

広域的法線評価と改良 Watershed 法に基づく メッシュモデルの特徴稜線抽出手法

溝口 知 広[†] 伊達 宏 昭[†]
金井 理[†] 岸 浪 建 史[†]

近年、リバースエンジニアリングや 3D データベースの検索など、様々なリソースからのメッシュモデルの入手が可能となり、工業製品開発における意匠設計段階でこのようなメッシュモデルを用いる機会が多くなった。しかし、工業製品開発の詳細設計のためには、最終的に設計対象のソリッドモデルを生成することが必要であり、製品開発期間短縮のため、メッシュモデルからソリッドモデルを自動で生成する技術が望まれている。そこで本研究では、メッシュモデルからのソリッドモデルの自動生成を支援するための、広域的法線評価と改良 Watershed 法に基づくメッシュモデルの特徴稜線抽出手法を提案する。ここで述べる特徴稜線とは、機能的に 1 つの面分として認識することが望ましい三角形集合からなる領域の境界線を意味する。本手法では、メッシュに含まれる測定ノイズやメッシュ分割方法の影響を受けにくい特徴稜線抽出が可能である。また、特徴稜線で囲まれた領域を 1 つの面分として認識でき、従来の特徴稜線抽出手法では抽出しにくかったフィレット面の境界線の特徴稜線として抽出することもできる。さらに、異なる抽出パラメータ設定より得られる特徴稜線群のうちユーザが要求する稜線のみを選択的に合成し、これらを 3 次元 CAD システムで読み込む機能も実現した。

Feature Edges Extraction on Mesh Models Based on Wide Range Normal Evaluation and Modified Watershed Method

TOMOHIRO MIZOGUCHI,[†] HIROAKI DATE,[†] SATOSHI KANAI[†]
and TAKESHI KISHINAMI[†]

Recently, mesh models are easily obtained from several resources such as reverse engineering and searching 3D database. These mesh models are often used at the style design stage in product development. But at the detail design stage, solid models are necessary, therefore the technique which automatically generates solid models from mesh models is required for more efficient product development. In this paper, we propose a new method of feature edges extraction of triangular mesh models based on wide range normal evaluation and segmentation using modified Watershed method for generating solid models from mesh models automatically. We define feature edges as boundary edges of the region consisting of a set of triangles which should be recognized as a face. Our method is robust for scanning noise and differences of mesh connectivity, and it allows us to recognize the region bounded by extracted feature edges as a face. Our method also enables the extraction of boundary edges of fillet surface as feature edges, which was difficult by previous methods. Moreover we implemented the functions for selecting and collecting required feature edges which are extracted at different parameter settings for extractions by user interactions, and importing resulting feature edges into 3D CAD system.

1. はじめに

近年、リバースエンジニアリングにおける 3 次元測定機の性能向上により、物理モデルからの高密度メッシュモデルの入手が可能となり、工業製品開発にお

ける意匠設計段階でこのようなメッシュモデルを用いる機会が多くなった¹⁾。また、3D データベース内のメッシュモデルが劇的に増加していることから、これらの検索から得られるメッシュモデルを再利用し、製品の意匠設計に用いることも可能であろう²⁾。しかし、工業製品開発の詳細設計のためには、最終的に設計対象のソリッドモデルを生成することが必要となるが、現状ではこのメッシュモデルからソリッドモデルを生成

[†] 北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology,
Hokkaido University

するために多くの時間を費やしている．そこで製品開発期間短縮のため、メッシュモデルからソリッドモデルを自動で生成する技術が望まれている．このようなメッシュモデルからのソリッドモデル生成に関する研究は、リバースエンジニアリングの分野でさかんに研究が行われているが、現状では複雑形状のソリッドモデルを自動で生成することは困難である^{3),4)}．

メッシュモデルからソリッドモデルを自動で生成するためのアプローチの1つに、まずメッシュモデルから特徴稜線を適切に抽出し、その特徴稜線を活かして曲面フィッティングを行うことが考えられる⁵⁾．ここで述べる特徴稜線とは、機能的に1つの面分として認識することが望ましい連結された多角形(三角形メッシュの場合は三角形)集合からなる領域の境界線を意味する．この観点から、特徴稜線抽出手法に対する要求条件として以下の4つがある．

- (a) ノイズやメッシュ分割方法にできるだけ影響を受けずに特徴稜線が抽出できること
- (b) 工業製品に多く含まれるフィレット面の境界線の特徴稜線として抽出できること
- (c) 抽出される特徴稜線がループをなし、これに囲まれた領域を1つの面分として認識できること
- (d) ユーザの利用目的に応じた特徴稜線が選択的に抽出できること

既存手法を調査した結果、これまでの手法には上記4つの要求条件すべてを満たす手法はない．そこで本研究では、上記の4つの要求条件をすべて満たすメッシュモデルからの特徴稜線抽出手法を提案することを目的とする．ただし、本研究では三角形メッシュモデルのみを対象とする．

2. 関連研究と提案手法の概要

メッシュモデルからの特徴稜線抽出手法には大きく2つの方法がある．1つは特徴稜線を直接抽出する手法であり、もう1つは領域分けの結果として得られる領域の境界線の特徴稜線として抽出する手法である．

2.1 特徴稜線を直接抽出する手法

渡辺ら⁶⁾は、三角形メッシュモデルの単純化の際に、形状特徴を表す稜線をできるだけ保存することを目的とした特徴稜線抽出手法を提案した．まず各稜線に対して、稜線接続性、隣接2平面間角度、位相距離で定められる部分領域内の湾曲度の3つを考慮して稜線特徴量を算出し、この値がユーザの指定した閾値以上の稜線をグルーピングし特徴稜線として抽出した．しかしこの手法では、位相距離により定められた領域内の稜線群を評価し稜線特徴量を算出するため、抽出され

る特徴稜線がメッシュの分割方法の影響を受けやすいといった欠点がある．さらに、閾値処理に基づき特徴稜線を抽出するため、特徴稜線がループを形成しないといった問題がある．

Meyerら⁷⁾の手法では、まずメッシュの二面角評価により処理の開始点を見つけ、次にその点から局所的な、不連続な形状をよく表現可能な *absoid* 曲面のフィッティングにより、本来の稜線の開始点を見つけ、その *absoid* 曲面から得られる特徴稜線を生成することを繰り返し行い、最後に抽出された特徴稜線の端点を接続する．しかしこの手法では、複雑な形状が少ない数の三角形で構成される場合に特徴稜線がループをなさないことがある．

Ohtakeら⁸⁾は、まず入力メッシュモデルから生成した近似陰関数曲面上での曲率を計算し、次にこれをもとに特徴稜線を検出し、最後に検出した特徴稜線をもとのメッシュモデルに投影することで、特徴稜線を抽出する手法を提案している．この手法では、メッシュから直接計算することは困難である高次の微分値を、入力メッシュから近似し生成した陰関数曲面上で計算し、これをもとにメッシュ上の山線・谷線を抽出している．しかし、この手法では、抽出される特徴稜線がループをなさず、ループを形成するには複雑な後処理が必要となるといった問題に加え、メッシュの向きにより異なる特徴稜線が抽出される、陰関数曲面上で高次の微分値を計算するため計算時間がかかるといった問題がある．

