

距離尺度に基づく形状表現法

徳 増 真 司^{†1} 野 中 士 郎^{†2} 仁 尾 都^{†3}
原 島 一 郎^{†2} 松 本 輝 夫^{†4}

機械系 CAD/CAM やコンピュータ・グラフィックスに適した図形処理法として、距離尺度に基づく統一的な形状表現法を提案した。本手法は、空間上の任意の点から対象とする形状の境界に至る最短距離に関するデータを生成する手順（これを“相対表現手順”と呼ぶ）と、対象とする形状を同一視して表現するものである。本手法では、通常必要とされる主要な図形処理を、相対表現手順という概念のもとで統一的に行っており、以下の特徴がある。(1) 広範な形状の定義、加工・編集と表示が可能、すなわち取り扱える形状の範囲が広い。(2) 可能なあらゆるデータの組合せに対して処理が破綻をきたさず有効な結果を保証できる、すなわち処理系が堅固である。(3) 目的に応じた表現精度の制御が容易である。(4) 図形処理法および、これにかかわる形状データの形式が簡明でデータ量が少なく、オブジェクト指向に適した形で処理系が単純化される。また、本手法に基づくモデラーのプロトタイプを作成し、簡単な形状表現の実験を行った結果、形状の表示処理速度に問題はあつたものの、従来提案されている図形処理法では実現が困難であつた形状編集操作等が確認でき、従来の手法を補完する形状モデリングの有力な一手法となりうる見通しを得た。

Geometric Modeling Method Based on Distance Measure

SHINJI TOKUMASU,^{†1} SHIRO NONAKA,^{†2} MISATO NIO,^{†3}
ICHIRO HARASHIMA^{†2} and TERUO MATSUMOTO^{†4}

The object-oriented geometric modeling method for mechanical CAD/CAM and Computer Graphics, that enables the uniform representation of object shape, is presented. In this method, shape of objects is regarded as the procedure which measures the distance between any given point and the boundary of objects. And principal shape manipulations are constructed uniformly by this idea of distance measure procedure. The characteristics of this method are the followings. (1) The range of representable shape is wide. (2) The modeling system can be solid. (3) The control of representation accuracy is easy. (4) The modeling system is simple and suitable for object-oriented form. Examination of shape representation is done, using the prototyped geometric modelers based on this method. The result of the examination shows that this modeling method has the functions, such as moving objects representation, which are difficult to be realized by usual methods.

1. はじめに

現状の機械系 CAD は、2 次元、3 次元の形状モデラーを中心に、製図、各種解析あるいは NC テープ作

成支援等、主として機械形状の寸法が決定した後の設計支援を行っている。しかし、今後益々必要となってくる設計の初期段階の支援を考える場合、設計が不完全であいまいな設計要求から次第に完全で厳密な設計解へと進むことから、形状モデラーに対しても、それに対応できる図形処理の堅固さや柔軟性が要求されている。堅固さとは、形状の生成や加工・操作の処理を繰り返す際、あるいは複雑な操作の処理を加えた場合に、モデラー自身がハングアップしないことであり、柔軟性とは、詳細で、かつ固定した精度でモデルを扱うのではなく、上記のような設計の各段階において、自在に詳細さや精度を制御して作業が進められることである。本論文では、上記の問題をカバーするために、新たに距離尺度に基づく形状表現法を提案し、その機

†1 神奈川県立理工学情報工学科

Department of Information and Computer Sciences,
Kanagawa Institute of Technology

†2 株式会社日立製作所日立研究所情報制御第2研究部

The 2nd Department of Systems Research, Hitachi Research Laboratory, Hitachi Ltd.

†3 株式会社日立製作所システム事業部都市開発・住宅システム部
Home Systems Department, Systems Engineering Division, Hitachi Ltd.

†4 株式会社日立情報ネットワークシステムエンジニアリング本部
System Engineering Division, Hitachi Information Networks Ltd.

能的な側面から、その方式の実現性と有効性に関する検討を行った結果を報告する。なお、本論文はすでにシンポジウムで発表した論文を整理し、さらに3次元モデルの検証例を加えて拡充したものである^{4)~7)}。

2. 従来の形状表現法の問題点

機械系CADやCGのための形状表現法として必要要件は、目的に応じて多少異なるが、共通するものとして、以下の5項目があげられる。

- ① 形状の表現力が豊かで、形状に係る特性値（マスコプロパティ等）の出力が可能であること。
- ② 処理系が堅固であること。
- ③ 目的に応じた表現精度の制御が可能であること。
- ④ 形状データの表現形式が簡明であり、データ量が少ないこと。
- ⑤ 処理速度が高速であること。

以上の項目に対し、従来から用いられている主な形状表現法（B-rep法、CSG法、Oct-tree法）を検討すると以下のことがいえる。

(1) B-rep (Boundary Representation) 法¹⁾

これは、形状を位相データおよび幾何データの組合せで構造的に表現し、これを基に形状間の集合演算が可能とするものである。この手法の大きな特徴は自由曲面を含む広範な形状表現能力にあり、①に対してはよいが、③は一般に困難であり、数値計算誤差との関連や例外処理の多さから②や④に対しても問題がある。また、形状の加工・編集に時間を要し、⑤に対しても問題がある。

