

ボリュームレンダリングを高速化するための in-place 回転による キャッシュヒット率の向上

御前 雄嗣[†] 伊野 文彦[‡] 萩原 兼一[‡]

大阪大学基礎工学部[†] 大阪大学大学院情報科学研究科[‡]

1 はじめに

ボリュームレンダリング (VR) は 3 次元データを可視化する技術である. この技術は医用 CT 像や流体シミュレーションの分析を視覚的に支援する.

VR はメモリ集中型の応用であり, 参照したデータの再利用が局所的である. したがって, テクスタチャキャッシュ (TC) のヒット率向上による高速化が試みられている. Sugimoto ら[1]は, 視点の位置に応じてスレッドブロック (TB) [2]の形状を適切に選択することにより, TC ヒット率を向上した. ただし, 特定の視点において, 参照ストライドが大きくなり, TC ヒット率を向上できなかった.

そこで, 本論文では, TC ヒット率向上による VR の高速化を目的として, 視点の位置に応じてボリュームデータを in-place 回転する手法を提案する. ここで, in-place 処理とは, 作業用メモリ領域の大きさが $O(1)$ で済む処理を指す. メモリ参照のストライド幅が大きくなる視点に対しては, レンダリングの直前にボリュームデータを動的に in-place 回転することにより, ストライド幅を削減し, TC ヒット率の向上を図る.

2 提案手法

Sugimoto らの手法では, スクリーンに対して最も平行な面が yz 平面であるときに (図 1 左),

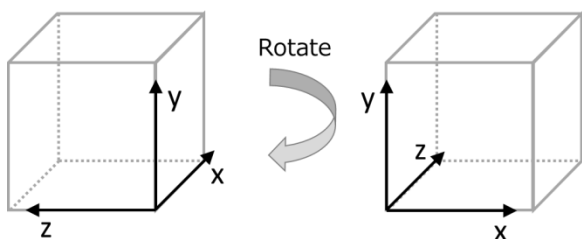


図 1 回転による参照ストライド幅の削減

Increasing Cache Hit Rate by In-Place Rotation for Fast Volume Rendering

[†] Yuji MISAKI [‡] Fumihiko INO [‡] Kenichi HAGIHARA

[†] School of Engineering Science, Osaka University

[‡] Graduate School of Information Science and Technology, Osaka University

TC ヒット率が低下してしまう. この場合, ストライド幅に関して有利な x 軸がスクリーンと平行になるよう, 提案手法はボリュームデータを動的に回転させる. 具体的には (図 1 右), y 軸を中心にボリュームデータを 90 度回転させ, yz 平面への参照を xy 平面への参照に置換する. さらに, Sugimoto らの手法, すなわち TB 形状の適切な選択を併用する.

回転前および回転後のデータをビデオメモリに保持しておけば, これらの使い分けにより提案手法を実現できる. しかし, ボリュームデータの 2 倍のビデオメモリ容量が必要である. そこで, 我々は in-place 処理を目指す.

2.1 in-place 回転

図 2 に, xz 平面を 90 度回転させる様子を示す. 図中において, T1~T4 の各々は $t \times t$ 個のボクセルを含むタイルを表す. 回転後に T1 は T2 の位置へ移動する. 同様に, T2, T3 および T4 について移動先を考えれば, T1~T4 の移動は巡回置換で表せる.

提案手法は, in-place 回転を実現するために, この巡回置換を 1 つの TB (サイズ $t \times t$) に担当させる. つまり, TB は 4 回の移動を反復し, 各々はタイルの退避および上書きを処理する. 例えば, T1 の移動は以下のように実現する.

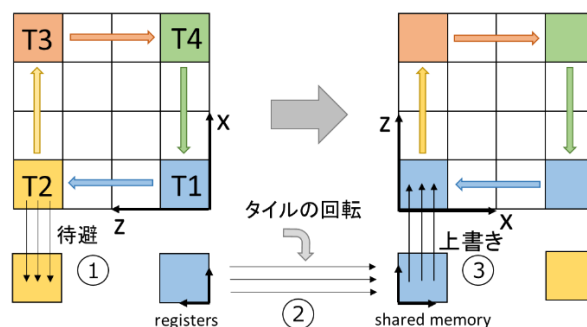


図 2 タイル T1 の in-place 移動と T2 の退避

1. T2 をレジスタに退避する (図 2 の①)
 2. レジスタに退避済の T1 を回転し, 共有メモリに書き込む (図 2 の②)
 3. 共有メモリ上の回転済データを T2 に上書きする (図 2 の③)
2. において書き込み先を共有メモリにする理由は, メモリコアレスシングを実現して実行効率を高めるためである. 共有メモリに起因して 2. の前後には同期が必要となる.