2.2 領域分けに基づく特徴稜線抽出手法

Manganら⁹⁾やSunら^{10),11)}は、リバースエンジニアリングにより得られるノイズを多く含んだ三角形メッシュモデルの領域分け手法を提案した．これらの手法では、まず各頂点の特徴量を算出し、次に算出した特徴量をもとに Watershed 法を用いて領域分けを行うことで、各領域の境界線の特徴稜線として抽出することができる．Watershed 法を用いると、抽出される特徴稜線は頂点特徴量に大きく依存し、変化してしまう．したがって、特徴稜線抽出において、メッシュに含まれる測定ノイズやメッシュ分割方法の影響を受けにくくするためには、Manganらのようにメッシュを局所的に評価するのではなく、Sunらのようにメッシュを広域的に評価し、頂点特徴量を算出する必要がある．しかし、Watershed 法では頂点集合からなる領域分けを行うため、隣接領域間どの領域にも属さない三角形集合からなる隙間領域が生じ、その結果として、本来隣接領域の境界線として共有されるべき特徴稜線が共有されないといった問題がある．

Pulla ら¹²⁾ は、特徴量を算出するために、まず入力メッシュの各頂点の近傍において近似ベジェ曲面を生成し、このベジェ曲面より各頂点の近似曲率を算出し、これを頂点特徴量とする手法を提案した。この手法では、近似ベジェ曲面を生成する範囲を広域的に設定すれば、入力メッシュに含まれるノイズの影響は低減されるが、シャープな稜線上の頂点における特徴量に大きな誤差を含む。また、算出される特徴量はその近似精度に依存し、近似精度を上げると計算時間がかかるといった問題がある。また Razdan ら¹³⁾ は、従来の Watershed 法では適切に領域分けされなかった、ソリッドモデルやパラメトリック曲面モデルのテセレーションから得られる三角形メッシュモデルなどに多く見られる、1つの領域として認識したい面内に頂点を含まない面(たとえば、長方形が2つの三角形から構成される場合など)を含むメッシュモデルを適切に領域分けできる、Watershed 法をベースにした手法を提案した。頂点特徴量算出には Pulla ら¹²⁾ の手法を用いている。この手法では、すべての三角形がいずれかの領域に属する領域分けが行えるため、隣接領域間に隙間領域が生じず、その結果として抽出される特徴稜線は一致するといった利点がある。

Vieira ら¹⁴⁾ は、入力メッシュの各頂点において近似双二次曲面を生成し、この近似曲面より各頂点の曲率を近似し、次にこの曲率に対してスムージングを適用することでノイズの影響を受けにくく特徴量を算出できる手法を提案した。さらに、この曲率を用いてシード領域を生成し、これを拡大していくことで、頂点集合からなる領域分けを行った。しかし、この手法では頂点集合からなる領域を生成するため、隣接領域間などの領域にも属さない三角形集合からなる隙間領域が生じ、その結果として、本来隣接領域の境界線として共有されるべき特徴稜線が共有されないといった問題があり、さらに、頂点数の少ない領域が生成され、これらの領域の境界線は微小な特徴稜線として抽出されるため、後の応用に利用しにくいといった問題がある。

Katz ら¹⁵⁾ は、過度な領域分けを回避でき、滑らかな境界線を抽出できる階層的な手法を提案している。この手法では、メッシュの双対グラフに対するカットを、最大流量アルゴリズムを用いて解くことにより領域分けを行っている。しかし、この手法では、工業製品の領域分けに必要な面分単位での領域分けが困難であり、処理時間がかかるといった問題がある。

これまでに述べたすべての既存手法に共通の問題として、いずれの手法もフィレット面の境界線の特徴稜線として適切に抽出できていないこと、また、1通り

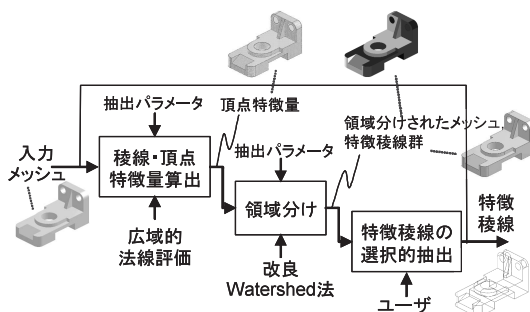


図1 アルゴリズムの概要

Fig. 1 An overview of our algorithm.

の抽出パラメータ設定(たとえば、特徴量を算出する際のメッシュの評価範囲を定めるパラメータ)では要求するすべての特徴稜線の抽出が困難であるといったことがあげられる。

2.3 本研究の概要

本研究ではこれまでに述べた問題を解決する手法を提案する。図1に示すように、本手法は大きく3つのステップからなる：

- (1) メッシュ上において、ユークリッド距離で定めた領域内の三角形集合の法線を広域的に評価することにより稜線や頂点の特徴量を算出する。さらに、頂点特徴量の離散化、および差分化を行うことにより、要求条件(a),(b)を満たす特徴稜線が抽出できる。
- (2) ステップ(1)で算出した特徴量に基づいて、頂点をグルーピング可能な Watershed 法²⁾を改良した手法を用いて連結三角形集合からなる領域分けを行い、この領域の境界線の特徴稜線として抽出する。これにより要求条件(c)を満たす特徴稜線が抽出できる。
- (3) ステップ(2)で抽出された特徴稜線のうち、ユーザの要求する特徴稜線のみを対話的に選択させる。これにより要求条件(d)を満たす特徴稜線が抽出できる。

さらに、これら3つのステップを、異なる抽出パラメータ設定で数回繰り返し実行し、各パラメータ設定で抽出される特徴稜線群を対話的に集約する。

3. 広域的な法線評価に基づく特徴量算出

本章では、本手法の第1ステップである、広域的な法線評価に基づく特徴量算出手法について述べる。

3.1 稜線特徴量算出

本手法では、まず稜線の特徴量を算出し、この稜線特徴量をもとに頂点特徴量を算出する。稜線特徴量算

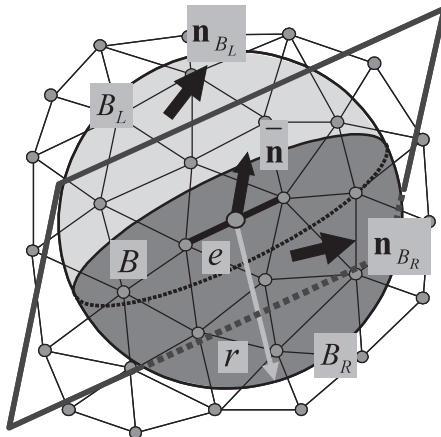


図 2 稜線特徴量算出
Fig. 2 Calculation of edge feature value.

出の最も単純な方法として、稜線を共有する 2 平面の二面角を計算する手法が考えられるが、この手法では、二面角の値が大きなシャープな特徴稜線の抽出には有効だが、リバースエンジニアリングから得られるノイズを多く含んだメッシュモデルに対しては、局所的な形状に基づいた特徴量算出しが行っていないため、算出される特徴量がノイズの影響を受けやすく、隣接 2 平面の二面角の値が小さな特徴稜線の抽出は難しい。

