효율적인 BCC 볼륨데이터의 GPU 등가면 레이캐스팅 *

김민호[0], 김현준, Aaliya Sarfaraz
서울시립대학교 컴퓨터과학부
(minhokim, kamu1324)@uos.ac.kr, mohdilya76@yahoo.com

Efficient GPU Isosurface Raycasting of BCC Volume Datasets

Minho Kim[0], Hyunjun Kim, Aaliya Sarfaraz
Dept. of Computer Science, University of Seoul

Abstract

This paper presents a real-time GPU isosurface raycaser that improves the performance by 4-7 folds of our previous method, while keeping the superior visual quality. Such an improvement can be achieved by incorporating an efficient empty-space skipping scheme and an analytic gradient computation. The empty-space skipping scheme is done by building a min/max octree computed from the BB(Bernstein-Bézier)-forms of spline pieces and the analytic gradient formula based on directional derivative box-spline kernels provides not only an accurate visual quality but also an improved performance.

1. Introduction

Real-time volume rendering is one of the most actively investigated research area where the modern graphics hardware plays a very important role due to their powerful parallel processing feature. On the other hand, it has been known for a while that the most widely used and hardware-friendly Cartesian lattice is not an efficient sampling lattice compared to other regular lattices such as BCC (Body-Centered Cubic) and FCC (Face-Centered Cubic) lattices. Specifically, it is shown that the BCC lattice is the optimal 3-dimensional sampling lattice and several reconstruction filters for BCC datasets have been proposed, one of which is the 7-direction quartic box-spline filter [1].

* 구두발표논문
* 본 논문은 요약논문(Extended Abstract)으로서, 본 논문의 원본 논문은 한국그래픽스학회논문지 19(2), 2013에 게재확정 되었음.
* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (No. 2010-0024007)

This paper extends our previous work [2] to improve the performance of GPU isosurface raycasting kernel by applying the analytic gradient computation anefficient empty space skipping method [3].

2. Efficient GPU isosurface raycasting of BCC volume datasets

2.1. Previous work

Kim and Lee [2] implemented a real-time GPU isosurface raycasting kernel by leveraging (1) transformations by encoding instead of lookup tables (2) faster data fetch by converting matrix-vector multiplication to a vector addition, and (3) an efficient evaluation scheme based on partially factored formula.

2.2. Analytic gradient computation

For isosurface rendering, we need to compute the gradient of the spline for correct shading. Usually the it is approximated by the finite difference method. But this requires six more evaluation of the spline and the delta value needs to be determined carefully to avoid both bad approximation and round-off error.
Let M_Ξ be the box-spline defined by the direction matrix Ξ. One of the properties of box-splines is that the directional derivative along the direction $\xi \in \Xi$ is the same as the backward difference of the box-spline defined by $\Xi \setminus \{ \xi \}$. Based on this property, we can compute the gradient of a spline analytically from three “directional derivative kernels” [4].
Analytic gradient computation not only results in a superior rendering result, but also improves the performance by 55-80%, since we do not need additional data fetch operations but can re-use those data already fetched for spline evaluation. Table 1

shows the the performance improvement.

Table 1. Performance comparison (FPS) between two gradient computation methods.

Dataset	Finite difference	Analytic	Speed-up
#1	22.3	39.4	1.77
#2	27.5	47.6	1.73
#3	21.7	35.0	1.61
#4	15.7	24.4	1.55

2.3. Empty space skipping

Compared to the direct volume rendering method, isosurface rendering does not require evaluating the volume field along the whole ray, but only finding the nearest intersection. Therefore we can improve the performance by discarding those voxels that are guaranteed not to include the isosurface, and this can be done in our scheme by leveraging the BB-form of the polynomial pieces. In other words, in the preprocessing step, we first build a hierarchical min/max octree based on the min/max values of all BB-coefficients. At each rendering step, those boxes that do not contain the isolevel are discarded in the vertex shader hence the lengths of rays are shortended significantly. Figure 1 visualizes the performance improvement due to our empty space skipping technique.

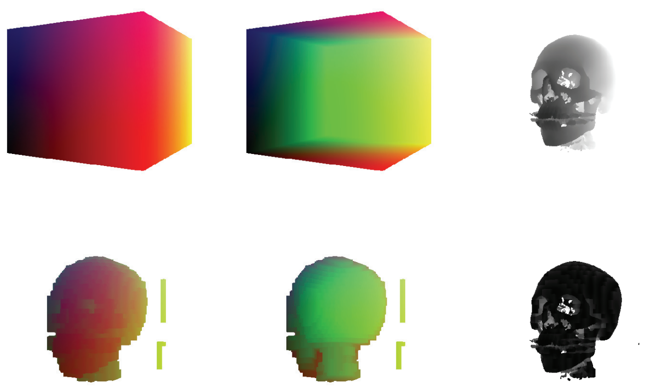

Figure 1. Visualization (top) without and (bottom) with the empty-space skipping method. (Left) entry and (middle) exit points for rays are normalized and color-coded. (Right) the number of evaluations for each ray before intersection color-coded in grayscale.

3. Results and discussion

Table 2 shows the performance comparison between difference GPU raycasting methods of BCC datasets. The overall rendering performance is improved by 4-5 folds compared to our previous work. When compared to other isosurface raycasting methods, our new method still outperforms them. But note that, while it requires some work, our optimization techniques can also be applied to other methods, too.

Table 2. Performance comparison for (a) previous method, (b) our new method, (c) bcc8 [5], and (d) bcc12 [6].

Dataset	(a)	(b)	(c)	(d)	(b)/(a)	(b)/(c)	(b)/(d)
#1	8.3	39.4	11.3	34.1	4.75	3.49	1.16
#2	7.6	47.6	10.3	34.0	6.26	4.62	1.40
#3	6.5	35.0	8.1	22.8	5.38	4.32	1.53
#4	3.2	24.4	4.6	13.0	7.63	5.30	1.88

4. Conclusion and future work

By incorporating two optimization techniques, we can improve the overall performance significantly while keeping the superior visual quality of our previous method. This enables our GPU isosurface raycasting practical for modern graphics hardware. While our optimization techniques are limited to isosurface rendering, we plan to investigate an efficient direct volume raycasting kernel of BCC volume datasets.

References

[1] M. Kim, Quartic Box-Spline Reconstruction on the BCC Lattice, *IEEE TVCG*, 19(2): 319-330, 2013.

[2] M. Kim and Y-J. Lee, Real-time BCC Volume Isosurface Ray Casting on the GPU, *Journal of KCGS*, 18(4): 25-34, 2012.

[3] M. Kim, GPU Isosurface Raycasting of FCC Datasets, *Graphical Models*, 75(2): 90-101, 2013.

[4] M. Kim, H. Kim, and A. Sarfaraz, Efficient GPU Isosurface Ray-casting of BCC Volume Datasets, *Journal of KCGS*, 19(2), 2013 (to appear).

[5] B. Finkbeiner, A. Entezari, D.V. De Ville, and T. Möller, Efficient volume rendering on the body centered cubic lattices using box splines, *Computers & Graphics*, 34(4): 409-423, 2010.

[6] B. Csébfalvi and Markus Hadwiger, Preltered B-spline reconstruction for hardware-accelerated rendering of optimally sampled volumetric data, In *Workshop Vision, Modeling, and Visualization*, 325-332, 2006.