# CELL-PROJECTION PARALLEL VOLUME RENDERING WITH EARLY RAY TERMINATION

Motohiro Takayama, Shin-ichiro Mori, Yuki Shinomoto, Masahiro Goshima, Yasuhiko Nakashima, Shinji Tomita
Kyoto University, Sakyo-ku, Kyoto 606-8501 JAPAN
email:revolver@lab3.kuis.kyoto-u.ac.jp

## ABSTRACT

A cell-projection parallel volume rendering system for unstructured grid volume data is proposed in this paper. For this system, the modified early ray termination scheme is proposed to prune the cell-projection of invisible cell. In order to alleviate load imbalance due to view-dependency in scan-conversion and the dynamic behavior of early-ray termination, the authors also implement dynamic load balancing mechanism into their system. Preliminary evaluation of this system shows the 4.32-times performance improvement compared to the system without early ray termination and dynamic load balancing.

## KEY WORDS

Volume Rendering, Unstructured Grid, Dynamic Load Balancing, Parallel Processing, Early Ray Termination, Cell-Projection

## 1 Introduction

PC-cluster based large-scale simulation has been rapidly increasing in popularity. We have noticed the strong demand for simultaneous visualization of large-scale simulation results on a PC cluster. Indeed, we have already been doing research on hardware acceleration techniques for structured grid volume rendering[1]. Now, we are going to tackle the unstructured grid volume rendering for simultaneous visualization of large-scale simulation results on a PC cluster. The volume rendering of unstructured grid volume data can be implemented by direct volume rendering or indirect volume rendering algorithms.

The indirect volume rendering algorithm converts the unstructured grid volume data into structured-grid volume data and then performs direct volume rendering with this structured grid volume data. Due to the regularity of the converted volume data, this direct volume rendering can benefit from hardware acceleration. On the other hand, however, it may suffer an unacceptable increase of data size in general.

Direct volume rendering algorithms can be classified into either ray-casting or projection methods. Projection methods may be further categorized as cell-projected[2], slice-projected, or vortex-projected schemes[3]. In this paper, we have focused on the cell-projection scheme, and we discuss its parallel implementation.

This paper is organized as follows. The next section introduces cell-projection volume rendering and its parallel implementation. In Section 3, we propose the conservative early ray termination scheme to prune unnecessary scan conversion of cells. Section 4 shows some experimental results, followed by the conclusion in Section 5.

## 2 Cell-Projection Volume Rendering

In the cell-projection(CP) scheme[2], unstructured grid volume data is rendered by the following three steps(Figure 1). For simplicity of discussion, we assume the cell to be a tetrahydra.

1. **Projection Phase**: Project a given three-dimensional data(cell) into a two-dimensional screen and find the projection area($R$) for each data(cell).
2. **Scan Conversion Phase**: Perform scan conversion for each projected cell. More precisely, for each pixel $P(x, y)$ in the projection area($R$) on the screen, calculate the cell's contribution(color(RGB) and opacity($\alpha$)) to pixel $P(x, y)$ and depth values ($z_{front}\ and\ z_{back}$) for both the front and back intersection points where the ray corresponding to pixel $P(x, y)$ intersects the cell. In the rest of this paper, we refer to the data structure consisting of these four parameters (RGB, $\alpha$, $z_{front}\ and\ z_{back}$) as ray segment. As the result of the scan conversion of a cell, ray segments for each pixels in the projection area($R$) are computed. For every cell, compute its ray segments. After that, for each pixel on the screen, gather the ray-segments corresponding to the pixel and make a depth-sorted list of these ray segments (Figure 2).
3. **Composition Phase**: Given the ray-segment lists for all pixels on the screen, calculate the pixel value(color) by compositing the depth-sorted ray segments in the ray-segment list from front to back by alpha blending using Porter-Duff's over operation[4]. Now, we finally obtain the volume rendered image.

Here, we have to note that, even in the scan conversion phase, one may perform composition of the neighboring ray segments in the ray-segment list for a ray if these two ray-segments are derived from the neighboring cells contacting each other along the ray. We refer to this composition in the scan conversion phase as partial composition(Figure 3).

