Transactions, SMiRT-22
San Francisco, California, USA - August 18-23, 2013
Division IX (include assigned division number from I to X)

Multivariate Volume Rendering for FEM Simulation Results

Koji Koyamada[1] and Naohisa Sakamoto[2]

1Professor, The Institute for the of Liberal Arts and Sciences, Kyoto University, Japan, (koyamada.koji.3w@kyoto-u.ac.jp)

2Chief Engineer, The Institute for the of Liberal Arts and Sciences, Kyoto University, Japan, (naohisas@viz.media.kyoto-u.ac.jp)

ABSTRACT

In this paper, we propose a technique for a fused rendering of multiple simulation results. In a visual analytic environment, it is important to help scientists form a hypothesis in a scientific discovery. If we assume that the hypothesis is composed of an explained variable and one or more explanatory variables, the visual analytic environment needs to help select the latter variables in the simulation result. In this case, a fused volume rendering plays an important role as a multivariate visualization, but the visibility-sorting hampers its realization. To solve the problem, we employ a stochastic rendering to implement a sorting-free rendering technique. To confirm the effectiveness, we apply the technique to several volume data sets.

INTRODUCTION

Recently, multivariate visualization techniques have attracted more attention in visual analytic environment as shown in H. Akiba and K.-L. Ma (2007) since they are useful for hypothesis forming and testing and its application. Multivariate visualization techniques are important in hypothesis forming and testing using a large-scale simulation result. In general, a high-performance computer system generates a multivariate volume dataset defined on unstructured grids which are widely used in the finite element method (FEM). We assume that the hypothesis is composed of an explained variable and one or more explanatory variables. For example, a probability of cracks in a solid structure can be regarded as the explained variable which can be obtained by measurement results. Stress/strain tensors, temperature or displacement can be regarded as the explanatory variables which can be obtained from FEM simulation results. A formed hypothesis can be well tested by using information visualization. The information visualization which helps scientists investigate the relationship, such as parallel coordinates or scatter plot matrix, is now widely used. In this visualization, multiple variables are visualized in the variable space.

How the variables are selected for forming the hypothesis? To help scientists, multiple variables need to be visualized in a coordinate space as well as in the variable space. Such a visualization, which we call as a fused visualization, can be implemented using scientific visualization techniques such as volume rendering, iso-surfaces, and slice contours. In the fused visualization, the transparency attributes of the visualization objects need to be well utilized in order to effectively represent multiple variables in a single pixel. When the transparent objects are rendered, they are generally processed in the visibility order, which is difficult to be determined especially in case of unstructured grids.

In the hypothesis forming, it is important to select an appropriate set of variables in the coordinate space where the volume datasets are defined. For the selection, multivariate volume visualization techniques are promising since we can easily find locations in which multiple variables have a relation with each other. But, the development of such fused visualization techniques is regarded as a challenging research in the visualization community since the visibility sorting becomes difficult especially for unstructured grids.

In this paper, we proposed a fused volume rendering by using a stochastic rendering technique which does not require the visibility-sorting. To confirm the effectiveness of the proposed technique, we apply the fused volume rendering to unstructured grids. In the remaining parts, we first describe a visual analytics and a multivariate visualization which can facilitate a hypothesis forming. Then, we explain our stochastic rendering technique and its application to finite element simulation results.

VISUAL ANALYTICS

J.J. Thomas and K.A. Cook (2005) define visual analytics as the science of analytical reasoning facilitated by interactive visual interfaces. Today, data is produced at an incredible rate and the ability to collect and store the data is increasing at a faster rate than the ability to analyze it. Over the last decades, a large number of automatic data analysis methods have been developed. However, the complex nature of many problems makes it indispensable to include human intelligence at an early stage in the data analysis process. The visual analytics methods allow decision makers to combine their human flexibility, creativity, and background knowledge with the enormous storage and processing capacities of today's computers to gain insight into complex problems. Using advanced visual interfaces, humans may directly interact with the data analysis capabilities of today's computer, allowing them to make well-informed decisions in complex situations.

