Sorting-free Pre-integrated Projected Tetrahedra

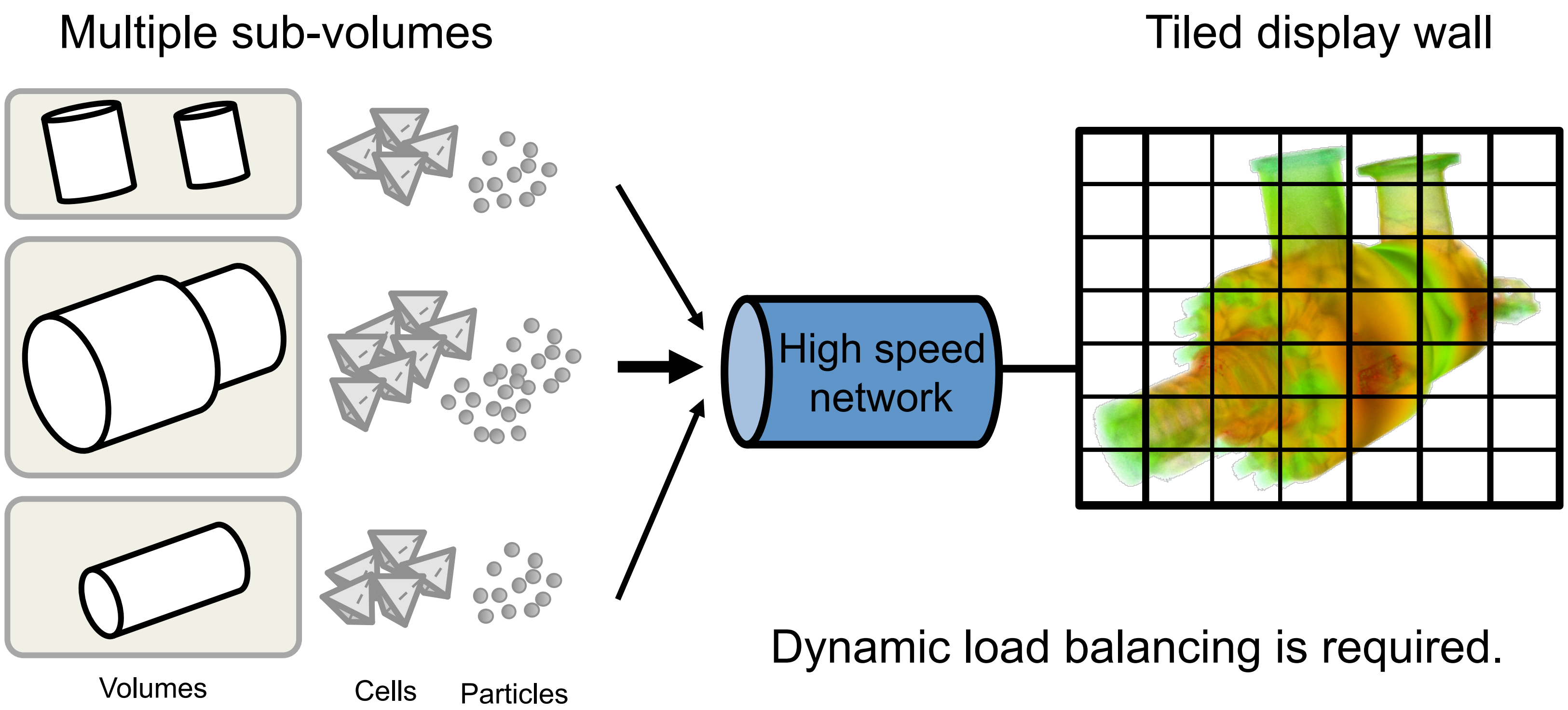
Naohisa Sakamoto, Takuma Kawamura,
Hiroshi Kuwano, Koji Koyamada

Graduate School of Engineering
Center for the Promotion of Excellence in Higher Education,
Kyoto University

Ultrascale Visualization Workshop 09
November 16, Portland, Oregon

Objective

- Large-scale irregular volume rendering
- Visualization of multiple sub-volumes on tiled display wall



Dynamic load balancing is required.

Contents

1. Related work
2. Particle-based volume rendering (PBVR)
3. Sorting-free pre-integrated projected tetrahedra (SPT)
    1. Sorting-free approach
    2. Implementation
    3. Experiment
4. Conclusion

Contents

Related Works (1/2)

- Performance improvement of visibility sorting
  - HAVS (1.3 fps for 1.4M tets, Steven et al. 2005)
  - Point-based Technique (0.3 fps for 6.3M tets, Erik et al. 2007)
- Technique without visibility sorting
  - Only absorption (Csébfalvi et al. 2003)
  - Only emission (Stefan et al. 2003)

Development of a technique without visibility sorting which considers both absorption and emission is required

