Particle-Based Rendering and its applications

2014/3/10

Kyoto university

Koji KOYAMADA

Shonan meeting 046

Contents

- Background
- Applications
- Rendering and particles
- Summary

Particle-Based Rendering and its applications

BACKGROUND

Big Data

Source: IDC's Digital Universe Study, sponsored by EMC, June 2011

年間1.8ゼタバイトのデータ生成

Supercomputer

2010

Tianhe-1A, NVIDIA

2011

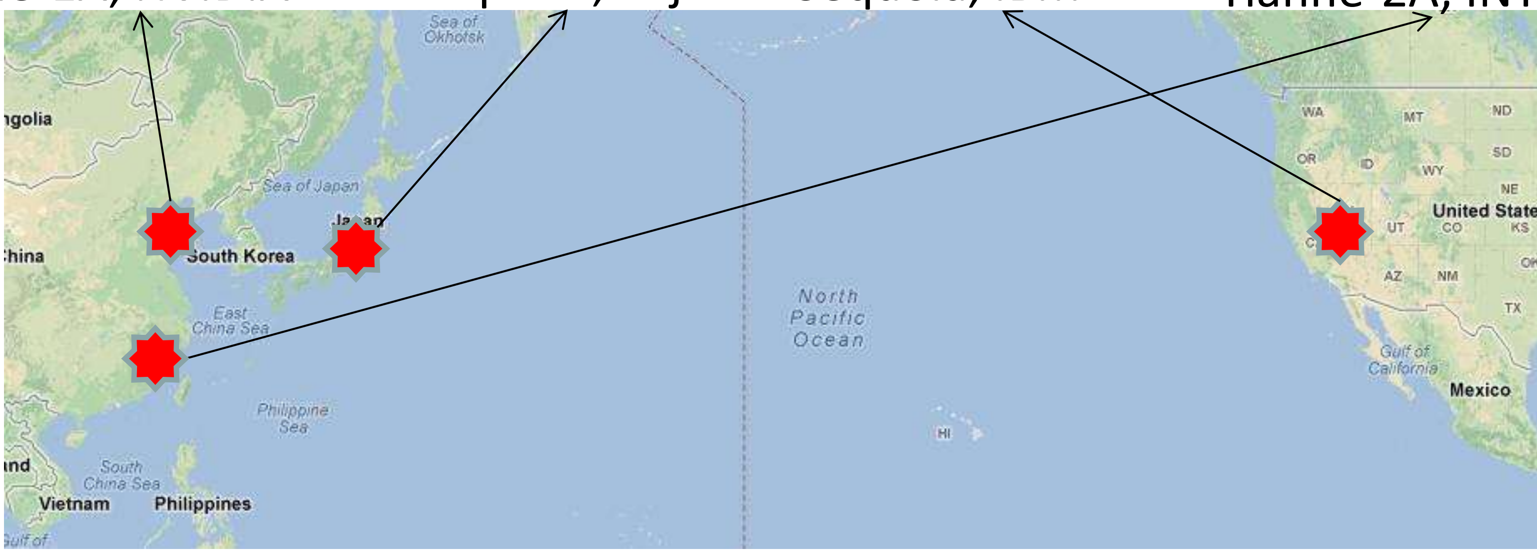
K computer, Fujitsu

2012

Sequoia, IBM

2013

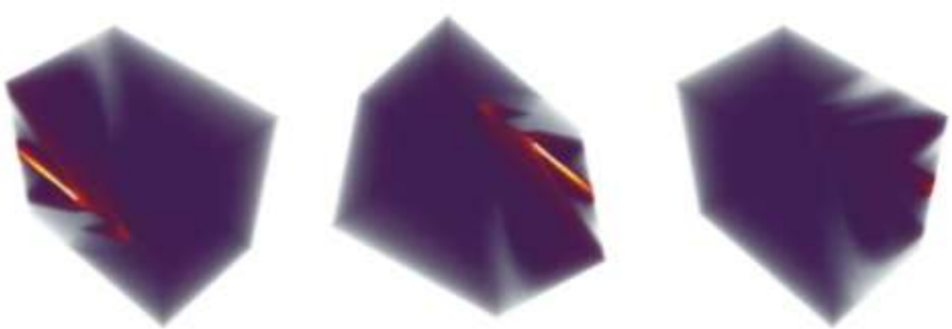
Tianhe-2A, INTEL



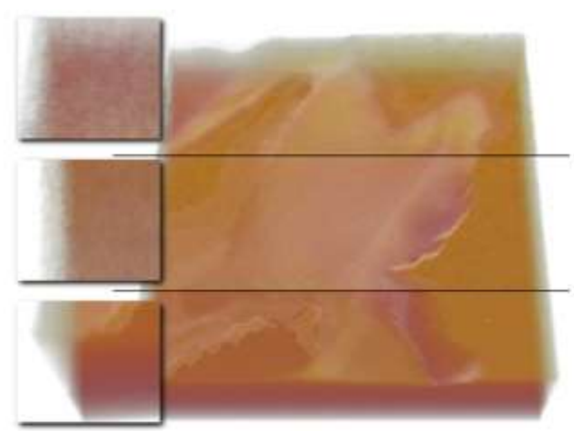
Requirement for efficient volume rendering to shorten the time to inspiration

Related work (1/2)

- Performance improvement of visibility sorting
  - HAVS [Steven et. al., 2005]
    - 1.3 fps for 1.4M tets.
  - Point-based technique [Erik.W et. al., 2007]
    - 0.3 fps for 6.3M tets.



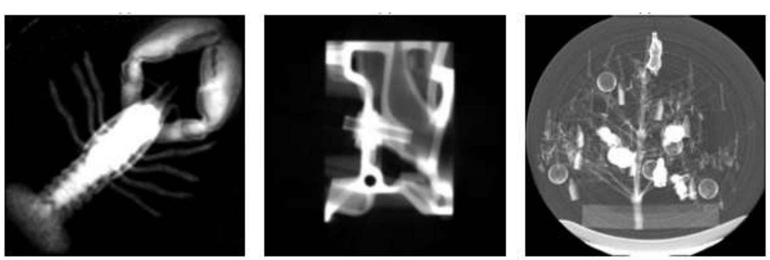
HAVS



Point-based technique

Related work(2/2)

- Sort-free algorithm by model simplification
  - Only absorption
    - Monte Carlo Volume Rendering [Cselfalvi et. al., 2003]
      - 1.4 fps for 16M points (for regular volumes)
  - Only emission
    - Point-based rendering [Zhou et. al., 2006]
      - 20 fps for 17.6M tets.



Monte Carlo Volume Rendering

Point-based rendering

Comparison

- Parameters
  - r: Radius of a particle
  - N_c: Number of cells

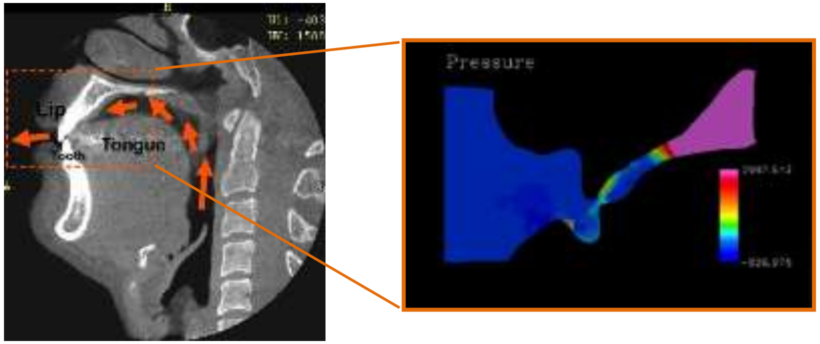
Method	**Number of primitives**	**Visibility sorting**
Particle-based	$O(1/r^2)$	Not required
Point-based	$O(N_c)$	Required

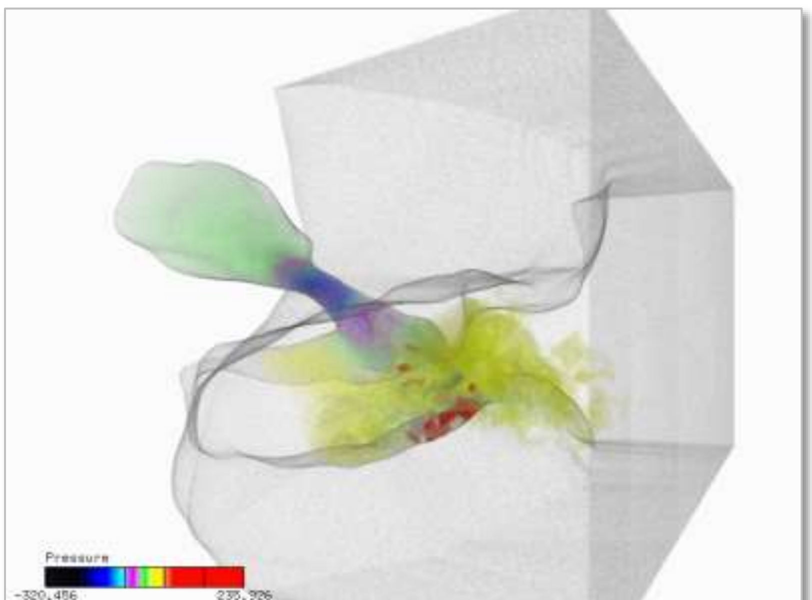
Particle-Based Rendering and its applications

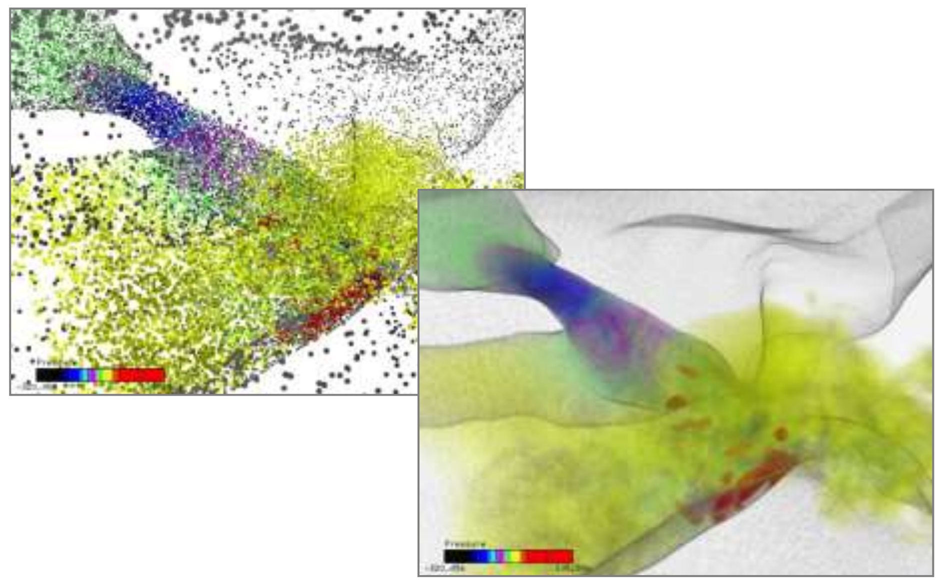
APPLICATIONS

Oral Airflow Simulation

- Cell type: hexahedron
- # of cells: 71 M
- Data size: 3.23 [Gbytes]
- # of particles: 0.73 M
- Particle generation time: 22.0 [sec]



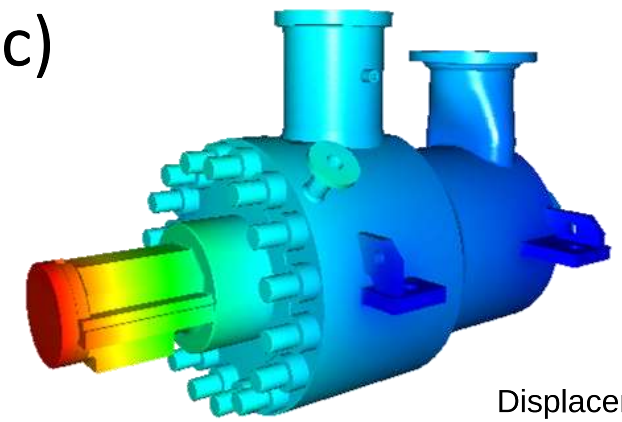




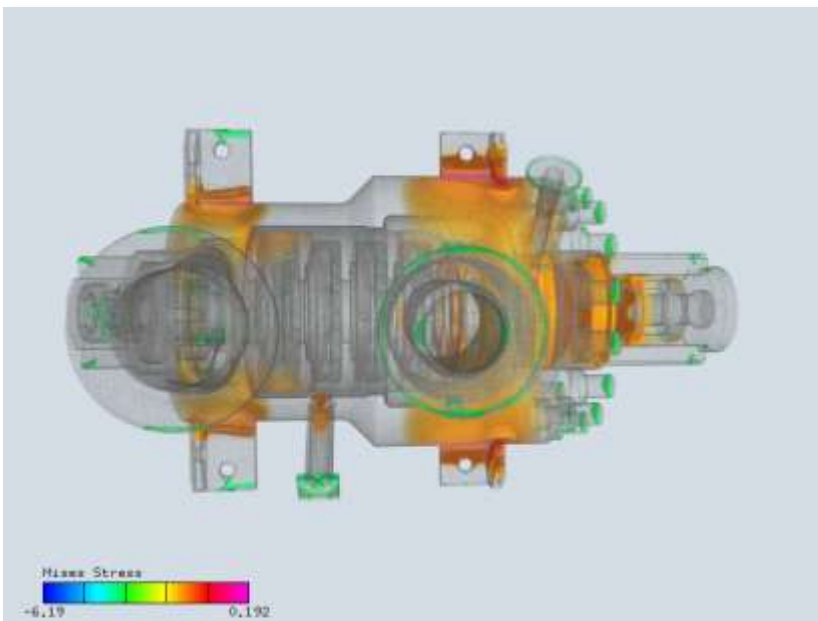
Naohisa Sakamoto, Takuma Kawamura, Koji Koyamada, Kazunori Nozaki: Improvement of particle-based volume rendering for visualizing irregular volume data sets. Computers & Graphics 34(1): 34-42 (2010)y

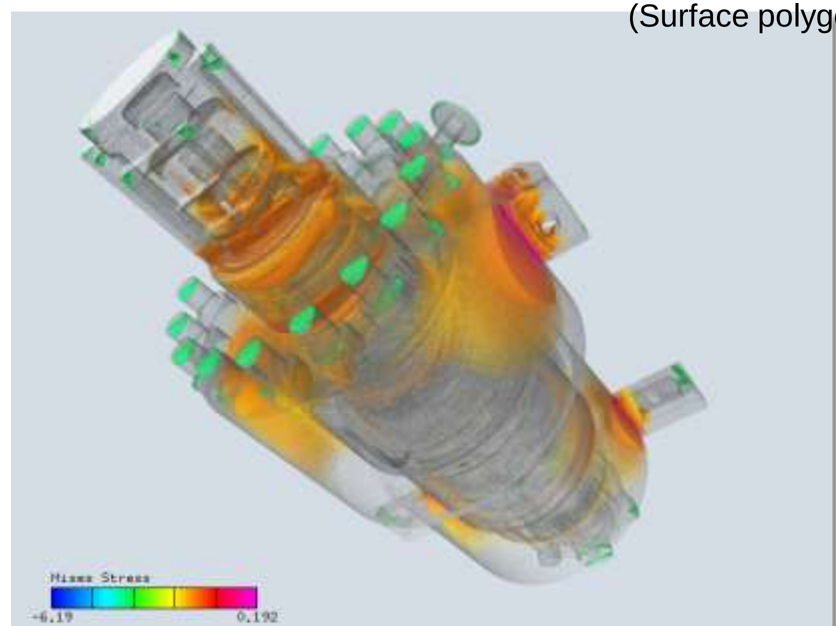
Structural Analysis on a Pump

- Cell type: tetrahedron (quadratic)
- # of cells: 26 M
- Data size: 1.93 [Gbytes]
- # of particles: 10 M



Displacement
(Surface polygon)

