

多面体の合同判定のための $n \log n$ アルゴリズム†

杉原 厚 吉††

2個の多面体が合同であるか否かを判定する効率のよいアルゴリズムを構成する。これは、多面体の稜線の数を n とするとき、計算時間の上限が $n \log n$ に比例するものである。また、3連結平面グラフの同形判定技法を利用しているにもかかわらず、対象とする多面体の頂点と稜線の作るグラフは3連結である必要も平面グラフである必要もないという著しい一般性をもっている。さらに、多面体の一部分が別の多面体の一部分と合同であるか否かも n^2 に比例した計算時間で判定できることを指摘する。本稿の成果は、多眼視などによって得られた立体形状情報から物体の種類を同定したり、立体形状設計のための計算機支援システムにおいて部品を組み立てる際のはめ合いがうまくいくか否かを判定したりするときに役立つものである。

1. まえがき

2個の多面体が合同であるか否かを判定することは、計算機による立体形状処理の種々の局面で必要とされる重要な問題である。たとえば、機械部品などを設計するための CAD システム^{1),2)}において部品同士のはめ合いがうまくいくか否かの判定、多眼視によって得られた形状情報^{3),4)}からの物体の同定などの問題は、多面体の合同判定問題に帰着できる。しかし、合同判定に限らず立体形状の処理アルゴリズムに関する研究はまだそれほど進んでいない。最近、計算幾何学とよばれる分野でとくに2次元幾何形状の処理に関する研究が急速に進んでいるが、3次元形状に関しては、凸包の構成法^{5),6)}、2組の点集合を分離する平面の構成法⁷⁾、2個の多面体の干渉の有無の判定法^{8),9)}などの成果があるだけである。

ところで、多面体の頂点と稜線の接続関係のみに着目すると、頂点を節点とし稜線を枝とする無向無ループ単純グラフが得られる。したがって、2個の多面体が構造的に等価であるか否かの判定問題は、2個のグラフが同形であるか否かの判定問題と深くかかわっていることが予想される。一方、グラフの同形判定問題は、一般のグラフに対しては効率のよいアルゴリズムは見つかっていないが¹⁰⁾、平面グラフに対しては多項式オーダーのアルゴリズムが知られている。とくに、3連結平面グラフに対しては、Weinberg¹¹⁾ が $O(n^2)$ 、Hopcroft and Tarjan¹²⁾ が $O(n \log n)$ 、Hopcroft and Wong¹³⁾ が $O(n)$ のアルゴリズムを提案してい

る。

本稿では、これらのアルゴリズムのうち文献¹²⁾に使用されている考え方を利用して、多面体の合同を判定するための効率のよいアルゴリズムを構成する。これは、多面体の稜線の数を n とするとき、計算時間の上限が漸近的に $n \log n$ に比例する方法である。

その上、文献¹²⁾のアルゴリズムが平面グラフのなかのさらに特殊なもの（3連結という性質をもったもの）のみを対象にしているのに対して、本稿で構成するアルゴリズムは、多面体の頂点と稜線の作るグラフが3連結であることも平面グラフであることも要求しない。すなわち、ここで対象とする多面体は、凸とは限らず（多面体が凸ならばその頂点と稜線の作るグラフは3連結である¹⁴⁾）凹であってもよく、さらに球に同相とは限らず（多面体が球と同相ならば頂点と稜線の作るグラフは平面グラフである）トーラスなどに同相であってもかまわない。このように、もとなっているアルゴリズムが強い制限を受けているにもかかわらず、本稿のアルゴリズムが広い適用範囲をもつという顕著な結果が得られるのは、次のような理由によるものである。

文献¹²⁾のアルゴリズムの基本になっている性質は、(i) 3連結平面グラフを枝が交差しないように球面に描く方法は2とおりにしかなく（ある描き方とそれを裏返した描き方）、(ii) 球面の上に枝が交差しないように描かれた二つのグラフの“描かれ方”が同じか否かを判定する効率のよいアルゴリズムが作れる、の二つである。一方、多面体が与えられた場合には、その頂点と稜線の作るグラフは、中に物質の詰まった物体の表面に初めから描かれているとみなすことができ、描き方が何とおりにあるかを心配する必要はない。したが

† An $n \log n$ Algorithm for Determining the Congruity of Polyhedra by KOKICHI SUGIHARA (Department of Information Science, Faculty of Engineering, Nagoya University).

†† 名古屋大学工学部情報工学科

って、性質(i)のなかの“3連結”という制限は除くことができる。また、二つのグラフを比べるときには接続関係という位相的情報のみが対象となるが、二つの多面体を比べるときには、稜線の長さや面の交角などの計量的情報も比較の対象となる。この差を利用すると、多面体の合同判定の場合には“球面上”とは限らず“表裏の区別のある面上”にグラフが描かれていればよいということがわかる(この性質を導くことが本稿の理論的内容の中心である)。したがって、性質(ii)の中の“球同相”という制限もはずすことができるわけである。

なお、3連結平面グラフの同形判定アルゴリズムには $O(n)$ のもの¹³⁾もある。しかし、これを提案している Hopcroft から自身もいっているように、このアルゴリズムはオーダは1次であっても係数が大きすぎて実用的ではなく、また平面グラフの性質を本質的に利用しているため一般の多面体の合同判定問題に応用することはむずかしいように思われる。

まず2章でいくつかの準備をしたあと、3章で2個の多面体が合同であるための必要十分条件を導く。4章では、それを利用した合同判定アルゴリズムを構成するとともに、それを実際の場面で使う際の注意や拡張の可能性などを論じる。また5章では、2個の多面体が部分的に合同であるか否かの判定も多項式時間でできることを指摘する。

