

# ワイヤフレームからソリッドへの一変換手法

桃井貞美

福井幸男

長野県情報技術試験場

工業技術院 製品科学研究所

形状モデリングにおけるワイヤフレームモデルは、三次元物体としての情報量が不足しているという理由から、三次元グラフィックのためのモデルとしては利用が制限されてきたが、形状を組み立てていく過程での柔軟性や計算機への負荷が少なく小規模なシステムへも適用可能であるなどの長所を持っている。ワイヤフレームからソリッドモデルへの変換が自動的に行えれば、ワイヤフレームの操作性の良さを生かした三次元モデリングが小規模なシステムでも可能となる。本論文では頂点座標と稜線の接続情報だけを持つワイヤフレームで記述された物体の形状を、稜線からループを探索し、そのループからメビウスの法則を満たす面の組合せを探索することによりソリッドモデルへ変換する手法について述べる。本手法により直線で記述されたワイヤフレームから、穴の開いた形状や曖昧な形状を含め、自動的にソリッドを生成することができた。

## A conversion algorithm from wireframe models to solid models

Sadayoshi MOMOI\*

Yukio FUKUI\*\*

\*Nagano Prefectural Research Institute for Information Technology  
( 1-1-7, Nomizo-nishi, Matsumoto, Nagano 399 JAPAN )

\*\* Industrial Products Research Institute  
( 1-1-4 Higashi, Tsukuba, Ibaraki 305 JAPAN )

A wireframe model is insufficient as a model of 3D shape for computer graphics. It has, however, such advantages, compared with a solid model, as simple data structure, less computational load, and consequently quick response, easy to implement on small computer systems. This paper reports a method that converts wireframes into solids. With this method, advantages of both models are utilized, that is, solid models are manipulated by wireframes with high interactive efficiency. This algorithm is to search for planar loops among possible edges, then to search for faces among loops, obeying Möbius' law. Ambiguous wireframes with holes are successfully converted into solids with this method.

## 1 はじめに

形状モデリングにおけるワイヤフレームモデルは、サーフェスモデル及びソリッドモデルに比較して形状を組み立てていく過程での柔軟性に優れており、また内部のデータ量が少なく計算機への負荷が比較的に軽い。パソコンレベルまで適用範囲が広がっている。しかしその柔軟性の裏返しとして、面情報の不足や三次元的に不合理な構造や曖昧な構造を容認することとなり、そのことがワイヤフレームモデルとソリッドモデルの相互変換を妨げる結果となっている。これらの問題点を解決すればワイヤフレームのすぐれた操作性のもとでソリッドモデルが作成できるようになり、モデリング方法の選択の幅が広がるばかりでなく、簡易なシステムでも三次元ソリッドのモデリングが可能となる。

これまでこの分野においては、G.Markowskyらが直線的な稜線からなる全ての物体に適用可能なアルゴリズムを報告している<sup>[1]</sup>が、処理段階において補助稜線を追加したり、物体を多数の部品化しそれぞれソリッドモデルデータとして蓄えるなど内部データ構造が複雑であるため小規模なシステムには負荷が大きすぎる。またグラフ理論やトポロジー的手法を用いて曲面にも適用可能な方法がいくつか報告されている<sup>[2]~[4]</sup>が、穴の開いた形状や曖昧な形状には適用できないものがほとんどである。

この論文では、頂点の座標と頂点間を結ぶ稜線のデータからなるワイヤフレームモデルを同一平面上の稜線のループ（以下ループという）からなる面の候補（以下面候補という）の組合せを探索することによりソリッドモデルに変換する一手法について報告する。本手法は、穴の開いた形状や曖昧な形状にも適用でき、また処理過程に於ける内部データ構造が簡潔であるため小規模なシステムへも応用可能である。