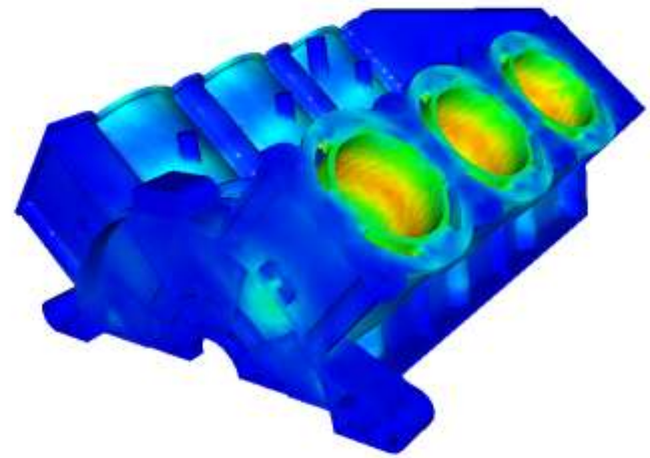


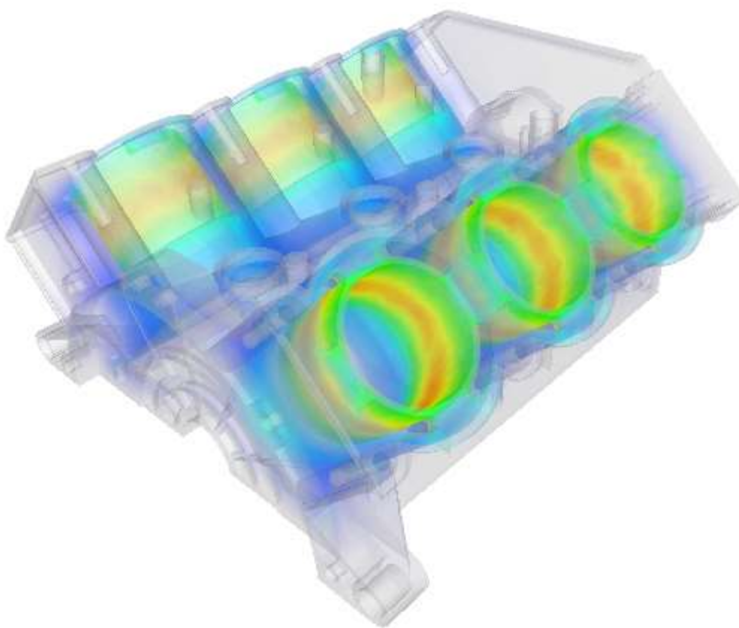


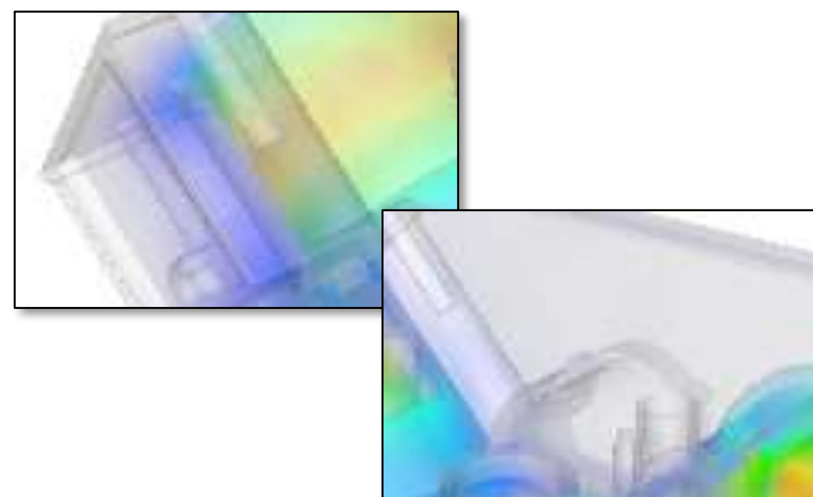
Naohisa Sakamoto, Hiroshi Kuwano, Takuma Kawamura, Koji Koyamada, Kazunori Nozaki,"Visualization of Large-scale CFD Simulation Results Using Distributed Particle-Based Volume Rendering",International Journal of Emerging Multidisciplinary Fluid Sciences, Vol.2, No.2, pp73-86,2010,

V6 engine data

- Cell type: tetrahedron
- # of cells: 282K
- Data size: 170 [Mbytes]

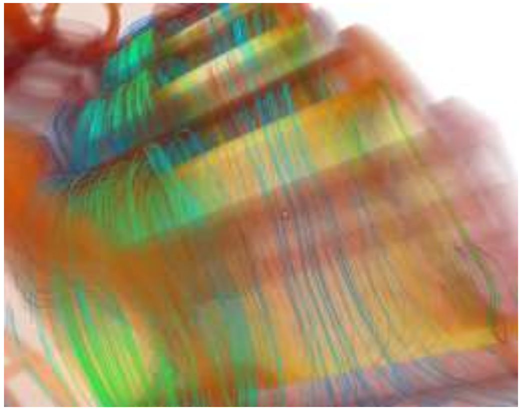


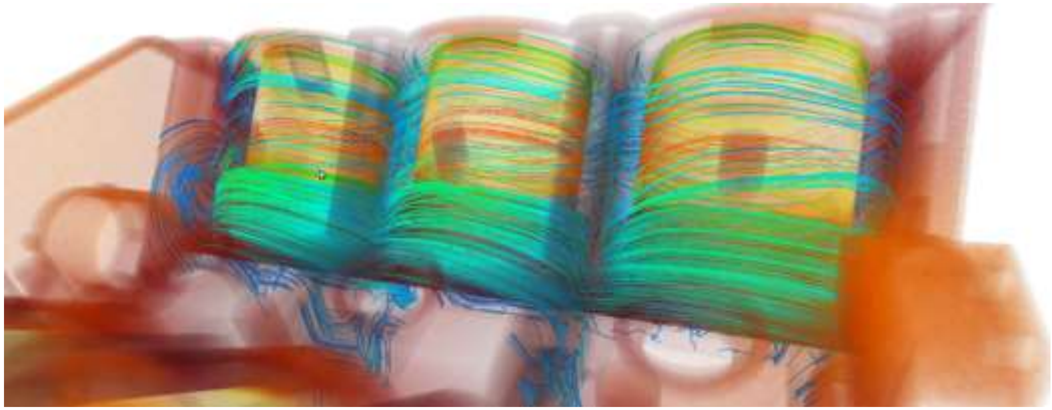




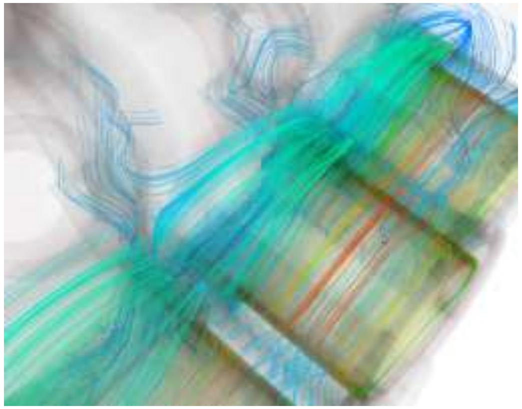
Volumes and hyper streamlines

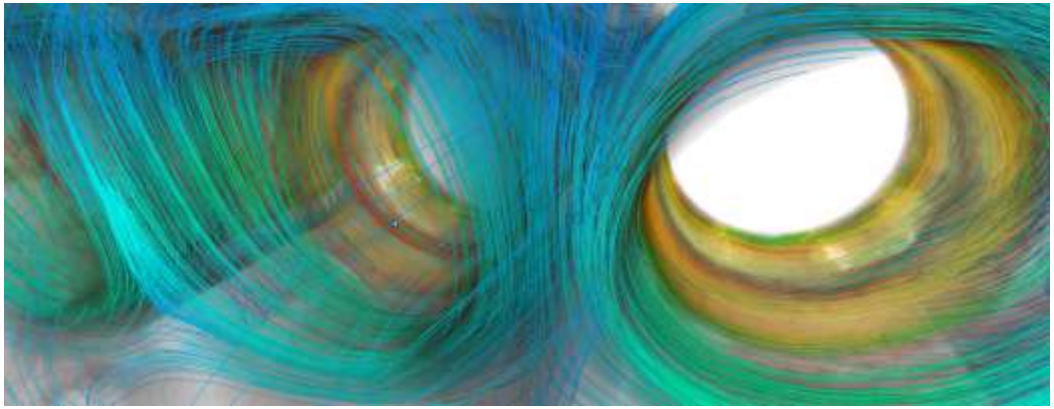
- Displacement (volume) and stress tensor (streamlines)





- Misses stress (volume) and stress tensor (streamlines)

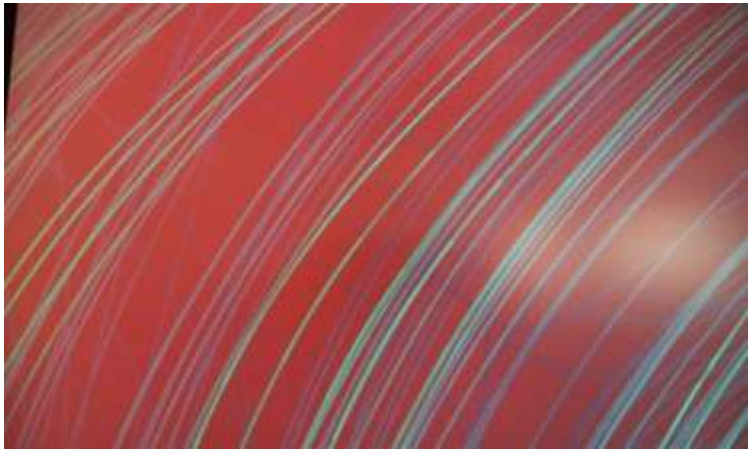




Rendering results on 40 LCDs

- Volume + boundary surfaces + hyper streamlines

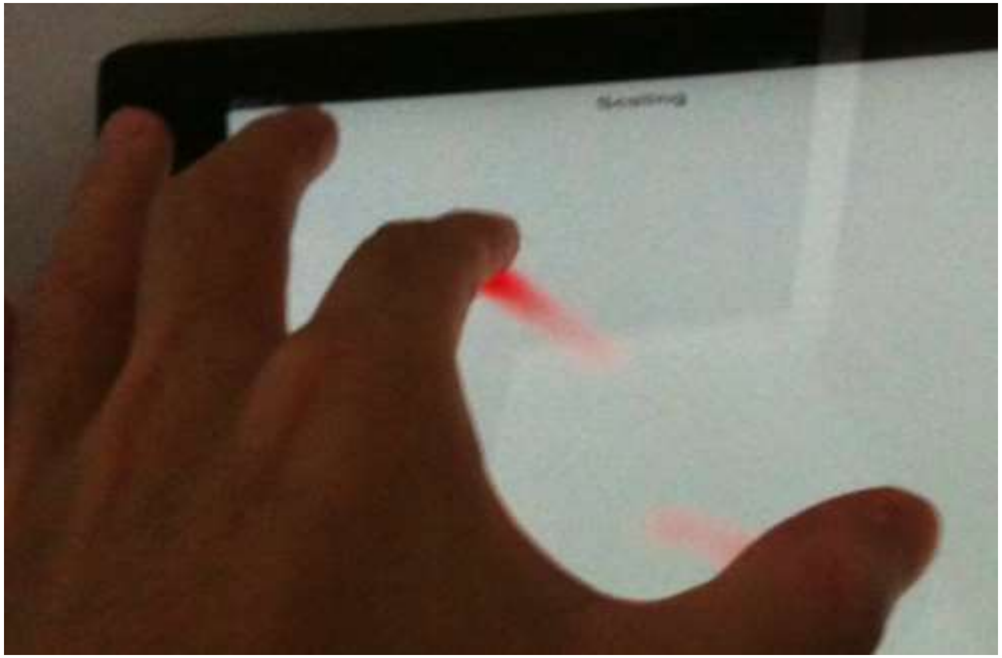


N.Sakamoto, J.Nishimura, K.Koyamada,"Stochastic Approach for Integrated Rendering of Volumes and Semi-transparent polygons on a High-resolution Display System(poster)",The 5th IEEE PacificVis Symposium,2012, Best Poster Award

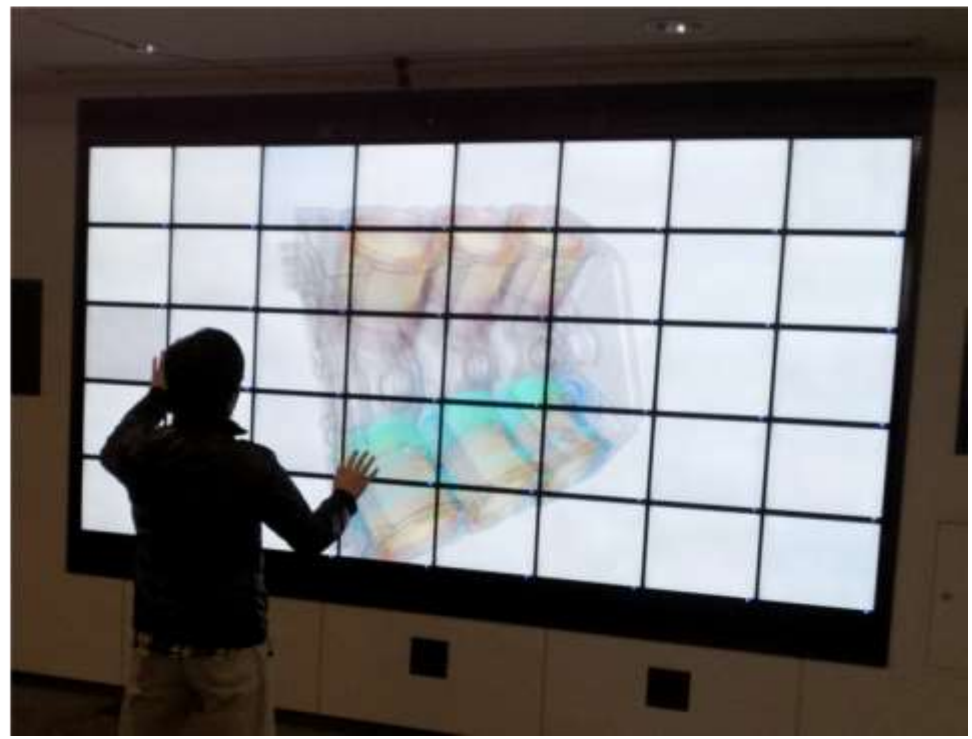
Multi-touch interface

- Using tablet PC
- Multi-touch operation



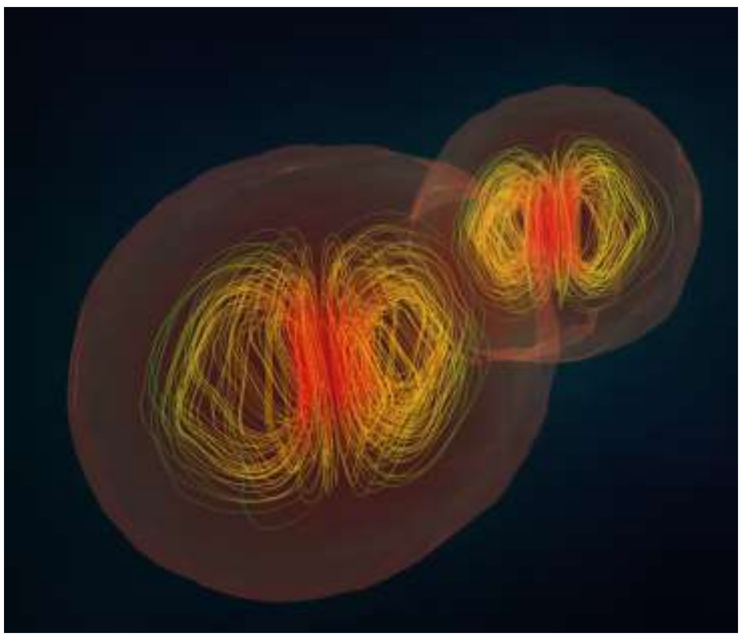
Gesture interface

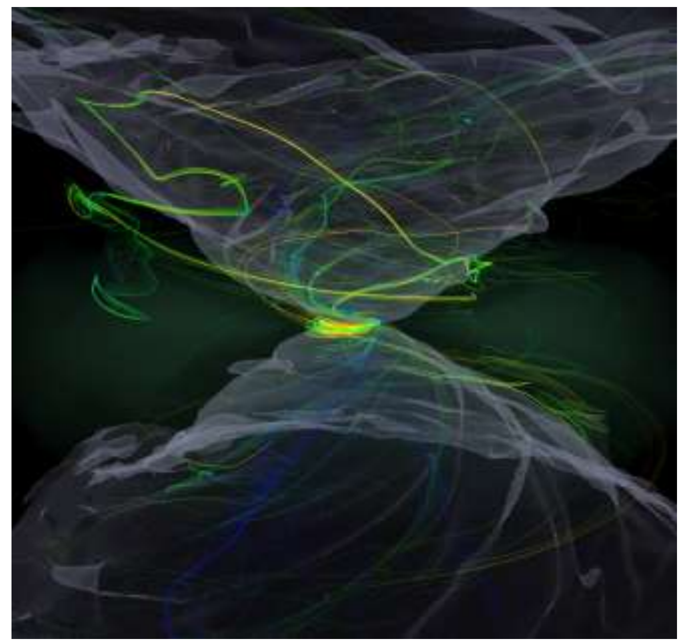
- Using stereo camera (Microsoft Kinect)
- Gesture operation



Visualization of astrophysics data

- Magnetic field and mass density around a simulated black hole which is generated by a collision of binary neutron star merger.

