

## 構成的形状記述 (CGM) と形状エンジン

瀬戸口 良三

東北大学 工学部 精密工学科

形状エンジンの果たすべき役割や効用など、その全体像を明らかにするため、まず、計算機による形状処理とそこでの固有な計算機利用技術について検討した。これらの考察から、形状処理における最大の問題点が形状記述にあることや、形状エンジンの構築のための基本ツールとしては、構成的形状記述 (CGM) が唯一期待できる形状記述であることを明らかにした。最後に、形状処理の中核的／敷衍的処理における形状エンジンの基本フレームや固有機能などについての考察をベースにして、プロトタイプとして現在構築しているCGMに基づく形状エンジンが、形状処理におけるシステムコアとして、能動的にダイナミックに機能することや、計算機における設計用ツールとして、インタラクティブな形状処理方式を採用することにより、リアル・タイムな処理が可能であることなどを示した。

On the Geometric Shape Processing Engine for the 3-D Entity  
based on the Constructive Geometric Modeling (CGM)

Ryozo SETOGUCHI

Department of Precision Engineering, Faculty of Engineering,  
TOHOKU UNIVERSITY

Aramaki Aza Aoba, Sendai, JAPAN 980

In this report, to relate the established utilities and performed role of the proposed Geometric Shape Processing Engine (GSPE) used as typical example for generating the 3-D entity, Computer Aided Shape Processing (CASP) and its application technologies can be examined. In this case, as for the most important problem for CASP, the limit of Geometric Modeling can be also examined. And moreover, the whole requirement for GSPE can be determined. From these results, it can be made clear that only the method called Constructive Geometric Modeling (CGM) can be adopted as the basic tool established for GSPE. And then it is concluded that the prototype engine followed by CGM can be well performed for the set up objectives on CASP.

## 1 計算機利用技術と形状処理

計算機に基づく形状処理は、図1に示すかたちのフローとして、実行するのが慣用的であり、一般的であると言える。例えば、計算機を媒体とする典型的な機械的視覚・認知処理(C.V.)においては、この流れ図の中で示した一連の操作に基づき、逐次/並列的処理として、視覚/認知処理に伴う形状処理を行う。また、形状処理の代表的ターゲット・フィールドであるCAD/CAMについて言えば、入力データ処理は、ここでの形状処理の基本的機能として、積極的に用いているのではなく、基本的機能の補完的機能として用いている。この種の機能は、本来形状データベースとのかかわりの中で用いられるのが一般的で本質的であると言える。ここでの基本的な機能は中核的/敷延的処理の機能にそのほとんどの部分を依存しており、それ以後の処理の流れを含めた流れが本来の処理機能の流れであると考えてよい。

計算機利用技術の一環として行われるこれらの形状処理は多種多様であるが、入力データ処理は一般に補間、平滑あるいはフェアリングなどの形状データの圧縮や復元の操作あるいは構成的なフィティング処理などのいわゆる形状処理における前処理が固有の処理機能である。ここでの処理機能は処理の流れの中での主体的処理としての機能と言うよりは、次ぎのステップと重畳する機能あるいは従属する機能として展開されることが多いようである。

中核的/敷延的処理は主に形状定義/形状修飾の処理機能として設計・構築されるべき性格の処理機能である。この場合、構築される中核的処理機能としては、拡張的補間の方法として定義できるB-Spline関数やBézier関数による形状生成の機能に基づく処理方式が用いられる。また、集合演算によるプリミティブ・ソリッド・モデルに基づく複製の形状生成の方法も用いられる。

ここでは同時に、形状処理の一環として、随伴的機能であるが、形状記述のためのデータ構造の構築が必要となる。これは、当然、その帰結として、形状データベースの構築を伴うことになる。このような形状記述に随伴する形状データ格納のための枠組みは、副次的ではあるが、必須の処理機能であり、これが定義した敷延的処理に対応する処理機能である。

