

GPU を用いたボリューム細分割の高速化に関する研究

内之宮 仁志 西尾 孝治 小堀 研一

大阪工業大学

1. はじめに

一般にCG分野で用いられる細分割は膨大な処理コストを必要とし、高速化が求められている。細分割には、形状の表面部分のみを分割する細分割曲面と形状の内部も分割するボリューム細分割の2つがある。このうち、細分割曲面の高速化手法は、これまでに多く提案されている。しかし、ボリューム細分割の高速化については、あまり研究がなされていない。

一方、近年グラフィックスカード(以下GPU)の描画処理の一部をプログラムにより任意に変更できる機能が登場した。これにより、GPUは様々な分野で利用されるようになってきている^[1]。

そこで、本研究では GPU を用いてボリューム細分割を高速に行う手法を提案する。提案手法では、位相情報の変更と変更された位相情報をもとに各頂点の座標の計算を行う処理を GPU で行うことにより高速化をはかる。

2. ボリューム細分割

ボリューム細分割とは、代表的な細分割手法の一つで四辺形メッシュを対象としたCatmull-Clark細分割手法を3次元空間に拡張した手法である。ボリューム細分割には、細分割ルールが簡単であるBajajらの六面体メッシュ細分割(以下、MLCA 細分割)^[2]が多く用いられている。MLCA細分割には、形状を構成する各六面体の位相を分割するBi-linear Subdivisionと分割後の各頂点の移動を行うAveragingの2つの処理がある。

2. 1 Bi-linear Subdivision

Bi-linear Subdivisionとは形状を構成する六面体の重心と面の重心、および稜線の中点に頂点を生成し、これらを結び新たな稜線を生成する。生成された稜線をもとに、新たな六面体を生成する処理のことである。

2. 2 Averaging

Averagingとは、分割前の初期頂点および Bi-linear Subdivisionにより新たに生成された各頂点を、滑らかな六面体メッシュが形成されるような位置に移動する処理のことである。

Averagingの処理では、形状内部に存在する各頂点は、図2.1(a)に示すように六面体の重心を求める。求めた重心をもとに、同図(b)に示すように頂点を共有している六面体の重心の平均を求め、求めた値を移動後の頂点座標とし移動させる。形状表面の各頂点の座標は、同図(c)に示すようにCatmull-Clark細分割手法のsmoothing処理を適用したときに得られる座標値を移動後の頂点座標とし、移動させる。

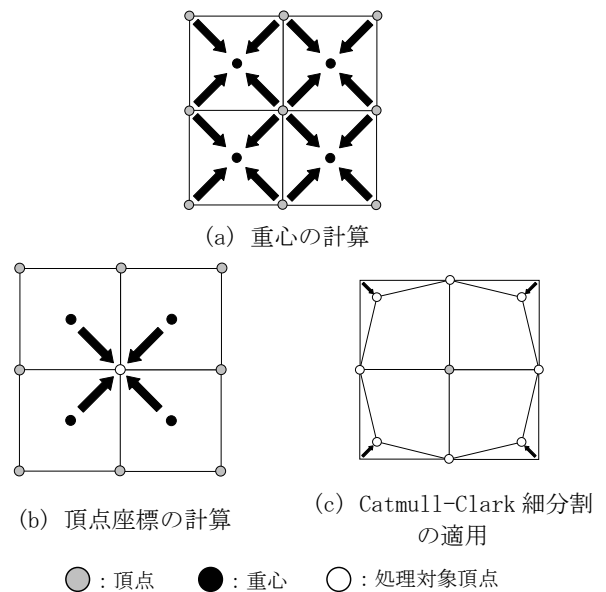


図 2.1 Averaging

3. 提案手法

提案手法では、MLCA細分割におけるBi-linear SubdivisionとAveragingの処理にGPUを用いることで高速化をはかる。提案手法の処理手順について図3.1を用いて概説する。

- ① GPU を用いて Bi-linear Subdivision と Averaging の処理を行うために、形状を構成する六面体と面、稜線および頂点に番号付けを行う。番号をもとに、位相情報をテ

A Study on Fast Method for Volumetric Subdivision by Using GPU

Hitoshi Uchinomiya, Koji Nishio, Ken-ichi Kobori
Osaka Institute of Technology

- スタチャに保存する。
- ② テクスチャに保存された形状データをもとに、一定の規則に従い Bi-linear Subdivision の処理を行う。Bi-linear Subdivision の処理を行った結果をテクスチャに保存する。
 - ③ ②の結果をもとに、GPU で Averaging を行うための頂点データを CPU で作成する。作成した頂点データを入力データとし GPU に入力する。入力された頂点データをもとに Averaging の処理を行い、その結果を出力する。