## 2 基本概念

ワイヤフレームが表現する形状は有限の体積をもつ物体で、多角形の内部で表現される面によって構成されており、稜線と稜線が接するような形状は含まないことを前提条件とする。このような前提条件のもとで1本の稜線は必ず2つの面に接している。本手法は稜線の探索により求めた面の候補について、この規則が成り立つような組合せを仮定を行ないながら探索することによって物体の形状を求めることを最大の特徴としている。求めた組合せはそれぞれ固有の形状を持った物体の候補（以下物体候補という）となるものであるが、この中には面と面が交差（以下交差という）するような立体的な干渉を持った物体候補も含まれている。交差を含んだ物体候補を取り除いた残りの物体候補がソリッドとして出力される。ワイヤフレームに曖昧な形状が含まれている場合は考えられる全ての形状のソリッドが出力される。

ワイヤフレームからソリッドを生成するまでの処理の流れを図1に示す。処理は大きく分けて5つの過程からなる。

- ① 同一平面上の閉じた稜線の連なり（以下ループという）を探索する過程。
- ② 探索したループどうしの包含関係を調べ最小単位のループである面の候補（以下面候補という）を抽出する過程。
- ③ 各稜線に面が2つとなるように面候補の組合せを探索し物体候補を求める過程。
- ④ 物体候補を構成する面の表裏を正しい向きに決定する過程。
- ⑤ 及び各物体候補の交差による立体干渉の有無を調べ正しい形状かどうかを評価する過程。

処理の過程で作成され保持されるデータは、ループを構成する稜線の並び順に関するデータ、ループの平面式に関するデータ及び物体候補を構成する面候補の組合せに関するデータで、容量的に計算機の負担にはならない程度のものである。データ構造が簡単であることもこの方法の特徴である。

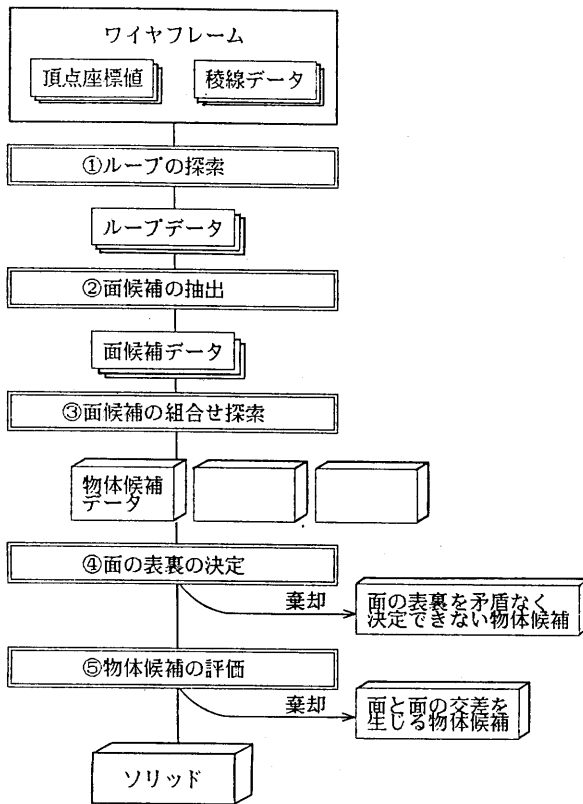


図 1 処理の流れ

### 3 入力データの定義

入力されるデータは頂点の座標値と頂点と頂点を直線で結ぶ稜線の接続状況に関する情報を備えたワイヤフレームとする。また入力データは次のような条件を満足しているものとする。

#### (1) 各稜線が接続する面は二つに限定

各稜線が接する実際の面は二つでなければならない。よって稜線上で二つの物体が接した形状を表現するときは同一座標上に二組の頂点と頂点を結ぶ稜線が定義されていなければならない。

#### (2) 穴の開いた面は多角形分割で記述

最終的に出力するソリッドのデータはウィングドエッジデータ構造を採用しているため面は多角形で表現可能なものに限定される。したがって穴の開いた面は多角形に分割して記述されていなければならない。

