

自動生成された多重解像度格子を用いた自由形状変形

Free-Form Deformation with Automatically Generated Multiresolution Lattices

小野 豊*
Yutaka Ono

*東京大学
University of Tokyo

陳 炳宇*
Bing-Yu Chen

西田 友是*
Tomoyuki Nishita

†中国浙江大學
Zhejiang University

馮 結青†
Jieqing Feng

要旨

直感的で効率的な形状変形手法の開発は CG における重要な分野の一つである。そのような手法の一つである自由形状変形 (free-form deformation, FFD) を用いると、ユーザは 3 次元モデルを包含する格子と呼ばれる制御点をいくつか動かすことで、容易にモデルの変形を行うことができる。FFD はモデルの大域の変形にも局所の変形にも用いることができるが、ユーザは適切な格子をマニュアルで定義するか、さもなければ平行六面体のような単純な形状の格子を用いなければならない。ゆえに、どちらの種類の変形も可能であるような格子を自動的に生成する手法が必要である。そこで本研究では、そのような格子を自動的に生成する新しい FFD の手法を提案する。我々の手法により生成された格子の制御点を動かすことにより、ユーザは大域の変形も局所の変形も容易に行うことができ、さらに複数の格子を組み合わせて用いることで、大域的な変形から局所的な変形、またはその逆といったモデルの階層的な変形も可能となる。

1. はじめに

近年、映画やコンピュータゲームなどの多くの分野において、複雑かつ詳細な三次元モデルが幅広く用いられるようになってきている。精巧なモデルは多くの人々を魅了し、プロだけではなく、趣味でそのようなモデルを制作するアマチュアのユーザも増えてきている。ゆえに、有用な 3 次元形状のデザインや編集、アニメーション手法の開発が、CG においてますます重要となってきている。

そのような 3 次元モデルの変形手法の一つに FFD [7] がある。FFD はモデルの表現法に非依存な手法であり、ユーザは数学的なバックグラウンドを意識せずに変形を行うことができる。さらに、FFD を用いると、他の手法では困難な滑らかな形状変形を容易に行うことができるため、FFD はプロのユーザの間にもアマチュアのユーザの間にも広く普及している。

FFD の手順は以下のようなものである。まず、ユーザが格子と呼ばれる制御点集合を配置することにより、変形可能領域を定義する。次に、モデルを変形可能領域に対しパラメータ化する。その後、ユーザがいくつかの制御点を動かすと、変形可能領域が歪められ、その歪みが自動的にモデルへと伝わり、モデルが変形される。

ユーザが適切な格子を設計することで、FFD は直感的かつ効率的なモデルの変形を可能とする。例えば、変形対象のモデルの点全てを包含する制御点数の少ない格子を用いることで、曲げ、ねじりといった大域的な変形が可能となり、変形対象の形状に近い適当な密度を持った格子を用いることで、隆起を加えるような局所的な変形が可能になる。しかし、オリジナルの FFD の手法は、格子として平行六面体のみが使用可能であったため、適切な格子を設計することや、格子と変形対象のモデルの対応関係を理解することなどが容易ではなかった。

この問題を解決するため、より一般的な形状の格子および変形可能領域を使用可能にする手法がいくつか提案されているが、extended free-form deformation (EFFD)

[2] などにみられるように、オリジナルの FFD の柔軟性を損なっているものがほとんどであった。MacCracken らの手法 [4] は、その柔軟性を保ちつつ、任意の形状の格子を用いることを可能にしたが、空間的計算量が大きいため、局所の変形に用いるような多くの制御点を含む格子を使用することは、実際的には不可能であった。さらに、いままで提案されてきた手法のほとんどは、使用可能な格子を用いての変形手法のみを言及していたため、適切な格子を自動的に生成する手法はいまだ提案されていない。そのため、ユーザは格子をマニュアルで定義せねばならず、その生産性が下がってしまっていた。

そこで本研究では、FFD の格子を自動生成し、オリジナルの FFD の柔軟性を保ちつつ、ユーザによる 3 次元モデルの変形を支援する手法を提案する。我々の手法は、まずモデルのバウンディングボックスを生成し、それを初期格子とする。その初期格子から階層的に新たな格子を生成することで、多重解像度格子という格子集合を作り出し、ユーザがその中から目的に沿った適切な格子、すなわち大域的な変形のための少数の制御点を持つ格子や、局所的な変形のための変形対象領域の形状をよく近似する格子を選択できるようにする。さらに、変形前の格子だけでなく、ユーザによって変形された格子から新たな格子の生成を行うことで、階層的な変形を行うことができる。階層的な変形を用いることで、格子をバウンディングボックスから再定義することなく、大域的な変形から局所的な変形、またはその逆といったようにモデルを階層的に変形することが可能となる。我々の手法はこのようにして、ユーザをマニュアルの格子の設定から解放し、FFD の直感性の強化とユーザの生産性の向上を可能にする。

