

ワイヤースケルトンモデルからの曲面モデルの構成法

井上 恵介[†] 嶋田 憲司^{††}

立体形状を意図して作られたワイヤースケルトンモデルを解釈して、表面に対応する曲面モデルを効率よく生成する方法を提案する。対象とする入力ワイヤースケルトンモデルは2連結な平面マルチグラフであり、球面と同相な閉曲面として複数の解釈が可能である。本手法では、まずグラフの3連結要素分解を用いて位相的に可能な解をすべて生成する。その後で、それぞれの解を構成する面ループに対してできるだけ単純な曲面を張り、得られた解（面分の集合）を幾何的に評価・比較することによって、全体として単純な形状を表す曲面モデルを出力する。同様の方法で、円板と同相な開いた曲面としての解釈も可能である。過去に作られたワイヤースケルトンモデルの再利用だけでなく、複数の表現形式を並行して管理するCADシステムの構築にも応用できる。

Surface Model Reconstruction from Wireframe CAD Models

KEISUKE INOUE [†] and KENJI SHIMADA^{††}

We propose an efficient method for reconstructing a surface model from a wireframe model. As input the method accepts a biconnected planar multigraph, which can be interpreted in multiple ways as a closed surface that is homeomorphic to a sphere. In this method we generate all topologically possible interpretations (i.e., all sets of face loops) of the input wireframe model by applying a triconnected component decomposition. Then we create the simplest possible surface geometry for each face loop in the interpretations and compare the solutions geometrically to output surface models representing simpler shapes. Similarly, we generate interpretations representing an open surface that is homeomorphic to a disc. Our method can be applied to construct a CAD system that supports multiple representations of an object as well as to extend the capabilities of existing CAD databases of wireframe models.

1. はじめに

機械部品を典型とする3次元幾何形状のモデリングにおいて、ワイヤースケルトンモデル、曲面モデル、立体モデルがよく使われている。上位の、すなわち後者のモデルほどシステムの構築や形状の操作は複雑になるが、形状記述手段としての完全性は高くなる。そのため、有限要素解析や加工工程設計など多くの用途への応用性が高く、CAD(Computer-Aided Design)の発展もこの流れに沿って進んできた。

しかし、立体モデルが下位のモデルの情報を包含するからと言って、それだけですべての要求を満たす訳ではない。とりわけ試行錯誤を伴うような設計過程においては、1つの設計対象について視点の異なった複数のモデル（例えばワイヤースケルトンモデルと立体モデル）を使うことで、より効率的な設計環境が実現で

きる。その際、モデルどうしの整合性は、できる限り自動的に維持されることが前提である。

特にワイヤースケルトンモデルには以下のような利点があり、他のモデルと並用することでその有用性がより高まると考えられる。

- 立体は3次元、曲面は2次元なのに対し、ワイヤースケルトンモデルは1次元である。つまり情報量が少なく、設計者の思考上も作業上も、形状操作が簡単である。
- 曲面式など詳細の決定を後回しにして、特徴曲線（稜線）によって形を決めてゆく設計が可能である。
- 計算機内に保持する情報が少なく、メモリやCPUへの負荷が軽い。

ワイヤースケルトンモデルを曲面または立体モデルに自動で変換する技術は、こうした利点を生かすために不可欠なものである。

設計者は多くの場合、ワイヤースケルトンモデルを曲面または立体形状を抽象化したものとして見ている。すなわち、実際に扱っているのは曲線でも頭の中には曲面や体積のイメージを持っている。しかし、ワイヤースケルトン

[†] 日本アイ・ビー・エム(株) 東京基礎研究所
Tokyo Research Laboratory, IBM Japan, Ltd.

^{††} カーネギーメロン大学 機械工学科
Mechanical Engineering, Carnegie Mellon University

フレームを曲面や立体として解釈することはそこに存在しない情報を補うことであり、いつも唯一の答があるわけではない。対象についての背景的知識を共有している設計者間でさえ、ワイヤーフレームのどこに面分があるかについて異なった解釈をすることがある。

このような難しさはあるが、ワイヤーフレームから曲面モデルを構築する技術の応用分野は広く、以下のようなものが挙げられる。

- CAD システムにおいて、曲面・立体形状を容易に入力し操作するためのしぐみを提供する。
- ワイヤフレームの形式で蓄積された膨大な設計資産を、変換することで再利用可能にする。
- 画像からの物体認識において、ワイヤーフレームに面を張ることでより正確なマッチングを行なう。

2. ワイヤフレームからの曲面の構成問題

2.1 問題の入出力

入力となるワイヤーフレームモデルは、空間中の曲線セグメント（以後、稜線と言う）の集合であり、以下の情報を持つものとする。

- 幾何情報: 多項式曲線であれば係数とパラメータの範囲、線分列であれば点列の座標値。
- 位相情報: どの稜線とどの稜線が端点を共有するか。但し、稜線は端点以外では他の稜線と交わらないとする。

位相情報については CAD 側では陽に管理されていないことも多いが、その場合は適当な許容値を設定して後から構成する。

曲面モデルは面分の集合である。各面分は一続きの稜線で囲まれた領域で、内部の幾何情報（曲面式など）を持っている。従って、ワイヤーフレームモデルに以下の 2 つの処理を行えば、曲面モデルが得られる。

- 位相の構築: 具体的にどの稜線の列を面分のループとみなすかを定める。
- 幾何の構築: それぞれの面分のループをどのような曲面で満たすかを定める。

さらに立体モデルを得るには、面分の全体が閉曲面をなし、かつ空間を二分しているときに、体積が有限の側を選べばよい。

ワイヤーフレームからの曲面の構成問題の解は、幾何情報が定義された面分の集合である。解が複数ある場合は、予め定義した基準によってそれらを評価し、不合理なものを排除したうえで、評価の高い順に出力する。

2.2 位相の構築と多義性

曲面モデルの位相を構築する際、ワイヤーフレー

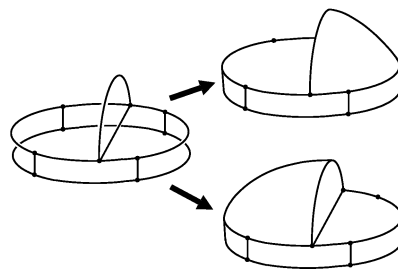


図 1 位相的な多義性の例

Fig. 1 Examples of multiple interpretations

ムの中でのループのとり方は非常に多く考えられる。従って、ワイヤーフレームを面分の組合せとして解釈する方法にも、膨大な可能性がある。そこで、以下に挙げるような制約によってその可能性を限定することが有効である。但し大前提として、解の中ですべての稜線が最低一度は利用されるものとする。

- (1) 面分の組合せが全体として閉じた曲面を表わす。
- (2) 同じ稜線を 3 枚以上の面分が共有することはない (2 次元多様体の条件)。
- (3) 面分どうしは稜線以外では交わらない。
- (4) 面分として、限定した種類の曲面 (平面や 2 次元曲面など) になるものだけを許す。