2. 準備

3次元ユークリッド空間を \mathbf{R}^3 で表す。有限個の平面多角形で囲まれた有界で閉じた \mathbf{R}^3 の部分集合を**多面体**という。したがって、ここでいう多面体は、たんなる物体形状ではなくて、3次元空間における位置や姿勢が固定されたものである。多面体と外の空間との境界をなす多角形を**面**、2個の面の境界をなす線分を**稜線**、稜線の端点を**頂点**とよぶ。多面体 P の面の集合を $F(P)$ 、稜線の集合を $E(P)$ 、頂点の集合を $V(P)$ で表し、 P の境界(表面)を ∂P で表す。任意の2点 $x, y \in \mathbf{R}^3$ に対して、 x と y の距離を $d(x, y)$ と表す。 \mathbf{R}^3 からそれ自身への1対1写像 T が、任意の $x, y \in \mathbf{R}^3$ に対して $d(x, y) = d(T(x), T(y))$ を満たすとき、 T を3次元合同変換とよぶ。合同変換 T が \mathbf{R}^3 の右手系の直交座標軸をやはり右手系の直交座標軸に移すとき、 T は**向きを変えない**という。任意の部分集合 $X \subseteq \mathbf{R}^3$ に対して $T(X) = \{T(x) | x \in X\}$ と定義する。2個の多面体 P_1, P_2 に対して、向きを変

えない3次元合同変換 T で $P_2 = T(P_1)$ を満たすものが存在するとき、 P_1 と P_2 は**合同**であるという。また、 $P_2 = T(P_1)$ のとき、合同変換 T から自然に導かれる $F(P_1)$ と $F(P_2)$ 、 $E(P_1)$ と $E(P_2)$ 、 $V(P_1)$ と $V(P_2)$ の間の1対1対応を**合同対応**という。連結な部分集合 $X \subseteq \mathbf{R}^3$ に対して、 X 内の任意の閉曲線が X 内での連続変形によって1点に変形できるとき、 X は**単連結**であるという。

多面体は次の仮定1, 2, 3を満たすものとする。

【仮定1】 多面体 P に対して、 $F(P)$ を節点集合とみなし、2個の面 $f_1, f_2 \in F(P)$ が稜線を共有するときかつそのときのみ f_1 と f_2 を結ぶ枝があるとみなしてできる無向グラフは連結である。

この仮定によって、非連結であったり二つの部分が1点のみで接しているような多面体は除かれる。全体がつながっていないかたり1点のみでつながっていたりする物体は全体を剛に保つことが物理的に不可能であるから、この仮定は何ら実質的制限にはなっていない。

【仮定2】 多面体 P の境界 ∂P 上の任意の点 x に対して、 x を中心とし十分小さな任意の半径 $r > 0$ をもった球の内部に属する点の集合 $S(x; r)$ を作ると、 $S(x; r) \cap P - \partial P$ は非空で単連結な領域となる。

これは、少々病理的な多面体を除外するために設けた仮定である。小球 $S(x; r)$ と多面体内部との共通部分が非空であることから、紙細工のような厚みのない物体は除かれ、この共通部分が単連結であることから、**図1(a), (b)**のような特殊な稜線や頂点をもった多面体が除かれる。その結果、仮定2を満たす多面体においては、面の表と裏が区別でき、各稜線はちょうど2個の面と隣接し、さらに頂点を多面体の外から眺めてそこに隣接するすべての面と稜線を反時計回りに並べる並べ方が出発点の任意性を除いて一意に決まる。

【仮定3】 多面体 P の各面からその境界をなす辺を除いた領域は単連結である。

この仮定は、 P の各面が穴のない(ただし面の外側

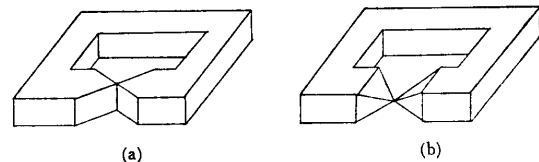


図1 仮定2に反する多面体

Fig. 1 Polyhedra to be excluded from consideration.

の境界と点で接する穴はあってもよい) 多角形であることを意味する. これによって, 1 個の面の上の稜線は互いに連結していることが保証される.

多面体 P に対して, P の頂点を節点とし稜線を枝とする無向グラフを $G(P)=(V(P), E(P))$ とおく. このグラフの各節点は少なくとも 3 本の枝をもつ. なぜなら, 多面体の頂点は 3 個以上の面と隣接しているからである. また, $E(P)$ の各要素を互いに逆方向の向きをもった 2 本の有向枝で置き替えてできる有向グラフを $\vec{G}(P)=(V(P), \vec{E}(P))$ とおく. すなわち

$$\vec{E}(P) = \{(v_1, v_2), (v_2, v_1) \mid \{v_1, v_2\} \in E(P)\}$$

である. $e=(v_1, v_2) \in \vec{E}(P)$ のとき, v_1, v_2 をそれぞれ e の始点, 終点という. また, e の向きを反転したものを e' と記す. グラフ $\vec{G}(P)$ に対して, 頂点 v_1 から頂点 v_n への有向路とは, $\vec{E}(P)$ の要素の列で $((v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n))$ の形をしたもののことである.

