Real-time Rendering of Heterogeneous Translucent Objects using Voxel Number Map

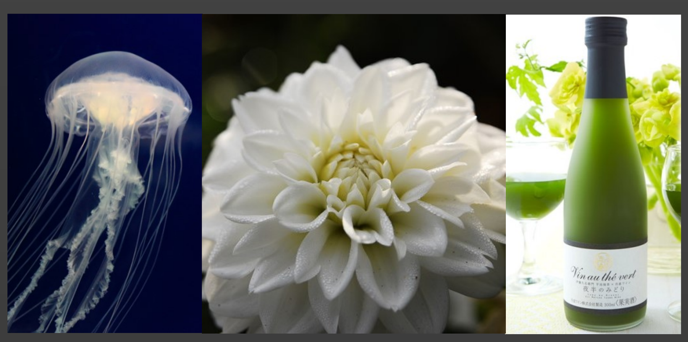
Keisuke Mochida[1], Midori Okamoto[1], Hiroyuki Kubo[2], Shigeo Morishima[3]

[1]Waseda University, [2]Nara Institute of Science and Technology, [3]Waseda Research Institute for Science and Engineering

Introduction

➢ Background

Rendering translucent objects having complicated structure in real-time is a challenging task.

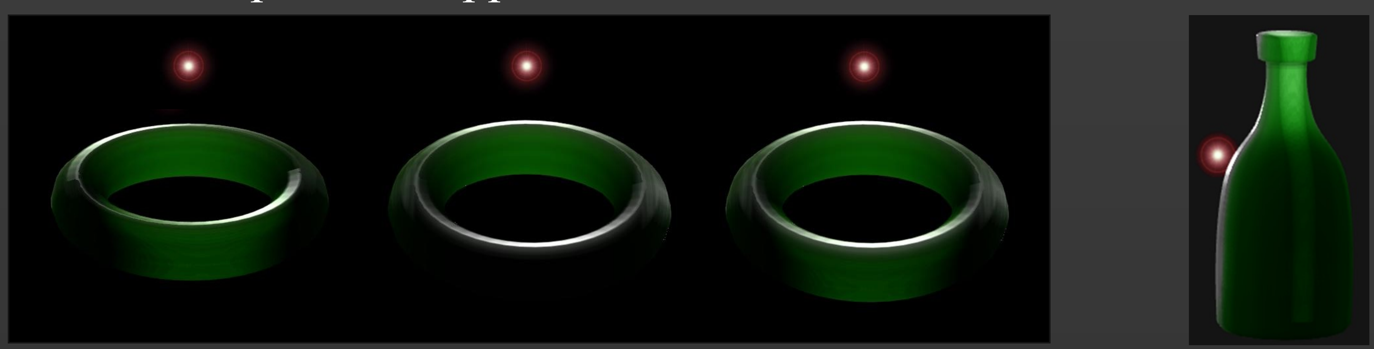


Ex) jellyfish, flower, bottle

Left: https://metimes.jp/images/1551
Center: http://photozou.jp/photo/show/146421/5447807
Right: http://item.rakuten.co.jp/itohkyuemon/10004058/

➢ Previous Work

Shadow Maps-based approaches



Ground Truth | Dachsbacher et al. | Kosaka et al. | Kosaka et al.

Limitation: In case of containing cavities inside, these methods can not generate accurate results.

➢ Purpose

Real-time rendering of heterogeneous translucent objects.

