

I-001

Gregory 曲面と GPGPU による、 G^1 連続性を保った高速形状変形手法A Fast shape modification with G^1 continuity using Gregory patch and GPGPU.

武田 巧視†
Takeda Takumi

渡辺 大地†
Watanabe Taichi

1 はじめに

近年、ハードウェアの性能が向上し、3DCG においても GPU を画像処理以外の目的で利用する GPGPU を用いて、複雑な計算を高速に並列演算する研究が盛んに行われている。

本論文では、曲面形状モデルを、滑らかなまま高速に形状を変形するために、Gregory 曲面 [1] と GPGPU を用いることを提案する。Gregory 曲面とはパラメトリック曲面の一種である。パラメトリック曲面は、一般に静的な描画においては、現在多く利用されている細分割曲面 [2][3][4] よりも高速に描画できる曲面である。しかし、形状変形を伴う動的な描画においては、曲面同士を滑らかに繋げるための両立性補正がネックになり、高速描画には向いていない。Gregory 曲面は、パラメトリック曲面の中でも両立性補正が簡単な曲面であり、静的動的に関わらず、高速描画が見込める。また、再帰的な細分割曲面に比べ、陽的なパラメトリック曲面は、GPGPU の並列処理にも向いていると考えられる。

2 Gregory 曲面

Gregory 曲面 [1] とは、Bézier 曲面を拡張した曲面表現式である [5][6]。この曲面表現式は、以下のような特徴を持っている。

1. 境界曲線は Bézier 曲線になる。
2. Bézier 曲面と同様に凸閉包性を持つ。
3. 特別な場合に、Bézier 曲面と等しくなる。

さらに、境界曲線を共有する 2 曲面の接続について述べる。曲面同士を滑らかに繋げる処理を両立性補正という。両立性補正によって、境界曲線上のすべての点で接平面が

一致するため、2 曲面を G^1 連続に接続できる。Gregory 曲面は u 方向と v 方向を独立に定義できるので、1 方向だけを意識すればよい。単純な計算なので、Bézier 曲面などより、 G^1 連続に接続する計算コストを非常に少なくできる。

3 検証

本章では、GPGPU を用いて Gregory 曲面の、 G^1 連続性を保った形状変形の高速度を実装したプログラムを使用し、その有用性を検証する。検証に用いた環境は以下の通りである。

CPU : Intel® Core™2 Duo 3.00GHz
メインメモリ : 2.00GB RAM
GPU : NVIDIA GeForce 9600 GT
解像度 : SXGA (1280 × 1024)

3.1 細分割曲面と Gregory 曲面

GPGPU を利用しない状態で、細分割曲面と Gregory 曲面の速度比較を行った。形状変形は行わず、静的な描画速度を比較する。比較方法は、曲面生成処理を 10 回繰り返した際にかかる時間を、分割数を変えて比較した。制御点 (制御ポリゴン) から曲面を生成し、ポリゴン化する際に互いの分割数を揃えた。細分割曲面の制御ポリゴンは 3×3 の四角形ポリゴンメッシュである。10 回計測を行った時の平均値は表 1 のようになる。このように、全体的に Gregory 曲面が速くなるという結果が得られた。

表 1 細分割曲面と Gregory 曲面の速度比較

分割数	12 × 12	24 × 24	48 × 48
細分割曲面	43.9 ms	237.6 ms	1489.6 ms
Gregory 曲面	18.1 ms	106.8 ms	744 ms

3.2 Gregory 曲面の両立性補正

Gregory 曲面の両立性補正処理の高速度を検証した。2 × 3 パッチで布がはためいているように見えるように稜

† 東京工科大学大学院 バイオ・情報メディア研究科
Graduate School of Bionics, Computer and Media Science,
Tokyo University of Technology
‡ 東京工科大学 メディア学部
Faculty of Media Science, Tokyo University of Technology