さて, 上で定義した $\vec{G}(P)$ 自身は抽象的なグラフであるが, 本稿では $\vec{G}(P)$ とそれを導出するもとになった多面体 P との複合概念のことも “グラフ $\vec{G}(P)$ ” とよぶことにする. すなわち, 今後はとくにことわらなくても, グラフ $\vec{G}(P)$ とは多面体 P の表面に描かれたグラフを指すものとする. したがって, たとえば, 各枝 $e \in \vec{E}(P)$ はそれに対応する P の稜線の位置や長さに相当する計量的な性質ももっており, また “頂点の回りで反時計回りに枝をたどる” などという言い方も意味をもつ.

グラフ $\vec{G}(P)$ の各枝 $e \in \vec{E}(P)$ に対して, e の終点を始点とする枝で e から出発して e の終点の回りを反時計回りに回ったとき最初に出会うものを $g_R(e)$, 時計回りに回ったとき最初に出会うものを $g_L(e)$ とする. g_R, g_L はそれぞれ $\vec{E}(P)$ の中の 1 対 1 写像で, したがって, 逆写像 g_R^{-1}, g_L^{-1} も $\vec{E}(P)$ のなかの 1 対 1 写像である (図 2 を参照). 枝 e の向きに向かって右側に隣接した面を $f_R(e)$, 左側に隣接した面を $f_L(e)$ とする. さらに, $f_R(e), f_L(e)$ が載っている表裏の区別のある無限平面を $H_R(e), H_L(e)$ で表す.

枝 $e \in \vec{E}(P)$ に対応する P の稜線の長さを l とし, 面 $f_R(e)$ と面 $f_L(e)$ のなす角を φ とする ($0 < \varphi < 2\pi$, $\varphi \neq \pi$). ただし, φ は多面体 P の内側を通過して計測した角度とする. したがって, $0 < \varphi < \pi$ なら稜線 e は尾根をなし, $\pi < \varphi < 2\pi$ なら稜線 e は谷をなす. また

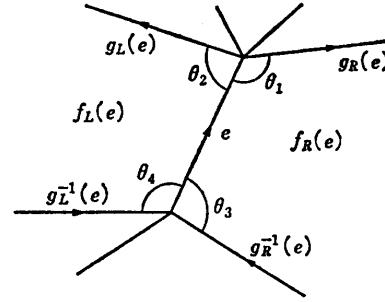


図 2 g_R, g_L, λ の定義
Fig. 2 Definition of g_R, g_L and λ .

図 2 のように, 枝 e と枝 $g_R(e), g_L(e)$ とのなす角をそれぞれ θ_1, θ_2 とする ($0 < \theta_1, \theta_2 < 2\pi$). そして, 枝 e に対する 4 次元ベクトル $\lambda(e)$ を

$$\lambda(e) = (l, \varphi, \theta_1, \theta_2)$$

と定義する.

グラフ $\vec{G}(P)$ の有向路 (e_1, \dots, e_n) は, 各 $i (1 \leq i \leq n-1)$ に対して $e_{i+1} = g_R(e_i)$ または $e_{i+1} = g_L(e_i)$ が成り立つとき, 原始的であるという. 原始的な有向路 $p = (e_1, \dots, e_n)$ に対して, $g_R(e_i) = e_{i+1}$ なら $c_i = 1$, $g_L(e_i) = e_{i+1}$ なら $c_i = -1$ とおき (P の各頂点は 3 本以上の稜線と隣接しているから常に $g_R(e_i) \neq g_L(e_i)$ が成り立ち, c_i は一義的に決まる), c_i を並べてできる $n-1$ 次元ベクトルを $\alpha(p) = (c_1, \dots, c_{n-1})$ とおく.

P_1, P_2 を二つの多面体とする. $\vec{G}(P_1)$ の原始的な有向路 $p_1 = (e_1, \dots, e_n)$ と $\vec{G}(P_2)$ の原始的な有向路 $p_2 = (e'_1, \dots, e'_n)$ が $\alpha(p_1) = \alpha(p_2)$ を満たすとき, p_2 は p_1 と同順路であるという. $e_1 \in \vec{E}(P_1), e'_1 \in \vec{E}(P_2)$ に対して, e_1 から始まる任意の原始的有向路 $p_1 = (e_1, \dots, e_n)$ と e'_1 から始まる p_1 と同順路の原始的有向路 $p_2 = (e'_1, \dots, e'_n)$ が常に $\lambda(e_i) = \lambda(e'_i)$ ($i = 1, \dots, n$) を満たすとき, e_1 と e'_1 は識別不可能であるという. また, この定義は P_1 と P_2 が同一の多面体の場合にも有効なものであるとみなすことにする. したがって, 1 個の多面体 P の枝集合 $\vec{E}(P)$ の要素同士にも識別不可能であるかないかという関係が定義され, 結局, 識別不可能という 2 項関係は, 任意個の多面体 P_1, P_2, \dots, P_i に関する枝集合 $\vec{E}(P_1) \cup \vec{E}(P_2) \cup \dots \cup \vec{E}(P_i)$ のなかの同値関係とみなすことができる.

