多重解像度補間メッシュ*

道川 隆士　金井　崇　藤田 将洋　千代倉 弘明
慶應義塾大学 環境情報学部

本論文は，メッシュモーフィングのための，多重解像度表現にもとづく新しいデータ表現について提案するものである．従来のメッシュモーフィングの手法は，合成操作をベースとしているため，補間メッシュの面の数が膨大になる等様々な問題を抱えていた．多重解像度補間メッシュは，ユーザが用意したベース補間メッシュを再帰的に細分割し，入力メッシュに近似させることで得ることができる．多重解像度補間メッシュはメッシュの構造が規則的なため，効率的なデータの保持が可能である．また本表現を用いることで従来とは異なる新しい補間方法が可能であることを示す．

1　はじめに

形状モデル間の滑らかな移行を実現する三次元モーフィング（*metamorphosis* or *morphing*）は，今やコンピュータグラフィックス (CG) での重要な研究トピックの一つである． 本研究では多角形，特に三角形メッシュ（以下メッシュ と呼ぶ）間のモーフィングについて扱う．

メッシュモーフィングを実現するにあたり次の二つの問題を解決する必要がある．ひとつはメッシュ間において 1 対 1 の対応関係を構築し補間メッシュを生成することである．もう一つは対応関係が構築された頂点を補間することで中間形状を生成することである．これらはそれぞれ対応問題，補間問題と呼ばれている [9]．これまでのメッシュモーフィングに関する研究の多く [9, 1, 5, 12, 8]　は前者の対応問題を主題としている．従来手法の共通点に補間メッシュを生成する際に合成操作を用いる点を挙げることができる．合成操作とはメッシュの和をとるものであり，この操作により頂点につき複数の座標値をもった補間メッシュを生成する．しかしこの合成操作には次に挙げる 3 点の問題点が存在する．第 1 点は合成操作を行うにあたりシビアな数値計算を必要とする点である．第 2 点は補間メッシュの面の数が膨大になり，かつメッシュの構造が不規則になる点である．第 3 点は，歪んだ三角形面が生成される点である．以上の結果，合成操作ベースの補間メッシュはレンダリングの質が低下する，補間制御が困難であるなどの問題があった．

本論文では上記の問題を解決するために，新たに多重解像度表現を導入した補間メッシュである，多重解像度補間メッシュ（*multiresolution interpolation mesh*）を提案する．本手法では，粗いベース補間メッシュを用意し，合成操作を用いるかわりに，再帰的にベース補間メッシュ細分割することで，合成操作を必要としない補間メッシュの生成を実現している．多重解像度補間メッシュには，次の 3 点の利点がある．第 1 点は，一様な 4 対 1 細分割によりメッシュの並びが規則的になる点である．これによりデータの効率的な格納が可能である．第 2 点は再帰的に細分割しているため補間形状に階層構造を持っている点である．

Multiresolution Interpolation Meshes, by Takashi MICHIKAWA, Takashi KANAI, Masahiro FUJITA, Hiroaki CHIYOKURA, Keio University, Faculty of Environmental Information.

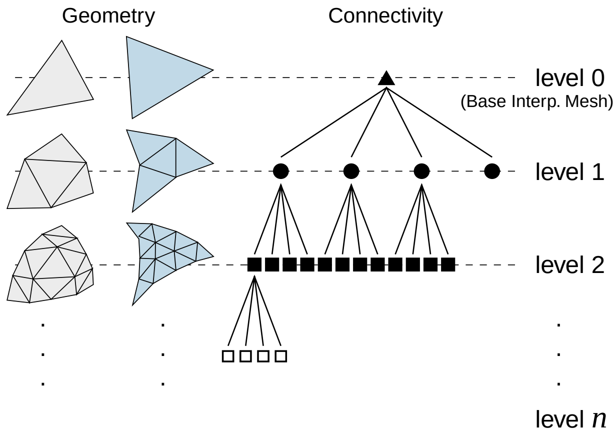


図 1: 多重解像度補間メッシュの四分木構造.

