
Zootopia. (c 2010, 2016 Disney.)


Coco (2017 Pixar)


Terminator (c 2015 Moving Picture Company)

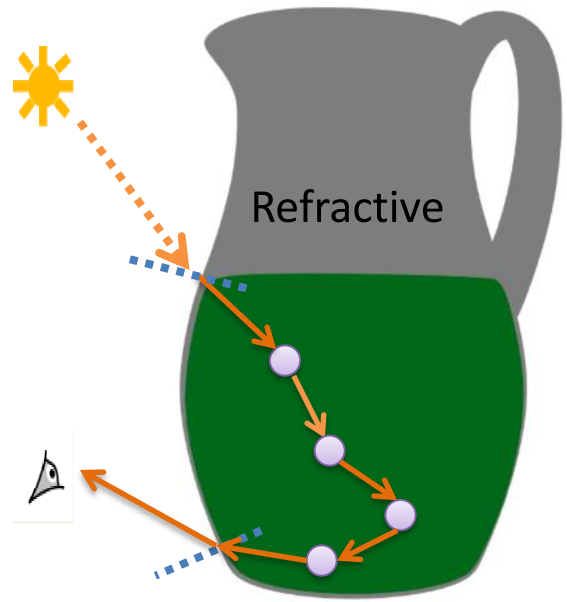
高效参与介质绘制方法研究

王贝贝

高维信息智能感知与系统
教育部重点实验室

南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

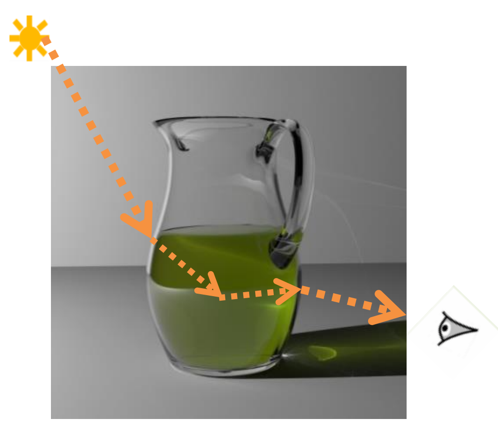
Participating Medium with refractive boundaries



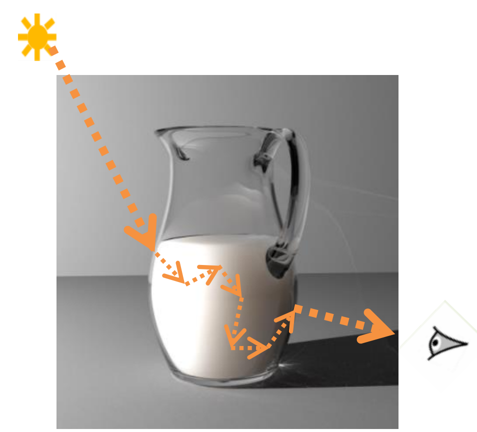
Describe a medium

Homogeneous Medium:

- **Absorption** coefficient
- **Scattering** coefficient
- **Phase function** (describes the angular distribution, might be **highly anisotropic)**

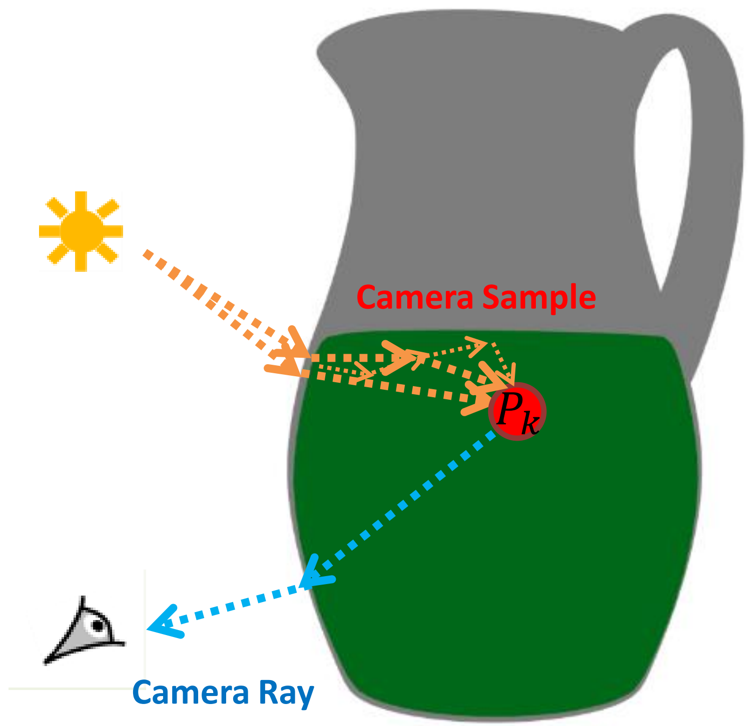


Low order scattering dominant



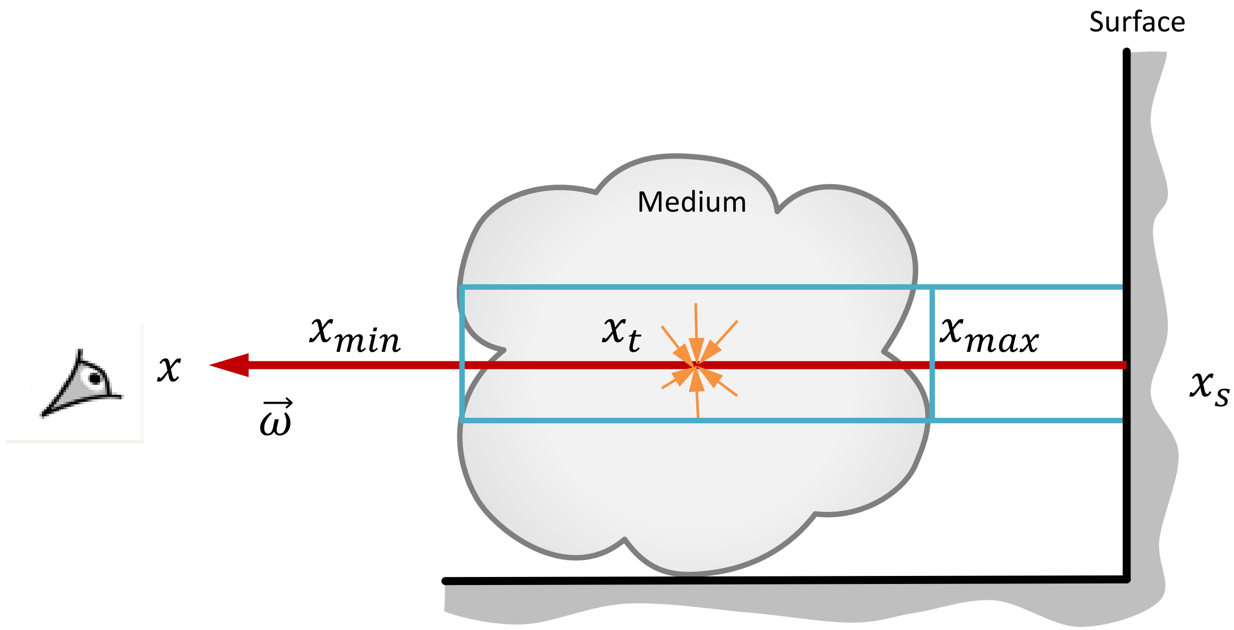
High order scattering dominant

Single / Double / Multiple Scattering



Single Scattering
Double Scattering
Multiple Scattering

Radiative Transfer Equation



$$L(x,\vec{\omega}) = \boxed{\int_{x_{min}}^{x_{max}}} T_r(x \leftrightarrow x_t)\sigma_s(x_t)\boxed{L_i(x_t,\vec{\omega})}dt + \boxed{T_r(x_{max} \leftrightarrow x_{min})L(x_s,\vec{\omega})}$$

$$T_r(x \leftrightarrow x_t) = e^{-\sigma_t \|x - x_t\|},\ \sigma_t = \sigma_a + \sigma_s$$

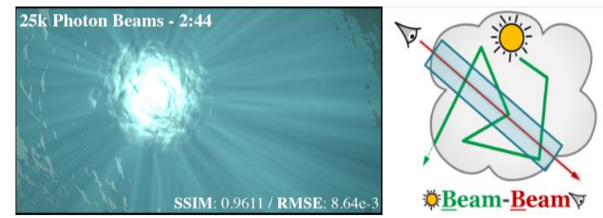
$$\boxed{L_i(x_t,\vec{\omega}) = \int_{\Omega_{4\pi}} p(x_t,\vec{\omega},\overrightarrow{\omega_t})L(x_t,\overrightarrow{\omega_t})d\omega_t}$$

Related Work

- Photon density estimation

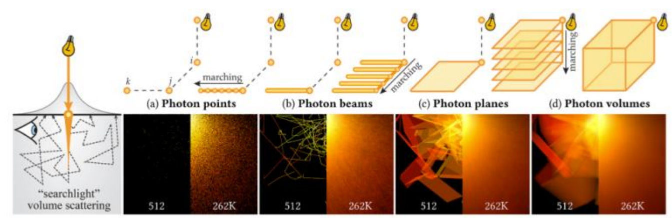
Volumetric photon mapping
[Jensen and Christensen ’98]

Beam radiance estimate
[Jarosz et al. ‘08]



Photon beams
[Jarosz et al. ‘11]

Unifying points, beams, and paths
[Krivanek et al. ‘14]



Beyond Points and Beams
[Benedikt Bitterli, Wojciech Jarosz ‘ 17]

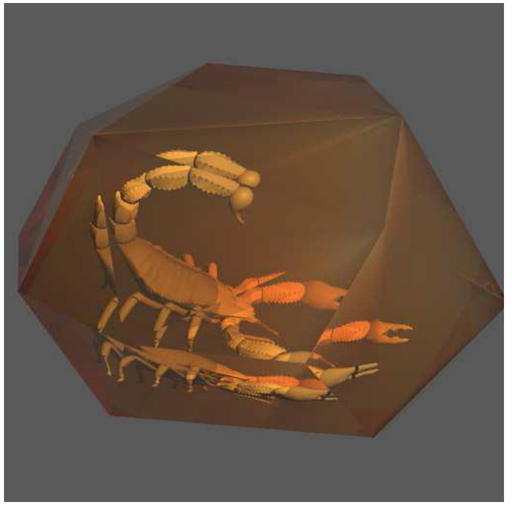
Related Work

- Multiple scattering

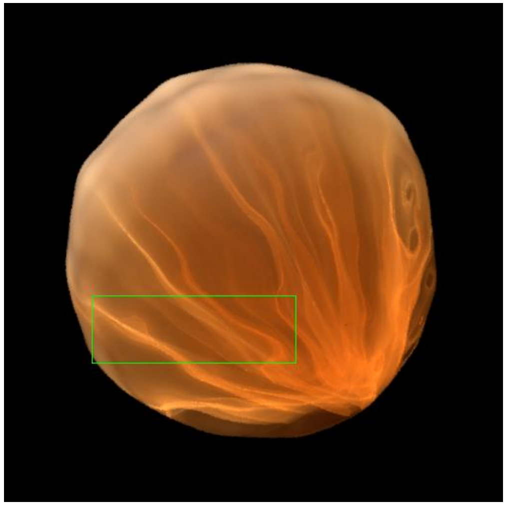
Virtual Ray Lights
[Novák et al 2012]

Related Work

- Single scattering

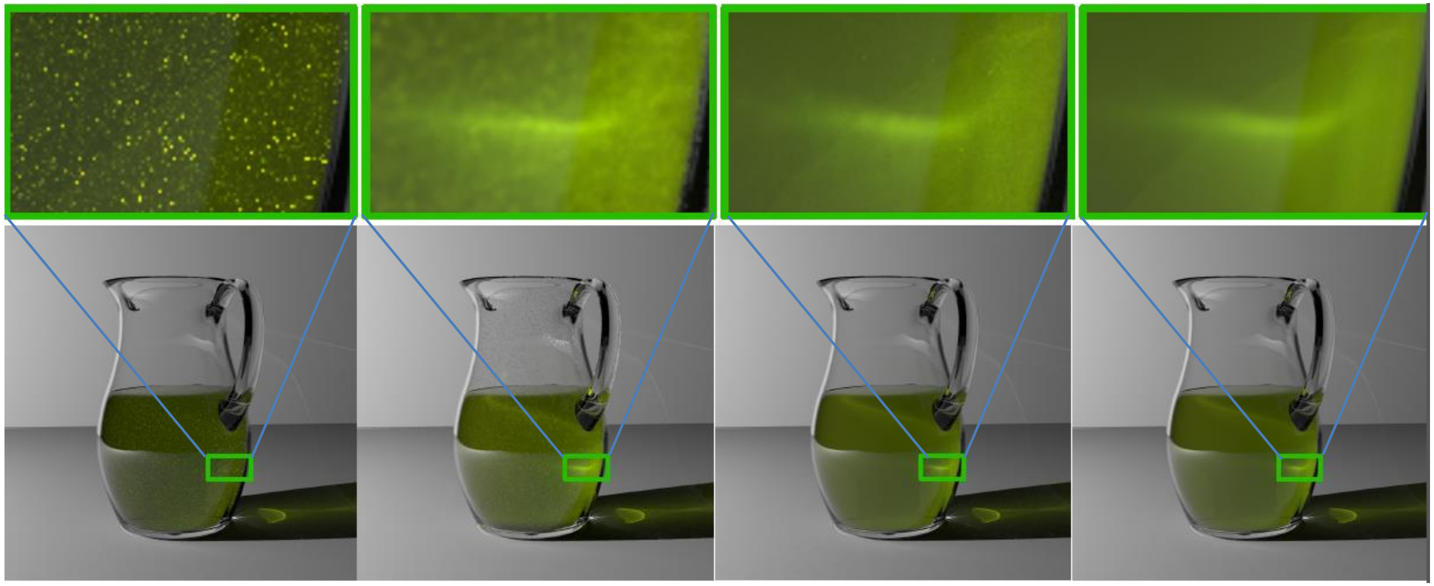


[Bruce Walter et al 09]



[Holzschuch 2015]

Motivation



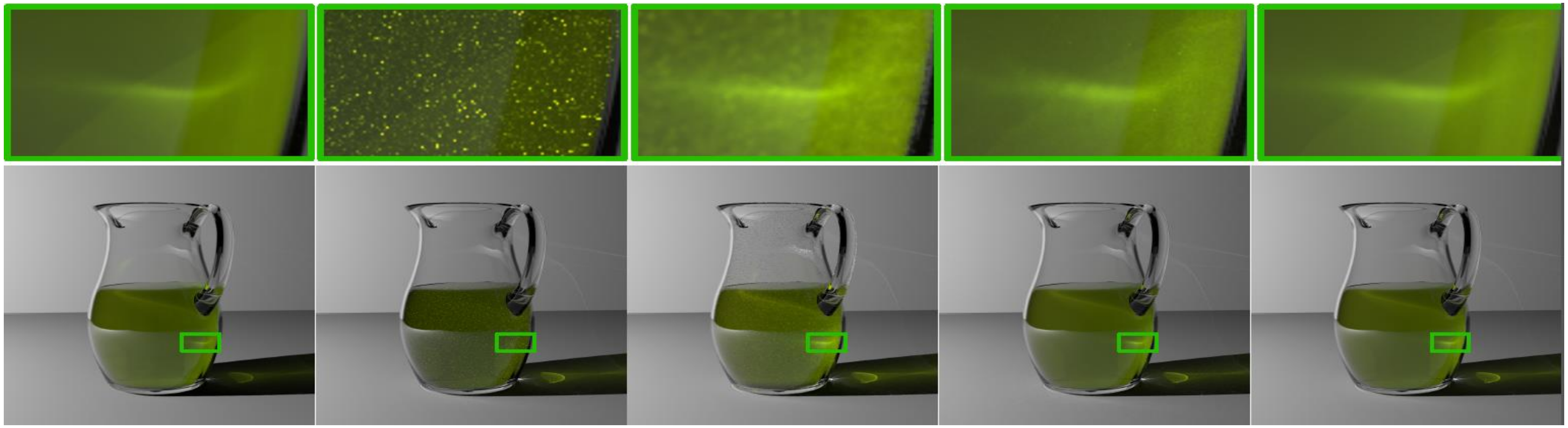
BDPT, 41 mins | PM with BRE, 41 mins | UPBP, 41 mins | UPBP, 6 hours

long time to converge/ noisy results

Point-Based Light Transport for Participating Media with Refractive Boundaries

Beibei Wang, Jean-Dominique Gascuel, Nicolas Holzschuch

EGSR 2016 (EI&I)