一方、出力データ処理はイメージの表示処理を中核とする機能に依存する形状処理の領域である。出力データ処理においては、まず基本的機能として、イメージ表示のための操作的処理が求められる。これは、一般に、回転、移動、反転そして拡大・縮小などの処理操作であり、これらの操作は図形・画像に対して直接施される。また、ディスプレイなどの出力装置に対する固有の表示機能として、ウィンドーイング/クリッピング処理が行われる。さらに、3次元図形に対しては、透視投影の技術を核に、立体視として、ステレオ・スコピックやアナグリフ処理が多用される。これは線図形としての固有の処理であり、この種の表示技術に対しては、隠れ線/隠れ面の処理が付加的に用いられる。出力データ処理にお

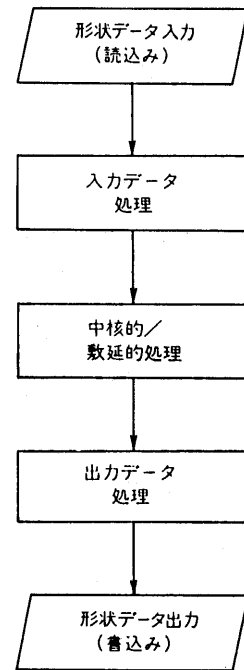


図1 計算機による  
形状処理

いては、更に高次表示処理として、実体の表現法の一つであるソリッドモデルに対し、リアルなイメージを提供する目的のために、レンダリングな処理が行われている。このような処理として、シェーディング処理、シャドーイング処理やテクスチャ処理、さらには背景処理などが行われているが、グローシェーディングやテクスチャ処理や背景のぼかし処理などに対しては、光線追跡法の技術がこれらの技術を補完するかたちで使われている。

出力データ処理は、このように表示装置に対するアクセス技術の性格を保有するもので、ディスプレイ装置などに対する表示環境を提供するものである。

この意味で、出力データ処理は、入力データ処理と異なり、独立した機能として構築できる。しかし、出力データ処理は、形状処理の全体の流れにおいて最も下流に位置する機能であり、当然、形状記述の機能に強く拘束される処理形態を取らざるを得ない。明らかに、出力データ処理のそれぞれの方法は形状記述に深く影響を受けるのであるから、形状記述に対し、本来的には、従属的に構築せざるを得ないのである。

しかし、概して、これまでに提案されてきた形状記述は、形状定義の機能として要求される形状生成／形状表現などの機能に対し、十分に完備されておらず、そこでの機能は便宜的であり、事例対応的であるように思われる。このような技術的背景から、これまで、出力データ処理はグラフィックス・エンジンとして構築され、形状処理の標準化は、これを中心に押し進められ、形状記述はその主たる座をこれに明け渡すかたちで、極めてマイナーな役割を演じてきたようである。

形状処理のシステム化は、形状記述を従とするかたちで、グラフィックス・エンジンを中核として行われている。これは、形状処理の流れから見ても、極めて不自然で、本末転倒なかたちであると言える。このことが形状処理の一貫性を阻害しているのは言うまでもないことである。

このような観点から、形状処理の中核的／敷衍的処理の機能に対応する形状エンジンの構築を提案し、形状エンジンのビジョンやそのフレーム、更には実現の可能性やその果たすべき役割と効用などについて検討してみる。

## 2 形状処理と形状記述

形状エンジンの具体的な骨格や性能などを議論する前に、形状エンジンの構築において、必要不可欠な基本ツールとなる形状記述について考察する。これは、また、形状処理の流れの中で展開される中核的処理においても、必須の機能として要求される形状定義／形状修飾の処理機能に深くかかわるものであり、そこでの処理機能のフレームを定めるものである。

ここでは、それ故、これまで提案された形状記述について概説し、それがかかわる形状処理における問題点について検討してみる。

提案された形状記述の多くは、そこでの形状生成の機能について言えば、拡張的補間として定義できる方法に依存するかたちで機能している。勿論、前述したように、集合演算に基礎を置く形状生成の方法も提案されているが、用いられる形状生成の方法は、プリミティブ・ソリッドに基づき形状を生成するデュプリケータリングな方法であり、生成される形状は、当然、マザーソリッドであるプリミティブソリッドの形状を転写するかたちで形成される。それ故、形状生成は母形状により拘束されるので制約的であり、この方法の限界点など形状処理における

