

距離尺度に基づく形状表現法

野中 士郎 徳増 眞司

機械系CAD/CAMやCGに適した、距離尺度に基づく統一的な形状表現法を提案する。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到る最短距離に関するデータを求める関数（これを“相対表現手順”と呼ぶ）を形状の計算機内部表現とするものであり、以下の特徴がある。（1）形状の表現力が豊かである。（2）処理系が堅固である。（3）システムが柔軟で拡張性が高い。

また、本手法に基づく形状モデラーを作成し、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るものの、上記の特徴が確認でき、従来の手法を補完する形状モデリングの一手法となり得る見通しを得た。

GEOMETRIC MODELLING METHOD BASED ON DISTANCE MEASURE

Shiro Nonaka Shinji Tokumasu

Hitachi Research Laboratory, Hitachi, Ltd.

4026 Kuji-cho, Hitachi-shi, Ibaraki-ken, 319-12 Japan

A new geometric modelling method for mechanical CAD/CAM and CG is presented. In this method, the internal description of a shape is defined as a procedure whose function is to derive a set of data which describes the relative positional relation between the shape and a given reference point. The characteristics of this method are the followings. (1)The range of representable shape is wide. (2)The modeller is robust. (3)The system can be flexible. And a prototype of the modeller demonstrated its wide capabilities as the new strong geometric modeller.

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テープ作成支援など、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、形状モデルに対しても、それに対応できる堅固さや柔軟性が要求されている。

本報では、その為の距離尺度に基づく形状表現法について報告する。

2. 従来の形状表現法の問題点

機械系CADやCGの為の形状表現法として必要な要件は、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、Octree法）を検討すると以下のことが言える。

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を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に対しては良いが、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、

別途個別的な手法が必要である。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（例えば表示）する場合に時間を要し、⑤に対しても問題がある。

3) Octree法⁽³⁾

これは、形状に対し、ある単位空間（通常立法体）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くかを判断し、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時、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、形状を手順として一元的に表現するものである。

従って、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。例えば球の場合には，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。例えば形状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，形状に係る諸特性（マス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/0（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。

$$(I/0, \rho, Q)$$

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/0（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。

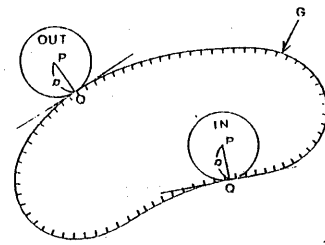
$$(I/0, \rho^-, \rho^+, Q)$$

ところで，第1種相対位置データは，

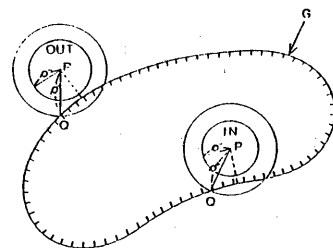
$$\rho^- = \rho^+ = \rho$$

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。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。



(a) 第1種相対位置データ



(b) 第2種相対位置データ

図1 相対位置データ

4. 形状モデラーの概要

形状モデラーの基本構成を図2に示す。

ここで，オブ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及び形状（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及び形状を用い、形状に係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係らず、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・編集及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。

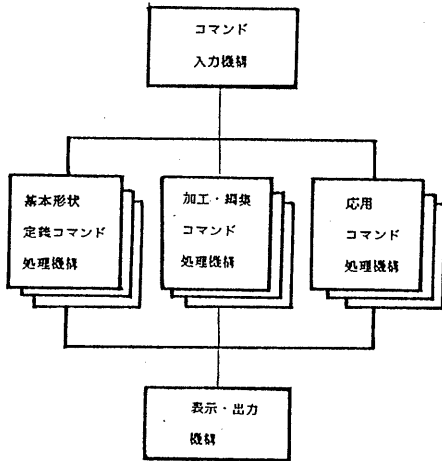


図2 モデラーの基本構成

以下、各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、以下のものである。

(1) 0次元形状 : 点

(2) 1次元形状 : 直線, 円, ...

(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, ...

