局所的3次元モーフィングに基づく3角形メッシュの融合演算

金井 崇[†] 鈴木宏正[††]
三谷 純[††] 木村文彦[††]

本論文では，3次元のモーフィングを利用した，メッシュの切り貼りに基づくモデリング手法を提案する．これは，従来行われてきた切り貼り操作にモーフィングのアイデアを適用し，ある形状の一部を接合部分を滑らかになるように，別の形状に接合することを実現するものである．またここでは，3つの幾何演算である剛体変換，スケール変換，制御線変形計算を組み合わせ，2つの境界を接合するためのアルゴリズムを提案している．本モデリングはリアルタイムでの処理を実現しており，ユーザが様々な形状をインタラクティブに構築することが可能になっている．

Triangular Mesh Fusion Based on Local 3D Metamorphosis

Takashi Kanai,[†] Hiromasa Suzuki,[††] Jun Mitani[††]
and Fumihiko Kimura[††]

This paper proposes a new mesh modeling scheme, called *mesh fusion*, based on three-dimensional (3D) mesh-based metamorphosis. We establish the attachment from a part of one mesh to a part of another with smooth boundaries, employing the traditional cutting and pasting operation in conjunction with a combination of meshes, applying the idea of 3D metamorphosis. We also offer an algorithm for adjusting two boundaries by using the combination of three geometrical operations *rigid transformation*, *scaling* and *deformation*. Our schematic offers a computation time swift enough that the user can create various shapes with interactive speed.

1. はじめに

三角形メッシュ（以下，メッシュと呼ぶ）で表される形状モデルは，コンピュータグラフィックスの分野でよく用いられている表現形式である．本研究の目的は，三角形メッシュによるモデリングの中で，2つの既存の形状（もしくはその一部）を使って，新しいモデルを創作するための方法を提供することである．ここでは，あるメッシュ（の一部）を別のメッシュ（の一部）上に接合する，という操作に焦点を当てる．この操作は，一般に切り貼り（カットアンドペースト，Cutting and Pasting）という操作に相当する．切り貼り操作は，2次元の描画用ソフトウェアやペイント用ソフトウェアなどでは必ず実装されている基本的な操作である．

三角形メッシュに対し切り貼り操作を実現するには，2つの形状の境界の繋ぎ目を滑らかに接合するための方法が必要である．このことを解決する1つの方法として，集合演算[12]を用いて2つのメッシュの和をとってから，境界付近にブレンディング[7]を適用する，という方法が考えられる．しかし，集合演算を実装している市販のモデリングシステムでは，プリミティブ形状もしくは自由曲面を対象としたものが多く，メッシュに対しては，集合演算とブレンディングの両方の機能を持つものは少ない．

一方，このような切り貼り操作は，メタボールなどの陰関数曲面形式では容易に実現することが可能である．陰関数曲面形式では，球や楕円体などのプリミティブ形状の演算により，境界を滑らかに接続したまま様々な形状を定義することができる．また過去には，基本立体間の切り貼り操作[15]，陰関数曲面へのテクスチャの貼り付け[14]，自由曲面間（Bスプライン曲面）の貼り付けやスライド操作[3]などの研究が見られる．しかしながら，これらはいずれもメッシュを対象にしたものではない．メッシュの場合，2つの部分形状どうし，あるいはその境界線どうしの対応づけがより複雑になり，したがって，上記の手法をそのまま適用するのは難しい．

我々が提案するメッシュ間の切り貼り（ここでは融合演算（*fusion*）と呼ぶことにする）は，これらの操

† 理化学研究所基礎科学特別研究員
Special Postdoctoral Researcher, RIKEN
†† 東京大学大学院工学系研究科
Graduate School of Engineering, University of Tokyo

作とは異なるタイプのものである．我々の手法はメッシュベースの 3 次元形状モーフィング[6),11)]の考え方に基づく．この手法は基本的には以下の 2 つのステップに分けて処理される．まず，あるメッシュ上の点がもう一方のメッシュ上のどの点に移るかを決定するために，面間の対応づけを行う必要がある．これは対応問題と呼ばれる．次に，メッシュ上のある点からもう一方のメッシュ上の点へ，どのような軌跡を描いて補間するかを決定する必要があり，これは補間問題と呼ばれる．このうち前者の対応問題を解決するために，我々が以前提案したモーフィング手法[8)]を用いることにする．

本手法で提案する融合演算の特徴は，後者の補間問題のための手法に工夫をしていることである．すなわち，切り貼りという点から考えれば，接合付近ではベースとなる形状から張り付けた形状へ形が変化し，その他は（ほぼ）張り付けた形状のままであってほしい．我々はこのことを，対応づけのときに作成したパラメータ空間を用いて，融合制御関数（*fusion contol function*）を定義し，この関数を修正することで解決している．この関数をインタラクティブに操作・修正することで，切り貼りとしての効果を含む様々な形状定義が可能となる．

また，一般的には異なる切り口の滑らかな接続を可能にするために，整合化アルゴリズムを提案する．これらの幾何演算は剛体変換，スケール変換，制御線変形計算の 3 つの操作から構成される．ここでは「できるだけ形状を変形したくない」という立場から，変形量をなるべく小さくするようにアルゴリズムを設計している．

さらに，ユーザにとってモデリングはインタラクティブに処理できることが重要である．本手法で提案しているすべてのアルゴリズムは高速であり，この条件を満足するものとなっている．

2. モーフィングに基づくメッシュの融合演算

2 つの三角形メッシュ $\mathcal{M}^1$，$\mathcal{M}^2$ のうち，これらのメッシュの部分領域である $\mathcal{F}^1 \subseteq \mathcal{M}^1$，$\mathcal{F}^2 \subseteq \mathcal{M}^2$ をタイル（*tile*）と呼ぶことにする．それぞれのタイルは円盤と位相同型なメッシュの一部の領域であるものとする．また，タイル $\mathcal{F}^1$，$\mathcal{F}^2$ の境界を $\partial\mathcal{F}^1$，$\partial\mathcal{F}^2$ として定義する．