2. 多重解像度格子の自動生成

我々の手法の詳細を述べる前に、本稿において使用される用語を [4] を参考に定義する。

- 格子とは、制御点集合と、その接続関係を定義する制御点対の集合の組である。
- 格子の辺とは、接続されている制御点対である。

- 格子の面とは、辺で接続された閉路である。
- 格子のセルとは、面によって囲まれる閉じた空間領域である。

また、以下を満たす格子を妥当な格子と呼ぶ。

- 辺上に位置する制御点は、全てその辺により接続されている。
- 全てのセルが閉じており、穴を持たない。
- 格子が自己交差していない。

2.1 八分木分割格子

我々の手法は、与えられたモデルの中の、変形対象の領域に対し、その領域の形状を階層的に近似する格子群をバウンディングボックスの3次元的な分割の繰り返しにより生成する。最低レベルの格子であるバウンディングボックスには、各軸に平行なバウンディングボックスのほかにも oriented bounding box (OBB) [3] や、体積最小のバウンディングボックスを用いることができる。OBB は、各軸がモデルの主成分の向きに沿っているため、多くの場合において、ユーザが最も変形を行い易い最低レベルの格子である。

最低レベルの格子を定義後、以下で定義する八分木分割規則を繰り返し用いて多重解像度格子を生成する。この規則は以下のような3種の新しい制御点を生成する。

- セル制御点: セルを定義する制御点の平均値で定まる点
- 面制御点: 面を定義する制御点の平均値で定まる点
- 辺制御点: 辺を定義する2つの制御点の中点

各分割のステップで、上記の3種の制御点が格子の各セル、面、辺についてそれぞれ生成され、それらの新しい制御点と古い制御点を再接続することで、より細かな格子が得られる。再接続は、(1) 新しいセル制御点を、そのセルを定義する各面から生成された新しい面制御点と結び、(2) 新しい面制御点を、その面を定義する各辺から生成された新しい辺制御点と結び、(3) 新しい辺制御点を、その辺を定義する古い制御点と結びことにより行われる。

上記の八分木分割規則を用いて、以下のように多重解像度格子を生成する。

1. 最低レベルの格子を定義する。
2. 現在のレベルの格子に八分木分割規則を適用し、新しい格子を生成する。
3. 新しく得られた格子から、モデルの点を一つも含まないセルを削除し、それにより得られた格子を一レベル細かな格子とする。
4. ユーザが指定した詳細度が得られるまで、ステップ2と3を繰り返す。

以下では、上の手法により生成された多重解像度格子を八分木分割格子と呼ぶ。

上の手続きにより生成された格子は、図1で示すように、与えられたモデルの形状を階層的に近似していくように配列された制御点からなるので、ユーザはどの

ようなレベルにおいても容易かつ直感的に変形を行うことができる。さらに、八分木分割規則は変形前の格子だけでなく、変形後の格子にも適用可能であるため、4節で述べる階層的な変形が可能となる。