そのため様々なレベルにおける面構造を容易に取得することでき，補間形状の多重解像度表現が可能になる．第 3 点は局所細分割によりメッシュ数の増加を抑えることができる点である．

このように数々の利点を持つ多重解像度補間メッシュを用いることで，従来とは異なった補間手法が可能になる．本論文では多重解像度補間メッシュを用いて実現する新しい補間手法についても述べる．一つは多重解像度を利用した補間形状編集，もう一つは 3 つ以上のメッシュを対象としたマルチターゲットモーフィングである．また本手法は，もとのメッシュを近似するメッシュしかできないが，それを画像表示技術で補えることも示す．

2　多重解像度補間メッシュ

補間メッシュ（*interpolation mesh*）とは，メッシュと同様，頂点／エッジ／面のグラフ構造からなる形状表現である．各頂点には複数の座標値と，必要に応じて座標値間の軌跡の情報を持つ．多重解像度補間メッシュは，補間メッシュを多重解像度表現にしたものである．すなわち，ベースとなる補間メッシュ（ベース補間メッシュ）から，段階的に詳細化された半規則的な（*semi-regular*）構造を持つ．解像度は詳細化レベルにより管理され，ベース補間メッシュの面を根とする 4 分木構造により，各レベルの面

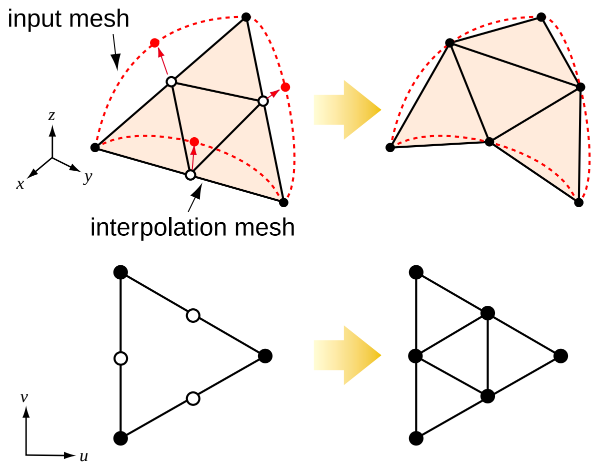


図 2: 細分割フィッティング処理.

が格納される (図1).

多重解像度補間メッシュの生成過程は次の 3 つのステップに分けられる．まずユーザーによりベース補間メッシュを生成する．次に，ベース補間メッシュを基に入力メッシュのグループ化，パラメータ化を行う．最後に，ベース補間メッシュに対して細分割フィッティング処理を行う．このうち最初の 2 つのステップは，[8] と似た手法を用いている．また細分割フィッティング処理は Guskov らの再メッシュ手法 [6] に近い．ただ，我々の手法は複数のメッシュを対象とした手法である点で異なっている．

2.1 ベース補間メッシュの作成

はじめに，複数の入力メッシュに対し，ベース補間メッシュとなる粗い多面体をユーザが作成する．各入力メッシュからそれぞれ頂点を一つずつ選択し，ベース補間メッシュの 1 つの頂点における座標値とする．次に，生成された頂点群を使ってベース補間メッシュの面 M^0 を作成する．このようなベース補間メッシュを生成する意義は 2 点ある．第 1 点はこの時点で粗い状態での 1 対 1 対応関係を構築するという点である．第 2 点は初期補間メッシュを通して最終的なモーフィング結果をデザインできるという点である．

このベース補間メッシュを生成するための指針としてはベース補間メッシュが入力メッシュを簡略化させたメッシュに近いメッシュであることが望ましい．入力メッシュとかけ離れたものであると細分割フィッティング処理の結果の質が悪化するためである．ただこの操作をメッシュの簡略化手法 [4] などを用いて自動化させることは困難である．一方のメッシュを簡略化させたメッシュを用いても，もう一方のメッシュにとって望ましいと限らないためである．

2.2 入力メッシュのグループ化とパラメータ化

各入力メッシュに対し，ベース補間メッシュのグラフ構造をもとに面を幾つかの領域（パッチ）にグループ化する．まず，ベース補間メッシュのエッジをもとに，入力メッシュ上の境界線を算出する．境界線は，エッジの両端点に対応する入力メッシュの頂点間の近似最短経路 [7] を計算することで得る．次に，算出された境界線を通る面を切断して三角形化する．最後に境界線で囲まれる面群を一つのパッチとしてまとめる．

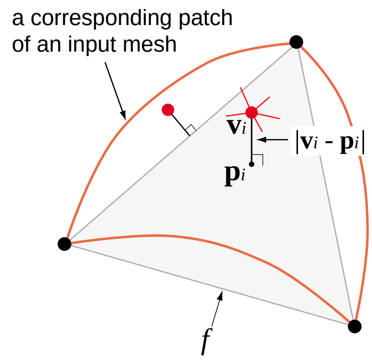


図 3: 誤差関数の設定.