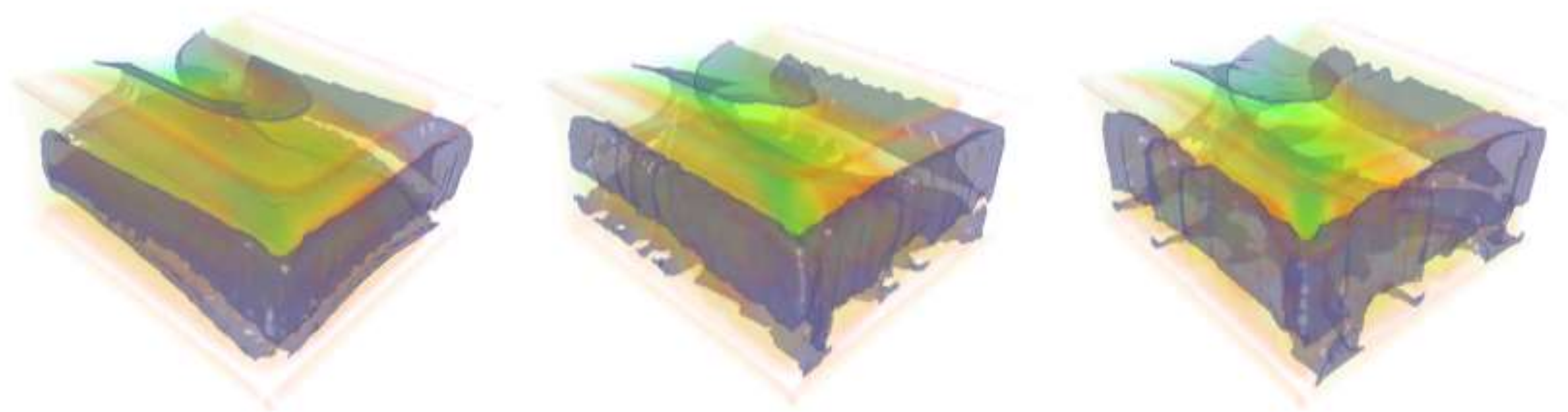




Data courtesy of Prof. Shibata, Dr. Kiuchi, Dr. Sekiguchi, Kyoto Univsersity

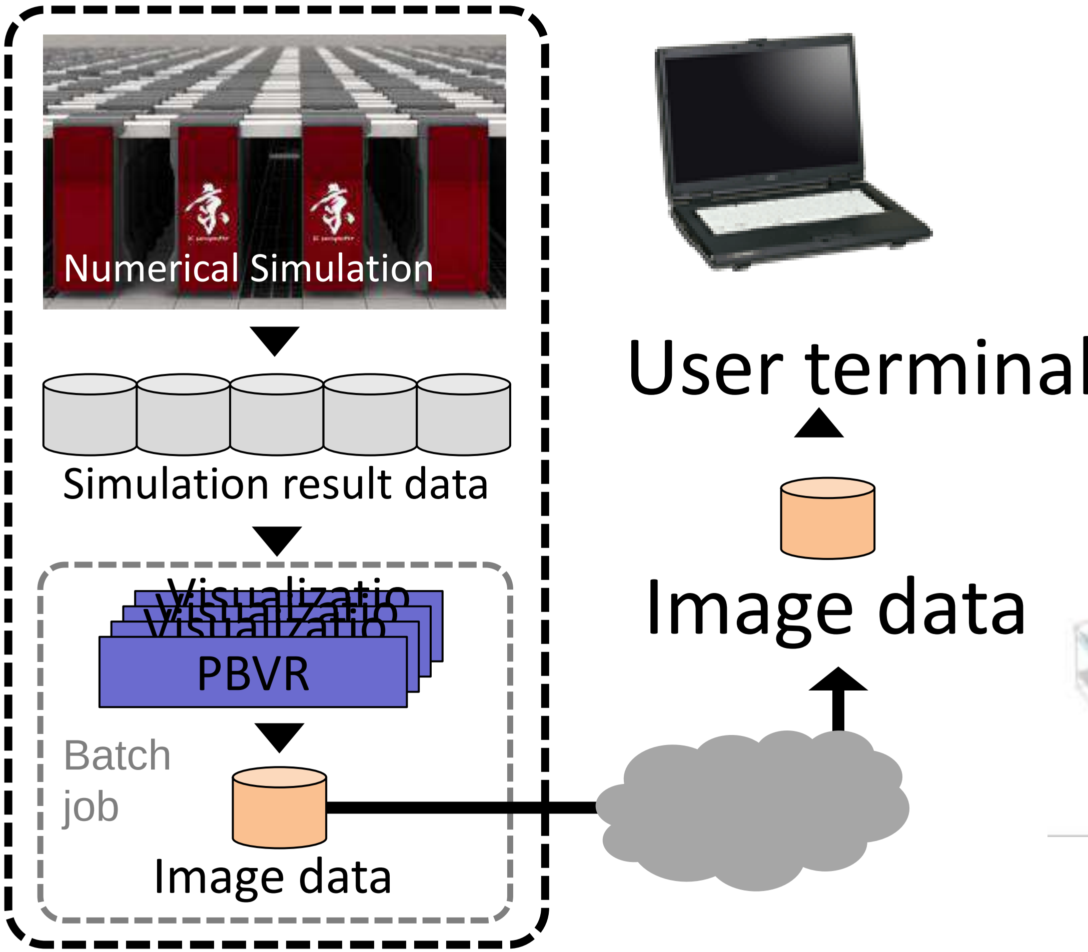
Visualization of fluid dynamics data

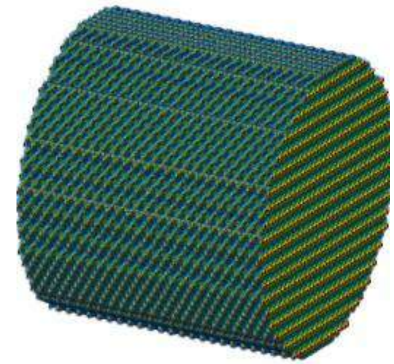
- Immiscible multi-phase fluid in a tank with vertical moving wall by MICS (Multiphase Incompressible flow solver with Collocated grid System)



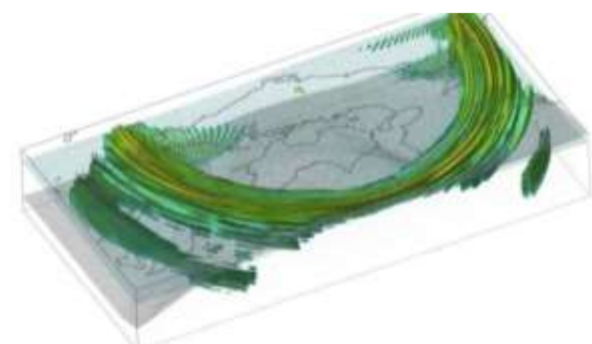
Data courtesy of Prof. Ushijima, Mr. Fujioka, Kyoto Univsersity

Parallel PBR on K computer





Visualization result of silicon nanowire (RSDFT)
RIKEN AICS and Univ. of Tokyo

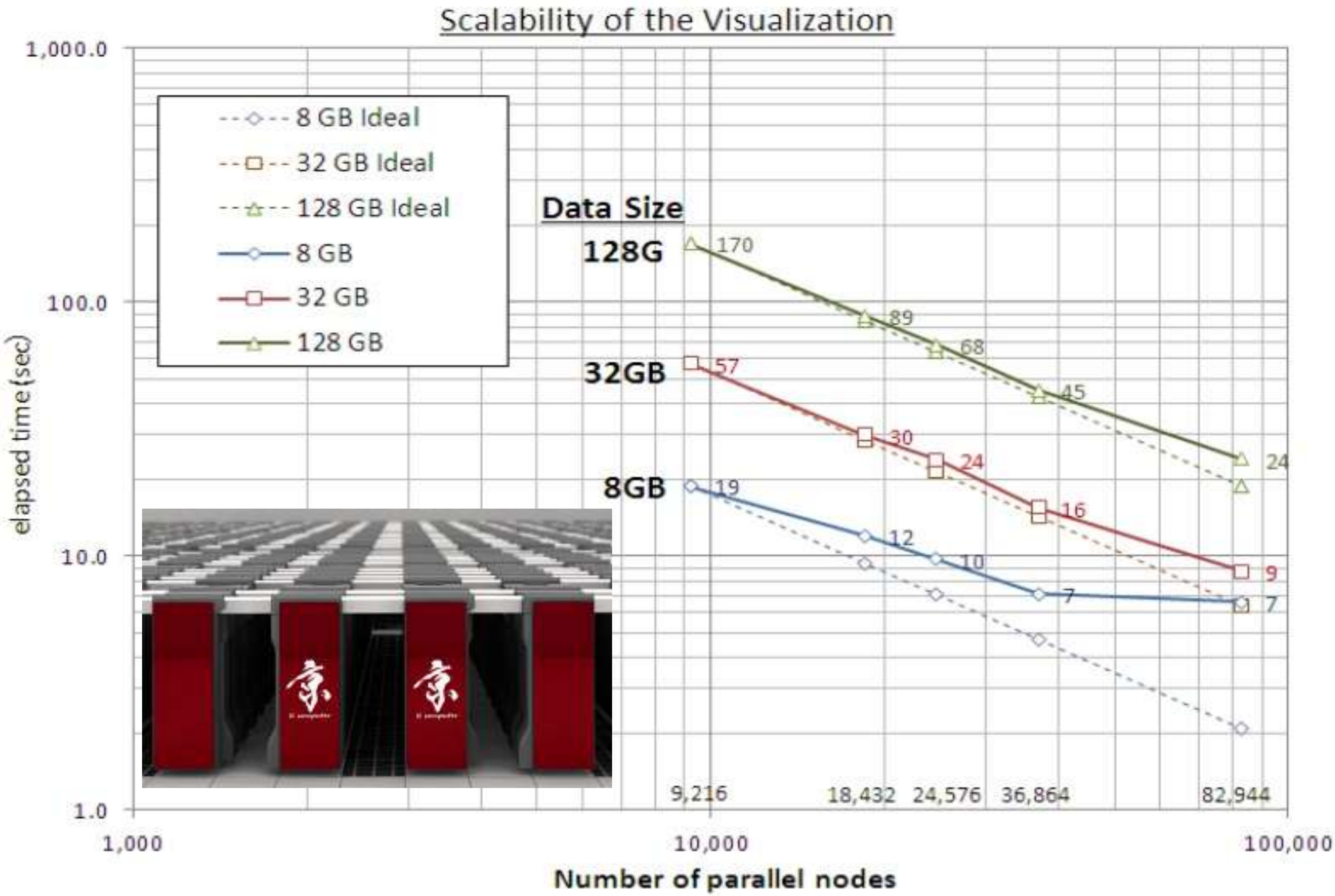


Visualization result of seismic wave
Earthquake Research Institute, Univ. of Tokyo

Ogasa et al., "Visualization technology for the K computer", Fujitsu Scientific & Technical Journal, Vol.48, No.3, 2012.

Visualization in K-computer system

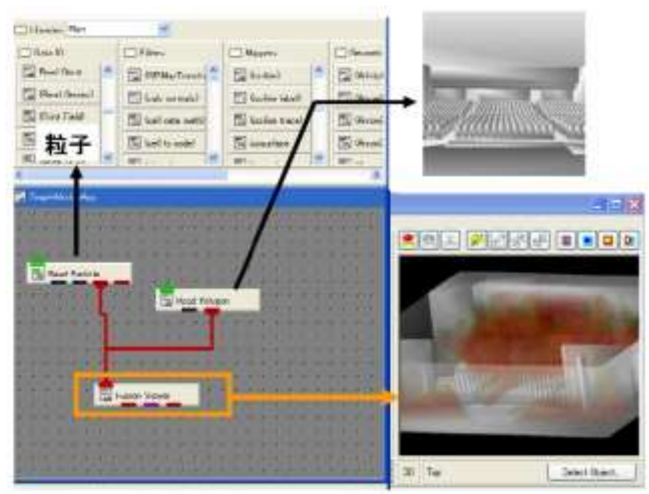
Fujitsu Sci. Tech. Journal, Vol.38 No.4, 2012



Technology transfer to AVS/Express

http://www.avs.com/solutions/express/

Since its first release in 1995, AVS/Express has been selected by thousands of corporations, solution providers, academic institutions and government agencies to provide powerful insight to nearly one million end-users in an array of industries.

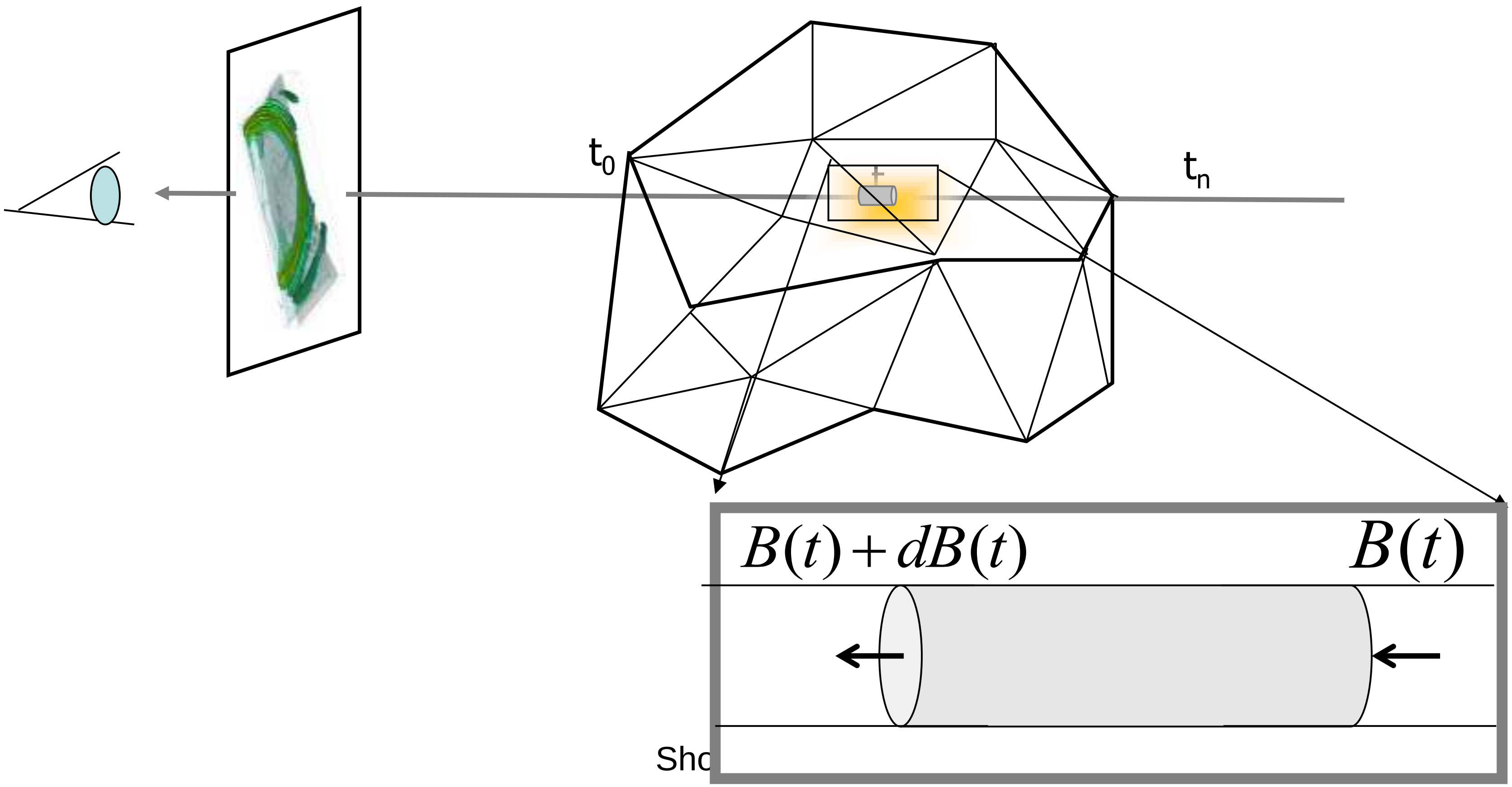


Particle-Based Rendering and its applications

RENDERING AND PARTICLES

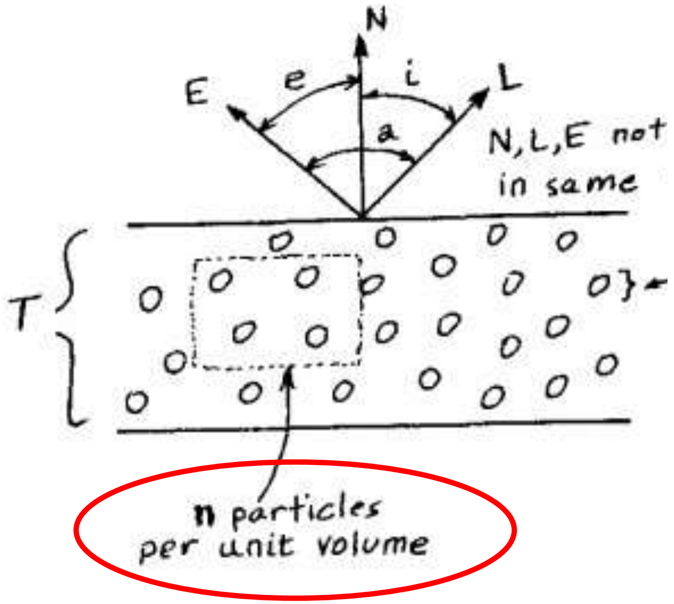
What is volume rendering?

