Chapter

Particle-Based Fused Rendering

Koji Koyamada and Naohisa Sakamoto

Abstract

In this chapter, we propose a fused rendering technique that can integrally handle multiple irregular volumes. Although there is a strong requirement for understanding large-scale datasets generated from coupled simulation techniques such as computational structure mechanics (CSM) and computational fluid dynamics (CFD), there is no fused rendering technique to the best of our knowledge. For this purpose, we can employ the particle-based volume rendering (PBVR) technique for each irregular volume dataset. Since the current PBVR technique regards an irregular cell as a planar footprint during depth evaluation, the straightforward employment causes some artifacts especially at the cell boundaries. To solve the problem, we calculate the depth value based on the assumption that the opacity describes the cumulative distribution function (CDF) of a probability variable, w, which shows a length from the entry point in the fragment interval in the cell. In our experiments, we applied our method to numerical simulation results in which two different irregular grid cells are defined in the same space and confirmed its effectiveness with respect to the image quality.

Keywords: volume rendering, irregular volume, unstructured grid

1. Introduction

Coupled analysis is important to solve complex phenomena. Several computational schemes such as computational fluid dynamics (CFD), computational structure mechanics (CSM), and computational electronic magnetics (CEM) are applied to the same geometrical domain, and high-performance computing (HPC) facility has been used for the computation. Since, in general, the requirement for the computational grid is different at each scheme, and the space is composed of multiple irregular volumes. Thus, a volume rendering technique which can handle multiple irregular volumes is expected.

Volume rendering can provide useful information because with this technique, it is possible to grasp the spatial distribution of the related physical quantities. It is a powerful technique for displaying volume datasets, especially three-dimensional scalar data fields, which are composed of scalar data values defined in three-dimensional space. In a CSM, CFD or CEM simulation, the three-dimensional space is composed of computational cells, the shapes of which are, for example, tetrahedra, prisms, and hexahedra. In a large-scale simulation model for complex physical phenomena, the number of computational cells may be more than one million.

Current volume rendering techniques require discrete sampling in which the composition is executed in the visibility order. In the volume rendering calculation, accumulating the optical depth is time consuming. The optical depth is accumulated so that we can efficiently calculate the occlusion of the scattered light from various lighting positions. Most volume rendering techniques accumulate the optical depth in order from front to back or from back to front along a viewing ray.

Although this sorted sampling is straightforward in structured grid data, it becomes more complicated if the computation is conducted in a distributed computing environment. In such an environment, the sub-volume dataset is stored in each distributed node. In this case, it is difficult for a sub-volume to be sorted along the viewing ray since the shape of the sub-volume may be concave. On the other hand, the shape of a computational cell is convex. The whole volume is subdivided into multiple sub-volumes so that the total data transmission cost is minimized.

To handle the problem, a particle-based rendering technique, which does not need visibility sorting, has been proposed [1]. In this technique, opaque emissive particles are employed for realizing sorting-free rendering. There are two approaches for the implementation, that is, an object-space approach and an image-space approach (see **Figure 1**). In the former approach, which we call object-space particle-based volume rendering (O-PBVR), a particle density function is estimated from a user-specified transfer function, and a set of opaque particles are generated at each computational cell. The generated particles are projected onto an image plane, and the projection is repeated to improve the image quality. Although O-PBVR shows good scalability for handling large-scale irregular volume [2], the current drawback of the technique is the generation of low-quality images in which particles are visible on the boundary surface polygons when viewed closely. Moreover, with O-PBVR, it is necessary to generate a large number of particles to obtain a high-resolution image.

In the latter approach, we proposed a sorting-free technique by regarding the brightness equation as the expected value of the luminosity of a sampling point along a viewing ray [3]. We applied the technique to a projected tetrahedral technique with pre-integration. We called this image-space particle-based volume rendering (I-PBVR). We conducted a thorough experimental analysis to construct the performance model [3]. The model suggests that I-PBVR is preferable to O-PBVR