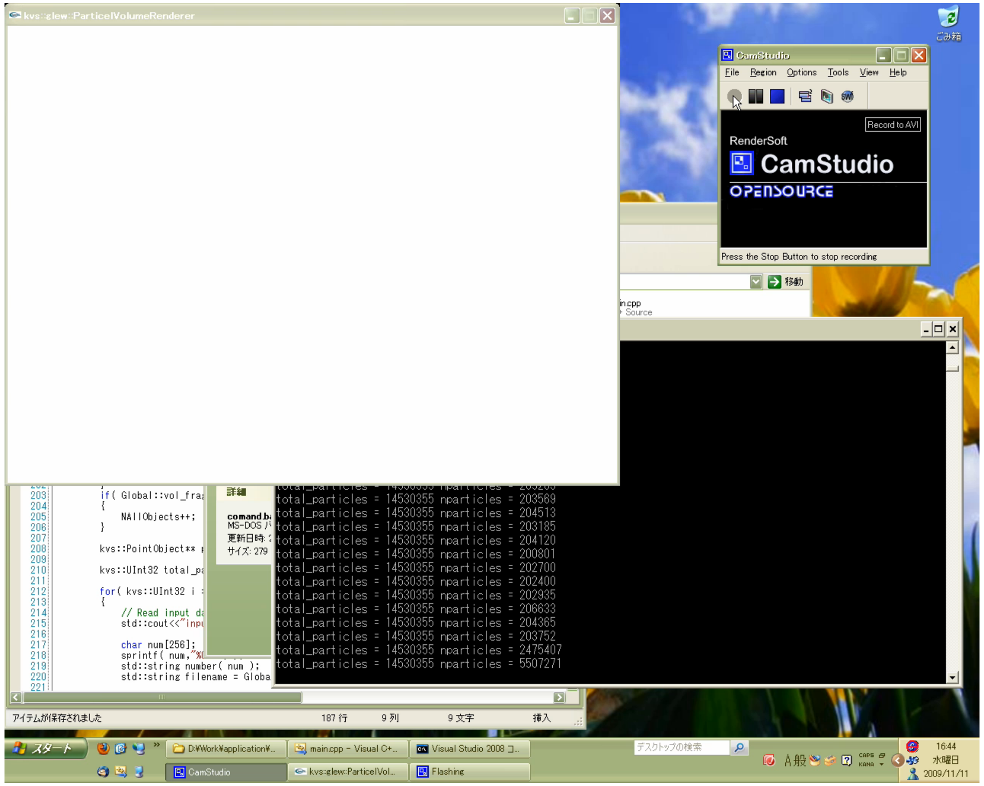
Related Works (2/2)

- Parallel volume rendering for Irregular volume
  - HPC system [Childs et. al., 2006]
    - 35 sec for 138M tets.
    - 128 procs (Opteron 2.4GHz), InfiniBand
  - PC cluster system (iRun) [Vo et. al., 2007]
    - 11.6 fps for 6.3M tets.
    - 4 PCs (Pen. D 3.0GHz, RAM 2.0GB, NVIDIA 7800 GTX)

Interactive rendering is required in order to realize our collaborative visualization environment.

Sorting-free approach

- Particle-based volume rendering [Sakamoto, et. al.,2007]
  - 7.0 fps for 208 M cells (15M particles) on a single PC (current version)

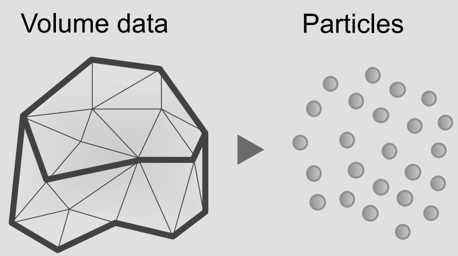


Intel Core 2 Duo 3.16GHz, 3.0GB RAM, NVIDIA GeForce 9800GT (512MB), 1280x1024 (SXGA)

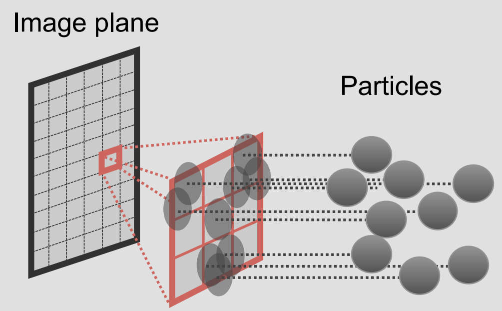
Contents

Particle-Based Volume Rendering

- Volume data is represented as particles.
  - Particle generation within the volume data
  - Particle projection into the image plane
- Visibility sorting is not required.