このうち純粋に位相的な制約は (1) と (2) である。(1) はすべての稜線が 2 つの面分に属していることを意味しており、「クラインの壺」のような場合を除き立体モデルへの変換が可能である。一方 (1) を満たす面分の組合せからいくつかの面分を外しても (2) は満たされており、1 枚の紙を切り抜いて折り曲げたような形状を表わすことができる。(3) の条件は幾何的でもあり位相的でもある。つまり、面分が稜線以外で交差するかは内部の曲面に依存するが、どんな曲面を張っても必ずどこかで交差が生じるような面ループの組合せもある。(4) は幾何的な制約であり、解が大きく限定されるので、過去の研究でも多く採用されている。

全体が閉曲面をなす場合に限っても、面ループの組合せとしての解釈は 1 つとは限らない (図 1)。このようにとき、複数の解は幾何的には同等でなくても、位相的には優劣はなく同格である。多義性の原因となるような曖昧な箇所が増えると、解釈の総数は組合せ的に増大する。

2.3 幾何の構築

曲面モデルの幾何の構築については、各面ループに対して完全に独立な方法と、隣接する面分との関係 (接平面連続性など) を考慮する方法がある。後者の場合は、境界曲線上の曲面法線など付加的な情報を与える必要があり、処理は複雑になる。

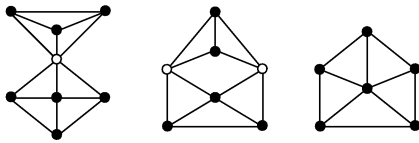


図2 n 連結なグラフの例 (左から $n=1, 2, 3$)
Fig.2 n -connected graphs (from left, $n=1, 2, 3$)

いずれにせよ境界曲線以外の情報はもともと存在しないので、曲面は無数に考えられる。実際には何らかの方針に基づき、面ループから一意に曲面を生成する。

3. 従来の手法

3.1 数学的準備

本節では、ワイヤフレームの位相に関する議論に必要な用語について、簡単に説明する。

頂点 (vertex) の集合 V と辺 (edge) の集合 E によって定義されるグラフを $G=(V, E)$ とする。辺は2つの頂点の組 (v, w) で表わされる。両端点が同一の辺を、ループ辺 (loop edge) という。辺の集合 E に、同一の頂点の組を結んだ辺が複数含まれることを許し、ループ辺が含まれることは許さないグラフを、マルチグラフ (multigraph) という。頂点を2つだけ含むマルチグラフをボンド (bond) といい、特に n 本の辺を含むボンドを n -ボンド (n -bond) という。

グラフ上の経路 (path) とは、頂点と辺が交互に現われ、かつ最初と最後が頂点である列をいう。マルチグラフにおいて任意の頂点の組に対して経路が存在するとき、そのグラフは連結 (connected) である。同様に、グラフから任意の $(n-1)$ 個の頂点とそれに接続する辺を外しても残ったグラフが連結であるとき、 n 連結 (n -connected) であるという (図2)。経路の最初と最後の頂点が同一であり、かつそれ以外の頂点がすべて異なるとき、その経路を閉路 (cycle) という。閉路を含まない連結なグラフを木 (tree) という。

辺どうしが共通の頂点以外で交わらないようにグラフ全体を曲面上に書くやり方を、グラフの埋め込み (embedding) という。埋め込んだ際に辺で分離される曲面領域を面分 (face) という。平面上に埋め込むことが可能なグラフを、平面グラフ (planar graph) という。平面グラフは球面に埋め込むことができ、その逆も成り立つ。平面への埋め込みでは外側に無限の大きさの面分ができるが、球面への埋め込みではすべてが有限の面分になるので、以後、球面への埋め込みによって説明を行なう。

3 連結な平面グラフについて、埋め込み (すなわち面分の組合せ) が1つだけ存在することが、グラフ理

論から知られている¹⁾。逆に、2 連結でかつ3 連結でない平面グラフには複数の埋め込みが存在する。

グラフの閉曲面上への埋め込みを考えると、その曲面は位相的には「境界のない2次元多様体」である。境界のない2次元多様体は、向き付け可能 (orientable) なものと、不可能なもの (例えばクラインの壺) にさらに分けられるが、立体の表面になり得るのは前者だけである。

向き付け可能な2次元多様体は、種数 (genus) によって分類できる。種数は球面で0、トーラス面で1、以後ハンドルの数が増すにつれて増えてゆく。

あるグラフが埋め込める閉曲面の種数の最小値を、そのグラフの種数という。平面グラフの種数は0であり、非平面グラフの種数は1以上である。

3.2 関連研究

ワイヤフレームモデルから立体モデルすなわち閉じた曲面モデルを構築する問題は、1970年代から研究されてきた。

初期の研究では、全体として閉曲面をなす面分の集合の位相を構築するために、ワイヤフレームの幾何情報が重要な役目を果たしていた。Markowsky²⁾ は、入力ワイヤフレームへの位相的な制約は小さいものの、対象形状が平面だけで構成された多面体であるという制約を採用して、面ループを決定した。桃井³⁾ の手法も、複数の解を扱うという特徴はあるが、対象とする形状について同様の制約があった。これらのような幾何情報に基づいて曲面モデルの位相を決定する手法には、以下の問題がある。

- 対象となる形状が限定される。自由曲面を含んでいたりと、そもそも稜線以外の幾何をまだ考えていないような初期設計段階のワイヤフレームは扱えない。
- 位相的な判定 (面ループの発見) が入力モデルの幾何に依存しているため、幾何的な誤差の悪影響を受ける。

これらの問題を避けるため、ワイヤの接続関係だけから面ループを決定する位相的な方法が研究された。Hanrahan⁴⁾ や Dutton⁵⁾ は、この問題を閉曲面へのグラフの埋め込みの問題として扱った。グラフ理論の研究成果 (グラフの平面性検査⁶⁾ など) を利用して埋め込みを決定し、幾何情報を使わずに面ループの組合せを得た。しかし、彼らが課した3 連結グラフの条件は、埋め込みが一意に決まる利点はあるものの、本論文で採用した2 連結グラフの条件に比べると入力を強く限定してしまう。

Ganter⁷⁾ は、閉曲面への埋め込み問題として考え

るのではなく、グラフの独立な閉路の集合をまず決定した。それらの閉路どうしを繰り返し結合することにより、個々の閉路の長さを短くし、全体の辺の共有を削減する。しかし、この発見的ルールに基づく方法では立体を囲む閉曲面が導ける保証はなく、行き詰まりを解消する対話的な操作が必要であった。Courter⁸⁾はこの手法を改良して対話的操作をなくしたが、複数の解釈があり得る入力(2連結なグラフ)に対しては、最初に見つかった解を返すだけであった。