- Technique used to display a 2D projection of a 3D discretely sampled data set.



Volume rendering and particles

- The number of particles follow the poisson process



All other particles are assumed to be of radius p also. For such a particle to be completely outside the cylinder its center must be outside this volume. Statistically, then, the attenuation of light traversing a cylinder of radius p and volume V is P(0;V) = the probability of 0 particles in volume V. Now the expected number of particles in a given volume V is nV. If n is small this can be modelled as a Poisson process and

$$P(0;V) = \exp(-nV)$$

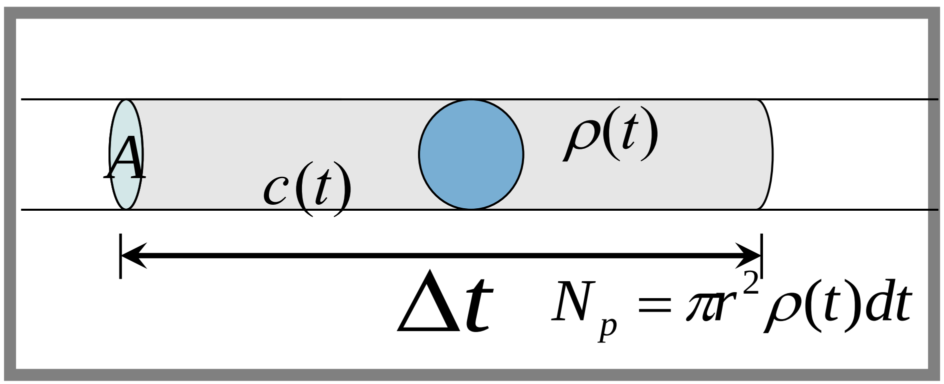
J. F. Blinn, "LIGHT REFLECTION FUNCTIONS FOR SIMULATION OF CLOUDS AND DUSTY SURFACES,"

Opacity definition

- Probability that there are more than one particles

$$P(N = k) = \frac{\exp(-N_p) N_p{}^k}{k!}$$

$$\alpha = 1 - \exp(-N_p)$$



$$\rho = -\frac{\log(1-\alpha)}{\pi r^2 \Delta t}$$

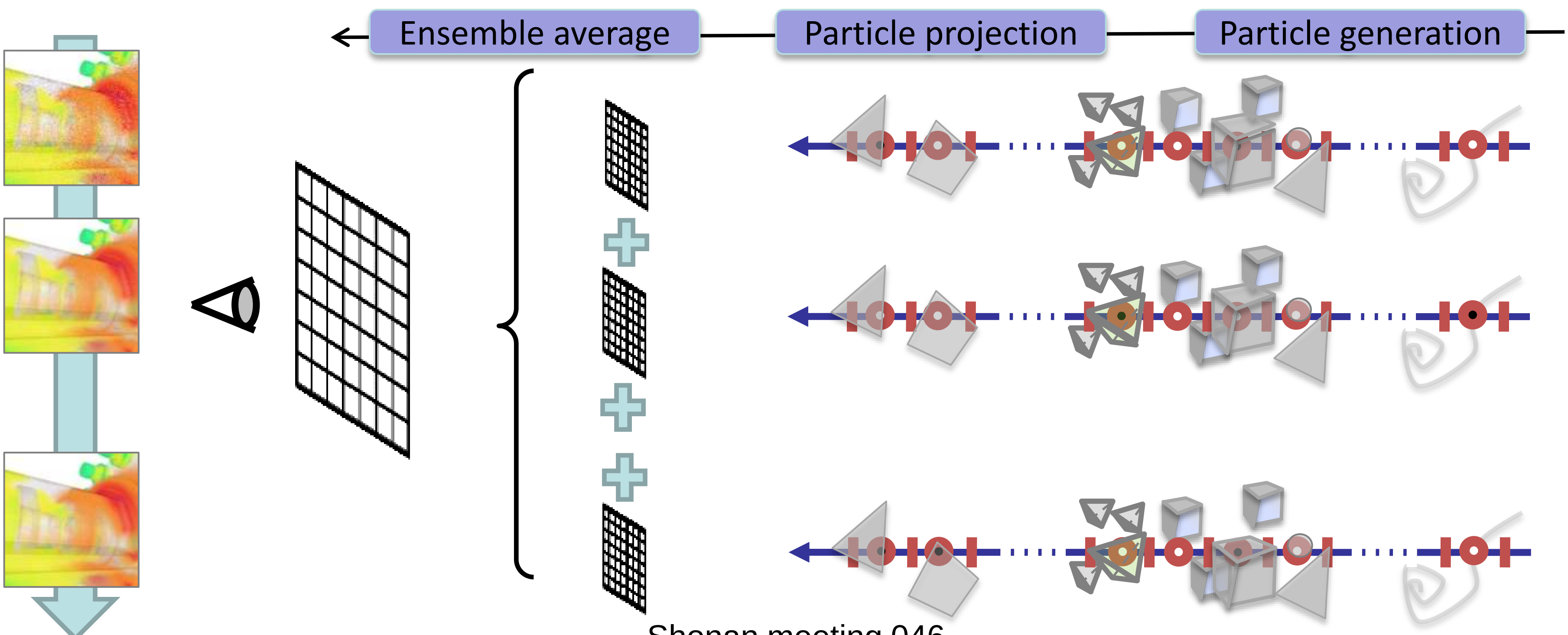
$$P(N = 0) = \exp(-N_p)$$

Particle-Based Rendering and its applications

PARTICLE-BASED RENDERING

Particle-based rendering (PBR)

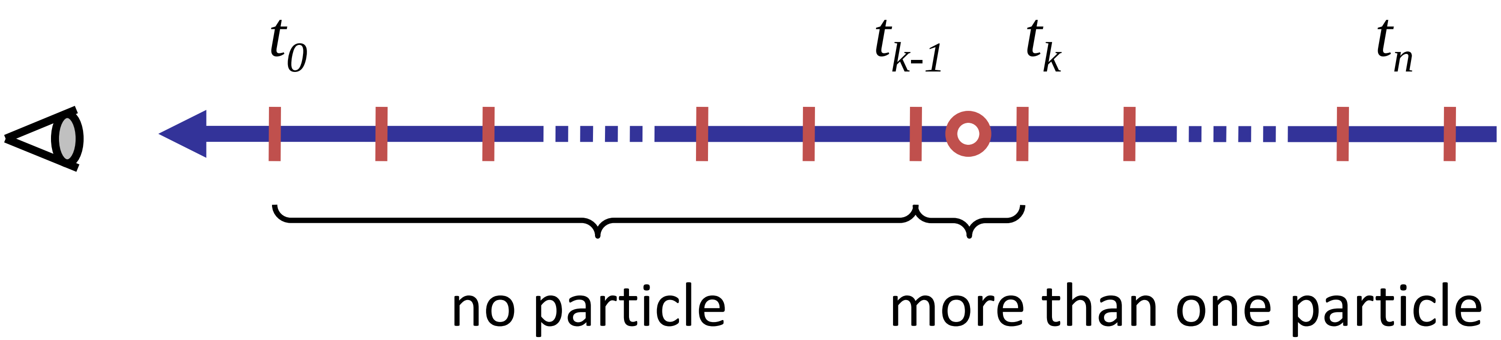
- Generate a set of **opaque** particles
- Project the particles onto an image plane
- Use an ensemble average



Relationship between conventional rendering and PBR

- Brightness = Expected value of particle colors

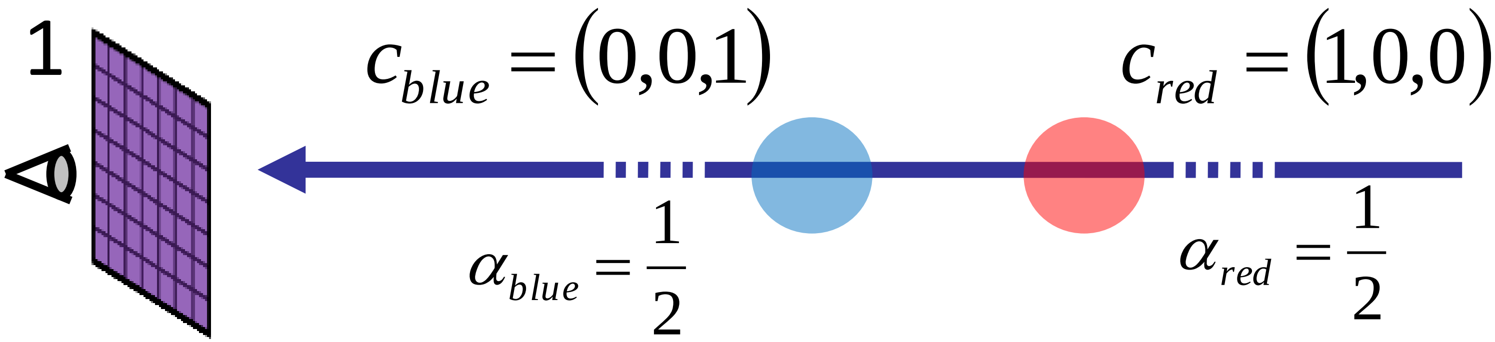
$$B = \sum_{i=1}^{n} c_i \times \underbrace{\left(\alpha_i \prod_{j=1}^{i-1} (1-\alpha_j)\right)}_{P_k} \qquad B = \sum_{k=1}^{n} c_k P_k$$



Probability of $B = c_k$ $\qquad$ $P_k = \alpha_k \prod_{j=1}^{k-1} (1-\alpha_j)$

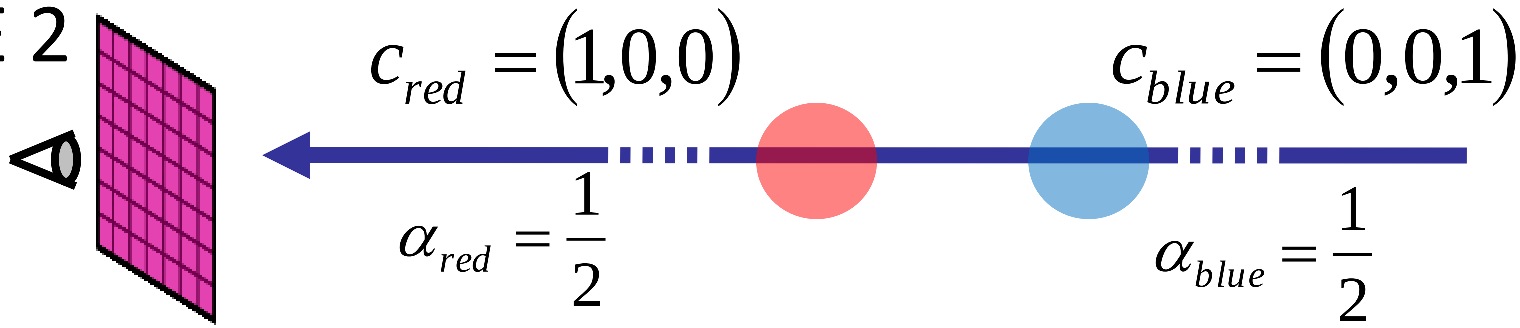
Simple cases

CASE 1



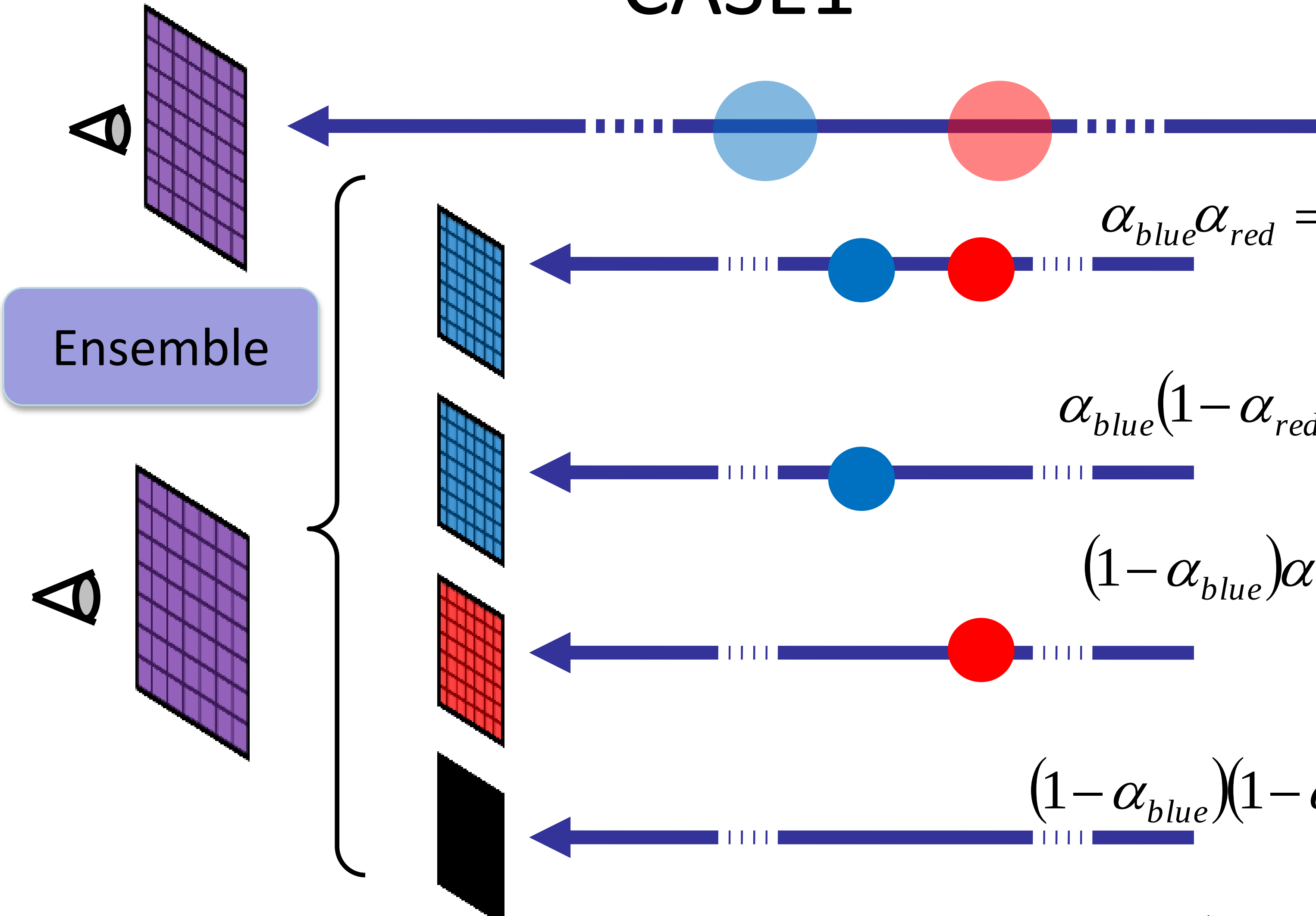
$$B = (1-\alpha_{blue})\alpha_{red}c_{red} + \alpha_{blue}c_{blue} = \left(\frac{1}{4}, 0, \frac{1}{2}\right)$$

CASE 2



$$B = (1-\alpha_{red})\alpha_{blue}c_{blue} + \alpha_{red}c_{red} = \left(\frac{1}{2}, 0, \frac{1}{4}\right)$$