3. 基本定理

本章では, アルゴリズムを構成するための基礎とな

る性質を導く。

【補助定理 1】 多面体 P は仮定 2 を満たし $p=(e_1, \dots, e_n)$ は $\vec{G}(P)$ の任意の原始的有向路であるとする。稜線 e_1 と平面 $H_R(e_1)$ の 3 次元空間における位置、および $\lambda(e_i), i=1, \dots, n$, が与えられたとき、稜線 e_n と平面 $H_R(e_n), H_L(e_n)$ の 3 次元空間での位置は一義的に定まる。

(証明) 稜線 e_i と平面 $H_R(e_i)$ が空間に固定されているとき、 $e_{i+1}, H_R(e_{i+1}), H_L(e_{i+1})$ が空間に固定されることを示せばよい。いま e_i と $H_R(e_i)$ が空間に固定されているものとする。(i) $e_{i+1}=g_R(e_i)$ のとき: $H_R(e_{i+1})=H_R(e_i)$ であるから $H_R(e_{i+1})$ の位置は確定する。 e_i と e_{i+1} のなす角および e_{i+1} の長さが $\lambda(e_i)$ の第 3 成分と $\lambda(e_{i+1})$ の第 1 成分として与えられているから e_{i+1} の位置が確定する。また $\lambda(e_{i+1})$ の第 2 成分より $H_L(e_{i+1})$ の位置も確定する。(ii) $e_{i+1}=g_L(e_i)$ のとき: $\lambda(e_i)$ の第 2 成分から $H_L(e_i)$ の位置が確定するから、右と左を入れ替えて(i)と同様の議論をすれば、 $e_{i+1}, H_R(e_{i+1}), H_L(e_{i+1})$ の位置が確定することがいえる。(証了)

【補助定理 2】 多面体 P_1, P_2 は仮定 2 を満たすとする。 $p_1=(e_1, \dots, e_n)$ は $\vec{G}(P_1)$ の任意の原始的有向路、 $p_2=(e'_1, \dots, e'_n)$ は任意の枝 e'_1 から始まり p_1 と同順路の $\vec{G}(P_2)$ の原始的有向路であり、さらに $i=1, \dots, n$ に対して $\lambda(e_i)=\lambda(e'_i)$ が成り立つとする。このとき、 $e'_1=T(e_1), H_R(e'_1)=T(H_R(e_1))$ を満たす 3 次元合同変換 T は、 $e'_n=T(e_n), H_R(e'_n)=T(H_R(e_n)), H_L(e'_n)=T(H_L(e_n))$ を満たす。

(証明) 補助定理 1 より、 $e_1, H_R(e_1)$ と $e_n, H_R(e_n), H_L(e_n)$ の相対的位置関係は、 $e'_1, H_R(e'_1)$ と $e'_n, H_R(e'_n), H_L(e'_n)$ の相対的位置関係と等しい。したがって、上の補助定理が成り立つ。(証了)

【補助定理 3】 多面体 P は仮定 1, 2, 3 を満たすものとする。任意の $e, e' \in \vec{E}(P)$ に対して、 e から始まり e' または $(e')'$ で終わる原始的有向路が存在する。

(証明) 仮定 1 より、 P の面の列 f_1, f_2, \dots, f_n で、 $1 \leq i \leq n-1$ に対して f_i と f_{i+1} はある稜線 (これを e_i とおく、ただし e_i の向きは少しあとまで伏せておく) を共有し $f_1=f_R(e), f_n=f_R(e')$ となるものが存在する。そこで、 e から始まる次のような原始的有向路を作ることができる。すなわち、まず e から面 f_1 に沿って e_1 まで進み (e_1 の向きはこの進行方向と一致するものを採用する)、次に f_2 に沿って e_2 まで進

む (同様にここで e_2 の向きを決める)、ということを繰り返して e_{n-1} まで有向路を延ばす。仮定 3 より、1 個の面の境界をなす稜線は互いに連結しており、面 f_i に沿って e_{i-1} から e_i へ常に到達可能である。また、この有向路は常に左右どちらかの面に沿って進んでいるから原始的である。最後に e_{n-1} から f_n に沿って進めば、いつか e' あるいは $(e)'$ に到達する。(証了)

【定理 1】 P_1, P_2 は仮定 1, 2, 3 を満たす任意の多面体であるとする。 $e_1 \in \vec{E}(P_1)$ を $e_2 \in \vec{E}(P_2)$ に向きも含めて対応させる P_1 と P_2 の合同対応が存在するための必要十分条件は、 e_1 と e_2 が識別不可能であることである。

(証明) e_1 を e_2 に対応させる合同対応があれば、明らかに e_1 と e_2 は識別不可能である。いま、逆に、 e_1 と e_2 は識別不可能であるとする。このとき、 $e_2=T(e_1), H_R(e_2)=T(H_R(e_1))$ を満たす合同変換 T は一義的に定まる¹⁵⁾。 $e_3 \in \vec{E}(P_1)$ を任意の枝とする。補助定理 3 より、 e_1 から出発して e_3 または $(e_3)'$ で終わる原始的有向路 p_1 が存在する。 e_2 から始まる p_1 と同順路の原始的有向路を p_2 とし、その最後の枝を $e_4 \in \vec{E}(P_2)$ とする。 e_1 と e_2 は識別不可能であるから、補助定理 2 より、 $e_4=T(e_3), H_R(e_4)=T(H_R(e_3)), H_L(e_4)=T(H_L(e_3))$ が成り立つ。すなわち、 P_1 の任意の稜線あるいは面に対して P_2 のある稜線あるいは面で変換 T によって一致するものがあることが示された。 P_2 の任意の稜線と逆変換 T^{-1} に対して同様の議論を行えば、 P_1 の稜線および面と P_2 の稜線および面が変換 T によって 1 対 1 対応することがわかる。したがって、 P_1 と P_2 は合同である。(証了)

4. 合同判定法

4.1 基本アルゴリズム

定理 1 と文献¹²⁾のアルゴリズムの基礎となっている定理とは、内容も導出方法もまったく異なっているが、“形式”は同じである。したがって、文献¹²⁾のアルゴリズム技法をここでも利用することができる。本節では、この方針に従って、2 個の多面体 P_1, P_2 が合同であるか否かを判定するアルゴリズムを構成する。