表 1 実験環境

OS	Windows 7 Professional 64-bit
CPU	Intel Core i7-3770K 3.5GHz
主記憶容量	16GB
GPU	NVIDIA GeForce GTX 680
ビデオメモリ容量	2GB
CUDA バージョン	6.5
データサイズ	1024 ³ ボクセル
スクリーンサイズ	1024 ² ピクセル

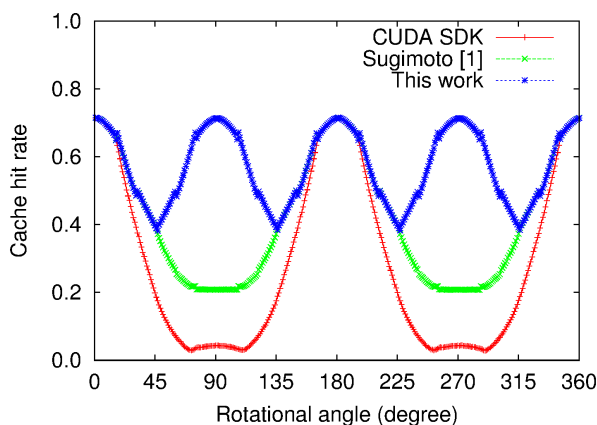


図 3 TC ヒット率

3 評価実験

提案手法を評価するために, TC ヒット率およびフレームレートを計測した. 具体的には, y 軸を中心として視点を 1 度刻みで回転させた. 比較対象として, CUDA に付随しているサンプルコード volumeRender[3]および先行研究[1]を用意した. 表 1 に実験環境を示す.

図 3 および図 4 に, 視点ごとの TC ヒット率およびフレームレートを示す. あらかじめボリュームを回転する提案手法では, 本来 yz 平面を参照していた部分 (図 3 の 45~135 度および 225~315 度) の TC ヒット率が xy 平面参照時 (図 3 の 135~225 度) の TC ヒット率に置き換わり, TC ヒット率が向上している. また, TC ヒット率の向上にともない, フレームレートが最大で 3.7 倍

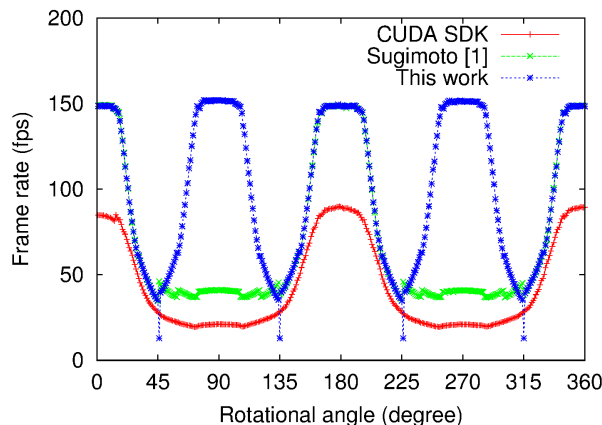


図 4 フレームレート

向上した (図 4 : 90 度および 270 度).

図 4 から, in-place 回転時の 45 度, 135 度, 225 度および 315 度において, フレームレートが 13fps まで低下している. 性能低下の原因は, 回転時のオーバヘッドにある. in-place 回転時のメモリ読み書きスループットは 43GB/s であり, GPU のピークメモリバンド幅 192GB/s に対し, 実行効率が 22%に留まる. 実行効率が低い原因はスレッド間の同期にある. 実際に, 誤りを承知したうえで同期処理を除いた場合, 実行効率は 61%に向上した.

4 まとめ

本論文では, TC ヒット率向上による VR の高速化を目的として, 視点の位置に応じてボリュームデータを in-place 回転する手法を提案した. 結果として, フレームレートを最大 3.7 倍まで向上できた.

今後の課題として, in-place 回転時におけるメモリ読み書きスループットの向上が挙げられる.

謝辞

本研究は科研費 24560458 の補助による.

参考文献

- [1] Yuki Sugimoto, Fumihiko Ino, and Kenichi Hagihara, "Improving Cache Locality for GPU-based Volume Rendering," *Parallel Computing*, Vol. 40, No. 5/6, pp. 59-69, May. 2014.
- [2] NVIDIA Corporation, "CUDA C Programming Guide Version 6.5," http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf, Aug. 2014.
- [3] NVIDIA Corporation, "CUDA Code Samples," <http://developer.nvidia.com/cuda-code-samples/>, Aug. 2014.