次にグループ化された各パッチ毎に，メッシュのパラメータ化をすることで，二次元パラメータ空間上の座標値を計算する．ここでパラメータ化を行うのは次に述べる細分割フィッティング処理を安定的に行うためである．パラメータ化には Floater の手法 [3] を用いている．

2.3 細分割フィッティング処理

ベース補間メッシュをもとにした細分割フィッティング処理により，各入力メッシュを近似する多重解像度補間メッシュを生成する．図2に処理の手順を示す．ある詳細化レベル i からレベルを 1 つ上げた詳細化レベル $i+1$ の補間メッシュを得るには次の 2 ステップを経て行われる．まず，レベル i の補間メッシュの各面を 4 対 1 細分割により 4 つの小面に分割する．この時新しく生成される頂点のパラメータ座標も同時に計算する．この座標は分割前の稜線の両端点のパラメータ座標を足して 2 で割った値である．新しく生成される頂点の座標 $\mathbf{v}$ をパラメータ化の逆写像（重心写像）を利用して計算する．まず，新しく生成される頂点の包含面 $f = \{v_\alpha, v_\beta, v_\gamma\}$ を探し，重心座標値を (α, β, γ) とする．その係数を用いて $\mathbf{v}$ を式 (1) のとおり求める．

$$\mathbf{v} = \alpha\mathbf{v}_i + \beta\mathbf{v}_j + \gamma\mathbf{v}_k. \quad (1)$$

この計算により，新しい頂点は各入力メッシュの面上に置かれる．この処理を繰り返すことで，段階的に詳細化された多重解像度補間メッシュを得る．

局所細分化処理

ここで述べた一様な細分割処理では，近似精度の高い領域も分割してしまうので，必然的に面の数が増えてしまう．この問題に対処するために，より少ない面数で高い近似精度を保持できるよう，フィッティングアルゴリズムを修正する．

局所詳細化に先立ち，ものさしとして誤差関数 E を補間メッシュの各面 f に対して式 (2) のとおり設定する．

$$E^1(f) = \max\{|\mathbf{v}_i - \mathbf{p}_i|\}, \quad i = 1...n, \quad (2)$$

ここで $\mathbf{v}_i$ はパラメータ空間において f に含まれる入力メッシュ S^1 の頂点であり，$\mathbf{p}_i$ は $\mathbf{v}_i$ に最も近い f 上の点を表す．これをもう一つのメッシュ S^2 に対しても行い $E^2(f)$ を得る． そして最終的には双方の値のうち大きい方を採用する．ただこれでは形状の大きさによって値が変化してしまうので，バウンディングボックスの対角線の長さ $B(S^1)$， $B(S^2)$ で割ることにより平均化を行い， 大きい方を値として採用する (式 (3)).

$$E(f) = \max\left(\frac{E^1(f)}{B(S^1)}, \frac{E^2(f)}{B(S^2)}\right). \tag{3}$$

次に誤差関数を基に行う詳細化処理について述べる．本処理はユーザーによって設定した閾値 ϵ を基に $E(f)$ が閾値が超えた場合に細分割処理を行う様にしている．本処理は次の手順で行われる．

1. $\mathcal{M}^0$ にある全ての面をリストに加える．
2. リストにある全ての面に関して $E(f)$ を求める．
3. $E(f) \le \epsilon$ の面に関してはリストから削除する．
4. リストにあるループに関して 4 対 1 細分割処理を行いリストに登録する．
5. 2.， 3.， 4. をリストが空になるまで繰り返す．