計算機モデルはランダム・アクセス・モデル¹⁶⁾を使用する。また、 P_1, P_2 の形状は適当なデータ構造¹⁷⁾のもとに計算機に入っており、任意の枝 $e \in \vec{E}(P_i)$ ($i=1, 2$) に対して、 $g_R(e), g_L(e), g_R^{-1}(e), g_L^{-1}(e), e$ の

始点・終点の3次元座標値, e の両側の面の平面方程式などは $O(1)$ の時間で参照できるものとする。このとき, 次のアルゴリズムによって多面体 P_1, P_2 が合同であるか否かを判定することができる。このアルゴリズムに現れる $B(1), B(2), \dots$ は, 初期には $\vec{E}(P_1) \cup \vec{E}(P_2)$ のある粗い分割を表し, 最終的には識別不可能な枝集合への分割を表すものである。

[アルゴリズム 1: 多面体の合同判定法]

(ステップ 1) すべての $e \in \vec{E}(P_1) \cup \vec{E}(P_2)$ に対して $\lambda(e)$ を計算する。

(ステップ 2) 任意の $e_1, e_2 \in \vec{E}(P_1) \cup \vec{E}(P_2)$ に対して $\lambda(e_1) = \lambda(e_2)$ かつそのときのみ e_1 と e_2 が同一成分に属するように $\vec{E}(P_1) \cup \vec{E}(P_2)$ を分割する。この分割によって生じた成分を $B(1), B(2), \dots, B(k)$ とする。 $B(i) \subseteq \vec{E}(P_1)$ または $B(i) \subseteq \vec{E}(P_2)$ を満たす $i(1 \leq i \leq k)$ が存在すれば P_1 と P_2 は合同ではないと判定して処理を終了する。

(ステップ 3) 図 3 に示す細分手続きによって, ステップ 2 で得られた分割を細分する。その結果得られる成分を $B(1), \dots, B(l)$ とする。

```

begin
1 WAIT ← {(1, R), (1, L), ..., (k, R), (k, L)};
2 l ← k;
3 while WAIT not empty do
  begin
4 select and delete (i, D) from WAIT;
5 MOVE ← {σD-1(e) | e ∈ B(i)};
6 for each j such that B(j) ∩ MOVE ≠ ∅ and B(j) ⊄ MOVE do
  begin
7 l ← l + 1;
8 create new block B(l);
9 B(l) ← B(j) ∩ MOVE;
10 B(j) ← B(j) - B(l);
11 if (j, D) ∈ WAIT then add (l, D) to WAIT
  else
12 if |B(j)| ≤ |B(l)| then add (j, D) to WAIT
13 else add (l, D) to WAIT
  end
  end
end
end

```

図 3 細分手続き
Fig. 3 Sub-partitioning procedure.

(ステップ 4) $B(1) \cap \vec{E}(P_1) \neq \emptyset$ かつ $B(1) \cap \vec{E}(P_2) \neq \emptyset$ ならば P_1 と P_2 は合同であると判定し, それ以外のときは合同ではないと判定する。(アルゴリズム了)

図 3 の細分手続きの記述に使用した言語は Pidgin ALGOL¹⁶⁾ であり, また文字 D は R または L を値と

してとる変数である。この手続きは, ステップ 2 で得られた分割 $B(1), \dots, B(k)$ を入力したとき, この分割の細分で互いに識別不可能な枝同士が同一成分に属するようなもの $B(1), \dots, B(l)$ を出力するもので, 文献¹²⁾の手続きと等価なものである。

[補助定理 4] 図 3 の細分手続きの最終結果 $B(1), \dots, B(l)$ は, 枝集合 $\vec{E}(P_1) \cup \vec{E}(P_2)$ の識別不可能な同値類への分割に一致する。また $|E(P_1)| = |E(P_2)| = n$ のとき, この手続きの実行時間の上限は $n \log n$ に比例する。

証明は文献¹²⁾と同じなので省略する (文献¹⁶⁾ではもっと一般的な枠組のなかでいっそう詳しい証明がなされている)。

[定理 2] P_1, P_2 は仮定 1, 2, 3 を満たす多面体で, $|E(P_1)| = |E(P_2)| = n$ とする。このとき, アルゴリズム 1 は P_1 と P_2 が合同であるか否かを正しく判定し, その処理時間の上限は $O(n \log n)$ である。

(証明) アルゴリズム 1 が終了するのは, ステップ 2 または 4 においてである。ステップ 2 において $B(i) \subseteq \vec{E}(P_1)$ ならば, $B(i)$ に属する枝は $\vec{E}(P_2)$ のどの枝とも識別不可能ではなく, 定理 1 より P_1 と P_2 は合同ではない。 $B(i) \subseteq \vec{E}(P_2)$ の場合も同様である。一方, 補助定理 4 より, ステップ 3 で得られた分割結果は枝集合の識別不可能な同値類への分割であるから, 定理 1 より, ステップ 4 での判定結果は正しい。

各枝 e に対する $\lambda(e)$ の計算は $O(1)$ の時間で実行できるから, ステップ 1 は $O(n)$ で実行できる。 $\lambda(e)$ は 4 次元ベクトルであるから, 4 文字の単語を辞書式に並べる技法¹⁶⁾を利用すると, ステップ 2 は $O(n \log n)$ で実行できる。ステップ 3 は補助定理 4 より $O(n \log n)$ で実行でき, ステップ 4 は $O(n)$ で実行できる。したがって, 全体の実行に要する時間は $O(n \log n)$ である。(証了)