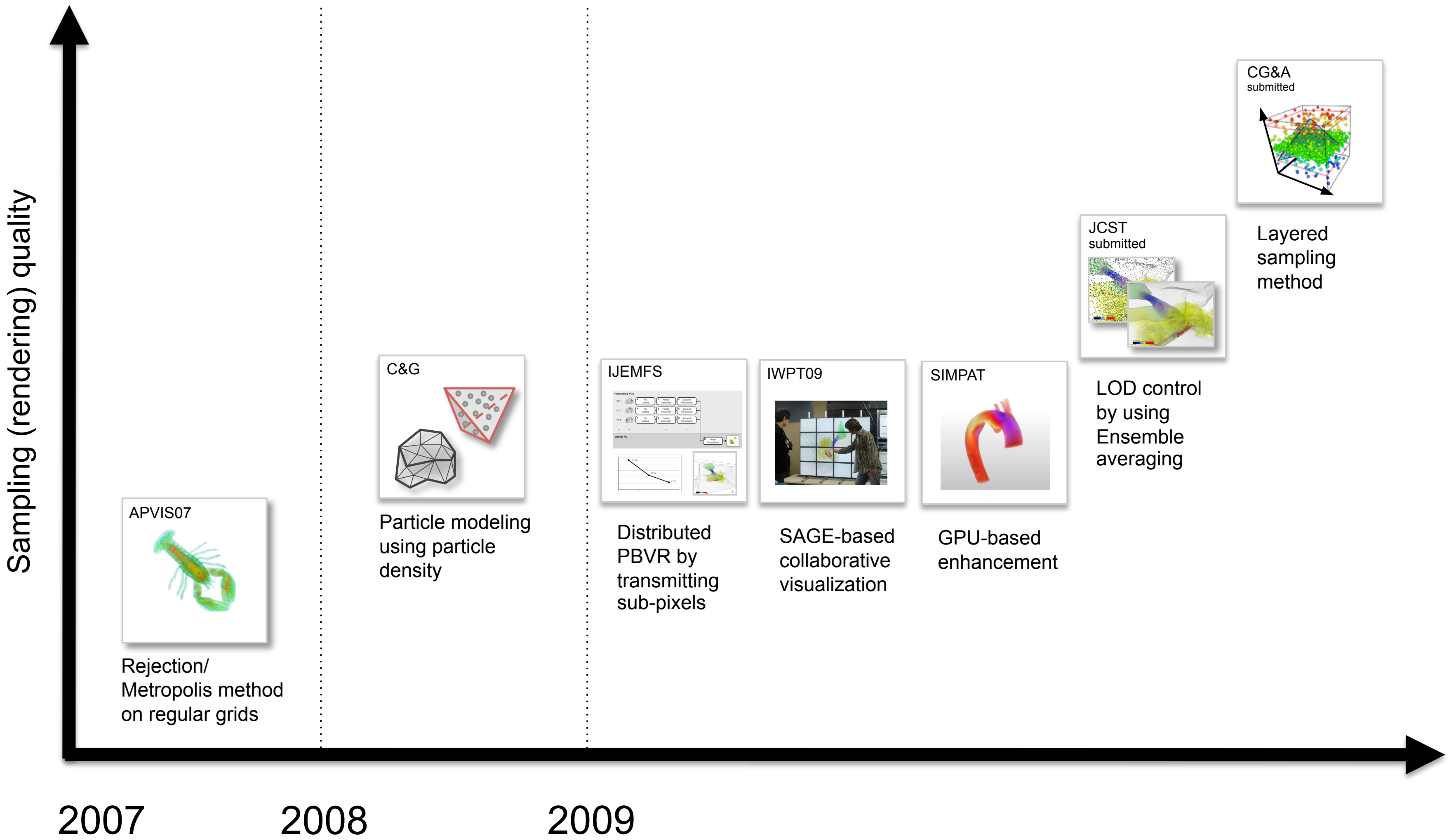


A. *Particle generation within the volume data*



B. *Particle projection onto the image plane*

Particle-Based Volume Rendering techniques (PBVR)



TOC

Sorting-free approach

- Ray casting (Brightness equation)

$$B_0 = \sum_{i=1}^{n} c_i \times (\alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j))$$

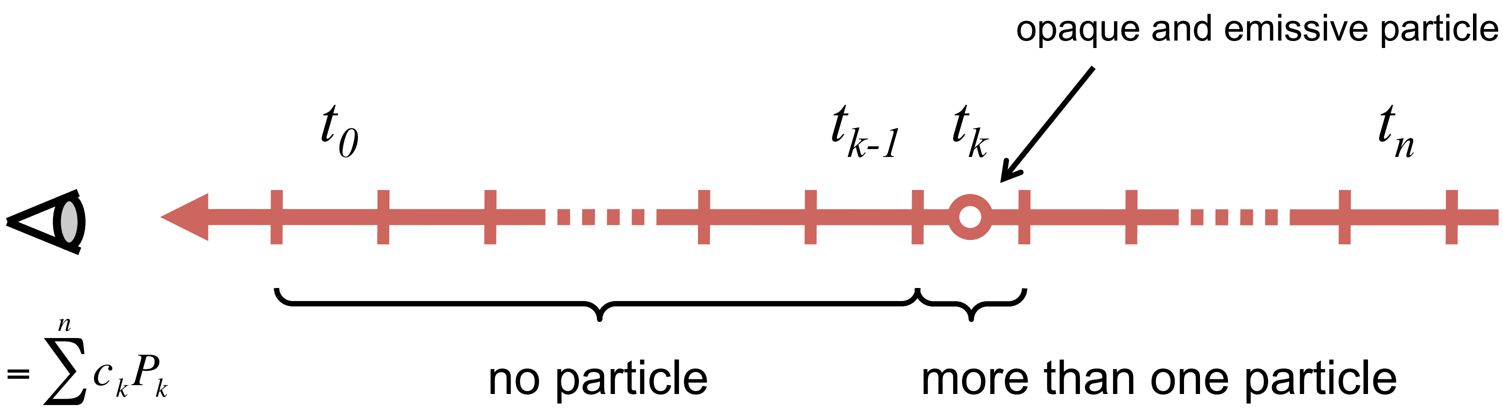
- Sorting-free approach
  - Brightness = Expected value of luminosity

$$B_0 = \sum_{k=1}^{n} c_k P_k$$

$$P_k = \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j) \quad \Rightarrow \quad \text{Probability of “} c_k = B_0 \text{”}$$