問題点は明らかである。従って、検討すべきターゲットは、拡張的補間に基礎を置く方法に限定する。

言うまでもなく、検討の対象として選択した拡張的補間も、基本的には、関数近似に基礎を置く数学的方法である。関数近似の方法は、標本点とその点上の接線などのかたちで付与される初期条件に基づき、設定した関数のパラメータを定め、標本点の間の欠落している形状構成点を内挿点として求める方法であると定義できる。この場合、設定する関数は(区分)多項式として表現するのが慣用的であり、設定した関数のパラメータは単純に境界条件により定めるという定式化の方法もあるが、一般的な方法としては、境界条件の重み付けとして定式化する方法を踏襲するために、基底関数を用いて定式化する方法を用いる。空間図形の場合には、混合させ関数を導入して、定式化する方法を使うことが多い。この場合、混合させ関数を用いて、付与された境界条件に基づき定式化する間接的方法が基本的には用いられる。同時に、それと等価な表現形式の定式化として、基底関数を用いて、付与された境界条件に基づき定式化する直接的方法も使われる。この種の方法の一つの範疇としては、3次 Spline関数による方法や Coonsや糖坂の提案した方法があるが、これらの方法は、形状生成において、ほとんど制御的に実行することを期待できない方式である。これらの方法は、わづかに、初期条件として付与する構成点の位置や接線の方向あるいは大きさなどによる形状変更が期待できるだけであり、本質的には、補間の枠の中にある方法である。

一方、補間の枠をかなりの程度越えることができたと言う意味で、拡張的補間として定義できる方法としては、Bézier関数や B-Spline関数などに基づく形状生成の方法がある。B-Spline関数は Bezier関数の拡張的方法であり、この意味で、これらの方法は同系統に属する方法と言える。すなわち、B-Spline関数の基底関数は Bezier関数の基底関数を含むものであり、2つの方法の相異点はノットベクトルの導入とそれに依存して拡張された部分だけである。それ故、ここでは、特に Bézier関数による方法に限定して述べることにする。

Bézier関数などによる拡張的補間の方法は、境界条件として、ポリゴンやポリヘドロンを形成する手順が必要である。形成したポリゴンなどは、形状生成の初期条件の役割が付与されており、制御ポリゴンと呼んでいる。形状生成はこれを基準図形として実施される。この場合、導入した基底関数の重み付けの機能に基づき、生成した基準図形の頂点系列に従い、所望の形状は生成される。すなわち、Bézier関数に基づく形状生成は、図2において示した一連の手順に基づき、実行される。この方式における演算処理としては、Bézier Polinomialの係数である荷重因子の Bernstein基底に基づく算定あるいは混合させ関数による係数の算定と制御ポリゴン頂点に基づく形状構成点の位置座標の算定の2つの手続きが必要となる。ここでの形状の生成/変更は、基本的には、形成した制御ポリゴンの頂点

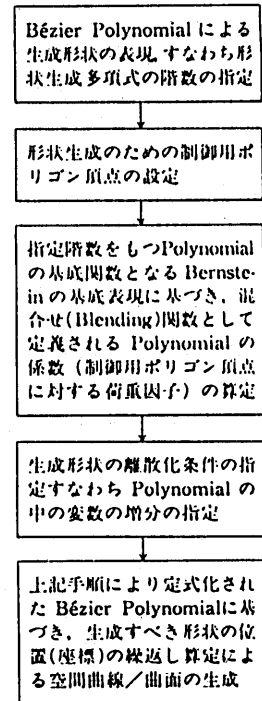


図2 Bézier関数による形状生成

の位置を変更することにより行うのである。この意味で、拡張的補間の方法は、制御ポリゴンに基づく形状生成の方法であると定義するのであるが、この種の間接的な制御操作に基づき形状を生成するのでは、形状生成の機能として、それほど効果的に機能するとは考えられない。確かに、生成すべき形状の概略形状が制御ポリゴンを通してかなり把握でき、判断できると言えるかも知れないが、Coonsや穂坂などの方法と比較して、格段に改善されているとは言えないようである。