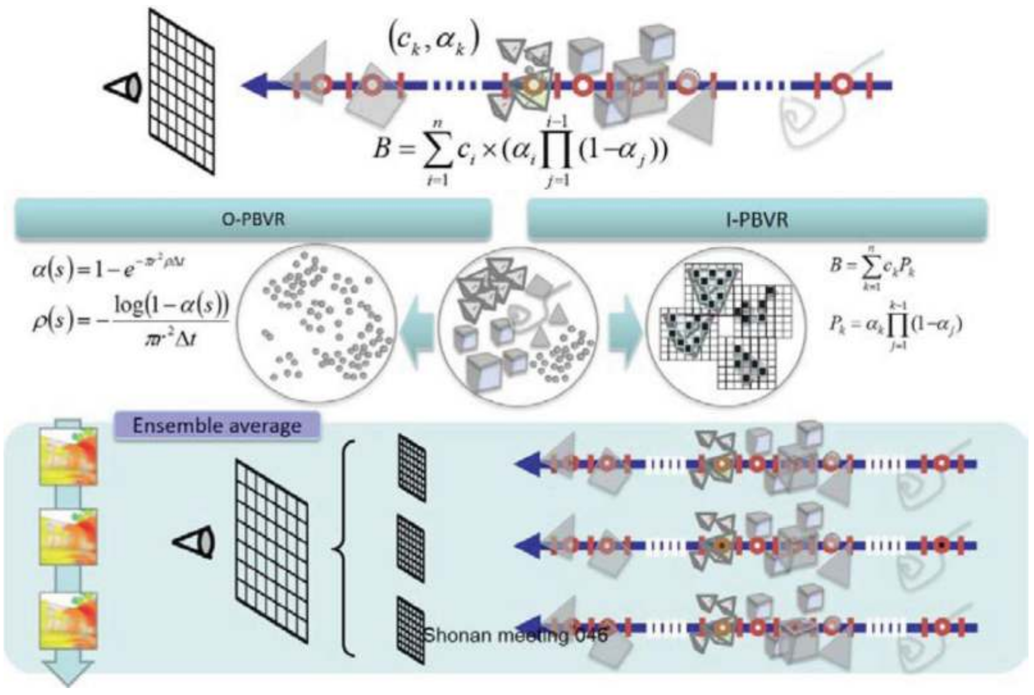
(c_k, α_k)
$B = \sum_{i=1}^{n} c_i \times (\alpha_i \prod_{j=1}^{i-1} (1-\alpha_j))$
O-PBVR
I-PBVR
$\alpha(s) = 1 - e^{-\pi r^2 \rho \Delta t}$
$\rho(s) = -\frac{\log(1-\alpha(s))}{\pi r^2 \Delta t}$
$B = \sum_{k=1}^{n} c_k P_k$
$P_k = \alpha_k \prod_{j=1}^{k-1} (1-\alpha_j)$
Ensemble average
Shonan meeting 046

Figure 1.
Overview of PBVR processes that employ the ensemble average.

when a volume dataset is rendered in a high resolution. In addition, I-PBVR becomes a feasible choice for rendering irregular volume data when particle generation becomes frequent.

To understand the relationships between variables in the irregular volume data, it is necessary to integrate multiple volumes, into a single volume rendering. In this chapter, we improve the I-PBVR technique in terms of its extensibility to multiple volumes and propose a new technique for semi-transparent rendering which can integrally handle multiple volumes without visibility sorting. I-PBVR regards a tetrahedral cell as a triangle footprint on the image plane. When a single volume is rendered, each footprint does not intersect with the other one since the volume is composed of cells that are not overlapped with others. When multiple volumes are rendered, the footprints may intersect with others. This intersection causes a problem in which the cell boundaries are visible when dealing with multiple volumes. Thus, in this chapter, our research question is “How can we realize a fused volume rendering technique which is free from artifact?” The hypothesis to the question is “If we employ an adequate probability process to sample particles along a viewing in a grid cell, the artifact is not noticeable.” In the early age of volume rendering, Blinn assumed that the number of particles is distributed in a volume space according to the Poisson distribution. If we take an interval that is a part of the viewing ray cut by cell faces, the distance between particles is distributed according to the exponent distribution.

In the remaining part, we first describe related work and the basic theory of PBVR and then propose a fused rendering technique using PBVR. Finally, we make a comparative work to confirm the effectiveness of the proposed technique. To test the hypothesis, we design the following experiments for the sampling along the interval:

1. The sampling is made at the entry, exit or middle points along an interval in the grid cell.

2. The sampling is made at a random position along an interval in the grid cell.

3. The sampling is made according to a probability process, which is determined based on an opacity along the interval.

2. Related work

In the particle-based modeling of Saturn’s ring, Blinn assumed that the number of particles follows the Poisson distribution although he did not describe it in detail [4]. The assumption led to a definition of opacity which was an important keyword for the volume rendering. Then, volume rendering has been the focus of intensive study for nearly three decades [5–7]. The volume rendering of unstructured volume data has received much attention, and several approaches have been proposed. Extensive literature and surveys on volume rendering are available that address unstructured volume data [8, 9]. A concern has often been visibility sorting, which causes a severe bottleneck in the interactive exploration.

To solve the problem recognized by many volume rendering researchers, we returned to a density emitter model and presented the basic concept for this approach. The proposed PBVR technique represents the 3D scalar fields as a set of particles and considers both emission and absorption effects [1, 2]. The particle density is derived from a user-specified transfer function and is used to estimate the number of particles to be generated in a given volume dataset. Because the particles can be considered fully opaque, no visibility sorting processing is required during the rendering process, which is advantageous from a distributed processing perspective.

The development of a fused rendering technique began in the seismic or medical imaging field. Lu Cai et al. developed a fused volume rendering technique for multiple seismic attribute volume data by the way of planar slices or horizon slices and revealed a variety of geological phenomena more effectively and clearly in order to provide a true three-dimensional perspective view. Their method can address only regular volume datasets [10].

