

ソリッドモデラ (Kernel-Solid) における立体集合演算の一手法

長田義弘 小堀研一 西岡郁夫
シャープ株式会社 技術本部 コンピュータシステム研究所

境界表現 (Boundary-Representation) による多面体モデルにおける立体集合演算を高速に行う手法について報告する。

一般的に、ソリッドモデルの和、差、積を求める立体集合演算は遅いとされている。しかし対話的にソリッド形状を定義するには、立体集合演算を高速に行う必要がある。そこで幾何学的な情報となる実数計算を出来るだけ減らし、位相的な情報を活用することにより計算速度の向上を実現した。具体的な方法として、最初に効率良く二面の交線を求め、交線と立体との関係をリストとして持つ。そして、このリスト情報を主に用いて稜線のトリム、面ループの接続等、立体集合演算の処理を行う。リストを用いることにより、計算速度の向上のみならず、自由曲面処理への移行も容易となる。また、計算誤差の影響も減少した。

この手法を現在開発中のソリッドモデラ (Kernel-Solid) に組み込み、対話的な速度による立体集合演算が可能となった。

A Method of Set Operations in Solid Modeller (Kernel - Solid)

Yoshihiro Nagata, Ken-ichi Kobori and Ikuo Nishioka
Computer Systems Laboratories, Engineering Center, Sharp Corporation
2613-1, Ichinomoto, Tenri, Nara, 632 Japan

This paper describes a method of fast set operations for polyhedra based on Boundary-Representation.

In general, it requires much time to get union, difference and product sets of solid objects. So, it is important to realize fast set operations for defining solid objects in order to be suited to an interactive environment. To reduce processing time, the method presented here reduces the use of geometric information by efficient use of topological information. After finding intersection lines of faces, the relationship between the intersection lines and the object is stored as a list. Most of set operation process can be done using only this relationship. As a result, not only can the method offers speed, but it could be adapted to nonplanar surfaces. The algorithm has been implemented on our solid modeller (Kernel - Solid), and we have put it to practical use.

1. はじめに

設計、生産業務の省力化、自動化を目的として、現在、多くのCAD/CAMシステムが開発され、実用化されている。このうち三次元形状定義システムには、データ構造からワイヤフレームモデル、サーフェスモデル、ソリッドモデルの3種類のものがある。NC加工などCAD/CAMの適用分野の大半は、サーフェスモデルを用いて処理することが出来るが、NC加工シミュレーションや形状の体積、重心の計算などには、ソリッドモデルの利用が必要となる。しかしソリッドモデルは、モデルの情報量が多くデータ構造も複雑なため、システムの信頼性が低い、処理速度が遅い等の理由で、その有用性にもかかわらず他のモデルに比べて実用化が進んでいない。このソリッドモデルの基本となるのが立体集合演算であり、これを高速に実行することで対話的な形状定義が可能となる。

ソリッドモデルのデータ構造には、「TIPS」¹⁾が用いているCSG表現(Constructive Solid Geometry)と、「Build」²⁾が用いている境界表現(Boundary Representation)とがある。本手法では境界表現を用いた多面体モデルで立体集合演算を行う。

関連した研究として、千代倉³⁾らによる「DESIGN BASE」の開発がある。「DESIGN BASE」での立体集合演算は、基本操作を用いてデータベースの操作を行うため、交線データの重複があるが、本手法では、直接データベースを操作し、集合演算の結果となる新たな立体を発生している。

以下、立体集合演算の処理方法について報告する。

2. 立体集合演算の概要

本手法では、2つの立体との和を次の4ステップで求める。(図1)