Agarwal⁹⁾は、入力ワイヤーフレームを位相的な単体(4面体)の集合に逐次分割する手法を提案した。しかし、単体に分割するには立体としての解釈が結局必要であり、分割に幾何情報を利用しないのであれば、各頂点ですべての可能な辺の並び順を考えなくてはならない。それを行なって埋め込みを数え上げる体系的な方法は示されなかった。

4. アルゴリズムの概要

4.1 入力ワイヤーフレームの範囲

入力されたワイヤーフレームモデルは、位相情報だけに注目すればグラフである。入力グラフとして、本論文では2連結な平面マルチグラフを対象とする。

多くの従来手法で対象とされてきた3連結なグラフは、集合としては2連結なグラフの部分集合であり、2連結なグラフのほうが格段に範囲は広い。但し、これにより球面(または平面)への埋め込みの一意性が失われる。つまり位相的な根拠だけから解を1つに決めることはできなくなり、位相的には同等な複数の解を考慮することが必要になる。

また、一般に任意のグラフを2連結なグラフに分解することができる。つまり2連結なグラフを扱うことは、将来一般のグラフを処理する際の基礎となる。

グラフの平面性については、平面グラフ(種数が0)だけを対象にする。つまり、図3のようなトーラス面上にしか埋め込めないグラフ(種数が1)は扱えない。本論文の手法で種数が1以上の場合を除いたのは、種数が1以上のグラフ特有の以下の事情による。

- 3連結の場合でも、最小種数の閉曲面への埋め込みが一般に複数ある。
- 最小種数の閉曲面への(すべての)埋め込みを生成する、効率のよいアルゴリズムが知られていない。

4.2 処理の流れ

本論文の手法は以下の3つのステップで構成される。

- (1) (埋め込みの生成) グラフの球面上へのすべての埋め込みを、位相的な考慮によって生成する。

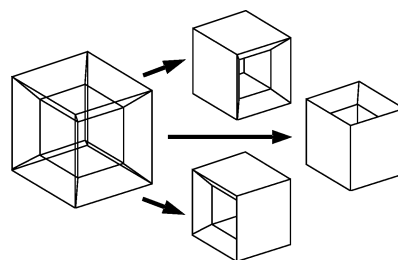


図3 トーラス面になら埋め込み可能なワイヤーフレーム
Fig. 3 Wireframe embeddable only on the torus

- (2) (曲面幾何形状の生成) 生成されたすべての面ループに、なるべく単純な曲面を張る。
- (3) (解の絞り込み) すべての埋め込みを、前のステップで生成した曲面情報も考慮して幾何的に評価する。解として不合理なものを排除し、残った解を妥当性に基いて順位づけする。

ステップ(1)では、幾何情報をまったく利用しない。ステップ(2)と(3)では幾何情報を利用するが、利用のしかたを限定することで、わずかな幾何的誤差が最終的な解に大きな影響を及ぼすのを最小限に抑えることができる。

各ステップは理論上は完全に独立だが、実装上は一部並行して処理を進めることで、結果に悪影響を与えず効率を向上させることができる。

また、全体にわたる方針として、ワイヤーフレームをなるべく単純に解釈するものとする。種数が0の2次元多様体(すなわち球面)への埋め込みだけを考慮するのも、そのためである。さらにステップ(2)と(3)でも単純な解釈を優先する方針を採用する。但し「単純さ」の定義のしかたは、対象形状の分野(種類)に依存している。

この後の5-7章では、ステップ(1)-(3)の処理についてそれぞれ説明する。

5. 埋め込みの生成

埋め込み生成の方針は、まず入力グラフを埋め込みが明らかな要素グラフに分解する。次に、要素グラフを埋め込んだ位相的な球面を逐次貼り合わせることで、すべての埋め込みを組合せ的に生成する。

5.1 3連結要素分解

本節では、2連結グラフの「3連結要素分解(triconnected component decomposition)」について説明する。詳細については原論文(10)を参照されたい。

この手法によって、2連結なマルチグラフは3種類の要素グラフに分解される。処理の流れは、グラフをまずできるだけ細かく分割し(第1段階)、次にそれ

を局所的に結合する (第 2 段階).

5.1.1 第 1 段階 (グラフの分割)

入力グラフを G とし, G 中の頂点の組を $\{a, b\}$ とする. このとき, G の辺を同値類 E_1, E_2, \dots, E_n に分け, 各同値類に含まれる 2 本の辺が a, b のどちらも端点以外には含まない共通の経路上にあるようにできる (図 4). 同値類 E_i を, $\{a, b\}$ についての G の分割クラス (separation class) と呼ぶ. 2 つ以上の分割クラスがあり, かつ以下の例外条件をいずれも満たさないとき, $\{a, b\}$ を G の分割ペア (separation pair) という.

- ちょうど 2 つの分割クラスがあり, 一方が辺 1 本だけのとき.
- ちょうど 3 つの分割クラスがあり, 各クラスがそれぞれ辺 1 本だけのとき.

G が 2 連結でかつ G の中に分割ペアが存在しないとき, G は 3 連結である.

$\{a, b\}$ についての G の分割クラスを E_1, E_2, \dots, E_n とする. ここで, $|E'| \geq 2, |E''| \geq 2$ を満たすように以下の式で E' と E'' を定義する.

$$E' = \bigcup_{i=1}^k E_i, \quad E'' = \bigcup_{i=k+1}^n E_i$$

これを使って G_1 と G_2 を下式で定義し, これを $\{a, b\}$ についての G の分割グラフ (separation graph) と呼ぶ.

$$G_1 = (V(E'), E' \cup \{(a, b)\}),$$

$$G_2 = (V(E''), E'' \cup \{(a, b)\})$$

但し, $G = (V, E)$ は頂点の集合 V と辺の集合 E によって定義されるグラフを表わし, $V(E)$ は E のすべての辺の端点からなる頂点の集合である.

分割の際, 両方のグラフにそれぞれ新しい辺 (a, b) を追加した. これを仮想辺 (virtual edge) といい, (a, b, i) (i は分割操作の識別子) と記述する.

3 連結要素分解の第 1 段階では, 入力グラフに対し, これ以上分割できなくなるまで再帰的に分割操作を適用する. 最終状態での分割グラフを分割要素 (split component) と呼ぶ. 図 5 のように, 分割要素は 3-ボンドグラフ, 3 角形グラフ, 3 連結なグラフの 3 種類である (3 連結なグラフについては一例を示した). 分割要素への分解は, 入力グラフに対して一意ではない.