CASE1



$$\alpha_{blue}\alpha_{red} = \frac{1}{4}$$

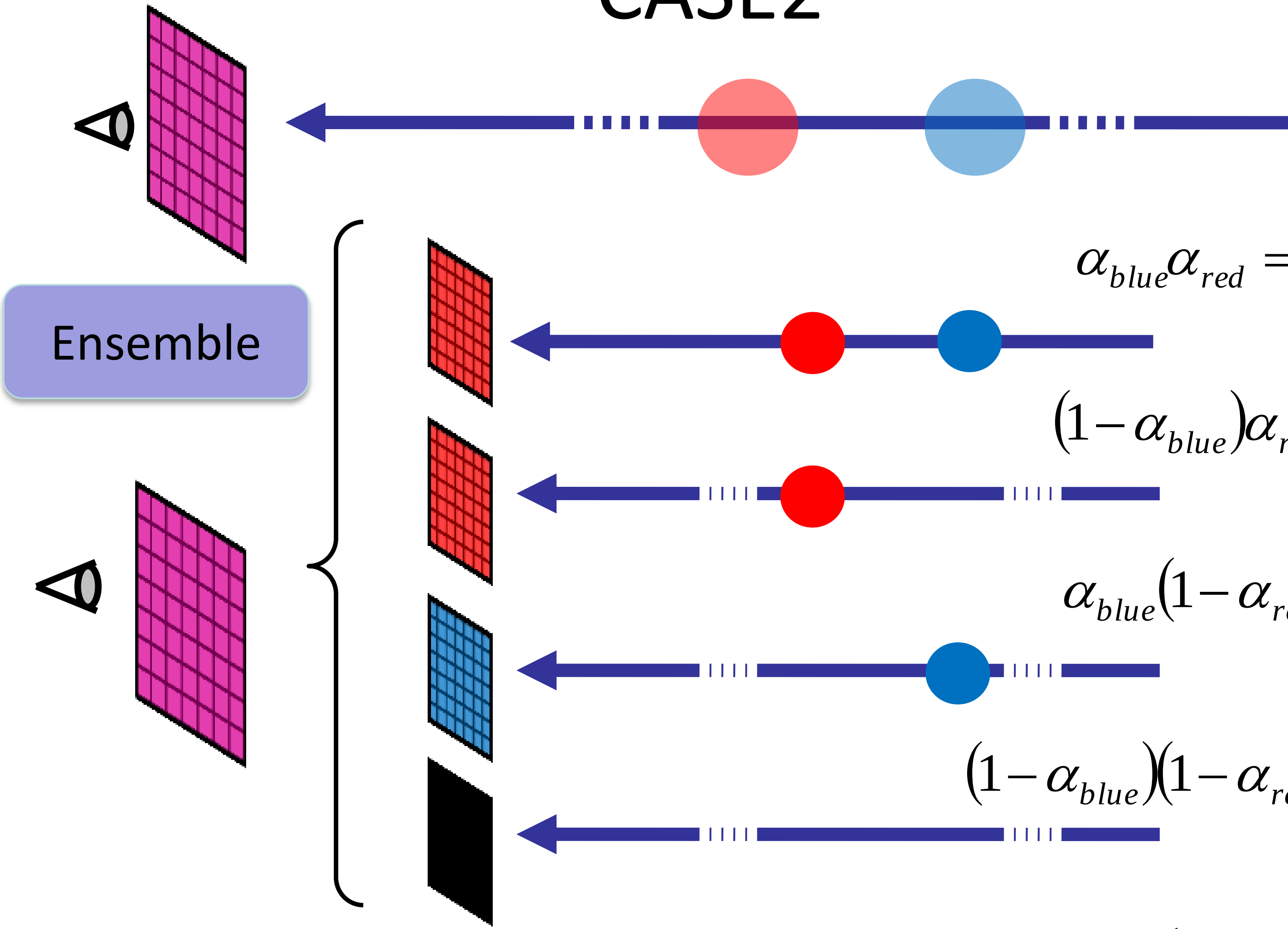
$$\alpha_{blue}(1-\alpha_{red}) = \frac{1}{4}$$

$$(1-\alpha_{blue})\alpha_{red} = \frac{1}{4}$$

$$(1-\alpha_{blue})(1-\alpha_{red}) = \frac{1}{4}$$

$$B = \frac{1}{4}[(0,0,1)+(0,0,1)+(1,0,0)+(0,0,0)] = \left(\frac{1}{4},0,\frac{1}{2}\right)$$

CASE2



$$\alpha_{blue}\alpha_{red} = \frac{1}{4}$$

$$(1-\alpha_{blue})\alpha_{red} = \frac{1}{4}$$

$$\alpha_{blue}(1-\alpha_{red}) = \frac{1}{4}$$

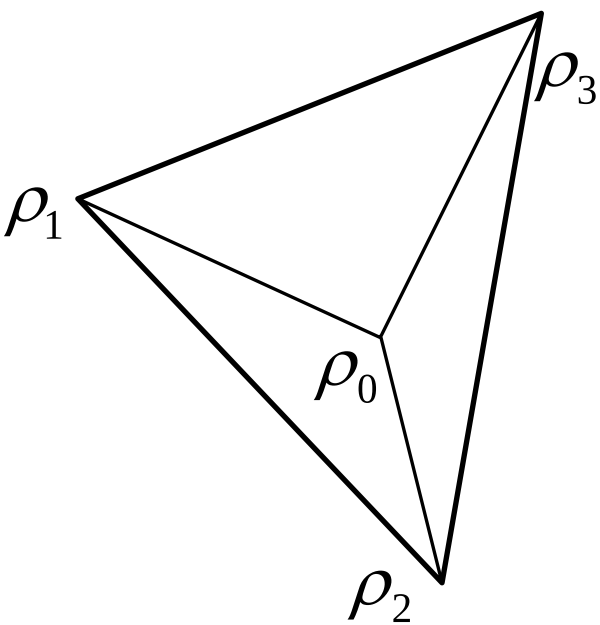
$$(1-\alpha_{blue})(1-\alpha_{red}) = \frac{1}{4}$$

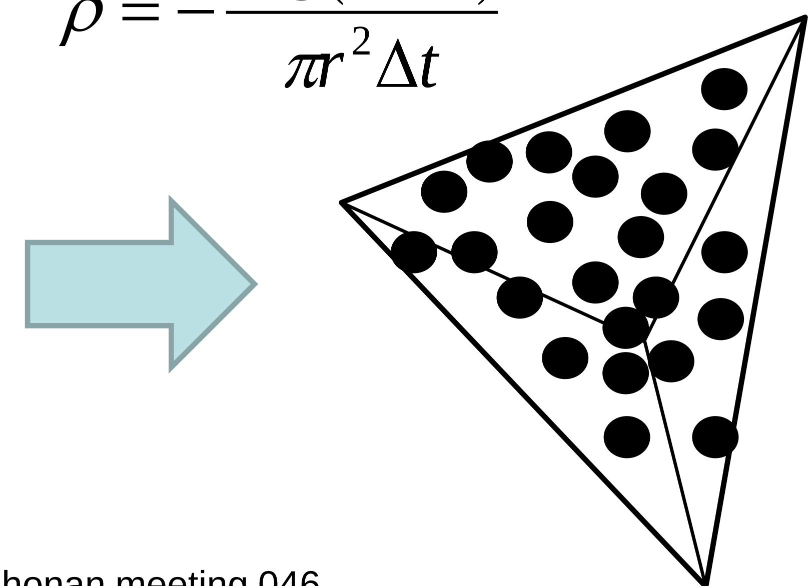
$$B = \frac{1}{4}\left[(0,0,1)+(0,0,1)+(1,0,0)+(0,0,0)\right] = \left(\frac{1}{2},0,\frac{1}{4}\right)$$

PBR in an object space (O-PBR)

- Generate a set of **opaque** particles in each cell
- Project the particles onto an image plane
- Use an ensemble average

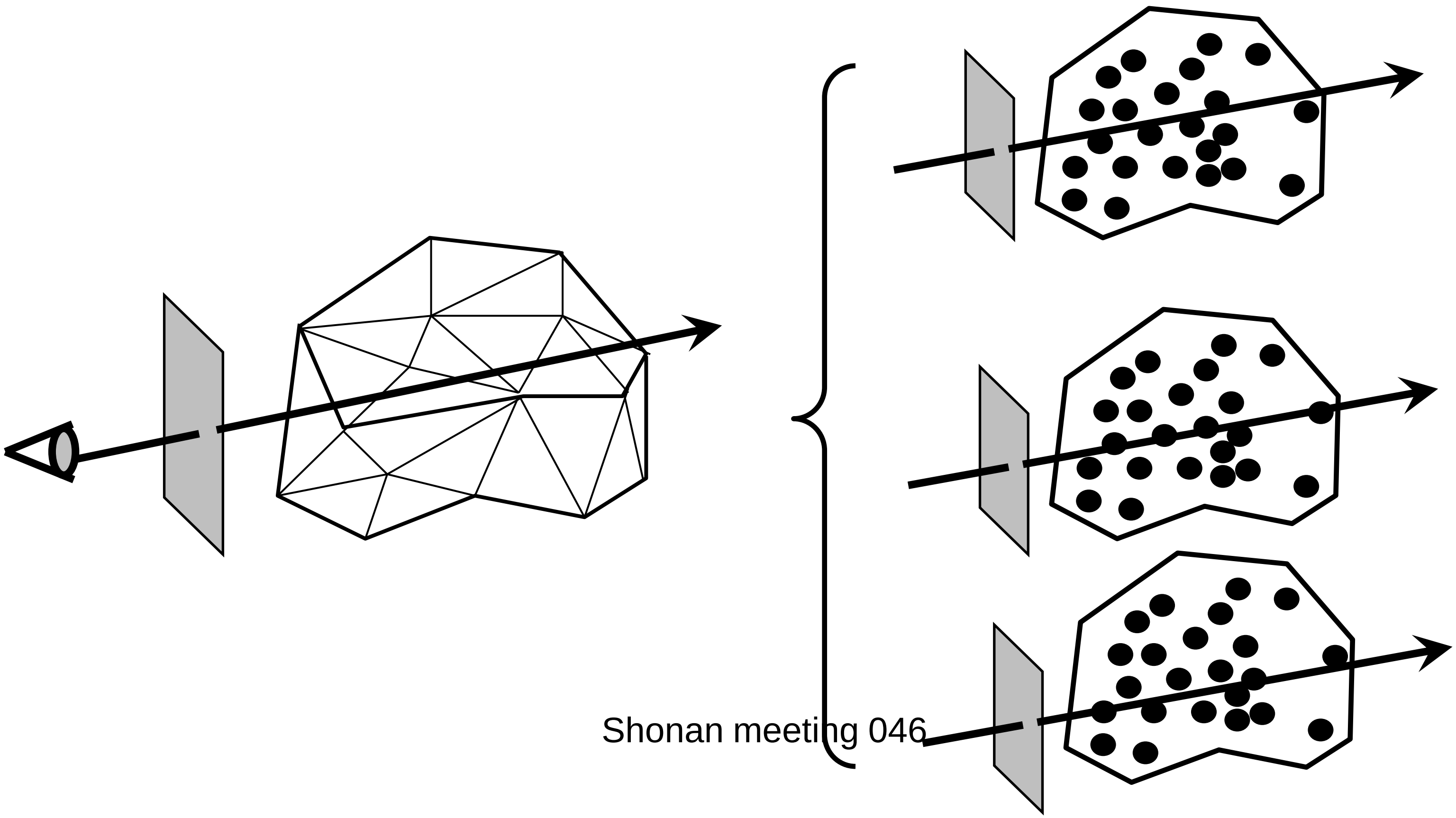
$$\rho = -\frac{\log(1-\alpha)}{\pi r^2 \Delta t}$$





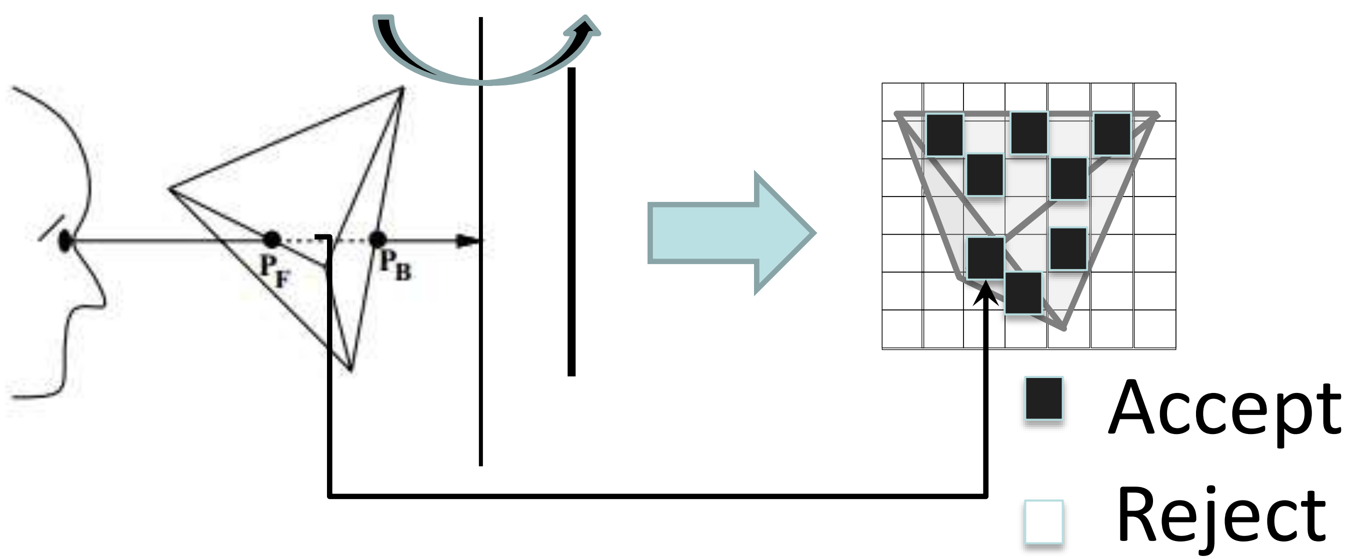
O-PBR procedure

- Generate a set of **opaque** particles in each cell
- Project the primitives onto an image plane
- Use an ensemble average



PBR in an image space (I-PBR)

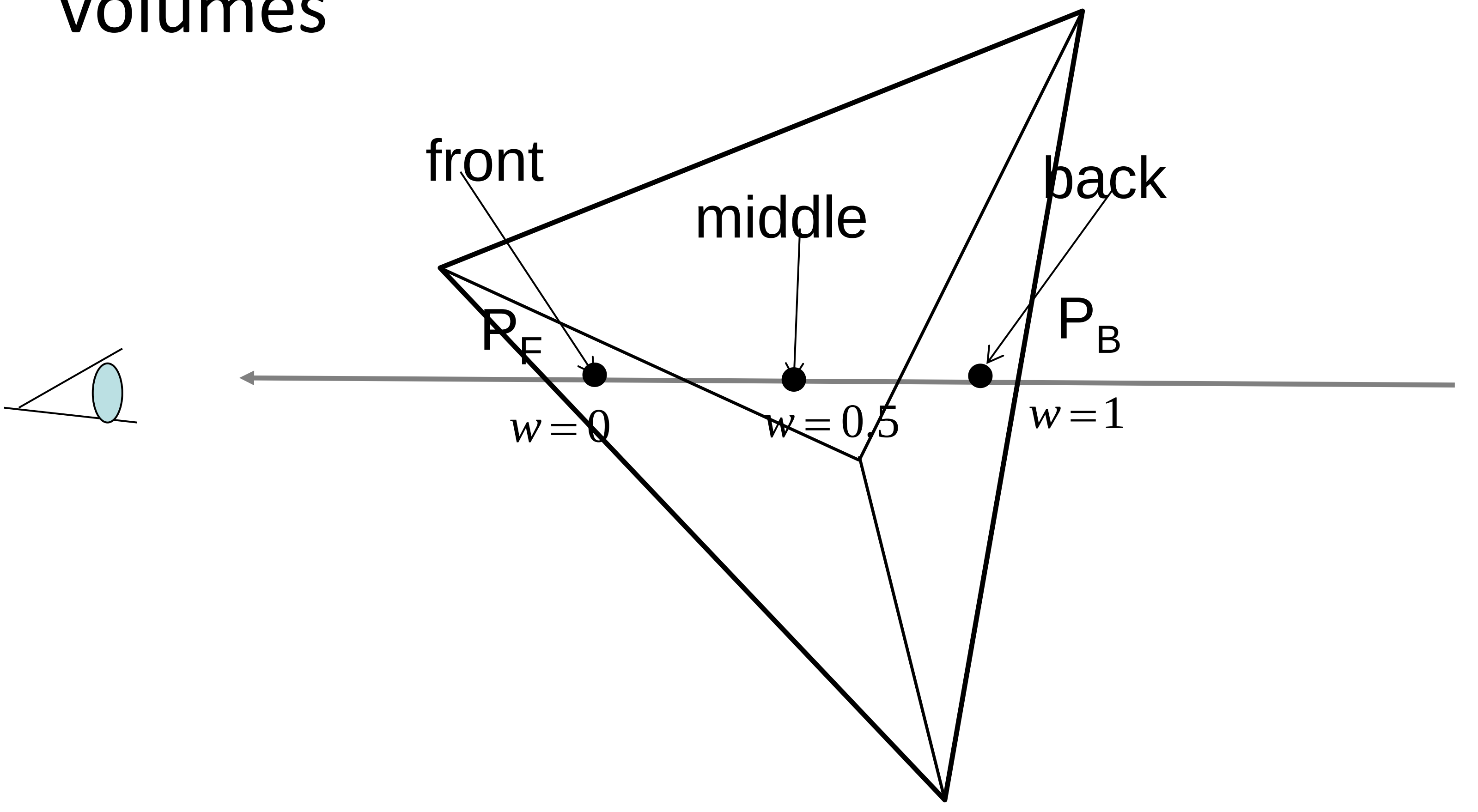
- Generate a set of **opaque** particles in each footprint
  1. Evaluate an opacity at each fragment
  2. Accept the fragment if it > a random number
- Project the primitives onto an image plane
- Use an ensemble average



P. Shirley and A. Tachman "A Polygonal Approximation to Direct Scalar Volume Rendering,"

How to determine the depth ?