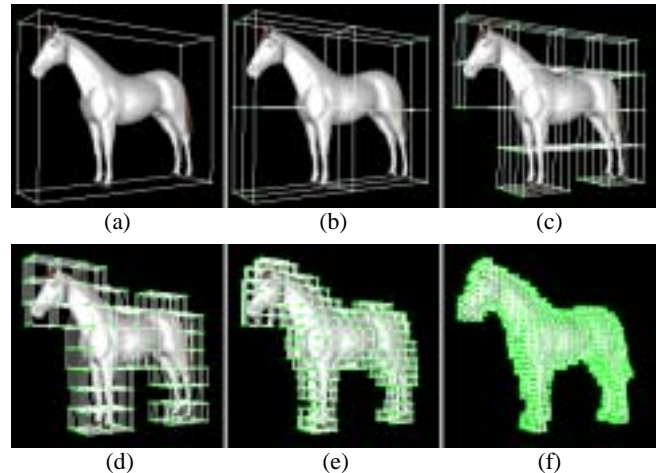


図1: 馬のモデル全体に対して生成した八分木分割格子。(a): 最低レベルの格子。(b) ~ (f): 分割により得られた格子。

2.2 八分木分割格子の拡張

八分木分割格子は、階層的にモデルの形状を近似していくが、格子のレベルが低く、各セルの大きさがモデルの点によって占められる領域に比べ非常に大きい場合などに、境界部においてあまりよい形状近似が得られないことがある。モデルと格子の対応を直感的に理解するためには、格子がなるべくモデルの形状に近いことが望まれるので、このような状態は改善する必要がある。そこで、境界部の制御点に境界軸方向への移動の自由度を与え、ある一定のオフセット値を保ったまま、モデルのできるだけ近くにそれらを配置できるようにした。この手法によって生成された八分木分割格子を以下では収縮八分木分割格子と呼ぶ(図2(a))。収縮八分木分割格子は、少数のセルよりなる低いレベルの格子を用いて変形を行うときに有用である。一方、高いレベルの格子をこの手法により生成した場合、適切なオフセット値が見つからなければ、散乱したような見た目の制御点を配置するので、通常収縮八分木分割格子は、大域的な変形にのみ適用する。

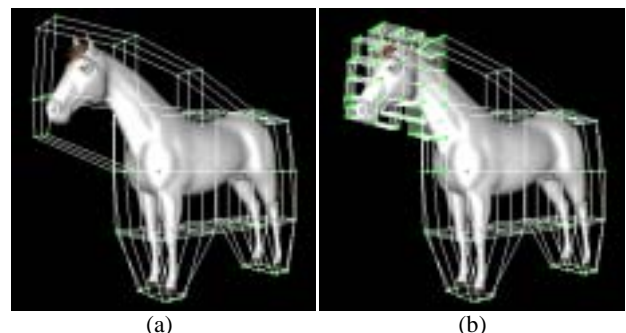


図2: 図1(c)に対応する、(a): 収縮八分木分割格子と (b): 局所八分木分割格子。

ユーザは多重解像度格子の中から適切な格子を選択す

ることで、どのようなレベルの変形も行うことができるが、ユーザが変形対象の一部のみに局所的な変形をほどこせるようにするため、我々の手法では、八分木分割規則を格子の一部のユーザが指定したセルのみに適用することを許している。この手法は [5] で提案されている局所的な分割と同様のものである。これによって格子の局所的な部分の位相的な構造は複雑になるが、結果として得られる格子は図 3 に示すように依然として妥当である。この手法により、図 2 (b) のように、より柔軟な形状の格子が利用可能となり、ユーザがモデルの一部分を局所的に変形し、残りの部分を大域的に変形するといったことを、格子の再定義をすることなしに一つの格子で可能とする。

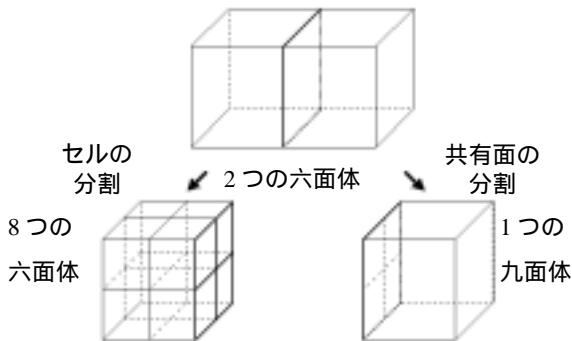


図 3: 六面体の局所的な分割。

3. 一つの格子を用いての変形手法

提案法の変形手法は、前節で述べた多重解像度格子だけではなく、2節で述べた任意の妥当な格子に対して適用可能である。我々は、格子により定義される変形可能領域に対し、[4] で提案された Catmull-Clark 細分割ボリュームの手法を用いてモデルの各点をパラメータ化し、それを用いて変形を行う。ところで、B スプラインボリュームは Catmull-Clark 細分割ボリュームの特別なケースなので、格子が六面体セルのみにより定義されている場合は、格子を B スプラインボリュームとみなすことができるが、格子が局所的に分割されているセルを持つ場合、B スプラインボリュームとしてパラメータ化を行うことができない。そのため、提案手法では Catmull-Clark 細分割を用いる。Catmull-Clark 細分割規則を繰り返し格子に適用することで、なめらかな変形可能領域を次第に近似していく細分割格子列が得られる。図 6 (c) と (e) はそれぞれ (b) と (d) の格子に Catmull-Clark 細分割を 3 回適用した細分割格子を示している。モデルの各点は、細分割格子列に含まれる格子のセルのうち、その点を含む体積最小のものを用いてパラメータ化される。ここで述べられている細分割格子は前節で述べた分割格子と異なることに注意してほしい。前者の格子はパラメータ化のためにのみ用いられるのに対し、後者は実際に表示され、ユーザによって変形が行われる格子である。