5.1.2 第 2 段階 (局所的な結合)

つづく第 2 段階では, 入力に対して一意的な分解を得るために, 分割要素を局所的に結合する. $G_1 = (V_1, E_1)$ と $G_2 = (V_2, E_2)$ を, どちらも同一の仮想辺 (a, b, i) を含む分割グラフとする. このとき

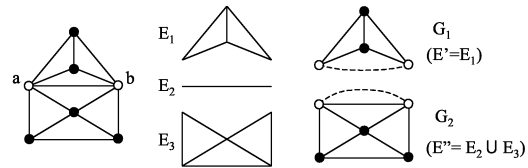


図 4 同値類への分割と分割グラフ G_1, G_2
Fig. 4 Equivalence classes and split graphs G_1, G_2

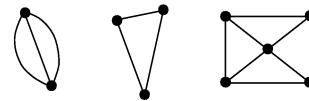


図 5 3 種類の分割要素
Fig. 5 Three types of split components

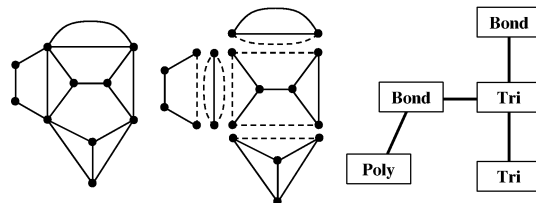


図 6 グラフとその 3 連結要素分解および構造グラフ (左から順に)
Fig. 6 Input graph, its triconnected component decomposition and the structure graph (from left to right)

$G = (V_1 \cup V_2, (E_1 - \{(a, b, i)\}) \cup (E_2 - \{(a, b, i)\}))$ と定義し, G を G_1 と G_2 の結合グラフ (merge graph) という. 分割要素の集合から始めて, ボンドの組に対してこの結合操作をできるだけ適用すると, いくつかの n -ボンドが得られる. 3 角形グラフの組に対しても同様にすると, いくつかの多角形グラフが得られる.

その結果, n -ボンドグラフ, 多角形グラフ, 3 連結なグラフ, の 3 種類の要素グラフの集まりを得る (図 6). これを G の 3 連結要素 (triconnected component of G) という. 3 連結要素への分解は, 入力グラフに対して一意である.

5.2 構造グラフ

本節では, すべての埋め込みを生成するため, 3 連結要素分解の結果を利用して新たなグラフを定義する.

3 連結要素分解の後, 3 連結要素を頂点, 1 組の仮想辺を両側の頂点 (3 連結要素に相当) をつなぐ辺とみなして, 新しいグラフを作る (図 6). これを構造グラフ (structure graph) と呼ぶ.

構造グラフに関して以下の補助定理が成り立つ.

補助定理 1 構造グラフは木である.

(証明は付録を参照)

5.3 各 3 連結要素の埋め込みの数え上げ

一般の 2 連結平面マルチグラフの埋め込み生成につ

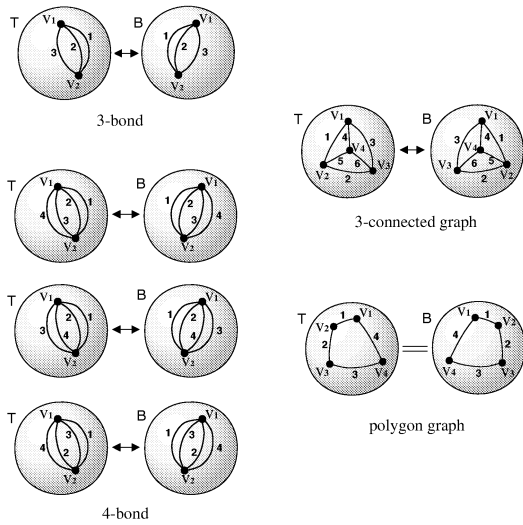


図 7 3 連結要素の球面上への埋め込みの例 (T と B は表と裏)
Fig.7 Triconnected components embedded on the sphere (T:top, B:bottom)

いて議論する前に、各タイプの 3 連結要素について、種数が 0 の向き付け可能閉曲面 (以下単に球面と呼ぶ) への埋め込みについて考える。図 7 に例を示す。

まず n -ボンドグラフでは、球面上への埋め込みの総数は、 n 本の辺の円順列の数すなわち $(n-1)!$ である。但し埋め込み全体の集合は、互いに鏡像をなすような $(n-1)!/2$ 個の埋め込みのペアを含んでいる。言い換えれば、 n -ボンドグラフには $(n-1)!/2$ 個の「状態」が存在し、球面を裏返して鏡像をとることで結局 $(n-1)!$ 個の埋め込みが生成できる。

次に多角形グラフ要素について、球面上への埋め込みはただ一つである。この埋め込みの鏡像は、単に多角形の内部と外部が入れ替わったものになるので、球面を裏返しても異なる埋め込みは得られない。

最後に 3 連結なグラフ要素について、以下の理由により球面への埋め込みが可能である。

- 入力グラフは、前提により平面グラフである。
- 元のグラフが平面グラフであれば、それを 3 連結要素分解してできる各グラフは平面グラフである。

実際に埋め込みを生成するには、非常に効率的な方法¹¹⁾ が利用できる。3 連結なグラフの場合、埋め込んだ球面を裏返してできる鏡像の埋め込みは、もとの埋め込みとは一致しない。

各 3 連結要素グラフの埋め込みについて、表 1 にまとめた。

5.4 2 連結なグラフの埋め込み生成

本節では、前節までの結果を利用して、2 連結な平面マルチグラフのすべての埋め込みを生成する方法を

表 1 3 連結要素の埋め込みの数
Table 1 Number of embeddings of each triconnected component

	状態の数	面の裏返し	埋め込みの数
n -ボンドグラフ	$(n-1)!/2$	あり	$(n-1)!$
多角形グラフ	1	なし	1
3 連結なグラフ	1	あり	2

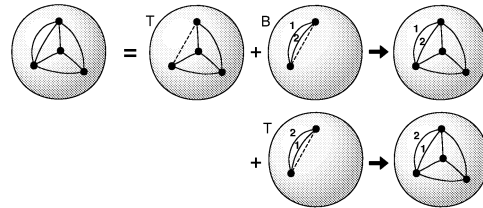


図 8 球面の裏返しによる埋め込み生成
Fig.8 Embedding by reversal of the sphere

説明する。

図 8 のように、ある 2 連結グラフが 2 つの 3 連結要素グラフから成っているものとする。一方は 3 連結なグラフであり、他方は 3-ボンドグラフである。曲面の裏返しを考慮すると、各要素グラフにはそれぞれ 2 つの埋め込みがある。

グラフ全体の球面上への埋め込みを生成するには、対応する仮想辺を結合する。これはちょうど分割の逆操作に相当する。この例では、球面の表どうしを貼り合わせるようにするやり方と、一方を裏返しにして貼り合わせるようにするやり方の 2 つがあり、結局 2 つの異なる埋め込みが生成できる。但し、グラフ全体の埋め込みについては、埋め込みとその鏡像を区別せずに 1 つとして数えている。