図 1 に本手法の概要を示す．提案するメッシュの融合演算は，以下の 4 つのステップからなる．まず最初に，それぞれのメッシュから融合するための領域，すなわち，タイルを抽出する（図 1 (a)）．この抽出は，ユーザが 2 つのメッシュのそれぞれから，同じ数の頂点を閉ループになるよう指定することで行う．図 1 (a) の境界線上にある点が，ユーザにより指定された頂点である．2 つ目に $\partial\mathcal{F}^1$ と $\partial\mathcal{F}^2$ が滑らかに接合するよう，整合化アルゴリズムを $\mathcal{F}^1$ に適用する（図 1 (b)）．このアルゴリズムの詳細については 3 章で述べる．3 つ目に，2 つのタイルから，これらのパラメトリゼーションである 2 次元空間上の単位円内にある凸多角形 $\mathcal{H}^1$，$\mathcal{H}^2$ とその合成 $\mathcal{H}^c$，および合成タイル $\mathcal{F}^c$ を生成する（図 1 (c)）．$\mathcal{F}^c(\mathcal{H}^c)$ は $\mathcal{F}^1(\mathcal{H}^1)$ と $\mathcal{F}^2(\mathcal{H}^2)$ のグラフ構造を合わせ持つメッシュとなる．最後に，$\mathcal{H}^c$ を使った融合制御関数を定義し，この関数をインタラクティ

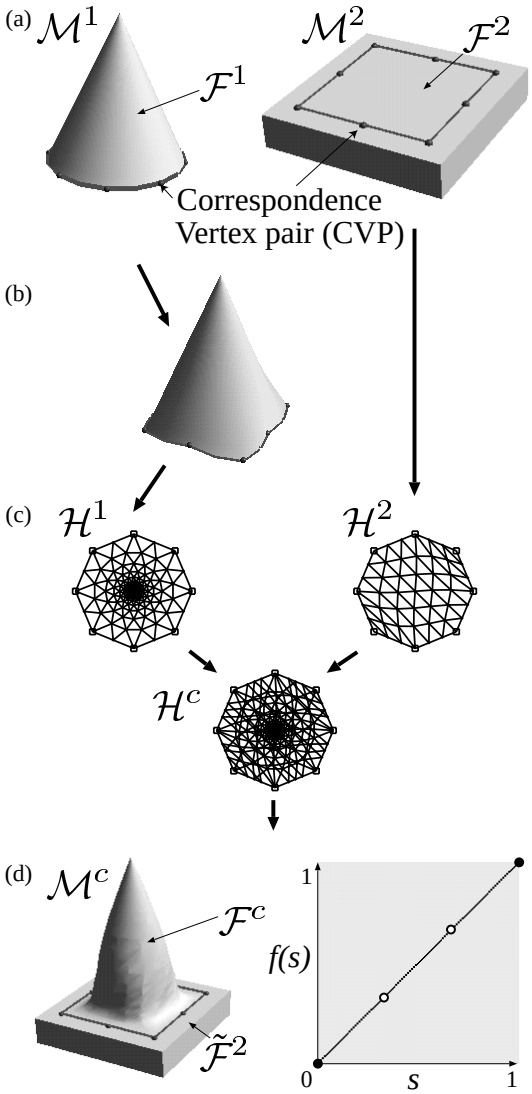


図 1 メッシュの切り貼り操作の手順
Fig. 1 An Overview of mesh fusion.

ブに操作することで最終形状を決定する（図1(d)）．最終形状 $\mathcal{M}^c$ は，$\mathcal{F}^c \cup \tilde{\mathcal{F}}^2$ ($\tilde{\mathcal{F}}^2 \equiv \mathcal{M}^2 \setminus \mathcal{F}^2$) という形で出力される．

2.1 タイルの抽出

2つのメッシュから融合すべきタイルを抽出する．この抽出プロセスは，ユーザによる頂点の対応づけとメッシュの切断，という2つのサブプロセスからなる．

頂点の対応づけのプロセスでは，ユーザは2つのメッシュの中からそれぞれ1つずつの頂点を選択する．この対応づけされた頂点を対応頂点（*correspondence vertex*, CV），2つの頂点のペアを対応頂点対（*correspondence vertex pair*, CVP）と呼ぶことにする．CVPは，2つのタイルの大まかな指定として，そして2.2節と3章で説明する幾何演算の参照として用いられる．これらのCVPの指定の操作を，融合したい領域を囲むように，メッシュ上にある閉ループを作るようにして行う（**図2**）．

メッシュの切断のプロセスでは，それぞれのメッシュ上で，指定したCVのうち隣接する頂点を通るようなメッシュ上の曲線を定義し，メッシュを分割する．ここでは，メッシュ上の曲線定義を2頂点間の最短経路問題として考える．しかし，最短経路を厳密に求める[13]のは計算負荷が大きく，またここでは必ずしも正確な経路が欲しいわけではない．

代わりに，我々は2つの頂点間の近似最短経路算出方法[18]を用いた．この方法はLanthierらの方法[10]に似ているが，我々の方法は収束計算に基づき，同じ近似精度の経路を算出するための空間量がより少ない，実装がより単純などの利点を持つ．最短経路は $\mathcal{M}$ の一部の頂点とエッジ，そして，付加的な頂点とエッジから構成される部分グラフの選択的詳細化により計算される．最短経路の探索は単にグラフのエッジの長さの加算のみで行われるため，誤差の蓄積しやすい数値計算（たとえば面の回転演算など）を必要としないのも利点である．図2において，2つの球の間に描かれている太線が，本経路探索法によって求めた境界である．ただし，求まった経路を構成する頂点やエッジ（それぞれ境界頂点，境界エッジと呼ぶ）は，メッシュの頂点，エッジではない場合がある．その場合は，境界エッジが通る面を，その境界エッジを含むように三角形分割する．

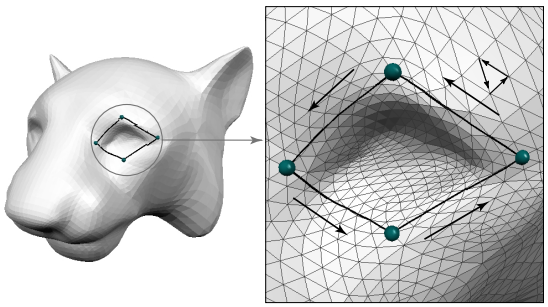


図2 CVを反時計回りに選択している様子
Fig. 2 Selecting CVs as a counter-clockwise direction.