3. Particle-based volume rendering

3.1 Definition of opacity

In image generation in volume rendering, it is thought that giving a viewing ray (viewpoint and direction), when there is no other emissive particle between the particle and the viewpoint, the energy from the particle reaches the viewpoint. Let us consider a certain section on the viewing ray, where τ particles are distributed on average. Furthermore, suppose that this section is divided into M equal parts, M small sections are formed, and particles are present in k small sections. Here, it is assumed that there is at most one particle in each small section and k is a natural number in addition to 0. At this time, the probability p that there is a particle in the small section is as follows:

$$p = \frac{\tau}{M} \tag{1}$$

Such a distribution follows a binomial distribution, and its probability is given as follows:

$$P(X = k) = {}_M C_k p^k (1-p)^{M-k} \tag{2}$$

Here, the number of particles X is set as a random variable. When M is brought close to infinity while keeping τ = Mp constant, its distribution becomes the Poisson distribution of the average τ. The probability that there are k particles in the section is expressed by the following equation:

$$P(N = k) = \frac{\tau^k e^{-\tau}}{k!} \tag{3}$$

Here, e is the Napier's constant (e = 2.71828 ...), and k! is the factorial of k. Thus, the probability is a positive real number. The Poisson distribution is often employed in the context of the number of occurrences of events within the interval defined in the time domain, but in volume rendering, it is considered not in the time domain but in the space domain. Eq. 3 represents the probability that k particles exist when the average number of particles is τ. When k = 0, it represents the situation in which no particle exists in the section on the viewing ray.

$$P(N = 0) = e^{-\tau} \tag{4}$$

If there are many particles, the rate at which energy reaches the viewpoint becomes small, and it is easier to define the passing distance of light by the number of particles rather than the actual distance. Therefore, the average particle number τ

is also called the optical thickness. Additionally, a negative sign is given to this optical thickness, and an index is taken, that is, Eq. 4 is called transparency (t), and it shows the ease of light transmission. The opacity α is obtained by subtracting this transparency from 1 and represents the probability that one or more particles exist in the section.

3.2 Volume rendering

In traditional computer graphics, it is assumed that all light is radiated from the outside, the particles constituting the object are treated as reflectors, and the light scattering and absorption are repeated inside the object. On the other hand, in the volume rendering technique proposed by Sabella in 1988, the internal structure of the volume data can be known by treating the particle as a radiator in addition to a reflector (particle emission model) for the purpose of visualizing the scalar volume.

In Blinn's model and Kajiya's model, the radiant energy is only reflected from the light source and energy emission (luminescence) by the particles themselves has not been considered. However, in the model of Sabella, from the standpoint of visualizing the volume data, we assume that the particles themselves emit light.

To accurately simulate the light scattering phenomenon inside the object, complicated analysis using radiation theory becomes necessary, and it is necessary to solve the scattering equation derived from that theory. In the particle emission model, focusing only on the viewing ray direction, we use a simple equation describing the transmission of optical energy with volume data. This equation can be formulated as follows by considering the difference in radiant intensity (luminance value) B in a cylindrical tube of minute length.

$$\begin{aligned} dB(t)A &= -absorbed + emitted \\ &= (-B(t) + c(t)) \times \pi r^2 \rho(t) A dt \\ \frac{dB(t)}{dt} &= (-B(t) + c(t)) \times \pi r^2 \rho(t) \end{aligned} \tag{5}$$

Here, $\rho(t)$ is the particle density (the number of particles per unit volume), r is the particle radius, and c (t) is the light emission amount per unit area.

$$B_0 = B(t_0) = \int_{t_n}^{t_0} c(t) \times \pi r^2 \rho(t) \times \exp\left(-\int_{t}^{t_0} \pi r^2 \rho(\lambda) d\lambda\right) dt \tag{6}$$

Eq. 6 is integrated in the interval in which the parameters $t_0$and t_n represent the nearest and the farthest points, respectively, from the viewpoint among the intersection points of the volume data and the viewing ray.

This is called the brightness equation. Generally, the brightness value B and the light emission amount c (t) are composed of three components of red, green, and blue. Eq. 6 shows that energy emitted from a point on the viewing ray reaches the viewpoint by receiving attenuation represented by an exponential term. Note that the exponent term represents the optical thickness τ, so it is equal to the transparency calculated by assuming a Poisson distribution.

In the particle emission model in volume rendering, from the viewpoint of visualization of scalar data, scalar values interpolated in particle positions are converted into color data (composed of three components of red, green, and blue). This conversion table is called a transfer function together with a conversion table to opacity described below.

In volume rendering, by performing shading processing, it is possible to effectively express shading on the isosurface inherent in the scalar volume. Particularly in the case of three-dimensional medical images, it is necessary to visualize complicated structures such as bones, muscles, blood vessels, etc. as isosurfaces, and shading processing becomes important. In shading processing targeting the scalar volume data, luminance value calculation using the gradient vector obtained by interpolation calculation inside the grid cell is performed.

To numerically calculate the brightness equation represented by Eq. 6, an integration area defined on the viewing ray is divided by a step width in which particle emission can be regarded as constant.

At this time, the k-th light emission amount c (t) is regarded as a constant and is set as c_k. In the calculation of integration, the integral of the exponent part is divided into the integral section and others. For those that are divided outside the integration interval, sections corresponding to the division sections are divided and expressed with product signs. Each element of the product symbol is an index obtained by attaching a minus sign to the optical thickness and represents the transparency described above.