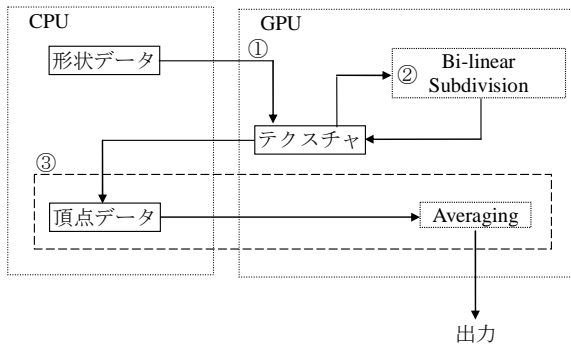


図 3.1 提案手法の処理手順

提案手法では、以上の処理を一度行うことで、一回の MLCA 細分割の結果を得ることができる。

3. 1 Bi-linear Subdivision 処理

提案手法では、形状を構成する六面体と面、稜線、および頂点をそれぞれテクスチャとして GPU に保存する。保存された形状データをもとに、GPU を用いて一定の規則に従い、六面体と面、稜線の位相を変更し、Bi-linear Subdivision の処理を行う。

3. 2 Averaging 処理

Averaging の処理は形状を構成する頂点が内部か外部かで変わる。形状内部にある頂点は、Bi-linear Subdivision の処理を行った結果をもとに、GPU で各六面体の重心を求め、その結果をテクスチャに保存する。保存された各六面体の重心をもとに、各頂点が共有する六面体の重心の平均を求め、その値を細分割後の頂点座標として各頂点を移動させる。

形状外部にある各頂点は、各頂点が接続する頂点の番号を、各頂点の頂点データであるカラー値とテクスチャ座標に割り当てる。これを入力データとして GPU に入力し、GPU で Averaging の処理を行うことで、形状表面にある各頂点の細分割後の頂点座標を得る。

4. 実験と考察

提案手法の有効性を検証するため、実験を行った。実験では、初期形状を構成する六面体の数を変化させ、CPU で行った場合と提案手法で行った場合とで処理時間の比較を行った。実験環境は、CPU : Pentium4 3.8GHz, メモリ : 1 GB, GPU : GeForce 6800GT で実験を行った。実験結果を図 4.1 に示す。

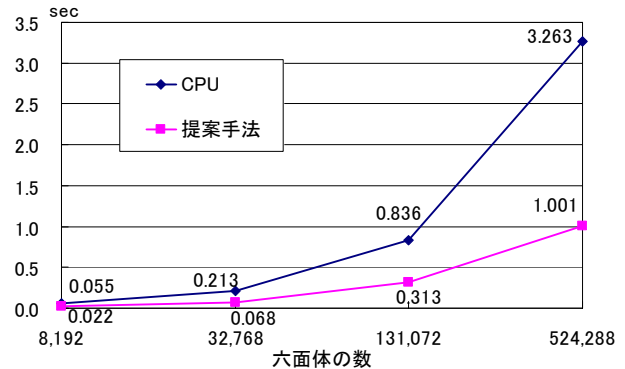


図 4.1 実験結果

実験結果より、提案手法は CPU で行うより、六面体の数が数千個であれば約 1/2 の処理時間で、数万個以上であれば約 1/3 の処理時間で細分割を処理できることが確認できた。また、六面体の数が増えるにしたがって、その差が大きくなることから、提案手法は初期形状の六面体の数が多いほど有効であると考えられる。

5. おわりに

本研究では、GPU を用いてボリューム細分割の一つである MLCA 細分割の高速化を行った。実験結果より、CPU で MLCA 細分割を行う場合と比べ、処理時間が約 1/2~1/3 に削減できていることが確認できた。

今後の課題として、形状の表面にしかメッシュがない場合でもボリューム細分割が行える手法を考案することや、提案手法を CAD やシミュレーションに応用することを考えている。

<参考文献>

- [1] 宮田一乗, 高橋誠史, 黒田篤: "GPU コンピューティングの動向と将来像", 芸術科学会論文誌, Vol.4, No.1, pp.13-19, (2005)
- [2] C.Bajaj, S.Schaefer, J.Warren, G.Xu: "A Subdivision scheme for hexahedra meshes", The Visual Computer, Vol.18, No.5-6, pp343-356, (2002)