すべての最短経路を計算すると，メッシュは2つの領域に分けられることになる．この領域のどちらがタイルになるかを決めるために，我々は頂点の対応づけのときに規則性を持たせるようにした．すなわち，メッシュの面の向きが，その頂点が反時計回りに定義されているときに外向きであるとすると，この頂点の対応づけも同様に反時計回りに定義する（図2）．すると，境界エッジに対して向きづけができるので，境界エッジの始点から終点に向かう方向に対して左側にある面群がタイルに属する面となる．タイルを構成する面群は，この性質を利用して欲張り法により集めることができる．

この一連のタイル抽出手法をとった主な理由として，ユーザの負荷をできるだけ軽減したい，という考えがある．ユーザは少数の頂点をピックするだけでよいし，生成されるタイルの境界は滑らかになる．境界上の点をすべてユーザが指定する方法だと，大変骨の折れる作業になるうえに，境界が $\mathcal{M}$ のエッジのみから構成されるので，凹凸の激しい線になる場合がある．

また我々は実装しなかったが，我々の方法よりもより緻密に境界を指定したい，というのであれば，Krishnamathyらの方法[9]も有効であると思われる．ただ，ここで生成される経路はメッシュの面上を通らないので，手法の修正が必要になる．

2.2 面間の対応づけの構築

メッシュから抽出された2つのタイルを融合するために，面上の任意の点間に対する1対1の対応づけを構築する．我々の手法の基本的な考え方は，2つのタイルを2次元平面上のある同じ凸多角形内に埋め込み，その凸多角形内で，対応を構築するためのモデルを生成する，ということである．

まず，2つのタイルの境界を構成する n 個の頂点（CV）を，2次元平面上の原点が重心となる同じ正 n 角形の頂点に写像する．残りの境界上の頂点は，頂点間のエッジの長さの比に等しくなるように n 角形の辺上に写される．タイルの内部の頂点は，調和写像の近似解法[4]を用いて求める．これを埋め込みタイルと呼ぶ．次に，これらの埋め込みタイル $\mathcal{H}^1$，$\mathcal{H}^2$ を合成し，2つのグラフ構造を合わせ持つような埋め込み合

成タイル $\mathcal{H}^c$ を生成する．2つのエッジの交点を求めてエッジを分割してから，もとの頂点と交点，それに分割されたエッジを用いて面の再構築を行う．このようにして定義された $\mathcal{H}^c$ から，重心座標を使い3次元の座標を求めることで，合成タイル $\mathcal{F}^c$ を生成する．$\mathcal{F}^c$ の各頂点は2つの3次元座標値を持ち，1つは $\mathcal{F}^1$ の，もう1つは $\mathcal{F}^2$ の面上の点に等しい．これらの頂点から融合形状の頂点は線形補間によって求められる．

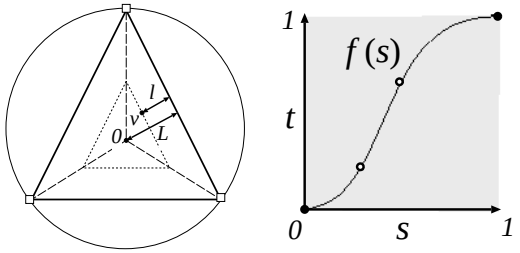
最終的なメッシュである $\mathcal{M}^c$ は，$\mathcal{F}^c$ と $\tilde{\mathcal{F}}^2$ の和集合となる．$\mathcal{F}^c$ と $\tilde{\mathcal{F}}^2$ を位相的に接続するために，$\partial\tilde{\mathcal{F}}^2$ に隣接する $\tilde{\mathcal{F}}^2$ の面を，$\mathcal{F}^c$ のエッジに従い分割する．

2.3 融合制御関数による融合形状の制御

メッシュの融合演算をメッシュの切り貼りという点から考えると，$\mathcal{F}^c$ の境界付近の形状は，ベースとなるタイル $\mathcal{F}^2$ のそれに近く，境界から離れた形状は張り付けるタイル $\mathcal{F}^1$ のそれに近くなってほしい．また，中間的な形状を生成する場合には，$\mathcal{F}^1$ と $\mathcal{F}^2$ の平均のような形状になってほしい．そこで，我々はこれらの要求を同時に満たすことのできる方法を提案する．埋め込み合成タイル $\mathcal{H}^c$ が単位円内にあることに注目し，融合の度合いを変化させるための融合制御関数（Fusion Control Function, FCF）を定義する．

図3 (a) に単位円と埋め込み合成タイル $\mathcal{H}^c$ の関係を示す．$\mathcal{H}^c$ の1つの頂点 v の持つパラメータ値を $s = l/L$ とする．ここで L は単位円の中心から境界（正 n 角形の辺）までの最短距離，l は v から境界までの最短距離である．s は境界に近い頂点ほど0に近付き，逆に円の中心に近付くほど1となるような，0から1の間の値を持つ．s は $\mathcal{H}^c$ の計算が終了すると同時に，各頂点につき一度だけ算出する．

FCF $f(s)$ は，2次元空間で4点を補間する3次のBスプライン曲線として定義する（図3(b)）．これらの点は，次の条件のもとで自由に動かせるようになっている．(1) 両端の2点はそれぞれ $s = 0$，$s = 1$ の直線上にある．(2) 2つの隣接する点の s 軸での間隔は0よりも大きい．(3) $f(s)$ は $0 \le f(s) \le 1$ である．特に，(3) の条件を満たすために，$f(s) < 0$ や $f(s) > 1$ の場合には，それぞれ $f(s) = 0$，$f(s) = 1$ になるよう補正している．



(a) (b)

図3 融合制御関数：(a) 埋め込み合成タイルと単位円との関係，(b) Bスプライン曲線で表される融合制御関数

Fig. 3 Fusion control function: (a) The relationship between an combined embedding and a unit circle, (b) A B-spline curve as a FCF.

合成タイルの頂点 $v \in \mathcal{F}^c$ の融合座標値を $\mathbf{v}^c$，対応する $\mathcal{F}^1$，$\mathcal{F}^2$ 上の座標値をそれぞれ $\mathbf{v}^1$，$\mathbf{v}^2$ とすると，$\mathbf{v}^c$ は次式で求められる：

$$\mathbf{v}^c = f(s)\mathbf{v}^1 + (1 - f(s))\mathbf{v}^2. \quad (1)$$

我々は上記のような式を設定したが，他の式を定義することも可能であるように思われる．特にBスプライン曲線による定義では，もし $\mathcal{F}^c$ の詳細な設定が必要であれば，移動できる点を増やすことも可能である．