As a result, it can be seen that a term of the same form as the exponent term of the divided section outside the integral section and the transparency are included. Transparency takes values from 0 to 1 by definition. The value obtained by subtracting this transparency from 1 is called opacity, as mentioned above, and may be a target of transfer function, and opacity is used in volume rendering in many cases. That is, in the k-th integral interval, the opacity α_k is defined as follows.

$$\alpha_k = 1 - \exp\left(-\int_{t_k}^{t_{k-1}} \pi r^2 \rho(\lambda) d\lambda\right) = 1 - \exp\left(-\pi r^2 \rho_k \Delta t\right) \tag{7}$$

where Δt is the length of the integration interval, and ρ_k is the average particle density in the integral interval k. By introducing this opacity, Eq. 6 is as follows:

$$B_k = c_k \alpha_k \prod_{j=0}^{k-2} \left(1 - \alpha_{j+1}\right) = c_k \alpha_k \prod_{j=1}^{k-1} \left(1 - \alpha_j\right) \tag{8}$$

By adding all the terms described by Eq. 8, the brightness is as follows:

$$B = \sum_{k=1}^{n} \left[c_k \times \alpha_k \prod_{j=1}^{k-1} \left(1 - \alpha_j\right)\right] \tag{9}$$

Normally, this opacity is converted from the scalar value S in the transfer function specified by the user. That is, the opacity is a function of the scalar value S.

$$\alpha = \alpha(S(x, y, z)) \tag{10}$$

If the length of the integration interval is a value Δt‘ different from the predetermined Δt, it is necessary to make the following correction.

$$\begin{aligned} \alpha_k &= 1 - \exp\left(-\pi r^2 \rho_k \Delta t\right) \\ \alpha'_k &= 1 - \exp\left(-\pi r^2 \rho_k \Delta t'\right) \\ \therefore \alpha'_k &= 1 - (1 - \alpha_k)^{\frac{\Delta t'}{\Delta t}} \end{aligned} \tag{11}$$

In volume rendering, the user should have a large opacity (maximum value is 1.0) for the scalar value to be emphasized and a small opacity (minimum value of 0.0 for the less important scalar value) to set the transfer function. By doing so, the relationship between the scalar value and the opacity can be defined, and according to Eq. 11, the following particle density is defined in the three-dimensional region in which the volume data are defined.

$$\rho_k = \frac{\log(1-\alpha_k)}{\pi r^2 \Delta t} \tag{12}$$

Eq. 12 defines the opacity when a transfer function is set in the three-dimensional region, and the particle density is determined when the particle diameter is determined; thus, particles can be generated using an appropriate method. By allocating colors to particles and projecting them on the image plane, volume rendering can be realized.

During the exploration phase, the opacity is often modified in order to change an emphasized region. If we keep the particle radius as defined in the first place, we need to re-generate particles which requires a significant computational time. To avoid the additional particle generation process, we need to change the particle radius on the condition that the particle density stays the same. If we change the opacity from α_k to α'_k, the new radius r' becomes as follows:

$$r' = r\sqrt{\frac{\log(1-\alpha_k)}{\log(1-\alpha'_k)}} \tag{13}$$

3.3 O-PBVR

O-PBVR is comprised of three processes: particle generation, particle projection, and the ensemble average [1]. The first process constructs a density field and generates particles consistent with the density function. The density is derived from a user-specified transfer function that converts a scalar to an opacity data value, and it describes the probability that a particle is present at a given point in space. The second process projects particles onto an image plane, and the corresponding particle buffer stores the particles. Each pixel on the image plane contains sub-pixels (i.e., divided pixels), and the number of division is called the sub-pixel level. This sub-pixel processing is equivalent to an ensemble average which repeats the first and second processes in sequence, N times, and calculates the resulting brightness values by averaging the accumulating color values.

3.3.1 Particle generation

The particle model considers three particle attributes: shape, size, and density. The particle shape is assumed to be spherical, as in the density emitter model. The size of the sphere is characterized by its diameter, which is the same as a side length of a sub-pixel. The particle density ρ can be estimated from the user-specified transfer function. To generate a rendering image equivalent in quality to the volume ray-casting result, the above relation must estimate the particle density function. The number of particles N in a volume cell is calculated as

$$N = \int_{\mathrm{Cell}} \rho dv \tag{14}$$

The particles are generated cell-by-cell. In each cell, particle locations are calculated stochastically in the local coordinate system, which may be one of a variety of types (e.g., barycentric coordinate). Because this technique generates particles with uniform sampling in each cell, blocky noise occurs in the rendering result. To solve this problem, Metropolis sampling for O-PBVR is presented [1]. Metropolis sampling, which uses a ratio of density at the current position to that at the candidate position, is widely used as an efficient Monte Carlo technique in chemistry and physics.

3.3.2 Particle projection

Using the aforementioned particle generation method, we can generate particles in a volume space according to the density function $\rho(x)$. By projecting these particles onto the image plane, we calculate the brightness values of the corresponding pixels. We also perform particle occlusion with the Z-buffer algorithm during this projection stage. This incorporates the effects of particle collision, which prevents some particles from reaching the image plane.

In the present method, we assume that the particles are completely opaque. Thus, neither alpha blending nor visibility ordering is required. However, when the number of projected particles is small, for instance one per pixel, it becomes difficult to produce a semi-transparent appearance. This problem can be solved by an ensemble average, that is, by accumulating a pixel value for particles generated multiple times and averaging their brightness values within the pixel.

3.4 I-PBVR

I-PBVR is comprised of three processes: cell projection, stochastic rasterization, and ensemble average. The first process decomposes a cell into several tetrahedral cells and splits each of the tetrahedral cells into a set of triangles on the projection plane. The second process renders the fragments of the triangles with a probability equal to the opacity value at each ray segment along a viewing ray. It projects particles onto the image plane, and the corresponding particle buffer stores the particles’ colors and depths. The third process repeats the first and second processes in sequence, N times, and calculates the resulting brightness values by averaging the accumulating color values.