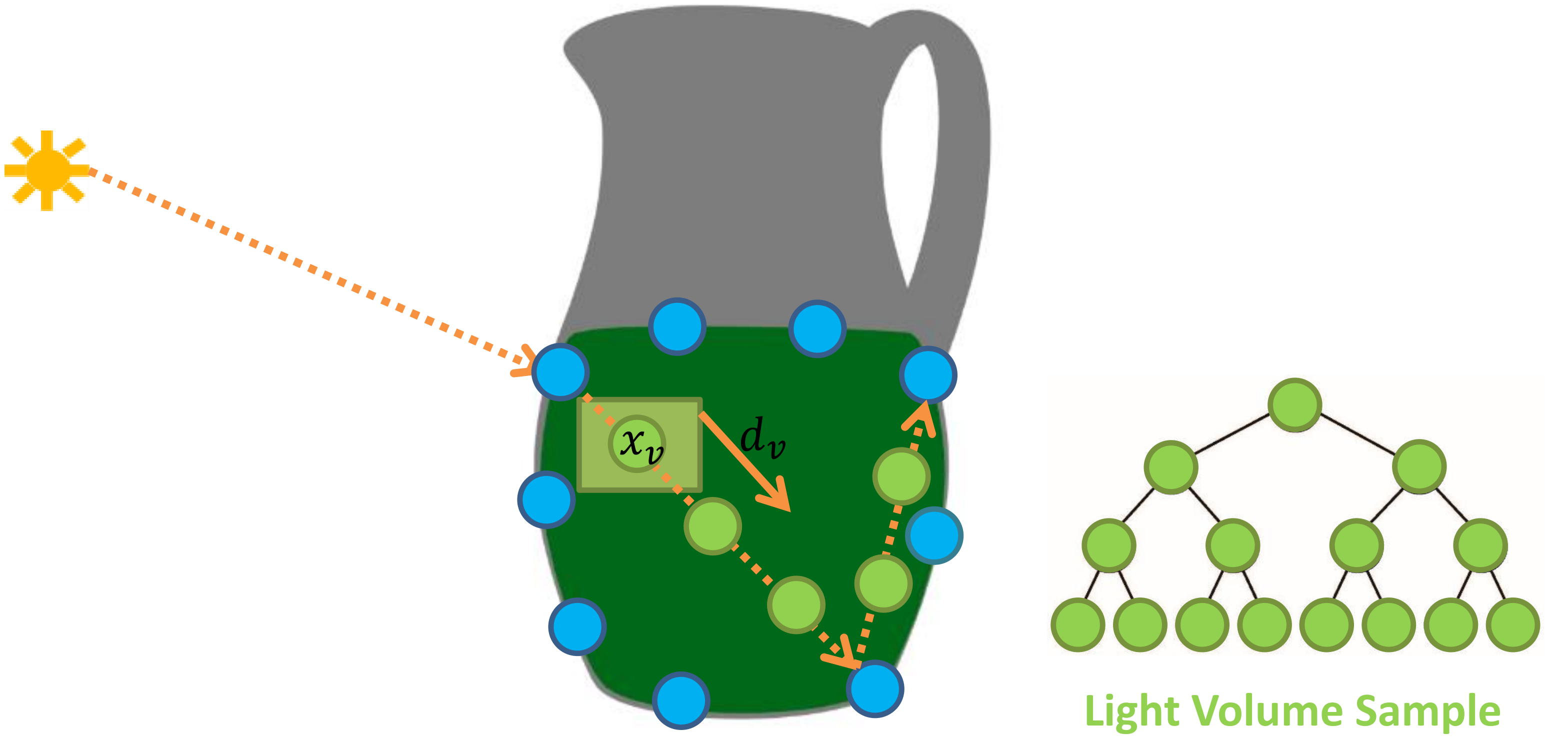


Our, 41 mins	BDPT, 41 mins	PM with BRE, 41 mins	UPBP, 41 mins	UPBP, 6 hours

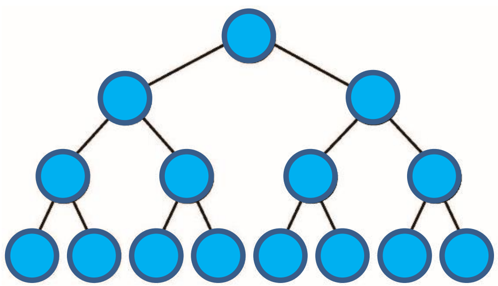
Our approach

- Inspired by PBGI, from **surface samples to volume samples**
- Efficient noise-free solution for:
  - single scattering: local query
  - double scattering: global traversal
  - multiple scattering: precomputation

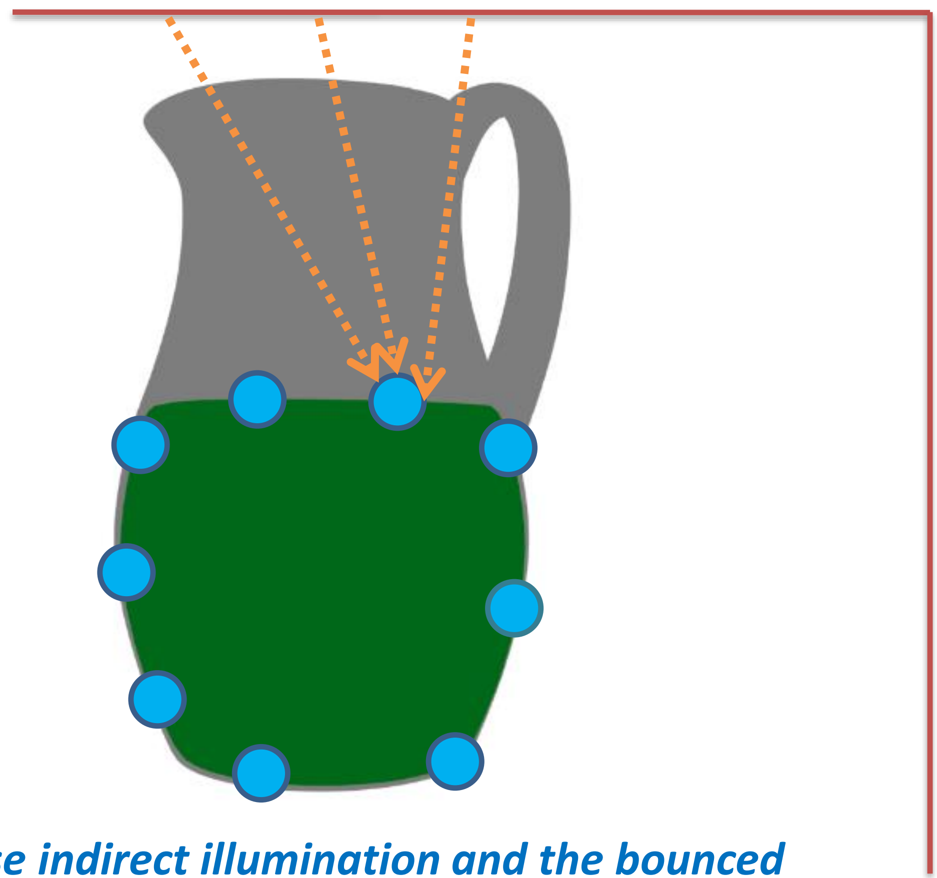
Our method – ***Caching volume samples***



Our method – ***Caching surface samples***

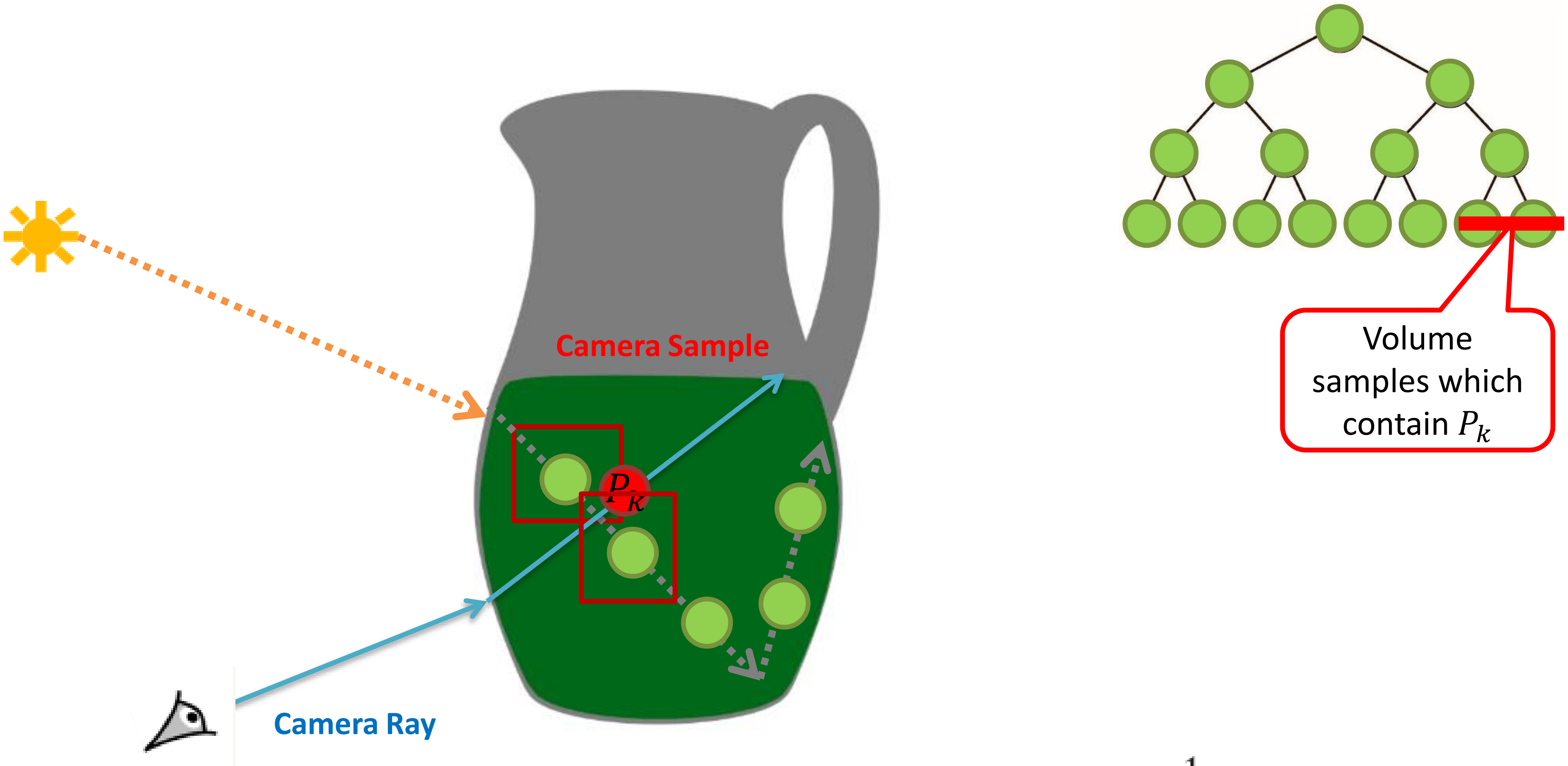


Light Surface Sample



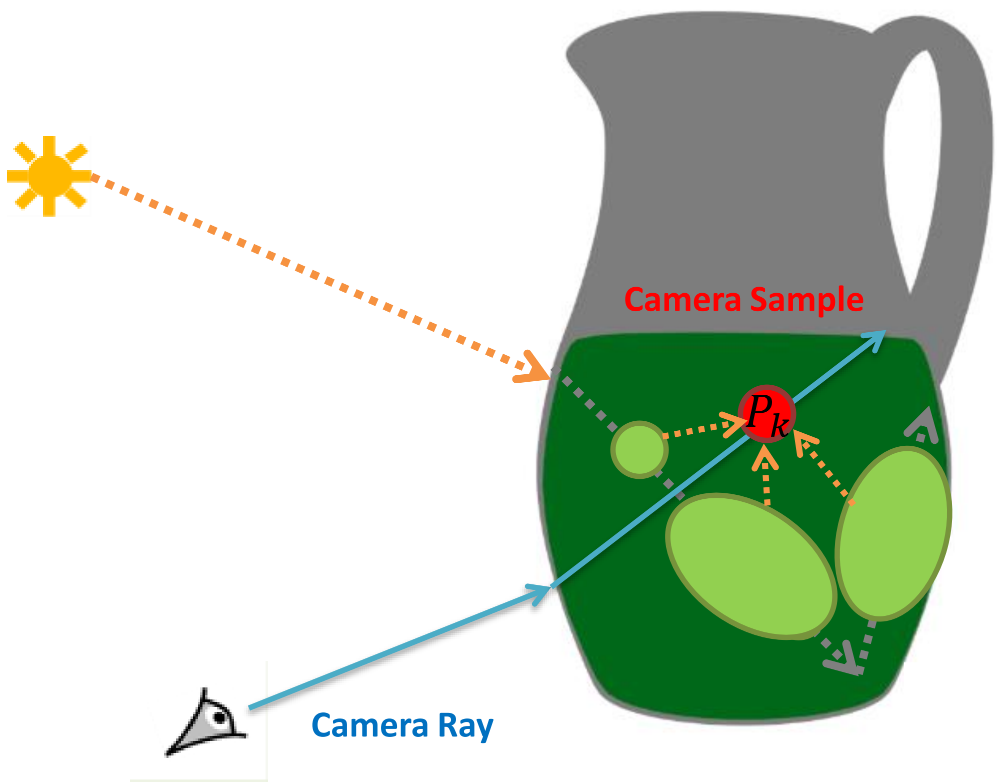
diffuse indirect illumination and the bounced multiple scattering

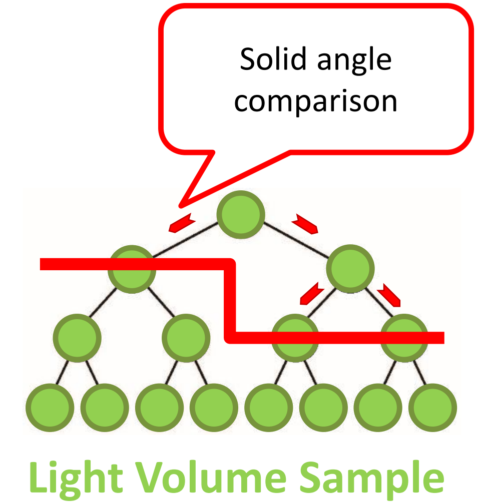
Our method – *Single scattering computation*



$$\text{single}(\boldsymbol{P}_k) = \mathrm{e}^{-\sigma_t d_k} \frac{1}{N} \sum_{v_i} \sigma_s p(\boldsymbol{o}, \boldsymbol{d}_{v_i}) \boldsymbol{I}_{v_i}.$$

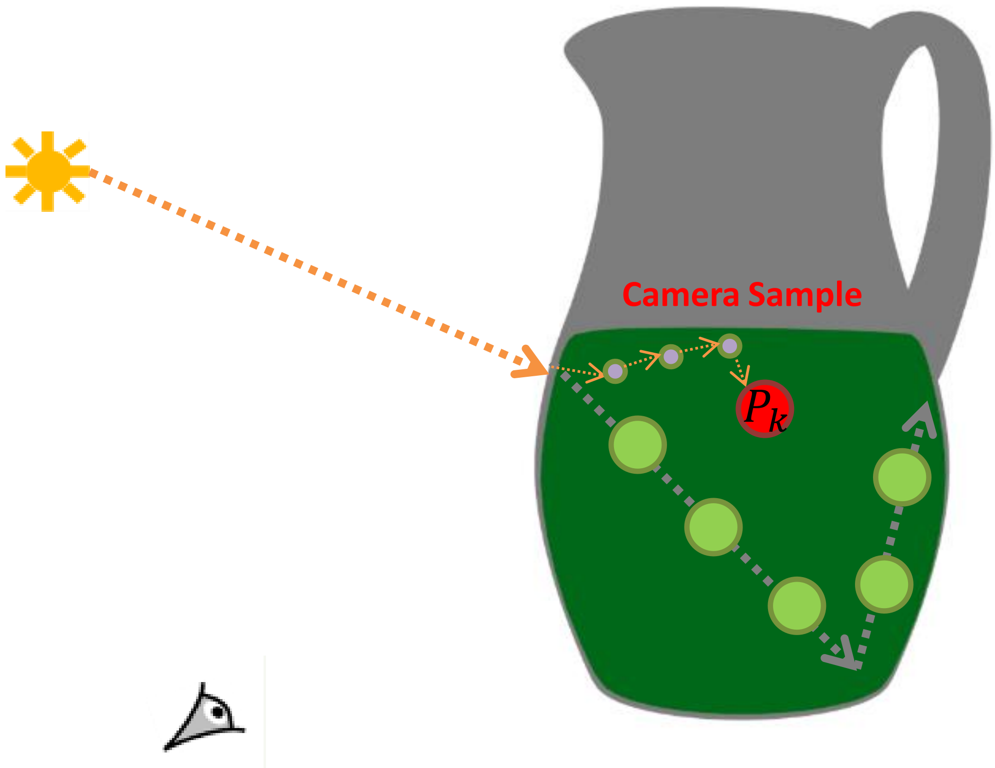
Our method – ***Double scattering computation***