そこで本研究では、稜線の近傍のユークリッド距離で定められる広域的な領域を評価し、特徴量を算出することで、この問題を解決する。まず稜線 e の中点を中心とする半径 r の球 B を考える。次に、球 B に含まれる三角形の面法線ベクトルの和 \bar{n} と稜線 e とで定義した平面を用いて球 B を、2 つの半球 B_R, B_L に分割する (図 2)。次に B_R, B_L 内の各三角形面分の面積で重み付けした面法線ベクトルの和 $\mathbf{n}_{B_R}, \mathbf{n}_{B_L}$ を式 (1), (2) で計算する。

$$\mathbf{n}_{B_R} = \sum_{f \in F_{B_R}} \mathbf{n}_f |f \cap B_R| \quad (1)$$

$$\mathbf{n}_{B_L} = \sum_{f \in F_{B_L}} \mathbf{n}_f |f \cap B_L| \quad (2)$$

ここで、 \mathbf{n}_f は三角形面分 f の単位面法線ベクトル、 F_{B_R}, F_{B_L} は B_R, B_L に含まれる三角形の集合、 $|f \cap B_R|, |f \cap B_L|$ は f の B_R, B_L に含まれる部分の面積である。ただし、球や、球を分割した際に定義した平面と交差する三角形面分の球に含まれる部分の面積は、交差する部分を線分で置き換えた多角形で近似する。最後に $\mathbf{n}_{B_R}, \mathbf{n}_{B_L}$ のなす角度を式 (3) で算出し、これを稜線 e の特徴量 $\theta_e(r)$ とする。

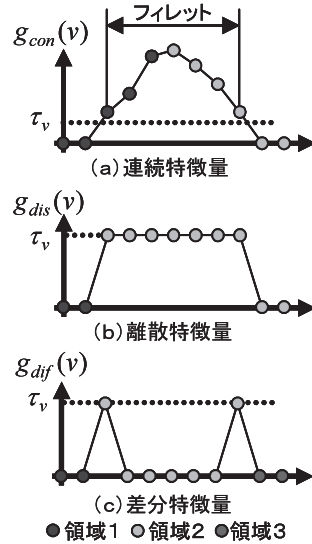


図 3 頂点特徴量算出
Fig. 3 Calculation of vertex feature value.

$$\theta_e(r) = \cos^{-1} \left(\frac{\mathbf{n}_{B_R} \cdot \mathbf{n}_{B_L}}{|\mathbf{n}_{B_R}| |\mathbf{n}_{B_L}|} \right) \quad (3)$$

3.2 基本特徴稜線・基本特徴頂点の分類

稜線特徴量 $\theta_e(r)$ の値が基本特徴稜線識別閾値 τ_θ より大きい稜線を基本特徴稜線、さらに基本特徴稜線の両端の頂点を基本特徴頂点と分類する。稜線の両端の頂点が基本特徴頂点であっても、必ずしもその稜線が基本特徴稜線とはならないことに注意する。これらは 4.2 節で述べる改良 Watershed 法で頂点を追加する際の参照データとして用いる。

3.3 頂点特徴量算出

次に、算出した稜線特徴量をもとに頂点特徴量を算出する。まず頂点 v に接続する稜線集合 $e^*(v)$ 内の稜線特徴量のうち最も大きな稜線特徴量を (この稜線特徴量が連続値をとることから) v の連続特徴量 $g_{con}(v)$ とする。

$$g_{con}(v) = \max_{e \in e^*(v)} \theta_e(r) \quad (4)$$

式 (4) で得られる連続特徴量では、図 3 (a) のように本来 1 つの領域として認識したいフィレット面の頂点特徴量にばらつきが生じる。フィレットの開始位置にある頂点が小さい値を持ち、中央に近いほど大きな値を持ち、終了位置では開始位置と同じ小さい値を持つ。これをもとに次章で述べる改良 Watershed 法を適用し領域分けを行うと、フィレット部が中央で 2 つに分割され、フィレットの中央線が特徴稜線として抽出されてしまう (図 4 (a))。そこで頂点特徴量離散化閾

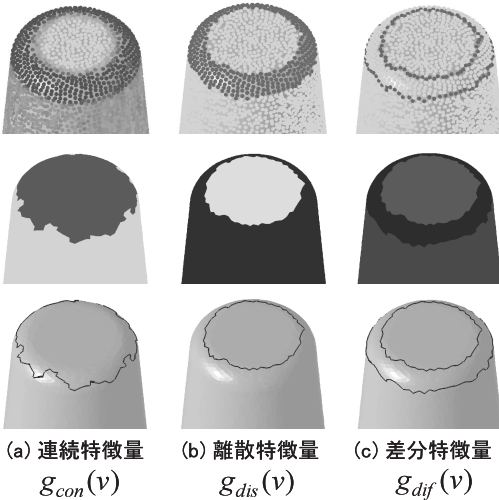


図 4 各特徴量に基づく特徴稜線抽出結果の比較 (上段: 頂点特徴量, 中段: 領域分け結果, 下段: 特徴稜線抽出結果)

Fig. 4 A comparison of feature edges extraction results using each feature value (upper: vertex feature value, middle: results of segmentation, bottom: results of feature edges extraction).

値 τ_v を用いてこの特徴量を離散化 (2 値化) したものを (この特徴量が離散値をとることから) 離散特徴量 $g_{dis}(v)$ とし, これによりフィレット面の頂点の特徴量を一定値にする (図 3 (b)). さらにこの離散特徴量では Watershed 法の性質上, 抽出したいフィレット部はそれに滑らかに接続する片側の領域と同じ領域と認識され, フィレット面の片側の境界線しか抽出できない (図 4 (b)). そこで, 頂点 v の 1 近傍頂点との離散特徴量の差分を式 (5) で計算し (図 3 (c)), これを差分特徴量 $g_{dif}(v)$ とする. これを用いることで, フィレット面の境界線が適切に抽出できる (図 4 (c)).

$$g_{dif}(v) = \begin{cases} g_{dis}(v) - \min_{v' \in v^*} g_{dis}(v') \\ (g_{dis}(v) \geq \min_{v' \in v^*} g_{dis}(v')) \\ 0 \quad otherwise \end{cases} \quad (5)$$

ここで v^* は v の隣接頂点集合である.

本手法では, シャープなメッシュ稜線のみを抽出したい場合には, 式 (4) の連続特徴量 $g_{con}(v)$ を頂点特徴量とし, 改良 Watershed 法による領域分けを行う. 一方, フィレット部の境界線の特徴稜線として抽出したい場合には, 連続特徴量を算出した後, 離散特徴量の算出を経て, 式 (5) の差分特徴量 $g_{dif}(v)$ を求め, これを頂点特徴量とし, 同様に, 改良 Watershed 法による領域分けを行う. 現在のところ, これらの抽出方法の使い分けは, 与えられたメッシュの形状の違いと

利用目的の違いによりユーザが判断し, 決定している.

4. 改良 Watershed 法

本章では, Razdan らの三角形集合からなる領域分けが行える Watershed 法¹³⁾ をベースにし, 抽出精度向上のための改良を加えた, 改良 Watershed 法について述べる.

4.1 Watershed 法

Watershed 法は, 画像の領域分け手法として提案された手法であり, 画像処理の分野で今でもさかんに研究が行われている¹⁶⁾. この Watershed 法は, 後に Mangan ら⁹⁾ により 3 次元メッシュモデルの領域分け手法に拡張された.

Mangan らの Watershed 法では, 水が高い所から低い所へ流れ落ちるように特徴量の大きい頂点から小さい頂点へとたどり, その過程でたどった頂点に共通の領域ラベルを割り当て, 同じラベルを持つ頂点を 1 つの領域にグルーピングする. Watershed 法の概要を図 5 に示す. ここでは (a) のモデルを図中の平面で切った断面図 (b) と, この断面図を展開し, 横軸に頂点のインデックス, 縦軸に先に算出した頂点連続特徴量を高さに見なした 2 次元のマップ (c) を作り, この図でアルゴリズムを説明する. (a) では頂点特徴量を色分けしており, 黒い頂点が高い特徴量, 白い頂点が高い特徴量を持つ. Watershed 法は次の 3 つのステップからなる.