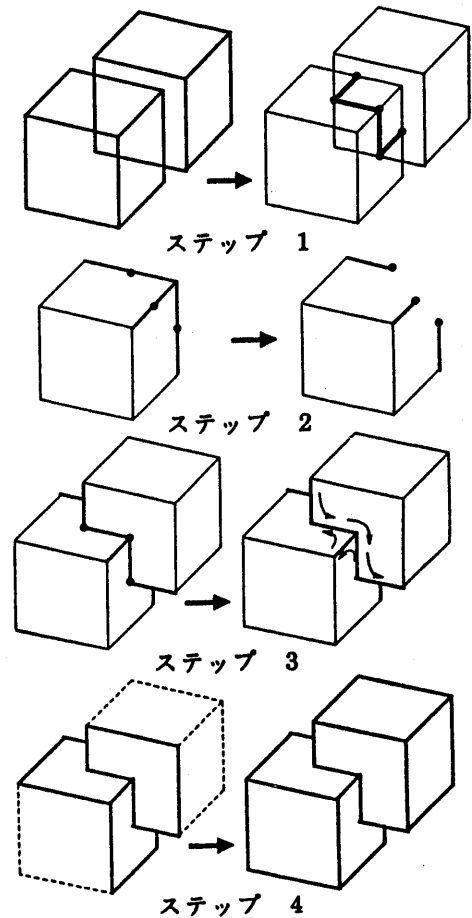


図1 立体集合演算の概要

ステップ1) 交線計算

2つの立体間で交差する2面の交線を求める。こ

ここで、交線の端点を交点とする。関係のある頂点、稜線からこの交点へのポイントリスト⁴⁾を作成する。

ステップ2) 稜線のトリム

稜線上の交点をリストを用いて集め、稜線方向にソートした後、位相的に新たな稜線となる部分を求める。

ステップ3) 局所的な面ループの連結

各交点に接している交線、稜線から、この交点を通る面ループを集め、面ループを連結する。

ステップ4) 立体の生成

ステップ3の連結情報と、もとの立体の面ループから新たな面ループを完成する。またこの面ループに隣接するすべての面を加えることにより新たな立体が完成する。また、立体の反転を組み合わせるにより、差、積も求まる。⁵⁾ 次章以下にアルゴリズムの詳細を述べる。

3. 交線計算 (ステップ1)

立体Aの面FAと立体Bの面FBの交線を次の方法で求める。(図2)

- (1) 面を含む直方体を用いた簡単なラフチェックにより、FA,FBの組のうち交差しないものを除く。⁶⁾
- (2) FBを無限平面と考え、FAとの交線を求める。
- (3) 無限平面FAとFBとの交線を求める。
- (4) (2)と(3)の交線の2次元的な積集合が交線となる。

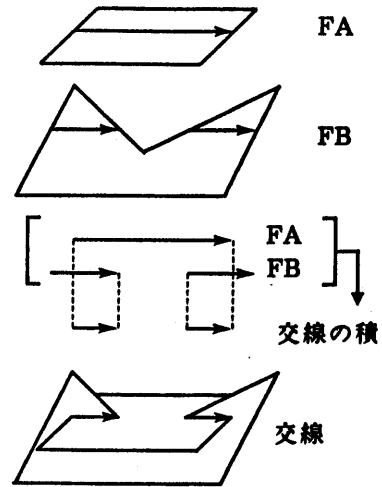


図2 平面FA,FBの交線

そこで、FAと無限平面FBの交線を求める手法について述べる。

まずFAのすべての稜線とFBとの交線を求める。FAのある稜線の始点をVB、終点をVTとする。この稜線がFBと交差するか調べるために、FBの平面の方程式 f_s に頂点座標を代入する。(図3)

$f_s(VB) \cdot f_s(VT) > 0$ のときは、VB,VTが平面に対して同じ側にあるので、この稜線には交点がないと判定できる。 $f_s(VB) \cdot f_s(VT) < 0$ のとき、この稜線には交点があると判定できる。また、 $f_s(V)$ の大きさは平面と頂点との距離に比例するので、この比率でVB,VTを内分する点が交点になる。ここで、交線の方角をFA,FBの法線ベクトルの外積の方角に同じとすると、稜線の向きとFBの法線ベクトルとの関係から、この交点が交線の始点、終点のどちらか判断出来る。(図4)

$f_s(V) = 0$ の場合は、頂点が面に含まれており、他の頂点の $f_s(V)$ の値により、始点、終点、または交点にならない点かを判断する。(図5)

$f_s(VB) = 0, f_s(VT) = 0$ となった場合は、稜線が平

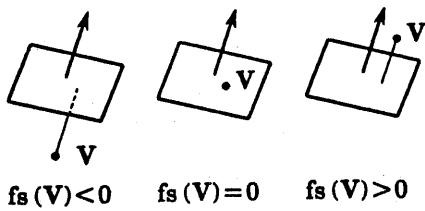


図3 頂点Vと平面の位置関係

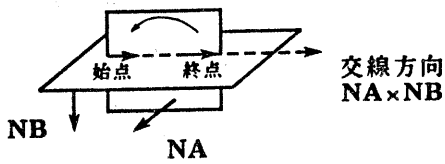


図4 平面FA,FBの法線ベクトル NA,NBと交線

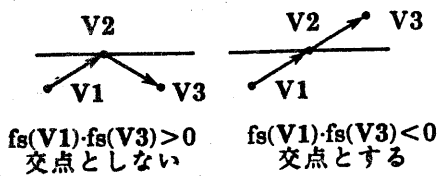


図5 $fs(V_2)=0$ の場合の処理

面に含まれているので、ここではVB,VTとも交点としておき、後で処理する。

FAの全部の稜線と無限平面FBとの交点を求めた後、これを交線方向にソートし、始点と終点を組み合わせるにより交線が求まる。

同様に、FBと無限平面FAとの交線を求め、2つの交線の積をとることにより、FAとFBの交線が求まる。もし積の結果、長さが0となった場合は2面が図6の様に接する場合なので、交線としない。