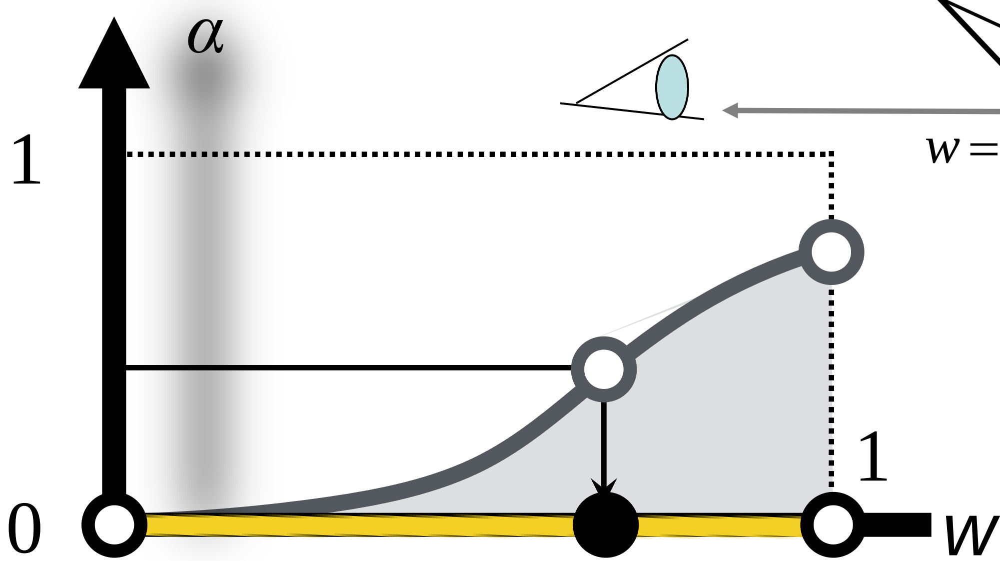
- Any location is OK for a single volume
- But, it may cause an artifact for multiple volumes

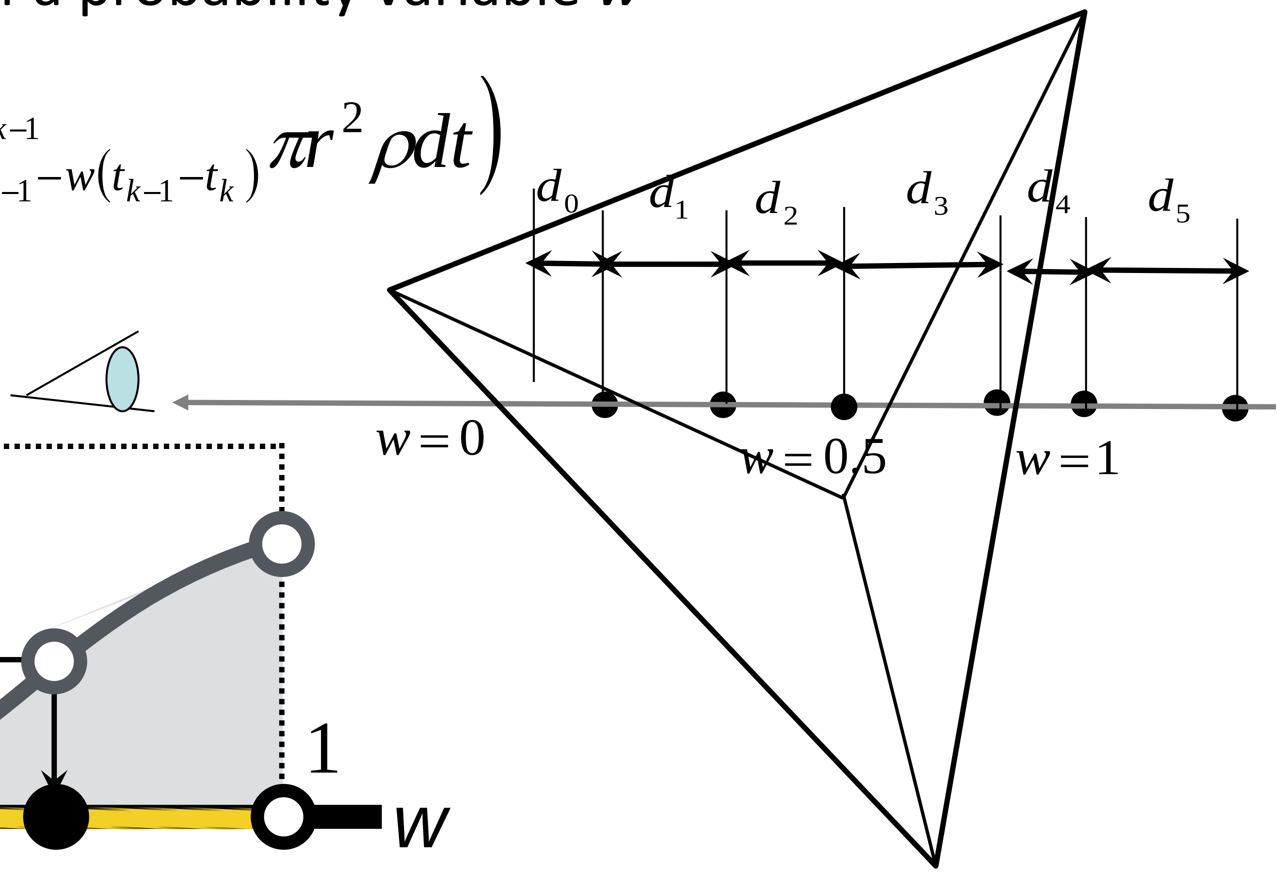


Use of exponential random numbers

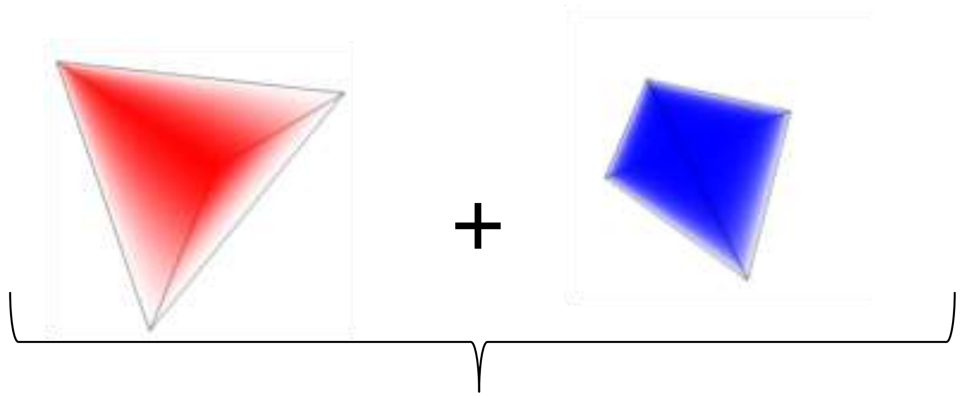
- The distance between particles follow the exponential distribution if their number follows the poisson distribution
- The opacity $\alpha(w)$ is regarded as a cumulative distribution function (CDF) of a probability variable w

$$\alpha(w) = 1 - \exp\left(-\int_{t_{k-1} - w(t_{k-1} - t_k)}^{t_{k-1}} \pi r^2 \rho \, dt\right)$$

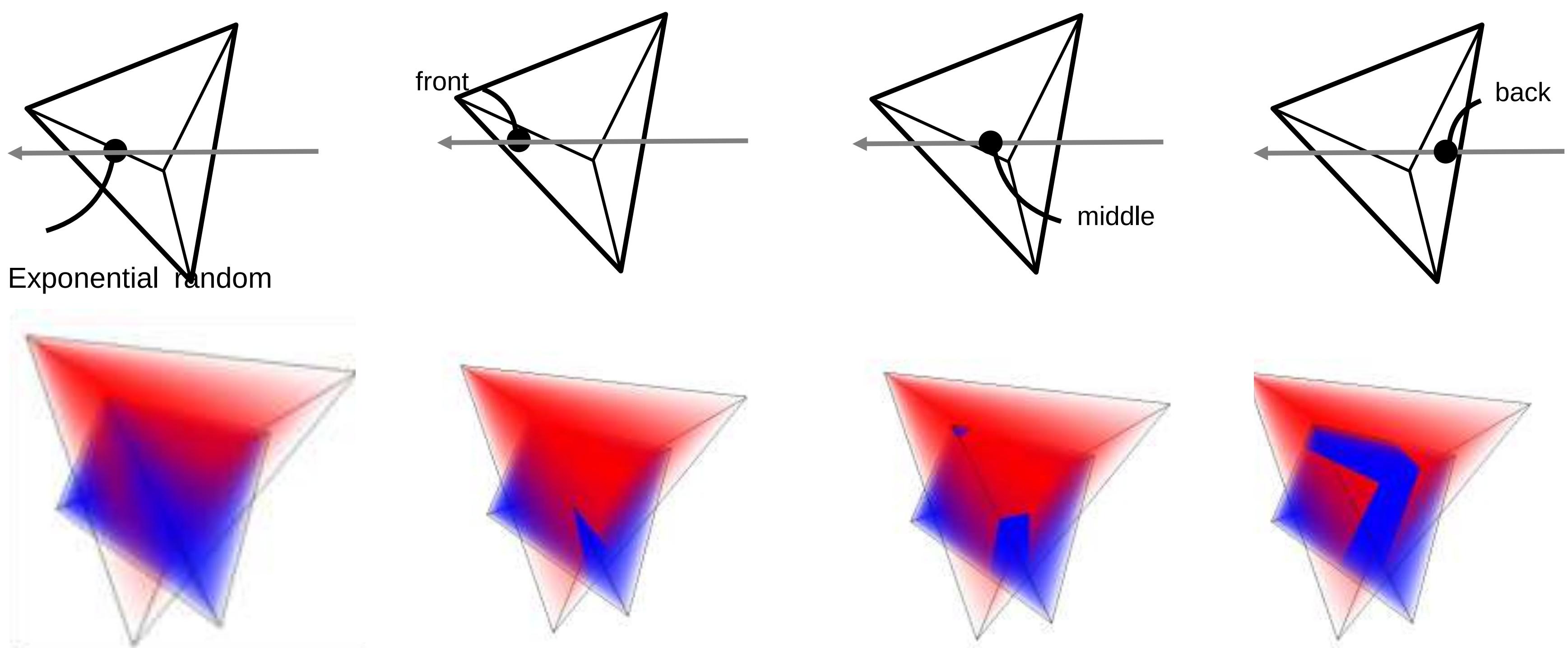




Fusion of two tetrahedra cells



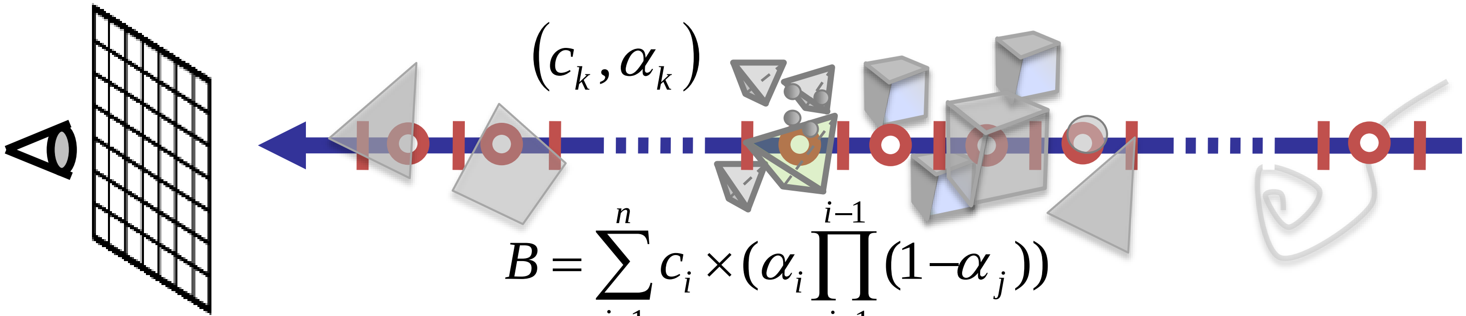
Demo



Particle-Based Rendering and its applications

SUMMARY

Particle-Based Rendering

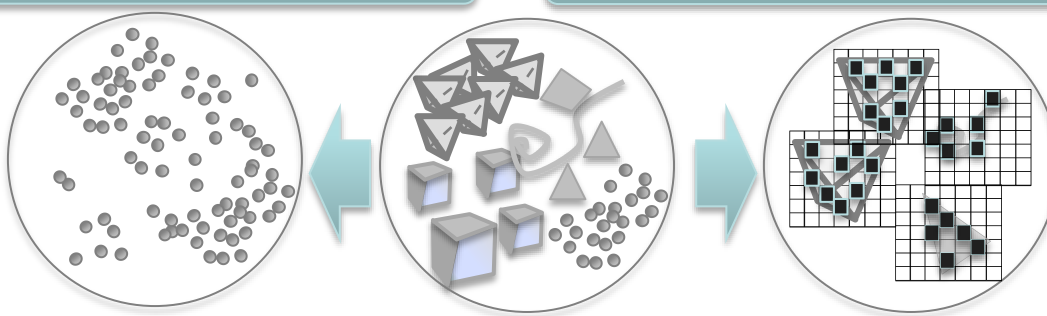


$$B = \sum_{i=1}^{n} c_i \times (\alpha_i \prod_{j=1}^{i-1} (1-\alpha_j))$$

O-PBR	I-PBR

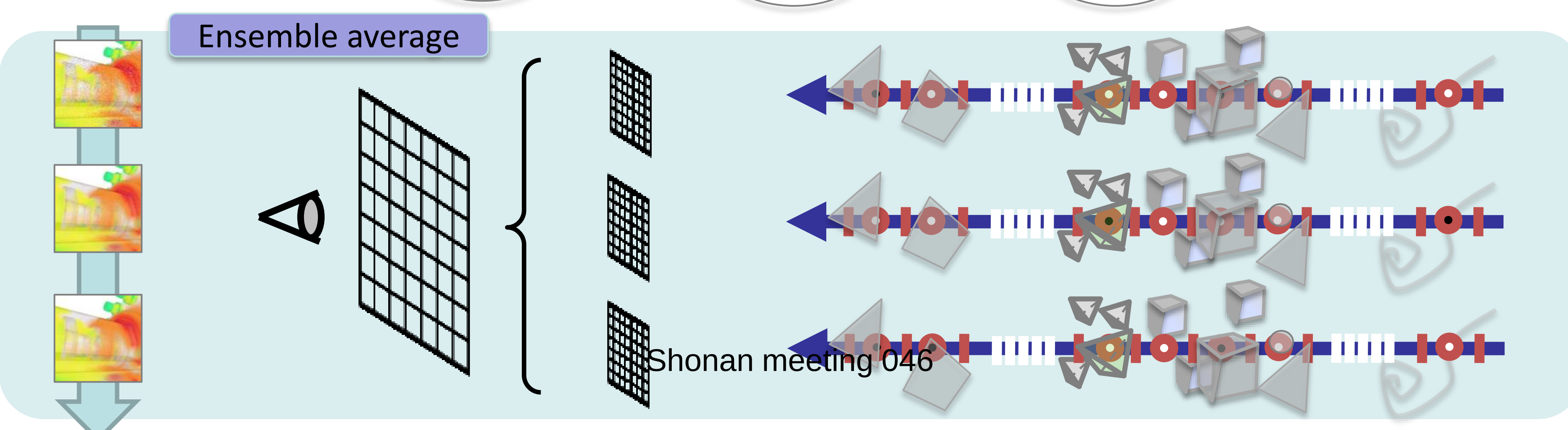
$$\alpha(s) = 1 - e^{-\pi r^2 \rho \Delta t}$$

$$\rho(s) = -\frac{\log(1-\alpha(s))}{\pi r^2 \Delta t}$$



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

$$P_k = \alpha_k \prod_{j=1}^{k-1} (1-\alpha_j)$$



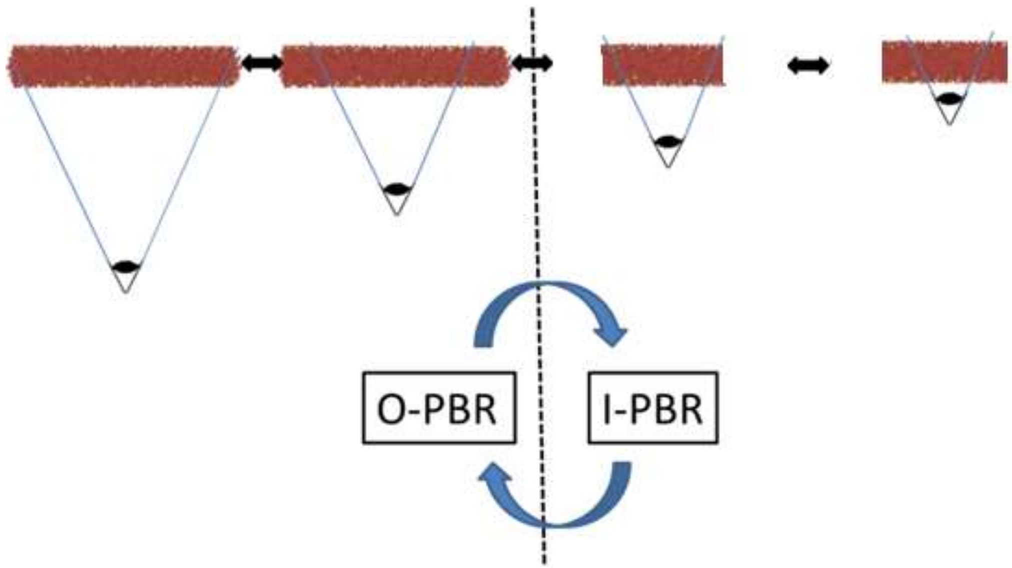
Comparison of O-PBR and I-PBR

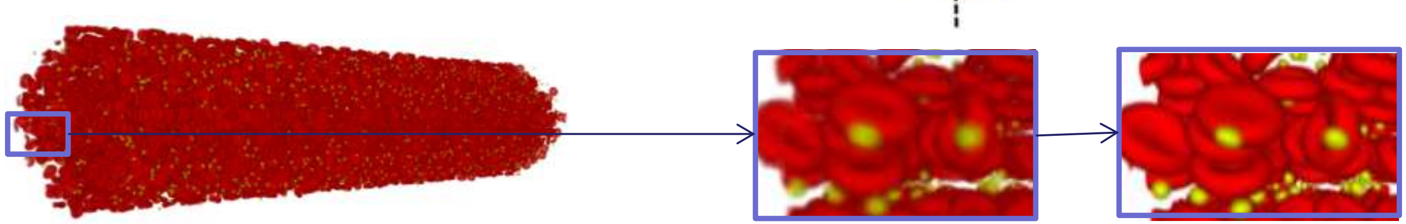
- Parameters
  - N_c: Number of cells
  - N_P: Number of particles

