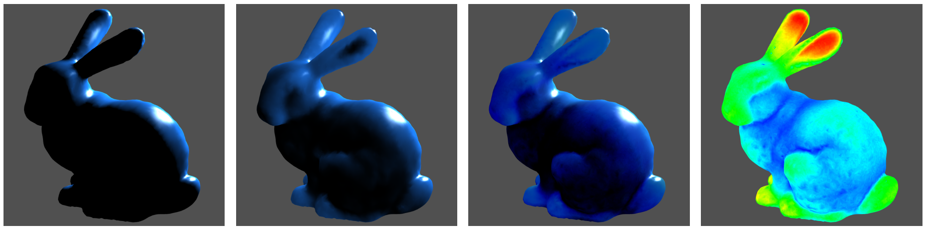
Real-time Rendering of Subsurface Scattering according to Translucency Magnitude

Hiroyuki Kubo* NAIST
Kohe Tokoi Wakayama University
Yasuhiro Mukaigawa NAIST



(a) Lambert (b) Curvature-based (c) Our method (d)Trans. Magnitude

Figure 1: *Examples of the bunny made of blue jewel*

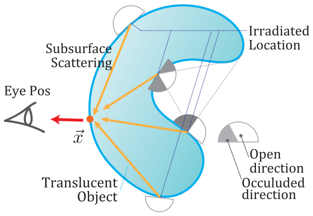


Figure 2: *Light transport*

1 Introduction

To synthesize realistic translucent materials in computer graphics, it is necessary to simulate the effect of subsurface scattering. In previous works, several methods are proposed for rendering such materials in real-time. The screen space subsurface scattering (SSSS) is developed by Jimenez et al. [2009], yet the speed of rendering is not very practical for low-end computational environment, because screen space techniques require huge number of texture samplings. We previously propose a curvature-based shading method [Kubo et al. 2010] which approximates the effect of subsurface scattering according to the curvature. Since the curvature is determined by the surface shape of neighbors, it is not able to compute the effect of scattering light from the behind of the object. In this paper, we propose a novel shading method depending on the translucency magnitude which represents the significance of the subsurface scattering effect. According to the translucency magnitude, we modulate the reflectance to imitate the effect of subsurface scattering. Since this modulation is very simple to compute, we are able to render translucent materials in real-time not only in high-end workstations but also low-end mobile devices.

2 Translucency Magnitude based Shading

In this paper, we propose the translucency magnitude which represents the significance of the subsurface scattering effects. The translucency magnitude is determined at every location $\vec{x}$ on the surface of the input 3D mesh. If a micro region dS, nearby $\vec{x}$, is highly irradiated, the effect of subsurface scattering at the position $\vec{x}$ tends to be more noticeable, although the irradiated light is extinct along the distance $|\vec{x}-\vec{s}|$. The irradiance of each location $\vec{s}$ depends on the light condition, however, we approximate the irradiance intensity as the ambient occlusion independent of the light (Figure2). Thus, we determine the translucency magnitude $t_m(\vec{x})$ as

$$t_m(\vec{x}) = \int_S R(|\vec{x}-\vec{s}|)(1-A_o(\vec{s}))dS. \tag{1}$$

$R(\cdot)$ and $A_o(\vec{s})$ are the individual scattering profile of the translucent material, and the ambient occlusion at the location $\vec{s}$ on the surface, respectively. Prior to rendering, we pre-compute the translucency magnitude for every vertices. During the run-time, we modulate the diffuse reflectance from the Lambertian to the function below to mimic the subsurface scattering effect according to the value of the translucency magnitude, similar to [Kubo et al. 2010]:

$$f_r(\theta, t_m) = k_d \left\{ \max(\cos\theta, 0) * \frac{1}{\sqrt{2\pi t_m}} \exp\left(-\frac{\theta^2}{2t_m^2}\right) \right\} (\theta).$$

k_d and θ are the diffuse albedo and the incident light angle, respectively. Since $f_r(\theta, t_m)$ is a function of two variances, we prepare 2 dimensional look-up-table prior to rendering. During the rendering, we are possible to acquire the radiance only sampling the look-up-table once for each color channel.

3 Results and Conclusions

Figure1-(a), (b) and (c) are synthesized images of Stanford bunny made of blue jewel rendered by using Lambert shading, Curvature-based shading [Kubo et al. 2010], and our method. Moreover, figure1-(d) is a pseudo color of translucency magnitude. Compared with the opaque appearance of Lambert shading, our method is able to synthesize realistic translucent material. Curvature-based shading method (b) brings unrealistic dark region on the bunny's ear. Instead, our method (c) synthesizes more plausible image than curvature-based shading. Compared with Lambert shading which is one of the simplest shading method, our method requires only one texture sampling operation for each color channel, additionally. Our shading method is implemented in HLSL pixel shader, we are able to render these images over 1,000 fps with Intel®HD Graphics 4400. Furthermore, we also develop a WebGL-based renderer. This enables us to render translucent materials in real-time even on the low-end mobile browsers such as PowerVR 6430. In our future work, we will apply our techniques to deformation objects by computing translucency magnitude on the GPU.

References

JIMENEZ, J., SUNDSTEDT, V., AND GUTIERREZ, D. 2009. Screen-space perceptual rendering of human skin. *ACM Transactions on Applied Perception 6*, 4, 23:1–23:15.

KUBO, H., DOBASHI, Y., AND MORISHIMA, S. 2010. Curvature-dependent reflectance function for rendering translucent materials. In *ACM SIGGRAPH 2010 Talks*, ACM, New York, NY, USA, SIGGRAPH '10, 46:1–46:1.

*e-mail:hkubo@is.naist.jp