次に稜線が平面に含まれている場合の処理について述べる。図7の場合(FA,F1),(FA,F2)の2本の交線が考えられるが、面と稜線の交差⁵⁾を調べることにより、F2は立体の内部と判断され、(FA,F1)

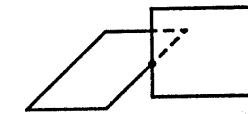


図6 長さが0の交線

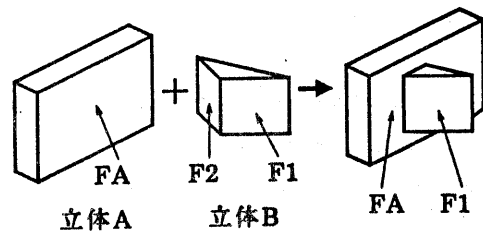


図7 稜線が面に含まれる場合

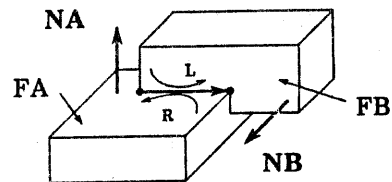


図8 交線方向と面ループの方向

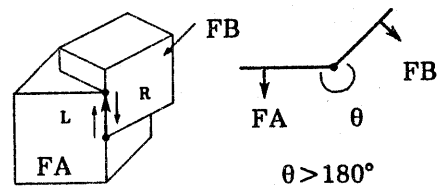


図9 交線方向が逆になる場合

の交線だけをデータに入れることが出来る。

この段階で交線にWinged-Edge⁶⁾に必要な、面ループの情報を次の方法で得る。

交線の方向がFAとFBの法線ベクトルの外積で

あるから、一意に左ループがFB、右ループがFAとなる。(図8) 但し図9の様に2立体の稜線が一致する場合、面の角度が 180° 以上となることもあり、このとき外積ベクトルの方向が反対になるので交線の方向も反対になり、左ループがFA、右ループがFBとなる。また、面の角度が 180° のとき外積ベクトルの大きさは0となるが、これはFA,FBが同一平面上の場合であり、ここでは交線を求めずに、ステップ3の面ループの連結等の際、FA,FBを同一ループと見なすことにより、処理出来る。

以下の処理の高速化を図るため、このステップでは、次のデータを作成している。

(1) 頂点フラグ

もとの立体の頂点と一致する交点があるとき、この頂点から交点へのポイントを持つ。一致する交点がないときは、0とする。

(2) 稜線リスト

稜線から、この稜線上にある交点へのポイントの集合を持つ。

(3) 交点リスト

交点から、この交点に接続している交線へのポイントの集合を持つ。

(4) 交線データ

交線の始点、終点となる交点へのポイントと左、右ループの面へのポイントを持つ。

4. 稜線のトリム(ステップ2)

ステップ1で作成した稜線リストを用いて、稜線上のすべての交点について次の処理を行う。(図10)

まず、交点に接続している交線の左、右ループのうち、稜線のループと同じループの数と向きを

調べることにより、この交点が新たな稜線の始点、終点のどちらになるか判断出来る。そして、稜線方向に交点をソートし、始点、終点データをもとに、稜線がトリム出来る。トリムされた稜線を交線と同じ形でデータに加え、以下のステップで利用する。

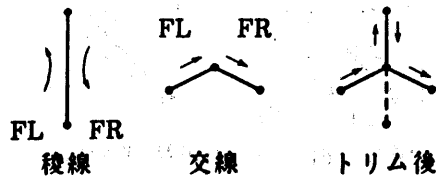


図10 稜線のトリム

5. 局所的な面ループの連結(ステップ3)

これまでの処理で、立体集合演算によって新たに発生する稜線はすべて作成出来た。ここでは、次の方法により交点における面ループの連結を行う。

(1) 交点に接続している交線のループのうち、この交点に入るループと出るループの中から、同じものを任意に連結する。

(2) (1)で連結出来ないループがある場合、この交点に対応した元の立体の頂点に接続している稜線を用いてループを連結する。

(2)の処理は、頂点上にある交点(図11)や、ステップ2で稜線をトリムした後に残った頂点(図12)等で行われる。

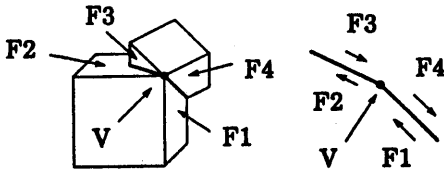


図11 面ループの連結(その1)