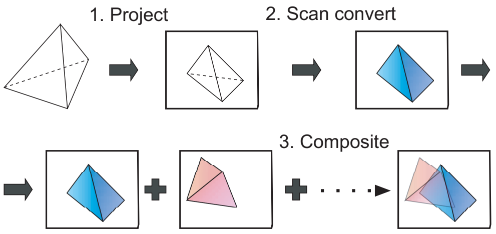


Figure 1. Cell Projection

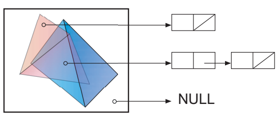


Figure 2.
Ray-Segment List

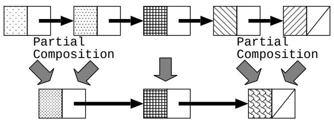


Figure 3.
Partial Composition

The resultant of the partial composition of two ray segments is again a ray segment. If we can apply partial composition successfully when a new ray segment is inserted into the ray-segment list, we can reduce the length of the ray-segment list and list manipulation overhead due to the length of the list.

Moreover, the opacity of a partially composed ray segment is higher than the opacity of each ray segments used in this partial composition. This feature greatly helps us to develop the optimization scheme described below.

## 3 Parallel Implementation

### 3.1 Fundamental Implementation

As we have mentioned before, the goal of our research is the simultaneous visualization of simulation results on a PC cluster. So, without a lack of generality, we can assume that unstructured volume data has already been distributed among the nodes of the PC cluster. However, we have to remember that the best data distribution pattern for the simulation may not be the best pattern for the visualization. This is the starting point of our parallel volume rendering(PVR) implementation. Thus, it is desirable to assume neither the preprocessing based technique like visibility sorting nor the shared memory based implementation[5].

In the following discussion, we assume the system has $N$ working nodes(**WN**s) for computation and one control node(**CN**) for global management of the load distribution and user interface, including final image output.

Once the data(cell) has been generated and distributed among the working nodes (WNs), the computation for the projection and scan conversion phases can easily be parallelized since there is no essential dependency between these computations for each cell. So, as the first step, each WN performs the projection and scan conversion of the cells which are assigned to the WN and generates its own ray-segment lists.

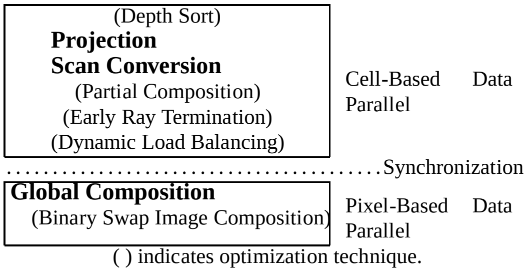


Figure 4. Parallel Implementation

When all WNs complete this computation, the composition phase starts as the second step. Now, we can utilize the pixel-level parallelism, so we assign the computation for a certain region of the screen to each WNs and perform the parallel composition as follows. For each pixel in the assigned region, each node 1) gathers from other nodes the ray-segment lists corresponding to the pixel, 2) merges them into a single depth-sorted ray-segment list, and then 3) calculates the pixel value(color) by compositing the depth-sorted ray-segments in the ray-segment list from front to back by alpha blending using Porter-Duff's over operation[4]. Now, each nodes finally obtains the volume rendered image for its own region. Then, each WNs sends its image to the control node to output the overall volume rendered image onto the display. Though we cannot explain the detailed implementation of the global composition phase due to the page limit of this paper, we have adopted the Binary-Swap Image Composition(BSC) scheme[6] to reduce both the communication overhead and the load imbalance.

We also refer to this composition phase as the final composition phase or global composition phase if we need to distinguish it from the partial(local) composition in the scan conversion phase.

Our fundamental parallel implementation itself is quite simple. The most important feature of our fundamental implementation is that it aggressively applies the partial composition in the scan conversion phase so that it can reduce the data size of each ray-segment lists required for exchange in the final composition phase.

The previous work by Ma et al.[3] proposed the parallel implementation which executes both the scan conversion and (global) composition processes concurrently. Matsui et al. [7] have also adopted a similar technique for their structured-grid parallel volume rendering system. We also considered adopting this idea. But our conclusion has been negative so far, because 1) it may incur the problem of frequent asynchronous interprocessor communication, 2) our partial composition scheme may reduce the cost of global composition, and also it can be thought of as a concurrent execution of the projection phase and the composition phase, and furthermore, 3) we are currently developing a technique like pipelining which may hide the cost(latency) of the global composition in some other work.

## 3.2 Optimizations

### 3.2.1 Approximate Depth Sorting

By adopting partial composition, if the program could process the scan conversion of the cells in depth-sorted order, it could reduce the length of the ray-segment list, and thus it could reduce the list manipulation cost at the same time. However, the depth order of cells may change pixel by pixel, in general. So, it is difficult, or rather impossible, to determine a perfect depth order of cells for volume rendering.