(2) CSG (Constructive Solid Geometry) 法²⁾

これは、形状を基本形状の集合演算の過程を木構造で示すことにより表現するものである。なお、結果として得られた形状の特性量計算や表示には、別途個別的手法が必要である。この手法の大きな特徴は、簡明な形状表現法で、データ量が少ない点にあるが、表現できる形状がプリミティブの集合演算で得られるものに限られており、①に弱点を持っている。また、形状生成に関しては高速処理が実現できるが、生成された形状を応用（たとえば表示）する場合に時間を要し、⑤に対しても問題がある。

(3) Oct-tree 法³⁾

これは、形状に対し、ある単位空間（通常立方体）が形状に完全に含まれているか、含まれていないか、あるいは部分的に含まれているかを判断し、部分的に含まれている場合には、その単位空間を分割して新たな単位空間とし、同様の判定を再帰的に繰り返すもので、8分木（Oct-tree）表現となる。この手法も簡明

で高速な図形処理法であり、②、③および⑤に対してよいが、①に関し、面や線のように、次元の縮退した形状の処理は不得手である。また、実用精度が必要な加工情報を作成しようとする、④のデータ量が膨大なものとなるという弱点を持っている。

以上述べたように、広範囲の用途に適用でき、高速な処理を実現でき、堅固で簡明な形状表現法としては、従来の提案されている手法はいずれも不十分である。そこで、従来の手法を補完し、これらの問題点を解決する新たな形状表現法を提案する。

3. 距離尺度モデルの基本的な考え方

3.1 手法の概要

本手法では、空間上の任意の点に対し、対象とする形状との最短距離に関する相対位置データを求める手順・関数（これを「相対表現手順」と呼ぶ）を考え、この相対表現手順を形状の計算機内部表現とする。すなわち、モデルの定義空間内の1点（この点を参照点と呼ぶ）を与えたとき、この点が対象とするオブジェクトの内部の点であるか、外部の点であるかにかかわらず、その点からオブジェクトの境界に至る距離を含めて、その点とオブジェクトの関係を示すデータ（前記相対位置データ）を求める手順と形状を同一視し、形状を手順として一元的に表現するものである。相対表現手順の具体的な形式については次章で述べるが、種々の形状を規定する相対表現手順を考えた場合、手順の内容は各々異なるものの、それぞれの手順へのアクセスの形式および生成される相対位置データの形式は同じとなる。

この形状の規定方法、すなわち相対表現手順の生成方法は、以下の2種類である。

- (1) 基本形状定義：形状を規定する一連のデータを用いて、相対表現手順を生成するものである。たとえば球の場合には、中心位置と半径を与えて、任意の点からの相対位置データを計算する手順を生成する。
- (2) 加工・編集：生成済の形状の相対表現手順を利用する形で、形状または形状間の加工・編集操作によりつくられた形状に対し、任意の点からの相対位置データを計算する手順を生成する。たとえば形状Aのオフセットを行う場合、オフセット量と形状Aの相対表現手順を与えて、手順を生成する。

さらに、形状にかかわる諸特性（マスコプロパティや概形等）も、形状の相対表現手順自身を用いて計算することができる。このように、通常必要とされる主要な形状処理は、相対表現手順という唯一の概念のもとで統一的に行う。

3.2 相対位置データ

空間上の任意の点と対象とする形状との相対位置データとしては、以下の2種類を考える。なお、説明の都合上、以降図は2次元のものを用いる。

(1) 第1種相対位置データ

図1(a)に示すように、空間上の点Pが、形状Gの内にあるか外にあるかを示す内外判定データI/O（内にあるときはIN、外にあるときはOUT）と、点Pから形状Gの境界への最短距離 ρ と、点Pに対して最短距離にある形状Gの境界上の点Qからなるデータの組であり、以下の形式で表現できる。

$$(I/O, \rho, Q)$$

(2) 第2種相対位置データ

図1(b)に示すように、空間上の点Pが、形状Gの内にあるか外にあるかを示す内外判定データI/O（内：IN、外：OUT）と、点Pから形状Gの境界への最短距離下界 ρ^- と、点Pから形状Gの境界への最短距離上界 ρ^+ と、点Pから最短距離上界 ρ^+ を与える形状Gの境界上の点Qからなるデータの組で表現する。