Method	Particle generation	Rendering
O-PBR	$O(N_c)$ CPU	$O(N_P)$ GPU
I-PBR	$O(N_c)$ GPU	$O(N_c)$ GPU

Integration of O-PBR and I-PBR

- Large-scale blood flow dataset
  - Red blood cells
  - Platelets
  - Vascular wall
  - Velocity vector
- Resolution: 3,072 x 640 x 640
- Time steps: 100





D. Hara, N. Sakamoto, K. Koyamada, K. Sugiyama, “Adaptive visualization of large-scale blood flow using particle-based rendering,” The 7th IEEE PacificVis Symposium Poster,2014

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)

$$
\begin{aligned}
T^{SPT} &= L_r\left(5.67\times10^{-5}n_c + 4.51\times10^{-6}whL_s{}^2\right) \\
M^{SPT} &= 20.6n_c + 7.1whL_s{}^2
\end{aligned}
$$

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

$$
\begin{aligned}
T^{PBVR} &= L_r\left(9.96\times10^{-7}n_p + 1.47\times10^{-5}whL_s{}^2\right) \\
M^{PBVR} &= 22.0n_p + 6.86whL_s{}^2
\end{aligned}
$$

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

$$
\begin{aligned}
T^{HAVS} &= 1.88\times10^{-6}n_c + 1.71\times10^{-6}wh + 0.275k \\
M^{HAVS} &= 37.65n_c + 76.3wh
\end{aligned}
$$

Performance evaluation (2/2)

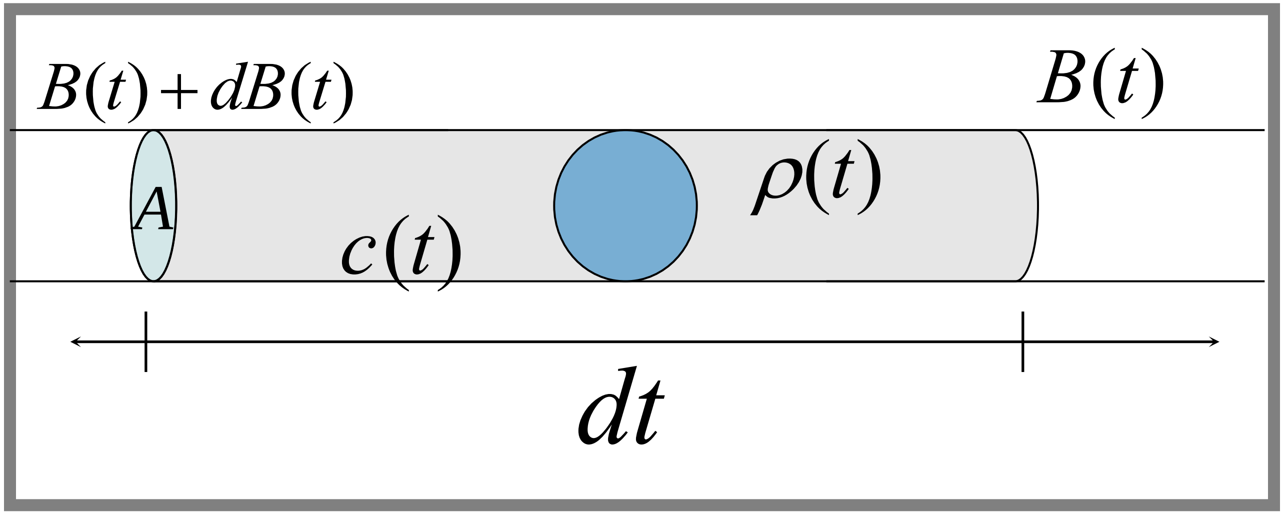
- Computational time
  - HAVS < SPT, PBVR
- GPU memory resources
  - HAVS > SPT, PBVR
- SPT vs. PBVR (512GB VRAM, 1024x1024)
  - if # of tets. < 24.5M SPT
  - otherwise PBVR

Stochastic rendering for facilitating a high-precision modeling

VISUALIZATION

Optical model of volume rendering

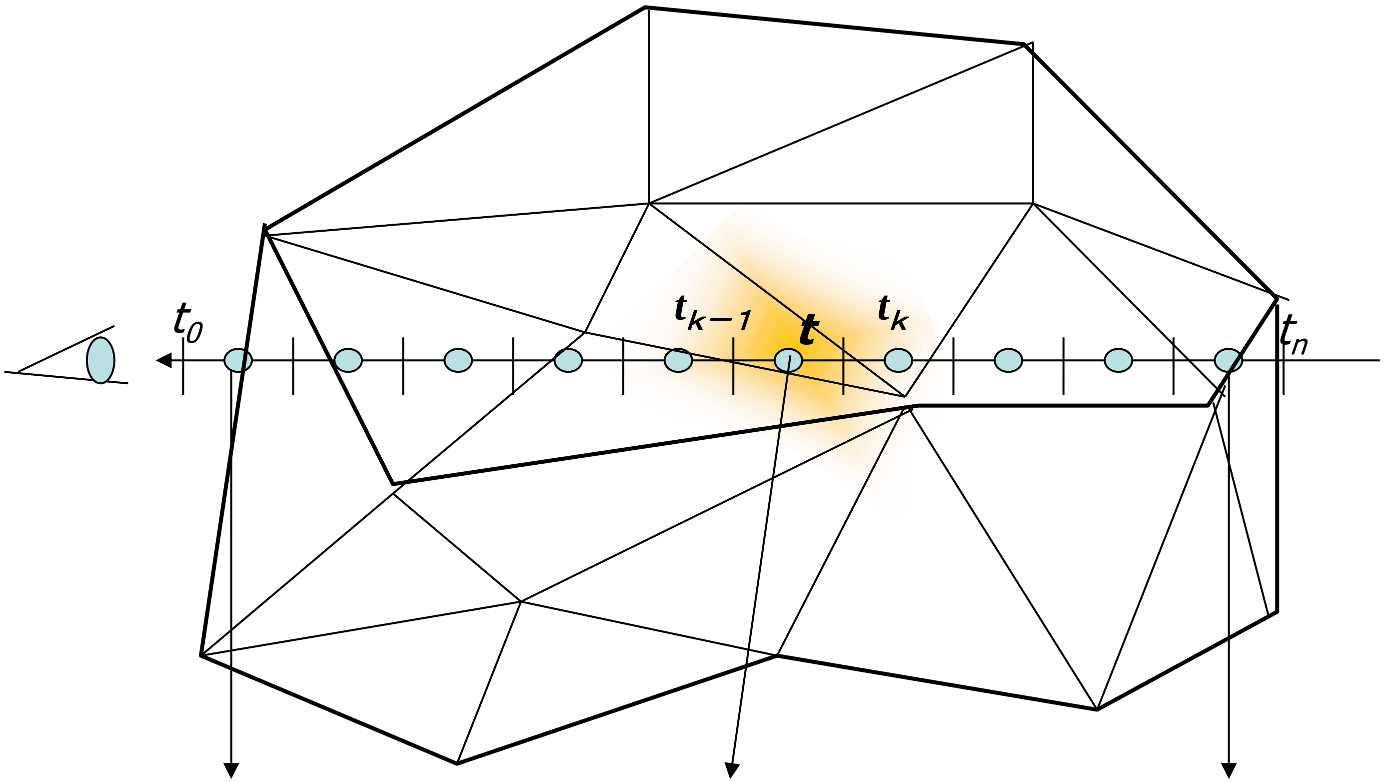
Sabella, 1988



$B(t)$: Brightness $\rho(t)$: particle density r : particle radius $c(t)$: luminosity

$$
\begin{aligned}
dB(t)A &= -absorbed + emitted \\
&= (-B(t)+c(t)) \times \pi r^2 \rho(t) A dt \\
\frac{dB(t)}{dt} &= (-B(t)+c(t)) \times \pi r^2 \rho(t)
\end{aligned}
$$

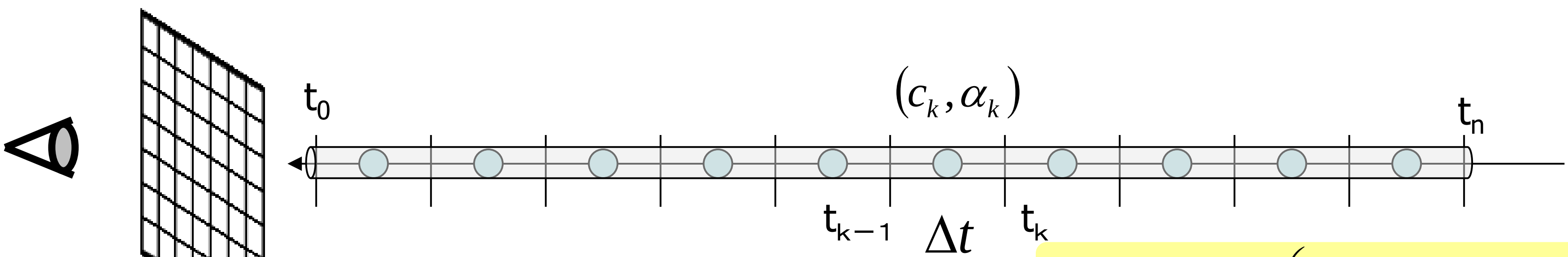
Discretization



$$B = B_1 + \bullet\bullet + B_k + \bullet\bullet + B_n$$

Brightness equation

- Sum of contributions from all the segments



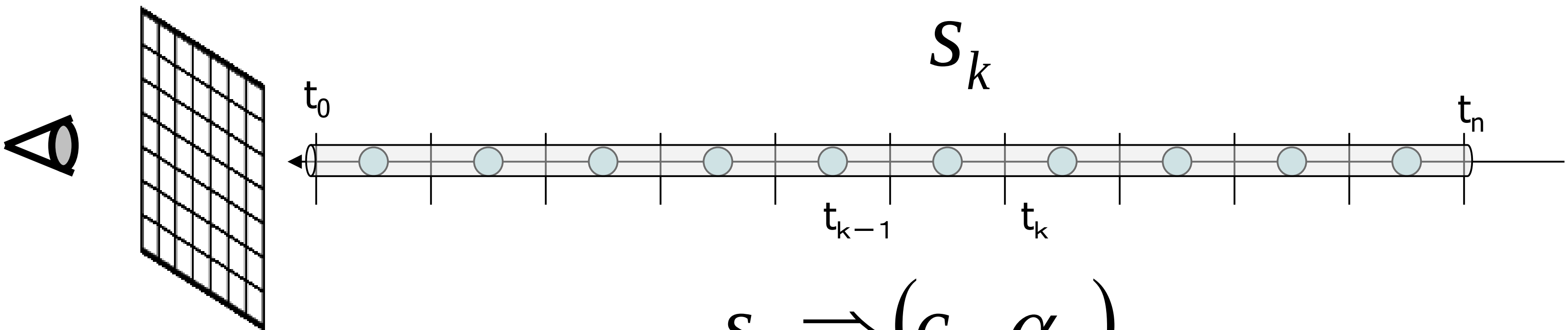
$$\alpha_k = 1 - \exp\left(- \int_{t_k}^{t_{k-1}} \pi r^2 \rho(\lambda) d\lambda \right) = 1 - \exp\left(- N_{p_k} \right)$$

N_{Pk}: Number of particles in the k-th ray segment

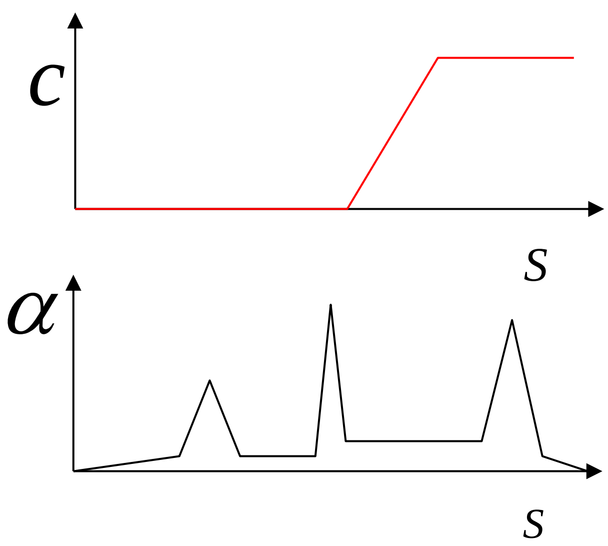
$$B = \sum_{i=1}^{n} c_i \times (\alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j))$$

Transfer function

- Transform scalar value into color and opacity



$$s_k \Rightarrow (c_k, \alpha_k)$$



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

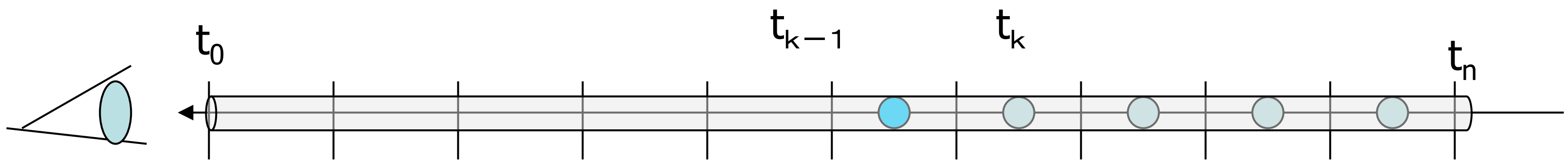
Demo

Back-to-front algorithm

- Derive a recurrence relation by adding a particle from back
- Blend intermediate brightness and particle color
- Assume that all the segments are sorted in the visibility order

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

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



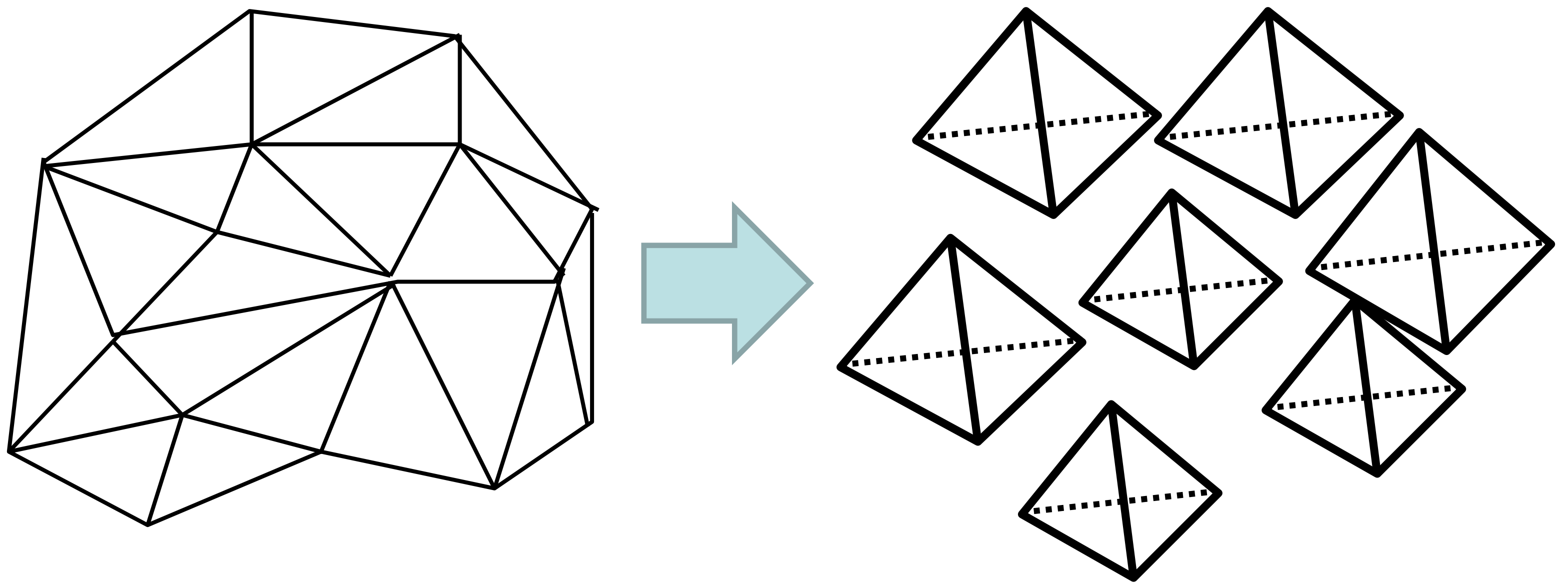
$$B^{k-1} = c_k \alpha_k + (1-\alpha_k) B^k$$