Our solution to this issue is approximate depth sorting which sorts the cells in the depth order of their center of gravity. Instead of determining the near-perfect depth order for paying the computational cost for both sorting and list manipulation, as in an octree search, we chose a rather simple and less costly sorting scheme, since the order of cells should be recalculated every time the viewpoint changes. Here, we have to mention that this approximation doesn't cause any quality error in the final image.

### 3.2.2 Early Ray Termination

Early ray termination(ERT)[8] is an optimization technique proposed for front-to-back ray-casting volume rendering. The concept of ERT is that the objects which are located behind less transparent objects (voxel, cell, and so on) may have very few or no contributions to the final pixel color even in volume rendering, thus, it is possible to terminate the composition computation along the ray before the ray passes through the volume data space. Therefore, the ERT contributes significantly to the reduction of the volume rendering cost, though its effect depends on the opacity of each objects.

Since ERT is a pixel-based optimization technique, it is easily implemented in the composition phase of our algorithm. However, the most time-consuming process in cell-projection parallel volume rendering(CP-PVR) is the scan conversion time. Thus, if we could introduce the concept of ERT into the scan conversion phase, we could reduce the computation time much more. The conservative early ray termination scheme, which we used to call ERT-table scheme[9], has been proposed for this purpose. The section 4 discusses this issue.

### 3.2.3 Dynamic Load Balancing

In the parallel implementation of the Cell-Projection Parallel Volume Rendering, we have to pay attention to the load imbalance due to 1)the initial data distribution, 2) the view dependency of scan conversion time, and 3)the run-time dynamic behavior of the ERT.

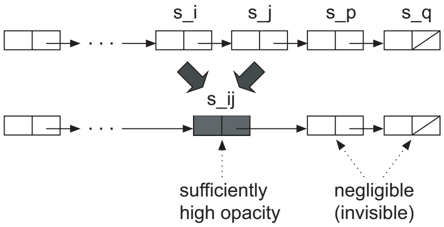


Figure 5. ERT using Partially Accumulated Opacity

The preprocessing-based static load balancing schemes like [3, 10] cannot deal with the third source of load imbalance. The self-scheduling based dynamic load balancing scheme[5] does not work well for the non shared memory system like PC cluster. Thus, in order to solve these three sources of load imbalance, we choose the distributed work-stealing scheme as the dynamic load balancing(DLB) scheme[11].

## 4 Conservative Early Ray Termination

### 4.1 The Concept of the Conservative Early Ray Termination

In order to apply the concept of ERT into the scan conversion phase, we have integrated the following two concepts into our conservative early ray termination scheme (CERT); 1) visibility checking based on the partially accumulated opacity, and 2) subscreen based comparison for the pruning of unnecessary scan conversions.

To explain the first concept, let's assume that we have been given an ray-segment list for a pixel as shown in the Figure 5. In this list, we notice the four ray segments ($s_i, s_j, s_p, and s_q$). By performing the partial composition with $s_i and s_j$, a new ray segment $s_{ij}$ is generated in the list below in this figure. If the calculated opacity of $s_{ij}$, which is what we call the partially accumulated opacity, is sufficiently high, the remaining ray segments ($s_p and s_q$ in this figure) linked behind $s_{ij}$ do not contribute anything to the pixel, thus, they can be negligible. It means that the decision of the visibility of the cell can be done regardless of the status of ray segments linked in front of the partially composed ray segment. This feature makes us possible to apply the concept of ERT in the scan conversion phase, because it is impossible to successively accumulate the opacity from the front of the ray-segment list to a certain ray segment during scan conversion phase.

Now, we can redefine the condition to neglect the scan conversion of a cell $c$ as follows: (C1) there exists a non-empty ray-segment list for all the pixel corresponding to the projection area $R$ for the cell $c$, and (C2) there exists a partially composed ray segment whose opacity is sufficiently high in each of these ray-segment lists, and (C3) at least one of such ray-segments in each of the lists is located in front of the cell $c$.