containing cavities / different media inside

Method

Input: Mesh Model

Precomputation Part: Precomputation of Optical Thickness

Run-time Part: Computation of Radiance

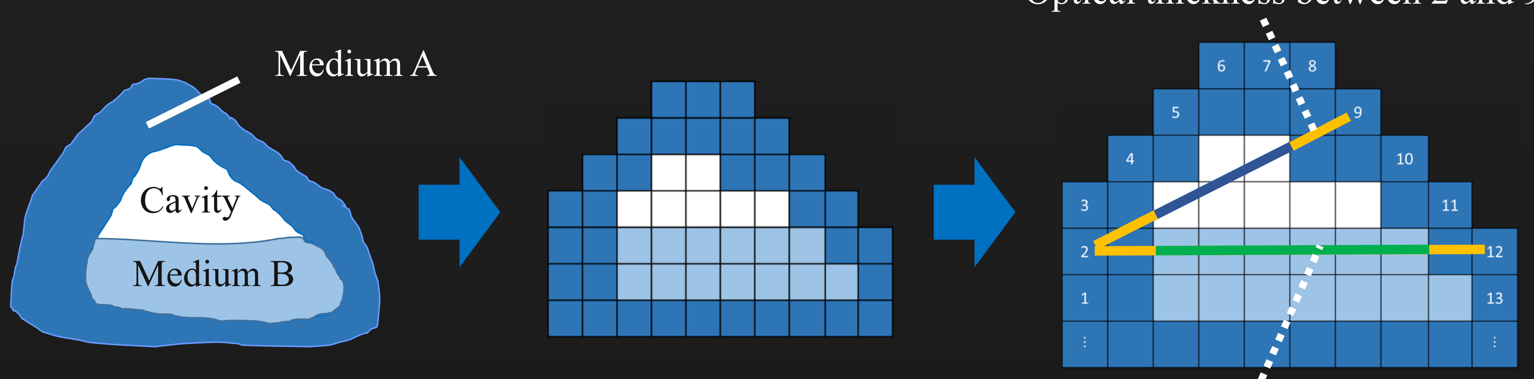
Output: Rendering Results

Optical Thickness $\tau(\boldsymbol{x}_{in}, \boldsymbol{x}_{out})$

- A measusre of how transparent media are to radiation passing through inside.

➢ Precomputation Part

Precomputation of Optical Thickness τ



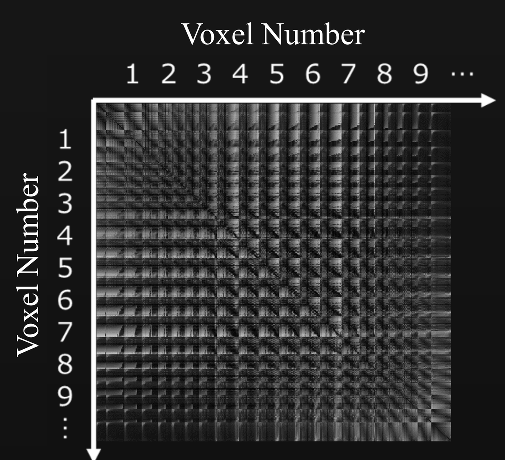
Mesh Data | Voxel Data • Discretization | Optical thickness between 2 and 12. • Precomputation

1. Extracting & Numbering each surface voxel.
2. Computing the optical thickness between two surface voxels.
3. Storing optical thickness into every pixel of Look Up Table (LUT).

Optical thickness $\tau(\boldsymbol{x}_{in}, \boldsymbol{x}_{out})$ is :

$$\tau(\boldsymbol{x}_{in}, \boldsymbol{x}_{out}) = \int_{\boldsymbol{x}_{in}}^{\boldsymbol{x}_{out}} \sigma_t(\boldsymbol{x}')d\boldsymbol{x}'$$

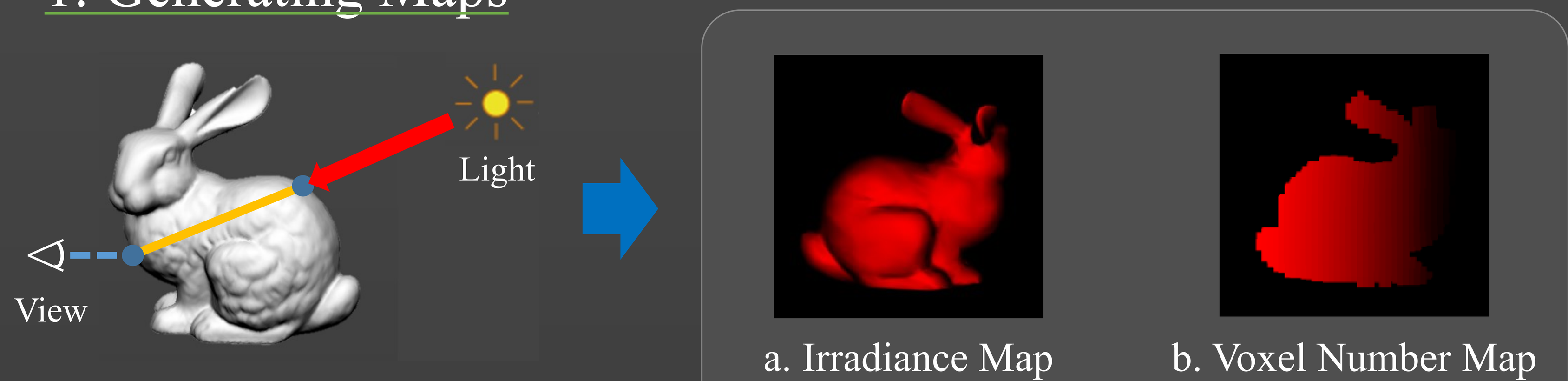
$\sigma_t(\boldsymbol{x}')$: *extinction coefficiet*