#### (3) 孤立した頂点、稜線及び面を持たない

ここでは三次元的に体積をもつ物体のソリッドを生成することを目的としているので、孤立した頂点や稜線など体積を持たない形状は事前に取り除かれていなければならない。

#### (4) 位相的な整合性と自己非干渉性

物体内部を貫くような稜線、稜線の交差、始点と終点が同一頂点である稜線など位相的に不合理なデータは含まれてはならない。

## 4 変換アルゴリズムの概要

### 4.1 ループの探索

最初に行なう処理は同一平面上でエッジが構成するループの探索である。ある頂点を出発点としてその頂点から出ている稜線を一本取り出す。さらにその稜線の先に接続する稜線を一本取り出す。次にこの二本の稜線が決定する平面を調べる。最初の二本の稜線が直線上に並ぶ場合はさらに先の稜線を探索して平面を決定する。この平面上に沿って稜線を探索していく。そして稜線が最初の頂点に戻ってきたところで一連の稜線を一つのループとして記憶する。探索の際に途中の頂点を2回通るようなものは棄却する。また出発点の頂点が異なるだけで実際には同一の稜線で構成されているループは一つにまとめる。この時点で出力されるループは単なる稜線の連なりであって、実際の面でない架空の面に相当するものや同一平面上の複数の面からなるループも含まれている。

#### 4. 2 面候補の抽出

次に各ループどうしの包含関係を調べ、他のループを包含しているものを棄却し、残ったループを面候補とする。図2に面候補の抽出例を示す。ループ①、②及び④は他のループを包含しているので棄却し、ループ③、⑤及び⑥が面候補として取り出される。この段階ではまだ面候補の中に架空の面に相当するものが含まれている。

#### 4. 3 物体候補の探索

この段階では面候補群の中から各稜線に接する面候補が二つとなるような面候補の組合せを探索する。探索はまず各稜線ごとの面候補の数を調べ、1本の稜線に面候補が二つしかないものについてその面候補を真の面として確定する。3つ以上面候補がある稜線について既に確定した面が二つある場合は他の面候補は棄却すべき面として確定する。このように連鎖的に面を確定していき、これ以上確定できないという段階まできたらまだ面の確定していない稜線について仮定を行なう。つまり面候補の中から二つを真の面として選ぶ組合せを考え、その組合せについて順次仮定を行なって面の確定を進めて行く。全ての稜線について不都合なく二つの面が確定できた時点でそれを面の集合体としての物体の候補（以下物体候補という）とする。探索の途中で矛盾が生じた場合にはその仮定を棄却し次の仮定を行ない探索を進める。ここで探索された物体候補としての面の集合体はある一つの形状を特定することができる。対称性が低い形状は架空の面が少なく、形状はこの時点で一つの候補に絞られるが、対称性の高い形状や曖昧な表現を含む形状は複数の候補が取り出される。この段階では物体候補は面の表裏が揃っておらず、また交差が含まれている場合がある。

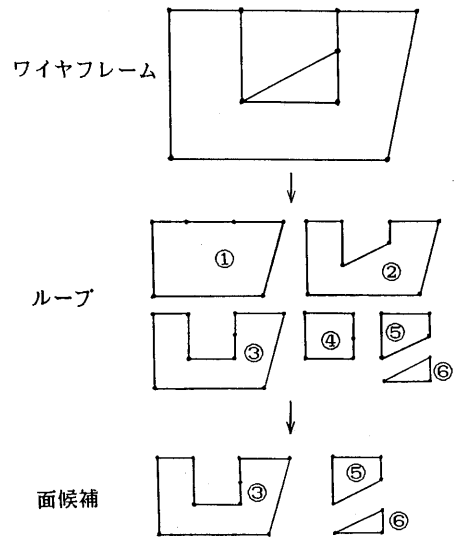


図2 面候補の抽出

#### 4. 4 面の表裏の決定

この段階では物体の外部から見たときに面の表側が見えるように、物体候補を構成する全ての面の表側を決定する。一つの物体においては全ての面がお互いの稜線を接してつながっているため、物体上の一つの面の表を決定できれば他の面も接している稜線上のループの向きだけで連鎖的に決定できる。つまり隣あう面ではその接している稜線でのループの向きが反対向きとなることを利用して決定してゆく。このアルゴリズムでは物体からはるか遠い位置においた視点からみた視線が最初に物体と交わった点を含む面は表が見えるということを利用して最初の面の向きを決定している。交差が存在する物体候補のうち一部のものは連鎖的にループの向きを決定していく過程においてループの向きに不都合を生じる。これを検出することにより交差が存在する物体候補の一部を取り除くことができる。

