Distribution Function (BSSRDF) for homogeneous translucent materials [1], [2]. However, none of these methods could be generalized to provide proper outputs for heterogeneous translucent materials. The characteristics of having structural deficiencies, impurities and composite structures inside the object volume of heterogeneous translucent materials require approaches with a different view point [3], [4], [5], [6], [7]. On the other hand, high storage needs and computational costs of these algorithms remain to be a major problem to be resolved.

In this study, we use the Singular Value Decomposition (SVD) method for the BSSRDF representation of heterogeneous translucent materials. The proposed approach was implemented in C++ and included in the source codes of Mitsuba renderer project [8]. The integration plugin which has already been available for use by [9], was modified and imported into the three dimensional (3D) Blender modeling tool [10].

Our plugin helps to render heterogeneous translucent materials accurately and efficiently. As it can be seen in Figure 7 and Figure 8, the rendering output of the plugin gives heterogeneous subsurface scattering effects visually plausibly.

## II. Related Work

The problem of representing BSSRDF for heterogeneous translucent materials has an extensive literature. Major efforts have been devoted to the development of some approximation models. The underlying approaches broadly can be classified into two groups. The first group includes the techniques that extend the Jensen's *Dipole Diffusion Approximation Model* [1] and the second group consists of the techniques that are based on development of new material models.

Jensen's Dipole Diffusion Approximation model reduces computation time of eight dimensional BSSRDF [11] to acceptable rates. This approach is effective on homogeneous translucent materials and extended by many researchers [12], [13], [14], [15], [16]. Nevertheless, the main observation of light being isotropic and modeling homogeneous translucent materials make this model inappropriate for heterogeneous BSSRDF representation.

Jakob et al. [12] extended the Dipole Diffusion Approximation model by using anisotropic approach. This model improved the Jensen et al.'s model, however, the output of this model does not give visually plausible heterogeneous subsurface scattering effects. Mertens et al. [13] modeled human skin by using an interactive method to achieve local subsurface scattering. Donner and Jensen's [14] study is based on using multiple dipoles and they represented their study on paper and human skin. Another study was presented by Jimenez et al. [15] in which human skin was represented. Considering the psychological states of human face, Jimenez et al. [16] also modeled facial appearance for different regions on the face.

The second group of techniques includes Goesele et al.'s [5] compact model depending on underlying geometry, Tong et al.'s [6] model of quasi-homogeneous materials and Song et al.'s [7] SubEdit representation which allows interactive editing and rendering of translucent materials. Although these techniques classified in this group have made some improvements on heterogeneous BSSRDF representation, their design issues still prevent them to provide efficient solution for the problem.

The Peers et al.s' work [17] was an important step for the solution of the problem. In their study, they employed the Non-Negative Matrix Factorization (NMF) algorithm. Replacing the Tucker-based factorization on tensor decomposition with the NMF algorithm, the same algorithm was also used by Kurt et al. [3]. Kurt [18] also used Singular Value Decomposition method on tensor decomposition and reported that the underlying algorithm has improved the computational efficiency for the data with two dimensional (2D) matrices [18]. This was the main motivation in our work and following his work, his corresponding algorithm was imported through our plugin and an efficient heterogeneous subsurface scattering representation was put into service for 3D modeling tools.

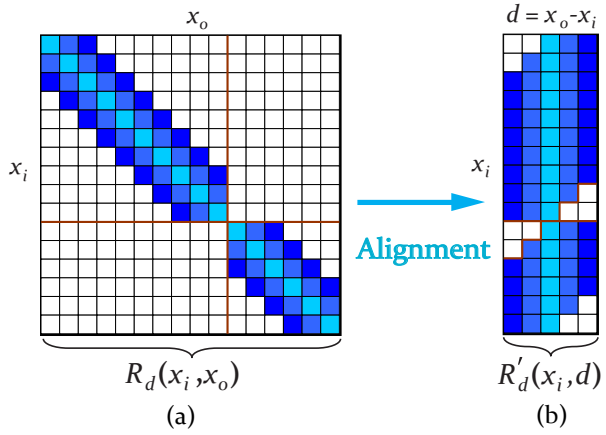



Fig. 1. The diffuse BSSRDF matrix (a) is transformed by first aligning the diagonal by a change of variables to $R'_d(x_i, d)$ (b) [3].