Brightness as expected value

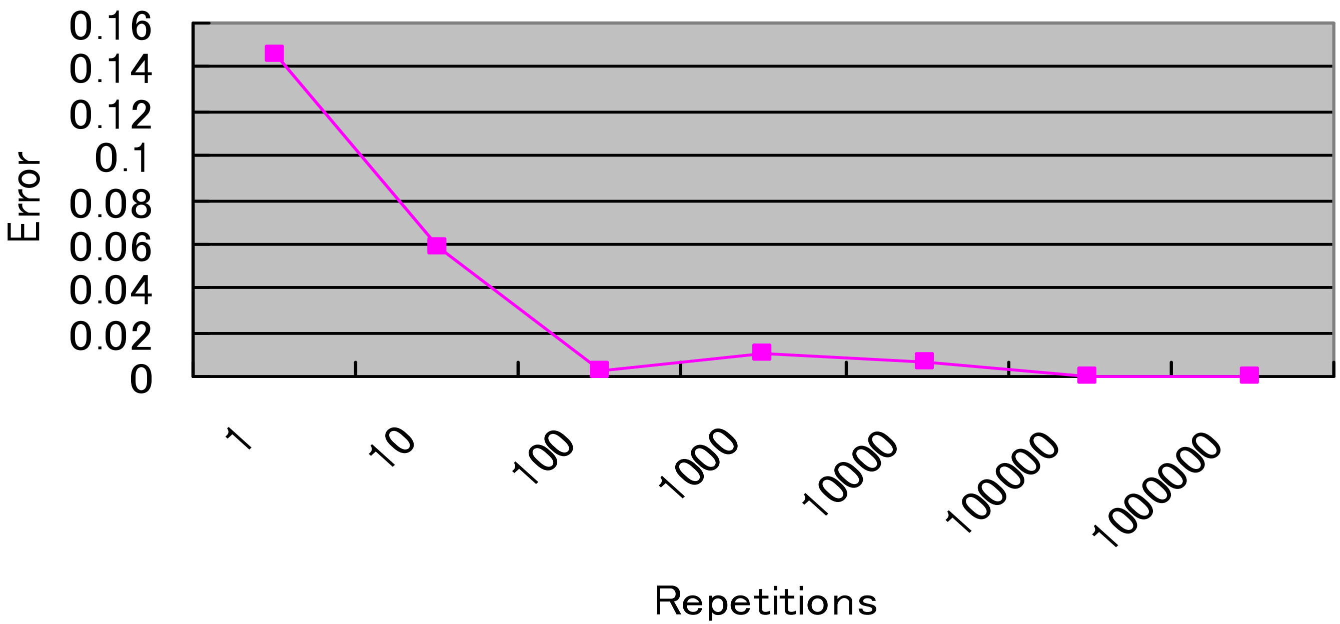
- An event where there is no particle from the first to the (k-1)-th ray segment and more than one particle in the k-th ray segment.



$$B_0 = \sum_{k=1}^{n} c_k P_k$$

Volume rendering can be approximated by repeating this events multiple times

Accuracy of Sorting-free rendering



- The error is defined as the absolute difference between the true and approximated brightness values.
- The true brightness value is calculated by generating random numbers as opacities and luminosities in all of the ray-segments

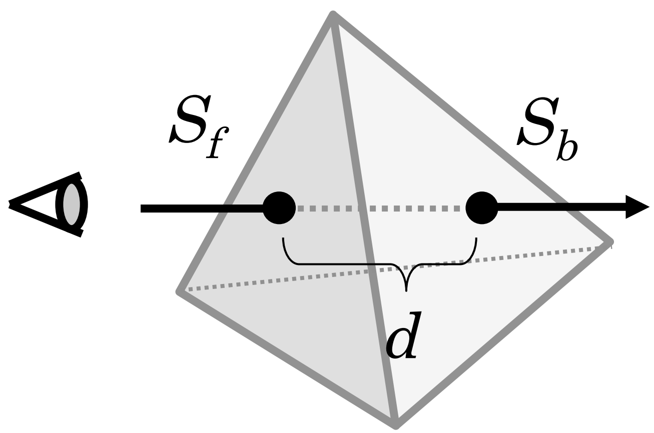
TOC

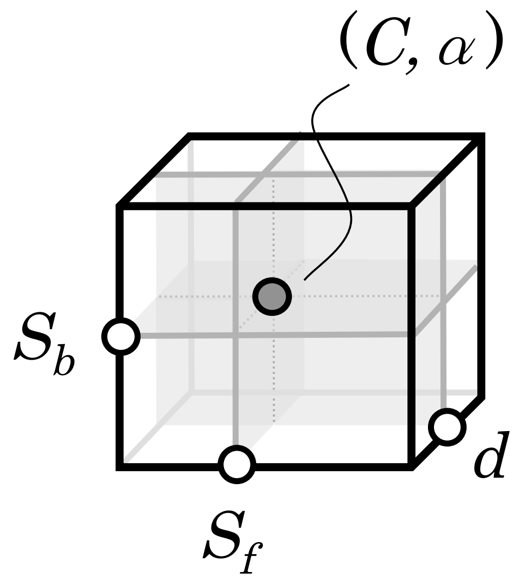
Implementation

- Sorting-free Projected Tetrahedra (SPT)
  - Pre-integration
  - Stochastic color composition

Pre-integration with 3D texture

- The pre-integration computes the lookup tables mapping three integration parameters (scalar value at the front: s_f, scalar value at the back: s_b, and length of the segment: d) to the pre-integrated color C and opacity α.