#### 4. 5 物体候補の評価

面の表裏を矛盾なく決定できた物体候補について、その形状が交差を含まない三次元的に正しいものかどうかを評価する。交差は図3のような三種類に分類できる。

◆(a)は面の内部を稜線が貫いているもので稜線が構成する面と交差を生ずる。

(a)型の検査はまず頂点 $V1$ 及び $V2$ から面 $F$ を含む平面に垂直に下ろした方向に向かうベクトル $v1$ 、 $v2$ を求める。 $v1 \cdot v2 < 0$ であればさらに稜線 $V1V2$ と平面との交点 $P$ を求める。そして $P$ が面 $F$ の内部にあれば交差が存在する。

◆(b)は面 $F1$ を別の面 $F2$ のループが頂点 $V2$ を $F1$ 上において貫いているもので $F1$ と $F2$ が交差している。(b)型の検査はまず面 $F1$ と $F2$ の法線ベクトル $f1$ 、 $f2$ の外積を求める。 $f1 \times f2 \neq 0$ ならば $F2$ の各頂点が $F1$ を含む平面上にあるかどうか調べる。平面上に頂点が存在した場合には、その頂点 $V2$ が $F1$ の内部にあるかどうか調べる。内部にあったならば $V2$ の両側の頂点 $V1$ 、 $V3$ から平面に下した垂線方向のベクトル $v1$ 、 $v3$ を求める。 $v1 \cdot v3 < 0$ ならば(b)型の交差が存在する。

◆(c)は面 $F1$ と面 $F2$ が頂点 $V1$ と $V2$ を共有して交差しているものである。(c)型の検査は二つの面 $F1$ 、 $F2$ について法線ベクトル $f1$ 、 $f2$ を調べ、 $f1 \cdot f2 \neq 0$ ならば共有している頂点(以下共有頂点という)の数を調べる。共有頂点数が二つ以上ならば各共有頂点における交差を調べる。共有頂点における交差は、共有頂点における二つの面の方向を調べることによって検出する<sup>15)</sup>。まずここで $F1$ のループが $V1$ から出て行く稜線方向を $x$ 軸、 $F1$ の法線ベクトル方向を $z$ 軸とし右手系の直交座標系を考える。 $F1$ を含む平面と $F2$ を含む平面の交差線を求め、共有頂点 $V1$ から出て交差線上の向きの異なる単位ベクトル $i1$ 、 $i2$ を求める。 $F1$ のループが $V1$ に入ってくる方向の稜線に向かうベクトル $e$ を求め、 $x$ 軸から $e$ への角度 $\theta0$ を求める。また $x$ 軸から $i1$ 、 $i2$ への角度 $\theta1$ 、 $\theta2$ を求める。 $0 < \theta1 < \theta0$ ならば $i1$ は $V1$ において $F1$ の内部へ向かっている。 $0 < \theta2 < \theta0$ ならば $i2$ は $V1$ において $F1$ の内部へ向かっている。同様に $F2$ に対する $i1$ 、 $i2$ の向きを調べる。もし $i1$ 、 $i2$ のどちらかが $F1$ 、 $F2$ 両方の内部に向かっているとき $F1$ と $F2$ は交差していることがわかる。