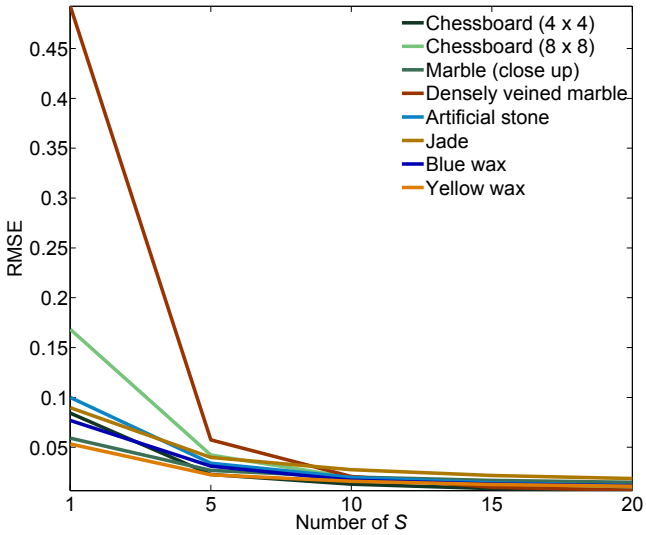



Fig. 2. The root-mean-square error (RMSE) values of SVD-based subsurface scattering model with different values of $S$ parameter [18].

## III. Subsurface Scattering with SVD Method

### A. Preparation of the Test Data

Heterogeneous translucent materials are represented by BSSRDF [11], [1]:

$$L_o(x_o, \vec{\omega}_o) = \int_A \int_{\Omega^+} L_i(x_i, \vec{\omega}_i) S(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o)(\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i dx_i. \quad (1)$$

This function relates to the outgoing radiance at one point to the incident flux at another. Eq. (1) can be separated into a local and a global component where the local component represents the reflected light and the global component represents the scattering light in the material volume [3]. The global component is represented by the diffuse BSSRDF [17], [3]

$$S_d(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o) = \frac{1}{\pi} F_i(x_i, \vec{\omega}_i) R_d(x_i, x_o) F_o(x_o, \vec{\omega}_o). \quad (2)$$

Eq. (2) represents the diffuse BSSRDF $S_d$ with a four dimensional (4D) spatial subsurface scattering component $R_d$ and the directionally dependent components $F_i$ and $F_o$. The directionally dependent components are ignored and $R_d$ is focused in the modeling process [17], [3], [5], [7].

For the factorization operation, 4D $R_d$ value is transformed into 2D matrix. As it can be seen in Figure 1, Kurt et al. [3] reorganized this matrix by changing the variable $d = x_o - x_i$. With this reorganization, $R'_d$ matrix is found which is a more compact matrix for representing measured heterogeneous translucent materials.

### B. Factorization

The subsurface scattering data is factorized by Singular Value Decomposition (SVD) as explained below. This method is a subset of tensor decomposition methods used in the representation of subsurface scattering effects. The study of Peers et al. [17] based on NMF algorithm and the study of Kurt et al. [3] based on Tucker-based factorization are examples of other tensor decomposition methods.

The subsurface scattering data has multi-dimensional features. It can be represented through a tensor model. For example, a 2D matrix can be considered as a second degree tensor. Thus, the subsurface scattering data in the form of a matrix provides a convenient form of data set for tensor decomposition operation.

In SVD operation an $M \times N$ matrix is defined as the product of an $U$ matrix with dimensions $M \times K$ and a matrix $V$ with dimensions $K \times N$ and a core tensor with dimensions $K \times K$. This is a similar decomposition operation in Kurt et al.'s study [3], however, if the value of $K$ is chosen to be 1 that is a scalar then the dimensions of $U$ and $V$ matrices become $M \times 1$ and $1 \times N$, respectively, and the core tensor becomes a scalar [18].