In the visual analytics process, a problem is first identified by observing phenomena in the world and then a hypothesis, that is a solution to the problem, is formed (Figure 1). If we assume that the hypothesis is composed of an explained variable and one or more explanatory variables, we need to select these variables in the problem domain. When we consider a problem which can be represented by a computational model, we can select the explanatory variables in the computer simulation result which includes a lot of variables. In general, we can select the explained variable in the measurement result of the phenomena which is modeled in the computer simulation. Once we select the explanatory variables which the hypothesis is composed of, we can employ some data mining or information visualization techniques in order to test it. In general, it is difficult to select appropriate explanatory variables among a lot of variables which are generated from the simulation result. In this phase, a multivariate visualization plays an important role.

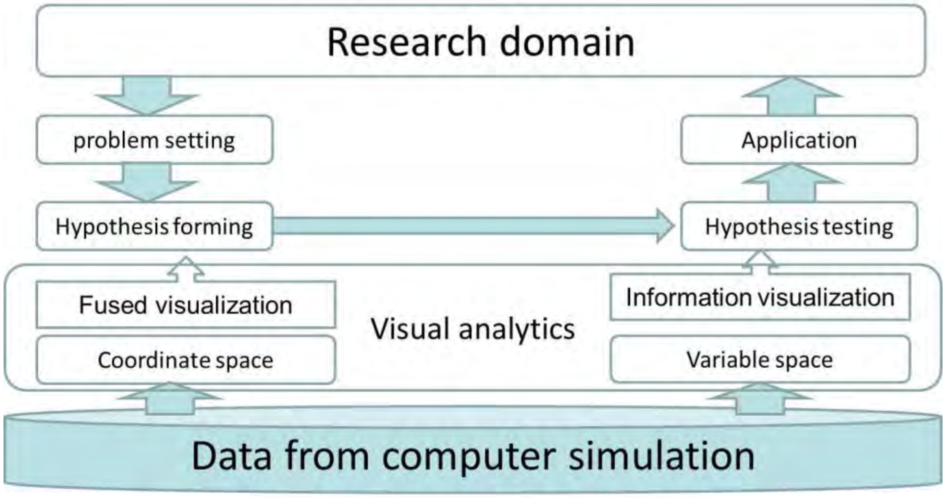

Figure 1. Visual analytics process

.

MULTIVARIATE VISUALIZATION

To support scientific knowledge discovery and hypothesis testing on data sets of ever increasing size and complexity, visualization community focuses on the multivariate visualization. The multivariate visualization is an active research field with numerous applications in diverse areas ranging from science communities and engineering design to industry and financial markets, in which the correlations between many attributes are of vital interest.

The terms multidimensional and multivariate are often used vaguely. Strictly speaking, multidimensional refers to the dimensionality of the independent dimensions while multivariate refers to that of the dependent variables. The more appropriate term for multivariate data visualization should be multidimensional multivariate data visualization. Nevertheless, a set of multivariate data is in high dimensionality and can possibly be regarded as multidimensional because the key relationships between the attributes are generally unknown in advance. The multidimensional property is therefore implied in common usage.

Multivariate visualization techniques are important in hypothesis forming and testing which are essential components in the scientific discovery. As for the hypothesis testing, information visualization techniques are often used to identify the relationships in multivariate linear models. The model is composed of multiple variables that are selected in advance. When we consider a numerical simulation result in the scientific discovery, we think that fusing of visualization results is important for the selection of the multiple variables. In this case, a volume rendering technique is suitable, but there is no adequate technique for volume rendering of unstructured grids which are often used for the finite element method. In the following discussion, we call the multivariate volume visualization as a fused visualization.

STOCHASTIC RENDERING

To realize the fused visualization techniques, we focus on a stochastic rendering, which is developed by us in order to integrally handle multiple objects including volumes and surfaces without visibility sorting. It is actually used to visualize a large scale simulation result generated by the K computer as shown in A. Ogasa et. al (2012). To describe the technique, we revisited a brightness equation in the volume rendering algorithm and reconsider the definition of opacity which is usually derived from a user-specified transfer function.