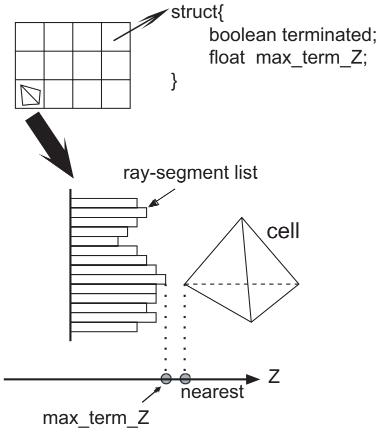


Figure 6. ERT in the scan conversion phase

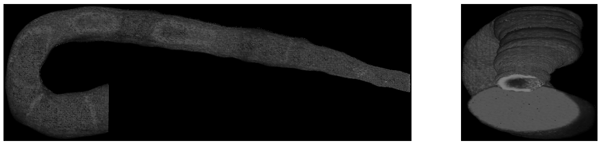
However, this condition still requires the per pixel based search on the ray-segment lists which may incur a significant overhead. In order to alleviate this overhead, we have introduced a small look up table, which we call the ERT-table. Here, we assume that the screen is subdivided into subscreens. Then, the ERT-table represents the status of each subscreen. Each entry of the ERT-table holds 1) a one bit information whether the conditions C1 and C2 are satisfied corresponding to each subscreen $S$, instead of the projection area of each cell, and if they are satisfied, 2)the maximum depth value ($max\_term\_Z$) which is the rear-most depth among all the subscreen's per-pixel-front-most partially accumulated ray-segments whose opacity exceed the threshold.

With this ERT-table, the decision of whether the scan conversion of a cell is required is made as follows; 1)After computing the projection area($R$) of a cell, check the corresponding entry in the ERT-table if the conditions C1 and C2 are satisfied, and if they are satisfied, 2)compare the ($max\_term\_Z$) and the depth value of the nearest vertex in the cell($nearest$). If $max\_term\_Z < nearest$, then the remaining process concerning the cell can be terminated.

Figure 6 illustrates the concept of the conservative early ray termination scheme.

An important feature of the conservative early ray termination scheme(CERT) is that it checks only a sufficient, though not necessary, condition for the termination. This approximation may possibly reduce the effect of ERT, thus we call it conservative, but it significantly reduces the cost to maintain the information in the ERT-table and therefore it becomes possible to apply the ERT technique in the scan conversion phase.

We should mention again that the partial composition used in our CP PVR implementation contributes to the increase of the efficiency of ERT because it increases the opacity of each ray-segments.



(a) Viewpoint A (b) Viewpoint B

Figure 7. Sample Data (Human Aorta)

### 4.2 Weak Sharing of ERT Information among PCs

The previous section introduced the ERT technique into cell-projection(CP) volume rendering by configuring the ERT-table with each working nodes (WNs). Now, we are going to extend this technique to our parallel program(CP-PVR). The fundamental idea is that the accuracy, or completeness, of the information in the ERT-table may increase if WNs exchange their own information. The typical situation is as follows. Let's assume two working nodes:WNa and WNb where WNb holds cells which are located behind the cells in WNa. In this situation, if WNa terminates its scan conversion due to ERT, WNb has no need to continue its scan conversion. If WNa and WNb exchange the ERT information in their ERT-tables, WNb can also benefit from ERT. This is what we call ERT sharing. The frequent exchange of ERT information may increase the accuracy of ERT information; however, it may incur undesired inter-processor communication. Thus, we have introduced the weak sharing of ERT information into our CP PVR program. The meaning of weak sharing is that the frequency of ERT information exchange is far beneath the frequency of ERT information updates at each WN.

