

適応的四面体ボリュームメッシュに基づくオンラインリメッシュを伴う 変形計算の GPU による高速化の検討

佐々木 康行†

田川 和義††

田中 弘美†††

†立命館大学 情報理工学部

††立命館グローバル・イノベーション研究機構

†††立命館大学 理工学研究科

1 はじめに

医療分野では CT や MRI などの医療用断層画像撮影装置の高性能化に伴い通常では不可視である人体の内部構造などを可視化することが可能になった。そこで可視化された三次元のボリュームデータを用いた手術シミュレーションシステムの開発が注目されている。しかし、CT や MRI などから得られる医療用のボリュームデータのデータサイズは膨大であり、コンピュータの高速化が進んだ昨今であっても実時間でのシミュレーションを行うことは難しい。実世界に忠実な「体験」を提供するためには、実世界に存在する様々な物体が有する力学的特性を表現すると共に、実世界の挙動や振る舞いに忠実なシミュレーションを実現することが課題である。そのため、筆者らの研究室では、適応的四面体ボリュームメッシュとオンラインリメッシュ法 [1] を用いた仮想柔軟物体の高速な変形計算手法を提案しているが、大規模なボリュームデータを力覚の実時間 (1kHz 以上が必要といわれている) で取り扱うことは難しかった。そこでこれをさらに高速化することを目的とし、Rasmusson らが提案した、Explicit Addressing [2] を基にオンラインリメッシュを伴う仮想柔軟物体のための GPGPU (CUDA) に適した並列計算手法の検討を行う。

2 適応的四面体モデル

本研究では、変形の大きな箇所では四面体を細かく、変形の小さな箇所では四面体を荒くする適応的メッシュを用いることにより、計算時間を減らしつつ、シミュレーションの精度を保っている。

図 1 は、3 次元空間中に配置したノード数 27 の立方体の左上手前のノードに強制変位を与えることにより生じた状態の変移を示す。強制変位が与えられたノ

ド付近のメッシュが細かくなり、変位がなくなるとメッシュが粗くなるのが分かる。

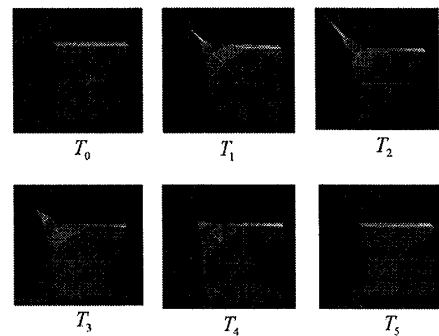


図 1: 適応的四面体モデル

3 提案手法

GPU はシンプルな演算ユニットを多数搭載しており、これらが並列に動作する構成になっているため、CPU に比べ並列性の高い演算処理を行う場合、高い処理性能が得られる。一方でバス幅の制約や DRAM の特性からメモリアクセスに非常に厳しい制約がある。そのため、効率的なデータアクセスのためには、並列に動作する演算ユニットがメモリ上の連続するアドレスをアクセスする必要がある。

オンラインリメッシュを伴うボリュームデータでは、四面体の分割と統合を行うため、実際にモデルを構成しているアクティブなノードと、構成していない非アクティブなノードが混在している。そのため、計算に必要なデータが不規則に存在することになり、GPU に取って効率の悪いデータ構造になってしまう。

そこで本手法では、ボリュームデータをノードリストと接続表に分け、それぞれの特性に合った管理を行うことにより、オンラインリメッシュを伴う変形物体の CUDA に適した並列化手法を提案する。

図 3 は、2 次元物体における (a) 対象物体と (b) 接続表と (c) ノードリストの関係を示した例である。(a) の円がノードであり、その中に書いてある数字がそのノ

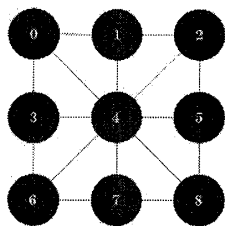
Examination of the speedup by GPU of the transformation calculation with the online re-mesh based on adaptive tetrahedron volume mesh