最後に詳細化レベルが異なる面が隣接する部分に見られる T-Vertex を除去する．

2.4 フィッティング結果

我々の多重解像度補間メッシュの構築方法の評価のために，図4 に venus (24,000 faces) と tiger (8,064 faces) を入力したときの多重解像度補間メッシュの構築結果を示す．(a) には入力メッシュ S を， (b) にはこれらのメッシュからユーザが作成したベース補間メッシュ M^0 を示す．ベース補間メッシュは，対応頂点を形状全体に一様に置くことに気をつけて作成した．さらに特徴的な部位（たとえばトラの耳など）に関しては対応頂点の数を増やしている．これは，より少ない分割回数で近似の精度が高くなるようにするためである． (c) には一様細分割フィッティングによる多重解像度補間メッシュの構築例として，細分割レベルが 2 の多重解像度補間メッシュ (M^2) と レベルが 4 の多重解像度補間メッシュ (M^4) を示す． また (d) には局所細分割フィッティングによる多重解像度補間メッシュの構築例として，異なる ϵ を設定した場合 (0.01, 0.001) の多重解像度補間メッシュの構築例を示す．

表1 には，図4 の多重解像度補間メッシュの構築例におけるメッシュのサイズ，誤差，計算時間をそれぞれ示す．なお我々はアルゴリズムの中で最大距離による誤差評価を行っているが，より厳しい評価として誤差評価のツールである IRI-CNR Metro tool [2] を用いた．表の中の数値は，その中の mean square error (L^2-norm) をメッシュのバウンディングボックスの対角線の長さに対する % を示している． 表を見ると， 4 回の一様細分割フィッティング結果 M^4 に比べ局所細分割フィッティングにおける $\epsilon = 0.01$ の結果は，誤差が同程度であるにもかかわらず，サイズはおおよそ 40% で済んでいる．このことは，誤差評価による局所細分割フィッティング手法の効果が現れていると言える．図からも窪みなどの曲率の大きい特徴的な部位が重点的に細かく分割されているのが確認できる．ただ，どちらか一方の形状の誤差をもとに細分割を行う，という本手法の性質から，片方のメッシュのある領域が誤差の小さい領域であっても，もう一方のメッシュの対応する領域が誤差が大きいために余計に分割されてしまう，という現象が起こる．

計算時間は PentiumIII 1GHz CPU， 512MB Memory の AT 互換 PC 環境を使用して計測している．我々のコードは十分に最適化されていないため，計算時間の改良の余地は残されている． pre. と書いてある時間は，メッシュの領域分割のための近似最短経路の算出，メッシュのグループ化，およびメッシュのパラメータ化の時間である．このうち近似最短経路の算出が最も計算時間がかかる．これは各メッシュに対し計 81 本の経路を計算するためである．他の計算時間は，ベース補間メッシュ M^0 からの細分割フィッティングにかかった時間である．一様細分割フィッティングよりも，局所細分割フィッティングの方が相対的に計算時間がかかるが，これはメッシュの各頂点からの誤差を計測する必要があるためである．

3 多重解像度補間メッシュを用いたモーフィング

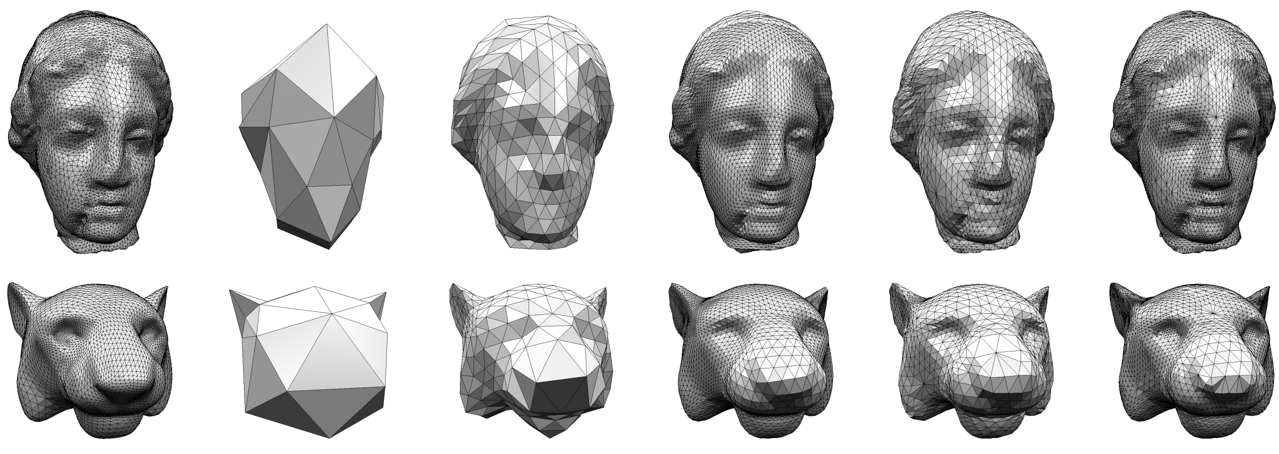
多重解像度補間メッシュは，リアルタイム 3D モーフィングのための大変強力なツールである．このことを示すために，本節では三つの補間手法，法線マップモーフィング，多重解像度表現を利用した補間経路の編集，そしてマルチターゲットモーフィングについて説明する．