3.1 Catmull-Clark 細分割ボリュームの性質

Catmull-Clark 細分割規則の詳細は [4] を参照されたい。この規則により生成された細分割格子は、いくつかの特出すべき性質を持つ。一つの性質は各制御点の接続

関係に関するものである。これについて述べるために、制御点の、あるセルにおける価数を、そのセル内のその制御点を含む辺の数と定義する。格子に細分割を施すとき、細分割規則は各セルについて、各制御点で新しいセルを生成する。価数 v の制御点で生成される新しいセルは、 $2v$ 個の四辺形の面と、 2 個の価数 v の制御点、 $2v$ 個の価数 3 の制御点を生成する。例えば、価数 3 の制御点で生成される新しいセルは、 6 つの四辺形の面と、 8 つの価数 3 の制御点より成る新しいセル、すなわち六面体のセルを生成する。ここで、通常点を価数 3 の制御点、特異点を価数が 3 ではない制御点と定義すると、局所的な分割が施されていない単純な八分木分割格子に Catmull-Clark 細分割規則を適用した場合は、通常点と六面体セルのみが生成され、細分割を適用するたびにセルの数が 8 倍になることがわかる。特異点が現れるのはセルが局所的に分割されている場合のみであるため、細分割に必要なメモリ量も、細分割のたびにおよそ 8 倍になる。

もう一つの性質は、細分割格子の局所性に関するものである。Catmull-Clark 細分割曲面 [1] と同様、Catmull-Clark 細分割ボリュームの各制御点は、ボリューム中の一部の領域にのみ影響を及ぼすことができる。この性質を以下で再帰的に定義する 2 つの制御点により定義される二変数対称関数 $dist$ を用いて説明する。

- $dist(p, p) = 0$ 、ここで、 p は格子の制御点である。
- $dist(p, q) = 1$ 、ここで、 p と q は同じセルに含まれる異なる制御点である。
- $dist(p, q) = i + 1$ 、ここで、 p と q は異なる制御点で、 $q \notin \{r \mid dist(p, r) \leq i\}$ かつ、 $dist(p, s) = i$ 、 $dist(s, q) = 1$ なる制御点 s が存在する。

ここで、[4] の細分割規則から以下のことが導出できる。すなわち、格子のある制御点 p は、

$$D_{p,1} = \{q \mid dist(p, q) = 1\}$$

の外部の点によって構成されるセルや面、辺、制御点から生成される新しい制御点の位置には影響を及ぼさない。この性質は次節の変形のアルゴリズムで用いる。

3.2 変形のアルゴリズム

提案法では、[4] のアルゴリズムを基礎とした変形アルゴリズムを用いる。[4] の手法では、まずユーザが格子を定義し、モデルの各点を格子から得られる Catmull-Clark 細分割格子列を用いてパラメータ化する。すなわち、それぞれの点を、細分割格子列に含まれる格子のセルのうち、その点を含む体積最小のものへの参照と、そのセルの中での局所座標系によりタグづけする。その後、ユーザにより格子が変形され、変形された格子から細分割格子列が再計算され、それらと先に用いたパラメータによりモデルの各点が再配置される。しかし、パラメータ化と再配置に用いられる Catmull-Clark 細分割を格子に適用するには、多くの位相的な情報を格納する必要があり、しかもそれらの情報量は一度の細分割につき 8 倍に増加するため、図 1 の(e)や(f)にみられるような局所的な変形に用いる複雑な形状の格子には実質的に [4] の手法は適用不可能である。

幸い、ユーザが格子の一つの制御点 p を動かしたとき、

移動されるモデルの点は全て、

$$N_{p,2} = \bigcup_{i \leq 2} D_{p,i}$$

に対して生成される新たな制御点集合によって包含される。なぜなら、 p の位置は $N_{p,2}$ の境界部で生成される新しい制御点の位置に影響を及ぼさないからである。 $N_{p,2}$ の境界で生成される新しい制御点は、 $N_{p,3}$ による影響を受け、 $N_{p,2}$ の外部に位置しうが、 $N_{p,3}$ に含まれるモデルの点を、再配置される点の候補として考えれば十分である。さらに、 $N_{p,3}$ に対して生成される新しい制御点は $N_{p,4}$ により影響を受けるため、点の再配置を行うためには $D_{p,4}$ も考慮する必要がある。要約すると、ユーザが格子の一つの制御点 p を動かしたとき、図 4 に示すように $N_{p,4}$ により定義される領域を細分割し、 $N_{p,3}$ に含まれる点のパラメータ化と再配置を行えば十分である。