多面体 P が球と同様な場合には, 平面グラフの基本的性質から $|E(P)| \leq 3|V(P)|$ が成り立つ¹⁸⁾ので, 合同判定問題の規模を頂点数で表しても, 定理 2 と同じ手間の評価が得られる。

4.2 いくつかの改良・拡張など

本節では, アルゴリズム 1 を実際に使用する際の注意, 改良の可能性, 拡張の可能性などを列挙する。

(1) アルゴリズム 1 を実行する前に, 2 個の多面体の頂点数・稜線数などが一致するかどうかを確かめておくべきことはいうまでもない。もしこれらの一つでも一致しなければ, アルゴリズム 1 を実行する

までもなく、2個の多面体は合同ではないと判定できる。

(2) ステップ2での初期分割は、 $\lambda(e)$ の値の違いによる分割よりは細かくて識別不可能な同値類への分割より粗ければ、できるだけ細かいほうがよい。したがって、 $\lambda(e)$ 以外の情報でより細かい分割のために簡単に利用できるものは利用すべきであろう。たとえば、枝 e の始点、終点に隣接する稜線の本数を m_1, m_2 , 面 $f_R(e), f_L(e)$ を囲む稜線の本数を m_3, m_4 とおいて、8次元ベクトル $\lambda'(e) = (m_1, m_2, m_3, m_4, \lambda(e))$ を作り、 $\lambda'(e)$ の値の違いによる分割を初期分割として採用することができる。一般にこのほうがより細かい分割が得られ、後の実行時間を短縮することができる。

(3) $\lambda(e)$ の成分は実数値であるから、デジタル化の誤差を伴う実際の場面では、 $\lambda(e_1) = \lambda(e_2)$ が厳密に成り立つことは期待できない。したがって、何らかの基準で $\lambda(e_1)$ と $\lambda(e_2)$ が“ほぼ”等しいか否かを判定しなければならない。(たとえば、 ε をある小さな正数とし $\lambda(e_i) = (x_i, y_i, z_i, u_i)$ としたとき、 $\max(|x_i - x_j|, |y_i - y_j|, |z_i - z_j|, |u_i - u_j|) < \varepsilon$ を満たすときかつそのときに限って $\lambda(e_i)$ と $\lambda(e_j)$ はほぼ等しいと定義することができる。)しかし、ほぼ等しいという関係は一般には同値関係ではない。したがって、形状データの誤差が大きいとステップ2における分割が定義できなくなってしまうことがある。(2)で考えた $\lambda'(e)$ の初めの4成分を使って枝集合をあらかじめ分割しておく(この4成分は整数であるからデジタル化の誤差の影響を受けず分割はいつも定義される)ことによって、このような事態はある程度避けることができよう。しかし、本来は、アルゴリズム1は、与えられたデータが合同判定のために十分な精度をもっているという前提のもので使うべきである。大きな計測誤差が含まれる場合には、グラフ $G(P)$ そのものにも誤差が含まれるであろうから、初めから誤差の存在を考慮した手法¹⁹⁾を採用すべきであろう。

(4) ステップ3のなかの細分手続きは枝集合を識別不可能な同値類へ分割するものである。しかし、ただたんに合同であるか否かのみを問題にする場合には、必ずしも分割を最後まで実行する必要はない。すなわち、途中で $\vec{E}(P_1)$ あるいは $\vec{E}(P_2)$ の一方に含まれてしまう成分が生じたら、その時点で合同ではないと判定して処理を打ち切ることができる。この判定は、図3の10行目と11行目の間で実行すればよい。

(5) アルゴリズム1は3個以上の多面体の間の合

同関係を決定する場合にもそのまま利用できる。すなわち、 t 個の多面体 P_1, P_2, \dots, P_t に対して $\vec{E}(P_1) \cup \vec{E}(P_2) \cup \dots \cup \vec{E}(P_t)$ の分割を同じ方法で行うことによって、与えられた多面体集合を合同な同値類に分割できる。また、1個の多面体 P_1 に対して $\vec{E}(P_1)$ の分割を施せば、 P_1 のすべての自己合同対応を見つけることができる。

(6) アルゴリズム1は、2個の多面体が相似であるか否かの判定にも利用できる。そのためには、前処理としてあらかじめ多面体の大きさを正規化しておけばよい。この正規化は、2個の多面体の最も長い稜線同士の本長さが一致するように一方の多面体を拡大または縮小することによって、 $O(n)$ の時間で実行できる。

5. 部分的に合同な多面体の判定

ある多面体の一部分がもう一つの多面体の一部分と合同であるか否かを判定する問題は、多面体の合同判定と同様に(あるいはそれ以上に)重要な問題である。これは、たとえば距離計測^{20), 21)}によって得られる物体の“見えている部分”の形状データから物体の種類を同定するときなどに生じる問題である。

グラフの場合の部分同形判定問題は、一般に同形判定問題と比べて非常にむずかしいことが知られている²²⁾。これは、与えられた部分グラフがもとのグラフからどの枝を除いてできたものなのかという情報が与えられないため、調べるべき場合の数が膨大になるからである。一方、多面体の部分合同判定を必要とする実際の場面では、どの部分が見えていてどの部分が隠れているかという情報も同時に与えられることが多い。したがって、稜線が欠けているかもしれない範囲が限定されるため、グラフの場合ほどむずかしくはならない。