これを拡張すると、一般的な 2 連結平面マルチグラフについてのアルゴリズムは以下ようになる。

- (1) 入力グラフを 3 連結要素分解する。 n -ボンドグラフ要素、多角形グラフ要素、3 連結グラフ要素の数をそれぞれ $N_B(n), N_P, N_T$ とする。
- (2) 初期埋め込みとして、各グラフ要素の球面上への埋め込みを生成する。
- (3) 多角形グラフ要素だけしかないとすれば、 $N_P = 1$ のはずである。その場合は、入力グラフに対して自明な埋め込みが 1 つ存在するだけであり、アルゴリズムを終了する。
- (4) 構造グラフを作り、そこからボンドグラフ要素か 3 連結グラフ要素に相当する頂点を 1 つ選んでルート頂点とする。構造グラフは木であるので、ルート頂点を除くすべての頂点はただ 1 つの親頂点を持っている。

- (5) 子側の頂点が多角形グラフ要素であるような辺をすべて選び、それらを「不活性」辺とする。構造グラフの中には、 N_P 本の不活性辺と下式で表わされる M 本の「活性」辺が存在する。

$$M = \sum_{n \geq 3} N_B(n) + N_T - 1$$

+1 と -1 の 2 値を取る変数を、活性辺と同じ数だけ (つまり M 個) 用意する。

- (6) $n \geq 4$ の n -ボンドグラフ要素は、球面の裏返しによるもの以外にも複数の埋め込みを持つ。そこで、 $(\sum_{n \geq 4} N_B(i))$ 個の変数を用意する。各変数は、辺の数 n に依存して $(n-1)!/2$ 個の異なる値を取るものとする。
- (7) $(M + \sum_{i=4} N_B(i))$ 個の変数がそれぞれ許された値を取るものとして、すべての値の組み合わせに対して以下のことを行なう。
- (a) $n \geq 4$ のすべての n -ボンドグラフ要素について、変数の値に対応する埋め込みを並べ換えによって生成する。それをこの要素についての「参照埋め込み」と呼ぶ。その他のグラフ要素については、ステップ (2) で生成した初期埋め込みを、参照埋め込みとする。
- (b) 構造グラフのすべての頂点について、自分自身とルート頂点を結ぶ経路上の活性辺 (に対応する変数) に、-1 がいくつあるか数える。 m 個あるとして、その頂点に $(-1)^m$ を割り当てる。ルート頂点では $(-1)^0 = +1$ となる。
- (c) 構造グラフの各頂点について、頂点に割り当てられた数が -1 のときは参照埋め込みを裏返し、そうでなければ参照埋め込みをそのまま使う。埋め込みを裏返すには、そのグラフのすべての頂点で、接続する辺の並び順を逆転する。
- (d) 要素グラフを、それを埋め込んだ球面の対応する仮想辺を貼り合わせることで結合してゆき、グラフ全体の埋め込みを得る。

以上のアルゴリズムにより、すべての埋め込みを順次生成することができる。埋め込みの総数 N_{em} は次式で表される。

$$N_{em} = 2^{N_T-1} \cdot \prod_{n \geq 3} \{(n-1)!\}^{N_B(n)} \quad (1)$$

6. 曲面幾何形状の生成

6.1 曲面生成の目的

生成された球面上への埋め込みは、どれも同数の面ループからなる。面ループとは入力グラフの辺に対応する稜線の列で、閉じた曲線である。

前章の方法で生成されたすべての埋め込みに対して、含まれる面ループに曲面を張る。但し、面ループは複数の埋め込みに重複して現われるため、その総数は、埋め込みあたりの面分数と埋め込みの総数の積よりは格段に小さい。

面ループに曲面を張るのには、以下のような理由がある。

- 複数の埋め込みを比較して解を絞り込む際、曲面の幾何情報が利用できる。
- 出力モデルをいろいろな目的で利用するには、曲面の幾何が不可欠である。

6.2 曲面生成の手法

曲面生成に際しては次の基本方針をとる。

- 面ループに張る曲面は、隣接する面分とは独立に生成する。すなわち、周囲の面分が変わって曲面の張り方は同じである。
 - 面ループにはできるだけ「単純な」曲面を張る。曲面を張る具体的な手順は以下のようにした。
- (1) 面ループ上の 3 点を選び、平面を決定する。面ループ全体がその平面上にあるかどうか判定し、平面上にあればその平面を採用して終了する。
 - (2) 面ループ上の 9 点を選び、2 次曲面を決定する。面ループ全体がその曲面上にあり、かつその 2 次曲面が球面、円柱面、円錐面のどれかであるときは、その曲面を採用して終了する。
 - (3) 面ループを表わす点列と、そのいくつかの点での法線ベクトルを与え、それらを満たす自由曲面を近似メッシュとして生成する。

限られた 2 次曲面 (球面、円柱面、円錐面) だけ特別扱ったのは、この種の曲面が実用上頻繁に使われるためである。また自由曲面の生成については、物理モデルを使ってメッシュをなめらかに変形させる曲面生成手法¹²⁾ を利用した。

曲面の単純さを別な方法で定義することも、もちろん可能である。最終的には対象形状の分野 (種類) を考慮して決めるべきである。

7. 解の絞り込み

7.1 目的と手法の概要

5 章の議論により、入力グラフが 3 連結でないとき

複数の解が存在する。しかし、対象形状の性質（例えば工業的に加工可能な形状）を踏まえ、より単純な形状を表わす解釈を優先することで、解の絞り込みを行なう。絞り込みとは、明らかに不合理なもの（例えば曲面が自己干渉する解）を排除し、残ったものを順位づけすることである。

絞り込みの基準には、大きく分けて、単一の面分に関するものと複数の面分の組合せに関するものがある。

すべての面ループに曲面を張った後、何らかの基準で不合理と判定された解は考慮から外される。除外されなかった解については、各評価値の重み付き和などによって総合的な優先度を決定する。

次の2節では、そうした目的で利用する基準の例を紹介する。その選択と重み付けのしかたにより、形状の単純さを定義することになる。

7.2 単一面分の基準

単純な解釈を優先するという方針のもと、各面分に対して独立に適用できる基準には以下のようなものがある。

7.2.1 曲面の種類

6章で述べたように、各面ループにはなるべく単純な曲面を割り当てる。言い換えると、曲面が決まったときその面分の単純さが決まっている。

例えば、ある解に含まれる平面、基本的2次曲面（球面、円筒面、円錐面）、自由曲面の数をまとめて (n_p, n_q, n_f) と表わすと、 $(3, 3, 6)$ の解よりも $(5, 3, 4)$ の解のほうが単純とみなせる。あるいは、これを $e = 3n_p + n_q$ のような式で数値化し、他の評価値と結合してもよい。さらに、そもそも自由曲面は許さないという前提を採用すれば、 $n_f > 0$ となる解を排除することもできる。

7.2.2 面積

面分は、面積が小さいほど単純である。すなわち2つの解を比較するとき、面積の総和が小さいほうがより単純である。同じワイヤーフレームに対する解を比較した場合、総面積の小さな解は、角張った部分や曲面の折り返しが少ない形状を表わしている。

面分の面積を評価するには、曲面式やメッシュから計算するか、あるいはループ上の点列を $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}$ として以下のような式で概算する方法もある（但し‘ \times ’は外積を表わし、 $\mathbf{p}_n = \mathbf{p}_0$ とする）。これなら面ループの曲面を決める前に利用することもできる。

$$S = \frac{1}{2} \left| \sum_{i=0}^{n-1} (\mathbf{p}_i \times \mathbf{p}_{i+1}) \right|$$

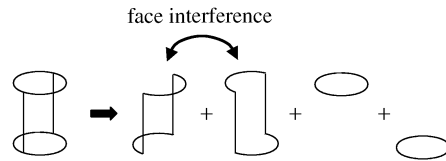


図9 干渉する面分を含む解
Fig.9 A solution with face interference



図10 稜線の並び順（右のほうが角度の和が大きい）
Fig.10 Edge ordering (The sum of angles is larger in the right.)