3.4.1 Cell projection

Projected tetrahedra (PT) is a technique for rendering a tetrahedral volume dataset using polygonal approximation, which regards a tetrahedral cell as triangles. In this technique, first the tetrahedral cells are sorted in the order of distance from a viewing point. Second, each tetrahedral cell is projected onto an image plane and subdivided into three or four PT triangles. In the original PT technique, the color and opacity are evaluated at the vertices and rasterizing of the PT triangle generates the fragments on the image plane. Finally, the colors are accumulated to calculate the pixel value using the back-to-front algorithm. In I-PBVR, although the particle radius is not explicitly specified, it actually becomes a pixel scale on the image plane.

To improve the accuracy of the pixel value, a pre-integration technique, proposed by Engel [11], is often employed in the rendering stage. The technique calculates the color and opacity in the ray segment in a more precise way than the conventional technique which just samples a scalar value at the middle point of the ray segment. If the color or the opacity changes drastically in the ray segment, this sampling may miss the important feature. The pre-integration assumes that the scalar is linearly distributed in the ray segment. In this assumption, the integrand can be transformed from a function of the distance to that of the scalar. The pre-integration computes the lookup tables mapping three integration parameters (scalar value at the front triangle face s_f, back one s_b, and length of the segment l in Eq. 18) to the pre-integrated color C and opacity α. By considering many combinations of scalar and distance values, the pre-integration table is stored as 3D texture in GPU.

3.4.2 Stochastic rasterization

From Eq. 9, we can regard a brightness calculation model as an expected value calculation in which there are n ray segments along a viewing ray, and the k-th particle occurs at the probability of α_k. Thus, the brightness can be regarded as the expected value of the luminosity from the ray segment:

$$B = \sum_{k=1}^{n} P_k c_k \tag{15}$$

where the possibility that the k-th luminosity c_k is equal to the brightness value can be described as follows by using the opacity value α_k:

$$P_k = \alpha_k \prod_{j=1}^{k-1} \left(1 - \alpha_j\right) \tag{16}$$

This represents an event in which there is no particle from the first to the (k-1)-th ray segment, and there is more than one particle in the k-th ray segment. In this case, the brightness B becomes c_k since opaque and emissive particles are used. Please note that the brightness is not contributed to by the ray segments from the k-th to the last segments since the (k-1)-th particle completely occludes these segments.

3.5 Ensemble average

In both O-PBVR and I-PBVR, a stochastic approach is employed to generate particles that are projected onto an image plane. The generation is repeated to make the average of the pixel values, which can be viewed as an ensemble average. An ensemble is an imaginary collection of notionally identical experiments. In the ensemble average, the total brightness is calculated by averaging the pixel values in all of the repetitions. We confirmed the fluctuation of the total brightness follows a large number and evaluated the minimum repetition, 65,536, that makes the total variance become within half of the minimum discretized brightness in the worst case [12]. This result suggests that little improvement can be expected in the brightness value when the repetition number exceeds 65,536 in most cases. When we interact the volume rendering image with some transformation such as translation, rotation, or scaling, we think much of the interaction by reducing the repetition number at the cost of the image quality. When we intend to improve the image quality, we stay still without any interaction.

4. Particle-based fused rendering

In I-PBVR, we generate a particle in an interval of a tetrahedral cell by regarding the opacity as a cumulative distribution function as shown in **Figure 2**. The opacity

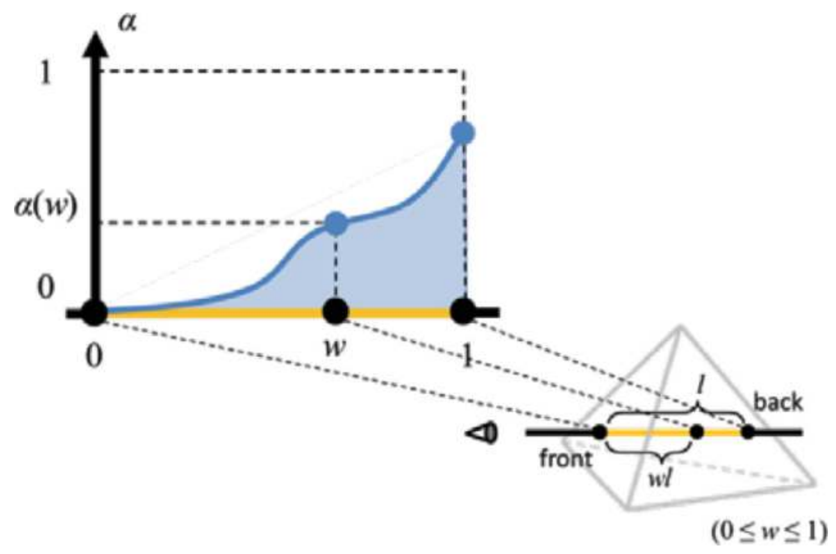

Figure 2.
Cumulative distribution function defined as the opacity between the entry point and the particle in the section where the viewing ray is cut into a tetrahedral cell.

can be represented as a function of a length from the entry point in the interval. Thus, the depth, which is the length from the entry point, can be regarded as a probability variable which follows a probability density function that is a derivative of the cumulative distribution function (CDF).