ステップ 1) 局所最小頂点抽出

頂点特徴量が局所最小 (その近傍の頂点がそれ自身より大きな特徴量を持つ) となる頂点集合を見つけ, それぞれの頂点に 0 から始まる固有の領域ラベルを割り当てる (図 5 (d)).

ステップ 2) 降下

領域ラベルのない頂点をラベル付けされた領域 (頂点集合) に到達するまで降下させ, その過程でたどった頂点すべてにその領域ラベルを割り当て, 同じ領域ラベルを持つ頂点を 1 つの領域にグルーピングする (図 5 (e)).

ステップ 3) 領域統合

ステップ 2) で頂点集合からなる領域分けを行った際に, 少ない数の頂点集合からなる小さな領域が得意. これらの領域の境界線として後に抽出される特徴稜線は, その利用の観点からみて好ましくなく, 除去することが望ましい. そこでまず, 領域内の最も低い連続特徴量を持つ頂点と, その領域の境界とその領域に隣接する領域の境界の頂点のうちの最も低い連続特徴量を持つ頂点 v_{bou} との連続特徴量の差分を計

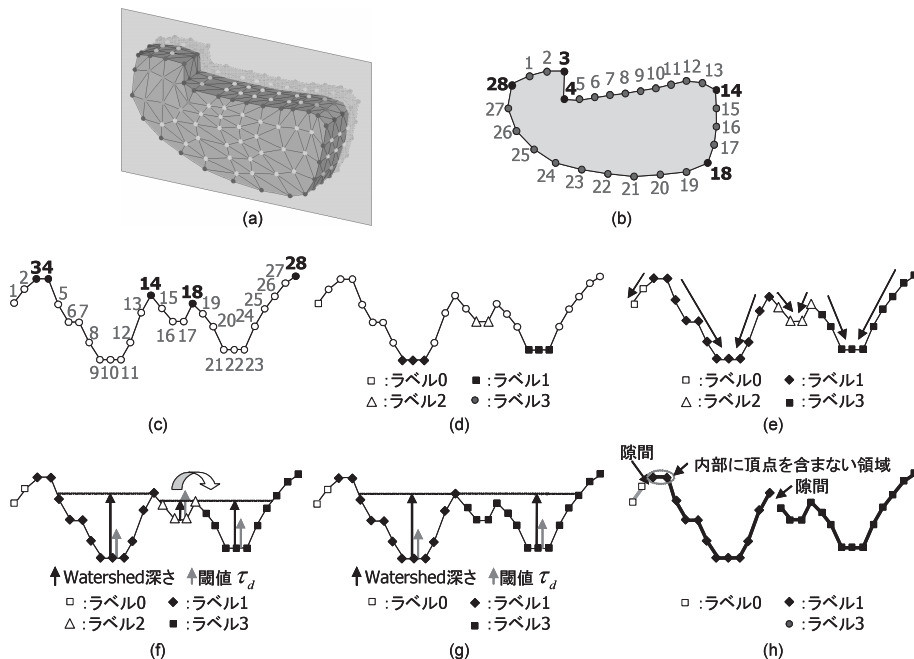


図 5 Watershed 法の概要
Fig. 5 An overview of Watershed method.

算する．この差分を Watershed 深さと呼ぶ．領域内の頂点数が領域内最小頂点数閾値 τ_n より少なく、かつ、Watershed 深さが領域統合閾値 τ_d より小さい領域を、 v_{bou} に隣接する領域と統合する．この処理を上記の条件を満たす領域がなくなるまで繰り返し、小さな領域を除去する (図 5 (f) , (g)) .

4.2 改良 Watershed 法

4.1 節で述べた問題点を解決するため、Razdan らの手法¹³⁾ に特徴稜線抽出精度向上のための改良を加えた改良 Watershed 法を提案する．この手法では、Watershed 法と三角形への領域ラベルの割当てに基づき、三角形集合からなる領域分けが可能である．改良 Watershed 法は 5 つのステップからなる．図 6 (a) は認識すべき面分の内部に頂点を含まないメッシュモデルの例である．このモデルは軸対象のモデルなので、ここでは、(a) のモデルを図中の白い線で切った断面図を展開し、横軸に頂点のインデックス、縦軸に先に算出した頂点連続特徴量を高さで見なした 2 次元のマップを作り、この図でアルゴリズムを説明する．この図では、線分はメッシュの三角形を表す．

ステップ 1) 頂点追加

1 つの領域として認識すべき面分集合内に頂点がない部分を適切に認識するため、3.2 節で述べた、メッシュ上の基本特徴稜線、基本特徴頂点の分布状況 (図 6 (b)) に基づき、図 7 に示す頂点追加規則に従って各辺の中

点に頂点を追加する (図 6 (c)) .

ステップ 2) 追加した頂点の特徴量算出

追加した頂点のうち、基本特徴稜線上にある頂点には全頂点のうち最大特徴量を、三角形の面上にある頂点には全頂点のうち最小特徴量を、それ以外の頂点にはその頂点が追加された稜線の特徴量を定義する (図 6 (d)) .

ステップ 3) Watershed 法

頂点を追加したメッシュモデルに Watershed 法を適用し各頂点に領域ラベルを割り当て、頂点集合からなる領域分けを行う (図 6 (e)) .

ステップ 4) 頂点の削除

基本特徴頂点にそれら自身の領域ラベルとともに 1 近傍頂点の領域ラベルも付加した後、ステップ 1) で追加した頂点をメッシュから削除する．これにより、基本特徴頂点は複数の領域ラベルを持つことになる (図 6 (f)) .

ステップ 5) 三角形への領域ラベル割当て

頂点ラベルの割当て状況に従って、図 8 に示すように各三角形に対して 1 つずつ領域ラベルを割り当てる (図 6 (g)) . 特に case2-case5 は、ソリッドモデルのテセレーションなどにより得られる、1 つの面分として認識したい領域内部に頂点を含まないために、適切に領域分けできなかったメッシュモデルを領域分けする際に有効である．一方 case6 は、リバースエンジニアリ

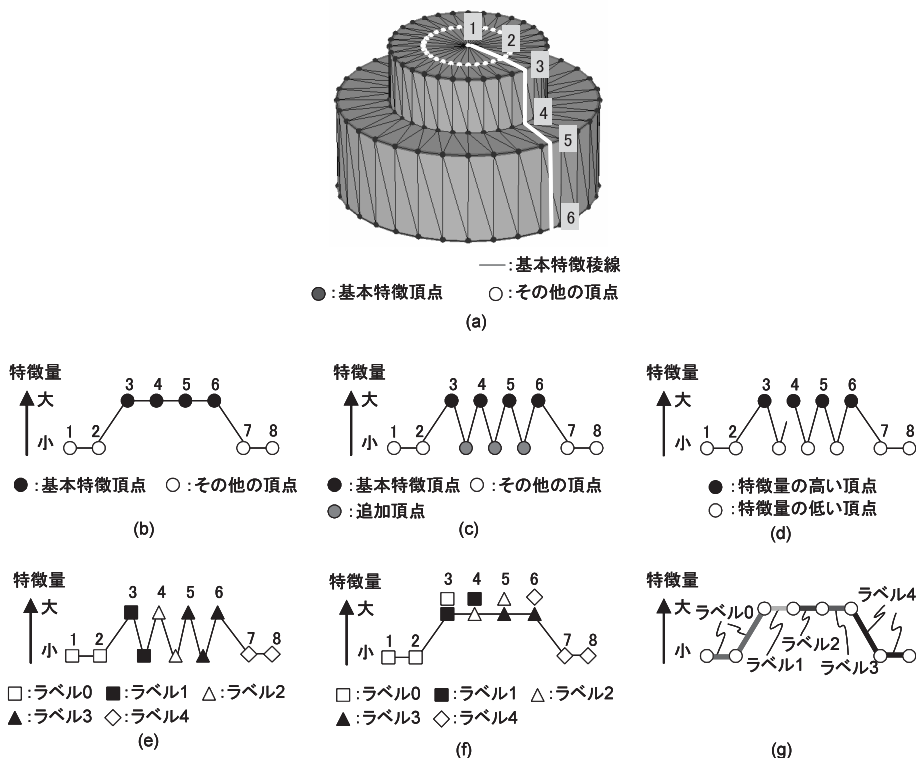


図 6 改良 Watershed 法の概要
Fig. 6 An overview of modified Watershed method.

