

## ボリュームレンダリングにおける効率的なテクスチャ参照の検討

杉本 祐樹 † 伊野 文彦 † 萩原 兼一 †

### 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。ボリュームの単位格子はボクセルと呼ばれ、それぞれ値を持つ。

VR では、スクリーン上のピクセル値を得るために、視点からピクセルに至るまでのボクセル値を積算する。このとき、ボクセル値は近接領域でのみ再利用できる。ゆえに、計算と比べてメモリの負荷が高く、メモリ集中型の応用である。したがって、メモリ帯域幅の高い GPU を用いて高速化を図ることが多い。具体的には、ボリュームをテクスチャメモリ (TM) に格納し、ハードウェアによる線形補間を利用する。

そこで、本研究では VR の性能向上を目的として、効率のよい TM 参照を実現する。ボリュームの軸に応じて参照ストライドが変動することに着目し、視点の位置に応じて適切なスレッドブロック (TB) の形状を選択する。これにより、一連のスレッドが同時に参照する TM 領域の局所性を高め、TC ヒット率の向上を狙う。

### 2. テクスチャメモリの参照機構

テクスチャメモリの物理アドレスと論理アドレスは図 1 のような対応関係をもち、 $u$  および  $v$  の 2 つの軸におけるメモリ参照に最適化されている。ボリュームレンダリングでは 3 次元のテクスチャにデータを格納する。したがって、ある軸方向のメモリ参照においては他の 2 つ軸方向と比較して参照ストライドが大きくなり、参照局所性が低くなる。参照局所性が低い軸の方向に対して同時にメモリを参照するとき、キャッシュミスが増加する。したがって、キャッシュミスを削減するためには参照局所性が高い軸の方向に対して同時にメモリを参照すればよい。

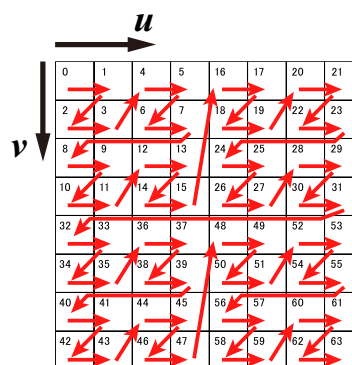


図 1 2 つの方向に最適化されたテクスチャメモリ

### 3. スレッドブロックのサイズと形状

TB の形状を選択する前段階として TB のサイズを決定する。1 つのマルチプロセッサは高さ 8 つの TB を割り当てて並列実行する。TB の数は、TB のサイズや使用するレジスタの数により決まる。また、ワーブ単位で実行するため、スレッド数は 32 の倍数がよい。したがって、8 つの TB を割り当てることができ、かつ 32 の倍数で最大のサイズを選ぶ。

各スレッドは 1 つのピクセル値を計算する。したがって、32 個のスレッドが 32 個のピクセル値を計算する。ワーブ形状は、TM に対する参照ストライドに基づいて決定する。ここで、参照ストライドの小さい軸の方向にワーブの形状を長く設定すれば、同時に参照するメモリ領域を局所化できる。ただし、プログラムはワーブの形状を指定できないため、TB の形状を指定することでワーブの形状を決定する。スレッド ID の若い方から順に 32 ごとに区切られた領域の形状がワーブの形状にあたる。

図 2 に 6 方向の視点位置からボリュームを見たときの参照ストライドの大小関係を示す。図中の矢印は太い直線、点線、細い直線の順に参照ストライドが大きくなることを表す。この参照ストライドの小さい方向に対してワーブの形状を長く設定する。これにより、参照局所性の高い軸方向のデータがキャッシュに多く