7.3 面分の組合せの基準

いくつかの面分の組合せについて、相対的位置関係を評価する基準を紹介する。

7.3.1 面分の干渉

面分どうしが、共有する稜線以外の場所で干渉することは許されない。図9のように、干渉する面分の組合せをもつ解は、不合理な解として排除する。

7.3.2 稜線の順番

埋め込みが1つ決まれば、すべての頂点に対して、隣接する辺の並び順が決定する。

頂点の中には、解によって異なる並びをとるものがある（但し、裏返して一致するものは同じとみなす）。そのような頂点は、多角形グラフを除いた複数の3連結要素に隣接しているか、あるいは $n \geq 4$ の n -ボンド要素に隣接している。

図10のように複数の並び順があるとき、隣接する面分の実体部分が占める角度の和を計算する。この値が、他の並び順の場合と比べて一定の比率を超えて大きいとき、その並び順を不合理として排除する。また、合理的な並び順が複数ある場合は、角度の和が小さいほど単純であるとみなす。

7.3.3 稜線での面の接続

稜線を介してつながる2枚の面分について、稜線上の法線がなす角は、接平面連続のとき0度である。この角度が180度に近いとき、両側の面分は、刃のように尖った形状かまたは非常に狭い溝を構成している。こうした形は実用的には加工しにくいいため、不合理として排除するか、単純でないとして優先度を下げることが考えられる。

8. 開いた曲面モデル

8.1 開いた曲面

これまで球面と同相な閉じた曲面モデルについて議論してきた。しかし、ワイヤフレームモデルは立体を表わすだけでなく、板金部品のような薄板形状を表わすのにも使われる。本章では、球面と同相な解から面分を1つ外して得られる解、すなわち円板と同相な解釈について検討する。

埋め込みの生成と各面分の曲面生成については、閉じた曲面の場合と同様である。違いがあるのは、取り外す面分の決定と絞り込みの方法である。

8.2 取り外す面分の決定

1つの埋め込みに N 枚の面分が含まれているとき、取り外す面分の選び方は N 通りある。閉じた曲面の場合を合わせると、1つの埋め込みから $(N+1)$ 通りの解が考えられる。その中から1つに絞る発見的な方法として、例えば以下の条件が成り立つとき面積最大の面分を取り外すというやり方がある。

$$\frac{S_{max}}{\sum S_i} > k_0 \quad \text{かつ} \quad \frac{S_{sec}}{S_{max}} < k_1$$

但し S_{max} , S_{sec} はそれぞれ1番目、2番目に大きな面分の面積、 k_0 , k_1 は1未満の正定数とする。この条件は、面積が小さいほど形状が単純であるという考え方に沿っている。

8.3 絞り込み方法の変更

前節の方法によって円板と同相な形状かを判定し、取り外す面分を決めたとする。絞り込みの際は、単独の面分の評価においても面分の組合せの評価においても、取り外す面分の効果をすべて無視すればよい。

9. 実験結果と考察

本手法を、組合せ理論計算のライブラリ LEDA^{13),14)} を用いて PC (Pentium III, 450MHz) 上で実装した。本章では例題の処理結果を示し、最後に計算量について考察する。

9.1 例題1 (球面と同相なモデル)

入力ワイヤフレームモデルを図11に、これを3連結要素分解して得た構造グラフを図12に示す。‘B’、‘P’、‘T’で始まる名前の頂点はそれぞれボンドグラフ要素、多角形グラフ要素、3連結なグラフ要素を表わす。また各要素の埋め込みについて、括弧内の‘top’と‘btm’は球面の表と裏、‘st=0/3’は3つの状態のうちの1つであることを示す。‘=anc’と付記されているのは、ルートとして選んだ頂点である。このワイヤフレームについてすべての埋め込みを生成する

と、表2のようにになる。埋め込みに含まれる61枚の面分のうち、49枚はすべての埋め込みに含まれており、このため幾何的な処理に入る前に既に確定した面分と判断できる。

表2 例題1の埋め込みのまとめ
Table 2 Summary of embeddings of Example 1

埋め込みの総数	3,456
埋め込み1つあたりの面分の数	61
全ての埋め込みに共通な面分の数	49
一部の埋め込みに現れる面分の数	61

ここで、稜線の並び順が複数ある頂点について稜線の順番の基準を適用する。該当する頂点はこのワイヤフレーム中に6つあり、どれも4本の稜線がつながっている。稜線の並び順は、埋め込みの中で実際に生じる並び順を表と裏を区別せずに数えればよいので、 n 本の稜線がつながる頂点では最大で $(n-1)!/2$ 通りある。各円順列に対して $r = (\text{角度の和})/2\pi$ を計算し、そのうち $r/r_{min} > 1.1$ を満たすものを不合理とみなし排除した。但し r_{min} は、その頂点に関して実在する並び順についての r の最小値である。結果を表3に示す。不合理な r の値を四角括弧で囲み、すべての埋め込みのうち実際にその並び順を含むものの割合を右の列に示した。

この基準に基づき、3,456通りの解から不合理なものをすべて排除すると、128通りの解が残った。それを面積の小さい順に並べ直し、面積最小の解を図13に示した。この場合、確かに望んだ埋め込みが得られた。

表3 稜線の順番の基準の適用結果
Table 3 Result of application of edge-ordering criterion

頂点の座標	円順列ごとの r (内角和/ 2π)			排除率 (%)
	(1,2,3,4)	(1,2,4,3)	(1,3,2,4)	
(-672, 0, 13)	[1.417]	[1.417]	0.998	66.7
(-371, 0, 211)	[1.091]	[1.091]	0.934	66.7
(-673, 0, 42)	[1.417]	[1.417]	0.998	66.7
(-402, 0, 182)	[0.914]	0.782	[0.915]	66.7
(-412, 0, 181)	[1.270]	0.999	[1.270]	66.7
(-364, 0, 225)	[1.185]	[1.185]	1.000	66.7