図4 に，FCFを使用するうえでのガイドラインとなる単純な例題を示す．この図では，円錐の一部を球の一部に融合しており，境界はともに円形状である．図4(a)に，FCFが線形の関数の場合の結果を示す．$\mathcal{F}^c$ は境界から徐々に $\mathcal{F}^2$ から $\mathcal{F}^1$ に移り変わる．図4(b)に，$s = 0$ の近傍を除き，$f(s) \approx 1$ である場合の結果を示す．切り貼りとしての効果は，このように関数を設定することで表れる．図4(c)に $f'(0) \approx 0$ である場合の結果を示す．$\mathcal{F}^c$ の境界付近は $\mathcal{F}^2$ の境界付近の形状となる．図4(d)に $f(s) = 0.5$ の場合の結果を示す．この場合，$\mathcal{F}^c$ のすべての形状が，$\mathcal{F}^1$ と $\mathcal{F}^2$ のちょうど中間の値を持つ．

3. タイル境界の整合化

前章までで説明した融合演算の手順で，形状は位相的には接続することができた．ここでは，一般的な場合への対処として，2つの異なる幾何形状を持つタイルの境界 $\partial\mathcal{F}^1$，$\partial\mathcal{F}^2$ を，3次元境界形状として接合したい．境界の形状をユーザが修正することも可能であると考えられるが，その場合かなり骨の折れる作業となってしまう．そこで本章では，境界付近での滑らかな接合を実現するための3つの幾何演算について説明し，これらを効率的に組み合わせた整合化アルゴリズムを提案する．

ここで，境界の “整合化” および “滑らかな接合” とは，次の2つの条件を満たすことであるとする．1つ目に，2つの境界での形状が一致すること，2つ目は，それぞれの境界に垂直な接線方向（境界導関数ベクトル）が一致することである．本章では前者の問題について扱う．後者の問題については5章で述べる．

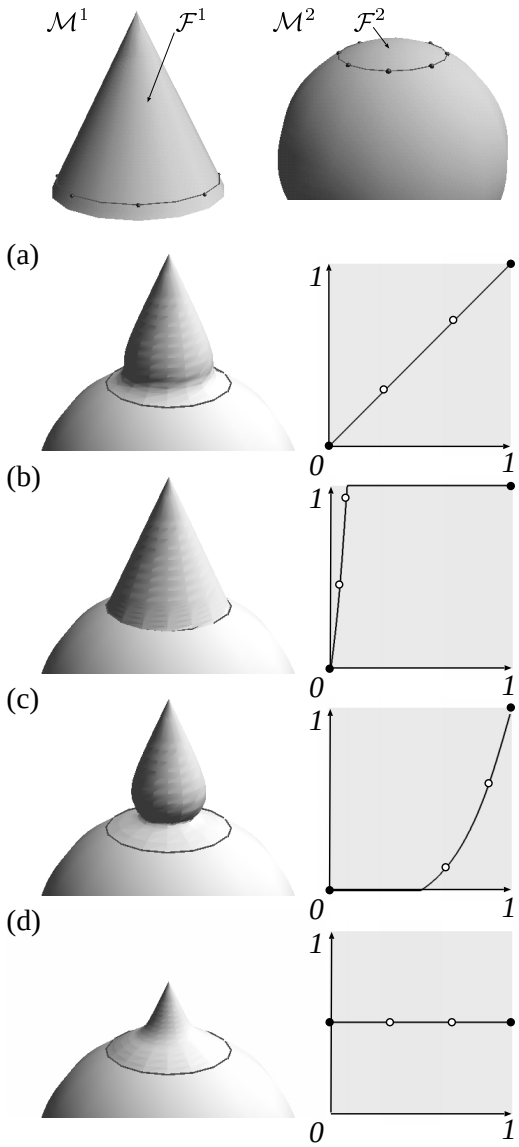


図 4 FCF を使った様々な融合形状
Fig. 4 Various shapes using a FCF.

3.1 剛体変換

形状をまったく変形しない幾何演算として，剛体変換を用いて $\partial\mathcal{F}^1$ を $\partial\mathcal{F}^2$ に近付けることを考える．ここでは CVP がなるべく近い位置にくるようにした．CVP の座標値をそれぞれ $\mathbf{p}_i^1, \mathbf{p}_i^2$ $(i = 1 \ldots m)$ とすると，

$$E_{trans} = \sum_{i=1}^{m} \left| \mathbf{p}_i^2 - (\mathbf{R}\mathbf{p}_i^1 + \mathbf{c}) \right|^2. \qquad (2)$$

が最小になるような回転行列 $\mathbf{R}$ と平行移動ベクトル

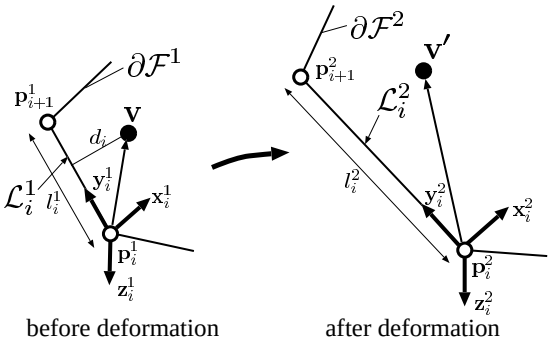


図 5 制御線変形 $\mathcal{L}_i^1 \rightarrow \mathcal{L}_i^2$ に従う $\mathcal{F}^1$ の頂点の変形
Fig. 5 The deformation of a vertex in $\mathcal{F}^1$ according to $\mathcal{L}_i^1 \rightarrow \mathcal{L}_i^2$.

$\mathbf{c}$ を求める．我々はこの最小自乗当てはめ問題を，特異値分解に基づくアルゴリズム[1)]を用いて解く．

3.2 スケール変換

形状を少々変形する幾何演算として，スケール変換によって $\partial\mathcal{F}^1$ を $\partial\mathcal{F}^2$ に近付けることを考える．これは，以下の式

$$E_{scale} = \sum_{i=1}^{m} \left| \mathbf{p}_i^2 - k\mathbf{p}_i^1 \right|^2. \qquad (3)$$

を最小とするようなスカラー値 k を，$\partial E_{scale}/\partial k = 0$ を解くことにより求める．

3.3 制御線変形計算

形状を大きく変形する幾何演算として，タイル全体を変形して境界を近付ける手法について考える．ここでは，$\partial\mathcal{F}^1$ の CV である $\mathbf{p}_i^1$ が $\mathbf{p}_i^2$ に（ほぼ）一致するような変形をしたい．我々は，画像のモーフィング手法[2)]を応用して，これらの境界の写像に基づく場を定義し，$\partial\mathcal{F}^1$ のすべての頂点を変形する制御線変形計算[17)]を用いる．