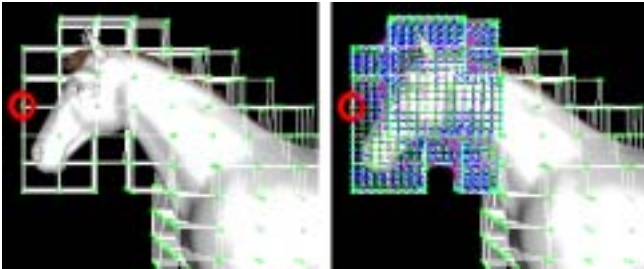


図 4: ユーザが、囲んである点を動かすときに生成される細分割格子。

これらを考慮した上で、多重解像度格子の一つを用いた変形手法を以下のように定める。

1. 変形対象のモデルに対し、多重解像度格子を生成する。
2. ユーザにより、多重解像度格子が一つ選択される。
3. 与えられたモデルと選択された格子を複製し、2つの組 $A = (L_A, M_A)$ と $B = (L_B, M_B)$ を生成する。以下のステップにおいて、組 A の格子 L_A とモデル M_A は実際に表示が行われ、ユーザによる変形を受けるが、組 B の格子 L_B とモデル M_B はパラメータ化のためにのみ用いられ、表示されることも、ユーザによって変形されることもない¹。
4. ユーザが格子 L_A の制御点集合 S_{p_A} を移動対象の制御点集合として特定する。
5. S_{p_A} に対応する格子 L_B の制御点集合 S_{p_B} を特定する。
6. 細分割格子列 $S_{L_B} = \{L_{B_1}, L_{B_2}, \dots, L_{B_n}\}$ を定義する。

ここで、 L_{B_i} は Catmull-Clark 細分割規則を $N_{S_{p_B},4} = \bigcup_{q \in S_{p_B}} N_{q,4}$ に対して i 回適用することで得られた細分割格子である。

7. $N_{S_{p_B},3}$ に含まれるモデル M_B の各点を、その点を含む S_{L_B} の格子のセルで最小のものを用いてパラメータ化する。
8. ユーザが S_{p_A} 中の点を動かすことにより、格子 L_A を変形する。
9. S_{L_B} を $N_{S_{p_B},4}$ に対応する格子 L_A の制御点集合 $N_{S_{p_A},4}$ を用いて再計算する。
10. $N_{S_{p_A},3}$ に含まれるモデル M_A の各点を、ステップ 7 で計算したモデル M_B の対応する点のパラメータを用いて再配置する。
11. ステップ 4 から 10 をユーザが変形を終了するまで繰り返す。

[4] のアルゴリズムと同様、上のアルゴリズムは変形可能領域を近似する Catmull-Clark 細分割を、ステップ 6 で使われている有限の定数値 n 回しか行わないため、近似的な変形のみ行うことができるが、変形結果は n を増加させることで改善される。提案法のアルゴリズムにおいて最も重要なのは、格子の一部の限られた領域のみを細分割するため、非常に多数のセルより成る格子に対しても適用可能であるという点である。一方、格子が持つ制御点数が十分少ない場合は、前処理として格子全体の細分割とモデルの全点のパラメータ化を一度だけ行えばよい。そのような場合、移動対象の制御点集合を代えるたびにパラメータ化を行う必要も、変形前の格子とモデルを保存しておく必要もない。

モデルの点のセル内での局所座標は、2通りの計算法で求められる。六面体のセルに対しては、三重線形補間を用い、他の種類のセルに対しては、セルを複数の四面体に分割し、その中で点を含む四面体について、4頂点の重みを計算する。

4. 複数の格子を用いての階層的な変形手法

ユーザが多重解像度格子の二つ以上の格子を使用した変形を行う場合、大域的な変形から局所的な変形、または局所的な変形から大域的な変形へとレベルを変えながらの変形となるため、変形手法は前節で述べた格子を一つだけ用いてのそれとは少し異なるものになる。