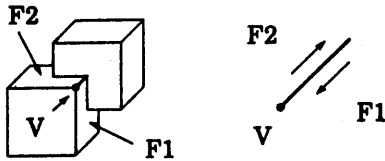


図12 面ループの連結(その2)

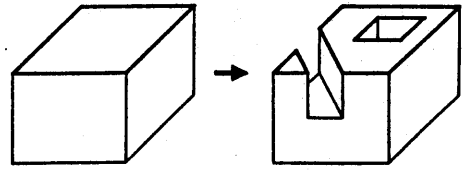


図13 穴あき面

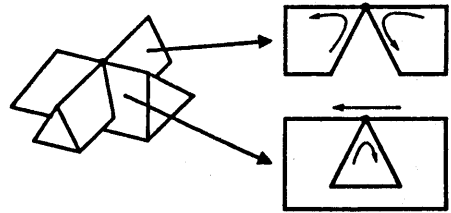


図14 面ループの分割

6. 立体の生成(ステップ4)

各頂点における面ループの連結データと元の立体の面ループの情報から、新たな面ループを完成する。また、この面ループに隣接する面ループをすべて取りだすことにより新たな立体が構成出来る。

7. 穴あき面の処理

ここでは、図13のような穴があいた形状の処理について述べる。穴となるための必要条件は、位相的な情報として同じ番号を持つ面ループが複数存在することである。また、親ループは左回り、子ループは右回りであるから、幾何データを用いてループの方向と包含関係調べるにより、穴あき面かどうか判断出来る。図14の場合、ステップ3では任意にループを連結しているため、出来た面ループが同じ頂点を2度通る場合がある。

このときは、この頂点で面ループを分割し、穴あき面の処理を行う必要がある。

8. 実行例

表1は、本手法を用いて立体集合演算を行った例である。計算時間は、DEC社のVAX11/780で測定した。使用言語はFORTRAN-77である。

9. おわりに

表1の測定結果から、本手法により対話的な速度で立体集合演算を行えることが明らかになった。このアルゴリズムは、現在開発中のソリッドモテラ(Kernel-Solid)に組み込み実現された。データベースを直接操作するため、処理は高速となったが、プログラムはかなり煩雑になった。また、このソリッドモテラでは、立体集合演算の結果出来たデータが正しいWinged-Edge構造になっている

かどうかを、自動的に判断するプログラムを組み込んでいる。これにより、データベースの信頼性が高まった。

今後は、NC加工、有限要素法等で必要とされる自由曲面を持った立体の集合演算に、本手法を展開することが課題となる。

10. 参考文献

- 1) 沖野：自動設計の方法論，養賢堂，p.19 (1982).
- 2) I.C.Braid, C.A.Lang : Computer-Aided Design of Mechanical Components with Volume Building Bricks, Proc. of PROLAMAT'73 (1973).
- 3) 島谷, 佐藤, 千代倉：逆操作を持った立体集合演算, グラフィクスとCAD 16-4, 情報処理学会 (Mar. 1985).
- 4) 野崎, 野下訳：アルゴリズムの設計と解析 I, サイエンス社, p.38.(1977).
- 5) 千代倉：ソリッドモデリング, 工業調査会, p.48 (1985).
- 6) B.G. Baumgart : "Geometric Modelling for Computer Vision," Report No. AIM-249, STAN-CS-74-463, Stanford Artificial Intelligence Laboratory, Stanford University (Oct. 1974)

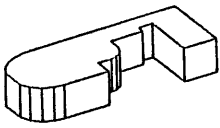
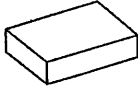
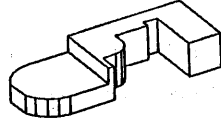
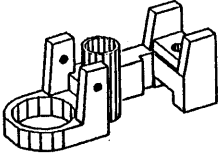
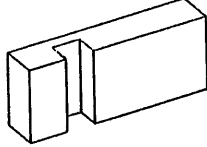
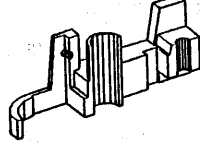


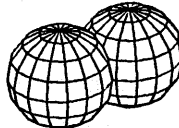
立体A	立体B	演算	演算結果	計算時間 (秒)
 26面	 6面	$A - B$	 28面	1.3
 150面	 10面	$A \times B$	 83面	4.6
 128面	 128面	$A + B$	 248面	7.3

表1 立体集合演算の実行例