線を動かし、Gregory 曲面を内挿する。Gregory 曲面は 10×10 分割のポリゴンメッシュで表現した。このとき、ポリゴンメッシュのための頂点配列の生成に GPGPU を利用している。結果、速度にほぼ変化はなく、Gregory 曲面では両立性補正において速度低下は起きないといえる結果になった。10 秒間形状変形を続けたときの平均 FPS を表 2 にまとめた。

表 2 Gregory 曲面の両立性補正の速度比較

両立性補正あり	185.3FPS
両立性補正なし	185.9FPS

3.3 Gregory 曲面と GPGPU

本プログラムでは、Gregory 曲面をポリゴン化する際の頂点配列の算出に GPGPU を利用した。

このプログラムにおいて、曲面の生成、及び変形後の再描画にかかる時間を計測した。GPGPU を利用しないプログラムでは、1 パッチの生成に 30~60 ミリ秒かかったが、GPGPU を利用した場合、ほぼ 1 ミリ秒以下の速度となった。結果、GPGPU を利用することで、曲面の動的な変形の高速度化が実現できた。

しかし、別の環境で計測したところ GPGPU を利用しない方が速度が速いという結果になった。利用した環境の構成は、以下のとおりである。

CPU: Intel® Core™2 Duo 2.40GHz
 メインメモリ: 4.00GB RAM
 GPU: NVIDIA GeForce 8400M GT
 解像度: SXGA (1280 × 1024)

GPGPU の使用で高速化できた環境を A、高速化できなかった環境を B として、曲面を動的に変形し、再描画にかかった時間 10 回分の平均値を表 3 にまとめる。GPGPU を利用することで、曲面の動的な変形の高速度化が実現できたことが分かるが、速度がハードウェアの性能に依存するという結果になった。

表 3 Gregory 曲面の速度比較

A	CPU のみ	44.0784248 ms
A	GPU 利用	0.3748625 ms
B	CPU のみ	53.5556265 ms
B	GPU 利用	378.1104234 ms

4 おわりに

本研究では、Gregory 曲面と GPGPU を用いて、 G^1 連続性を保った高速形状変形手法を提案した。

細分割曲面と Gregory 曲面の形状変形を伴わない静的描画における、Gregory 曲面の高速度性を検証した。また、Gregory 曲面の両立性補正処理が描画速度にほとんど影響を与えないことを検証した。この 2 つの検証から、形状変形を伴う動的な描画においても、Gregory 曲面が細分割曲面よりも高速であるといえる。

GPGPU を用いて、Gregory 曲面のさらなる高速描画も検証した。結果、ハードウェアの性能に依存するが、高速化に成功した。

本手法を用いることで、生き物のような滑らかな形状モデルの動的な変形を実現できるため、ゲーム等リアルタイム 3DCG コンテンツを制作する際の表現力の向上に役に立つだろう。

問題点として、今回は、Gregory 曲面を描画する際にポリゴン化しているが、滑らかな曲面を表現するためには大量のポリゴン数を必要とする。グラフィックメモリは限られているため、曲面の数が増えるほどメモリの解放と確保に関わるデッドロックが発生してしまう。また、GPU の演算性能によっても、描画時間が大きく変化する。そのため、今後の展望として、ポリゴン化することなく描画することで、使用するグラフィックメモリを減らすような対処等が必要になるだろう。

参考文献

- [1] J. A. Gregory, "Smooth interpolation without twist constraints", Computer Aided Geometric Design, R. E. Barnhill and R. F. Riesenfeld, ed., Academic Press, 1974.
- [2] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points", Computer Aided Design, 1978.
- [3] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological meshes", Computer Aided Design, pp.350-355, 1978.
- [4] C. Loop, "Smooth subdivision surfaces based on triangles", Master's thesis, University of Utah, Department of Mathematics, 1987.
- [5] H. Chiyokura and F. Kimura. "Design of solids with free-form surfaces". Computer Graphics, 17(3):289-298, 1983.
- [6] 鳥谷 浩志, 千代倉 弘明, 「3次元 CAD の基礎と応用」, 1991.