このような事情を考慮して、この種の制御操作を間接的制御と定義し、拡張的補間の方法を間接制御型形状生成法として位置づけるのが適切かと考えている。

拡張的補間の方法は、制御ポリゴンに依存して形状を生成するので、端点以外の付与された形状の内部構成点を自在に通過させるような制御はほとんど不可能に近いと言える。このため、例えばフィッチングの機能などを提供するのとは容易ではなく、かなりの問題点を内包しているようである。

勿論、提案されたこの種の形状記述に対し、形状の最適化記述の機能などを期待することはほとんどできない相談と考えられる。形状の最適記述とは、形状の空間的変化の方向と大きさの変化、例えば形状の曲率の変化に対応する形状記述のことであり、この種の記述能力は、ハイライト処理における線形補間などに効果的であり、形状の高品質の表示処理を容易に、しかも高速に行うことが期待できるのである。また、形状記述の階層的処理なども当然実現不可能である。これは、形状を逐次構成的に可制御的に生成する手段がないと期待できない機能である。従って、形状の記述精度の管理などを期待するのは、困難なことである。

現用の形状記述は、このような意味で、極めて制約的であり、問題が山積していると言える。

### 3 パーフェクト・ソリッド・モデルと構成的形状記述 (CGM)

前節の考察で述べたように、これまでに提案された形状記述に対する問題点として列挙した幾つかの要因の中で、例えば形状最適化記述の機能が構築できなければ、一貫した手続きによる完備した形状処理の機能が期待できるような形状エンジンの構築はほとんど絶望的であると言える。

一方、これから述べる構成的形状記述 (CGM) は『逐次的／並列的な処理方式により、構成的に、かつ可制御的に形状を生成／表現できる形状記述』である。

CGMは工学的フレームに基づく形状記述として構築されるが、ここで用いる方式は、当然、形状を表現するために定式化する数学的モデルと等価なモデルとなっている。CGMは形状の数学的特性、例えば近傍系や接続関係に関する位相構造は勿論、曲率やねじれ率などの幾何学的構造などを、制御パラメータなどのかたちとして組み込むことにより、必要十分な記述レベルに到達している。

CGMは、それ故、これまで提案された形状記述と異なり、形状処理に伴う問題点の多くを克服できる唯一の完備した形状記述と言える。このような特徴を備えたCGMの基本的機能は、図3の流れとして、手続き的に構築できる。

この方式の詳細については参考文献によることができるので、ここでは、その全体像を容易に把握できるように、CGMの特徴を要約するだけにとどめておく。すなわち、以下のように、要約できる。