処理時間は、3連結要素分解に0.08秒、全埋め込みの生成と評価(自由曲面の生成を除く)に4.05秒を要した。

9.2 例題2 (円板と同相なモデル)

入力の3次元ワイヤフレームを図14に示す。このワイヤフレームについてすべての埋め込みを生成すると表4のようにになる。幾何的な処理に入る前に、28枚の面分のうち23枚は既に確定していることがわ



図 11 例題 1 の入力ワイヤーフレーム
Fig.11 Input wireframe of Example 1

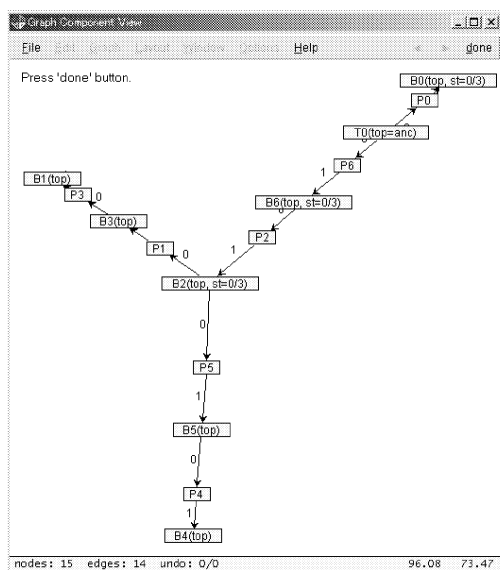


図 12 例題 1 の構造グラフ
Fig.12 Structure graph of Example 1

かる。

ここで、8.2 節で述べた取り外す面分の条件 (但し $k_0 = 0.3, k_1 = 0.5$) を使って 8 つの解を評価すると、どれも開いた曲面モデルと判定された。面積の小さい順に並べた結果が表 5 である。面積が最小の解を図 15 に示した。また面積が 2 番目に小さい解との比較を図 16 に示した。図に示した断面線からわかるように、右端の「へり」の部分が異なっている。

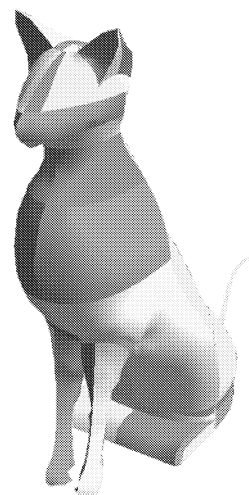


図 13 例題 1 の面積最小解
Fig.13 Minimum-area solution of Example 1

表 4 例題 2 の埋め込みのまとめ
Table 4 Summary of embeddings of Example 2

埋め込みの総数	8
埋め込み 1 つあたりの面分の数	28
全ての埋め込みに共通な面分の数	23
一部の埋め込みに現れる面分の数	12

処理時間は、3 連結要素分解に 0.02 秒、全埋め込みの生成と評価 (自由曲面の生成を除く) に 0.01 秒を要した。

表 5 例題 2 の解のまとめ
Table 5 Summary of solutions of Example 2

解の ID	開いた 曲面か	最大面分の 面積比 (%)	総面積	平面分の数
1	yes	44.4	156,627.9	24
3	yes	44.4	157,033.6	22
0	yes	44.1	157,262.0	23
2	yes	44.0	157,667.0	21
5	yes	39.1	195,368.2	23
7	yes	39.0	195,814.4	22
4	yes	38.7	196,002.3	22
6	yes	38.7	196,448.6	21

9.3 計算量

3 連結要素分解の計算量は理論上 $O(|V| + |E|)$ であり、各要素グラフについて平面性を判定して埋め込みを 1 つ決定する計算量も $O(|V| + |E|)$ である^{10),14)}。平面グラフでは Euler の公式より $|E| \leq |V| - 6$ であるので、結局ここまでの計算量は $O(|V|)$ になる。

埋め込みの総数は式 (1) のように指数的に増加し、つまり $O(\exp(k|V|))$ (k は正定数) である。従って、

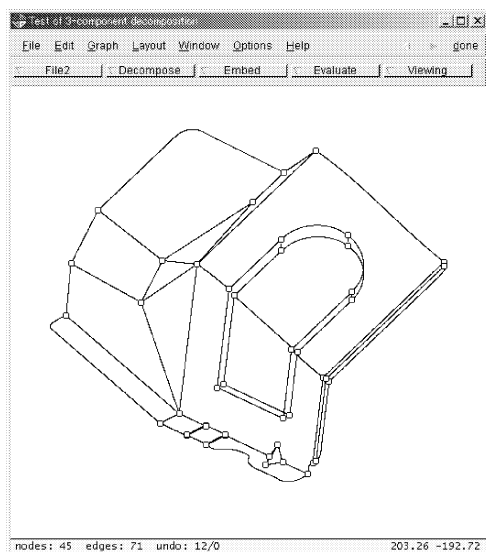


図 14 例題 2 の入力ワイヤーフレーム
Fig. 14 Input wireframe of Example 2

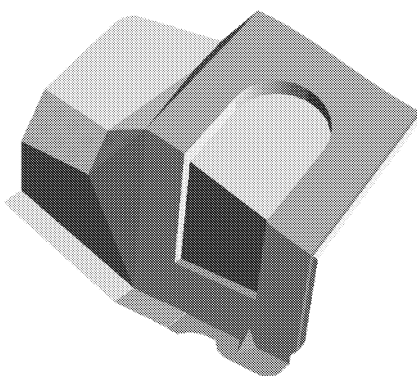


図 15 例題 2 の面積最小解
Fig. 15 Minimum-area solution of Example 2

曲面生成と絞り込みを各埋め込みごとに独立して行なうと、その計算量も $O(\exp(k|V|))$ になってしまう。

しかし曲面生成と絞り込みの計算の大部分は、個々の面分や、隣接する面分の組合せに対する処理である。面分は複数の埋め込みに現れるので、重複を避けることにより、全体の計算量を $O(m_f)$ に近づけることができる (m_f は異なる面分の総数)。

グラフが大きくなるととき 3 連結要素の数や辺の数が $O(|V|)$ だとすれば、特定の面分を含む埋め込みの数は $O(\exp(k|V|))$ で指数的に増加する。その結果 m_f は埋め込みあたりの面分の数と同じオーダーになり、曲面生成と絞り込みの計算量は $O(|V|)$ に近づく。

まとめると全体の計算量は、位相的な埋め込みの総数ではなく、入力モデルの頂点数に対してほぼ線形になる。

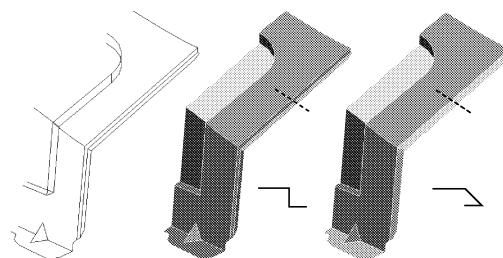


図 16 面積最小解と 2 番目に小さい解 (真中と右側)
Fig. 16 Solutions with minimum and next minimum area (middle and right)