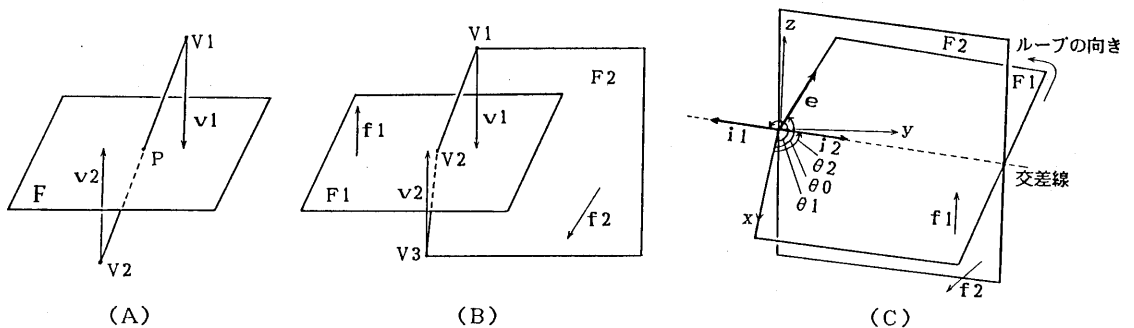


図3 物体候補における面の交差

## 5 実験例

本手法によってワイヤフレームからソリッドを作成した例を図4～図7に示す。

図4は六角形の回転体の例である。(a)がワイヤフレームで(b)が出力されるソリッドである。(d)は3つの物体候補のうちの一つであるが(e)のカットモデルを見ると内部に(C)型の交差が生じているのがわかる。ソリッドは(c)のカットモデルを見ても内部に交差は見られず、正しい形状であることがわかる。回転体では経験的に回転方向の角数が偶数の場合に交差を含んだ物体候補が生じる。また回転方向の角数が6の倍数以外では交差を含む物体候補は面の表裏の決定において不都合が生じ、交差を調べる前に棄却することができる。

図5は穴の開いた形状の例である。(a)がワイヤフレームで(b)が出力されるソリッド、(c)がそのカットモデルである。3つ出力される物体候補のうち2つは(d)のような形状をしており、(e)のカットモデルが示すように内部に(A)型の交差が生じている。

図6は(a)に示す曖昧な形状のワイヤフレームの変換例である。穴の位置が異なる(b)、(c)、(d)の3つのソリッドが出力される。(e)はそのうち1つのカットモデルである。

図7は(a)のワイヤフレームの変換例である。この例では1つの稜線に最高5つ、最低でも3つの面候補を持っている。探索される物体候補は3つであるが、うち2つは(C)型の交差を含んでおり、出力されるソリッドは(b)に示す1つである。(d)に交差を含む物体候補の1つを、また(e)にそのカットモデルを示す。

## 6 まとめ

各稜線に接続する面が2つになるような面候補の組合せを探索することにより、ワイヤフレームをソリッドに変換する一手法について述べた。本手法により、穴の開いた形状も処理できるので、ワイヤフレームの柔軟性を十分生かすことができる。またワイヤフレームの中に曖昧な形状が含まれていてもすべてソリッドへ変換してくれるので出力されたソリッドを見れば曖昧な部分を容易に知ることができる。今回ワイヤフレームの作成からソリッドを生成するまでの作業をすべてパソコン上で実現することができた。今後は処理速度を改善するなど、より実用的なものへ改良して行きたいと考えている。

### <参考文献>

- [1]G.Markowskey and M.A.Wesley:Fleshing Out Wire Frames,IBM J.RES.Develop., Vol.24,No.5,pp.582-597(1980)
- [2]S.M.Courter and J.A.Brewer:Automated Conversion of Curvilinear Wire-frame Models to Surface Boundary Models; A Topological Approach,SIGGRAPH,Vol.20, No.4,pp.171-178(1986)
- [3]清水,矢島,上西:ワイヤフレームからソリッドへの半自動変換,情報処理学会第29回全国大会,pp.1765-1766(1984)
- [4]二上,長田,小笠原,小堀:複合幾何モデルの開発,第3回NICOGRAPH論文集,pp.20-28(1987)
- [5]福井:多面体頂点での物体方向検出方法,情報処理学会第30回全国大会,pp.1997-1998(1985)