(1) 2次元形状や3次元形状の形状記述において、何等のインタフェースも必要としない同一の枠組みにより、一貫したかたちで形状処理ができる。

- (2) 曲面を多面体の極限構造として形成できるので、多面体から曲面までを同一の枠組みにより、一貫したかたちで記述できる。
- (3) 極めて単純な形状（シンプル・プリミティブ・ソリッド）から複雑な形状が自在に生成できる。
- (4) 形状記述において、自然発生的で計算機向きの整合したデータ構造が容易に構築できる構造化形状記述の機能が付与できる。
- (5) 形状記述において、数式表現を必要とする現用の間接的な方法の代わりに、形状データを中核とする手続き的処理として形状生成／形状表現の機能などが構築できる。いわゆるデータ・ディペンデントな処理を基本とする直接的な方法として形状処理が可能となる。このため、ユーザー・インタフェースをほとんど必要としない形状処理が可能となる。また、マン・マシン・インタフェースを容易に構成できるインタラクティブな形状処理の枠組みが構築できる。さらに、データ・フロー型の形状記述の枠組みの構築が可能となる。
- (6) 形状記述において、形状変化の極大・極小となる形状の特異点を自動的に決定できるので、形状の最適化記述が可能である。
- (7) 形状記述において、境界面を線状記述や面状記述として、自在に生成／表現できる。また、これらの記述方式は相互に自由に選択できる。
- (8) 逐次構造的に可制御的に形状を生成／表現する形状記述であるから、形状の記述精度を目的に応じて自在に管理できる。
- (9) 包絡線／包絡面に基礎を置く形状記述であるから、位相的關係が容易に確定できる。このため、自然に位相構造が定まる。また、ソリッドとして、必要不可欠な情報のすべてを必要に応じて自然に定めることができる。
- (10) 形状記述制御パラメータP、Q、R、Tはスカラー量であるから、形状の種々の変換操作に対して不変量となる。このことは、特にスケーリングの操作に対して、実際的で効果的な形状処理の機能を提供する。また、形状データベースの有効利用などについても、必要十分な機能を提供する。
- (11) 形状記述として、例えば記述能力、記述精度などに関し、必要十分であ

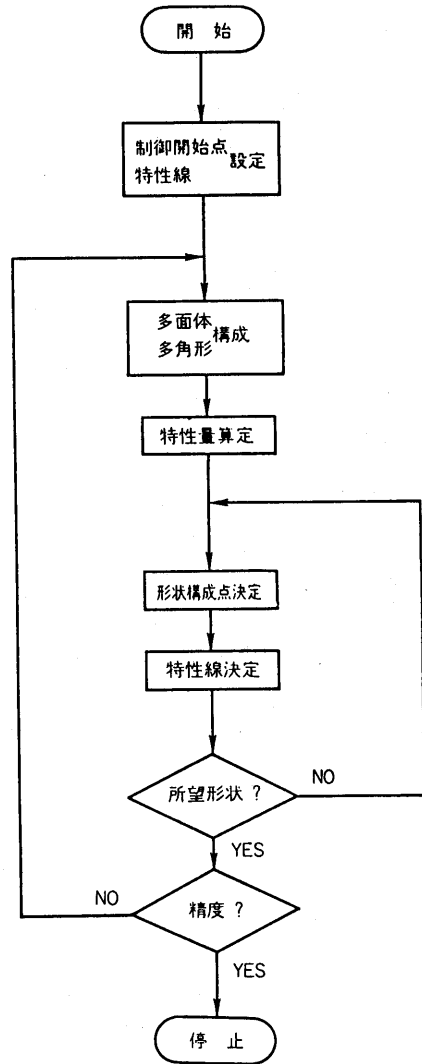


図3 構造的形状記述（CGM）  
の基本構造と基本機能

ると言えるから、種種の形状処理の段階において、極めて柔軟で、適応性に富んだ形状処理の固有技術を提供できる。また、特別な機能として、例えば形状の分解・復元の機能などを提供できるので、形状データによる形状記述として、関数曲面などは勿論のこと、任意の自由曲面も自在に分解し、一意に復元するかたちで記述できる。

(12) 逐次構成的な形状記述の手続きをベースにして、完全に独立したかたちで、形状の並列処理が容易に実行可能となる。

このような特徴をもつCGMモデルは、ワイヤー・フレームやサーフェイス・モデルは勿論のこと、その上位モデルであるB-RepsモデルやCSGモデルと完全に整合しており、包含的で、統合可能な形状記述である。また、多様な形状処理の機能を構築できる形状記述であり、これまでの形状処理の問題点をほぼクリアーしているので、いわゆるパーフェクト・ソリッド・モデルとして、形状エンジンの構築に対し、現状では唯一期待できる形状記述であると言える。

#### 4 構成的形状記述(CGM)に基づくプロトタイプ形状エンジン

形状エンジンの概念は、図4において示すようなかたちで、既に提示されている。しかし、図から判るように、この形状エンジンは、受動的に機能するような形態として構築され、展開されている。しかも、極めて複雑なインタフェースの構築により、システムの統合を試みている。これは、現用の形状記述の機能の限界に起因する結果と考えてよい。明らかに、システム機能のかなりの部分の肩代わりを、ユーザー・プログラムに求めており、システム機能に直接関与するかたちで、ユーザー・プログラムが機能することを期待している。ここでの形状エンジンは、それ故、静的なかたちで、システムの中に埋め込まれていると言える。