4.1 大域的な変形から局所的な変形

大域的な変形から局所的な変形を行う階層的な変形の手法は、以下のようなものである。

1. ユーザが低いレベルの格子を用いてモデルを大域的に変形する。
2. 変形された現在のレベルの格子に対し、八分木分割を施すことで、一レベル細かな格子を生成する。
3. ユーザが新しい格子を用いてモデルを局所的に変形する。
4. より局所的な変形が必要な場合、ステップ 2 と 3 を繰り返す。

3節のアルゴリズムを拡張してこのような階層的な変形を実現するのは比較的簡単ではあるが、八分木分割規則と Catmull-Clark 細分割規則の違い等により生じる

¹ ユーザによって変形が行われた格子は一般には妥当ではなく、パラメータの計算に用いることはできないので、 L_B と M_B が必要となる。

モデルの点の位置のずれの問題がある。

O と C をそれぞれ八分木分割と Catmull-Clark 細分割により定義される、低いレベルの格子から高いレベルの格子への写像とする。 L_0 を最低レベルの格子とし、 $L_i = O^i(L_0)$ をレベル i の多重解像度格子とする。変形を施す前に、まず現在のレベルの格子 L_i に対して Catmull-Clark 細分割規則を用いて、モデルの点集合 V のパラメータ P_i を $C^n(L_i)$ から求める必要がある。ここで、 n は Catmull-Clark 細分割の適用回数を表す有限の定数である。ユーザが L_i の変形を行い L_i' が得られたとすると、 V はパラメータ $P_i' = P_i$ を用いて再配置される。現在のレベルよりも局所的な変形を行うためにユーザが $L_{i+1}' = O(L_i')$ を生成したとすると、局所的な変形を行うために L_{i+1}' から P_{i+1}' を計算する必要がある。しかし、 L_{i+1}' は一般的には妥当ではないので、 P_{i+1}' を直接計算することは一般的にできないので、我々は $L_{i+1} = O(L_i)$ から計算した値 P_{i+1} を代わりに用いる。この場合、一般的に $P_{i+1} \neq P_{i+1}'$ となるので P_{i+1} により計算されたモデルの点の位置は、元の位置とわずかに異なる。これは O と C の違いにより生じる問題である。

この問題は多重解像度格子の生成において、八分木分割規則の代わりに Catmull-Clark 細分割規則を用いることで一部解消できるが、Catmull-Clark 細分割規則で生成される格子は時として散乱した配置の制御点から成るため、ユーザがそれらの制御点を動かして変形するのは容易ではない。さらに、その計算コストのため、高いレベルの格子を生成することは難しい。そこで、提案法では Catmull-Clark 細分割規則を多重解像度格子の生成に用いることはせず、上で述べた位置の違いを埋めるための参照テーブルを用いている。

4.2 局所的な変形から大域的な変形

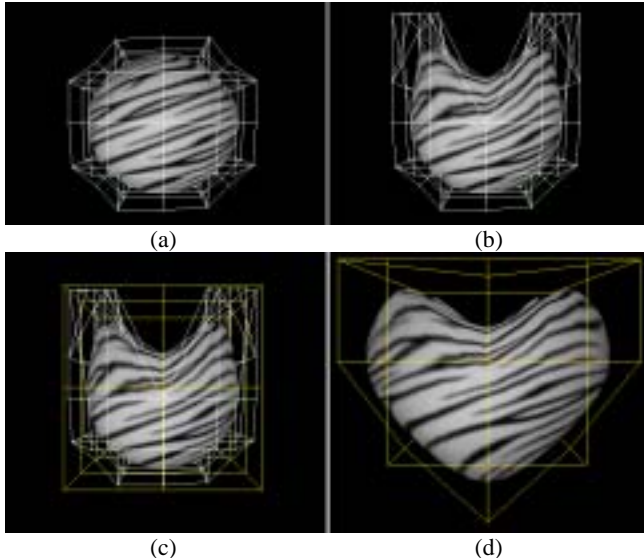


図 5: 局所的な変形から大域的な変形への階層的な変形。(a): モデルと高レベルの格子。(b): 局所的な変形がなされたモデル。(c): 高レベルな格子と変形されたモデルの両者を含む新しい低レベルの格子。(d): 低レベルの格子を用いて大域的な変形がなされたモデル。見やすくするために、高レベルな格子は表示していない。

高レベルの格子から、八分木分割規則の逆の操作をして得られる低レベルの格子は、一般に変形対象の領域を全ては含まないため、局所的な変形から大域的な変形を行うことは容易ではない。そこで、提案法ではモデルと高レベルの格子の両方を包含する新しい低レベルの格子を定義することで、この種の階層的な変形を可能にする。高レベルな格子は、新しい低レベルな格子の変形により、モデルとともに変形される。