Volume Rendering

In volume rendering, a density emitter model proposed by P. Sabella (1988) is often referenced. In this model, a small cylinder with a cross section area A and a dt distance is considered along a viewing ray. In the cylinder, it is assumed that opaque and emissive particles follow the Poisson distribution at the density ρ, which represents a number of particles in a unit volume. The Poisson distribution also assumes that the particles with the same radius r do not overlap each other in the cylinder. The luminosity of a particle, that is particle color, is represented as c(t) which shows an energy per unit area. The model expresses a difference in a brightness value, which is a pixel value, between both cross sections by using the absorption and the emission effect caused by particles.

$$\begin{aligned} dB(t)A &= -absorbed + emitted \\ &= (-B(t) + c(t)) \times \pi r^2 \rho(t) A dt \cdot \\ \frac{dB(t)}{dt} &= (-B(t) + c(t)) \times \pi r^2 \rho(t) \end{aligned} \qquad (1)$$

The density emitter model is in a form of a simple ordinary differential equation, which can be solved explicitly. The pixel value can be calculated as an integral in the interval from t_n to t_0 along a viewing ray. Here, we assume that the radius of the viewing ray is the same as that of the particle. The term $c(t)\pi r^2\rho(t)$ represents a total particle color generated from particles in the interval with length *dt*. The exponential term shows a probability that the particle color can arrive at the viewing point. The exponent shows a negative number of particles in the interval from *t* to t_0 along a viewing ray. This interval spans a distance from the reference point t to the viewing point.

$$B = \int_{t_n}^{t_0} c(t) \times \pi r^2 \rho(t) \times \exp(-\int_{t}^{t_0} \pi r^2 \rho(\lambda) d\lambda) dt \cdot \qquad (2)$$

$$B = \sum_{i=1}^{n} c_i \times (\alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)) \qquad (3)$$

This leads to two approaches, object space approach and image space approach. In the former one, we define a density function of emissive opaque particles. According to the density function, we generate particles in a given volume data set and project them onto an image plane. Since we use an opaque particle, no visibility sorting is required. In the latter approach, we regard the brightness equation as the expected value of the luminosity of a sampling point along a viewing ray, and we propose a sorting-free approach that simply controls the fragment rendering by using the evaluated opacity value to calculate a rendered image.

Definition of opacity

In order to numerically calculate a brightness value, the viewing ray is subdivided into a number of ray segments. In each ray segment, a new parameter, an opacity value is introduced to simplify the brightness equation. The exponent represents a negative number of particles in the ray segment. The exponential function of the negative number of particles is called as transparency, which will be given a stochastic meaning to. The opacity is defined as one subtracted by the transparency. If we assume that the length of the ray segment is subtle, the density function can be described using the opacity, the length and the particle radius.

$$\begin{aligned}\alpha_k &= 1-\exp\left(-\int_{t_k}^{t_{k-1}} \pi r^2 \rho(\lambda)d\lambda\right) \\ &= 1-\exp\left(-N_{p_k}\right)\end{aligned} \qquad (4)$$

In the density emitter model, the Poisson distribution has been assumed for the number of particles in a specified volume. If the expected number of particles in this volume is N_p, then the probability that there are k particles can be expressed in the formula. N_p can be calculated by integrating the density function in the volume, which is the ray segment. In the volume rendering, it is important to evaluate the probability that there is no particle which occludes the color of the particle which is located farer from the viewing point. By substituting k with zero, we can calculate the probability, which is equivalent to the transparency.

$$P(N=k)=\frac{\exp(-N_p){N_p}^k}{k!}. \qquad (5)$$

$$0 \le P(N=0)=\exp(-N_p) \le 1 : \mathit{transparency} \qquad (6)$$

$$0 \le \alpha = 1-\exp(-N_p) \le 1 : \mathit{opacity} \qquad (7)$$

Alpha-blending formula

In the simple implementation of the volume rendering, we first cast a viewing ray into a volume and then interpolate the scalar value at the center of the ray segment. The interpolated scalar is transformed into color and opacity by referring to a transfer function from a scalar value to a color or an opacity value. In many cases, the transfer function is specified by the user and the color is composed of three elements; red, green and blue. By using the discretized brightness equation, the pixel value can be calculated in order of the squared number of ray segments at each viewing ray.