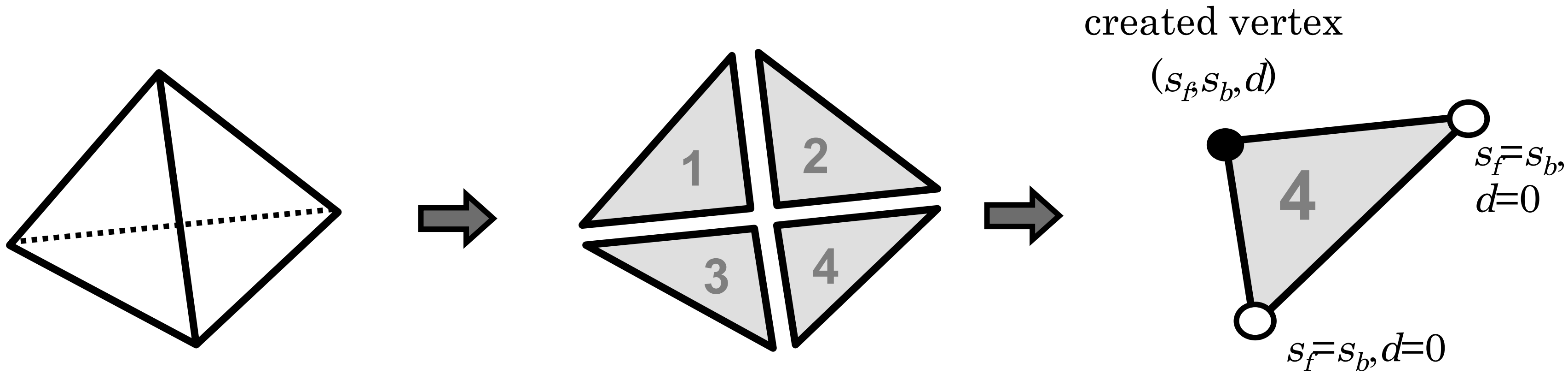




[K.Engel, et. al., 2001]

Classification and decomposition on geometry shader

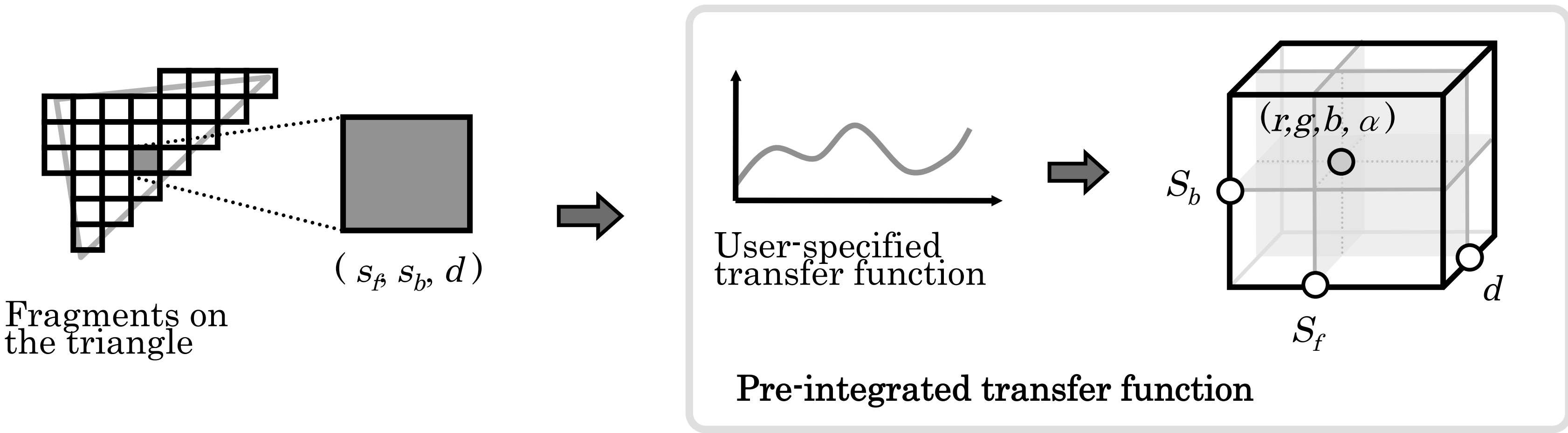
- The PT algorithm refers to the classification as splitting each of the cells into a set of triangles following four classes of projection patterns.