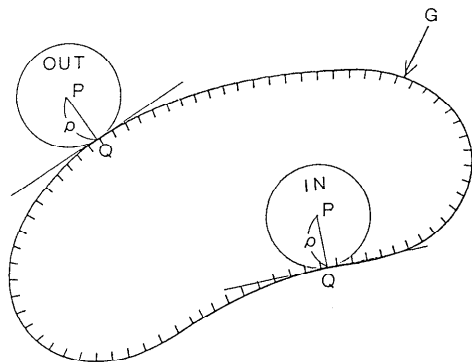
相対位置データは以下の形式で表現する。

$$(I/O, \rho^-, \rho^+, Q)$$

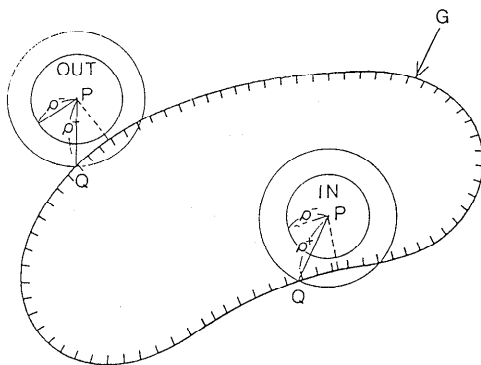
ところで、第1種相対位置データは、 $\rho^- = \rho^+ = \rho$ とすることにより、第2種相対位置データに含ませることができるので、第1種相対位置データの表現も第2種の表現形式を用いる。なお、第1種相対位置データを生成する相対表現手順のみで形状モデラーを構築することも可能であるが、正確な最短距離 ρ を算出するために多量の処理を必要とする場合があり、また第2種相対位置データで済む処理も多い。そこで、第2種相対位置データの最短距離下界 ρ^- および最短距離上界 ρ^+ を必要に応じて最短距離 ρ に近づける手続き（この操作の極限は第1種相対位置データを求めることに相当する）を別途用意し、通常は第1種および第2種の相対位置データを混在して利用し、必要に応じて上記手続きを使用する形態とする。すなわち、この手続きは例外的な処理であり、後述の4.2節(6)でのみ現れる。またこの相対位置データの形式はOct-treeのように、座標系に依存する形式をとっていないので、オブジェクトの平行移動や回転にも適合できる。これにより、柔軟性の高い形状モデラーの構築が可能となる。

4. モデラーの概要

形状モデラーの基本構成を図2に示す。ここで、オ



(a) 第1種相対位置データ
(内外判定データ、最短距離 ρ 、境界上の最短距離点Q)



(b) 第2種相対位置データ

図1 相対位置データ

Fig.1 Relative position data.

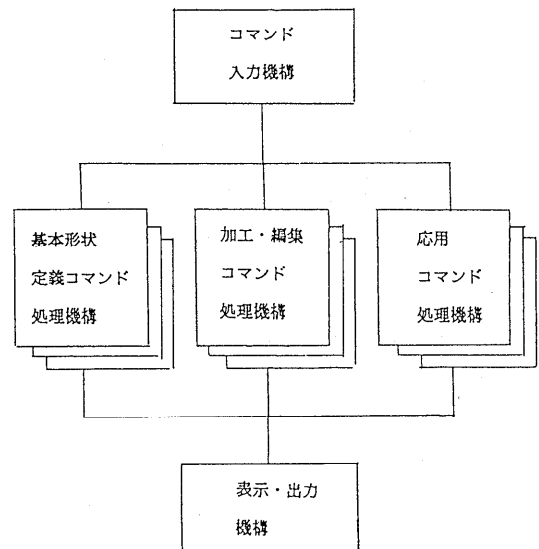


図2 モデラーの基本構成

Fig.2 Basic construction of geometric modeler.

プロジェクト指向の考え方を取り入れ、各コマンドはクラスとして表現する。基本形状定義コマンドは形状を規定する一連のデータにより、形状をインスタンスとして生成する。各インスタンス（形状）のメソッドが相対表現手順である。加工・編集コマンドも同様に操作データおよび形状（インスタンス）を用い、加工・編集された形状をインスタンスとして生成する。応用コマンドはパラメータおよび形状を用い、形状にかかわる諸特性を出力する。なお、生成された形状インスタンスはその種類にかかわらず、点の座標値をメッセージとして起動し、メソッドにより相対位置データを生成する。また、加工・編集および応用コマンドにおいて、形状インスタンスが入力されるが、その利用も形状の種類を意識しない形態で構成する。このように、全体をオブジェクト指向の形で構成できるので、基本形状定義コマンド、加工・編集コマンドや応用コマンドの追加や修正等が容易であり、モデラーとしての拡張性、柔軟性が高い。以下、コマンドの概要を説明する。

4.1 基本形状定義コマンド

これは、形状を規定する一連のデータに基づき、相対表現手順を生成する手段である。なお、相対表現手順の生成方法は以下の形式による。

(定義関数名, 形状を規定するデータ)

ここで対象とする形状は、それが埋め込まれるオブジェクトの定義空間の次元にのみ依存し、原理的にはオブジェクト自身の次元には依存しない。この理由は本モデルが“点”の概念しか持たないためであり、辺や面の概念を持たないからである。本モデラーでは、3.2 節に述べた任意の点 P に関する相対位置データを求める手順が存在する図形ならば取扱いが可能である。取扱い可能なオブジェクトの例を下記に記す。

- (1) 0次元形状：点
- (2) 1次元形状：直線，円，解析曲線
- (3) 2次元形状：矩形，三角形，円板
- (4) 3次元形状：直方体，球，円柱，トーラス

たとえば、円板の相対表現手順を生成する場合を考え、

$G := (\text{円板, 中心 } O, \text{半径 } r)$

とするとき、中心 O 、半径 r の円板の相対表現手順名（円板形状名）は G である。なお、生成される相対表現手順の内容を図3に示す。ここで、オブジェクト G の内部表現は上記のステートメント自身であり、それは LISP の一命令として生成される。したがって、このモデルの表現形式は、どちらかといえば CSG 形式に近いけれども、はるかに簡明である。また、B-rep

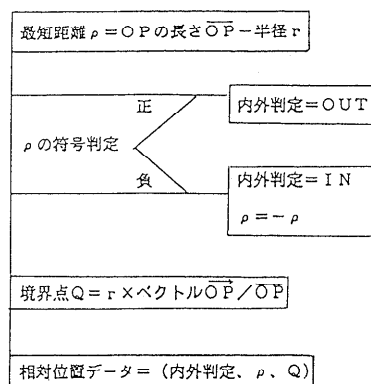
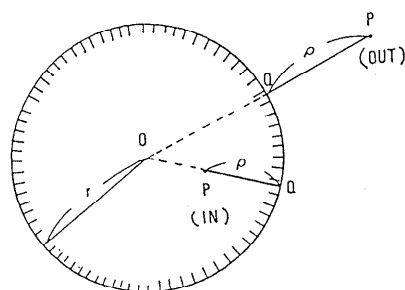


図3 半径 r の円板形状の相対表現手順

Fig. 3 Distance prove procedure of circular board.