変形手法は，**図 5** のような方式で行われる．まず，各タイル境界の隣接する CV の間にエッジ $\mathcal{L}$ を定義する．変形したい頂点の座標値を $\mathbf{v}$ とすると，はじめに $\mathbf{v}$ を各エッジ $\mathcal{L}_i^1$ によって定義される局所座標系 $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ での値 $\mathbf{u}$ に変換する．次に，$\mathcal{L}_i$ から対応するエッジ $\mathcal{L}_i^2$ へ移動することによる新座標 $\mathbf{u}'$ を計算し，その後世界座標系に逆変換して座標値 $\mathbf{v}_i'$ を得る．変形後の座標値 $\mathbf{v}'$ は，これらの各エッジにおける変形後の座標値の重みづけ和で表す．まとめると，以下のような式で表される：

$$\mathbf{u}_i = \begin{pmatrix} \mathbf{x}_i^1 & \mathbf{y}_i^1 & \mathbf{z}_i^1 \end{pmatrix}^{-1} (\mathbf{v} - \mathbf{p}_i^1), \qquad (4)$$

$$\mathbf{u}_i' = \frac{l_i^2}{l_i^1}\mathbf{u}_i, \qquad (5)$$

$$\mathbf{v}'_i = (\,\mathbf{x}_i^2 \quad \mathbf{y}_i^2 \quad \mathbf{z}_i^2\,)\,\mathbf{u}'_i + \mathbf{p}_i^2, \tag{6}$$

$$\mathbf{v}' = \frac{1}{\sum_i w_i} \sum_i w_i \mathbf{v}'_i, \tag{7}$$

ここで，l_i^1，l_i^2 はそれぞれ $\mathcal{L}_i^1$，$\mathcal{L}_i^2$ の長さを示す．

各エッジ $\mathcal{L}_i$ における局所座標系は次のようにして定義する：まず $\mathcal{L}_i$ の始点 $\mathbf{p}_i$ を原点とし，始点から終点への方向の単位ベクトルを $\mathbf{y}_i$ とする．直交座標を作るために，$\mathbf{y}_i$ と，対応頂点群の重心から局所座標の原点方向の単位ベクトルとの外積を $\mathbf{z}_i$ として，$\mathbf{y}_i$ と $\mathbf{z}_i$ の外積を $\mathbf{x}_i$ とする．

また，変形の影響を決める重み関数 w_i は以下の式により定義する：

$$w_i = \left(1 + \frac{d_i}{l_i^1}\right)^{-n}. \tag{8}$$

ここで d_i は頂点 $\mathbf{v}$ から制御線 $\mathcal{L}_i^1$ までの距離を示す．この重み関数により，制御線 $\mathcal{L}_i^1$ が長いほど，また $\mathbf{v}$ との距離が短いほど w_i の値は大きくなる．さらに，**図 6** に様々な n に対する重み関数を示す．n の値が大きくなるほど重み関数の影響は局所的に，すなわち d_i/l_i^1 の値による重み関数の変化がより大きくなる．なおここでは，いくつかの実験から $n = 5$ の関数を用いている．

3.4 整合化アルゴリズム

合成を行う前に，貼り付けるタイル $\mathcal{F}^1$ の方に以下の 4 つのステップからなる整合化アルゴリズムを施す．

STEP1 剛体変換を一度だけ施す．変換した $\mathcal{F}^1$ の CV と，対応する $\mathcal{F}^2$ の CV との平均距離がある微小な閾値 ϵ 内に収まれば **STEP4** へ．

STEP2 スケール変換と剛体変換の順に実行する．**STEP1** と同様，平均距離が閾値 ϵ 内に収まれば **STEP4** へ．そうでなければ **STEP2** を繰り返す．

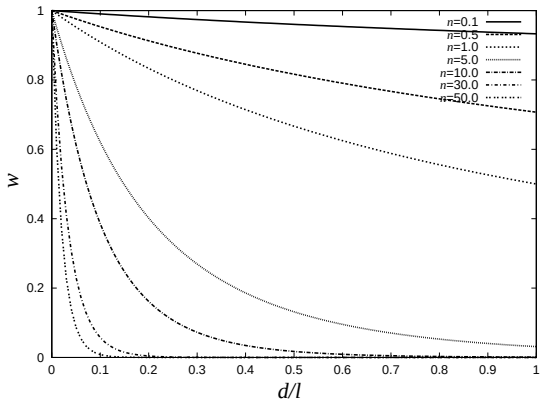


図 6 様々な n に対する重み関数 w_i
Fig. 6 A weight function w_i with various values of n.

STEP3 制御線変形計算を行う．

STEP4 処理を終了する．

このアルゴリズムは “できるだけ形状を変形したくない” ということを基本として設計されている．**STEP1** は，境界の形も大きさも同じ 2 つのタイルを繋ぎ合わせるためにまず実行される．この段階で処理が終了，すなわち境界が合致するようであれば，残りの 2 つの操作を使う必要はない．**STEP2** では，境界の形は同じであるが，大きさの異なる 2 つのタイルを繋ぎ合わせるために実行される．ある程度の許容量以下に 2 つの境界形状の差が収まるような繰り返し計算となるので，うまく収束するかが問題であるが，我々が実験したいくつかの例では，すべて 5 回以内の繰り返し計算で収束することができた．なおここでの閾値 ϵ は，$\mathcal{F}^1$ の境界エッジの長さの合計の 0.1%の値をとっている．**STEP3** は，境界の形状を一致させるために 1 回だけ行われる．このとき，**STEP2** までで CV 間の距離を最小にするような幾何演算をしているおかげで，ここでの境界付近の形状の変形量は小さくてすむ．

ただこのアルゴリズムは，あくまで，境界形状を “調整する” 程度の役割しか持たない．境界の形状が著しく異なる場合や，境界曲線の起伏が激しい場合など，タイルの大きな変形を必要とする場合には，このアルゴリズムを適用するのは適切ではない．式 (8) の n をどのような値に設定しても，境界付近は合致する一方で，形状全体の変形を制御するのは難しい．たとえば，n の値を大きくすると境界から離れたところまで変形が及ばず，また n を小さくすると今度はもとの形状を著しく損なってしまうということが起こる．

この問題に対処するために，上のアルゴリズムを適用する前に，あらかじめ FFD（Free-Form Deformation）[16] などを用いて，$\mathcal{F}^2$ の境界の形状に沿うようにユーザにより $\mathcal{F}^1$ を変形しておく．ただしその後に，境界の整合化が行われるので，FFD の変形の段階では境界を厳密に合わせる必要はない．

