1ZC-02

個別の変形を伴う複製描画のための空間計算量の削減

大河原 将　　藤代 一成
慶應義塾大学 理工学部情報工学科

1　序論

膨大な個数の物体によって構成されるシーンを描画する際，時間計算量だけでなく空間計算量の増加も大きな問題となる．そこで本稿では，基本形状を共有し，かつ，個別の変形を伴う複数の物体を描画する際の空間計算量削減手法を提案する．

基本形状をもとに別の物体を生成する際，効率の観点からしばしば複製を使用する．コンピュータグラフィックスの分野において，複製にはコピーとインスタンスが知られている．コピーは空間計算量が大きい代わりに編集の柔軟性が高く，インスタンスは空間計算量が小さい代わりに編集の幅に制限があるという相反する特徴をもっている．つまり，複数の物体が基本形状を共有し，かつ，それぞれの物体が個別の変形を伴う場合，コピーを使用せざるを得ず空間計算量が大きくなってしまう．これは大きなデータの複製を作る際や大量の複製を作る際に致命的な問題となる．

Shell Maps[1] は，個別の変形を伴う複製を描画する際の空間計算量を削減することに成功した．この手法は，サーフェスにジオメトリをマッピングすることで複製を表現する．これにより，小さい空間計算量で個別の変形を伴う複製を表現することは可能になったが，その変形や配置はマッピングされるサーフェスに依存してしまうという課題が残った．

2　枠組み

提案手法では自由形状変形（FFD）[2] の制御点によって複製を表現することで，複製描画にかかる空間計算量の大幅な削減を可能にした．

提案手法の特長は以下の 3 つである．

- 小さい空間計算量
- 個別の変形
- 自在な配置

これらの 3 つを同時に実現することにより，大規模で複雑なシーンを一般的なマシンで描画可能となる．その基本的な枠組みについては先行研究 [3] で既に提案されているが，そこでは大きな時間計算量が問題となっていた．提案手法では，これを GPU で高速化し，大きく改善した．

Space Reduction for Rendering Individually Deformed Clones
Masaru Ohkawara, Issei Fujishiro
Department of Information and Computer Science, Keio University

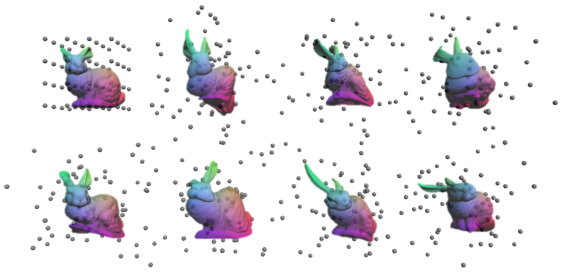

図 1: FFD の制御点によって表現された個別の変形を伴う 8 体の複製．一般に個別の変形を伴う複製を描画する際は，複製の個数だけジオメトリデータが必要となるが，提案手法は FFD の制御点によって複製を表現することで大幅に空間計算量を削減している．

3　提案手法

提案手法の処理は，**初期化**，**複製の探索**，**変形および描画**の 3 つのステップに大きく分けることができる．以下，各ステップを順に説明していく．

3.1　初期化

初期化のステップでは，オリジナルの登録と制御点の配置を行う．オリジナルは，任意の頂点の座標値を x, y, z 各軸について 0 ～ 1 の空間に正規化して登録する．制御点はオリジナルを囲むように軸平行に整列させる．これが制御点の初期位置となる．この状態から制御点を自由に移動させることで変形を伴う複製を表現する．

今回，オリジナルのデータ構造には色付き点群モデルを用いた．また，制御点は時間計算量と変形の柔軟性を考慮して x, y, z 各軸方向に 4 点ずつ計 64 点用いた．

3.2　複製の探索

複製の探索のステップでは，光線が衝突する可能性のある複製を探索する．このとき，FFD は時間計算量の大きい処理として知られているため，事前に大まかな衝突判定を行うことで FFD の計算回数を削減することが重要になる．FFD には凸包性が知られており，制御点によって定義される凸包の内部に変形後の形状が必ず含まれる．また，バウンディングボリュームは凸包を必ず含むことは明らかである．これらの性質を利用して，制御点によって定義されるバウンディングボリュームとの光線衝突判定を行い，光線が衝突する可能性のある複製を効率的に探索する．