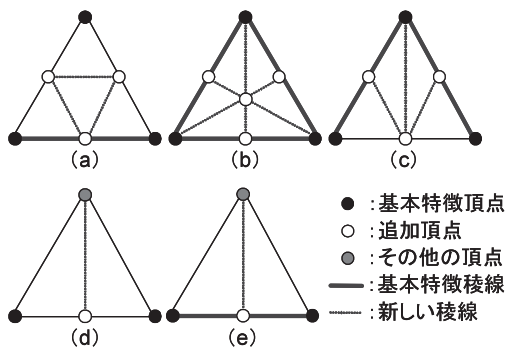


図 7 頂点追加規則
Fig. 7 Rules of vertex insertion.

| | 頂点ラベルの割り当て状況 | 三角形への割り当てラベル |
|-------|---------------------------------------|-----------------------|
| case1 | 3つとも単一の同じラベルを持つ | その頂点のラベル |
| case2 | 1つは単一ラベルを持ち、他2つは複数ラベルを持つ | 単一のラベルを持つ頂点のラベル |
| case3 | 3つとも複数ラベルを持ち共通ラベルが1つ | 共通ラベル |
| case4 | 3つの稜線が基本特徴稜線 | それ自身が領域 |
| case5 | 3つとも複数ラベルを持ち、共通ラベルが2つあり、基本特徴稜線が1つある場合 | 基本特徴稜線を共有する三角形と異なるラベル |
| case6 | その他 | 最も小さい特徴量の頂点のラベル |

図 8 三角形へのラベル割り当て規則
Fig. 8 Rules of allocating region label to a triangle.

ングから得られた高密度メッシュモデルに Watershed 法を適用した場合に生じる隙間領域の三角形に適切に領域ラベルを割り当てる際に有効である。これによりすべての三角形に1つずつ領域ラベルが割り当てられ、同じ領域ラベルを持つ三角形群を1つの面分としてグルーピングする。

以上の5つのステップにより、メッシュモデルに対し、連結三角形集合からなる領域分けが行え、最後に各領域の境界線の特徴稜線として抽出する。図9に従来の Watershed 法と、改良 Watershed 法を用いて領

域分けを行った結果を示す。従来の Watershed 法では、隣接領域間に隙間領域が生じ、その結果として、本来隣接領域の境界線として共有されるべき特徴稜線が共有されないといった問題(図9(a))、さらに、1つの面分として認識したい領域の内部に頂点がない場合に適切な領域分けができないといった問題を持つことが分かる(図9(b))。これに対し、改良 Watershed 法を用いることで、これらの問題が解決され、特徴稜

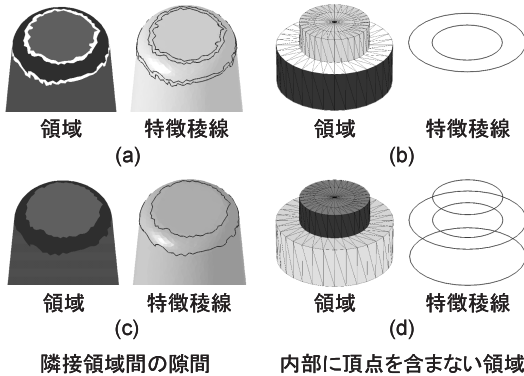


図 9 従来の Watershed 法 ((a), (b)) と改良 Watershed 法 ((c), (d)) による領域分けの比較

Fig. 9 An comparison of segmentation results using conventional Watershed method ((a), (b)) and modified Watershed method ((c), (d)).

線が共有され (図 9(c)), また, 適切な領域分けができていないことが確認できる (図 9(d)).

5. 特徴稜線の選択的抽出と集約

5.1 抽出パラメータの持つ定性的意味

本節では, まず, 本手法で用いる 5 つの抽出パラメータが特徴稜線抽出に与える効果と, 次に, 実際に特徴稜線を抽出する際の設定方法について述べる. パラメータ 1) 球の半径 r

これは稜線特徴量を算出する際の評価範囲を定めるために用いる. この値は厳密に設定する必要はなく, たとえば, 物理モデルを測定して得られるノイズを多く含んだメッシュモデルに対しては, 入力メッシュの Bounding Box の対角線の 50 分の 1 ~ 100 分の 1 程度に設定すると, 評価範囲が広域的になり, 形状がなだらかに変化する部分にある特徴稜線が抽出しやすくなる. また, ソリッドモデルの三角形分割から得られるメッシュモデルなどに多く見られる, 隣接 2 平面間の二面角が大きいクリースエッジを多く含んだメッシュに対しては, r を小さな値に設定すると, 評価範囲が局所的になり, シャープな特徴稜線が抽出しやすくなる.

パラメータ 2) 基本特徴稜線識別閾値 τ_0

この値は, ソリッドモデルの三角形分割から得られるメッシュモデルなどに多く見られる, 主にシャープな特徴稜線の抽出の際に用いる. 大きくする設定すると隣接面との二面角が大きい面の境界線のみが抽出でき, この値を小さく設定すると隣接面との二面角が大きい面の境界線だけでなく, 小さい面の境界線もが抽出できる. この値も特に厳密に設定する必要はない.

パラメータ 3) 頂点特徴量離散化閾値 τ_v

これはフィレット面の境界線の抽出位置を調整する際に用いる. この値は, フィレット面の境界線上にある頂点を持つ特徴量に一致するように, 適切に設定する必要がある. この適切値より小さく設定すると, 抽出したいフィレット面より広いフィレット面の境界線が抽出されてしまい, 大きくすると抽出したいフィレット面より狭いフィレット面の境界線が抽出される. 本研究では, この値をパー操作を用いて制御し, この閾値より連続特徴量の値が大きいかが, 小さいかを色分け表示している. これにより現状では, フィレット面の境界の決定は, 厳密な数値定義でなく, ユーザの目視による判断により行われる.

パラメータ 4) 領域統合閾値 τ_d

この値を小さく設定すると, 領域内の頂点特徴量の差が少ない領域分けが行われ, その境界線が抽出でき, 大きくすると, 領域内の頂点特徴量の差が大きい領域分けが行われ, その境界線のみ抽出できる. 仮に, この値をある値に設定し, 領域分けを行った結果, 小さな領域が多くできてしまった場合には, この値をさらに大きく設定することで, これらの小さな領域を隣接領域へ統合する. 逆に, 異なる領域として抽出したい 2 つ以上の領域が統合され 1 つの領域として抽出されてしまった場合には, この値を小さく設定しなおすことで, 要求する領域が抽出できる.

パラメータ 5) 領域内最小頂点数閾値 τ_n

この値を小さく設定すると, 少ない数の頂点集合からなる領域分けが行われ, その境界線が抽出でき, 大きく設定すると, 多い数の頂点集合からなる領域分けが行われ, その境界線のみ抽出できる. この値は厳密に設定する必要はなく, メッシュの前頂点数の 0.5 ~ 1% 程度の値に設定しておくとうよい.