3.1 法線マップの利用

多重解像度補間メッシュの形状は，各入力メッシュの近似形状にすぎないため，これらの入力メッシュの持つ正確かつ微細な凹凸を再現できない場合が多い．ただし，視覚的な効果が目的であるならば，この問題は法線マップ (*normal map*) を利用したテクスチャマッピング技術を利用することにより，ある程度改善することができる．

法線マップは，ポリゴンや曲面の法線情報を画像に変換して表現したものである．これは，これとは別に，高さマップを利用するバンプマッピングの場合，微細形状を表現するには，高さマップを法線摂動マップに変換して画像として形状に貼り付ける必要がある [10] ．ただし，詳細化されたもとのメッシュの法線情報が得られるのであれば，その情報を用いた方がより正確な視覚効果を得ることができる．

図5 に，多重解像度補間メッシュの形状のみの表示と，法線マップを用いたテクスチャマップによる表示の比較例を示す． (a) に入力メッシュを， (b) には一様細分割フィッティングによるレベル 4 の補間メッシュ M^4 （面数



S M^0 M^2 M^4 $\epsilon = .01$ $\epsilon = .001$

(a) (b) (c) (d)

図 4: 細分割フィッティング処理結果. (a) 入力メッシュ (venus,tiger). (b) ベース補間メッシュ (c) 一様細分割フィッティンッグ結果. 左: M^2, 右: M^4. (d) 局所細分化. 左: $\epsilon = .01$, 右:$\epsilon = .001$.

	size (#faces)						error (%)				
	S	M^0	M^2	M^4	$\epsilon = .01$	$\epsilon = .001$	M^0	M^2	M^4	$\epsilon = .01$	$\epsilon = .001$
venus	24,000	54	864	13,824	5,048	19,784	3.81	0.50	0.09	0.09	0.04
tiger	8,064						3.59	0.72	0.12	0.15	0.07

	time (sec.)				
	pre.	M^2	M^4	$\epsilon = .01$	$\epsilon = .001$
	51.2	1.4	32.3	20.9	84.1

表 1: 図 4 に対するメッシュの面の数，誤差，計算時間.

13，824) を示す．(c), (d) には，それぞれレベル 3 (M^2) ，レベル 4 (M^4) の補間メッシュに法線マップを適用した結果を示す．法線マップを適用することにより，M^3 でも入力メッシュとのかなりの視覚的均一性が得られることがわかる．多重解像度補間メッシュでは，四分木構造により解像度を自由に行き来できるので，どのレベルの補間メッシュでも，容易に同じ画像を貼り付けて表示することが可能であり，表示の際の LOD 制御に有効である．