[P.Shirley, et. al., 1990]

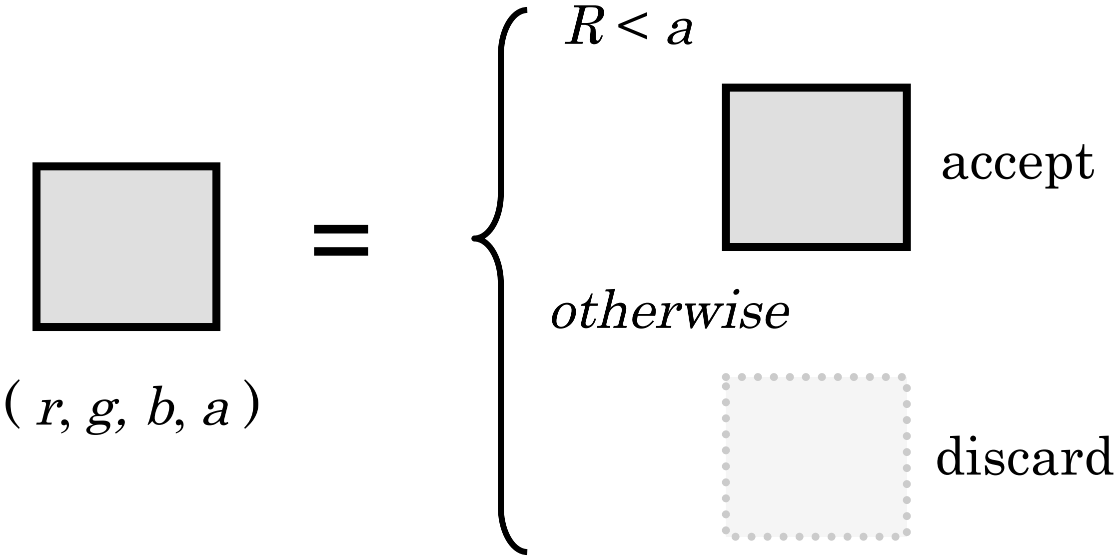
Stochastic color assignment on fragment shader (1/2)

- The stochastic color compositing is processed in the fragment shader by referring to the pre-integrated table.
- The first step is to obtain the color value of the current fragment.

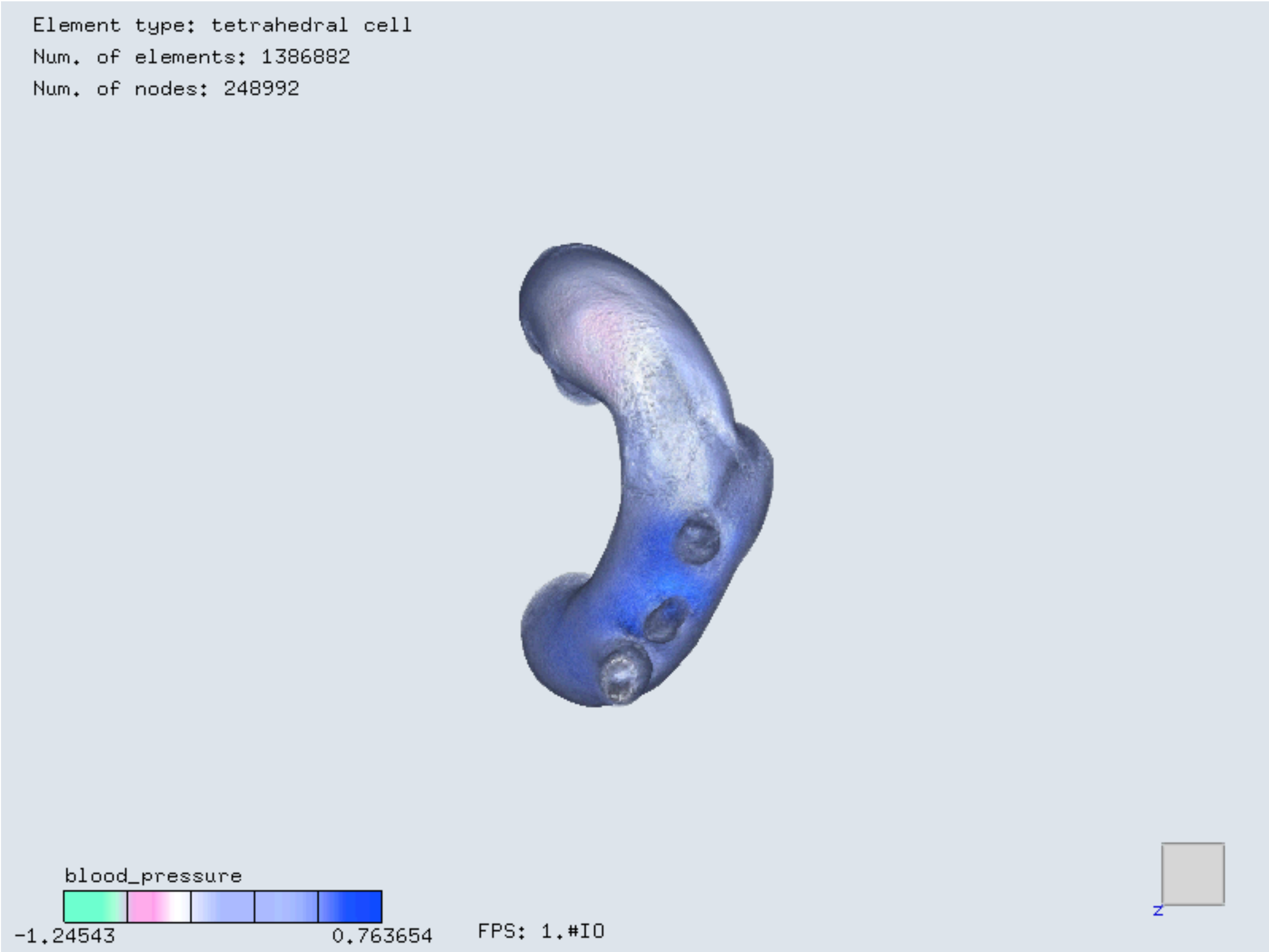


Stochastic color assignment on fragment shader (2/2)

- The next step is to decide whether this fragment should be accepted, based on its current opacity value.