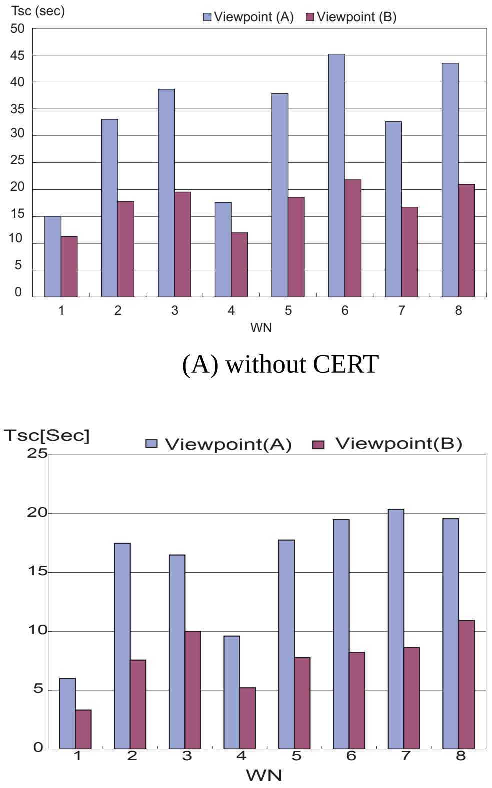
## 5 Evaluation

### 5.1 Experimental Environment

For a relatively large sample of volume data, we used segmented human aorta simulation data(Figure 7). This dataset consists of 307,565 unstructured cells of tetrahydra, with 62,475 vertices in total. Each vertex are attributed with floating-point numerical data of pressure $P[N/m^2]$ and velocity vector$(v_x, v_y, v_z)[m/s]$. In the following experiments, magnitude of velocity vector at each vertex is used for visualization. The average opacity of each vertex was set to 0.3 during the experiments.

The initial assignment of this dataset to each working node (WN) is determined such that 1) the number of cells assigned to each WN becomes roughly the same and 2) the cells inside a WN are geometrically close each others as long as possible.

Unless noticed especially, screen resolutions of $900 \times 300$ and $300 \times 300$ are used for viewpoints (A) and (B), respectively, in the following discussions. An 8-node PC cluster(Pentium4 3GHz, 1GbE) for the WNs and a 1-node



(A) without CERT

(B) with CERT

Figure 8. Scan Conversion Time on each WN w/o DLB

PC(Pentium4 2GHz, 1GbE) for the control node (CN) are used in our experiment.

As a preliminary experiment, we have measured the scan conversion time on each WN. Figure 8(a) shows the result of this experiment without the Conservative Early Ray Termination(CERT) and Dynamic Load Balancing(DLB). In this figure, we can observe the load imbalance due to the view dependency, in the viewpoint (A) in particular.

## 5.2 Effects of the Conservative Early Ray Termination

In order to investigate the effects of the conservative early ray termination (CERT) scheme, we first examined the scan conversion time in the sequential execution on a single WN. In the following experiments, we consider a ray can be terminated if the partially accumulated opacity of one of its ray segments becomes heigher than 0.9 when we apply the CERT scheme. Table 1 summarizes this results for both with and without the CERT scheme. In this experiment, a screen resolution of $300 \times 100$ is used for the viewpoint (A') because we couldn't complete the execution with the $900 \times 300$ screen for viewpoint (A) due to a lack of memory capacity.

Table 1. The Effect of CERT in Sequential Execution

| Viewpoint | Scan Conversion Time[sec] | |
|---|---|---|
| | without CERT | with CERT |
| (A') | 49.3 | 39.1 |
| (B) | 169.9 | 44.0 |

We can confirm a significant speedup of $3.86$ times for the viewpoint (B) and a moderate speedup of $1.26$ times for the viewpoint (A). This difference comes from the difference in the degree of overlapping of cells. Since many cells are overlapping each other in case of viewpoint(B), the possibility of early ray termination increases and thus, the CERT scheme performs well.

Figure 8(b) shows the scan conversion time in the 8-node parallel execution with the CERT scheme. By comparing the results in Figure 8(a) and (b), we can also confirm the effects of the CERT on each WN. The speedup ratio in the viewpoint (B) is a little bit diminished because the degree of overlapping of cells in each WN is decreased compared with that degree in the sequential execution.

Here, we have to note that the patterns of load imbalance both for viewpoints (A) and (B) differ between Figure 8(a) and (b). It is due to the run-time feature of ERT which cannot be estimated in advance to the execution.

## 5.3 Effects of the Sharing of ERT Information

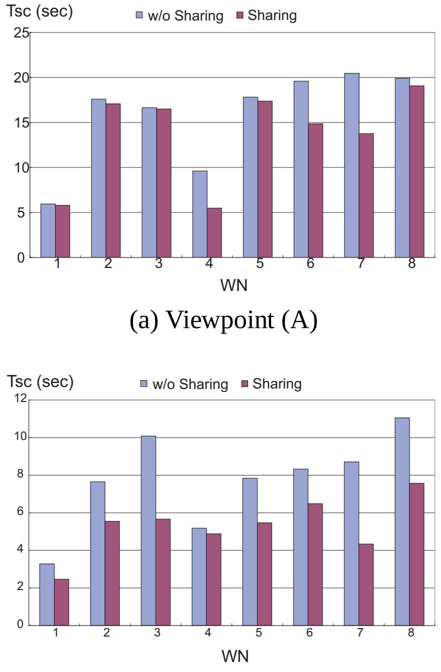
Figure 9(a) and (b) show how the scan conversion time on each WN changes if the sharing of ERT Information is applied for viewpoints (A) and (B). We can confirm the decreases of at most 43% and 50% in scan conversion time for viewpoints (a) and (b), respectively, when the ERT Information sharing among WNs is applied. However, the effective performance improvement as the parallel execution is 7% for the viewpoint (A) because the effect of the sharing is observed only in a few WNs. On the other hand, the effective performance improvement of 30% is achieved in viewpoint (B) where the effect of the sharing is appeared in every WN except WN4.