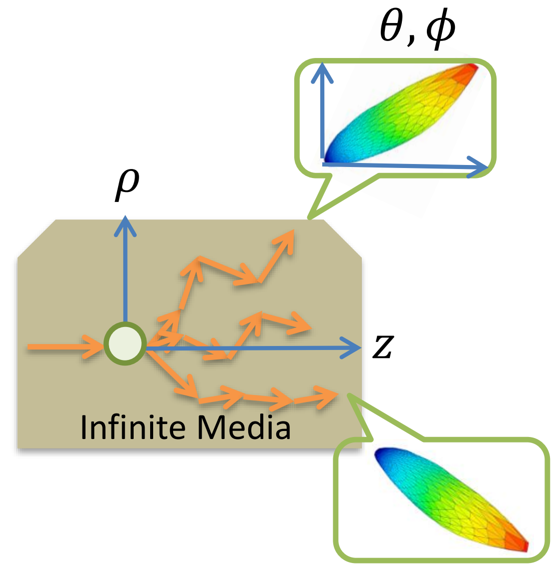




$$
\begin{aligned}
r_i &= \|\boldsymbol{P}_k - \boldsymbol{v}_i\| \\
\boldsymbol{m}_i &= (\boldsymbol{P}_k - \boldsymbol{v}_i)/\|\boldsymbol{P}_k - \boldsymbol{v}_i\| \\
\text{double}\,(\boldsymbol{P}_k) &= \mathrm{e}^{-\sigma_t d_k} \sum_{v_i} \frac{\mathrm{e}^{-\sigma_t r_i} I_{v_i}}{r_i^2} \sigma_s^2 p(\boldsymbol{o}, \boldsymbol{m}_i) p(\boldsymbol{m}_i, \boldsymbol{d}_{v_i}).
\end{aligned}
$$

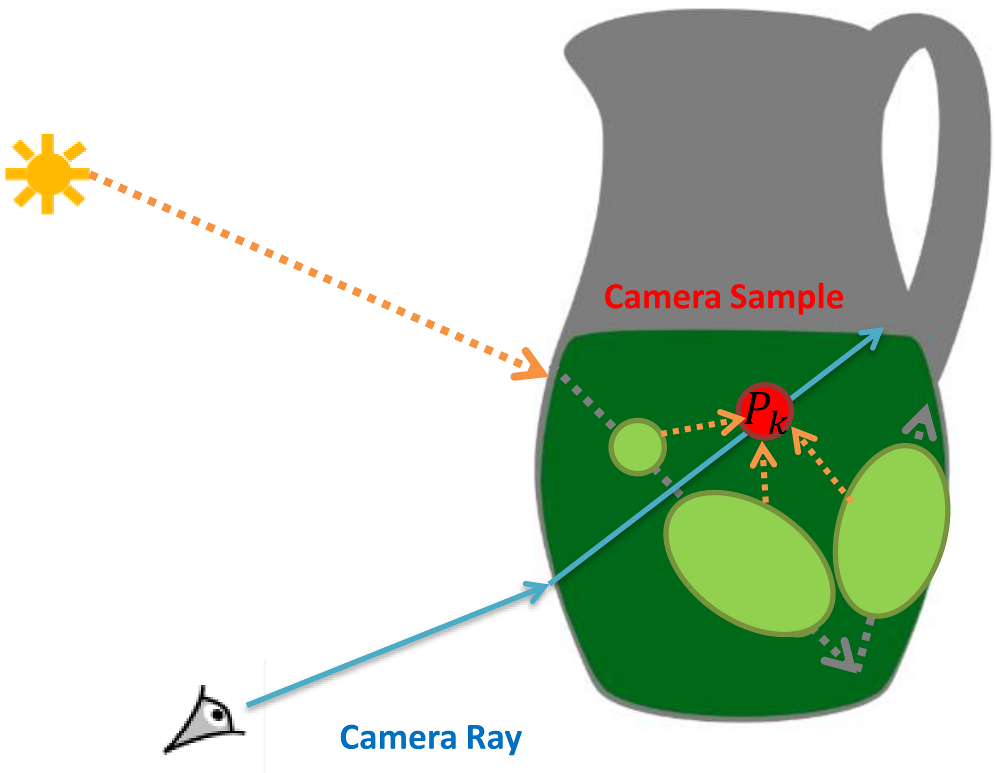
Our method – *Multiple scattering precomputation*

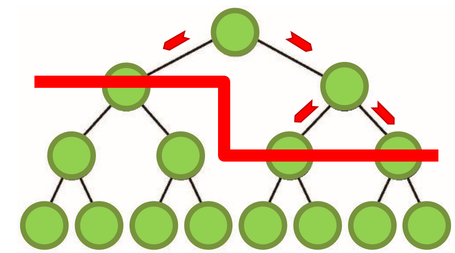




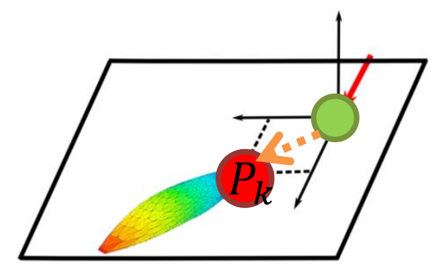
Precompute multiple scattering as a 4D table
(Shoot photons randomly to
the medium and record the interacts)
Independent of scenes

Our method – *Multiple scattering computation*



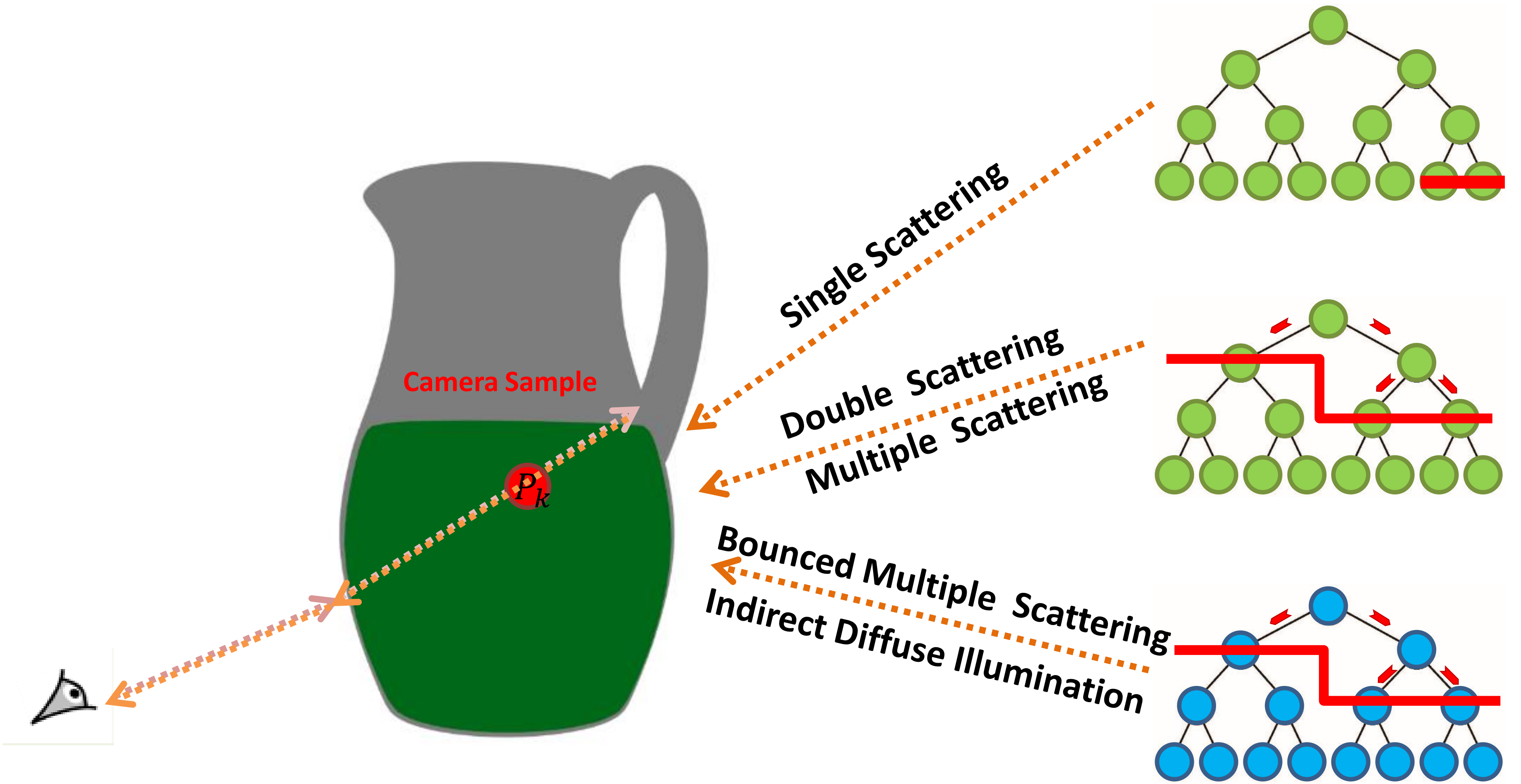


Light Volume Sample



$$
\begin{aligned}
z_k &= (\boldsymbol{P}_k - \boldsymbol{v}_i) \cdot \boldsymbol{d}_i \\
\rho_k &= \|(\boldsymbol{P}_k - \boldsymbol{v}_i) - z_k \boldsymbol{d}_i\| \\
\text{mult.}\,(\boldsymbol{P}_k) &= \mathrm{e}^{-\sigma_t d_k} \sum_{\boldsymbol{v}_i} L_o\left(\frac{\rho_k}{\ell}, \frac{z_k}{\ell}, T_{(\boldsymbol{v}_i, \boldsymbol{d}_i)}(\boldsymbol{o})\right).
\end{aligned}
$$

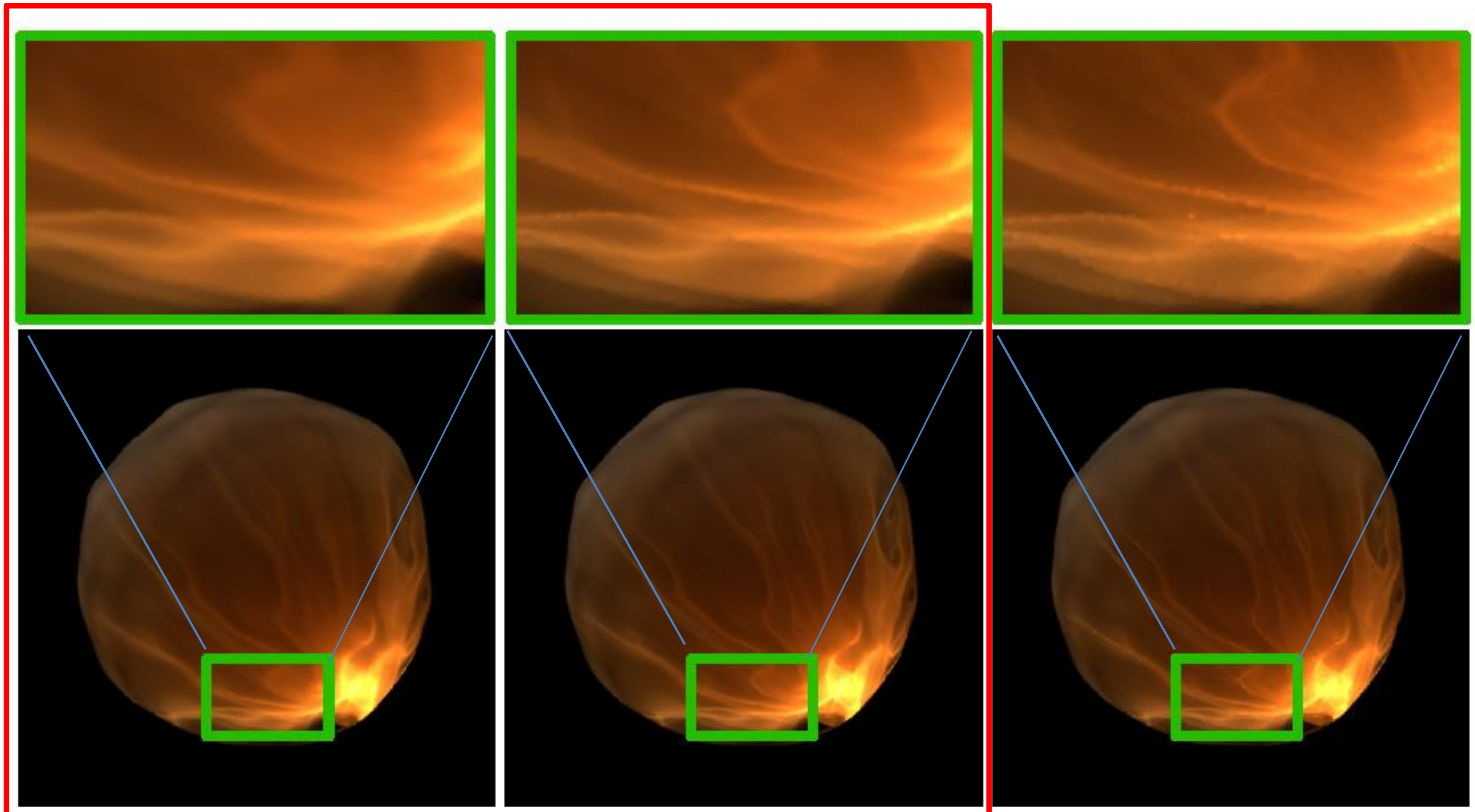
Our method - ***Full Solution***



Implementation

- Our technique: Mitsuba renderer, UPBP from SmallUPBP.
- Under bidirectional path tracing framework.
- Octree to organize volume/surface samples
- Camera ray sampling: exponential plus regular.
- Support point light only.