図 5 は局所的な変形から大域的な変形を行う階層的な変形の例である。まず初めに、図 5 (a) の球のモデルが高レベルの格子を用いてユーザによって局所的に変形され、図 5 (b) のようになる。次に、図 5 (c) のように、高レベルの格子と変形されたモデルの両者を含む低レベルの格子が新たに生成され、図 5 (d) のようにモデルが大域的に変形される。ユーザはこの後、さらに大域的な変形を続けることも、変形された高レベルの格子を用いて局所的な変形に戻ることも可能である。

5. 結果

図 6 は、収縮八分木分割格子を用いての大域的な変形の例である。ユーザは収縮八分木分割格子を用いることで、容易に大域的な変形を行うことができる。図 7 は階層的な変形の例である。モデルは始めに大域的な変形が施され、変形された格子から生成された高レベルな格子を用いて次第に局所的な変形がなされていく。変形された格子から生成された格子は、変形されたモデルの形状の特徴を捉えた形となるので、ユーザはモデルの変形を直感的に続けることができる。これが我々の手法の主な利点の一つである。

六面体への Catmull-Clark 細分割の回数	セル数	必要なメモリ量 (byte)
0	1×1×1	1,296
1	2×2×2	7,376
2	4×4×4	48,336
3	8×8×8	345,488
4	16×16×16	2,601,744
5	32×32×32	20,170,256

表 1: 実装したシステムにおいて、Catmull-Clark 細分割ポリュームを格納するのに必要となるメモリ量。

八分木分割の回数	フレームレート (fps)		
	495 頂点のモデル	22,886 頂点のモデル	53,417 頂点のモデル
0	50.0	4.54	3.69
1	25.4	3.57	2.73
2	5.12	2.54	2.19
3	3.13	1.97	1.80
4	4.44	2.56	2.30
5	4.16	2.02	1.90

表 2: 異なる複雑さを持つポリゴンモデルを異なるレベルの八分木分割格子を用いて変形した際のフレームレート。

我々の実装した FFD システムの変形に必要なメモリ量と変形中のフレームレートをそれぞれ表 1 と表 2 に示す。測定は、Pentium4 1.7GHz、256MB メモリ、GeForce3 グラフィックスアクセラレータを持つ PC 上で行った。格子の生成時間については、ほとんど無視

できる程度のものなので測定していない。格子の位相的な情報の格納には、radial-edge data structure [8] の変種である、loop edge data structure (LEDS) [6] に似た構造を用いている。Catmull-Clark 細分割を 2 回 $N_{p,4}$ に適用すると、少なくとも $32 \times 32 \times 32$ 個のセルが生じるので、それを行うには 20MB 以上のメモリが必要となることが表 1 から読み取れる。表 2 に示すフレームレートは、3 種の複雑さのポリゴンモデルを、ユーザが八分木分割格子の一つの制御点を動かすことにより変形した際の値である。Catmull-Clark 細分割はそれぞれ 2 回適用されている。変形の際に、最も時間がかかっているのはパラメータ化と、モデルの点の再配置である。

これらの表から、我々のシステムは依然として多量のメモリを必要とし、他の FFD システムを用いた場合よりも低速であるといえるかもしれないが、我々のシステムを用いると、ユーザが格子を定義する手間を省くことができ、階層的な変形による柔軟な変形が可能となる。他の特徴として、表 2 に示すように、バウンディングボックスに 3 回以上八分木分割を施した場合、フレームレートはほとんど減少しないか、またはわずかに増加する傾向があるということである。これは、我々の手法では Catmull-Clark 細分割を格子の限られた領域に適用することと、八分木分割を繰り返すたびに一つ一つのセルが小さくなり、再配置が必要なモデルの点数が減ることによる。そのため、より複雑な格子に対しても、格子の複雑さにはあまり依存しない、ほぼ一定のコスト以下で変形を行うことが可能である。

6. まとめと今後の課題

本研究では、多重解像度格子の自動生成手法と、限られた区間のみを細分割することにより複雑な格子にも適用可能な変形手法、および多重解像度格子を用いたモデルの階層的な変形手法を提案した。我々のシステムを用いることで、格子の詳細度をボタンなどで調整する簡単なインターフェイスにより、初心者も熟練者も容易にモデルの変形を行うことができる。