## 5.4 Overall Effects

Table 2 summarizes the overall effects of the optimization techniques used in our CP-PVR program. ”Base” stands for the parallel processing without CERT and DLB. From this table, we can confirm the 2.86- and 4.32-times speedup compared to the ”Base” implementation for viewpoints (A) and (B), respectively.

## 6 Conclusion

We have proposed a cell-projection parallel volume rendering system for simultaneous visualization of simulation results on a PC cluster. By adopting the conservative early



(b) Viewpoint (B)

Figure 9. Effects of the sharing of ERT information

Table 2. Overall Effects

| Scan Conversion Time T_sc[sec] | | |
|---|---|---|
| Optimization Level | (A) | (B) |
| Base | 45.37 | 25.48 |
| CERT | 20.38 | 10.93 |
| CERT + SHARE | 19.06 | 7.57 |
| CERT + DLB | 18.40 | 8.87 |
| CERT + DLB + SHARE | 15.82 | 5.89 |

CERT:Conservative Early Ray Termination
SHARE:ERT Information Sharing
DLB:Dynamic Load Balancing

ray termination and dynamic load balancing optimization techniques, it could achieve a 4.32-times performance improvement in our experimental environment. We would like to further investigate the various features of our program in the near future.

## 7 acknowledgment

We would like to thank Prof. T. Matsuzawa and Dr. M. Watanabe of JAIST for providing the aorta dataset. Part of this research was supported by the Grant-in-Aid for Scientific Research (S)(2)#16100001 from JSPS.

## References

[1] Yuki Maruyama, Satoshi Nakata, Motohiro Takayama, Tomoaki Tsumura, Masahiro Goshima, Shin ichiro Mori, Yasuhiko Nakashima, and Shinji Tomita. Parallel volume rendering with commodity graphics hardware(in japanese). In *IPSJ SIG Meeting(2003-ARC-154)*, pages 61–66.

[2] Nelson Max, Pat Hanrahan, and Roger Crawfis. Area and volume coherence for efficient visualization of 3D scalar functions. In *Computer Graphics (San Diego Workshop on Volume Visualization)*, volume 24 of *5*, pages 27–33, 1990.

[3] Kwan-Liu Ma and Thomas W. Crockett. Parallel visualization of large-scale aerodynamics calculations: A case study on the cray T3E. In *IEEE Parallel Rendering Symposium*, pages 95–104, 1997.

[4] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259. ACM Press, 1984.

[5] Ricardo Farias and Claudio T. Silva. Parallelizing the zsweep algorithm for distributed-shared memory architectures. In *Proc. of the Volume Graphics 2001*.

[6] Kwan-Liu Ma, James S. Painter, Charles D. Hansen, and Michael F. Krogh. Parallel volume rendering using binary-swap compositing. *IEEE Computer Graphics and Applications*, 14(4):59–68, 1994.

[7] M. Matsui, A. Takeuchi, F. Ino, and K. Hagiwara. Reducing the complexity of parallel volume rendering by propagating accumulated opacity. In *Technical Report of IEICE(CPSY2002-31)*, volume 103, pages 13–18, 2003.

[8] Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9–3:245–261, 1990.

[9] Motohiro Takayama. Dynamic load balancing on parallel visualization of large scale unstructured grid (in japanese). Master's thesis, Graduate School of Informatics, Kyoto University, 2004.

[10] L. Chen, I. Fujishiro, and K. Nakajima. Parallel performance optimization of large-scale unstructured data visualization for the earth simulator. In *Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization*, pages 133–140. Eurographics Association, 2002.

[11] Motohiro Takayama, Yuki Shinomoto, Masahiro Goshima, Shin ichiro Mori, Yasuhiko Nakashima, and Shinji Tomita. Implementation of cell-projection parallel volume rendering with dynamic load balancing. In *The 2004 Int'l Conf. on Parallel and Distributed Processing Techniques and Application*, 2004.