図 7（548 ページ参照）に上記の整合化アルゴリズムの各ステップを適用した結果を示す．$\mathcal{F}^1$ は FFD により曲げられた直方体のプレート形状に，立体文字群を貼り付けたものである．$\mathcal{F}^2$ は，ボトル形状の横のエッジをまたいだ四辺形の領域である．双方の 2 つの境界形状は同じ四辺形であるものの，そのアスペクト比は異なる．図 7 (b) には **STEP1** および **STEP2** を適用した後の結果を，図 7 (c) には **STEP3** を適用せずに接合した場合の結果を示す．図 7 (c) では，境界付近に凹凸が見られ，位相的には接続されているものの，

滑らかに接合していないことが分かる．図7(d)には**STEP3**を適用して接合した結果を示す．**STEP3**の効果により，立体文字がボトル形状の境界に整合するよう曲げられており，その結果，接合付近に望まれる幾何学的滑らかさが得られている様子が分かる．

4. 実　　装

我々は，提案した融合演算を行うためのプロトタイプシステムを，SGI OCTANE SI（MIPS R10000 175 MHz CPU）上で OpenGL と Motif を用いて C 言語により実装した．本システムは，2つの表示モードを用意しており，この2つの表示モードは，トグルボタンによって相互に切り替えられるようになっている．1つはタイル抽出のためのモードで，表示画面を2つ用意し，1つに $\mathcal{M}^1$ を，もう1つに $\mathcal{M}^2$ を表示している．このモードでは，CVP を生成・修正するための頂点の選択や，最短経路計算開始のボタンなどがマウスで行える．もう1つは，FCF を制御するためのモードで，1つに $\mathcal{M}^c$ を表示してあり，もう1つは FCF が表示してある．FCF の補間点を操作するごとに，$\mathcal{F}^c$ の形状をインタラクティブに変えられるようになっている．ただ，メッシュの FFD による変形機能は実装せず，Avid 社の SOFTIMAGE 3D^{TM} の形状変形機能（“Deforming with Lattices”）を用いて，事前に変形を行っている．

5. 結果と考察

図8に，マネキンの鼻から抽出した同じ領域に，円錐と半球の2つをそれぞれ融合させた場合の結果を示す．図8(b)，(c)には円錐との，図8(d)，(e)には半球とのそれぞれ2つずつの融合結果を示す．(b)，(d)は図4(b)の，(c)は図4(c)の，(e)は図4(d)の FCF を使用している．(b)，(d)において，鼻の形状は円錐もしくは半球とはいくぶん異なった形状が生成されているが，これは整合化によって境界が合致するように曲げられたからである．(c)は鼻の先端にしか円錐形状が表れていないが，これは関数が境界において $\mathcal{F}^2$ の連続性を重視した影響が現れている．

図9に，切り貼りとしての効果を表す2つの例を示す．(a)はある車のフロント付近の形状を別の車のに置き換えた例を，(b)はヘリコプタの前半分を鷲の頭に置き換えた例である．これらはいずれも図4(b)の FCF を使用している．

表1には，本例題において，最短経路の算出（Short. Path），境界の整合化（Adj.），面間の対応づけ（Corres.）に費やした各時間を表示している．この表にはユーザが CVP を生成する時間が表示されていないが，この作業は実際に頂点を数点ピックするだけの作業であり，1つの例題についてそれぞれ2〜3分程度しかかからなかった．また**表2**には，図1に示されている本例題のそれぞれの要素 $\mathcal{M}^1$，$\mathcal{M}^2$，$\mathcal{F}^1$，$\mathcal{F}^2$，$\mathcal{F}^c$，$\mathcal{M}^c$ の頂点と面の数，および CVP の数を示している．

表1を見ても分かるように，本研究で使用したアルゴリズムはいずれも高速であるため，インタラクティブな修正が可能である．本プロトタイプでは，境界の切口の形状を変えたいのであれば，境界における CVP の頂点を変えたり，増減したりしながら，その都度すべての処理を再計算するようになっているが，この操作はストレスなく行うことができる．また，面間の対応づけを行った後の FCF による修正は，F^c のすべての頂点の座標を1回ずつ書き換えるだけですむので，インタラクティブな処理が可能である．

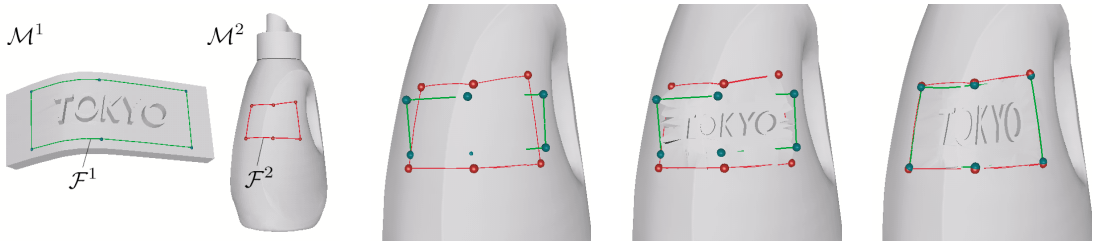
本手法の利点の1つとして，ユーザが境界における連続性の構築にあまり敏感になる必要はない，ということがあげられる．整合化のための制御線変形計算（3.3節）において，境界付近の連続性のための式，すなわち，境界に垂直な接線方向（横断導関数）を一致

表1　各例題における計算時間
Table 1　Computational times taken in our examples.

	Short. Path (Sec.)		Adj.	Corres.
	$\mathcal{M}^1$	$\mathcal{M}^2$	(Sec.)	(Sec.)
図7	2.95	2.77	0.19	0.75
図8(b)，(c)	0.45	7.05	0.08	0.87
図8(d)，(e)	0.33	7.05	0.05	0.53
図9(a)	3.67	0.87	1.03	0.52
図9(b)	2.32	2.93	0.63	3.53

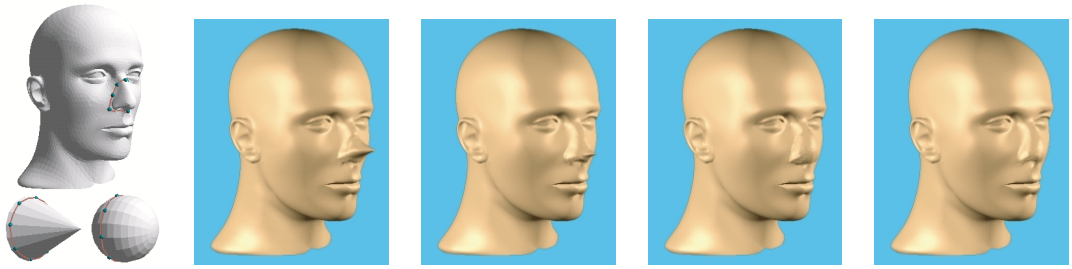
表2　各例題における頂点 v，面 f，CVP の数
Table 2　The number of vertices v, faces f, and CVPs in our examples.