$$B^{k-1}=\sum_{i=k}^{n} c_i \times (\alpha_i \prod_{j=k}^{i-1}(1-\alpha_j)) \qquad (8)$$

$$B^{k}=\sum_{i=k+1}^{n} c_i \times (\alpha_i \prod_{j=k+1}^{i-1}(1-\alpha_j)) \qquad (9)$$

If we transform the scalar into the color and the opacity in the descending order of the distance from the viewing point, we can adopt a back-to-front algorithm the performance of which can be estimated as an order of the number of the ray segments. Most volume rendering techniques employ the back-to-front algorithm using the following recurrence formula. If we calculate an intermediate pixel value, B_k, by taking consecutive (n-k) ray segments from the back end and we add the k-th ray segment adjacent to the ($k+1$)-th ray segment, the intermediate pixel value, , can be represented by using the alpha-blending formula.

$$B^{k-1}=c_k\alpha_k+(1-\alpha_k)B^k. \qquad (10)$$

Cell Projection

As shown in P. Shirley and A. Tuchman (1990), Projected tetrahedra (PT) is a technique for rendering a tetrahedral volume dataset using polygonal approximation, which regards a tetrahedral cell as triangles (Figure 2). In this technique, first all the tetrahedral cells are sorted in the order of the 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 the rasterizing the PT triangle generates the fragments of the image plane. Finally, the colors are accumulated to calculate the pixel value using the back-to-front algorithm.

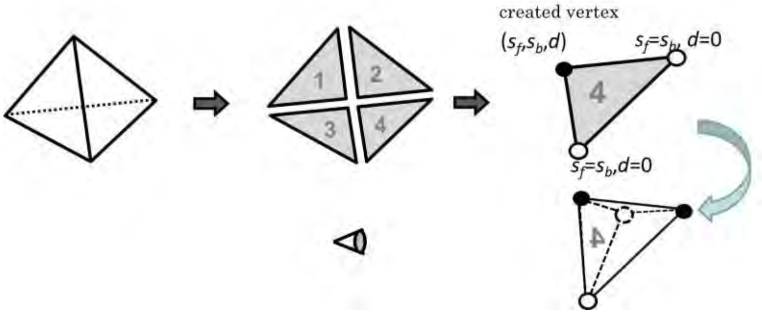

Figure 2. Subdivision into four PT triangles.

To improve the accuracy of the pixel value, a pre-integration technique, proposed by K. Engel, M. Kraus, and T. Ertl (2001), 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 d, 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 (Figure 3).

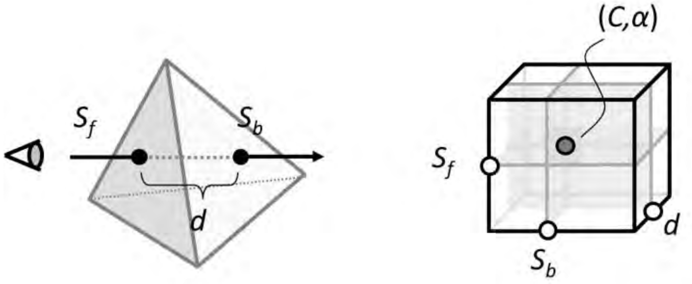

Figure 3. Pre-integration with 3-D texture..

Stochastic Approach

To design a sorting-free rendering algorithm, we just refocus on the discretized brightness equation which calculates a brightness value at each pixel by using colors and opacities along a viewing ray. As the length of the ray segment is not necessarily in the same equation, the segment can be a part of a volume cell, a polygon or a polyline. If we regard the color and the rest term of opacities as a probability variable and its probability, the brightness can be assumed as an expected value of the colors. Please note that the opacity is a probability that there is one or more opaque particle in the ray segment. We consider a probability event that a ray segment occurs at the probability of the opacity along a viewing ray.

$$B = \sum_{k=1}^{n} c_k P_k \qquad (11)$$

In the event, the probability that the color of the k-th ray segment arrives at the viewing point is P_k, which represents the product of the k-th opacity and the total transparencies of the (k-1) ray segments in the front of the k-th segment.