例えば、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。

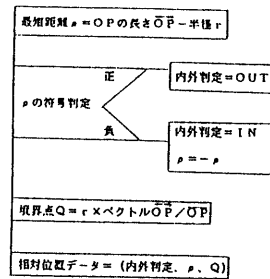
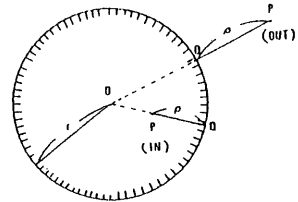


図3 円板形状の相対表現手順

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毎に異なり、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。

ここで、加工・編集操作としては以下のような

操作を考える。

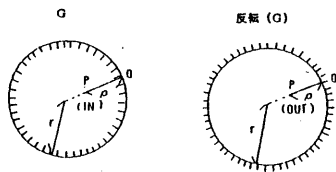
・反転操作，・集合演算（和，積，差），・平行移動，回転操作，・運動操作，・アフィン変換，・オフセット操作，・フレット操作，……。

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，形状の内と外との関係を反転して新たな形状をつくるものである。従って，生成する相対表現手順は，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）。



形状Gの相対表現手順 (内外判定, ρ^-, ρ^+, Q)
内外判定を逆転させる。 OUTならIN, INならOUT
相対位置データ (内外判定, ρ^-, ρ^+, Q)

図4 反転形状の相対表現手順

(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ようになる。

i) $I/O_1, I/O_2$ の両方がOUTの時（図5(a)）

ρ^- が小さい方の形状を G_1 とする。

ii-1) G_1 の相対位置データが第1種の時
(OUT, ρ_1^-, ρ_1^+, Q_1)

i-2) G_1 の相対位置データが第2種の時
(OUT, ρ_1^-, ∞, nil)

ii) $I/O_1, I/O_2$ の片方がINの時（図5(b)）

I/O がINの形状を G_1 とする。

ii-1) G_1 の相対位置データが第1種の時

ii-1-a) Q_1 が他方(I/O がOUT)の形状に含まれる時（図5(b-1)）

(IN, ρ_1^-, ∞, nil)

ii-1-b) Q_1 が他方(I/O がOUT)の形状に含まれない時（図5(b-2)）

(IN, ρ_1^-, ρ_1^+, Q_1)

ii-2) G_1 の相対位置データが第2種の時

(IN, ρ_1^-, ∞, nil)

iii) $I/O_1, I/O_2$ の両方がINの時（図5(c)）

(IN, $\min(\rho_1^-, \rho_2^-), \infty, nil$)

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。

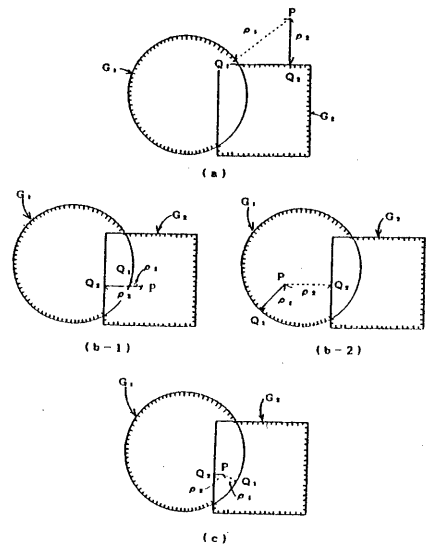


図5 集合演算（和集合）操作

(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に沿って運動した時に、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(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。

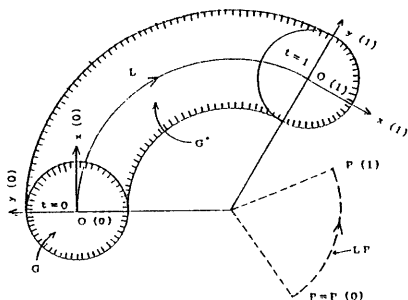


図6 運動操作

(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/0, ρ⁻, ρ⁺, Q)とすると、点Pに対する形状G*の相対位置データは、以下の様に構成される。

a) 相対位置データが第1種の時 (P = P₁)

i) I/0がOUTでρ > δの時

(OUT, ρ - δ, ρ - δ, 線分P₁Q₁上
P₁より距離ρ - δの点Q₁)

ii) I/0がOUTでρ ≤ δの時 (P = P₂)

(IN, δ - ρ, δ - ρ, 線分Q₂P₂の
延長上Q₂より距離δの点Q₂)

iii) I/0がINの時 (P = P₃)

(IN, ρ + δ, ρ + δ, 線分P₃Q₃の
延長上Q₃より距離δの点Q₃)

b) 相対位置データが第2種の時

iv) I/0がOUTでρ⁻ > δの時

(OUT, ρ⁻ - δ, ρ⁺ - δ, nil)

v) I/0がOUTでρ⁺ ≤ δの時

(IN, δ - ρ⁺, ∞, nil)

vi) I/0がINの時

(IN, ρ⁻ + δ, ∞, nil)