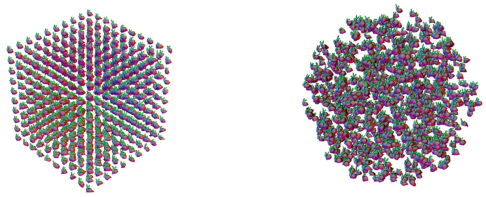

図 2: 自在な配置の例．左は立方体内に，右は球内にそれぞれ 100 体の個別に変形した Stanford Bunny が配置されている．

図 3: 繊維レベルの布への適用例．メモリ使用量は，コピーで表現すると 503 GB，提案手法で表現すると 58 MB となる．

3.3 変形および描画

変形および描画のステップでは，光線が衝突する可能性のある複製について，実際に変形処理を施し，描画する．提案手法は，オリジナルを複数の複製が共有するため，変形前のデータを書き換えることはできない．したがって，変形前のデータとは別に変形後のデータを新たに用意する必要がある．しかし，それでは複製ごとにオリジナルと同量のデータが発生してしまい，一時的に空間計算量が大幅に増加してしまう可能性がある．特に，このステップを並列処理する場合，このような一時的な空間計算量の増加が各スレッドで発生することは致命的な問題となり得る．そこで提案手法では，プリミティブとなる点群に対して変形と描画の処理を交互に繰り返すことで，新たに用意するデータを最大で 12B にとどめている．

4 結果

提案手法は単一の Nvidia GeForce GTX 980 GPU 上で CUDA を使用して実装した．

空間計算量について，コピーではオリジナルのデータサイズに比例して複製のデータサイズも増加してしまうのに対し，提案手法ではオリジナルのデータサイズに依存せず複製のデータサイズは常に一定となった．これは，コピーではオリジナルと同量のデータで複製を表現しているのに対し，提案手法では定数個の制御点で複製を表現しているという違いによる．ここで，オリジナルのデータサイズを n としたとき，コピーの空間計算量が $O(n)$ であるのに対し，提案手法は $O(1)$ となる．

インスタンスはオリジナルを再利用することで複製を表現しているため，オリジナルを共有する複製はすべて同一の形状になってしまう．一方で，提案手法は制御点を個別に移動することにより，すべての複製を独立して変形することが可能である．図 1 は個別の変形を伴う 8 体の Stanford Bunny を示している．通常，このような表現にはコピーが用いられるが，コピーでは提案手法のように空間計算量を小さくとどめることはできない．

関連研究の Shell Maps は，複製を配置する際サーフェスを参照するため，サーフェスに沿う形にしか配置できない．一方で，提案手法は図 2 のように任意空間への複製の充填のような自由度の高い配置にも対応可能である．これは，群衆シミュレーションへの適用の可能性も示唆している．

図 3 の画像は提案手法を繊維レベルの布に適用した例である．2,097,152 点群（50 MB）から成る布のスウォッチをオリジナルとして登録し，クロスシミュレーションに基づいて制御点組を 10,000 組配置し描画した．これをコピーで表現するには 503 GB を要するが，提案手法では 58 MB で表現が可能である．

5 結論

個別の変形を伴う複製描画のための空間計算量の削減手法を提案した．オリジナルのデータサイズに依存しない複製の表現により，通常潤沢な計算資源を必要とするシーンを単一の GPU で描画できることが確認できた．

提案手法の限界は，空間計算量の削減に伴い時間計算量がコピーと比較して約 3 倍増加してしまう点である．しかし，我々はこのことを悲観的には捉えていない．なぜなら，空間計算量が $O(n)$ から $O(1)$ に減少したことを考慮すると，時間計算量が約 3 倍増加したことは無視できるからである．

謝辞

本研究の一部は，マイクロソフトリサーチアジア CORE14 の支援により実施された．

参考文献

[1] Serban D. Porumbescu, Brian Budge, Louis Feng, and Kenneth I. Joy: “Shell Maps,” *ACM Transactions on Graphics,* Volume 24, Issue 3, Pages 626–633, July 2005.

[2] Thomas W. Sederberg, and Scott R. Parry: “Free-Form Deformation of Solid Geometric Models,” in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques,* August 1986, Pages 151–160.

[3] 大河原 将, 藤代 一成：「自由形状変形を伴う複製物体のための空間計算量を重視した描画」, 画像電子学会ビジュアルコンピューティングワークショップ 2018 in 下呂温泉, 2018 年 12 月 (講演要旨：画像電子学会誌, Volume 48, Number 1, 2019 年 1 月掲載予定)