3.2 多重解像度表現の利用

我々の多重解像度補間メッシュは多重解像度表現であるため，様々な利点を持つが，ここではそのうちの二つの利点について議論する．

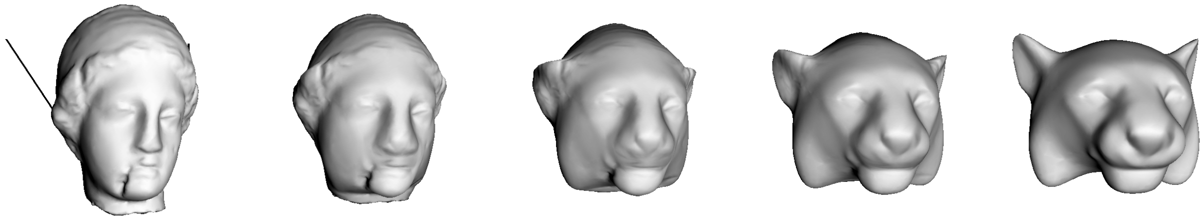
一つは，データとして詳細化部分での接続性情報を保持する必要がない点である．多重解像度補間メッシュにおいては，面は細分割接続性により管理されるため，面のインデックス情報はアプリケーション側で自動的に生成できる．データとして必要なのは，最も粗いベース補間メッシュのメッシュと，解像度レベル，パッチ数などのいくつかのヘッダ情報，および差分ベクトルである．このことにより，通常のメッシュデータフォーマットに比べ，データ量を大幅に削減することができる．例えば，図4 の例において，M^4 までのデータを 多重解像度補間メッシュにより管理すると，入力メッシュのデータサイズと比較して大体半分以下に抑えることが可能である．局所細分割フィッティングによる多重解像度補間メッシュの場合は，一様細分割の場合に比べ，頂点 1 つにつきそれが存在する位置を指定する整数値を 1 つ増やす必要があるので，データサイズが若干大きくなる．

もう一つは，多重解像度表現を利用した補間制御が行えることである．多重解像度補間メッシュのある詳細化レベル i において新しく生成される頂点 $\mathbf{p}^i$ は，一つ粗いレベルにおいて，$\mathbf{p}^i$ が定義されるエッジの端点 $\mathbf{p}_s^{i-1}, \mathbf{p}_e^{i-1}$ と差分ベクトル $\mathbf{D}_i$ を用いて式 (4) のように表すことができる．

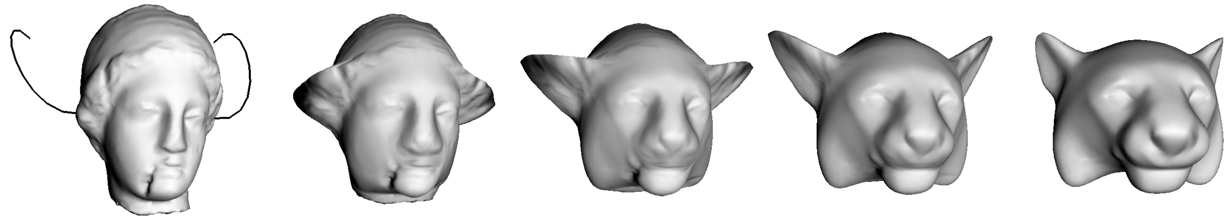
$$\mathbf{p}^i = \frac{1}{2}\left(\mathbf{p}_s^{i-1} + \mathbf{p}_e^{i-1}\right) + \mathbf{D}_i. \tag{4}$$

式 (4) により，多重解像度補間メッシュはベース補間メッシュと差分ベクトルの集合により書き直すことができる．多重解像度表現を利用した編集に関する研究 [14, 11] がそうであるように，粗いレベルの補間メッシュの補間経路を修正し，詳細化のための差分を加えることで，修正された細かい補間メッシュによる補間を実現できる．このように粗い状態で補間経路を設定して全体の補間を設定する手法は大渕ら [13] により提案されている．しかし，大渕らの手法は補間形状を 4 次元メッシュを切り出すことで表現されており，補間メッシュの形にはならない．

図6 に，多重解像度表現を利用した補間経路の簡単な制御例を示す．(a) では線形補間による補間例を，(b) は，ベース補間メッシュの補間経路を 3 次の ベジエ曲線で定義し，そのうち数点の頂点の経路に対して，中の 2 つの制御点を移動した場合の補間結果である．端の 2 点の制御点



(a)



(b)

図 6: 多重解像度表現による補間経路の制御.

(a) 入力メッシュ

(b) M^4

(c) M^3+ 法線マップ

(d) M^4+ 法線マップ

図 5: 法線マップによる多重解像度補間メッシュのテクスチャマッピング.

の位置は不変である.