図 5 や図 7 に示すように、我々の手法を用いることで、ユーザは大域的な変形と局所的な変形の両者を必要とする変形を容易に行うことができる。一方、他の手法を用いた場合、ユーザは適切な格子を変形の各レベルにおいて定義し、それがモデルにフィットするように制御点の微調整をする必要があるが、我々の多重解像度格子は自然にこの要件を満足することができる。

しかし、我々の手法は依然として多量のメモリを必要とするため、Catmull-Clark 細分割をごく少数回しか適用できない。Catmull-Clark 細分割により収束した空間を直接評価することにより、この欠点が解消できると思われる。その評価法とより使い易い格子の生成手法の開発が今後の課題である。

参考文献

- [1] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. In *Computer-Aided Design*, Vol. 10, pages 350-355, 1978.
- [2] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3D geometric modeling. In *ACM Computer Graphics (SIGGRAPH 90 Conference Proceedings)*, Vol. 24, No. 4,

pages 187-196, 1990.

- [3] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-Tree: a hierarchical structure for rapid interference detection. In *ACM SIGGRAPH 96 Conference Proceedings*, pages 171-180, 1996.
- [4] R. MacCracken and K. I. Joy. Free-form deformations with lattices of arbitrary topology. In *ACM SIGGRAPH 96 Conference Proceedings*, pages 181-188, 1996.
- [5] K. T. McDonnell, H. Qin, and R. A. Wlodarczyk. Virtual clay: a real-time sculpting system with haptic toolkits. In *ACM 2001 Symposium on Interactive 3D Graphics Proceedings*, pages 179-190, 2001.
- [6] S. A. McMains. Geometric algorithms and data representation for solid freeform fabrication. PhD thesis, Department of Computer Science, University of California, Berkeley, 2000.
- [7] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *ACM Computer Graphics (SIGGRAPH 86 Conference Proceedings)*, Vol. 20, No. 4, pages 151-160, 1986.
- [8] K. Weiler. The radial edge structure: a topological representation for non-manifold geometric boundary representations. In *Geometric Modeling for CAD Application (First IFIP WG5.2 Working Conference Rensselaerville)*, pages 3-36, 1988.

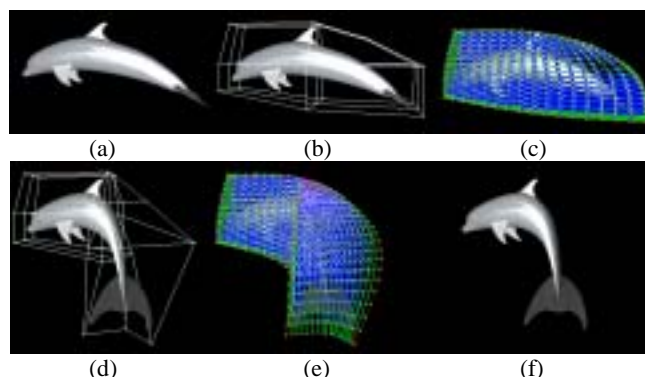


図 6: 収縮八分木分割格子を用いての大域的な変形。(c) と (e) で示されているのは、Catmull-Clark 細分割格子。

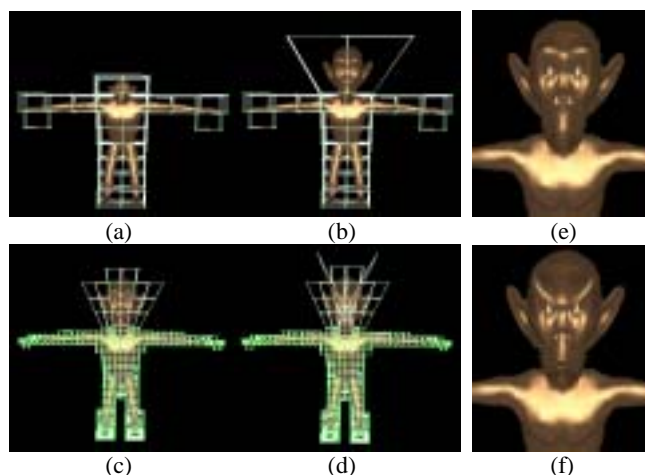


図 7: 階層的な変形。(a): 低レベルな格子と変形前のモデル。(b): 大域的に変形されたモデル。(c): 変形されたモデルと、(b) の格子から生成された高レベルな格子。(d): 局所的に変形されたモデル。(e): (b) と (c) のアップ。(f): (d) のアップ。