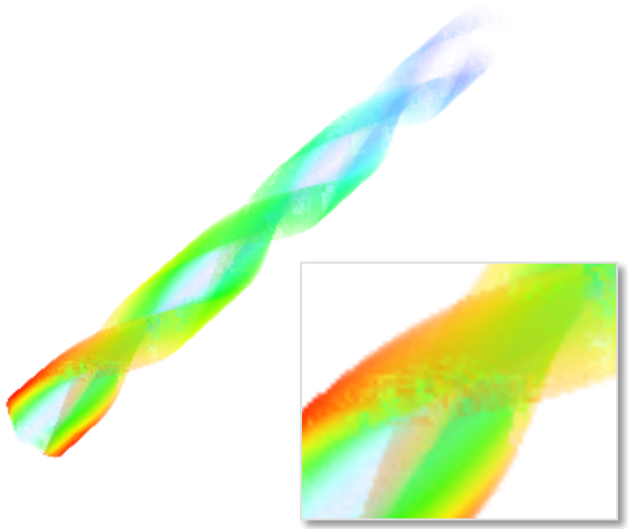
R : random number


Element type: tetrahedral cell
Num. of elements: 1386882
Num. of nodes: 248992
blood_pressure
-1.24543
0.763654
FPS: 1.#IO
z

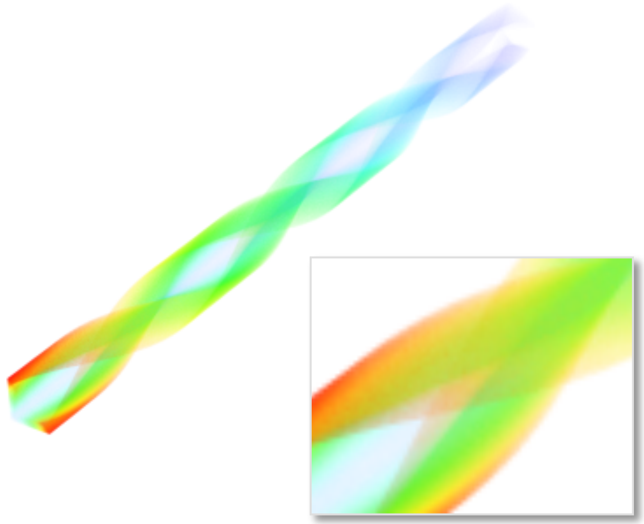
TOC

Influence of visibility sorting

- Comparison of rendering quality with and without visibility sorting (Drill data, 9.9M tets)



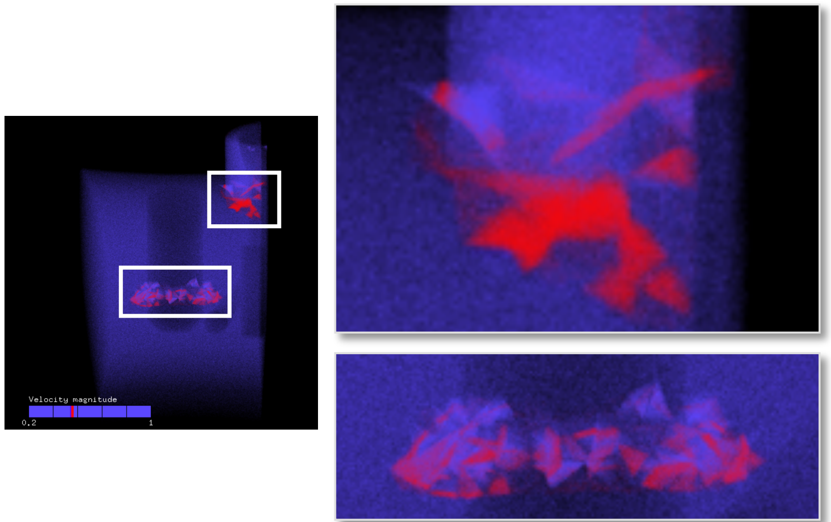
HAVS (with sorting)



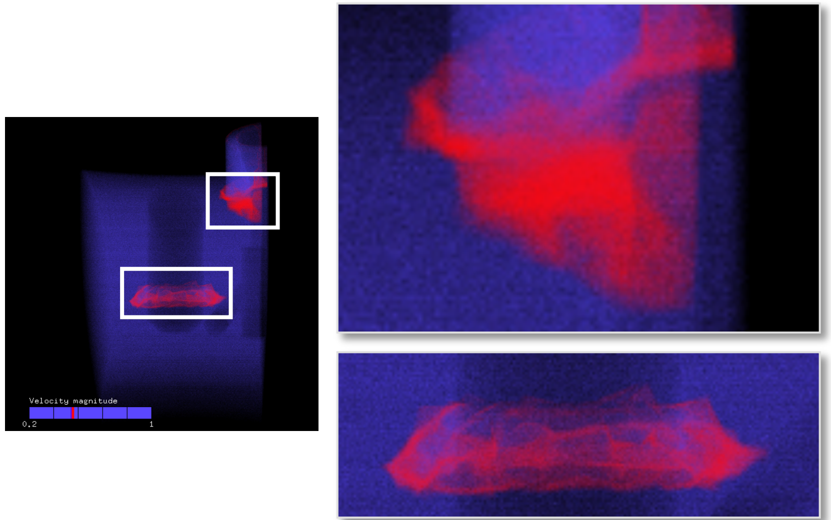
SPT (without sorting)

PBVR vs. SPT

- Difference between the quality of images rendered by PBVR and SPT when the transfer function varies precipitously.
- The opacity value of the transfer function drastically increases in the red colored value range.