が持つような操作の途中結果を反映するための複雑なデータ構造に基づく多量のデータ構造をいっさい持つておらず、データ量も格段に少なくなる。

4.2 加工・編集コマンド

これは、生成済の形状の相対表現手順を利用する形で、形状または形状間の加工・編集操作によりつくられた形状の相対表現手順を生成する手段である。

生成形式：(操作関数名, 操作データ,
相対表現手順名のリスト)

操作関数名は操作の種類ごとに異なり、それぞれ個別の操作データのもとで与えられた形状または形状間に加工・編集を加えて生成される新たな形状の相対表現手順を生成する手段を指す。なお、生成された相対表現手順は空間の任意の点の座標値を与えることによって第1種または第2種の相対位置データを生成する。以下、主な加工・編集操作および生成する相対表現手順について述べる。

(1) 形状の反転操作

生成形式：(反転, 手順)

この操作は、形状の内と外の関係を反転して新たな形状をつくるものである。したがって、生成する相対表

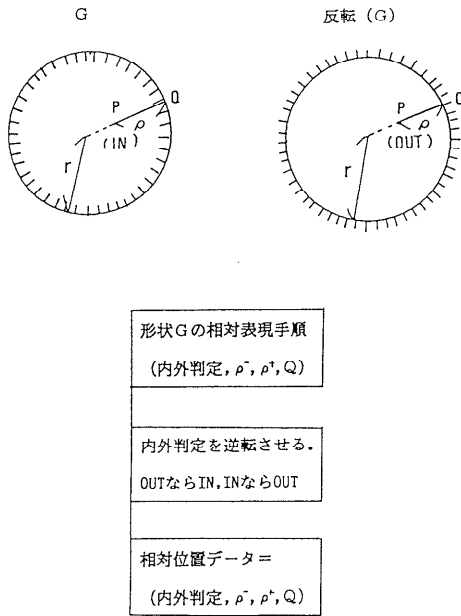


図4 反転形状の相対表現手順

Fig. 4 Distance probe procedure of complement.

現手順は、入力された手順が生成する相対位置データの内外判定を反転する手続きとして表現できる(図4)。

(2) 2つの形状の集合和

生成形式:(和, 手順1, 手順2)

この操作は、2つの形状の集合和によって新たな形状をつくるものである。今、図5に示すように形状 G_1 と形状 G_2 の集合和をとる場合を考え、形状 G_1 と形状 G_2 の点Pに対する相対位置データをそれぞれ $(I/O_1, \rho_1^-, \rho_1^+, Q_1)$ 、 $(I/O_2, \rho_2^-, \rho_2^+, Q_2)$ とすると、集合和の点Pに対する相対位置データは以下のようなになる。

- (a) I/O_1 , I/O_2 の両方がOUTのとき(図5(a)), ρ^- が小さい方の形状を G_i とする。
- (i) G_i の相対位置データが第1種のとき
(OUT, ρ_i^- , ρ_i^+ , Q_i)
- (ii) G_i の相対位置データが第2種のとき
(OUT, ρ_i^- , ∞ , nil)
- (b) I/O_1 , I/O_2 の片方がINのとき(図5(b)), I/O がINの形状を G_i とする。
- (i) G_i の相対位置データが第1種のとき
- (i-1) Q_i が他方(I/O がOUT)の形状に含まれるとき(図5(b-1))
(IN, ρ_i^- , ∞ , nil)
- (i-2) Q_i が他方(I/O がOUT)の形状に含まれないとき(図5(b-2))
(IN, ρ_i^- , ρ_i^+ , Q_i)

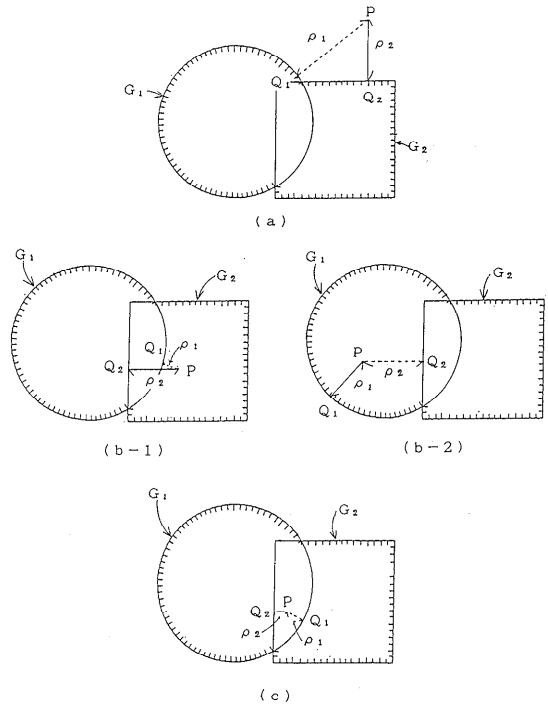


図5 集合演算(和集合)操作

Fig. 5 Distance probe procedure of union.