実行時における上記抽出パラメータは, 以下の 1 ~ 3 のステップで設定する.

1. 上記抽出パラメータ 1), 2) を設定し頂点連続特徴量を算出する.
2. ユーザがフィレット面の境界線を抽出したい場合にはパラメータ 3) を設定し, 特徴量の離散化と差分計算を行い, 頂点差分特徴量を算出する. フィレット面の境界線以外の特徴稜線を抽出する場合には, このステップは省略する.
3. 領域分けのためのパラメータ 4), 5) を設定し, ステップ 1, 2 で算出した頂点特徴量を用いて領域分けを行い, その境界線の特徴稜線として抽出する. このステップの間でパラメータ 1) ~ 3) を変更することはない.

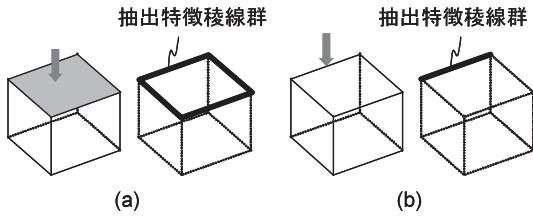


図 10 特徴稜線の選択的抽出
Fig. 10 Selective extraction of feature edges.

5.2 特徴稜線の選択的抽出と集約

3, 4 章で述べた手法を 1 通りのパラメータ設定で一度実行しただけでは, 要求する特徴稜線のすべてを抽出することは困難である. そこで, 以上の手法を異なる抽出パラメータ設定で繰り返し実行し, 各抽出パラメータ設定で抽出される特徴稜線のうちのユーザの要求する稜線をディスプレイ上で対話的に選択し, 集約することでこの問題を解決する.

本研究では, 効率の良いユーザの特徴稜線抽出のために, 2 つの特徴稜線選択手法を実装している. 第 1 は, 領域境界線ループの抽出である. これは, 領域内の面分の選択によって, 領域境界のループをなす特徴稜線群をすべて抽出できる (図 10 (a)). 第 2 は, 区分特徴稜線の抽出である. これは, 特徴稜線上のメッシュの稜線の選択によって, 3 つ以上の特徴稜線が交わる頂点で境界付けられた区分的な特徴稜線が抽出できる (図 10 (b)). これらを使い分けることで, あらゆる特徴稜線が選択的に集約でき, ユーザの要求する特徴稜線のみを抽出できる.

6. 実験結果と評価

図 11 に, ソリッドモデルの三角形分割により得られた円筒形状のメッシュモデルに, 同一サイズのフィレットをかけたモデルから, フィレット面の境界線の特徴稜線として抽出した結果を示す. このモデルには, 実際の測定器による測定誤差がガウス分布を持つことが一般的であることから, メッシュの平均稜線長さの 20% の値を標準偏差に持つ正規分布に従う乱数を発生させ, この量だけ頂点位置を頂点の法線方向に修正することで人為的にノイズを加えた. (a), (b) は, 頂点の特徴量を色分け表示したものであり, 頂点特徴量が高い頂点を黒で, 低い頂点を白で示している. (a) に示す局所的评价 (稜線特徴量にその稜線に隣接する 2 つの三角形の二面角を用いた) による頂点連続特徴量では, メッシュ全体に特徴量の高い頂点, 低い頂点が分布しており, これは頂点特徴量を加えたノイズの影響を受け, 特徴量を適切に評価できていないことを示

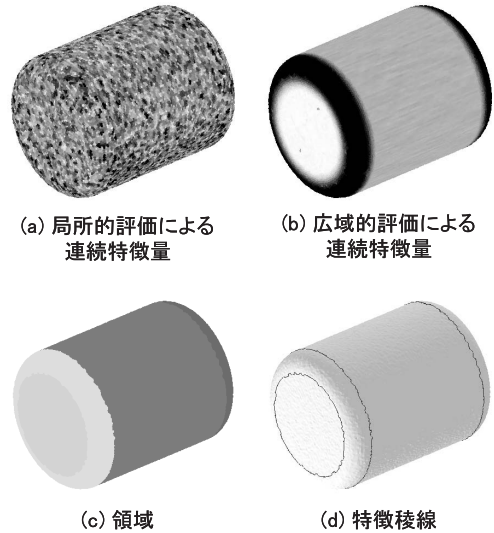
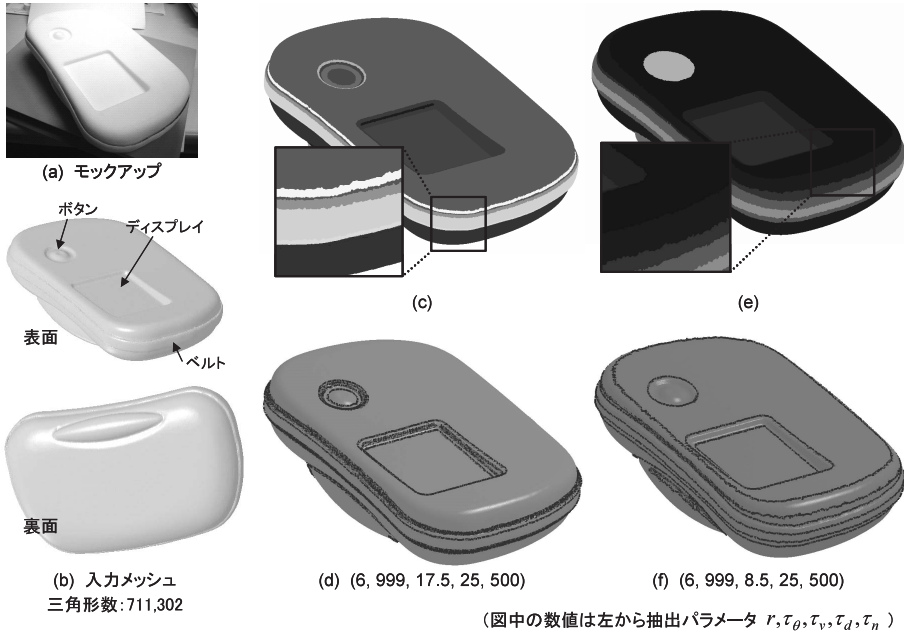


図 11 人為的なノイズを加えたメッシュモデルの特徴稜線抽出結果
Fig. 11 Results of feature edges extraction from a mesh model with artificial noise.

す. この特徴量をもとに頂点特徴量の離散化, 差分計算を行ってもフィレット面の境界線は抽出できないことが確認できた. これに対し, (b) に示す本研究で提案する広域的法線評価法による頂点連続特徴量では, 形状が湾曲しているフィレット面上の頂点の特徴量が他の頂点と比べて高い値を持ち, この特徴量をもとに頂点特徴量の離散化, 差分計算を行うことで, (c) のように領域分けが行え, さらに (d) に示されるように, その境界線をフィレット面の境界線として抽出できる. これより, 本手法で提案する頂点特徴量算出法が, 測定ノイズによる影響を受けにくいことが分かる.

図 12, 図 13 に, 硬質ウレタンフォームで作られた IT 機器デザインモックアップを, それぞれ非接触測定器 (Comet400¹⁸) で計測して得られた高密度点群から生成した高密度メッシュモデルに対する特徴稜線抽出結果を示す.