Results - Single Scattering

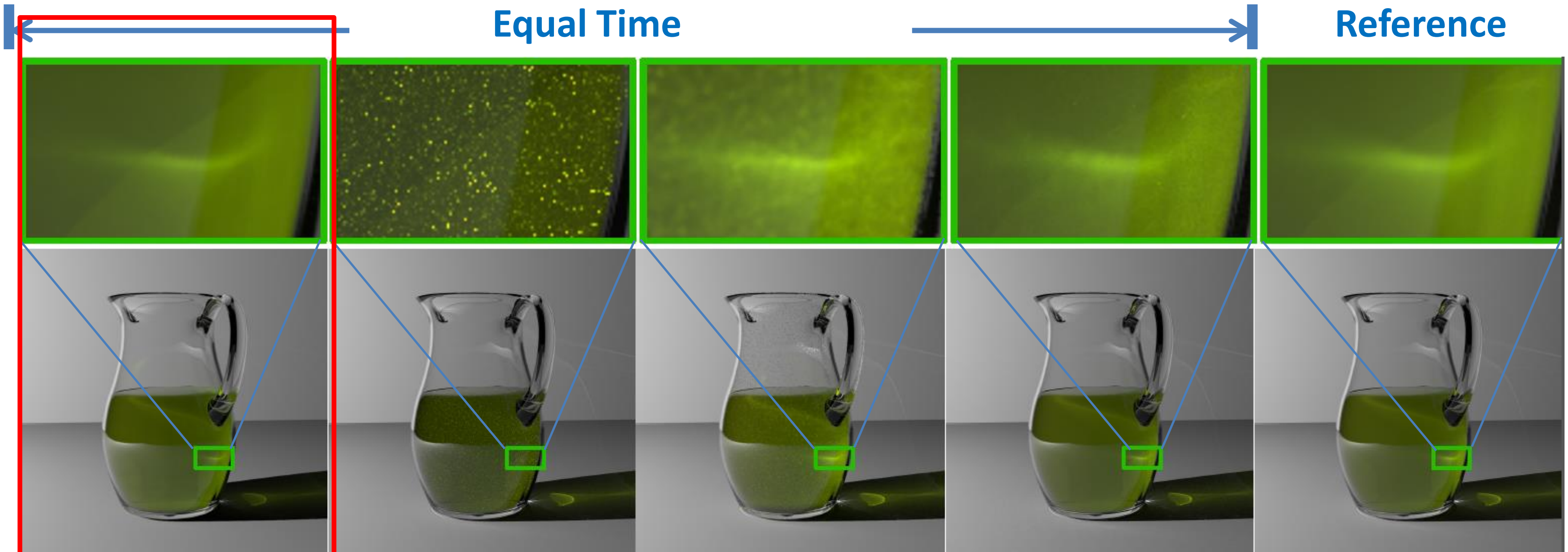


Ours, 1.2 GB, 15s

Ours, 2.35 GB, 46 s

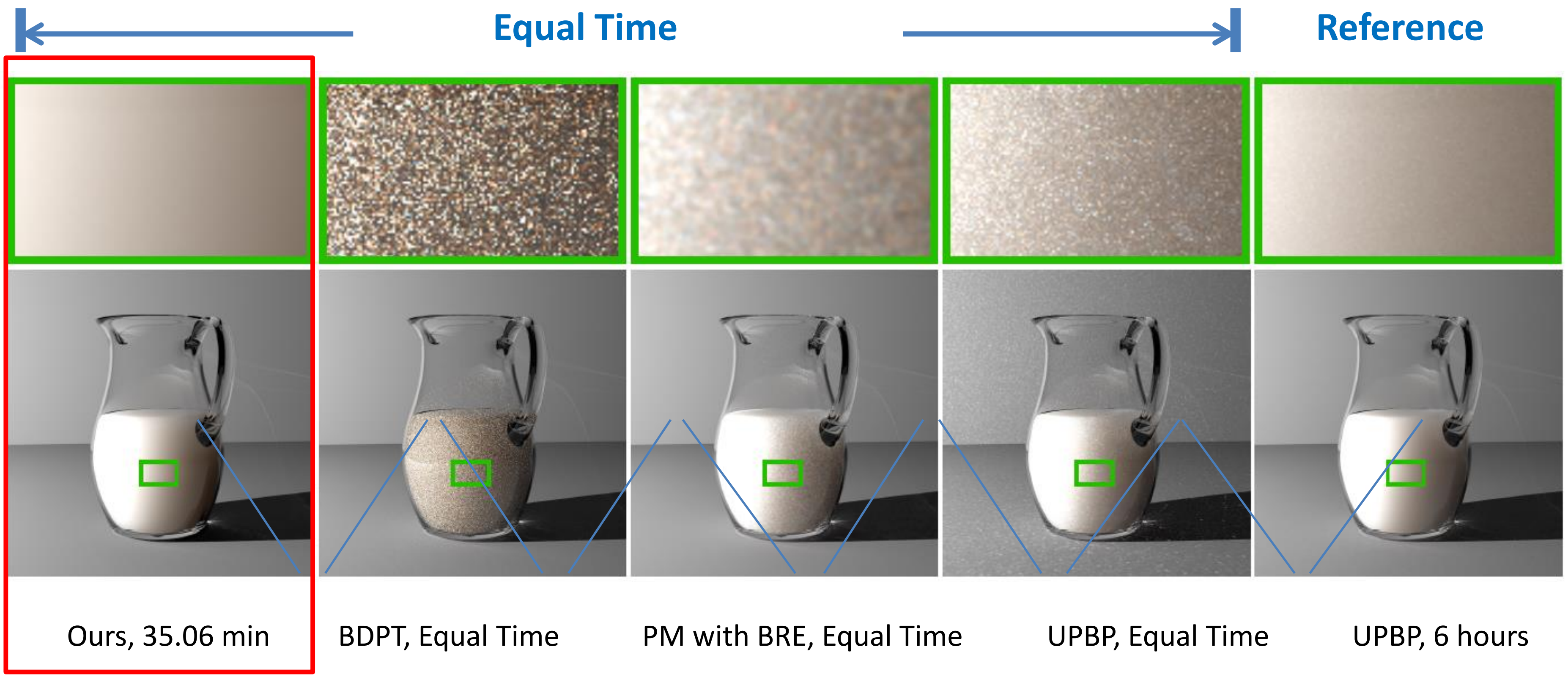
Holzschuch 2015, 46s, reference

Results



Ours, 41 mins | BDPT, 41 mins | PM with BRE, 41 mins | UPBP, 41 mins | UPBP, 6 hours, reference

Results


Equal Time
Reference
Ours, 35.06 min
BDPT, Equal Time
PM with BRE, Equal Time
UPBP, Equal Time
UPBP, 6 hours

Performance Measures

Scene	Max path	UPBP		Ours			Error
		Memory (GB)	*Rend. Time (h)*	*Memory (GB)*	*Pre. Time (s)*	*Rend. Time (m)*	*MSE*
Oil	11	16.25	6	1.21	22	41.13	1.1e-3
Wine	11	12.66	6	1.86	37	18.42	8.3e-5
Wax	50	13.41	3	1.49	32	3.07	2.4e-4
Milk	50	18.57	6	0.80	30	34.56	1.2e-3
Bumpy Sphere	50	6.75	3	1.34	33	2.43	3.2e-4

9x to 60x faster than UPBP (start-of-the art)

Point-Based Light Transport for Participating Media with Refractive Boundaries

Beibei Wang, Jean-Dominique Gascuel,
Nicolas Holzschuch

INRIA; Universite Grenoble-Alpes, LJK; CNRS, LJK

Conclusion

- Fast point-based method for single / double / multiple scattering computation.
- 9x to 60x faster than UPBP.
- Support both high order and low order scattering.
- Noise free.
- Easy to integrate to many rendering frameworks.

Can we do better?

Interactive Volumetric Rendering

- Provide **interactive rendering** of any homogenous participating media.
- Allow **Light editing / Material editing**.
- Produce pictures **visually identical** to reference solutions

Ours,
85.1 ms

[Wang et al. 2016],
13.7 mins

UPBP, 5 hours,
reference

Beibei Wang, Nicolas Holzschuch, *IEEE TVCG 2018*

Previous Work

- Multiple Scattering

Jimenez et al. [2009; 2015]

Fast, real-time models to compute sub-surface scattering effects in **high-albedo** materials.

(< 1ms)

Previous Work

- Without a refractive interface

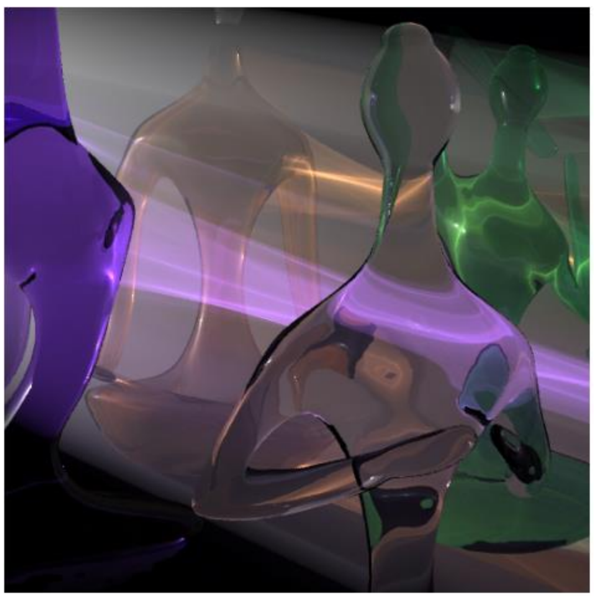
Sun et al. [2005]

Kaplanyan et al. [2010]

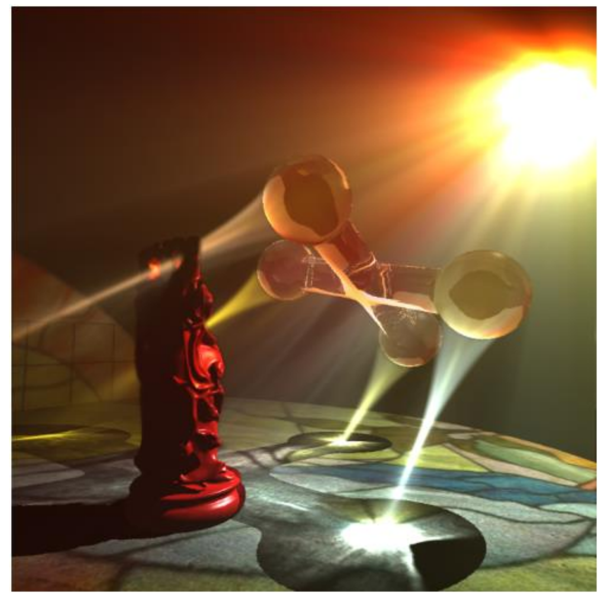
Billeter et al. [2012]

Previous Work

- Single Scattering, assume there is no refractive interface between the camera and volume caustics



Sun et al. [2010]



Hu et al. [2010]

Our approach

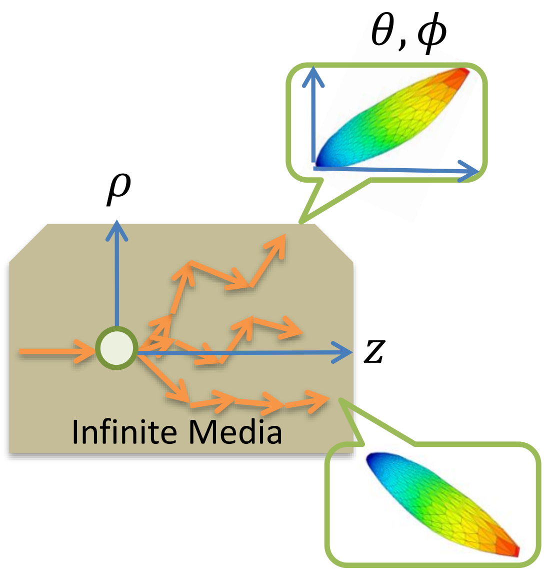
- Based on Point based method, we propose a GPU based implementation.
  - Interactive rendering of any homogenous media
  - Light editing
  - Material editing

Review Our Method

– Single scattering: local query
– Double scattering: global traversal
– Multiple scattering: precomputation

Precomputation for Entire Media Space

- In our CPU method, 4D for each material.
- Each material: mfp, albedo, g.



- So: 7 dimension: 10 * 10 * 10 * 128 * 64 * 36 * 18 * 1 float !!

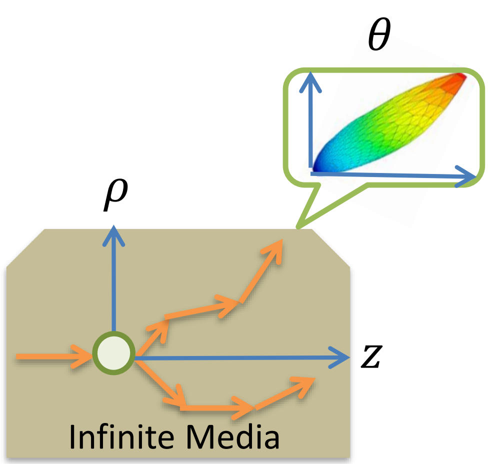
Precomputation: Multiple Scattering Effects

5D Structure

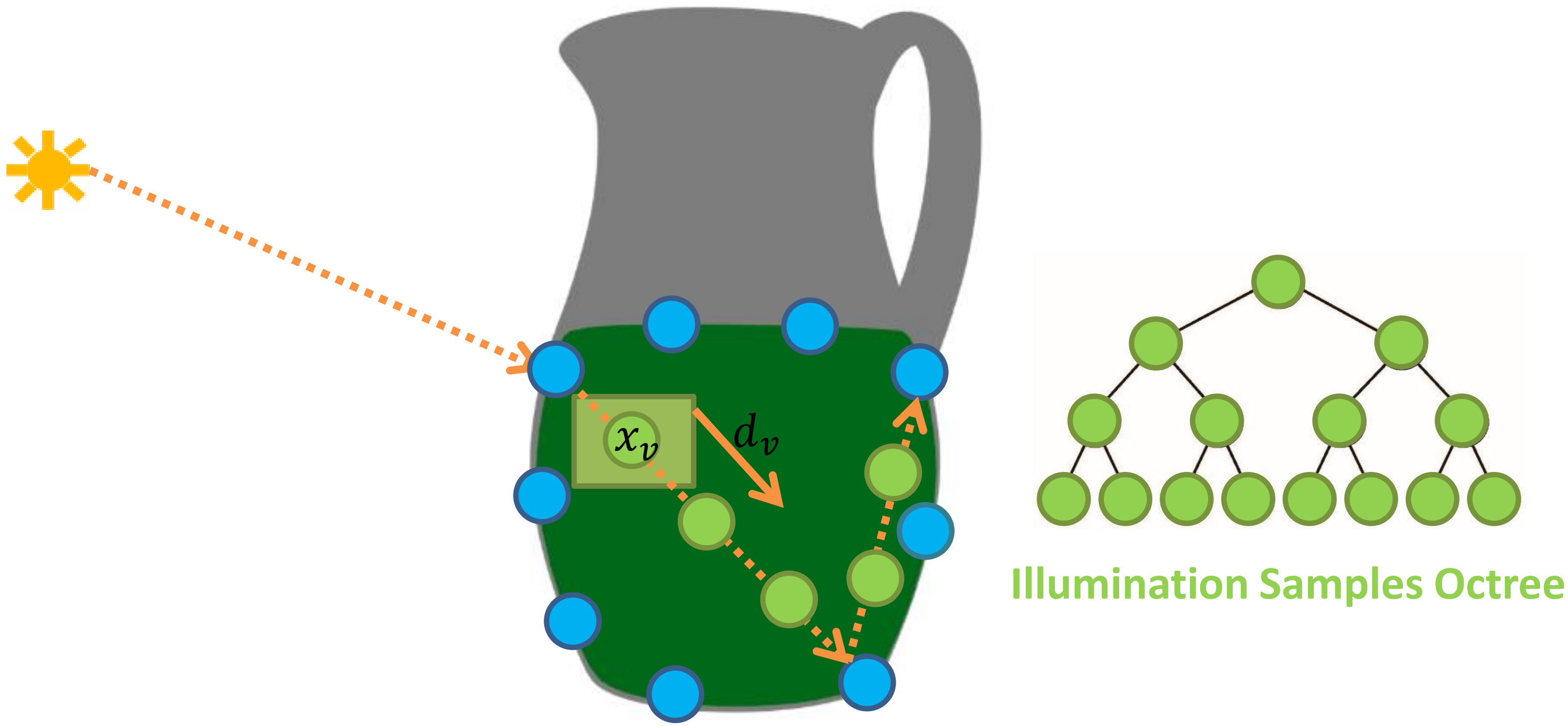
- 1D for material albedo
- 1D for phase function coefficient
- 3D for each sampled material

Overall cost: 600MB

At runtime, extract from this table to get the current materials.