- (ii) G_i の相対位置データが第2種のとき
(IN, ρ_i^- , ∞ , nil)

- (c) I/O_1 , I/O_2 の両方がINのとき(図5(c))
(IN, $\min(\rho_1^-, \rho_2^-)$, ∞ , nil)

なお, nilとは点Qを特定できないことを意味している。

この相対位置データを導出する手順が集合和の相対表現手順となる。

(3) 2つの形状の集合積

生成形式:(積, 手順1, 手順2)

この操作は、2つの形状の集合積によって新たな形状をつくるものであり、次のように構成する。

(積, 手順1, 手順2) :=

(反転, (和, (反転, 手順1), (反転, 手順2)))

(4) 2つの形状の集合差

生成形式:(差, 手順1, 手順2)

この操作は、2つの形状の集合差によって新たな形状をつくるものであり、次のように構成する。

(差, 手順1, 手順2) :=

(反転, (和, (反転, 手順1), 手順2))

(5) 形状の運動操作

生成形式:(運動, 母曲線L, 手順)

この操作は、形状が母曲線Lに沿って運動したとき

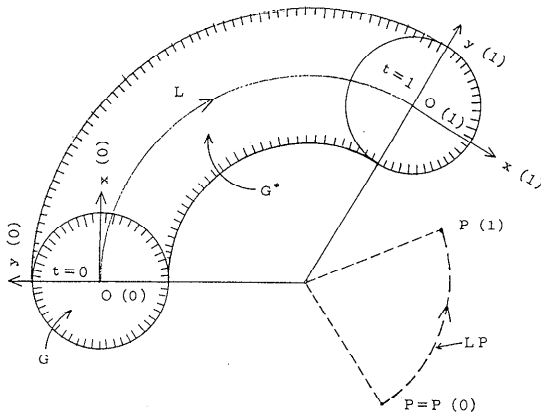


図 6 運動操作

Fig. 6 Distance probe procedure of motion.

に、空間を走査してできる新たな形状を定義するものである。母曲線 L は、

$$\{L(t) : 0 \leq t \leq 1\}$$

と表すものとし、 L に沿う運動は、元となる形状の 1 つの参照点が L 上を $L(0)$ から $L(1)$ まで形状とともに移動するものとする。図 6 は、形状 G が、母曲線である円弧 L 上を、座標系を L に沿って回転しつつ、参照点 O を基準に移動する運動操作を示している。ところで、運動操作によってできる形状 G^* に関する点 P の相対位置データは、形状 G を $t = 0$ に固定し、点 P を G の移動と逆に、 $P(0)$ から $P(1)$ へ移動させてできる軌跡（円弧 LP ）と、固定された形状 G との位置関係を調べることによって得られる。たとえば、移動軌跡 LP が形状 G と交差しなない場合には、点 P は形状 G^* の外にあり、点 P と形状 G^* との最短距離は曲線 LP と形状 G との最短距離である。また、移動軌跡 LP 上の 1 点でも形状 G に含まれる場合には、点 P は形状 G^* の内にある。ここでは、移動軌跡 LP を 2 分法的にセグメントに分割しつつ、相対位置データを求める。この相対位置データの導出する手順が運動操作の相対表現手順となる。

(6) 形状のオフセット操作

生成形式：(オフセット, オフセット量 δ , 手順)

この操作は、形状の境界をその形状の外側にオフセット量 δ (> 0) だけ移動させることによって新たな形状をつくるものである。図 7 に、形状 G よりその形状境界を δ だけオフセットした形状 G^* を作る例を示す。形状 G の点 P に対する相対位置データを $(I/O, \rho^-, \rho^+, Q)$ とすると、点 P に対する形状 G^* の相対位置データは、以下のように構成される。

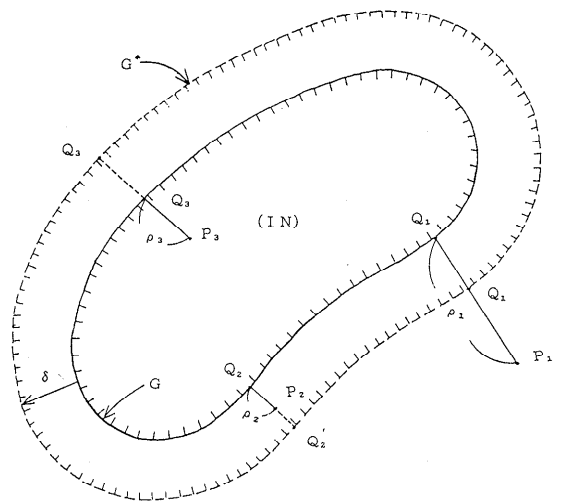


図 7 オフセット操作

Fig. 7 Distance probe procedure of offset.

- (a) 相対位置データが第 1 種のとき ($P = P_1$)
 - (i) I/O が OUT で $\rho > \delta$ のとき
(OUT, $\rho - \delta, \rho - \delta$, 線分 P_1Q_1 上 P_1 より距離 $\rho - \delta$ の点 Q_1')
 - (ii) I/O が OUT で $\rho \leq \delta$ のとき ($P = P_2$)
(IN, $\delta - \rho, \delta - \rho$, 線分 Q_2P_2 の延長上 Q_2 より距離 δ の点 Q_2')
 - (iii) I/O が IN のとき ($P = P_3$)
(IN, $\rho + \delta, \rho + \delta$, 線分 P_3Q_3 の延長上 Q_3 より距離 δ の点 Q_3')
- (b) 相対位置データが第 2 種のとき
 - (iv) I/O が OUT で $\rho^- > \delta$ のとき
(OUT, $\rho^- - \delta, \rho^+ - \delta$, nil)
 - (v) I/O が OUT で $\rho^+ \leq \delta$ のとき
(IN, $\delta - \rho^+, \infty$, nil)
 - (vi) I/O が IN のとき
(IN, $\rho^- + \delta, \infty$, nil)