3.3 マルチターゲットモーフィング

従来のメッシュモーフィングでは合成操作を用いるため，2 つの形状間の合成操作において，補間メッシュの面の数は，入力メッシュのうち面の数が多い方のおよそ 2.5 倍から 10 倍に膨れ上がることが知られている [8]．本手法では，補間メッシュの面の数はベース補間メッシュの面の数と細分割回数のみに依存し，入力メッシュには依存しない．そのため複数，特に 3 つ以上のメッシュを対象としたマルチターゲットモーフィング (*multi-target morphing*) を容易に実現できる．

n 個の入力メッシュ間の補間の場合，補間形状の座標値 $\mathbf{v}(t_1, t_2, ..., t_n)$ は式 (5) のとおりである．

$$\begin{aligned} \mathbf{v}(t_1, t_2, ..., t_n) &= \sum_{i=1}^{n} t_i \mathbf{v}^i, \quad \sum_{i=1}^{n} t_i = 1, \\ 0 \leq \quad t_i &\leq 1, \end{aligned} \tag{5}$$

ここで t_i は各メッシュにかかる重みである．

図7 に，図4 の結果にもう一つメッシュ (mannequin) を加えた，3 つの入力メッシュに対するマルチターゲットモーフィングの結果を示す．図7 は，M^4 における 補間結果である．尚この時の，補間メッシュのサイズは 13,824 であり，2 つの場合と変わらない結果が得られる．

4 まとめと今後の展望

本論文では，新しい補間メッシュ表現である多重解像度補間メッシュについて述べた．本手法はモーフィングに関して従来様々な問題を引き起こしてきた，合成操作を使わない方法であり，細分割フィッティング計算により補間メッシュを安定に計算することができる．また，細分割接続性に基づく階層的な半規則的構造をとることから，データの貯蔵に優れている，様々なレベルの解像度を持つメッシュを容易に取り出すことができる，補間の経路の編集ができる，など様々な利点をもたらすことを示した．また，法線マップを利用することにより，表示の効率化を図れることや，三つ以上のメッシュに対するモーフィングが行えるなど，従来の方法にない新しい特徴を示すことができた．

我々がこの新しい形状表現を生かすために，今後なすべきことは多い．そのうち特に興味があるのは，法線マップの補間による，モーフィングの際の様々な特殊効果の検討と，細分割曲面を含む非三角形メッシュへの適用，が挙げられる．