In this operation, $R'_d$ is taken into consideration as it is the most compact data. Another consideration in the factorization operation is making the data stay in the positive values which leads to physically correct results. This is achieved by another transformation that is [18]:

$$R'''_d(x_i, d) = \ln\left(\frac{R'_d(x_i, d)}{A} + B\right). \quad (3)$$

By choosing the most appropriate values for $A$ and $B$ to minimize the error values, $R'''_d$ matrix is factorized using SVD and error terms for each color channel is modeled using Bilgili et al.'s approach [19]. This procedure is repeated $S$ times to improve the accuracy of the approximation. Accordingly, $R'''_d$ becomes [18]:

$$R'''_d(x_i, d) \approx \sum_{j=1}^{S} f_j(x_i) h_j(d). \quad (4)$$

More details about SVD-based subsurface scattering representation can be found in Kurt's [18] work.

### C. Analysis

SVD-based subsurface scattering representation depends only on a single parameter, $S$ that is the number of terms in the factorization operation.

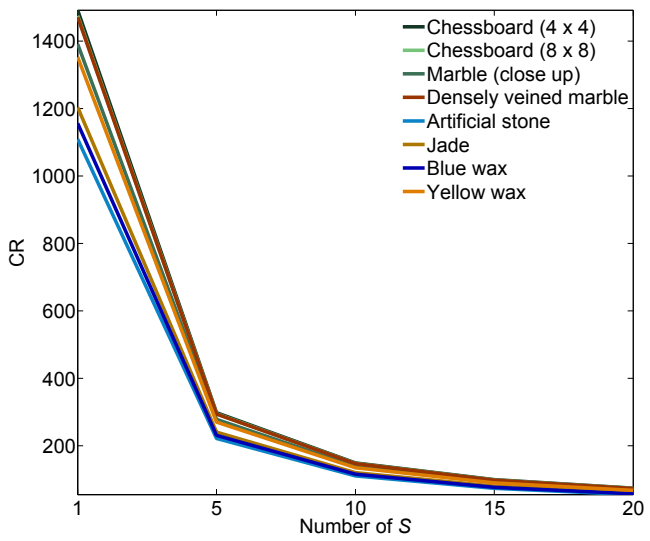



Fig. 3. The compression rates (CR) of SVD-based subsurface scattering model with different values of $S$ parameter [18].

The work of Kurt [18] shows that SVD-based subsurface scattering representation gives convincing results for representing heterogeneous translucent materials. In this work, it is also emphasized that the approximation results are visually acceptable for some test materials even for small number of iteration ($S \leq 5$). The effect of the number of iterations on the model errors for different materials is illustrated in Figure 2. This efficiency was the main motivation in this study, which leaded us to use SVD-based subsurface scattering representation in our plugin instead of other factorization methods such as Tucker-based factorization model [3].

Furthermore, the $S$ value should be chosen carefully since the factorization is applied for each channel, separately and the compression rate can be a problem for obtaining close approximations. The effect of the number of iterations on the compression ratio for different materials is illustrated in Figure 3.

## IV. The Integration Plugin

Constructing images through modeling and representing the BSSRDF for heterogeneous translucent materials is a complicated procedure. Developing a plugin to support the heterogeneous subsurface scattering effects of the underlying materials will provide a useful tool for many applications. In this work, we aim to develop a plugin for rendering purposes. Such a plugin must be an interface between the renderer and the software modeling tool. Because, the renderer which has the capability to represent the heterogeneous translucent materials is not the only task to be carried out. Details of lighting information, shading, texture mapping and others in the scene are defined by the 3D modeling tool and these details are also important properties that should be considered in the representation. Therefore, the plugin should send the details available on the scene to the renderer, and then the SVD-based subsurface scattering representation can be a solution for the common problem.

Our procedure for developing the plugin has two steps: the first step consists of the implementation of SVD-based subsurface scattering representation in the renderer, and the second step is to create a script that is able to send the details in the scene to the SVD-based subsurface scattering representation for a visually plausible rendering.