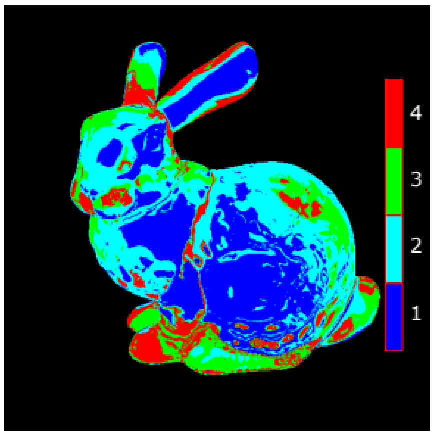


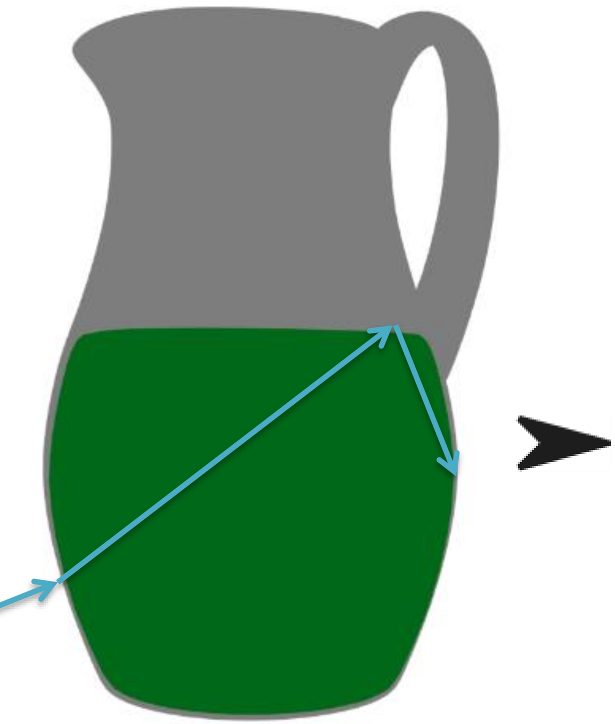
Caching Illumination Samples

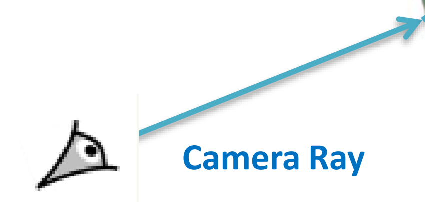


This is updated when the light sources change.

Rendering
Step 1: Generating Material Paths Buffer





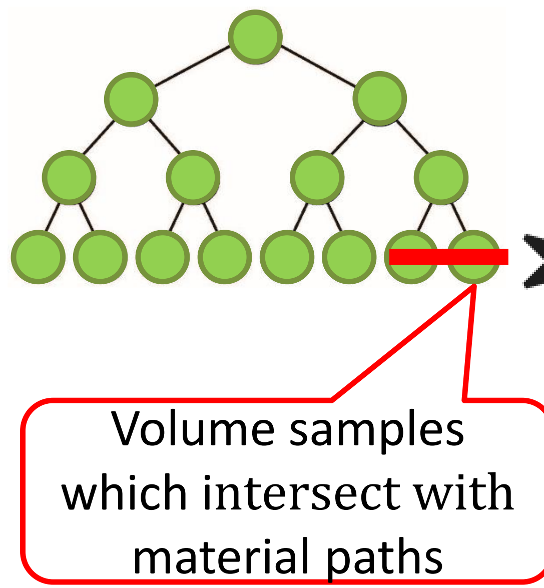



Material Paths Buffer


Surface Illumination

Rendering
Step 2: Computing Single Scattering

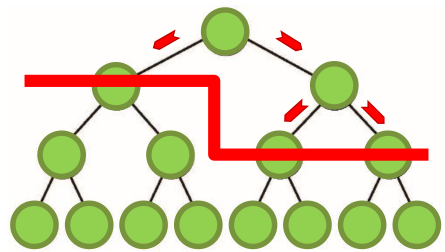
Material Paths Buffer

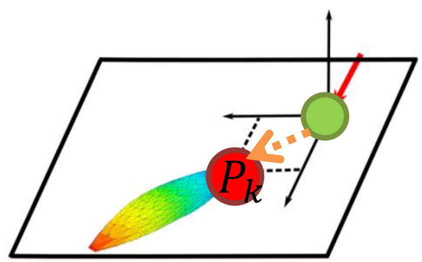


Single Scattering

Rendering
Step 3: Computing Multiple Scattering

Material Paths Buffer





Multiple Scattering

Full Solution

Surface Illumination

+

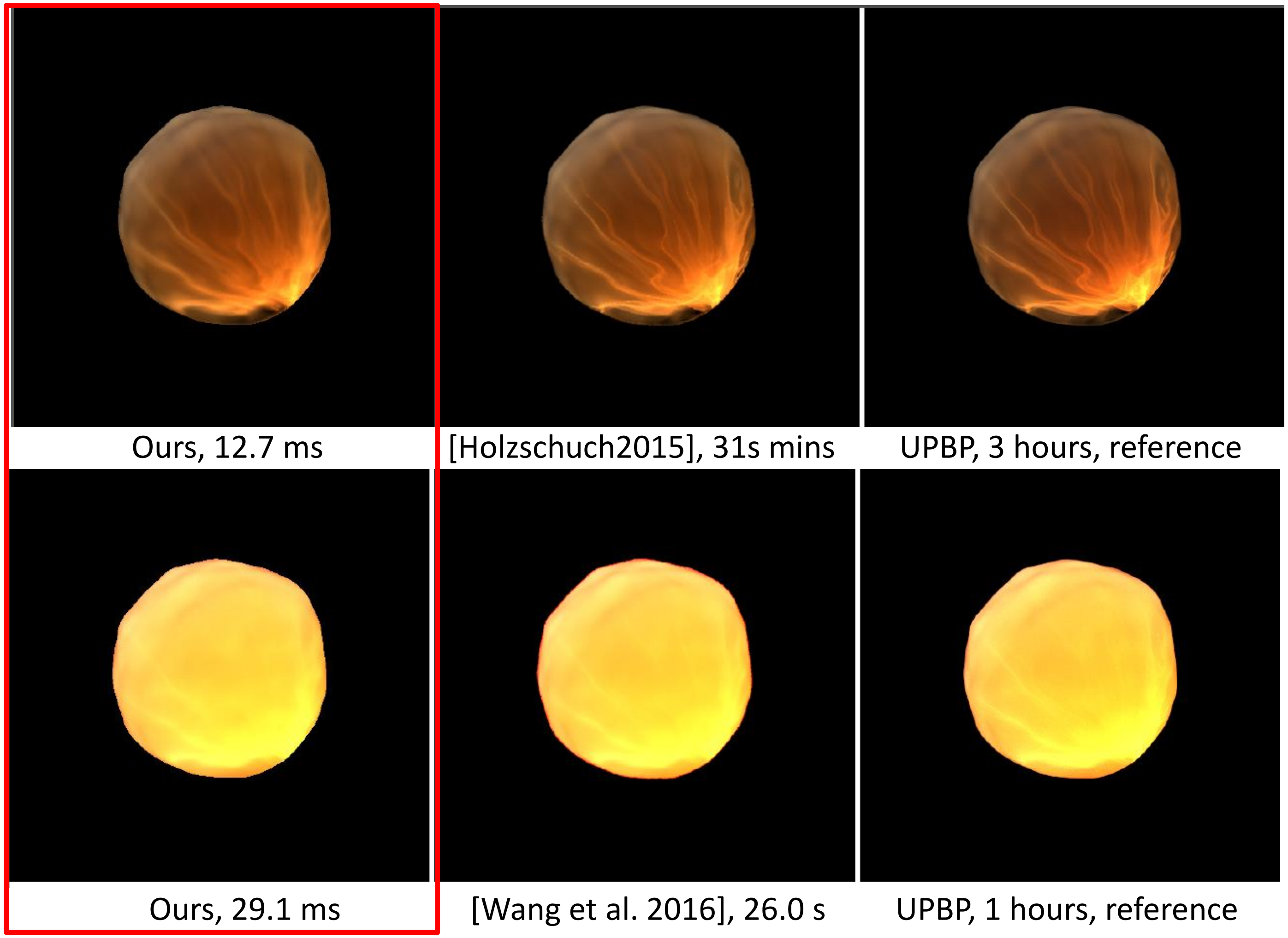
Single Scattering

+

Multiple Scattering

Full Solution

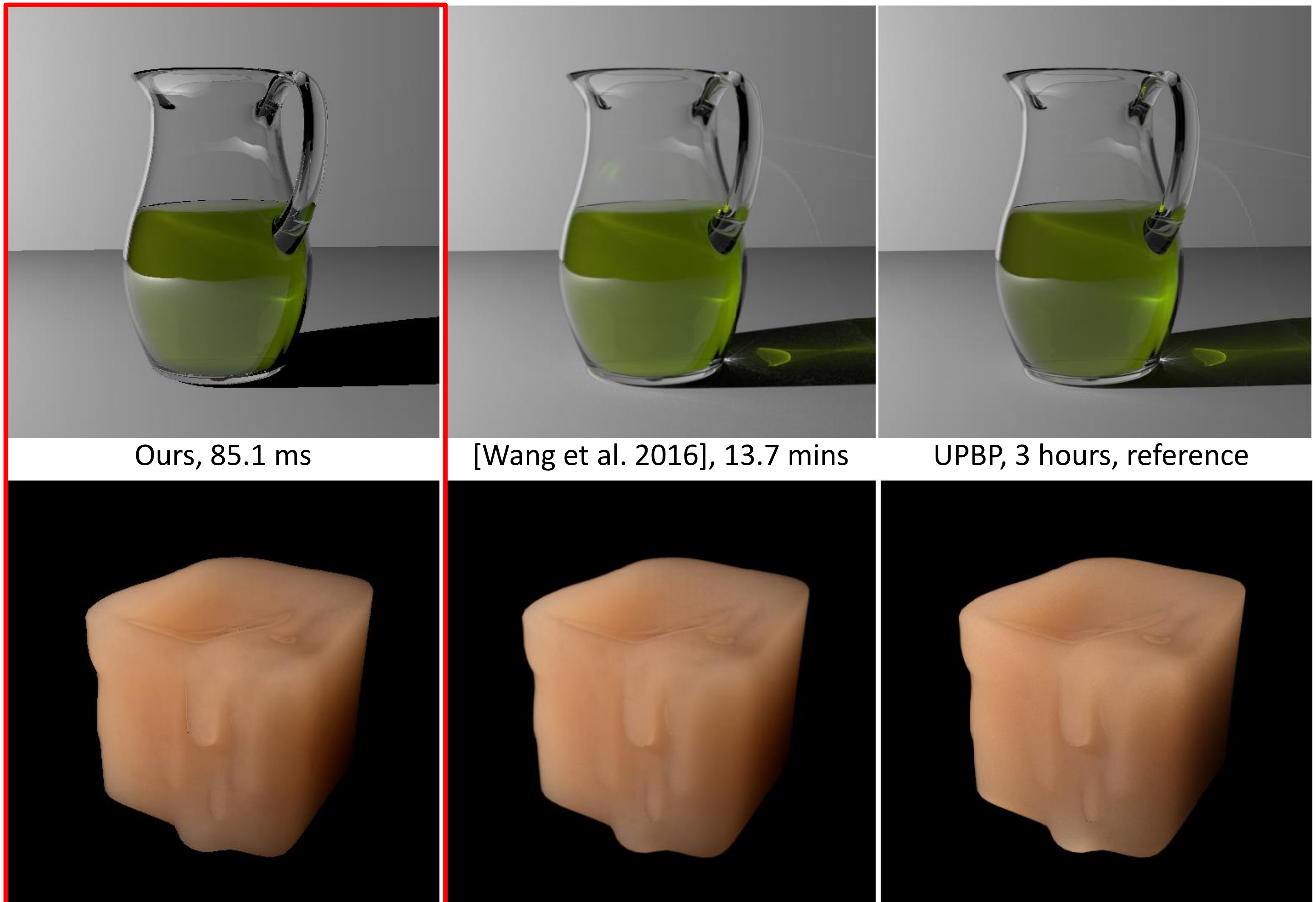
Results



Ours, 12.7 ms | [Holzschuch2015], 31s mins | UPBP, 3 hours, reference

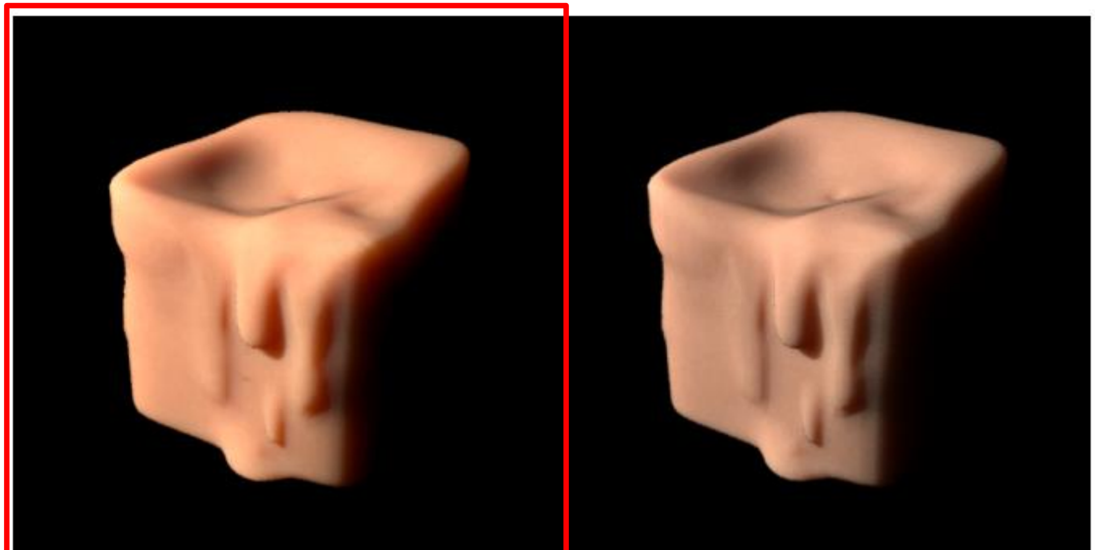
Ours, 29.1 ms | [Wang et al. 2016], 26.0 s | UPBP, 1 hours, reference

Results

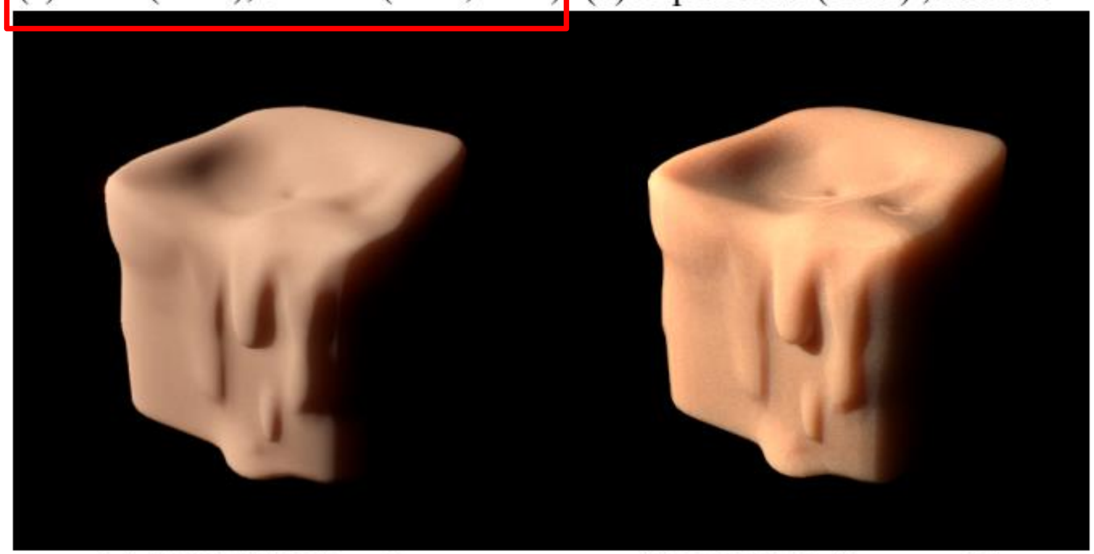


Ours, 85.1 ms | [Wang et al. 2016], 13.7 mins | UPBP, 3 hours, reference

Ours, 15.5 ms | [Wang et al. 2016], 26.6 s | UPBP, 3 hours, reference



(a) Ours (GPU), 72 ms. (CPU, 112s) (b) Dipole SSS (CPU) , 550 ms



(c) SSSS (GPU), < 1 ms (d) UPBP, Reference, 6 h

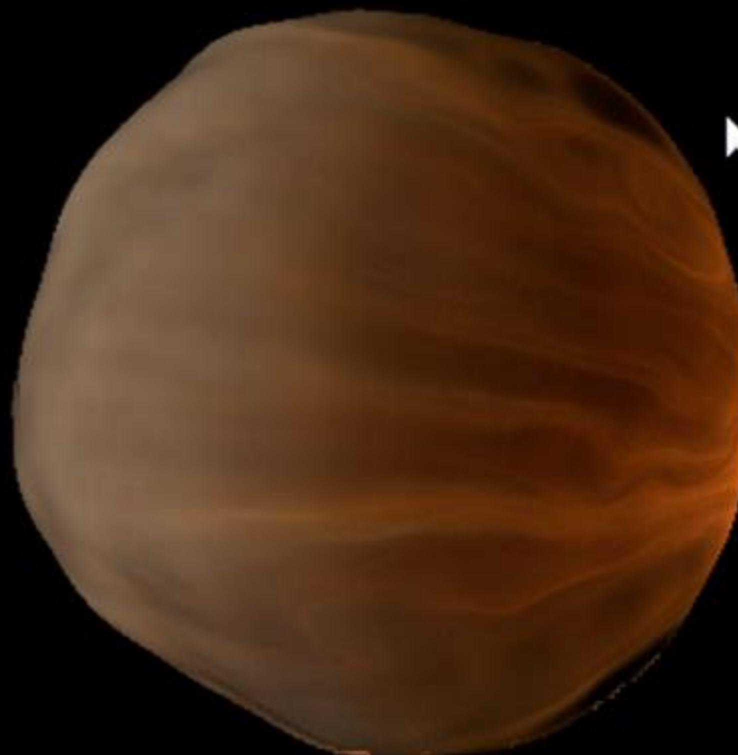
Performance Measures

Scene	UPBP			Ours (without editing)						Error
	Max path	*Memory (GB)*	*Rend. (h)*	*Memory (MB)*	*Cam. (ms)*	*Singl. (ms)*	*Mul. (ms)*	*Others. (ms)*	*Total. (ms)*	*MSE*
Oil	11	16.25	6	607	10.2	42.8	18.2	13.9	85.1	2.8e-3
Wax	50	13.41	3	601	1.5	1.9	6.1	6.0	15.5	7.1e-4
Bumpy Sphere	50	6.75	3	577	0.88	13.84	6.7	7.68	29.1	6.6e-4