但し、iv), v), vi)以外 (I/0がOUTでρ ≤ δ, ρ⁺ > δ) 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を中心とする空間を設定する(例えば、一辺がmin{2ρ⁻, 4δ})。この空間を相対表現手順を利用してOctree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作の相対表現手順となる。

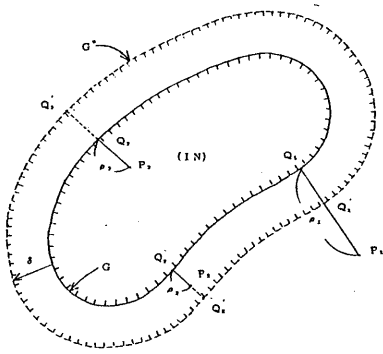


図7 オフセット操作

(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、形状自身に係る諸特性を、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, 空間の一边 H , 手順)

このコマンドは形状をOctree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、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 の球がセルを含む時、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時、セルは '半空' であると判定し、細分を繰り返す。

上記の基準が適用できない場合には、仮にセル

を '半空' であると見なして細分を繰り返す。

(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Octree表現より計算するものである。即ち、Octree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。

(3) 形状の表示

利用形式：(輪郭, 手順)

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Octree表現を利用し、'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ルを作成し、機能の確認を行った。

図8, 図9および図10は従来取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 (アフィン変換等) の結果を表している。

以上の実行例からもわかるように、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。

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。但し、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がある。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。従って、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ので、これらについても今後の研究課題としたい。

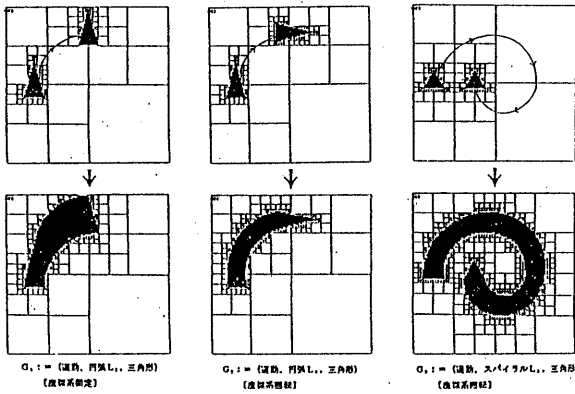


図8 形状の運動操作

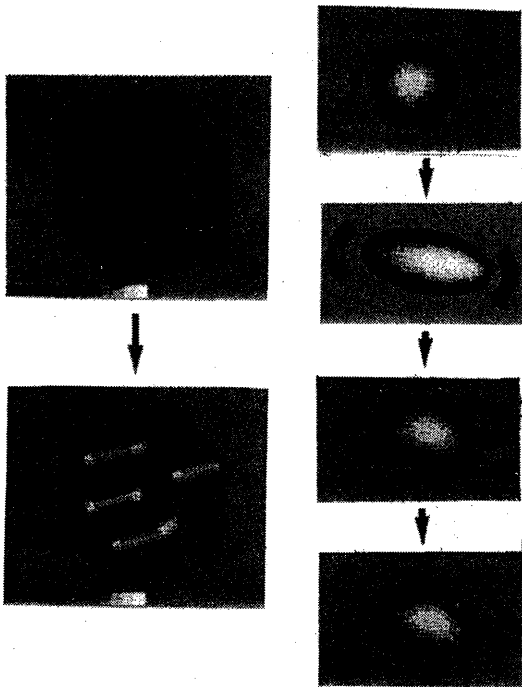


図9 フィレット操作

図10 各種変換操作

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。

- (a) 取扱える形状の範囲が広い。
- (b) 処理系が堅固である。
- (c) 表現精度の制御が容易である。
- (d) 処理系が簡明である。

今後の課題としては、

- (a) 本手法にあったアプリケーションの研究。
- (b) 並列処理を含めた処理高速化の研究。

があげられる。

【参考文献】

- (1) I.C.Braid, C.A.Lang: Computer-Aided Design of Mechanical Components with Volume Building Bricks, Proc. of PROLAMAT '73 (1973)
- (2) N.Okino, Y.Kakazu, H.Kubo: TIPS-1; Technical Information Processing System for Computer Aided Design, Drawing and Manufacturing, Proc. of PROLAMAT '73 (1973)
- (3) G.M.Hunter: Efficient Computation and Data Structures for Graphics, PhD. 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講演論文集, pp.1677-1678 (1988)
- (6) S.Nonaka, S.Tokumasu, Y.Kawashima and K.Takahashi: A New Geometric Model: Distance Measure Model, Advances in Design Automation Vol.1, pp.135-141 (1989)