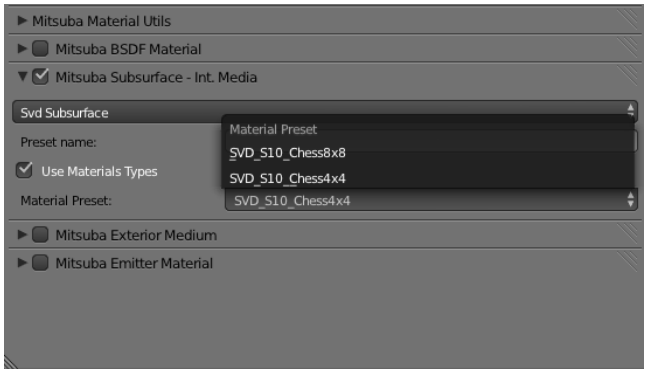



Fig. 4. The graphical user interface of our integration plugin in the Blender 3D Modeling Tool [10].

The 3D modeling tools are powerful software to model the details of the scene. Light sources, materials and their details can easily be defined using these tools. Our main choice was the Blender project [10] which is an open source developing platform and uses Python [20] as its plugin scripting language. These features were effective in our choice since Python has the ability to call C++ functions without the need of any software. It is also effective in the sense that an open source application would be useful for the documentation and faster development.

We chose Mitsuba renderer [8] as the default renderer because it is highly optimized and it has good performance, scalability, easy usability and robustness are the other features of this renderer includes. As the project is in C++ and Blender supports Python, the implementation of our plugin could become easier. The version of the Blender used in the this work is 2.69 which is compatible with Mitsuba version 0.5. We modified the source codes of these versions to finalize our study successfully.

The integrator class in Python script is defined as ”class mitsuba_sss_svd”. This class has control parameters for using different material types which are the possible parameters for the heterogeneous translucent materials. We tested our plugin on chessboard ($4 \times 4$) and chessboard ($8 \times 8$) heterogeneous translucent materials, which were measured by Peers et al. [17].

The integration plugin renders the material according to the material type choice and sends the data to the Mitsuba renderer .exe file, which consists of the necessary operations. These parameters are updated a function called ”params.update” which is defined in the Python script. The data is kept on the object of the material chosen, by using the default constructor self of Python language.

Graphical User Interface (GUI) of the integration plugin is shown in Figure 4. The representation of heterogeneous translucent materials is classified under the section of Subsurface Scattering of the already available integration plugin [9] and the details of the material type is chosen using the GUI of