Performance Measures

Scene	**Material Editing**				**Light Editing**			
	Table (ms)	*v. Tree (ms)*	*s. Tree (ms)*	*Total (ms)*	*Vol. (ms)*	*v. Tree (ms)*	*s. Tree (ms)*	*Total. (ms)*
Oil	0.04	0.28	12.5	12.8	4.4	11.2	12.7	28.3
Wax	0.06	0.16	4.5	4.72	6.3	18.4	4.4	19.1
Bumpy Sphere	0.06	0.18	7.0	7.24	3.2	10.1	6.5	19.8

Single Scattering Only.



Editing the position of the light source in real-time. [33 ms]

This is recorded in real time.

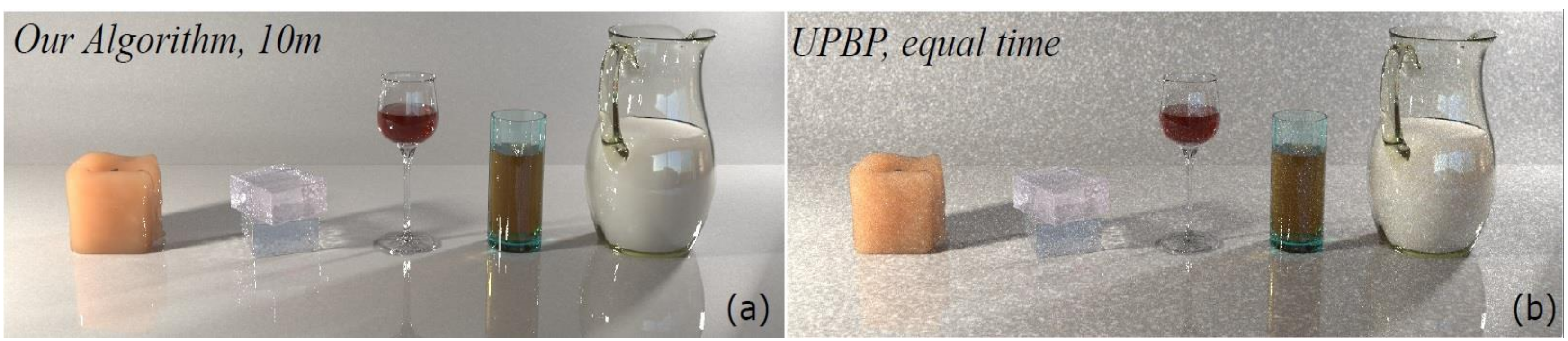
Conclusion

- Provide interactive rendering of any homogenous participating media.
- Allow Light editing / Material editing.
- Produce pictures visually identical to reference solutions.

Precomputed Multiple Scattering for Rapid Light Simulation in Participating Media

Beibei Wang, Liangsheng Ge, Nicolas Holzschuch.

IEEE TVCG 2019

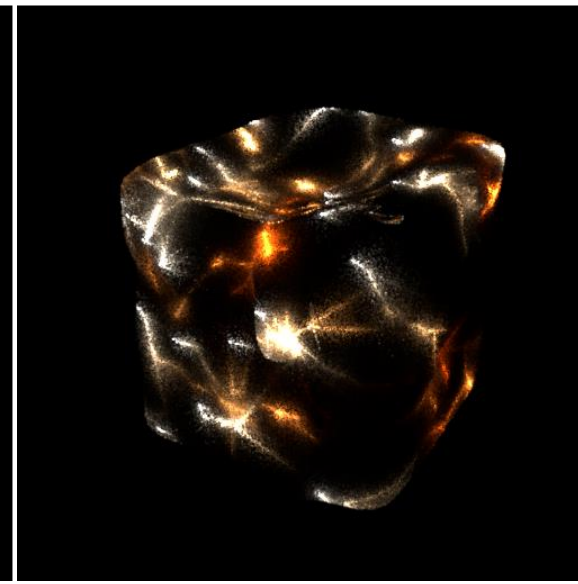


Motivation

Virtual Ray Lights

[Novák et al 2012]

VRL, multiple scattering, 1.8 mins



VRL, multiple scattering, 80 s

Multiple scattering with precomputation

- Precompute multiple scattering.

  -- assuming infinite media.

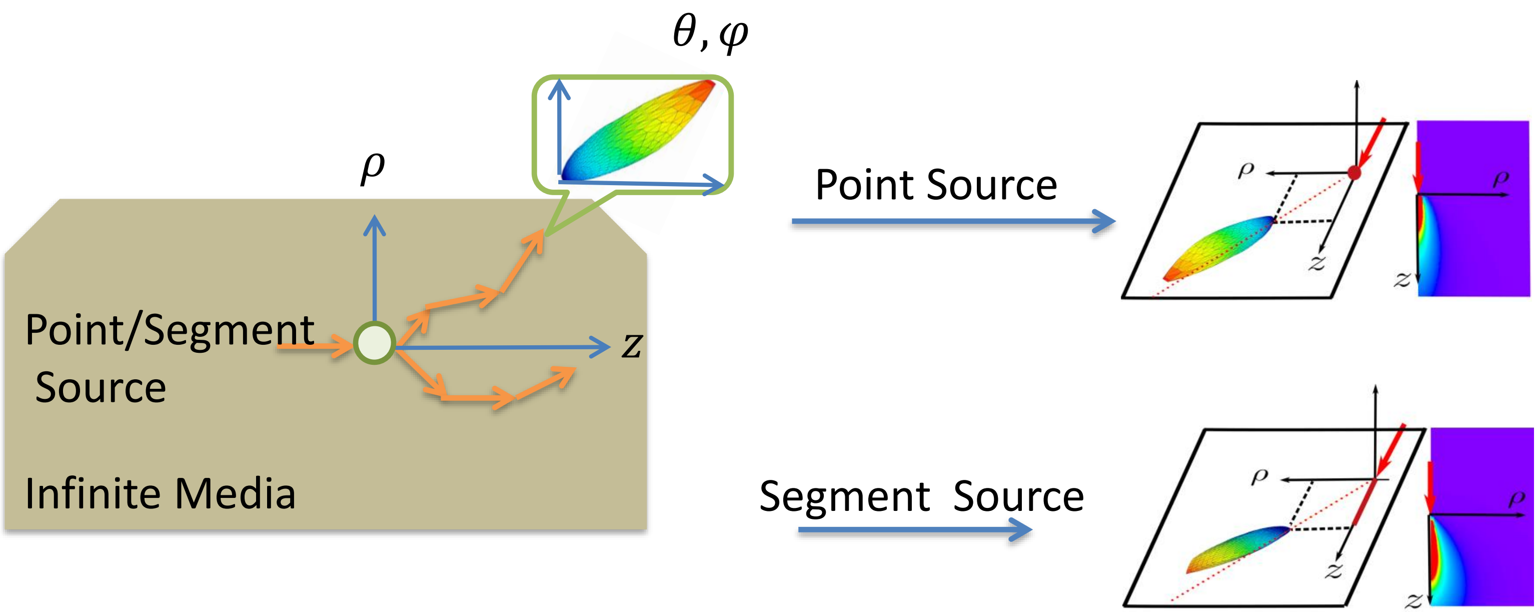
- Use this precomputation

  -- to extend existing algorithms.

  -- make them converge faster.

  -- multiple scattering only.

Precomputation Stage

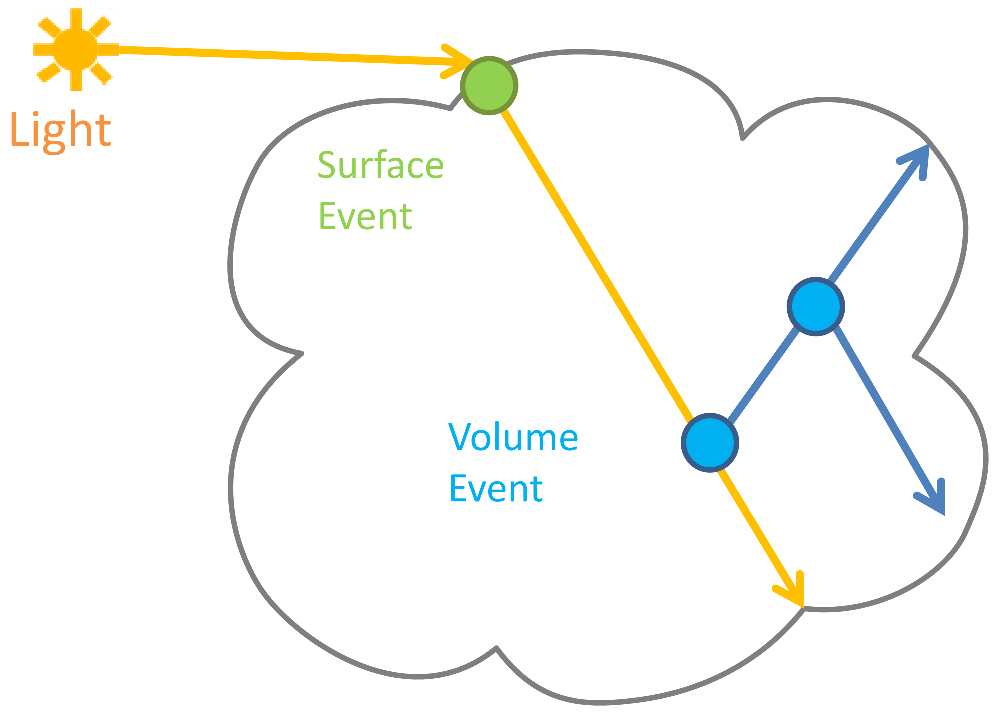


Precompute multiple scattering and store it in two tables

Apply to Existing Algorithms

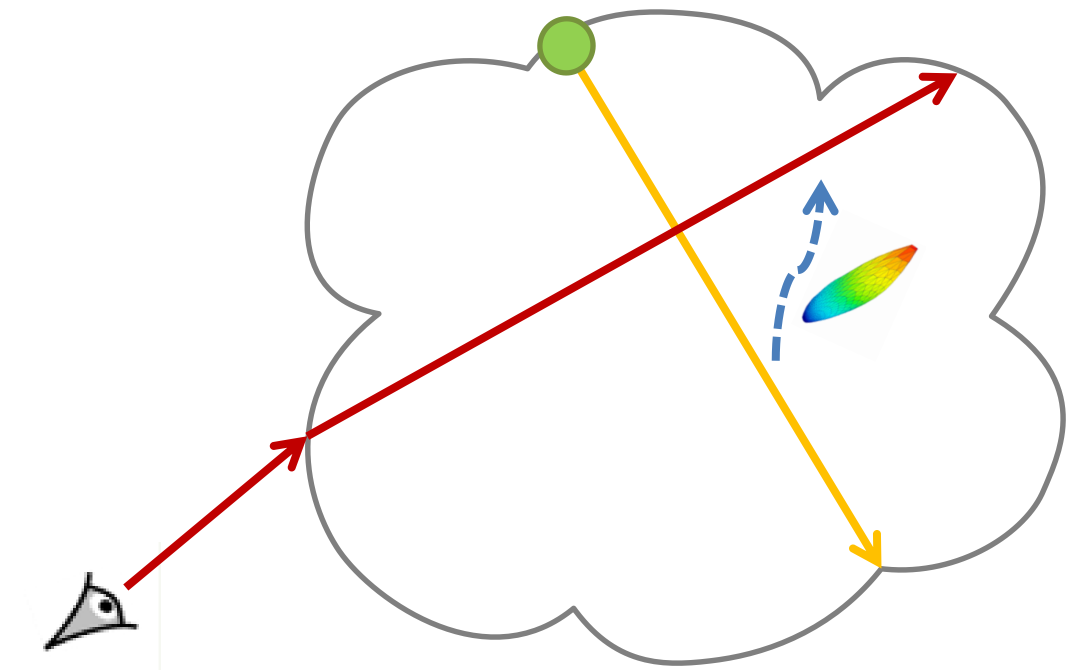
- Virtual Ray Lights (**VRL**).
- Unified Point, Beams and Paths (**UPBP**).
- Manifold-Exploration Metropolis Light Transport (**MEMLT**).

Application to VRL - Lighting Stage



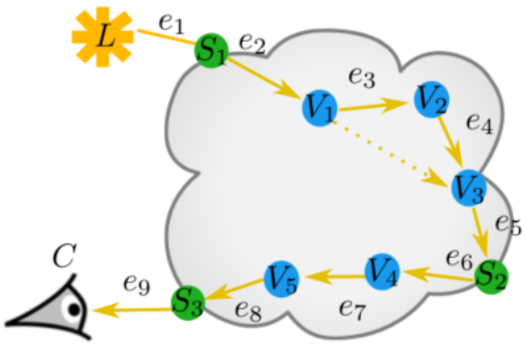
Store the rays (orange line) inside the medium after a surface event

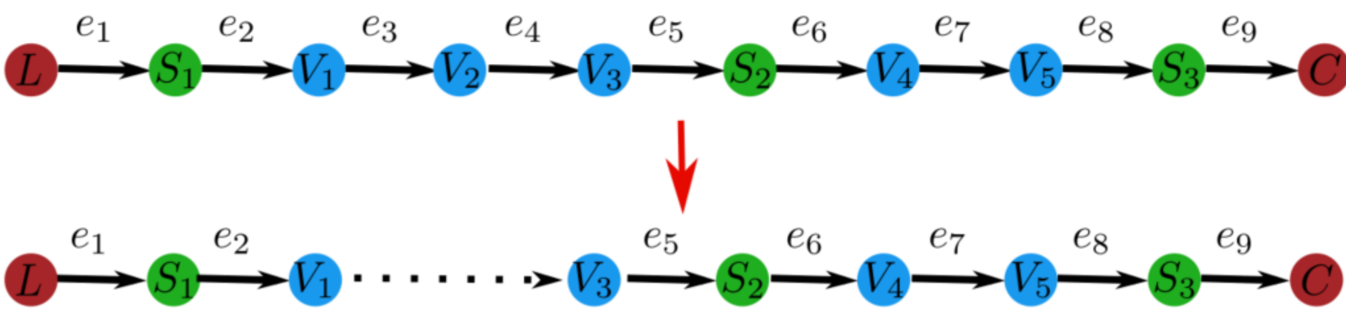
Application to VRL - Rendering Stage



Use the precomputed tables to compute multiple scattering from these light rays (orange line) to the camera rays (red line).