$$P_k = \alpha_k \prod_{j=1}^{k-1} (1-\alpha_j) \qquad (12)$$

Every day, we experience a scene which includes semi-transparent objects such as clouds, fires and fogs which are composed of opaque particles. In a moment, we see the color of one of the opaque particles. At the next moment, we see that of another one. We recognize the semi-transparency by averaging the colors which we see during the moments within an optical time resolution of the human's eye. If the recognition obeys the Ergodic theory, we can use an ensemble average of brightness values at all the probability events. In each event that is ensemble member, we control the occurrence of the ray segment at the probability of the opacity.

As we regard the brightness calculation as a probability event, we need to estimate a number of ensemble members which minimizes the brightness variance. As we can assume that the color and the probability are within the interval [0,1], the maximum of the brightness deviation becomes 1/2 that is a worst case. In this case, when we increase the number of the ensemble members, the deviation of the averaged brightness decreases inversely proportional to the square root of the number. If we represent a brightness value in 256 gradations, the minimum number of ensemble members can be estimated so that the deviation becomes smaller than 1/256

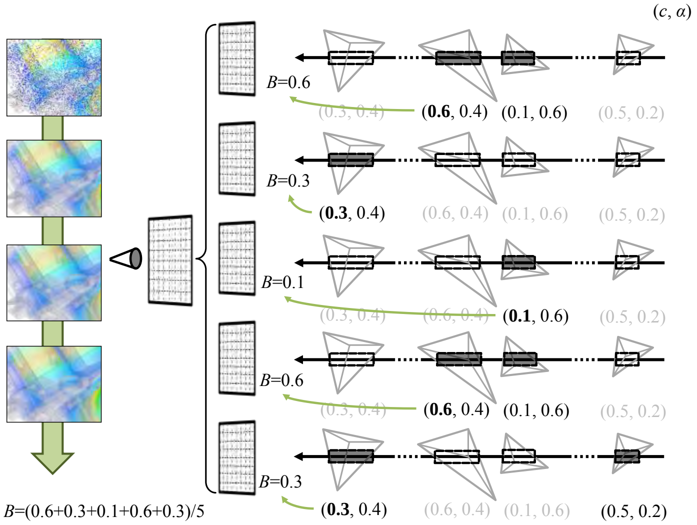

Figure 4. Ensemble average of brightness.

Based on the above discussion, we can design a stochastic rendering which does not require the visibility sorting of objects. First, we project objects onto an image plane. Second, we rasterize them to generate fragments and evaluate their opacities. By regarding the opacity as the probability and referring to the depth buffer, the fragment color is transferred to the frame buffer as a pixel value. An ensemble member is composed of the first, the second processes and the resulting image. After repeat the processes by adequate numbers which can be determined by referring to the minimum number of ensemble members discussed above, the ensemble average is employed to generate the final pixel value (Figure 4).

Our stochastic rendering uses the PT and the pre-integration to handle tetrahedral volume cells. The second process in the stochastic rendering includes the retrieval of colors and opacities in the 3D texture constructed in advance. As two scalars and a distance are defined at each vertex of the PT triangle, the fragment has the interpolated values. By using the interpolated scalars and the distance, the color and opacity can be retrieved. The retrieval is processed in the fragment shader by referring to the 3D texture that is the pre-integration table. We use a random number, a, to accept or discard the color. If the random number is smaller than the opacity, we accept the color. Otherwise, we discard the fragment (Figure 5).

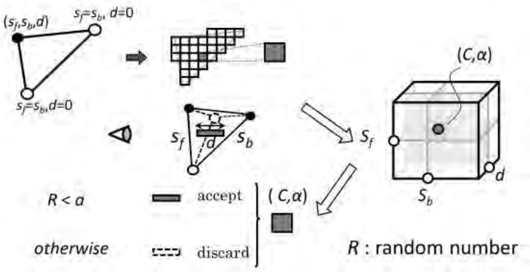

Figure 5. Use of the pre-integration table in PT

In the stochastic rendering of tetrahedral cells, the fragment has a property of its length which represents a distance between the entry and the exit points of the viewing ray. In the case of polygons and polylines, the length is zero. When we handle a single tetrahedral volume data set, the fragment has no overlapped region with another fragment. To check the occlusion of the fragment from a tetrahedral cell using the depth buffer, we can store any depth value between the entry and the exit points. Thus, in order to handle a single tetrahedral volume in the stochastic rendering, the depth buffer to store a single depth value is enough.