前章の技法がそのままこの問題に利用できるわけではないが、ある種の部分合同判定問題に対しては、計算時間が $O(n^2)$ の単純なアルゴリズムを構成することができる。このことを指摘するのが本章の目的である。

いま、多面体 P の面集合が二つの部分集合 $F^*(P)$ と $F^*(P)$ に分割され、前者に属する面の形状データだけが与えられているものとする。 $F^*(P)$ に属する面を見えている面とよぶ。見えている面の境界をなす稜線を見えている稜線とよび、その集合を $E^*(P)$ で表す。見えている稜線の端点を見えている頂点とよび、その集合を $V^*(P)$ で表す。このとき、別の多面体 P^*

(これをモデルという)の面集合 $F(P^*)$ の部分集合 X と 3次元合同変換 T が存在して, $F^v(P)$ の要素と X の要素が表裏の向きも含めて T によって 1対1対応するとき, P は $F^v(P)$ に関して P^* と部分的に合同であるという.

見えている部分に関して次の仮定を設ける.

[仮定 4] $F^v(P)$ に属する面を節点とみなし, $f_1, f_2 \in F^v(P)$ が稜線を共有しているときかつそのときのみ f_1 と f_2 を枝で結んでできる無向グラフは連結である.

$E^v(P)$ に属する枝を互いに逆方向の向きをもった 2本の枝で置き替えてできる有向グラフを $\vec{G}^v(P) = (V^v(P), \vec{E}^v(P))$ とする. そして, $\vec{E}^v(P)$ を次の三つの集合に分割する.

$$\vec{E}_{RL}^v(P) = \{e \mid e \text{ の両側の面が見えている}\}$$

$$\vec{E}_R^v(P) = \{e \mid e \text{ の右側の面のみが見えている}\}$$

$$\vec{E}_L^v(P) = \{e \mid e \text{ の左側の面のみが見えている}\}$$

任意の $e \in \vec{E}^v(P)$ に対して, $l, \varphi, \theta_1, \theta_2$ を 2章と同様に定義し, さらに e と $q_R^{-1}(e)$ のなす角を θ_3 , e と $q_L^{-1}(e)$ のなす角を θ_4 とおく (図 2 を参照). そして, 6次元ベクトル $\mu(e)$ を次のように定義する.

$$\mu(e) = \begin{cases} (l, \varphi, \theta_1, \theta_2, \theta_3, \theta_4) & e \in \vec{E}_{RL}^v(P) \\ (l, *, \theta_1, *, \theta_3, *) & e \in \vec{E}_R^v(P) \\ (l, *, *, \theta_2, *, \theta_4) & e \in \vec{E}_L^v(P) \end{cases}$$

ただし, $*$ は値が不定であることを表す記号である. $\mu(e_1)$ と $\mu(e_2)$ の一方または両方で不定な成分は無視し, 両方で値の確定している成分同士のみを比べてとき互いに等しいなら, $\mu(e_1) = \mu(e_2)$ と定義する. また, モデル多面体 P^* の枝 e に対しては, P^* のすべての部分が見えているものとみなして $\mu(e)$ を定義する.

枝 $e_1 \in \vec{E}^v(P)$, $e_1^* \in \vec{E}^v(P^*)$ に対して, $\vec{E}^v(P)$ の要素のみから成る任意の原始的有向路 $p_1 = (e_1, e_2, \dots, e_n)$ と, e_1^* から始まり p_1 と同順路の原始的有向路 $p_2 = (e_1^*, e_2^*, \dots, e_n^*)$ が常に $\mu(e_i) = \mu(e_i^*)$ ($i=1, \dots, n$) を満たすとき, e と e^* は $F^v(P)$ に関して部分的に識別不可能であるという.

このとき, 次の定理が成り立つ (証明の道筋は定理 1 の場合とほぼ同じなので省略する).

[定理 3] P と P^* を仮定 1, 2, 3 を満たす多面体とし, $F^v(P)$ は仮定 4 を満たすものとする. 任意の $e \in \vec{E}^v(P)$, $e^* \in \vec{E}^v(P^*)$ に対して, P が $F^v(P)$ に関して P^* と部分的に合同となる合同対応で向きも含めて

e を e^* に対応させるものが存在するための必要十分条件は, e と e^* が $F^v(P)$ に関して部分的に識別不可能であることである.

この定理の条件が満たされるか否かは単純な方法で判定できる. すなわち, まず $\vec{E}^v(P)$ の一つの要素 e_0 を選び固定する. 次に, $\vec{E}^v(P^*)$ の各要素 e^* に対して, e_0 と e^* が部分的に識別不可能であるか否かを調べ, 識別不可能なものが存在すれば部分的に合同であると判定し, それ以外のときは部分的に合同ではないと判定する. 部分的に識別不可能であるか否かを調べるためには, e_0 から出発してすべての枝を 1 回ずつ通るように原始的有向路をたどり (これは有向路の分岐点をプッシュ・ダウン・スタックに登録することによって実現できる), それと同順路な $\vec{G}^v(P^*)$ の原始的有向路の上の枝と μ の値を比べればよい. したがって, $|E(P^*)| = n$, $|E^v(P)| = m$ とすると, 部分的に合同であるか否かの判定に要する計算時間の上限は $O(mn)$ である.