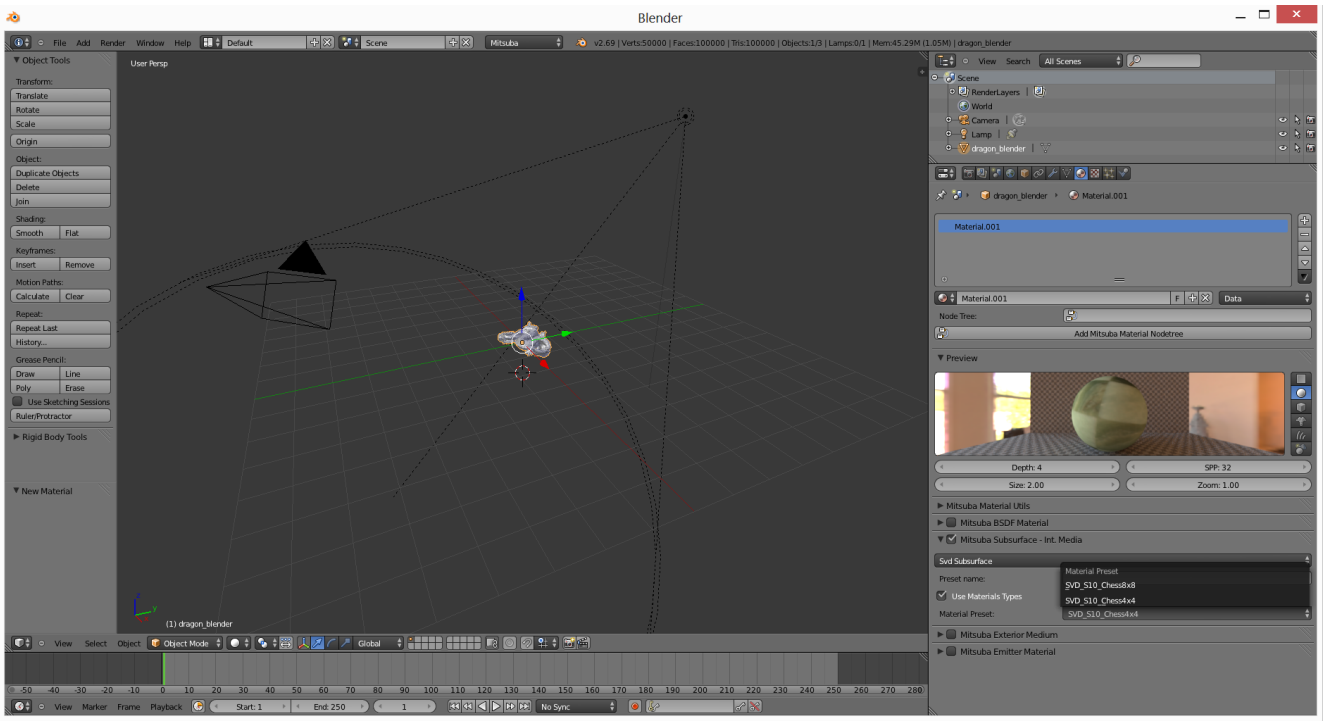



Fig. 5. A general overview of the Blender 3D Modeling Tool [10] and our integration plugin.

the plugin. As the Blender project [10] supports for modeling different types of materials and other details in the scene, the effect of the chosen operation is completed after the rendering operation. A general overview of the Blender 3D modeling tool and our integration plugin can be seen in Figure 5.

## V. Experimental Results

The proposed plugin was developed and tested by using various objects of some sample materials on which SVD-based heterogeneous subsurface representation were applied. Empirical results has shown that the rendering operation based on the underlying plugin simulates the heterogeneous subsurface scattering effects successfully. Furthermore, two critical issues that are the processing time the plugin requires to complete the rendering process and the storage requirements were checked to see if there is any preclusion.

The materials chosen for the test were applied on a dragon and a kitten objects. We checked whether the plugin could perform subsurface scattering effects on the objects as it does directly with Mitsuba renderer. It's important that SVD-based subsurface scattering representation is a texture-space based model. The texture coordinates on the objects were carefully determined by a 3D modeling tool and these texture coordinates were sent to Mitsuba renderer. Thus the details of the scene was exported to the renderer successfully.

Figure 6 shows the preview of subsurface scattering effects on a sphere solid object. A similar effect can be rendered on an object chosen in the scene by the artist. As it is seen in Figure 7(a), our plugin helps to render heterogeneous translucent materials correctly and it took 20 minutes to take this rendering output on a computer with i7-3630QM processor with 8GB RAM and NVIDIA GTX660M/2GB GDDR5. The storage of the output was 36.1MB. Figure 7(b) illustrates another tested material, chessboard $(8 \times 8)$ on a dragon object. The output needed 38.42MB on the same computer and the rendering process took only 20.9 minutes.

As it is seen in Figure 8, the last rendering test was done on a kitten object and on the same PC. Chessboard $(4 \times 4)$ and chessboard $(8 \times 8)$ materials were tested on the kitten object. It took 16.157 and 17.36 minutes to render chessboard $(4 \times 4)$ and chessboard $(8 \times 8)$ materials, respectively. The storage space needed 32.66MB and 34.72MB for rendering chessboard $(4 \times 4)$ and chessboard $(8 \times 8)$ materials, respectively.

## VI. Conclusion

In this paper, we presented an integration plugin for rendering heterogeneous translucent materials and empirically demonstrated that it enables to render visually plausible scenes. Our plugin relies on SVD-based subsurface scattering model which was proposed by Kurt [18]. In this paper, we also showed that the plugin works correctly by communicating with Mitsuba renderer and 3D modeling tool.