Stochastic rendering for facilitating a high-precision modeling

VOLUME RENDERING FOR IRREGULAR GRIDS

Irregular grids

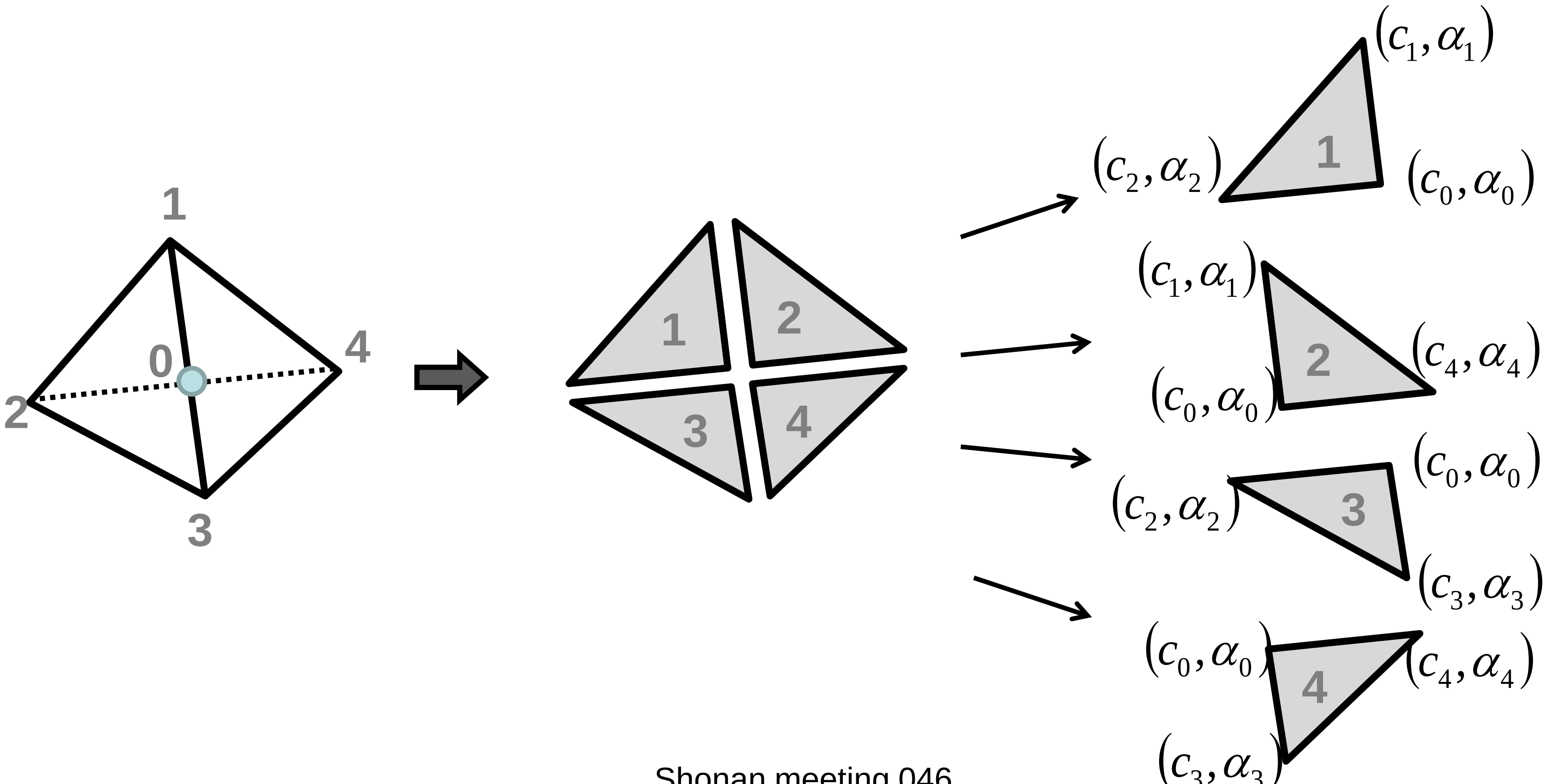
- Assume that irregular grids are composed of tetrahedral cells



Projected tetrahedra

1990, Shirley and Tuchman

- Split a triangle into 3 or 4 triangles on an image plane

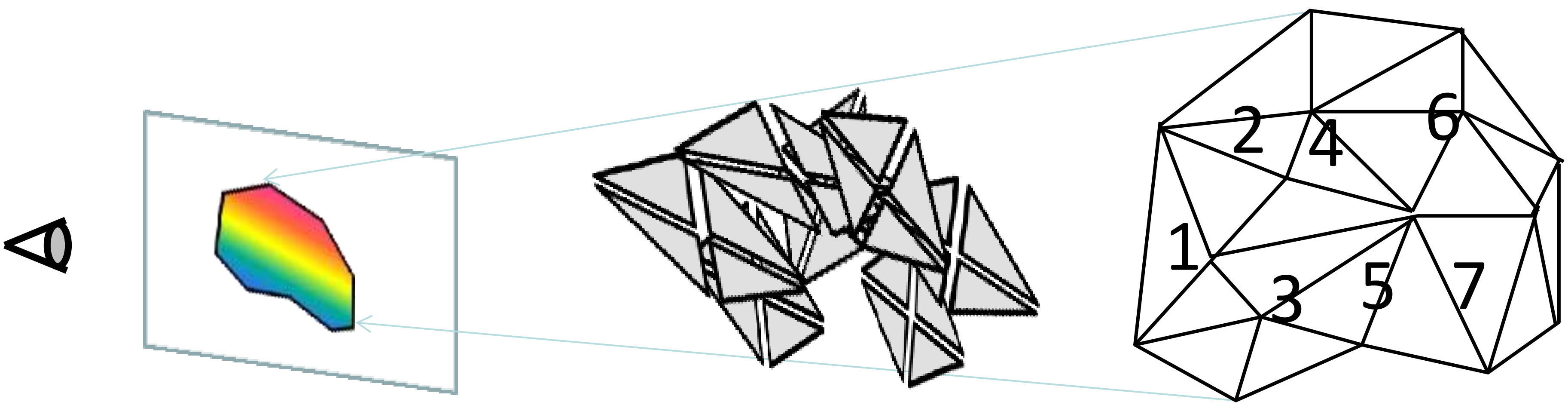


[P.Shirley, et. al., 1990]

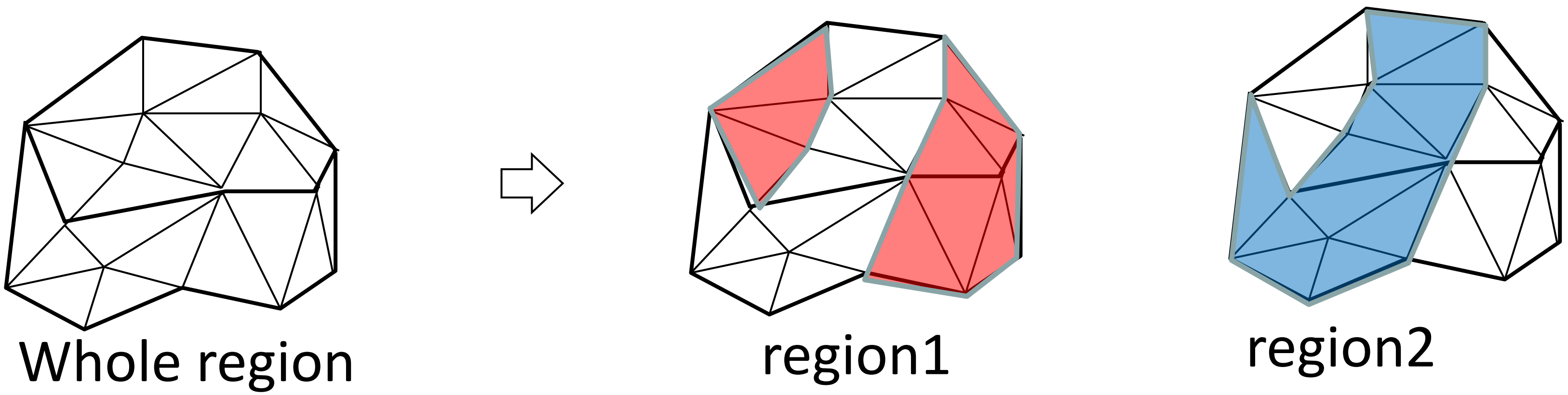
Projected tetrahedra

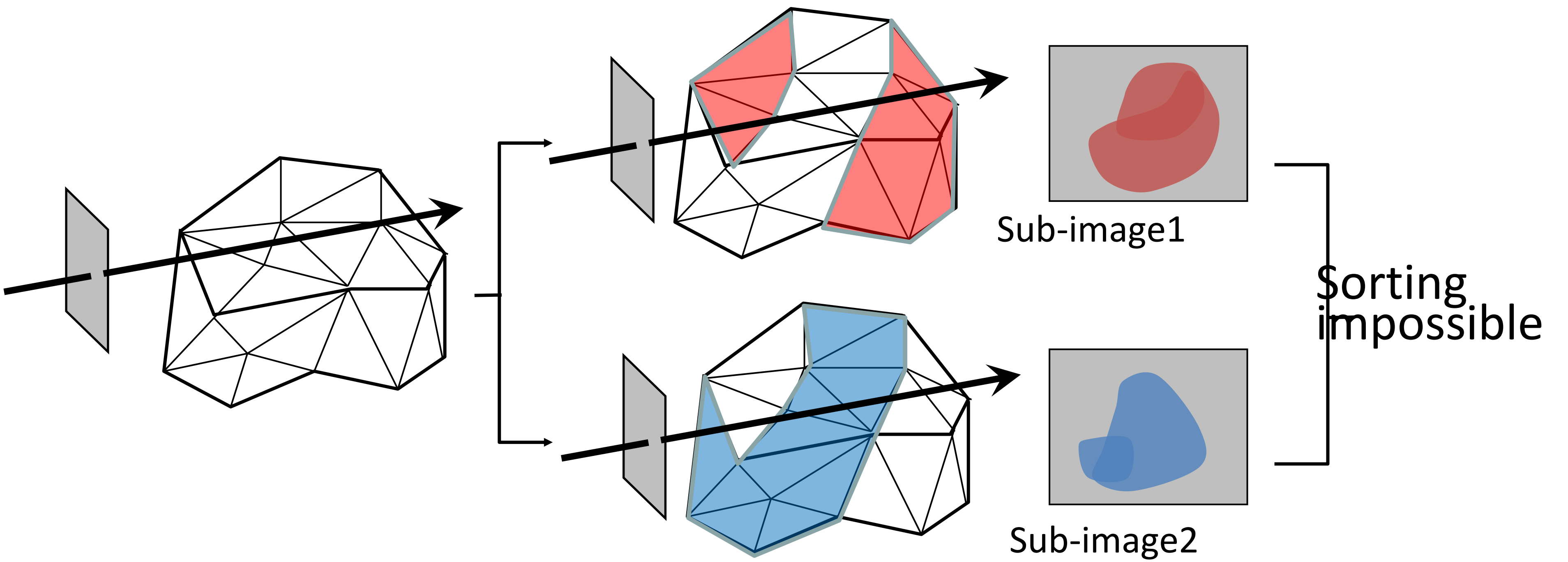
1990, Shirley and Tuchman

- Sort tetrahedral grid cells
- Project cells in the visibility order
- Subdivide the cells into 3 or 4 semi-transparent triangles
- Superimpose triangles in back-to-front order

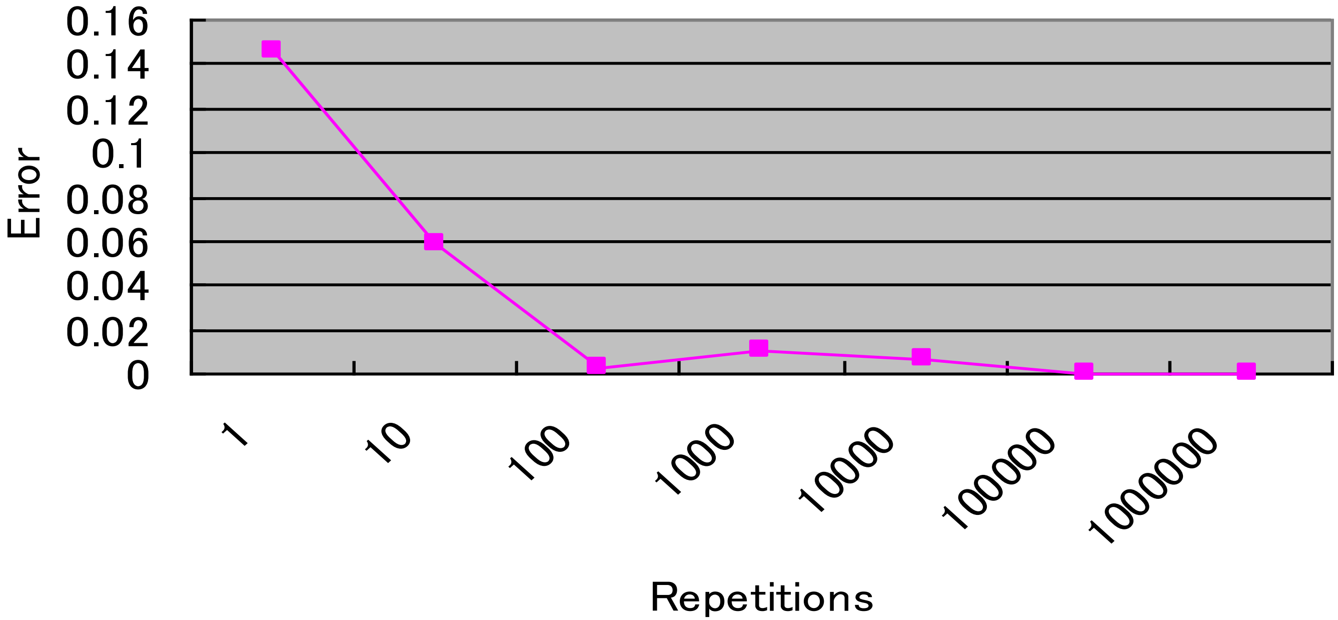


Sorting can be impossible





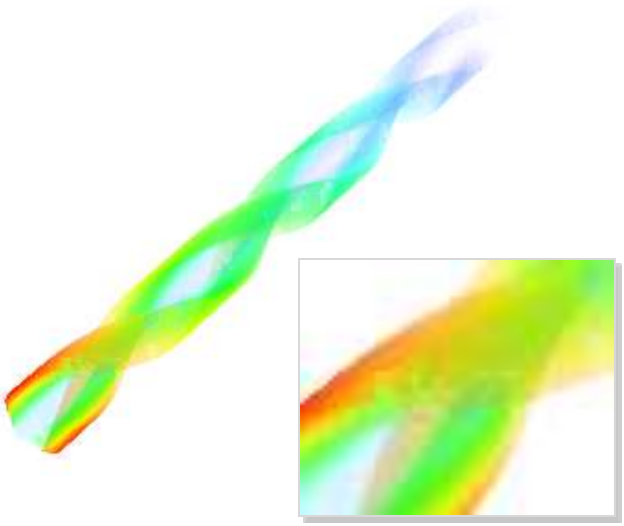
Accuracy of Sort-free rendering



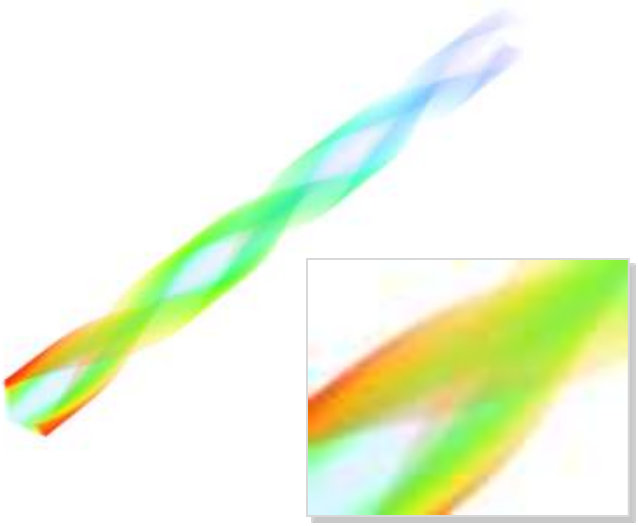
- The error is defined as the absolute difference between the true and approximated brightness values.
- The true brightness value is calculated by generating random numbers as opacities and luminosities in all of the ray-segments

Comparison with HAVS

- Drill data, 9.9M tets



HAVS (with sorting)



Ours(without sorting)