Generated LUT

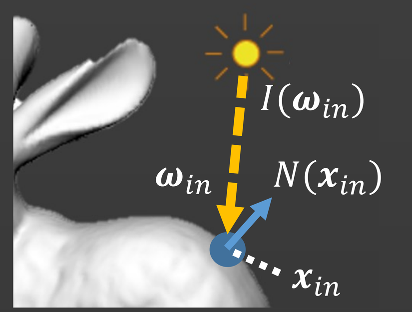
➢ Run-time Part

1. Generating Maps



a. Irradiance Map | b. Voxel Number Map (VNM)

a. Computing and storing irradiance $E(\boldsymbol{x}_{in})$.



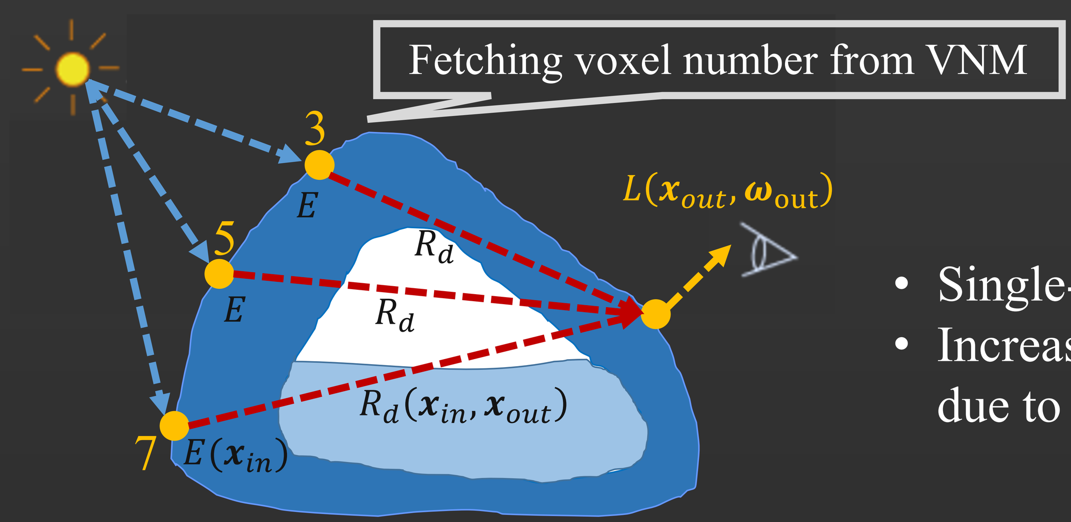
$$E(\boldsymbol{x}_{in}) = F_t(\eta, \boldsymbol{\omega}_{in})|N(\boldsymbol{x}_{in}) \cdot \boldsymbol{\omega}_{in}|I(\boldsymbol{\omega}_{in})$$

F_t: *Fresnel transmittance*

η: *Relative index of refraction*

b. Storing the number of the surface voxel.

2. Computation of Radiance



Assumption

- Single-scattering effects are dominant.
- Increase in the radiance due to in-scattering effects hardly occurs.

$$L(\boldsymbol{x}_{out}, \boldsymbol{\omega}_{\mathrm{out}}) = \frac{1}{\pi}\mathrm{F_t}(\eta, \boldsymbol{\omega}_{\mathrm{out}}) \int_S \underbrace{E(\boldsymbol{x}_{in})}_{\text{Fetched from Irradiance map}} R_d(\boldsymbol{x}_{in}, \boldsymbol{x}_{out})\, d\boldsymbol{x}_{in}$$

$$R_d(\boldsymbol{x}_{in}, \boldsymbol{x}_{out}) = e^{\underbrace{-\tau(\boldsymbol{x}_{in}, \boldsymbol{x}_{out})}_{\text{Fetched from LUT}}}$$

Results

Translucent cube containing a spherical cavity inside (2,020vertex)

Mesh Model	Our Result	TSM Result
	FPS: 405	FPS: 430

Translucent bottle containing wine inside (10,604vertex)

Mesh Model	Our Result	TSM Result
	FPS: 238 (Using 2 LUTs)	FPS: 420

Future Work

We consider effects of

- reflection and refraction at interfaces between the media inside objects using the precomputed refraction map.
- increase in the radiance due to in-scattering when generating LUT.

Demo

E-mail: k.mochida5420@gmail.com