When we consider the definition of the opacity, we find that it describes the CDF of a probability variable, w, which shows a length from the entry point in the fragment interval. The probability density function, that is, its derivative, describes an exponential distribution, which matches the theorem that the number of particles follows the Poisson distribution since the exponential distribution describes the distance between particles in a Poisson process.

The opacity in Eq. 7 can be used to express the CDF α (w) of the random variable w as follows:

$$\alpha(w) = 1 - \exp\left(-\int_{t_{k-1}-wl}^{t_{k-1}} \tau(\lambda)\mathrm{d}\lambda\right) \tag{17}$$

Let l be the width of the section where the viewing ray is cut by the cell. This equation represents the opacity calculated between the entry point and the position where the particles are located in the entry. This interval can be expressed as wl using the random variable w (see **Figure 2**). Furthermore, the theorem that the probability density function is represented by an exponential function is consistent with theorem that “when the number of particles in a certain section follows the Poisson distribution, the particle spacing follows the exponential distribution.”

In the pre-integration method, the opacity in the section is described as follows.

$$\alpha(s_f, s_b, l) = 1 - \exp\left(-\frac{l}{s_b - s_f}\left(T(s_b) - T(s_f)\right)\right) \tag{18}$$

Here, s_f and s_b are scalar data values interpolated and computed at the entry and exit points of the section, respectively, and l represents the width of the section as described above. $T(s)$ represents an integral expression for calculating the number of particles generated in the interval.

$$T(s) = \int_0^s \tau(\lambda)\mathrm{d}\lambda \tag{19}$$

In the proposed method, it is assumed that the scalar data value is linearly interpolated in the interval, and it is expressed as s (w) at the point expressed using the random variable w. This assumption arises from the linear distribution of scalar data in line segments defined in tetrahedrons when interpolation calculations using scalar data and volume coordinates defined at each vertex are performed in the tetrahedrons.

$$s(w) = (1-w)s_f + ws_b \tag{20}$$

Therefore, in the case of $s_f \neq s_b$, the opacity function α (w) expressed by Eq. 17 is expressed as follows.

$$\begin{aligned} \alpha(w) &= \alpha\left(s_f, s(w), wl\right) \\ &= 1 - \exp\left(-\frac{l}{s_b - s_f}\left(T(s(w)) - T\left(s_f\right)\right)\right) \end{aligned} \tag{21}$$

In the case of $s_f = s_b$, the opacity function α (w) expressed by Eq. 17 is expressed as follows:

$$\alpha(w) = 1 - \exp\left(-\tau\left(s_f\right) \cdot wl\right) \tag{22}$$

Here, reference is made to the following derivation process.

$$\begin{aligned} \alpha\left(s_f, s_b, l\right) &= \lim_{s_f \to s_b} \alpha\left(s_f, s_b, l\right) \\ &= \lim_{s_f \to s_b}\left(1 - \exp\left(-\frac{l}{s_b - s_f}\left(T(s_b) - T\left(s_f\right)\right)\right)\right) \\ &= 1 - \exp\left(-T'\left(s_f\right) \cdot l\right) \\ &= 1 - \exp\left(-\tau\left(s_f\right) \cdot l\right) \end{aligned} \tag{23}$$

4.1 Calculation of depth value by inverse function method

In this method, particles are placed at a position wl away from the start point of the section. At this time, assuming that the random variable w follows the probability density function such that the cumulative distribution function is the opacity α(w), this variable w can be generated using the inverse function method. In the inverse function method, when the random number R exists in the range of the interval $[0, \alpha\left(s_f, s_b, l\right)]$, the variable w is calculated using Eqs. 24 or 25. Eqs. 24 and 25 are derived from Eqs. 21 and 22 when $s_f \neq s_b$ and $s_f = s_b$, respectively.

$$\begin{aligned} w &= \alpha^{-1}(R) \\ &= \frac{1}{s_b - s_f}\left(T^{-1}\left(-\frac{s_b - s_f}{l}\log(1-R) + T\left(s_f\right)\right) - s_f\right) \end{aligned} \tag{24}$$

$$\begin{aligned} w &= \alpha^{-1}(R) \\ &= -\frac{\log(1-R)}{\tau\left(s_f\right) \cdot l} \end{aligned} \tag{25}$$

Using Eqs. 24 and 25, we can calculate the depth value of the particle.

$$D(w) = (1 - w) \cdot D_f + w \cdot D_b \tag{26}$$

Here, D_f and D_b represent the depth values at the start and end points of the section. Similarly, scalar data values at particle positions can be interpolated and color values can be calculated using the transfer function.

This method was implemented using OpenGL and the GPU shader described in GLSL. For the implementation of pre-integration, we used two-dimensional pre-integration to perform error correction [13] with perspective transformation, so we implemented the function T described in Eq. 19 as a two-dimensional texture in the GPU [14]. To determine the random variable w described in Eq. 20, an inverse function of T is required, but in order to realize efficient computation, the function value was calculated in advance and this was also implemented in the GPU as a two-dimensional texture.

5. Result and discussion

Experiments were conducted to evaluate the effectiveness of this method. The experiment used CPU: Intel Core i7 3.3GHz, MEM: 16GB, and GPU: Intel Iris Graphics 550.

To confirm the appropriateness of the depth value in this method, experiments were conducted on two irregular volume data consisting of a single tetrahedral cell. **Figure 3** visualizes each irregular volume data using a red/blue monochrome color map that increases the color value according to the data value.