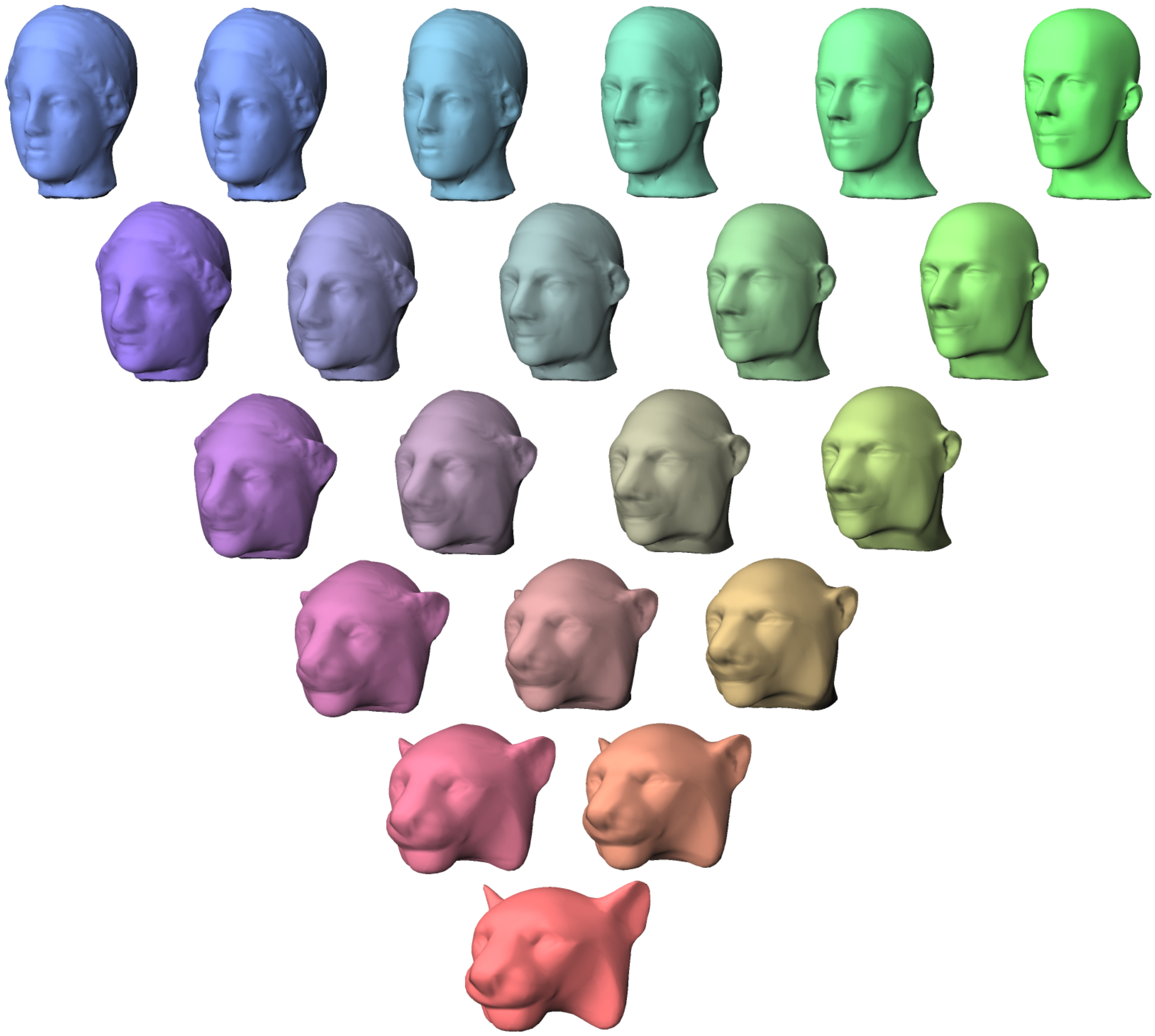


図 7: マルチターゲットモーフィング.

謝辞

本研究で用いたヴィーナスモデルは Cyberware Inc., マネキンモデルは Washington University, 虎モデルは Viewpoint DataLabs の提供によるものである．ここに謝意を表する．

参考文献

[1] M. Alexa. Merging polyhedral shapes with scattered features. *The Visual Computer*, 16(1):26–37, 2000.

[2] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[3] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250, 1997.

[4] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Computer Graphics (Proc. SIGGRAPH 97)*, pp. 209–216. ACM Press, New York, 1997.

[5] A. Gregory, A. State, M. Lin, D. Manocha, and M. Livingston. Interactive surface decomposition for polyhedral morphing. *The Visual Computer*, 15(9):453–470, 1999.

[6] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal meshes. In *Computer Graphics (Proc. SIGGRAPH2000)*, pp. 95–102. ACM Press, New York, 2000.

[7] T. Kanai and H. Suzuki. Approximate shortest path on a polyhedral surface based on selective refinement of the discrete graph and its applications. In *Proc. Geometric Modeling and Processing 2000*, pp. 241–250. IEEE CS Press, Los Alamitos CA, Apr. 2000.

[8] T. Kanai, H. Suzuki, and F. Kimura. Metamorphosis of arbitrary triangular meshes. *IEEE Computer Graphics and Applications*, 20(2):62–75, April 2000.

[9] J. R. Kent, W. E. Carlson, and R. E. Parent. Shape transformation for polyhedral objects. In *Computer Graphics (Proc. SIGGRAPH 92)*, pp. 47–54. ACM Press, New York, 1992.

[10] M. J. Kilgard. A practical and robust bump-mapping technique for today's GPUs. Game Developer's Conference 2000 Course "Advanced OpenGL Game Development", 2000. http://www.nvidia.com/Developer.nsf.

[11] L. P. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Computer Graphics (Proc. SIGGRAPH 98)*, pp. 105–114. ACM Press, New York, 1998.

[12] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Computer Graphics (Proc. SIGGRAPH 99)*, pp. 343–350. ACM Press, New York, 1999.

[13] R. Ohbuchi, Y. Kokojima, and S. Tahahashi. Blending shapes by using subdivision surfaces. *Computers and Graphics*, 25(1):41–58, 2001.

[14] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *Computer Graphics (Proc. SIGGRAPH 97)*, pp. 259–268. ACM Press, New York, 1997.