10. おわりに

ワイヤーフレームモデルからの曲面モデルの生成問題に対し、従来手法で多く採用されていた 3 連結グラフの制約を緩め、2 連結な平面グラフを扱う方法を提案した。2 連結なグラフでは一般に複数の解があるが、3 連結要素分解によって既知の埋め込みを持つ要素グラフに分解することで、すべての位相的な解が効率よく生成できた。その過程で稜線の幾何情報をまったく使わないため、幾何的誤差の悪影響を受けることがない。また、この時点では幾何的には非現実的な解も多く含まれているが、解を構成する個々の面分については、すべての解に含まれる確定した面分とその他の不確かな面分に区別されている。

すべての位相的な解を導いた後で、「単純な」解釈を優先するという方針に基づき、まず各面分に曲面を張った。それから、幾何情報を使った発見的な手法で解の絞り込みを行なった。曲面生成と絞り込みは対象形状の分野(種類)に依存するが、いくつかの典型的な基準を提案した。実際にそれらの基準を用いて、多数の解の中から有望な解に絞り込めることを実験で確かめた。

また、球面と同様な形状だけでなく、円板と同様な形状についても同様の手法が適用できることを示した。今後の課題として以下のようなことが挙げられる。

- 2 連結の制限を緩め、1-連結または非連結なワイヤーフレームモデルまで扱えるようにする。
- 平面グラフの制限を緩め、種数が 1 以上のワイヤーフレームも扱う。現在のアルゴリズムで 3 連結な要素グラフを埋め込む部分を、種数が 1 以上の場合にも拡張すれば、それをもとに組合せ的に全体の埋め込みが生成できる。
- 設計の過程で逐次変化してゆくワイヤーフレームモデルに対して、本手法を応用した対話的な環境を構築する。

謝辞 面分の境界曲線から自由曲面を生成するプログラムを提供していただいた山田敦氏に感謝します。

参 考 文 献

- 1) 前原潤, 根上生也: 幾何学的グラフ理論, 朝倉書店 (1992).
- 2) Markowsky, G. and Wesley, M. A.: *Fleshing Out Wire Frames*, *IBM Research and Development*, Vol. 24, No. 5, pp. 582-597 (1980).
- 3) 桃井貞美, 福井幸男: ワイヤーフレームからソリッドへの一変換手法, 情報処理学会論文誌, Vol. 31, No. 1, pp. 24-31 (1990).
- 4) Hanrahan, P. M.: *Creating Volume Models from Edge-Vertex Graphs*, *Proc. SIGGRAPH '82*, Vol. 16 (3), pp. 77-84 (1982).
- 5) Dutton, R. D. and Brigham, R. C.: *Efficiently Identifying the Faces of a Solid*, *Computers and Graphics in Mechanical Engineering*, Vol. 7, No. 2, pp. 143-147 (1983).
- 6) Hopcroft, J. E. and Tarjan, R. E.: *Efficient Planarity Testing*, *Journal of the ACM*, Vol. 4, No. 21, pp. 549-568 (1974).
- 7) Ganter, M. A. and Uicker Jr., J. J.: *From Wire-Frame to Solid-Geometric: Automated Conversion of Data Representations*, *Computers in Mechanical Engineering*, Vol. 2, No. 2, pp. 40-45 (1983).
- 8) Courter, S. M. and Brewer, III, J. A.: *Automated Conversion of Curvilinear Wire-Frame Models to Surface Boundary Models; A Topological Approach*, *Proc. SIGGRAPH '86*, Vol. 20 (4), pp. 171-178 (1986).
- 9) Agarwal, S. C. and Waggenspack Jr., W. N.: *Decomposition method for extracting surface topologies from wireframe models*, *Computer-Aided Design*, Vol. 24, No. 3, pp. 123-140 (1992).
- 10) Hopcroft, J. E. and Tarjan, R. E.: *Dividing a Graph Into Triconnected Components*, *SIAM Journal on Computing*, Vol. 2, No. 3, pp. 135-158 (1973).
- 11) Tarjan, R. E.: *An Efficient Planarity Algorithm*, Technical Report STAN-CS-244-71, Comput. Sci. Dept., Stanford University (1971).
- 12) Yamada, A., Shimada, K., Furuhashi, T. and Hou, K.: *A Discrete Spring Model to Generate Fair Curves and Surfaces*, *Proc. Pacific Graphics '99*, pp. 270-279 (1999).
- 13) Max-Planck-Institut für Informatik: *LEDA (Library of Efficient Data types and Algorithms)*. <http://www.mpi-sb.mpg.de/LEDA/>.

- 14) Mehlhorn, K. and Näher, S.: *LEDA: A Platform for Combinatorial and Geometric Computing*, Cambridge University Press (1999).

付録 補助定理 1 の証明

3 連結分解では, グラフの分割と結合を行なう. 1 回の分割につき, グラフの連結な成分の数は 1 つ増え, 対をなす仮想辺も 1 組増える. 結合は分割の逆操作であり, 1 回の結合につき連結な成分は 1 つ減り, 対をなす仮想辺も 1 組減る. 従って, 連結成分の数と仮想辺の組の数の差は常に一定である. 3 連結分解を始める前は連結成分は 1 つ, 仮想辺は 0 組なので, 次式が成り立つ.

$$(\text{連結成分の数}) - (\text{仮想辺の組の数}) = 1 - 0 = 1$$

連結成分, 仮想辺の組は, それぞれ構造グラフ G_S の頂点と辺に対応するので, 結局, 次式が成り立つ.

$$|V(G_S)| - |E(G_S)| = 1$$

G_S はその生成過程から明らかに連結である. 上式を満たす連結なグラフは木であることから, G_S は木である. (証明終)

(平成 12 年 9 月 28 日受付)

(平成 13 年 1 月 22 日採録)

井上 恵介 (正会員)



1987 年東京大学工学部精密機械工学科卒業. 1989 年同大学院精密機械工学専攻修士課程修了. 同年日本アイ・ビー・エム (株) 入社. 以来, 東京基礎研究所にて幾何モデリング, CAE 関連技術の研究開発に従事.

嶋田 憲司



1983 年東京大学工学部精密機械工学科卒業. 1985 年同大学院精密機械工学専攻修士課程修了. 同年日本アイ・ビー・エム (株) 入社, 東京基礎研究所にて形状処理, ロボティクス, コンピュータグラフィックス, 計算力学などの研究に従事. 1989 年より 1993 年までマサチューセッツ工科大学博士課程. Ph.D. 1996 年よりカーネギーメロン大学機械工学科とロボット研究所の助教授を兼任. 1994 年本会山下記念研究賞, 同年ニコグラフ論文コンテスト優秀論文賞, 2000 年米国立科学財団 (NSF) CAREER AWARD 受賞. ACM, ASME, IEEE, SIAM 各会員.