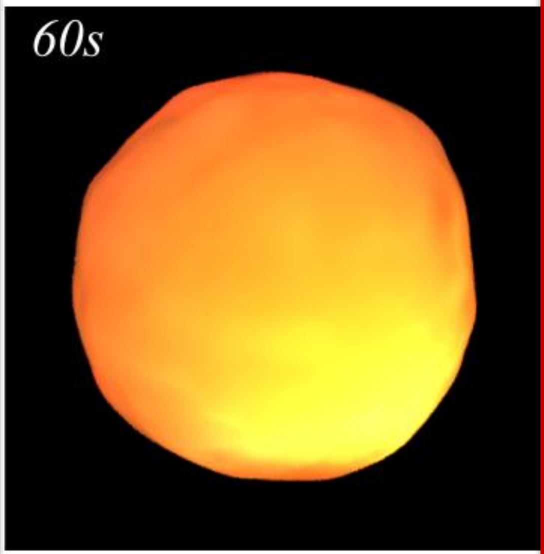
Application to MEMLT



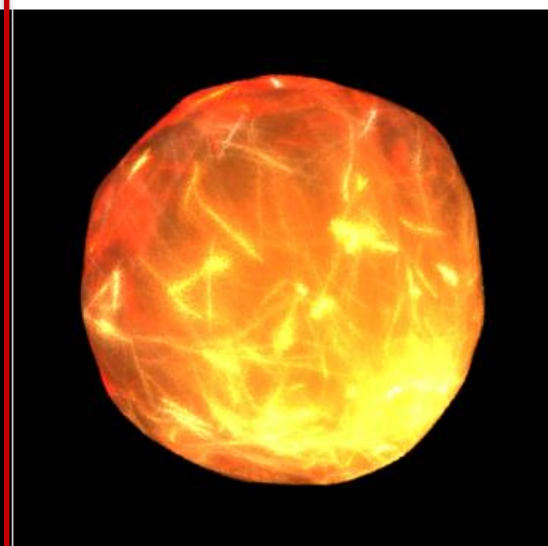


Results – Multiple scattering Only

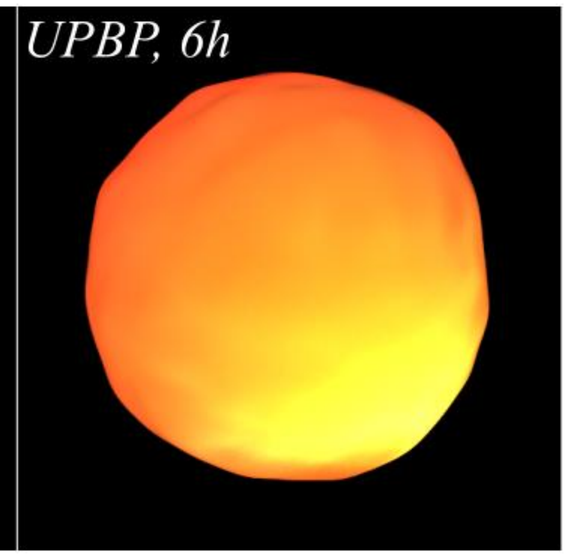
Our Method



Equal Time



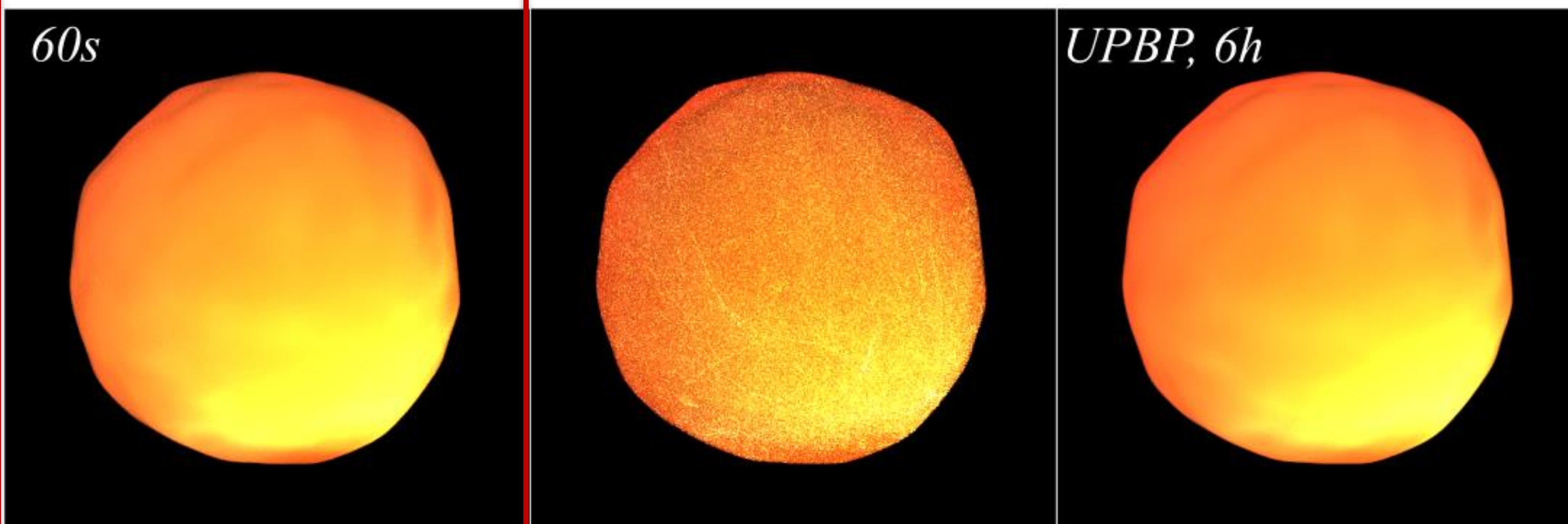
Reference



VRL

Results – Multiple scattering Only

Our Method **Equal Time** **Reference**



UPBP

Results – Multiple scattering Only

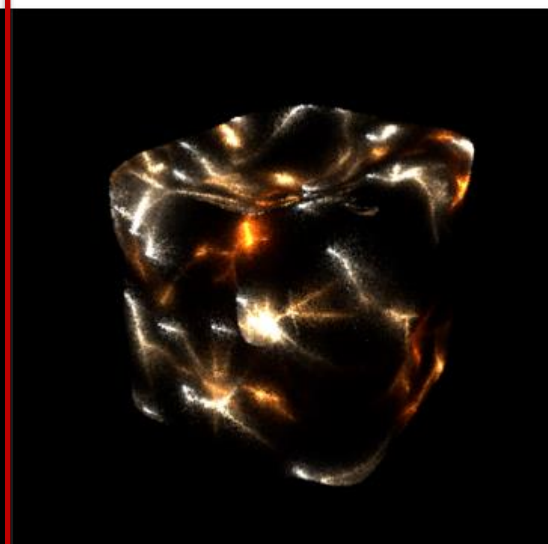
Our Method **Equal Time** **Reference**

MEMLT

Results – Multiple scattering Only

Our Method

Equal Time



Reference

VRL

Results – Multiple scattering Only

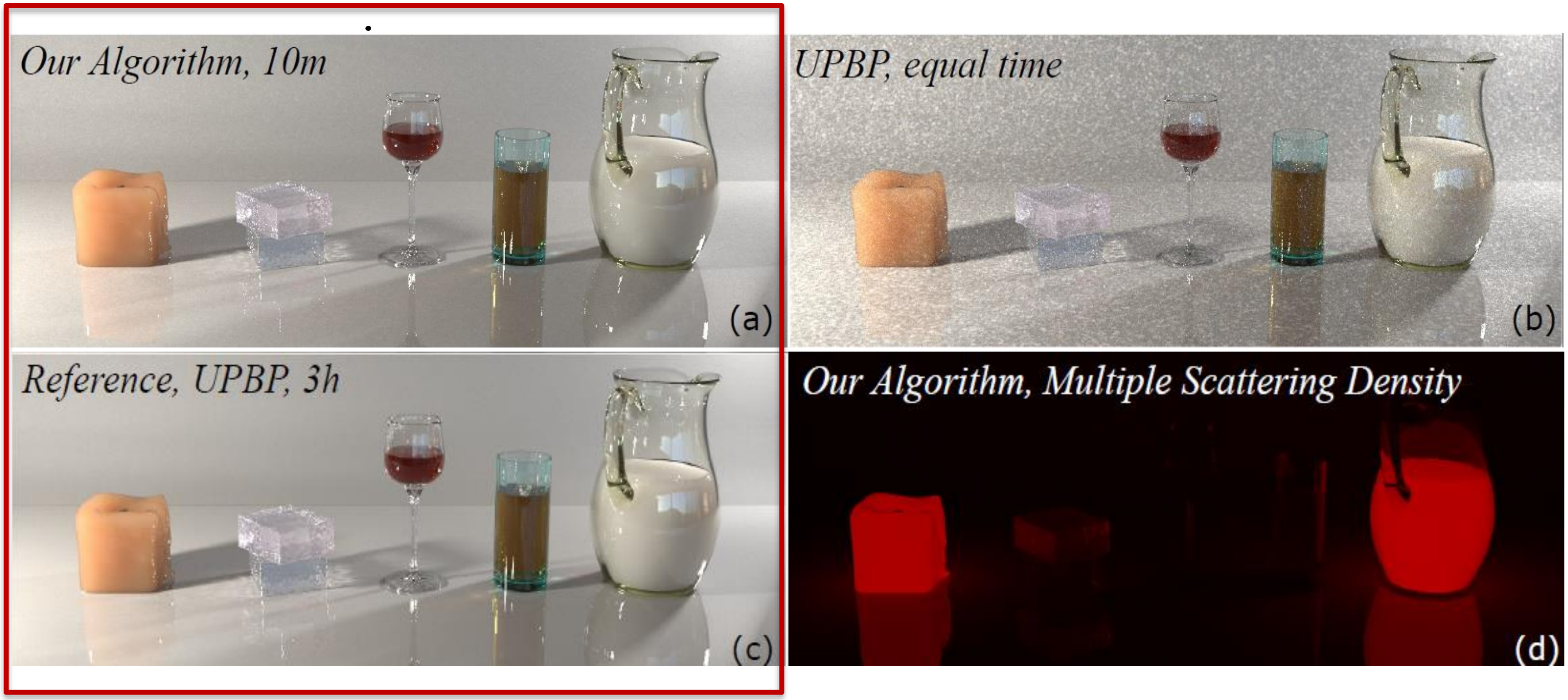
Our Method **Equal Time** **Reference**

UPBP

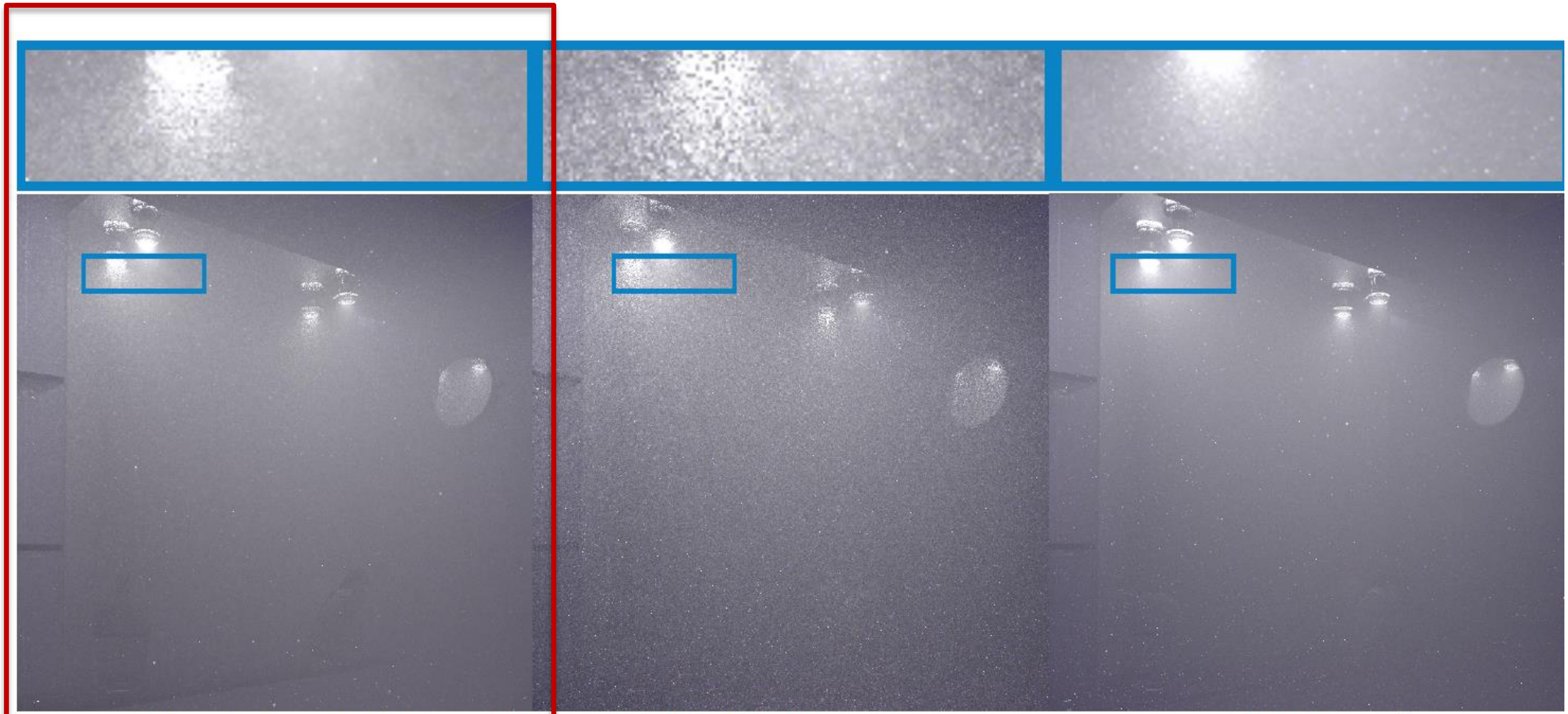
Results – Multiple scattering Only

MEMLT

Results – Full solution

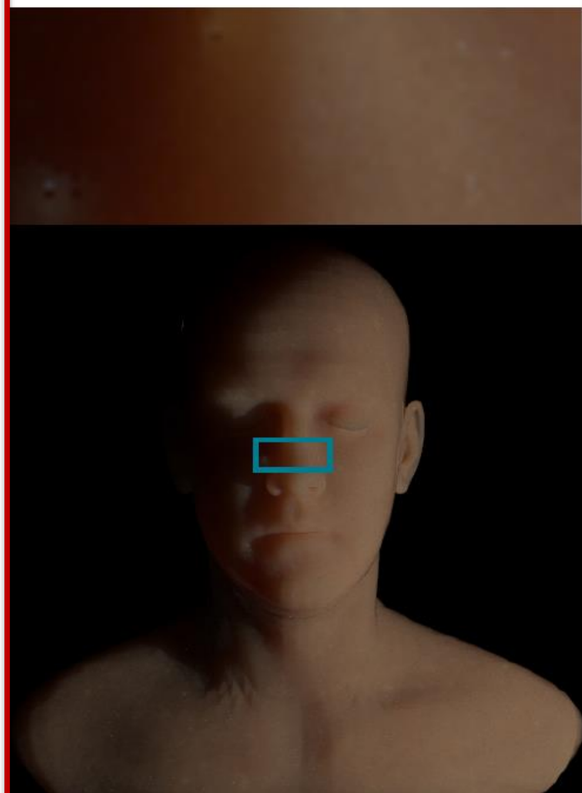


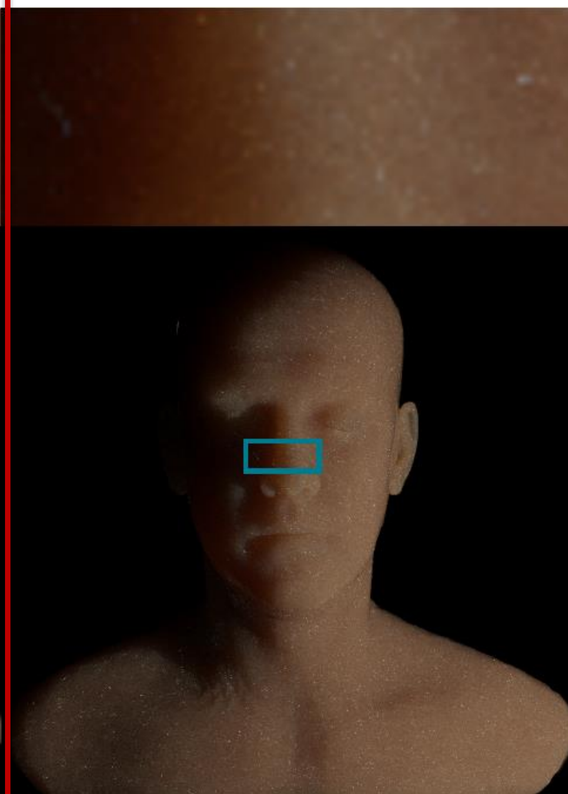
Results – Full solution

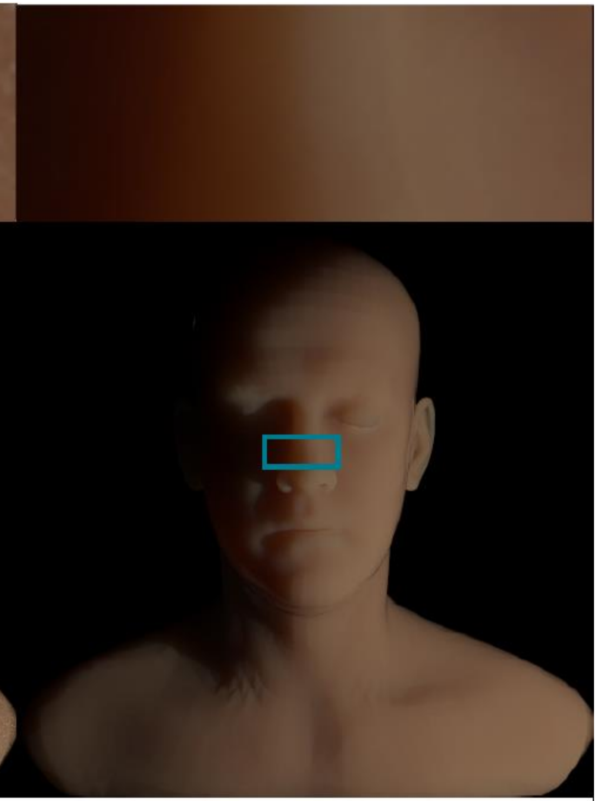


(a) *Our Algorithm (UPBP), 600 s* (b) *UPBP, Equal Time* (c) *UPBP, 6 h*

Results – Full solution


Our Algorithm (UPBP), 600 s


UPBP, Equal Time

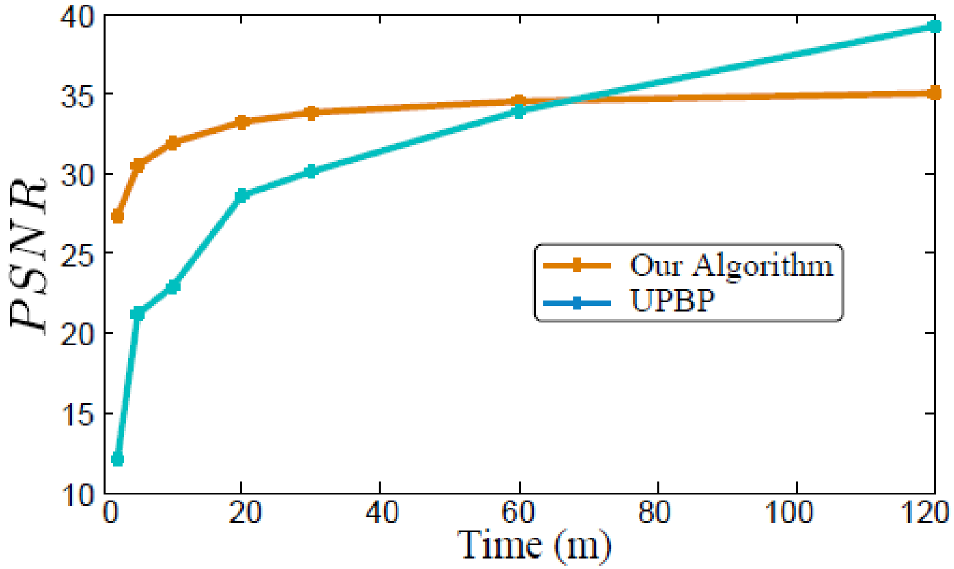

UPBP, Reference, 6h

CONVERGENCE SPEED COMPARISON (UPBP)

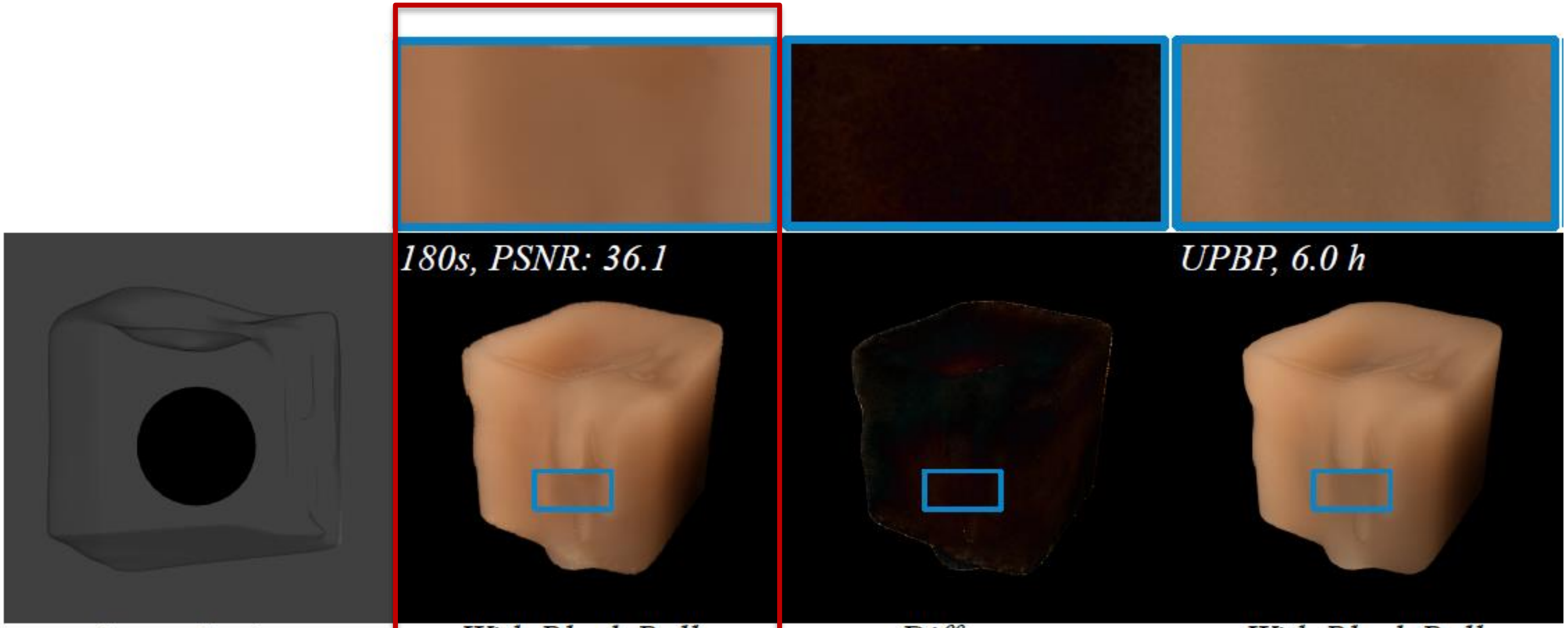
Bumpy Sphere Scene

Limitations

- Visibility between light rays and camera rays.
- Biased.

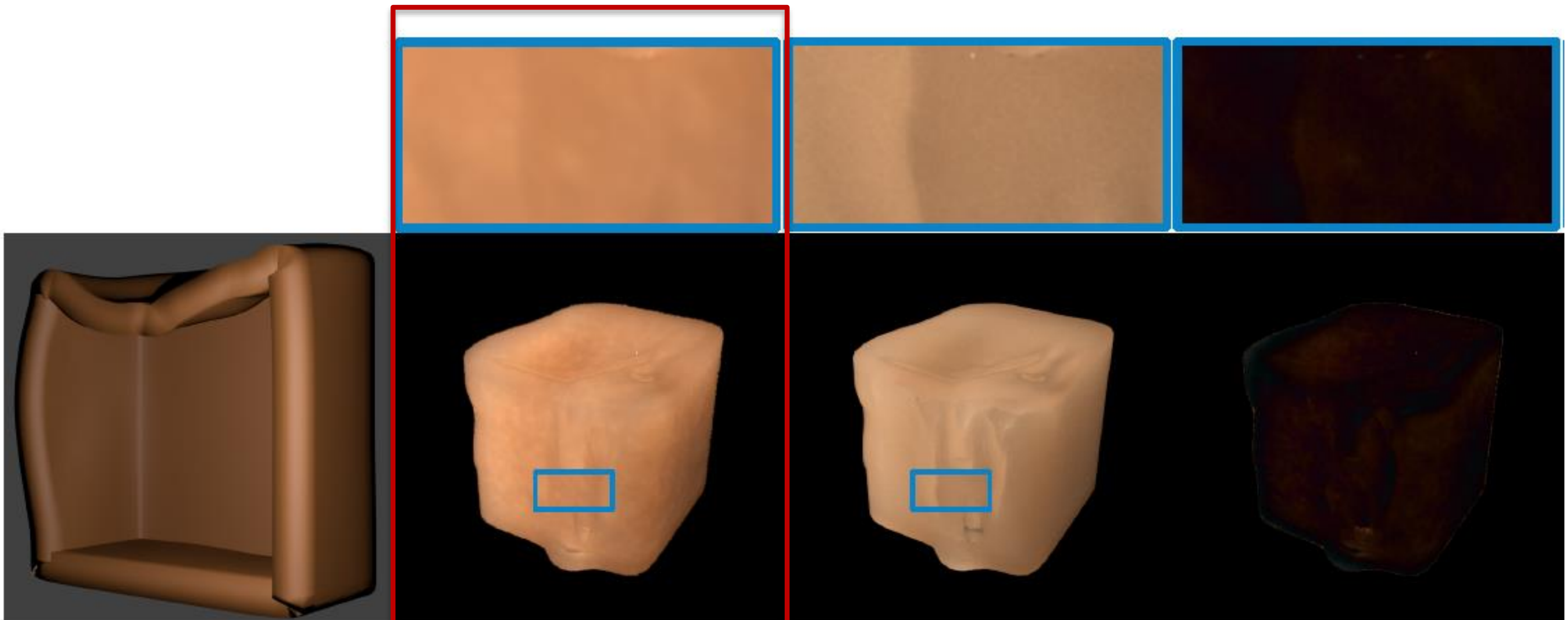


Limitations



Scene Setting | *With Black Balls, Ours (VRL)* | *Difference* | *With Black Balls, Reference*

Limitations



Scene Setting