ただし、(iv)、(v)、(vi) 以外 (I/O が OUT で $\rho^- \leq \delta, \rho^+ > \delta$) の場合、以下に示す手続きにより、第 1 種相対位置データを求め直し、上記 (i) または (ii) として処理する。

手続き：点 P を中心とする正方セル空間を設定する（たとえば、1 辺が 4δ ）。この空間を相対表現手順を利用して Quad-tree (3 次元の場合は Oct-tree) 表現する。この展開したセルのうちで、形状 G 内に含まれかつ点 P に最も近いセルの中心を点 Q とする。なお相対表現手順の利用法は、後述の形状の概形表現と同じである。

この相対位置データの導出する手順がオフセット操

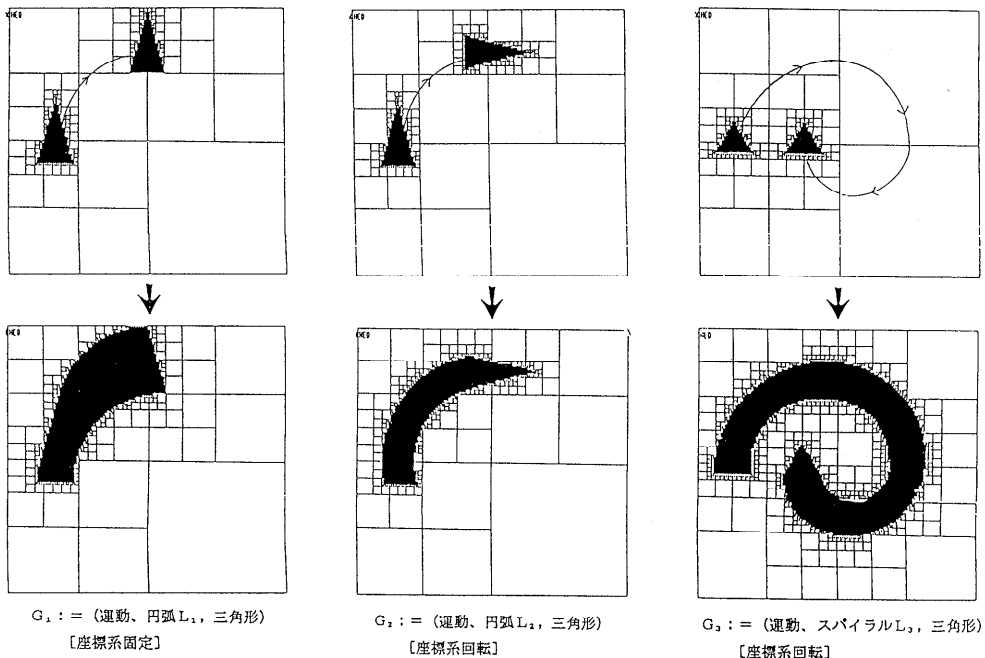


図 8 形状の運動操作の計算例
Fig. 8 An output example of motion.

作の相対表現手順となる。

(7) 形状のフィレット操作

生成形式: (フィレット, 凸部丸め量 δ_1 ,
凹部丸め量 δ_2 , 手順)

この操作は, 形状の凸部角点を半径 δ_1 の球で丸め, 凹部角点を半径 δ_2 の球で丸めて新たな形状をつくるものであり, 次のように構成できる。

(フィレット, δ_1, δ_2 , 手順) :=
(オフセット, δ_1 ,
(反転, (オフセット, $\delta_1 + \delta_2$, 反転,
(オフセット, δ_2 , 手順))))

4.3 応用コマンド

これは, 形状自身にかかわる諸特性を, 相対表現手順を用いて計算するための手段である。

以下, 主な応用コマンドの内容について述べる。

(1) 空間細分による形状の概形表現

利用形式: (概形, 空間の 1 辺 H, 手順)

このコマンドは形状を Quad-tree (3 次元の場合は Oct-tree) 表現に展開することにより, 概形表現を行うものである。具体的には, 1 辺 H の正方セルの中心点 P に対する形状 G の相対位置データを相対表現手順により求め, 以下の判定基準で判定を行う。

基準 1: 点 P を中心とした半径 ρ^- の球がセルを含むとき, 内外判定データの IN, OUT に応じて, セルは '満', '空' と判定する。

基準 2: 点 P を中心とした半径 ρ^- の球がセルに含まれるとき, セルは '半空' であると判定し, 元のセルの中心を通り, 各辺に平行な直線で 4 つ (ないしは Oct-tree の場合は 8 つ) の合同な正方セルに細分し, 同様な処理を繰り返す。

上記の基準が適用できない場合には, 仮にセルを '半空' であると見なして同様な細分を繰り返す。この再帰処理的は, 細分化されたセルの 1 辺の大きさがある一定値以下 (分解能) になるまで繰り返す。こうして取り残されたセルは 1/2 の確率で '満' か '空' に振り分ける。この結果, '満' の属性を得たセルの集合がオブジェクトの概形表現となる。

(2) 形状のマスプロパティの導出

利用形式: (応用関数名, 手順)