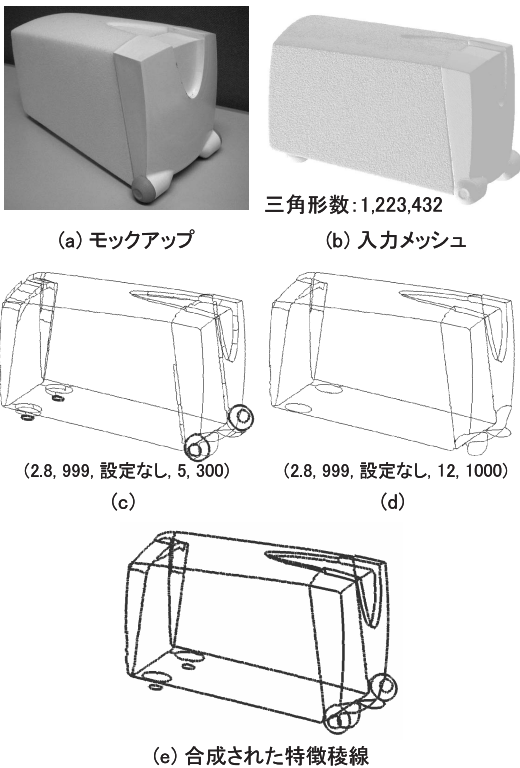
図 12 の例では, ユーザによるデザインモックアップの観察から, このモデルにはフィレット面が存在すると判断し, それらの境界線を抽出した. 同図 (c), (e) はそれぞれ図中に示される異なる抽出パラメータを用いて領域分けを行った結果である. また, (d), (f) はそれぞれ (c), (e) の領域の境界線の特徴稜線として抽出した結果である. このモデルには (b) で示すように, ボタン, ディスプレイ, ベルトの各境界部分に R の小さなフィレット面が存在する. (d) はこれらの境界線を抽出した結果である. さらに, モデルの表面に R の大きなフィレット面が存在する. (f) はこの境界



(図中の数値は左から抽出パラメータ $r, \tau_\theta, \tau_v, \tau_d, \tau_n$)

図 12 IT 機器筐体モックアップに対する特徴稜線抽出結果

Fig. 12 Results of feature edges extraction from a mock-up of housing of an IT device.



(図中の数値は左から抽出パラメータ $r, \tau_\theta, \tau_v, \tau_d, \tau_n$)

図 13 測定より得られたメッシュモデルに対する特徴稜線抽出結果
Fig. 13 Results of feature edges extraction from a scanned mesh model.

線を抽出した結果である。

一方、図 13 においては、ユーザによるデザインモックアップの観察から、このモデルにはフィレット面は存在しないと判断し、シャープなエッジで区切られる領域の境界線のみを抽出した。(c), (d) は、図に示された抽出パラメータ設定に従い特徴稜線を抽出した結果である。(c) では、抽出された特徴稜線のうち、ユーザにより選択された特徴稜線を太線で、選択されなかった特徴稜線を細線で示している。(e) は、(c) の選択された特徴稜線と、(d) のすべての特徴稜線を合成した結果である。

これらの結果より、本手法を用いることで、このような大規模測定メッシュからでも、測定ノイズの影響を受けにくく特徴稜線が抽出できることが確認できる。また、あるパラメータ設定では、ユーザの意図しない特徴稜線も抽出されるが、パラメータ設定を変更しながらインタラクティブに抽出された特徴稜線の集約処理を行うことにより、要求する特徴稜線を合成し、抽出できることも確認できる。抽出の処理時間は、Xeon 2.4GHz、メモリ 2GB の PC で、1 通りの抽出パラメータ設定での自動抽出処理に、図 12 のモデルでは約 3 分（頂点特徴量算出に約 2 分半、領域分けに基づく特徴稜線抽出に約 20 秒）、図 13 のモデルでは約 6 分（頂点特徴量算出に約 5 分半、領域分けに基づく特徴稜線抽出に約 20 秒）であった。

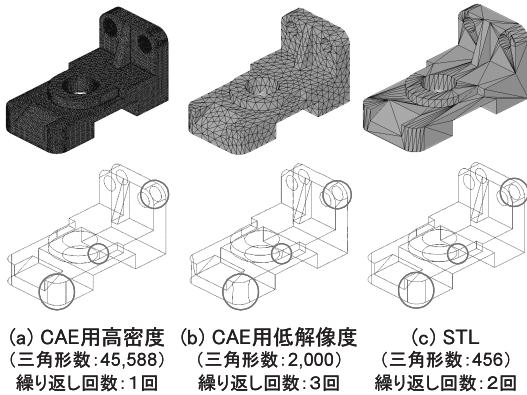


図 14 分割方法の異なるメッシュモデルに対する特徴稜線抽出結果
Fig. 14 Results of feature edges extraction from mesh models with different mesh connectivity.

図 14 に、ソリッドモデルを異なるメッシュ分割方法で分割し得られたメッシュモデルに対する特徴稜線抽出結果を示す。(a) は FEM メッシュにより分割して得られた高密度メッシュ、(b) は FEM メッシュにより分割して得られた低解像度メッシュ、(c) は STL データへの変換により得られたメッシュである。また、これらの結果は特徴稜線抽出手法を何度か繰り返して実行し、抽出された特徴稜線を集約した結果である。図 14 に示すように、分割方法の異なる 3 種類のメッシュモデルから、図の丸で囲まれた 5 カ所のフィレット面の境界線を含む、もとのソリッドモデル上に存在していたフィレット境界線のすべてを特徴稜線として抽出できることが確認できた。(a)、(b) に関しては、メッシュの稜線はソリッドモデルを生成した際に定義した稜線上に必ずしも存在していない。それゆえ若干の誤差があるものの、本来のソリッドモデル上の稜線に最も近いと思われる稜線が特徴稜線として抽出できている。

図 15 に、ユーザにより選択された特徴稜線の集約結果を示す。入力、ソリッドモデルを FEM メッシュにより分割して得られたひずみの小さい三角形からなるメッシュモデル (三角形数約 65,000) である。この図では、抽出された特徴稜線のうち、ユーザが選択した特徴稜線のみを表示している。(a)~(c) の 3 通りの抽出パラメータ設定での特徴稜線の集約結果が (d) である。(a) はシャープな稜線、(b) は比較的大きなフィレット半径を持つフィレット、(c) は比較的小さいフィレット半径を持つフィレットの境界線を抽出した結果である。これより、頂点特徴量算出アルゴリズムの変更と、異なる抽出パラメータ設定で抽出される特徴稜線が集約できることが分かる。処理時間は 1 通りの抽

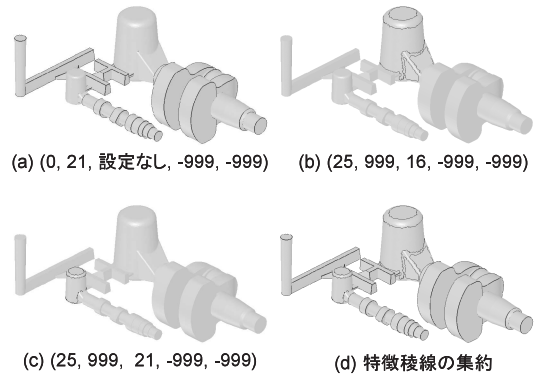


図 15 特徴稜線の集約結果
Fig. 15 Results of collection of feature edges.

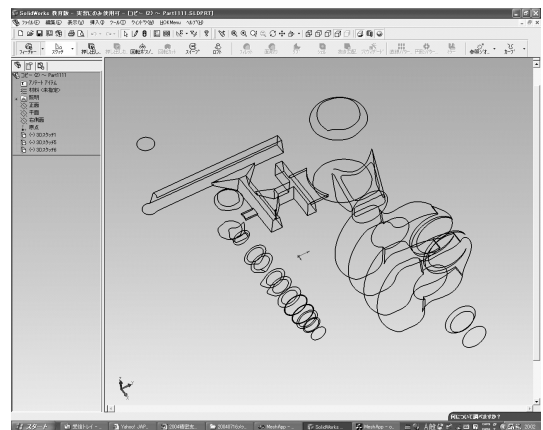


図 16 3次元 CAD システムで読み込んだ特徴稜線
Fig. 16 Imported feature edges in 3D CAD system.