Shell thickness: 3 mfp, *Ours (VRL), 20 min*

Shell thickness: 3 mfp, *Reference, UPBP, 6 h*

Difference

Conclusion

- New representation to **represent multiple scattering** in participating media.
- Easy to integrate with existing algorithms.
- Reduces noise, with **limited impact on accuracy**.
- Interesting for materials with **large albedo and small mean-free-path**.

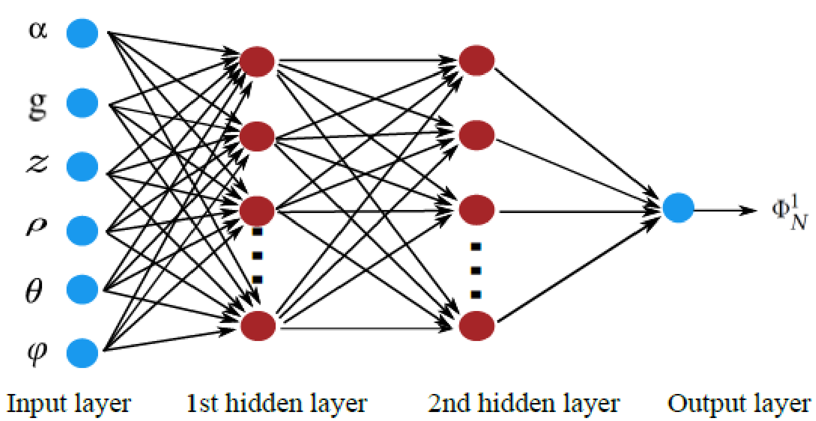
Future Work

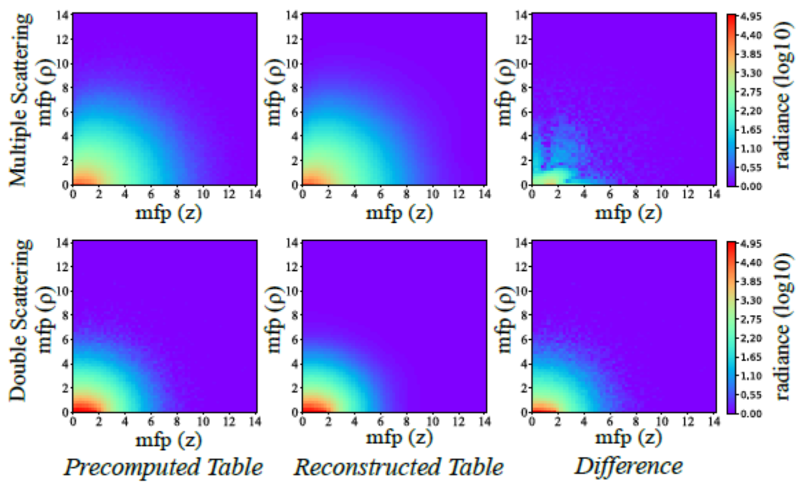
- More compact representation
- More accurate solution
  - using the table from further bounces instead of from the second one and automatic switch depend on the accuracy
- Layered materials with media

Interactive Simulation of Scattering Effects in Participating Media Using a Neural Network Model

Liangsheng Ge*, **Beibei Wang***, Lu Wang, Xiangxu Meng, Nicolas Holzschuch

IEEE TVCG 2020 (will be presented at I3D 2020)





Thanks! Question?

Beibei.wang@njust.edu.cn
https://wangningbei.github.io/

Precomputation Table

Scene	Memory (MB)	Time (s)
Oil	158.29	52
Wine	108.81	45
Wax	78.77	370
Milk	97.11	442

Material parameters and Precomputation times with 500 M particles, 20608 lobes and 36 x 18 directions for each lobe.

EGSR 2016

Diffusion Comparison

Our Algorithm (UPBP), 600 s *Diffusion Approx. (UPBP), 600 s* *Difference* *UPBP, 600 s* *UPBP, Reference, 6h*

TVCG 2019

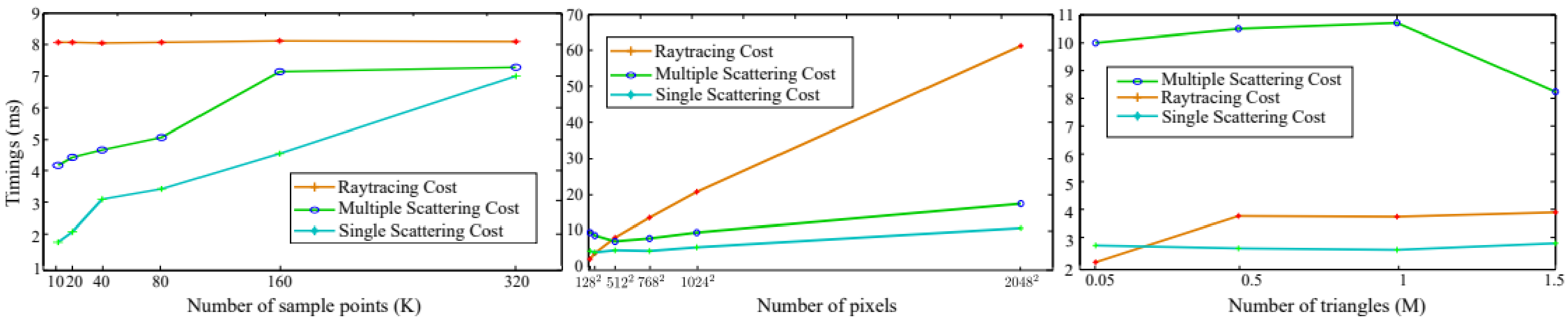


Fig. 25. Performance of the GPU implementation, over varying parameters: points, pixels and scene complexity.

TVCG 2018