The plugin may be developed by adding different subsurface scattering models for rendering heterogeneous translucent materials.

## VII. Future Works

This study enables the use of SVD-based subsurface scattering model for rendering heterogeneous translucent materials on a 3D modeling tool. As the proposed plugin is compatible with Blender project [10] and Mitsuba renderer [8], the functionality is strictly related with the correct versions of these tools.

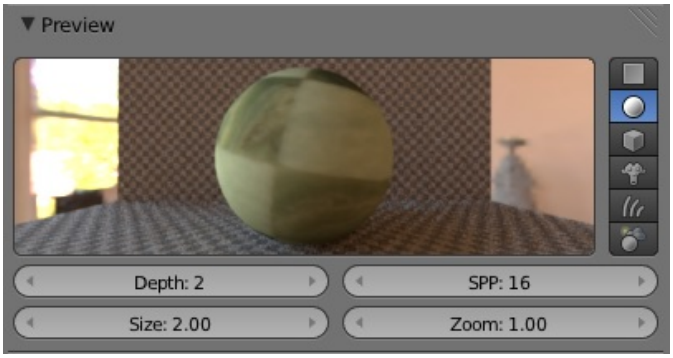
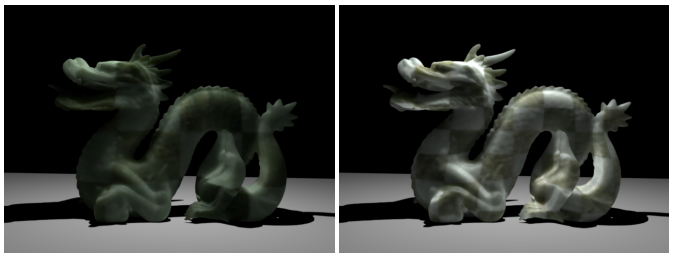



Fig. 6. A preview scene of our integration plugin in the Blender 3D Modeling Tool. Chessboard ($4 \times 4$) material was represented in this rendering.



(a) (b)

Fig. 7. A dragon under spot lighting was rendered using our integration plugin. The value of $S$ parameter in SVD-based subsurface scattering model was selected as 10. (a) chessboard ($4 \times 4$), (b) chessboard ($8 \times 8$) materials.

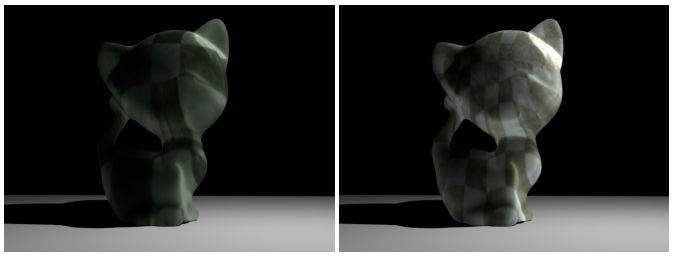
As a future development, the availability of other factorization based models such as Tucker-based factorization model can be supported in the integration plugin. There is also a lack of the availability of different subsurface scattering data which will be supported in the future versions.

It is also important that there are lots of different tools for 3D modeling which brings the need for extending this plugin for different platforms. The compatibility of plugin for other platforms is also considered as a future study.

## ACKNOWLEDGEMENTS

This work was supported by the Scientific and Technical Research Council of Turkey (Project No: 111E208). We would like to thank Pieter Peers, Yue Dong and Xin Tong for sharing their measured heterogeneous subsurface scattering data sets.

We would also like to thank the developer of the Mitsuba renderer project Wenzel Jakob and the developers of the exporter plugin Fransesc Juhé and Bartosz Styperek for their support in the development process.



(a) (b)

Fig. 8. A kitten under spot lighting was rendered using our integration plugin. The value of $S$ parameter in SVD-based subsurface scattering model was selected as 10. (a) chessboard ($4 \times 4$), (b) chessboard ($8 \times 8$) materials.