EXPERIMENTS

The proposed stochastic rendering technique has been applied to the fused volume rendering of unstructured grid volume data sets. To confirm its effectiveness, we conduct several experiments in order to verify the effectiveness of our proposed method. All experiments are performed on a PC with an Intel Core2 Quad 2.83 GHz CPU, 8 GB of RAM, and NVIDIA GeForce GTX 580 1.5GB GPU. And all results of rendering performance evaluations are shown by the repetition level of one since the rendering time of our method is proportionally increased as a number of repetitions increasing (Figure 6).

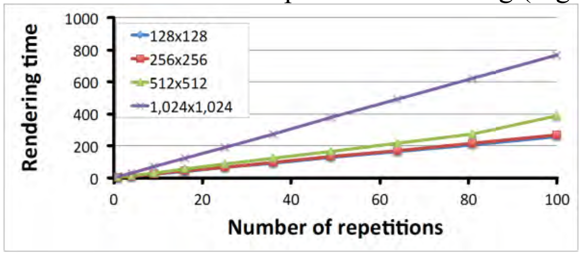

Figure 6. Rendering time of the V6 engine data with different repetition levels

Results of a single volume

In order to verify the image quality and rendering performance, we visualize the V6 engine dataset with the proposed techniques, and show the result in Figure 7. The obtained rendering speed per single repetition is 3.39 milliseconds. Figure 8 shows the result of error evaluation of the rendering images. Here the error is defined as the maximum absolute difference value [0,1] between pixels on the ideal image rendered with the repetition level of 30,000 and corresponding pixels.

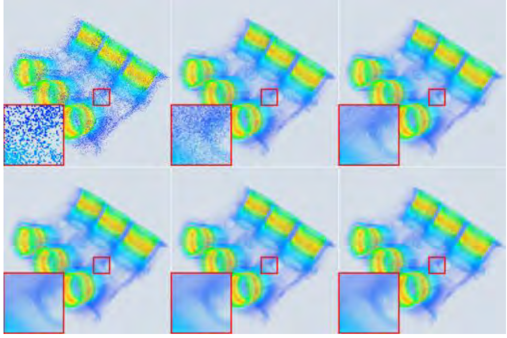

Figure 7. Rendering results (Repetition level: 1, 10, 100, 1000, 10,000, 30,000)

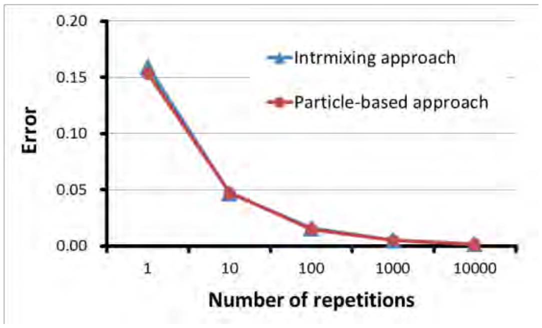

Figure 8. Error evaluation of the intermixing and particle-based approaches

Results of two volumes

In order to verify the effectiveness of the fused rendering, we apply the technique to two volumes which contain the misses stress and the absolute displacement scalars. To clearly figure out one scalar from another, we assign warm and cold colors to the former and latter scalars, respectively. In Figure 9 (a) and (b), we show the rendering results of the V6 engine data composed of 282,128 tetrahedral cells and their boundary surfaces. As shown in Figure 8 (c), we can understand how two scalars are related to each other.