	CVPs	$\mathcal{M}^1$		$\mathcal{M}^2$		$\mathcal{F}^1$ ($\mathcal{H}^1$)		$\mathcal{F}^2$ ($\mathcal{H}^2$)		$\mathcal{F}^c$ ($\mathcal{H}^c$)		$\mathcal{M}^c$	
		v	f	v	f	v	f	v	f	v	f	v	f
図7	6	1,293	2,582	3,589	7,084	875	1,677	55	74	1,835	3,569	5,141	10,244
図8(b)，(c)	8	434	864	6,769	13,472	393	760	404	721	2,072	4,041	8,592	16,906
図8(d)，(e)	8	242	480	6,769	13,472	121	216	452	805	1,682	3,249	8,182	16,059
図9(a)	7	6,775	13,502	1,300	2,580	521	1,003	62	103	1,502	2,961	2,759	5,470
図9(b)	5	5,625	11,167	5,188	10,332	1,541	3,045	1,158	2,244	8,325	16,548	12,425	24,666



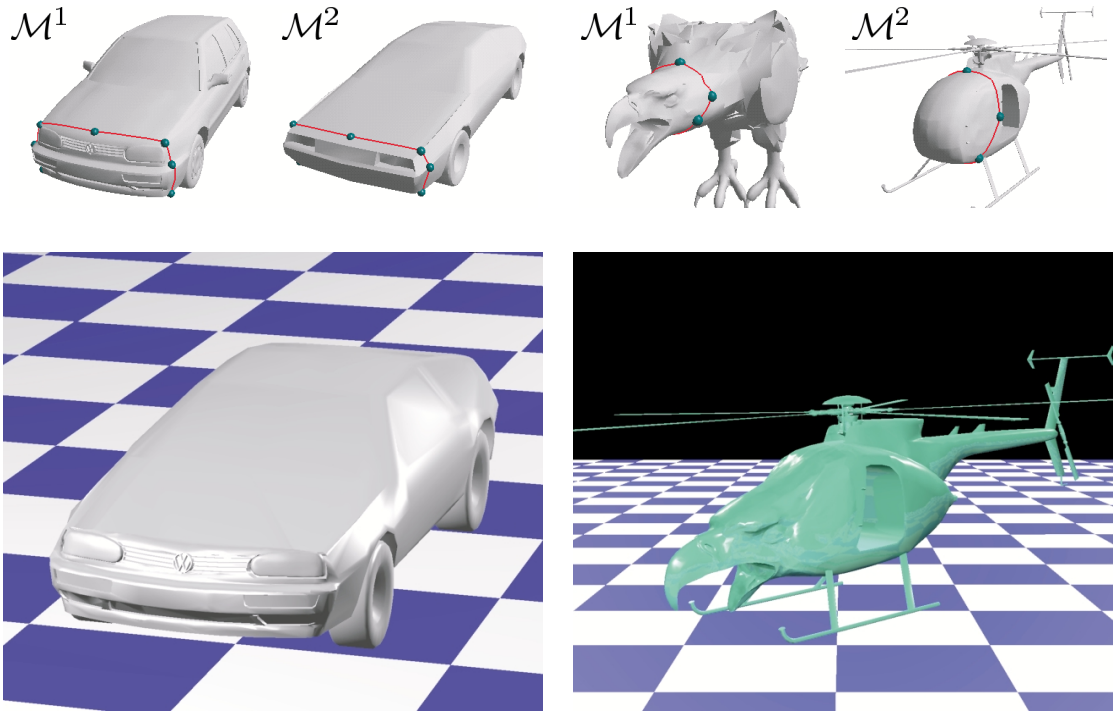
(a) (b) (c) (d)

図 7 整合化アルゴリズムを用いた 2 つの境界の接合
Fig. 7 The adjustment of two boundaries using our algorithm.



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

図 8 マネキンの鼻に対する様々な融合演算結果
Fig. 8 Various fusion results about a nose of “mannequin”.



(a) (b)

図 9 切り貼りとしての効果を示す融合演算結果：(a) “Delowagen”, (b) “Washicopter”
Fig. 9 Fusion results as the example of cutting and pasting: (a) “Delowagen”, (b) “Washicopter” (“Washi” in Japanese means “Eagle”).

するための式は，計算式 (4)〜(7) の中には含める必要がない．なぜならば，このことは FCF で制御できるからである．たとえば，図 4 (c)〜(d) のように $f'(0) \approx 0$ となるよう曲線を設計すれば，$\tilde{\mathcal{F}}^2$ の境界に垂直な接線方向が $\mathcal{F}^c$ の境界付近の形状に反映される．ただ図 4 (b) のように，必ずしも滑らかな接続が必要とされるわけではない．FCF による修正は，このように不連続な接続にも対処することができる．

本手法の 1 つの問題点として，ここで生成される $\mathcal{F}^c$ は面の数が劇的に増えてしまうことである．これは 3 次元モーフィングのための手法を採り入れているからである．たとえば切り貼りの場合では，明らかに，境界付近以外は面の数を多くする必要はない．これに対しては，関数の修正の後に，なるべく高速なメッシュの簡略化手法（たとえば Garland らの手法[5]など）を適用することが考えられる．しかし，インタラクティブな FCF の修正という観点から見ると，これらの操作の追加はいささか計算負荷が大きく，むしろその面の多い形状のまま表示する方が効率が良いように思われる．もし簡略化を行うとすれば，FCF の値により局所的に簡略化の度合いを変えられ，しかもインタラクティブなパフォーマンスに耐えうる程度の高速な手法が望ましい．