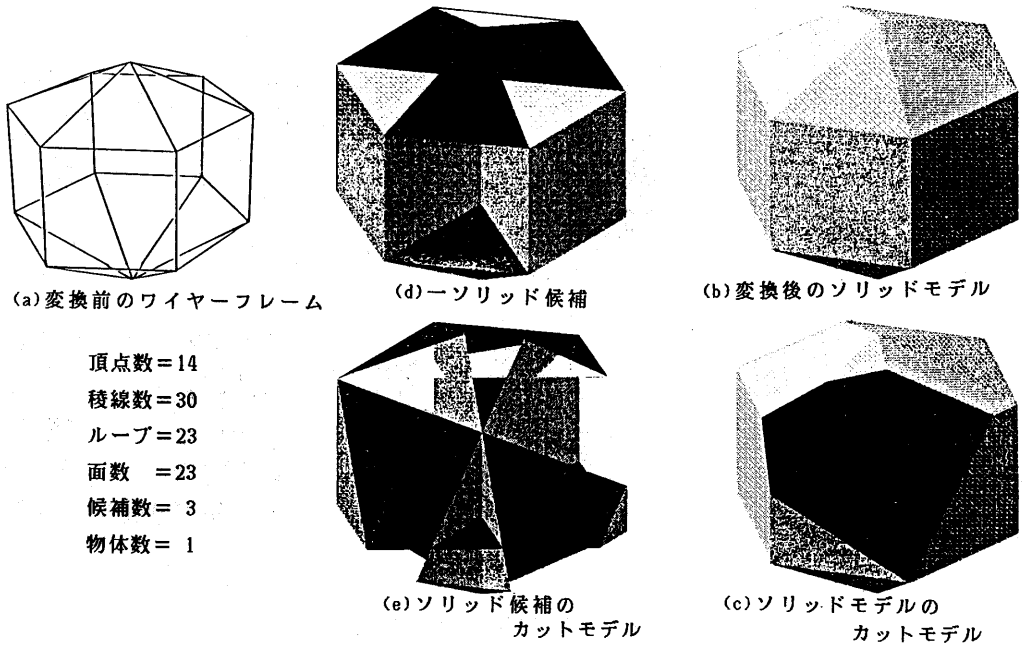


図4 穴のない回転体の変換例

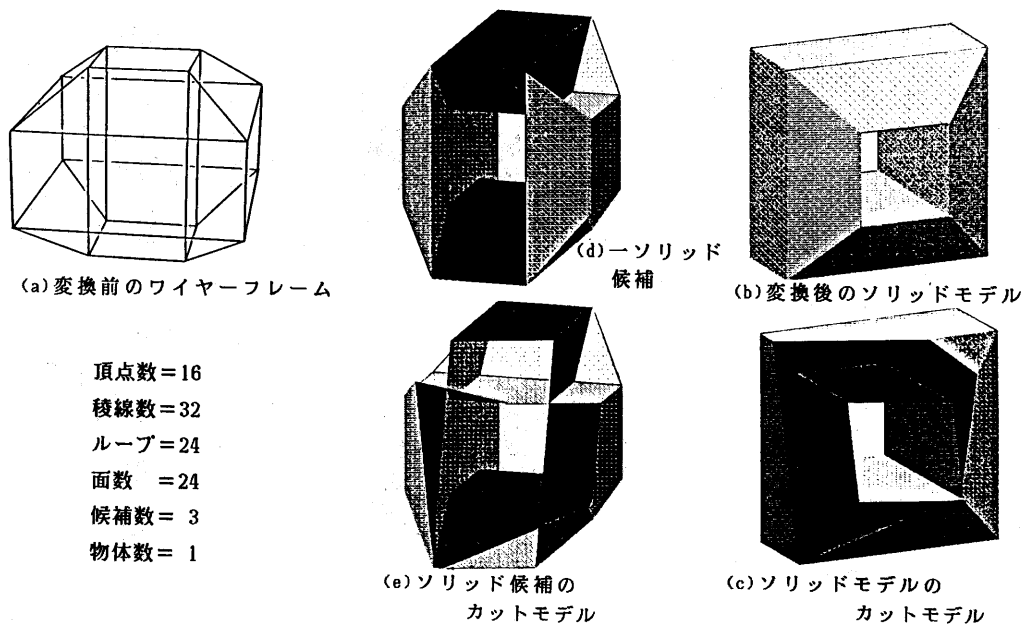
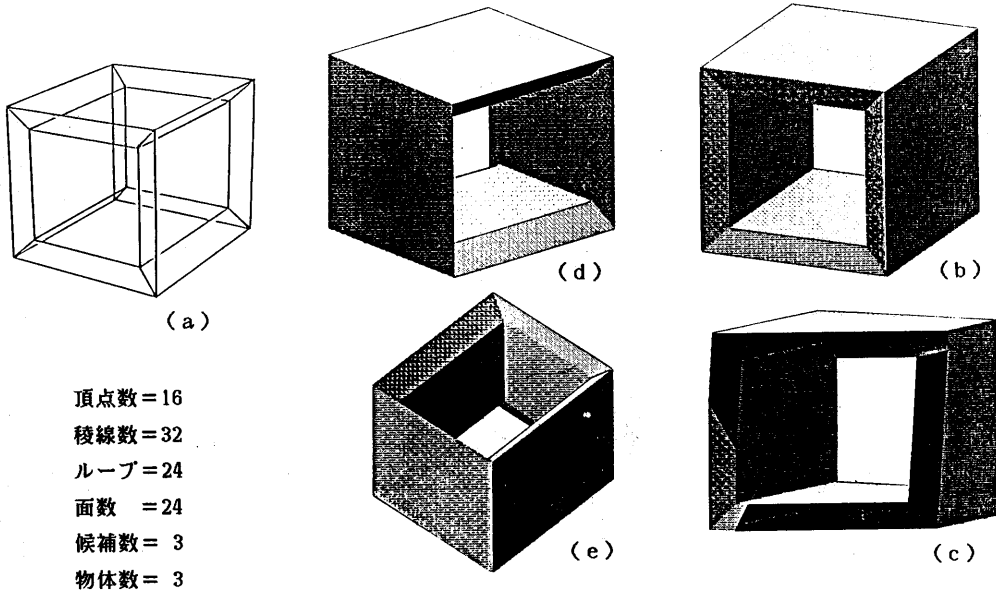
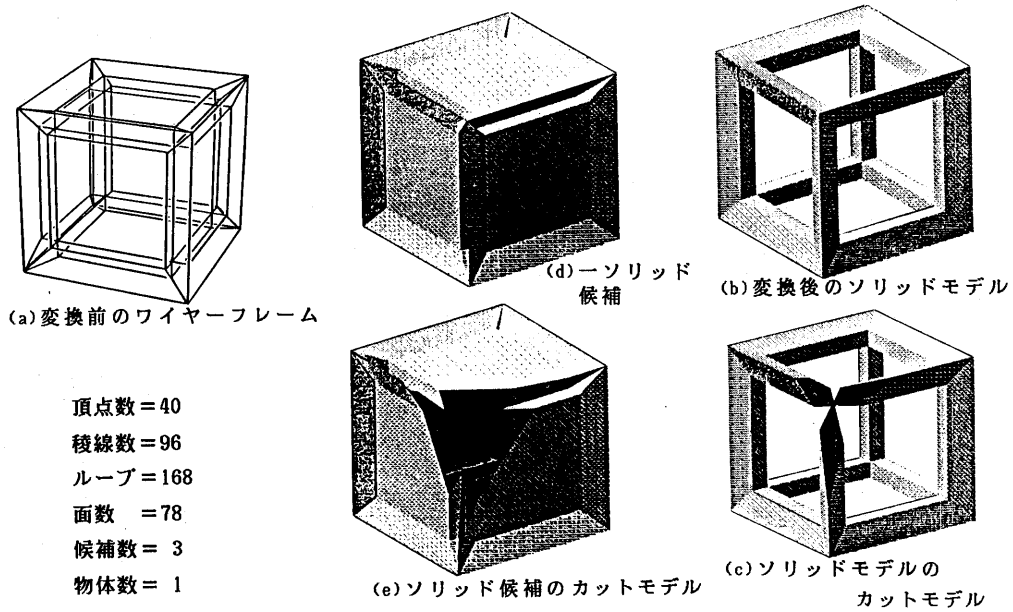


図5 面候補と稜線が交差する場合の変換例



頂点数 = 16  
 稜線数 = 32  
 ループ = 24  
 面数 = 24  
 候補数 = 3  
 物体数 = 3

図6 曖昧な形状の変換例  
 (b) (d) (e) 変換後の各ソリッドモデル



頂点数 = 40  
 稜線数 = 96  
 ループ = 168  
 面数 = 78  
 候補数 = 3  
 物体数 = 1

図7 多数の面候補が生じる場合の変換例