出パラメータ設定での自動抽出処理で、約 4 秒~26 秒であった。

図 11~15 により、メッシュに含まれる測定ノイズの影響を受けにくく特徴量算出が行えること、抽出された特徴稜線がグループをなすこと、また、フィレット面の境界線が特徴稜線として抽出できることが確認できた。さらに、図 15 のようにして得られた特徴稜線を 3 次元 CAD システム (SolidWorks) 内に読み込んだ結果を図 16 に示す。これより、集約された特徴稜線が 3 次元 CAD システムに正しく読み込まれていることが確認できた。

7. 結論と今後の課題

本研究では、メッシュモデルからのソリッドモデルの自動生成を目的とした、広域的法線評価と改良 Watershed 法に基づく特徴稜線抽出手法を提案した。

シミュレーション, および, 実測定データによる検証より, 本手法を用いることで, メッシュに含まれるノイズやメッシュ分割方法の影響を受けにくく特徴稜線が抽出できること, またフィレット面の境界線の特徴稜線として抽出できること, 抽出された特徴稜線がループをなし, それに囲まれる領域を1つの面分として認識できること, 利用目的に応じた特徴稜線が選択的に抽出できることを確認した.

さらに本研究では, 抽出した特徴稜線の3次元CADシステムへの読み込みも実現した. これらの特徴稜線をワイヤフレームとして利用し, 既存のCADの機能(押し出し, スウィープなど)を用いて, ソリッドモデルを生成することも可能である. これらはソリッドモデルをゼロから作ることに比べて, 大幅な作業時間の短縮を可能にすると思われるが, これらに関しては, 今後詳細な考察とアルゴリズムの検討が必要である.

今後の課題としては, 抽出された特徴稜線の抽出精度評価, 処理時間の短縮などがあげられる. また, 本手法により抽出される特徴稜線を活かして, これに囲まれる領域内の頂点に対して解析曲面とNURBSをフィッティングすることから, ソリッドモデルの自動生成の実現が可能となる. ただし, 一部の領域では, 解析曲面単位でのフィッティングを考えると, さらに領域分けが必要になるものもあると思われる.

参 考 文 献

- 1) 鈴木宏正: 3D デジタルコンテンツのためのモデリング技術: 3次元メッシュモデルの生成と表現, 情報処理, Vol.41, No.10, pp.1103-1107 (2000).
- 2) Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S. and Dobkin, D.: Modeling by Example, *ACM Trans. Graphics*, Vol.23, No.3, pp.609-612 (2004).
- 3) Thompson, W.B., Owen, J.C., James, H., Stark, S.R. and Henderson, T.C.: Feature-Based Reverse Engineering of Mechanical Parts, *IEEE Trans. Robotics and Automation*, Vol.15, No.1, pp.1-10 (1999).
- 4) Várady, T., Martin, R.R. and Cox, J.: Reverse engineering of geometric models, *Computer-Aided Design*, Vol.29, No.4, pp.255-268 (1997).
- 5) Benkő, P., Martin, R.R. and Várady, T.: Algorithms for reverse engineering boundary representation models, *Computer-Aided Design*, Vol.33, No.11, pp.839-851 (2001).
- 6) 渡辺大地, 千代倉弘明: 任意三角形メッシュからの特徴稜線抽出, 電子情報通信学会論文誌 D-II, Vol.J83, No.5, pp.1344-1352 (2000).
- 7) Meyer, A. and Marin, P.: Segmentation of 3D triangulated data points using edges constructed with a C1 discontinuous surface fitting, *Computer-Aided Design*, Vol.36, No.13, pp.1327-1336 (2004).
- 8) Ohtake, Y., Belyaev, A. and Seidel, H.P.: Ridge-Valley Lines on Meshes via Implicit Surface Fitting, *ACM Trans. Graphics*, Vol.23, No.3, pp.609-612 (2004).
- 9) Mangan, A.P. and Whitaker, R.T.: Partitioning 3D Surface Meshes Using Watershed Segmentation, *IEEE Trans. Visualization and Computer Graphics*, Vol.5, No.4, pp.308-321 (1999).
- 10) Sun, Y., Page, D.L., Paik, J.K., Koschan, A. and Abidi, M.A.: Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing, *Proc. IEEE ICIP*, Vol.3, pp.825-828 (2002).
- 11) Sun, Y.: Surface Modeling and Analysis Using Range Images, Smoothing, Registration, Integration and Segmentation, *A Dissertation Presented for the Doctor of Philosophy Degree*, The University of Tennessee, Knoxville (2002).
- 12) Pulla, S., Razdan, A. and Farin, G.: Improved Curvature Estimation for Watershed Segmentation of 3-Dimensional Meshes, Arizona state University, Technical Report (2001).
- 13) Razdan, A. and Bae, M.S.: A hybrid approach to feature segmentation of triangle meshes, *Computer-Aided Design*, Vol.35, No.9, pp.783-789 (2003).
- 14) Vieira, M. and Shimada, K.: Segmentation of Noisy Laser-Scanner Generated Meshes with Piecewise Polynomial Approximations, *Proc. DETC/CIE* (2004).
- 15) Katz, S. and Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts, *ACM Trans. Graphics*, Vol.22, No.3, pp.954-961 (2003).
- 16) Bleau, A. and Leon, L.J.: Watershed-Based Segmentation and Region Merging, *Computer Vision and Image Understanding*, Vol.77, pp.317-370 (2000).
- 17) Roerdink, J.B.T.M. and Meijster, A.: Watershed Transform: Definitions, Algorithms and Parallelization Strategies, *Fundamenta Informaticae*, Vol.41, pp.87-228, IOS Press (2001).
- 18) <http://www.tbts.co.jp>

(平成 16 年 12 月 13 日受付)

(平成 17 年 10 月 11 日採録)



溝口 知広

2004年北海道大学工学部卒業。同年北海道大学大学院情報科学研究科システム情報科学専攻修士課程入学、現在に至る。形状モデリングに関する研究に従事。精密工学会会員。



伊達 宏昭

2003年北海道大学大学院工学研究科博士課程修了。同年同研究科COE研究員、2005年北海道大学大学院情報科学研究科学術研究員、現在に至る。博士(工学)。形状処理工学、メッシュモデリングに関する研究に従事。精密工学会会員。



金井 理

1987年北海道大学大学院工学研究科博士課程修了。1989年東京工業大学工学部助教授、1996年北海道大学大学院工学研究科助教授、2004年北海道大学大学院情報科学研究科助教授、現在に至る。博士(工学)。デジタル幾何処理に関する研究に従事。精密工学会、ACM、IEEE各会員。精密工学会「デジタルスタイルデザイン研究分科会」幹事。



岸浪 建史

1971年北海道大学大学院工学研究科博士課程修了。1972年北海道大学工学部助教授を経て1988年北海道大学工学部教授、2004年より北海道大学大学院情報科学研究科教授、理事・副学長兼務。博士(工学)。生産情報工学、情報モデル、生産システムに関する研究に従事。日本機械学会、精密工学会、型技術協会各会員。1993年型技術協会技術賞、1996年日本機械学会FA部門功績賞、1999年精密工学会賞受賞。