6. 結論と展望

この論文では，3 次元形状モーフィングに基づいたメッシュの新しい形状モデリングの手法である融合演算について示した．メッシュの一部の形状を切り貼りするだけでなく，メッシュを合成するためのモーフィングの概念を用いることで，滑らかな補間をともなう接着方法を実現している．さらに，異なる接合部分の形状を滑らかに接合するための整合化アルゴリズムを提案した．これらの一連の操作は形状の局所的な操作ですむために高速であり，さらに合成の度合いをインタラクティブに修正することが可能であることを，いくつかの例題を通じて確認した．

本手法の利点の 1 つに，比較的簡単にメッシュの切り貼りができることがあげられる．我々はこの気軽さをいかした本手法の拡張，特に工業用意匠デザインの発想支援などの応用に興味を持っている．

謝辞 例題で用いたマネキンは，ワシントン大学のグラフィックス研究室で作成されたメッシュデータを使用している．また，デロリアン，虎，ボトル，フォルクスワーゲン，ヘリコプター，鷲は Viewpoint DataLabs Inc. のメッシュデータである．テルアビブ大学の Daniel Cohen-Or 教授には，特異値分解に基づく剛体変換アルゴリズムについてご教授いただいた．また，この研究の一部は（財）大川情報通信基金の支援を受けた．ここに感謝の意を表する．

参 考 文 献

1) Arun, K.S., Huang, T.S. and Blostein, S.D.: Least-Squares Fitting of Two 3-D Point Sets, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.PAMI-9, No.5, pp.698–700 (1987).
2) Beier, T. and Neely, S.: Feature-Based Image Metamorphosis, *Computer Graphics* (*Proc. SIGGRAPH 92*), pp.35–42, ACM Press, New York (1992).
3) Chan, L.K.Y., Mann, S. and Bartels, R.: World Space Surface Pasting, *Proc. Graphics Interface '97*, pp.146–154, Morgan Kaufmann Publishers, San Francisco, CA (1997).
4) Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M. and Stuetzle, W.: Multiresolution Analysis of Arbitrary Meshes, *Computer Graphics* (*Proc. SIGGRAPH 95*), pp.173–182, ACM Press, New York (1995).
5) Garland, M. and Heckbert, P.S.: Surface Simplification Using Quadric Error Metrics, *Computer Graphics* (*Proc. SIGGRAPH 97*), pp.209–216, ACM Press, New York (1997).
6) Gomes, J., Darsa, L., Costa, B. and Velho, L.: *Warping and Morphing of Graphical Objects*, Morgan Kaufmann, San Francisco, CA (1998).
7) Hoschek, J. and Lasser, D.: *Fundamentals of Computer Aided Geometric Design*, A K Peters, Natick, Massachusetts (1993).
8) Kanai, T., Suzuki, H. and Kimura, F.: Three-dimensional Geometric Metamorphosis Based on Harmonic Maps, *The Visual Computer*, Vol.14, No.4, pp.166–176 (1998).
9) Krishnamurthy, V. and Levoy, M.: Fitting Smooth Surfaces to Dense Polygon Meshes, *Computer Graphics* (*Proc. SIGGRAPH 96*), pp.313–324, ACM Press, New York (1996).
10) Lanthier, M., Maheshwari, A. and Sack, J.-R.: Approximating Weighted Shortest Paths on Polyhedral Surfaces, *Proc. 13th ACM Sympo. on Computational Geometry*, pp.274–283, ACM Press, New York (1997).
11) Lazarus, F. and Verroust, A.: Three Dimensional Metamorphosis: A Survey, *The Visual Computer*, Vol.14, No.8/9, pp.373–389 (1998).
12) Mäntylä, M.: *An Introduction to Solid Modeling*, Computer Science Press, Rockville, Maryland (1988).
13) Mitchell, J.S.B., Mount, D.M. and Papadim-

itriou, C.H.: The Discrete Geodesic Problem, *SIAM J. Computing*, Vol.16, No.4, pp.647–668 (1987).
14) Pedersen, H.K.: Decorating Implicit Surfaces, *Computer Graphics* (*Proc. SIGGRAPH 95*), pp.291–300, ACM Press, New York (1995).
15) Ranta, M., Inui, M., Kimura, F. and Mäntylä, M.: Cut and Paste Based Modeling with Boundary Features, *Proc. 2nd Sympo. on Solid Modeling and Applications*, pp.303–312, ACM Press, New York (1993).
16) Sederberg, T.W. and Parry, S.R.: Free-Form Deformation of Solid Geometric Models, *Computer Graphics* (*Proc. SIGGRAPH 86*), pp.151–160, ACM Press, New York (1986).
17) 金井 崇，土岐健一，橋本善久，鈴木宏正，寺沢幹雄，木村文彦：制御線群にもとづく 3 次元形状変形とその応用，グラフィクスと CAD/Visual Computing 合同シンポジウム論文集，pp.9–16，画像電子学会 (1997).
18) 金井 崇，鈴木宏正：離散グラフの選択的詳細化に基づく多面体上の近似最短経路算出とその応用，情報処理学会研究報告，99-CG-97, pp.13–18 (1999).

(平成 11 年 8 月 31 日受付)
(平成 11 年 11 月 4 日採録)

金井 崇（正会員）

昭和 44 年生．平成 10 年東京大学大学院工学系研究科精密機械工学専攻博士課程修了．同年理化学研究所に入所．現在基礎科学特別研究員として同研究所素形材工学研究室に所属．形状モデリングの CAD/CAM および CG への応用に関する研究に従事．工学博士．精密工学会，ACM，IEEE CS 各会員．

鈴木 宏正（正会員）

昭和 32 年生．昭和 61 年東京大学大学院工学系研究科精密機械工学専攻博士課程修了・工学博士．昭和 62 年東京大学助手（教養学部），昭和 63 年同講師，平成 2 年同助教授，平成 6 年同大学院工学系研究科助教授（精密機械工学専攻）．現在に至る．形状モデリング，CAD/CAM システムの研究に従事．精密工学会，日本機械学会，日本図学会，ACM，IEEE 各会員．

三谷 純

昭和 50 年生．平成 10 年東京大学工学部精密機械工学科卒業．現在東京大学大学院工学系研究科情報工学専攻修士課程在学中．形状モデリングの CG への応用に関する研究に従事．

木村 文彦（正会員）

昭和 20 年生．昭和 49 年東京大学大学院工学系研究科航空工学専攻博士課程修了．同年通産省電子技術総合研究所パターン情報部研究員．昭和 54 年東京大学工学部精密機械工学科助教授，昭和 62 年同教授．現在に至る．形状モデリング，CAD/CAM，生産システム，等の研究に従事．工学博士．精密工学会，日本機械学会，ACM，IEEE 等会員．