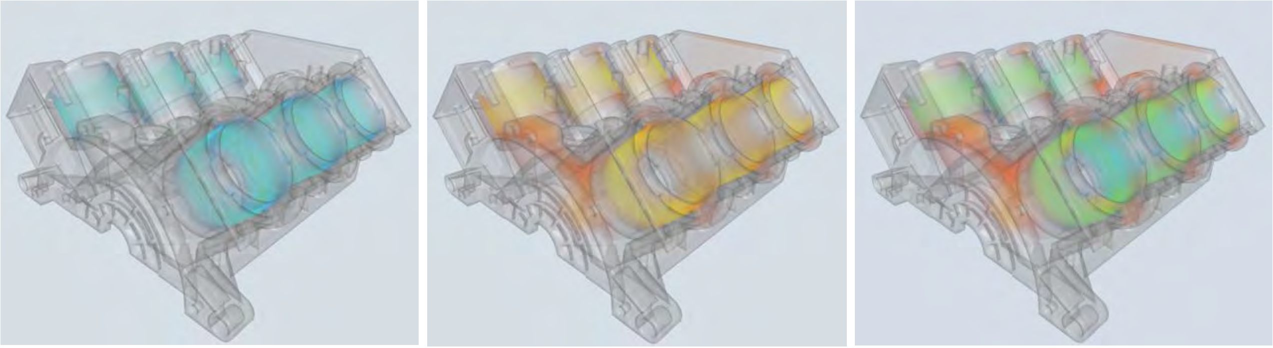

Figure 9. The rendering results of the V6 engine data composed of 282,128 tetrahedral cells and their boundary surfaces. The distributions of (a) misses stress values and (b) absolute values of displacement, and (c) the fused visualization result of (a) and (b).

CONCLUSION

In this paper, we have proposed a fused visualization technique for rendering multiple unstructured grid volumes using a stochastic approach. This technique calculates a brightness as the expected value of the luminosity from the ray segment and can improve the accuracy of volume rendering that is related to the repetition process. Since the main point of this technique is sorting-free, it enables to realize a multivariate visualization of unstructured grid volumes, which can facilitate a hypothesis forming. In the future, we will apply this fused visualization technique in the real world problem.

REFERENCES

H. Akiba and K.-L. Ma (2007). “A tri-space visualization interface for analyzing time-varying multivariate volume data,” In Proceedings of Eurographics/IEEE VGTC Symposium on Visualization, pages 115 -122.

J.J. Thomas and K.A. Cook (2005), Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Press.

A. Ogasa et. al (2012), "Visualization Technology for the K-computer," Fujitsu Scientific Engineering Journal, 48 (3), pp. 348-356.

P. Sabella (1988), A Rendering Algorithm for Visualizing 3D Scalar Field, Computer Graphics, 22 (4), pp. 51-58.

P. Shirley and A. Tuchman (1990), A Polygonal Approximation to Direct Scalar Volume Rendering, In Proceedings of San Diego Workshop on Volume Visualization, pp.63-70.

K. Engel, M. Kraus, and T. Ertl (2001), High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading, In Proc. Of Eurographics/SIGGRAPH Workshop on Graphics Hardware, pp. 9-16.

COPYRIGHT TRANSFER AGREEMENT

The below text in italic is for your information and agreement prior to submittal of the paper.

The author(s) warrants that the submitted manuscript is the original work of the author(s) and has never been published in its present form.

The Lead Author, with the consent of all other authors, by submitting the manuscript for publication in SMiRT-22 transactions, hereby transfers copyright interest in the submitted manuscript to IASMiRT subject to the following.

- *The Lead Author and all coauthors retain the right to revise, adapt, prepare derivative/ expanded works, present orally, or distribute the work.*
- *In all instances where the work is prepared as a "work made for hire" for an employer, the employer(s) of the author(s) retain(s) the right to revise, adapt, prepare derivative/expanded works, publish, reprint, reproduce, and distribute the work provided that such use is for the promotion of its business enterprise and does not imply the endorsement of IASMiRT.*
- *It is recognized that an author who is a U.S. Government employee and who has participated in the submitted work does not own copyright in it.*

Note: If the manuscript is not accepted by IASMiRT or is withdrawn prior to acceptance by IASMiRT, this copyright transfer will be null and void.