As a comparative experiment, as in the conventional method, the depth value in the tetrahedral cell is visualized by fixing the relative position in the section like the entry point, the middle point, and the exit point. It turns out that the proposed method realizes satisfactory visualization at the intersection of the two tetrahedral lattices. According to the result of the conventional method, a change in unnatural color value can be visually confirmed.

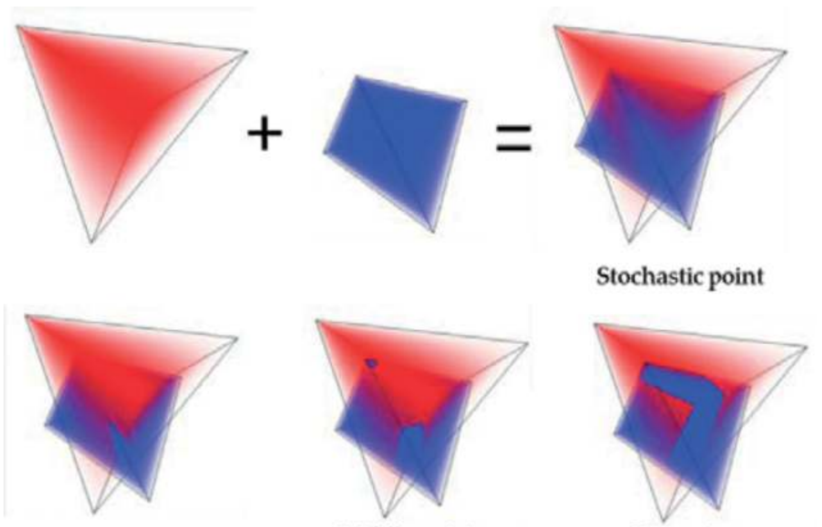

Figure 3.
Application example of proposed method for intersecting tetrahedral cell (red and blue color maps were used for each tetrahedral cell).

Experiments were conducted using two irregular volume data of appropriate size. These are obtained as a result of computational fluid dynamics calculation and are called "Tank." This calculation relates to a physical phenomenon when a pipe-like valve installed in a gas tank filled with a high-pressure state is instantly opened. Therefore, the important variables are pressure data and velocity absolute value data (both are scalar data). Pressure data and velocity absolute data were calculated using 9827 and 516 tetrahedral cells, respectively. **Figure 5** shows the volume rendering display of the fused mixed irregular volume data with different cells. In this experiment, red color is assigned to pressure data, and blue color is assigned to velocity absolute value data (**Figure 4**).

In **Figure 5**, we compare the proposed method (a), the previous methods (b), (c), and (d) in which the relative position in the section is fixed as the entry point, the middle point, and the exit point for the depth values in the tetrahedral cells and a method in which a random position is located between the entry and exit points (e). In the five figures, in addition to presenting the overall visualization result (grid line presence/absence) and the local visualization result (grid line presence/

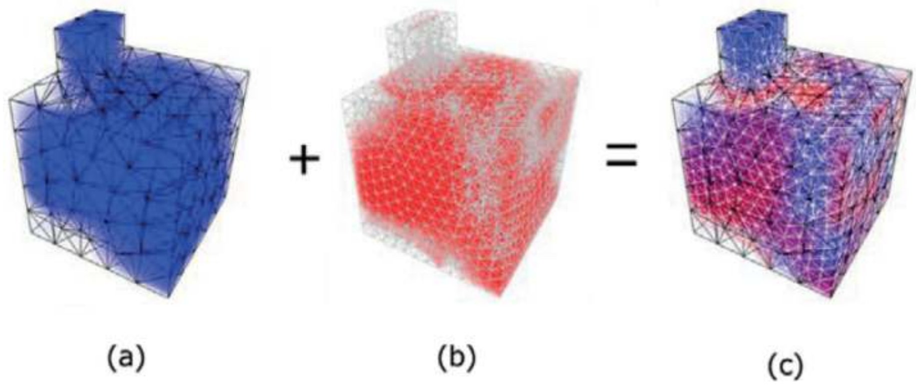

Figure 4.
Example of application to "Tank" data. (a) Velocity data defined by 516 cells, (b) pressure data defined by 9827 cells, (c) (a)+(b) fused visualization.