PBVR (repetitions: 144)



SPT(repetitions: 144)

Performance evaluation (1/2)

- Performance model (*T*: computing time [ms], *M*: GPU memory [byte])
  - SPT (nc: # of cells, Lr: repetition level, Ls: subpixel level, wxh: image resolution)

$$
\begin{aligned}
T^{SPT} &= L_r\left(5.67\times 10^{-5} n_c + 4.51\times 10^{-6} whL_s^{\,2}\right) \\
M^{SPT} &= 20.6 n_c + 7.1 whL_s^{\,2}
\end{aligned}
$$

  - PBVR (np: # of particles, Lr: repetition level, Ls: subpixel level, wxh: image resolution)

$$
\begin{aligned}
T^{PBVR} &= L_r\left(9.96\times 10^{-7} n_p + 1.47\times 10^{-5} whL_s^{\,2}\right) \\
M^{PBVR} &= 22.0 n_p + 6.86 whL_s^{\,2}
\end{aligned}
$$

  - HAVS (nc: # of cells, k:k-buffer size, wxh: image resolution)

$$
\begin{aligned}
T^{HAVS} &= 1.88\times 10^{-6} n_c + 1.71\times 10^{-6} wh + 0.275k \\
M^{HAVS} &= 37.65 n_c + 76.3 wh
\end{aligned}
$$

Performance evaluation (2/2)

- Computational time
  - HAVS < SPT, PBVR
- GPU memory resources
  - HAVS > SPT, PBVR
- SPT vs. PBVR (512GB VRAM, 1024x1024)
  - if # of tets. < 24.5M SPT
  - otherwise PBVR

Rendering result of large-scale volume dataset

- 13M tetrahedral cells
  - Image resolution: 1024x1024
  - Repetition level: 144

	SPT	PBVR*
Rendering time [sec]	3.1	N/A

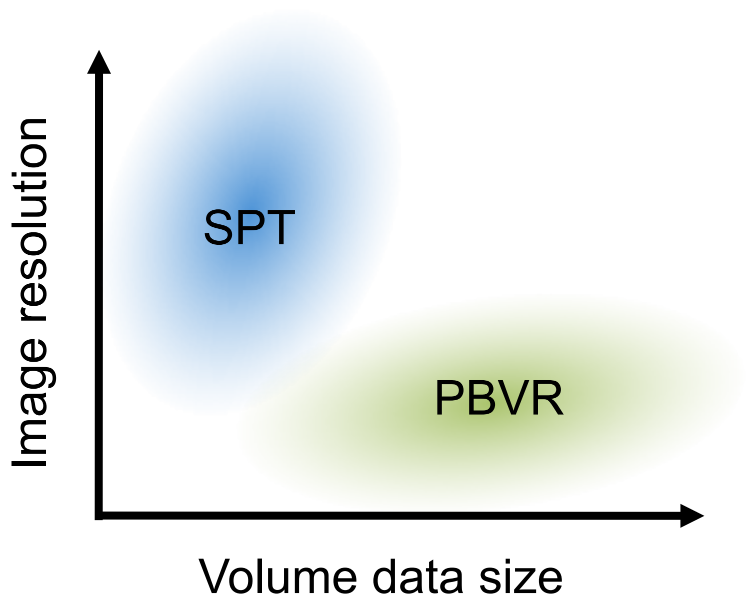
* 2.8G particles = 71GB VRAM

Intel Core 2 Quad 2.83 GHz, 8 GB RAM, Nvidia GeForce GTX 280 1.5GB VRAM

TOC

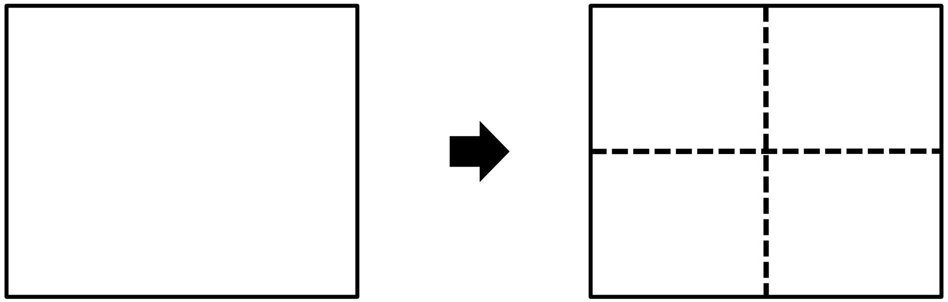
Conclusion

- Sorting-free Projected Tetrahedra (SPT)
  - NOT particle-based technique
  - Suitable for high resolution rendering
  - Easy implementation to parallelize



Future work

- Acceleration the rendering process by subdividing frame buffer



- Dynamic load balancing on distributed environment