†Yasuyuki SASAKI ††Kazuyoshi TAGAWA †††Hiromi TANAKA
†College of Information Science and Engineering, Ritsumeikan University

††Ritsumeikan Global Innovation Research Organization

†††Graduate School of Science and Engineering, Ritsumeikan University

ドの ID である。



(a) 対象物体

ノードID	0	1	2	3	4	...
接続ノード数	2	3	3	3	3	...
接続ノードID	1	0	1	0	0	...
接続ノードIDとの初期長	1.00	1.00	1.00	1.00	1.41	...
接続ノードID	2	4	4	4	1	...
接続ノードIDとの初期長	1.00	1.00	1.41	1.41	1.00	...

(b) 接続表

アクティブフラグ	0	1	2	3	4	...
X座標値	0.00	1.00	2.00	0.00	1.00	...
Y座標値	2.00	2.00	2.00	1.00	1.00	...
Z座標値	0.00	0.00	0.00	0.00	0.00	...

(c) ノードリスト

図 2: 2次元物体における例

3.1 接続表

接続表とは、アクティブなノードに対して、バネ・ダンパにより繋がっているノード数と繋がっているノード ID とそのノード間の初期長を持つ表である。一つのノードの最大隣接ノード数は 26 個である。

接続表は連続的な読み出し期待できることから GlobalMemory に格納する。接続表の各列の計算を GPU のそれぞれの演算ユニットに担当させることで結合アクセスによる高速な読み出しを実現する。

3.2 ノードリスト

ノードリストとは、各ノードのアクティブフラグと X 座標、Y 座標、Z 座標の値をそれぞれ持つリストである。

ノードリストへのアクセスは、接続表から行っており、ランダムアクセスになってしまう。このため、ノードリストを Cache が有効である TextureMemory に格納する。ノード番号と配列の番号を対応させることで、テーブル参照により高速にアクセスすることができる。

3.3 オンラインリメッシュへの対応

オンラインリメッシュを行うとノードが増減するため、接続表にアクティブなノードと非アクティブなノードの情報が混在することになる。この様な状態になると、計算を行わないプロセッサが出てくるが、結合ア

クセスを行うために、それらにもデータを送る必要があるため、メモリの転送時間が余計に掛かってしまう。

そのためオンラインリメッシュになり生じた非アクティブなノードを、リストから削除することとする。

4 提案手法の評価実験

提案手法の評価実験として、CPU で計算を行った時、GPU で接続表の更新がある時と接続表の更新がない時の比較を行った。

実験環境として OS:CentOS 5.3、CPU:Intel(R) Xeon X5355 2.66GHz、Memory:8GB、GPU:NVIDIA GeForce 8800GT を用いた。ノード数 729 の立方体モデルの変形計算に掛かった時間を計測した。

CPU で計算を行ったときは、17.7[ms]、GPU で接続表の更新がある時は、1.4[ms]、接続表の更新がない時は 1.0[ms] 要した。いずれの場合においても GPU を用いて変形計算を行うことで、高速に変形計算を行うことができることが実証された。

5 まとめ

本研究では統合開発環境である CUDA を用いて、オンラインリメッシュを伴う柔軟物体の変形を表現するシミュレータを並列化する手法の検討を行い、CUDA の特性に合わせてデータを書き換えることで、高速化を行うことができた。今後は、接続表の更新方法を工夫することでさらなる高速化を図る。

参考文献

- [1] 高間康文, 辻野圭則, 堀洋介, 田中弘美: "柔軟物体の適応的四面体ボリュームメッシュのオンラインリメッシュ法", 日本バーチャルリアリティ学会論文誌, Vol.13, No.1, pp.66-78, 2008.
- [2] Allan Rasmusson, Jesper Mosegaard, and Thomas Sangild Sorensen: "Exploring Parallel Algorithms for Volumetric Mass-Spring-Damper Models in CUDA", Proceedings of the 4th international symposium on Biomedical Simulation, pp.49-58, 2008
- [3] NVIDIA Corporation: "NVIDIA CUDA ProgrammingGuide".
- [4] PHANToMTM, SensAble Technologies, INC. : <http://www.sensable.com>