このコマンドは, 形状の特性量である, マス・プロパティ (面積, 体積, 重心, 重心まわりの 2 次モーメント) を, (1) の方法によって得られる Quad-tree (ないし Oct-tree) 表現より計算するものである。すなわち, Quad-tree (ないし Oct-tree) 表現の '満' または '半空' のセルについて, 諸量を計算し, それらの総和をとる方法を用いる。たとえば Quad-tree の場合, オブジェクトの面積は, '満' のセルの各面積を集計した値となる。

(3) 形状の表示

利用形式: (輪郭, 手順)

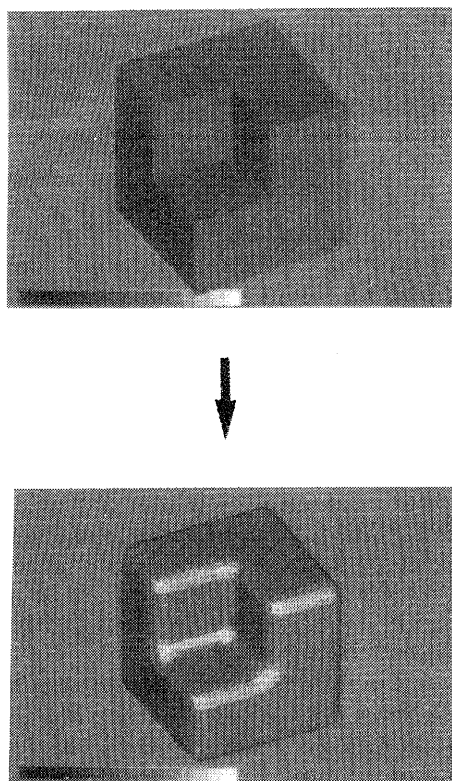


図9 フィレット操作例

Fig. 9 An output example of fillet operation.

このコマンドは、形状の輪郭（境界）を表示するものである。(1)の方法で得られる Quad-tree (ないし Oct-tree) 表現を利用し、表示の分解能以下となった‘半空’のセルについて、セルの中心点 P に対する形状 G の相対位置データより、最短距離点 Q を求め、 Q を通り線分 PQ に垂直な直線がセルによって切り取られる線分(群)を表示する。

5. 実行例および検討結果

本提案手法に基づき、基本機能のみの簡単なモデラーのプロトタイプを作成し、本モデルの実現性と有効性に関し今回は主に機能的な面について検証を行った。図8、図9および図10は従来取り扱うことが困難であった形状の加工・編集の例である。図8は運動操作(2次元)を表しており、図9はフィレット操作の結果、図10は各種変換操作(鏡像変換等)の結果を表している。実際、たとえば運動した物体がスキャンしたボリュームを形状表現した例はあるが、微分幾何学的な複雑な処理を必要としており、鏡像変換等の非線形的な変換も含めて、筆者らの知る限り一般の汎用モデラーではサポートされていない⁸⁾。また、本モ

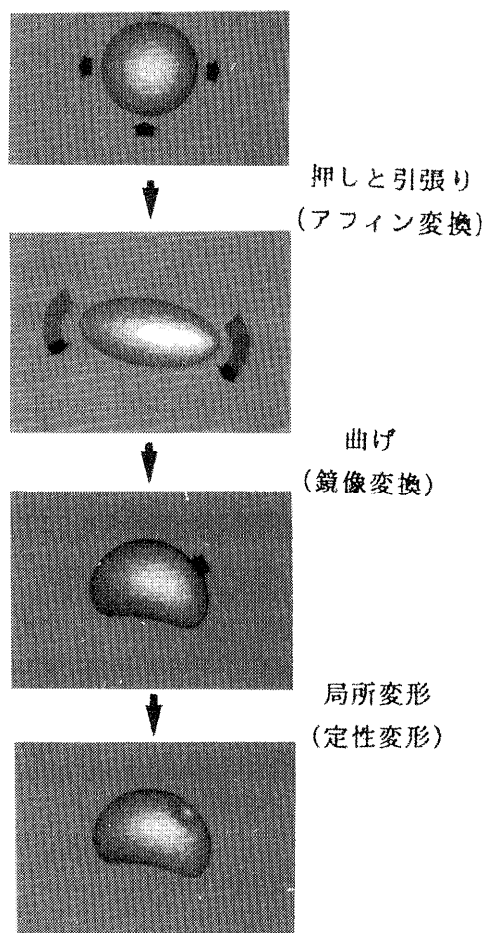


図10 各種変換操作

Fig. 10 An output example of mapping operation.

デラーのように操作の途中結果を保存しない CSG タイプではフィレット間の相互干渉や巨大半径を取り扱うことが可能な本格的なフィレット機能は存在しない。

以上の実行例からも分かるように、本手法は、種々の強力な加工・編集操作により、取り扱える形状の範囲が広く、操作自体もオブジェクト指向に適した形で簡明な処理系となっている。また、表現精度に関しても、処理の打ち切り精度を変更することにより、制御が容易に行える。処理系自体に関しても、すべての処理を相対表現手順のもとで統一化しているため、形状の複雑さや特異性に依存した個別処理や例外的処理がなく、堅固な系として実現できる。処理性能についての定量的な検討結果については、今回は割愛するが、少なくとも本実験の範囲では処理時間は対話処理許容限度内に入っていた。しかし、本手法は、必要に応じて行う空間分割による第1種相対位置データの生成に代表されるように、計算時間を要する処理があり、実用

上重要である処理の高速性の点で問題がある。ただし、計算時間を要す処理の多くは、空間的に独立な多数の処理に分解可能であるため、並列処理による高速化の可能性はある。今後の検討課題としたい。