6. むすび

2個の多面体が合同であるか否かを判定する効率のよいアルゴリズムを構成した. これは, 3連結平面グラフの同形判定技法を利用したものであるにもかかわらず, 対象とする多面体の頂点と稜線の作るグラフは 3連結である必要も平面グラフである必要もないという著しい一般性をもっている. また, ある多面体の一部分がもう一つの多面体の一部分と合同であるか否かの判定も多項式時間で実行できることを指摘した. これは, アルゴリズム技法上の顕著な工夫を含むものではないが, グラフの部分同形の判定が極端にむずかしい問題であるという事実との対比において興味深い結果であると思われる.

このように合同・部分合同のいずれの場合もグラフの同形・部分同形の判定法より効率のよいアルゴリズムが構成できた最大の理由は, グラフは接続関係という位相的情報しかもっていないのに対して, 多面体は稜線の長さなどの計量的情報ももっており, それが利用できたという点にある.

多面体形状の情報処理に関して解決すべき問題はまだまだたくさんある (とくに処理効率の改善, 処理効率の下限の評価などの問題). 一方, グラフ構造の処理に関してはすでに多くの研究がなされ, 種々の有益な処理手法が知られていると同時に各種の問題の性質もよくわかってきている. そこで, 本稿でも行なったよう

に、グラフ処理研究の成果が利用できないかという観点から形状処理の問題を眺めてみることは、これらの問題へ接近するための一つの強力な指針になると思われる。

謝辞 有益なご討論をいただいた名古屋大学工学部 杉江昇教授、東京大学工学部伊理正夫教授に感謝の意を表す。

参 考 文 献

- 1) Braid, I. C.: The Synthesis of Solids Bounded by Many Faces, *Comm. ACM*, Vol. 18, pp. 209-216 (1975).
- 2) Encarnacao, J. (ed.): *Computer Aided Design*, Lecture Notes in Computer Science, No. 89, p. 459, Springer, Berlin (1980).
- 3) 津吉敏明, 佐藤幸男, 藤田広一: 運動軌跡からの多面体頂点の座標測定, *信学論*, Vol. 62-D, No. 11, pp. 726-733 (1979).
- 4) Asada, M., Yachida, M. and Tsuji, S.: Understanding of Three-Dimensional Motions in Blocks World, *Tech. Rep.*, 81-01, Osaka Univ. (1981).
- 5) Appel, A. and Will, P. M.: Determining the Three-Dimensional Convex Hull of a Polyhedron, *IBM J. Res. Dev.* November, 1976, pp. 590-601.
- 6) Preparata, F. P. and Hong, S. J.: Convex Hulls of Finite Sets of Points in Two and Three Dimensions, *Comm. ACM*, Vol. 20, No. 2, pp. 87-93 (1977).
- 7) Muller, D. E. and Preparata, F. P.: Finding the Intersection of Two Convex Polyhedra, *Theor. Comput. Sci.*, Vol. 7, pp. 217-236 (1978).
- 8) Boyse, J. W.: Interference Detection among Solids and Surfaces, *Comm. ACM*, Vol. 22, No. 1, pp. 3-9 (1979).
- 9) Lozano-Pérez, T. and Wesley, M. A.: An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles, *Comm. ACM*, Vol. 22, No. 10, pp. 560-570 (1979).
- 10) Corneil, D. G. and Kirkpatrick, D. G.: A Theoretical Analysis of Various Heuristics for the Graph Isomorphism Problem, *SIAM J. Comput.*, Vol. 9, pp. 281-297 (1980).
- 11) Weinberg, L.: A Simple and Efficient Algorithm for Determining Isomorphism of Planar Triply Connected Graphs, *IEEE Trans. Circuit Theory*, Vol. CT-13, No. 2, pp. 142-148 (1966).
- 12) Hopcroft, J. E. and Tarjan, R. E.: A $V \log V$ Algorithm for Isomorphism of Triconnected Planar Graphs, *J. Comput. Syst. Sci.*, Vol. 7, pp. 323-331 (1973).
- 13) Hopcroft, J. E. and Wong, J. K.: Linear Time Algorithm for Isomorphism of Planar Graphs, Proc. 6th Annual ACM Symp. on Theory of Computing, pp. 172-184 (1974).
- 14) Barnette, D. and Grünbaum, B.: On Steinitz's Theorem Concerning Convex 3-Polytopes and on Some Properties of Planar Graphs, in *The Many Facets of Graph Theory*, Lecture Notes on Mathematics, No. 110, pp. 27-40 Springer, Berlin (1969).
- 15) 彌永昌吉: 幾何学序説, p. 478, 岩波, 東京 (1968).
- 16) Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *The Design and Analysis of Computer Algorithms*, p. 470, Addison-Wesley, Reading (1974).
- 17) Kimura, F. and Hosaka, M.: Program Package GEOMAP—Reference Manual, preprint (1977).
- 18) リウ, C. L. (伊理正夫, 伊理由美訳): 組合せ数学入門 I, II, p. 435, 共立, 東京 (1972).
- 19) Ikeuchi, K.: Recognition of 3-D Objects Using the Extended Gaussian Image, Proc. 7th Int. Joint Conf. Artif. Intell., pp. 595-600 (1981).
- 20) 杉原厚吉: 頂点辞書を利用した距離画像解析, *情処論*, Vol. 20, No. 1, pp. 39-44 (1979).
- 21) Oshima, M. and Shirai, Y.: A Scene Description Method Using Three-Dimensional Information, *Pattern Recogn.*, Vol. 11, No. 1, pp. 9-17 (1979).
- 22) Garey, M. R. and Johnson, D. S.: *Computers and Intractability*, p. 338, W. H. Freeman and Company, San Francisco (1979).

(昭和57年1月5日受付)
(昭和57年5月19日採録)