## REFERENCES

[1] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," in *Proc. SIGGRAPH '01*, 2001, pp. 511–518.

[2] H. W. Jensen, "Global illumination using photon maps," in *Proceedings of the Eurographics Workshop on Rendering Techniques '96*. London, UK, UK: Springer-Verlag, 1996, pp. 21–30.

[3] M. Kurt, A. Öztürk, and P. Peers, "A compact tucker-based factorization model for heterogeneous subsurface scattering," in *Proceedings of the 11th Theory and Practice of Computer Graphics*, ser. TPCG '13. Bath, United Kingdom: Eurographics Association, 2013, pp. 85–92.

[4] E. d'Eon and G. Irving, "A quantized-diffusion model for rendering translucent materials," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 56:1–56:14, Jul. 2011, (Proc. SIGGRAPH '11).

[5] M. Goesele, H. P. A. Lensch, J. Lang, C. Fuchs, and H.-P. Seidel, "DISCO: acquisition of translucent objects," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 835–844, Aug. 2004, (Proc. SIGGRAPH '04).

[6] X. Tong, J. Wang, S. Lin, B. Guo, and H.-Y. Shum, "Modeling and rendering of quasi-homogeneous materials," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 1054–1061, Jul. 2005, (Proc. SIGGRAPH '05).

[7] Y. Song, X. Tong, F. Pellacini, and P. Peers, "Subedit: a representation for editing measured heterogeneous subsurface scattering," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 31:1–31:10, Jul. 2009, (Proc. SIGGRAPH '09).

[8] W. Jakob, "Mitsuba renderer," 2013, http://www.mitsuba-renderer.org.

[9] B. Styperek and F. Juhé, "The blender plugin," 2011, https://www.mitsuba-renderer.org/plugins.html.

[10] Blender, "Blender foundation," 2003, http://www.blender.org/.

[11] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, "Geometrical considerations and nomenclature for reflectance," National Bureau of Standards (US), Monograph, Oct. 1977.

[12] W. Jakob, A. Arbree, J. T. Moon, K. Bala, and S. Marschner, "A radiative transfer framework for rendering materials with anisotropic structure," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 53:1–53:13, Jul. 2010, (Proc. SIGGRAPH '10).

[13] T. Mertens, J. Kautz, P. Bekaert, F. V. Reeth, and H.-P. Seidel, "Efficient rendering of local subsurface scattering," *Computer Graphics Forum*, vol. 24, no. 1, pp. 41–49, 2005.

[14] C. Donner and H. W. Jensen, "Light diffusion in multi-layered translucent materials," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 1032–1039, Jul. 2005, (Proc. SIGGRAPH '05).

[15] J. Jimenez, D. Whelan, V. Sundstedt, and D. Gutierrez, "Real-time realistic skin translucency," *IEEE Computer Graphics and Applications*, vol. 30, no. 4, pp. 32–41, 2010.

[16] J. Jimenez, T. Scully, N. Barbosa, C. Donner, X. Alvarez, T. Vieira, P. Matts, V. Orvalho, D. Gutierrez, and T. Weyrich, "A practical appearance model for dynamic facial color," *ACM Transactions on Graphics*, vol. 29, no. 6, pp. 141:1–141:10, Dec. 2010, (Proc. SIGGRAPH Asia '10).

[17] P. Peers, K. vom Berge, W. Matusik, R. Ramamoorthi, J. Lawrence, S. Rusinkiewicz, and P. Dutré, "A compact factored representation of heterogeneous subsurface scattering," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 746–753, Jul. 2006, (Proc. SIGGRAPH '06).

[18] M. Kurt, "An efficient model for subsurface scattering in translucent materials," Ph.D. dissertation, International Computer Institute, Ege University, Izmir, Turkey, January 2014, 122 pages.

[19] A. Bilgili, A. Öztürk, and M. Kurt, "A general brdf representation based on tensor decomposition," *Computer Graphics Forum*, vol. 30, no. 8, pp. 2427–2439, December 2011.

[20] G. V. Rossum, "Python," 1989, https://www.python.org/.