また、本手法では、形状の構成要素としての面、稜線、点を陽に表現することが若干難しい。したがって、これらの情報を必要とする処理に関しては、他のモデラーとの協調も必要である。このように、本手法には、応用によって適、不適があるものの、本提案手法の特徴をおおいに活かせる広範囲の応用が考えられる。本モデルは、定義空間を等質的に表現する、すなわち、オブジェクトによって占められた空間に対してそれ以外の部分をあたかも補オブジェクトのように扱う、形状モデラーというよりスペースモデルと呼ぶべき特徴がある。したがって空間の認識と理解を要する問題として

- (1) 設計用、CG 用形状モデルの作成
- (2) 2/3 次元画像処理
- (3) 物体の配置、組立て、据付け、分解、搬入、搬出のための手順設計
- (4) NC 加工手順設計
- (5) ロボットの自律走行、視覚障害者の歩行支援
- (6) 要素分割の要らない有限要素法等があるが、これらが適用対象にあげられる。これらについても今後の研究課題としたい。

6. おわりに

機械系 CAD や CG に適した距離尺度に基づく形状表現法を開発した。本手法は、空間上の任意の点に対して対象とする形状の境界までの最短距離に関するデータを生成する手順（相対表現手順）を、対象形状と同一視し、形状を一元的に表現するものであり、以下の特徴がある。

- (1) 取り扱える形状の範囲が広い。
- (2) 処理系が堅固である。
- (3) 表現精度の制御が容易である。
- (4) 処理系が簡明である。

今回、本手法に基づくモデラーのプロトタイピングを行い、上記事項に関する実現性と有効性を検証するとともに、基本的な問題点についても明らかにした。今後の課題としては、

- (1) 本手法にあったアプリケーションの研究
 - (2) 並列処理を含めた処理高速化の研究
- があげられる。

謝辞 最後に、本研究について様々な御討論をいただいた（株）日立製作所日立研究所の各位に感謝の意

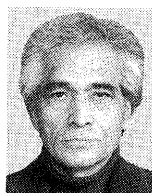
を表します。

参考文献

- 1) Braid, I.C. and Lang, C.A.: Computer-Aided Design of Mechanical Components with Volume Building Bricks, *Proc. PROLAMAT '73* (1973).
- 2) Okino, N., Kakazu, Y. and Kubo, H.: TIPS-1; Technical Information Processing System for Computer Aided Design, Drawing and Manufacturing, *Proc. PROLAMAT '73* (1973).
- 3) Hunter, G.M.: Efficient Computation and Data Structures for Graphics, Ph.D. Dissertation, Electrical Engineering and Computer Science Department, Princeton University (1978).
- 4) 野中, 川島, 徳増: 距離尺度に基づく統一的な形状表現法の研究, 第 6 回設計自動化工学講演論文集, pp.31-33 (1988).
- 5) 野中, 川島, 徳増: オブジェクト指向型の図形処理法, 第 37 回情報処理学会全国大会論文集, pp.1677-1678 (1988).
- 6) Nonaka, S., Tokumasu, S., Kawashima, Y. and Takahashi, K.: A New Geometric Model: Distance Measure Model, *ASME Proc. Design Automation Conference, Advances in Design Automation*, Vol.1, pp.135-141 (1989).
- 7) 野中, 徳増: 距離尺度に基づく形状表現法, グラフィクスと CAD, 50-3, pp.17-24 (1991).
- 8) Shiroma, Y., Kawashima, Y. and Kakazu, Y.: Generating Condition for Bounding Surfaces and Edges of Trajectory Volumes, *Proc. Sapporo CG Symposium* (1990).

(平成 9 年 1 月 29 日受付)

(平成 9 年 11 月 5 日採録)



徳増 真司（正会員）

1940 年生。1963 年横浜国立大学電気工学科卒業。（株）日立製作所日立研究所入社。1973～74 年米国スタンフォード大学大学院 OR 学科修士課程修了（社費留学 OR 修士），1984 年工学博士（京都大学），1995 年退社。現在神奈川工科大学情報工学科教授。生産情報システム（CAD/CAM/CAE, CG 等）システム工学, OR, 計算幾何学の研究に従事。1989 年河内賞（記念技術賞）受賞：「原子力発電プラント三次元 CAD システムの開発」。米国 IEEE, OR 学会, 電気学会各会員。

**野中 士郎（正会員）**

1955年生。1979年東京大学工学部卒業。同年（株）日立製作所に入社。同社日立研究所にて機械系CADの研究を経て現在製品管理の研究開発に従事。現在日立研究所主任研究員。機械学会会員。

**仁尾 都（正会員）**

1945年生。1969年京都大学理学部卒業。工学博士。同年（株）日立製作所に入社。CAD端末、グラフィックス、3次元CAD/CAM、高齢者支援情報システムを研究。現在システム事業部勤務。電子情報通信学会、日本健康医学会各会員。

**原島 一郎**

1963年生。1988年北海道大学工学部大学院修士課程電気工学専攻修了。同年（株）日立製作所入社。機械系CAD/CAM、形状処理、エンジニアリング情報制御技術の研究開発に従事。精密工学会、計測自動制御学会各会員。

**松本 輝夫（正会員）**

1938年生。1962年熊本大学機械工学科卒業。工学博士。同年（株）日立製作所に入社。2次元図形処理プラント配管のCAD/CAMを研究。現在（株）日立情報ネットワークに勤務。日本機会学会、日本社会情報学会、日本オペレーションズ・リサーチ学会各会員。