† 大阪大学大学院情報科学研究科

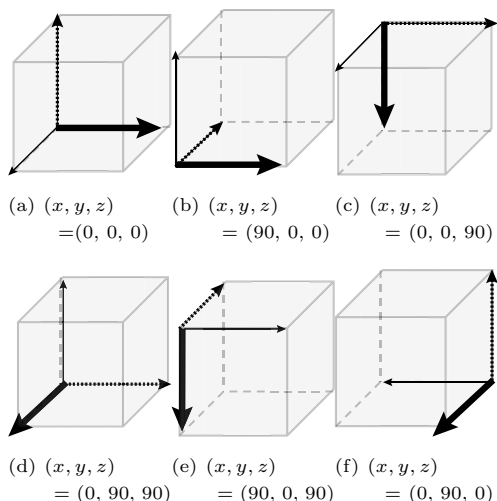


図 2 3つの次元における参照ストライドの大小関係

格納されるため TC ヒット率を向上できる。したがって、図 2(a), 2(b) および 2(d) では横方向に、図 2(c), 2(e) および 2(f) では縦方向にワーブ形状が長くなるように TB の形状を設定する。また、積算時に同じタイミングで近傍領域を参照することで計算の局所性を高められるため、適宜スレッド間で同期をとる<sup>1)</sup>。これにより、TC ヒット率の向上が期待できる。

#### 4. 実験

TC ヒット率および実行時間を調べるために、TB のサイズを 128 で固定し、図 2 の 6 つの視点位置について TC ヒット率と実行時間を計測した。表 1 に実験環境を示す。また、表 2 に CUDA SDK のサンプルプログラムによる VR の実行時間および TC ヒット率を、表 3 に実行時間が最短時の実行時間および TC ヒット率の結果とそのときの TB およびの形状のまとめを示す。サンプルプログラムの TB サイズは  $16 \times 16 = 256$  で同時実行 TB の個数は 4 個である。

表 2 および 3 から参照のストライドの小さい軸の方向がスクリーンに垂直なときは TC ヒット率が低く、実行時間が長い。このとき、実行時間を 20% 短縮でき

表 1 実験環境

OS	Windows 7 Professional 64-bit
CPU	Intel Core i7 930 2.80 GHz
主記憶	12 GB
GPU	NVIDIA GeForce GTX 480
ビデオメモリ	1536 MB
CUDA バージョン	CUDA 3.2
ビデオドライバ	260.61

表 2 CUDA サンプルプログラムの実行時間および TC ヒット率

視点位置 (x, y, z)	実行時間 (ms)	TC ヒット率 (%)
(0, 0, 0)	10.4	90.0
(90, 0, 0)	10.4	85.8
(0, 0, 90)	10.4	85.7
(0, 90, 90)	12.2	65.5
(90, 0, 90)	11.7	84.0
(0, 90, 0)	15.3	63.4

表 3 各視点位置における実行時間および TC ヒット率の結果と TB およびワーブの形状

視点位置 (x, y, z)	実行時間 (ms)	TC ヒット率 (%)	TB	ワーブ
(0, 0, 0)	10.3	90.8	8 × 16	8 × 4
(90, 0, 0)	10.3	83.0	8 × 16	8 × 4
(0, 0, 90)	10.3	95.4	8 × 16	8 × 4
(0, 90, 90)	11.1	92.7	16 × 8	16 × 2
(90, 0, 90)	10.7	93.9	2 × 64	2 × 16
(0, 90, 0)	12.1	71.9	4 × 32	4 × 8

た。また、TC ヒット率も図 2(b) の視点以外についてはサンプルプログラムよりも高く、概ね TC ヒット率の向上により実行時間を短縮できた。TB の形状については、図 2(c) の視点以外については 3 章で予想した通りの結果である。図 2(e) については縦横比が異なり、参照ストライドの最大方向と最小方向の差が大きいために最大方向への TB の形状が他の視点位置よりも長いときに実行時間が短くなったものと考えられる。

#### 5. まとめ

本稿では、VR における 3 次元テクスチャメモリの効率的なデータ参照を目的として、視点の位置により TB の形状を適切に設定することを提案した。結果として、ワーブ単位でのメモリ参照の局所性を高め、TC ヒット率を向上した。また、参照ストライドの小さい軸の方向がスクリーンに垂直な場合に提案手法の効果が大きく、実行時間を最大 20% の短縮できた。今後の課題として、具体的な TB の形状を決定することが挙げられる。

謝辞 本研究の一部は科学研究費補助金基盤研究 (B) (2330007) の補助による。

#### 参考文献

- 成瀬彰, 住元真司, 久門耕一. GPGPU 上での流体アプリケーションの高速化手法 ~ 1GPU で姫野ベンチマーク 60GFLOPS 超 ~. 情処研報, 2008-HPC-117, pp. 49-54, October 2008.