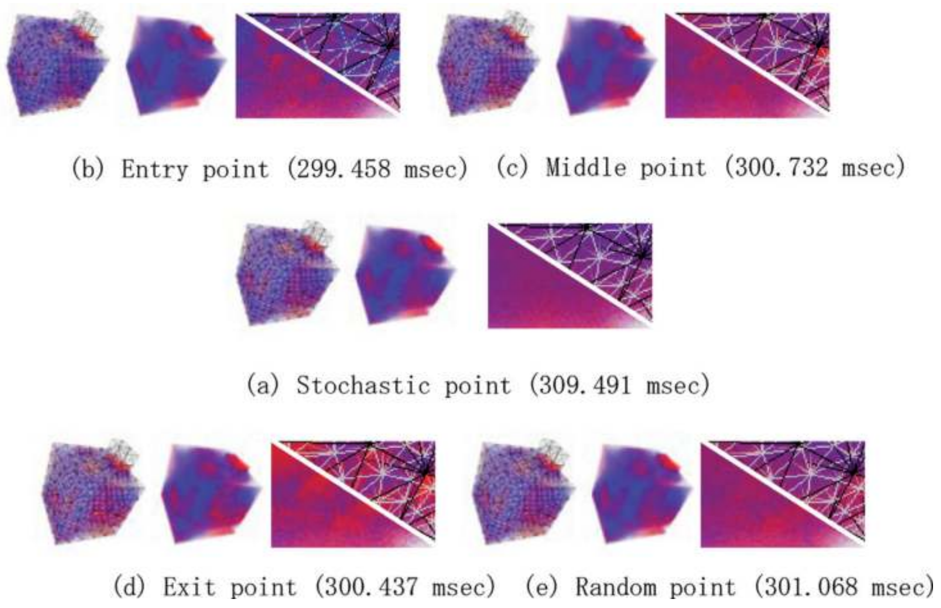
(b) Entry point (299.458 msec)
(c) Middle point (300.732 msec)
(a) Stochastic point (309.491 msec)
(d) Exit point (300.437 msec)
(e) Random point (301.068 msec)

Figure 5.
Comparison of application example to "Tank" data by the proposed method and conventional methods.

absence), the time required for the visualization result is described. Obviously, with the conventional method, it can be understood that artifacts due to improper setting of the depth value are visible in the visualization results. In particular, the trend is noticeable in the visualization results (b–d) where the depth value is set at the fixed point of the section. Even with the random position, it is more noticeable than that of the proposed method (e). Additionally, it can be seen that there is almost no difference in the calculation time required for the visualization.

6. Conclusion

In this chapter, we proposed a volume rendering algorithm method for multiple irregular volume data. In this method, the tetrahedral grid constituting the volume data is projected on the image plane, and the opacity is used to control the presence/absence of drawing at pixel expansion. To efficiently perform volume rendering of multiple irregular volume data, we developed a method for stochastically arranging particles in the section where the viewing ray is cut off by tetrahedrons. In this arrangement method, the particle position is calculated by inverse function method, considering the particle position as a random variable and the cumulative distribution function as opacity.

In the experiments for confirming the effectiveness of this method, we prepared two types of irregular volume data with different cells and confirmed the effectiveness of the proposed method in terms of the presence/absence of artifacts and calculation time at the intersection of the cells.

Although this time we concerned the proposal of the visualization method itself, we would like to use this method to elucidate the causal relationship between variables in important physical phenomena. For example, in order to clarify the influence of coherent vortices on heat transport in thermal fluid phenomena, it is necessary to combine scalar data representing a second invariant representing a vortex region and scalar data representing a heat flux absolute value related to heat transport. By using this method for the visualization of two kinds of time series irregular volume data, we would like to figure out a visual correlation between the multiple variables.

Author details

Koji Koyamada[1]* and Naohisa Sakamoto[2]

1 Kyoto University, Kyoto, Japan

2 Kobe University, Kobe, Japan

*Address all correspondence to: koyamada.koji.3w@kyoto-u.ac.jp

References

[1] Sakamoto N, Nonaka J, Koyamada K, Tanaka S. Particle-based volume rendering. In: Proceedings of Asia-Pacific Symposium on Visualization (APVIS 2007); 2007. pp. 129-132

[2] Sakamoto N, Kawamura T, Koyamada K. Improvement of particle-based volume rendering for visualizing irregular volume data sets. Computers & Graphics. 2010;**34**(1):34-42

[3] Sakamoto N, Kawamura T, Kuwano H, Koyamada K. Sorting-free pre-intergrated projected tetrahedra. In: Proceedings of the 2009 Workshop on Ultrascale Visualization 2009; 2009. pp. 11-18

[4] Blinn J. Light reflection function for simulation of clouds and dusty surfaces. Computers and Graphics. 1982;**16**(3): 21-29

[5] Sabella P. A rendering algorithm for visualizing 3D scalar field. Computers and Graphics. 1988;**22**(4):51-58

[6] Drebin RA, Carpenter L, Hanrahan P. Volume rendering. Computers and Graphics. 1988;**22**(4):65-74

[7] Levoy M. Display of surfaces from volume data. IEEE Computer Graphics and Applications. 1988;**8**(3):29-37

[8] Hansen C, Johnson C. The Visualization Handbook. Elsevier; 2005

[9] Silva C, Comba J, Callahan S, Bernardon F. A survey of GPU-based volume rendering on unstructured grids. Brazillian Journal of Theoretic and Applied Computing. 2005;**12**(2):9-29

[10] Lu Cai, Yuan Mingkai, Wang Qi, Kang Kun. Application of multi-attributes fused volume rendering techniques in 3D seismic interpretation. 2014:1609–1613

[11] Engel K, Kraus M, Ertl T. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In: Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware; 2001. pp. 9-16

[12] Sakamoto N, Koyamada K. Stoachastic approach for integrated rendering of volumes and semi-transparent surfaces. In: Proceedings of the 2012 Workshop on Ultrascale Visualization 2012; 2012

[13] Meredith J, Ma KL. Multiresolution view-dependent splat-based volume rendering of large irregular data. In: Proc. IEEE 2001 Symp. on Parallel and Large-Data Visualization and Graphics; 2001. pp. 93-155

[14] Westover L. Footprint evaluation for volume rendering. Computers and Graphics. 1990;**24**(4):367-376