これに対し、プロトタイプとして、現在開発している形状エンジンは能動的に機能するもので、形状処理システムのコア(核)をなすものである。それは、ダイナミクスなかたちで、システムに埋め込まれるもので、インタラクティブに形状処理ができることを意図している。また、グラフィックス・エンジンが、受動的なかたちで、本来的に機能できるように、形状エンジンの構築と展開を考えている。形状エンジンは、それ故、形状処理の中核的/敷延的機能を受け持つもので、

完備した処理機能が提供できるように、当然種種の考慮が払われている。この場合、入力データ処理は、その本質から、ここでの付加的処理機能として、外延的機能のかたちで構築できることが基本的に要求される。また、これまでに構築されてきた形状エンジンに対しても、簡易インタフェースにより、容易に接続可能であることが、ここでの基本的要件と考えている。

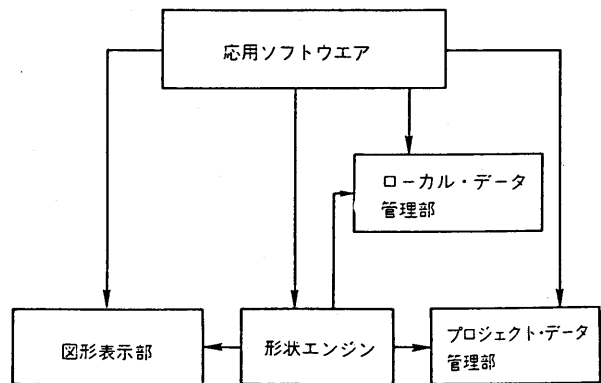


図4 CAD/CAEのための  
現用形状エンジン

これらの構成要件から、形状エンジン開発のための基本ツールとしては、CGMによる方法の他に、構築のための技法が無いことは明らかである。

図5として示した形状エンジンは、計算機援用統合型製造(CIM)システムにおいて、システム・コアとして機能するプロトタイプである。現在構築しているエンジンは、システムの水平・垂直の両方向に対し、完備したかたちで、自在に機能するものとして、機能展開が計られている。この場合、水平方向の機能は、設計支援の機能と定義できる機能であり、表示機能としてのグラフィックス・エンジンと設計データ管理機能としてのエンジニアリング・データベースなどの、これ

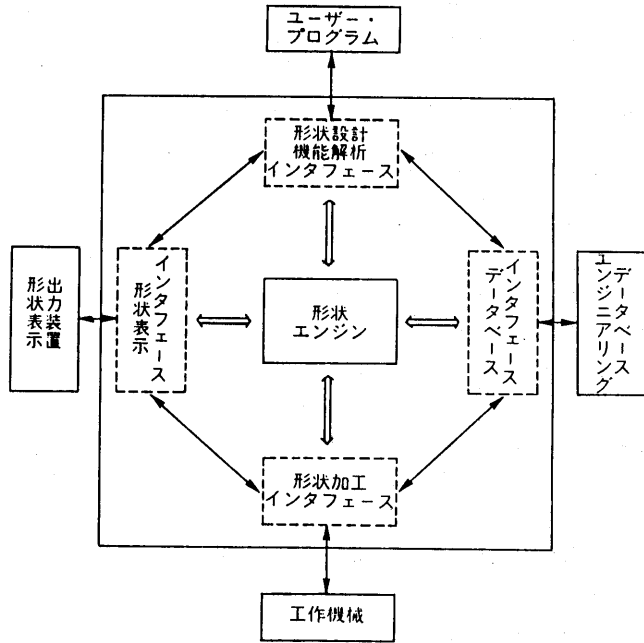


図5 CGMによるプロトタイプ形状エンジン

までに展開された機能により、十全に補完できるものである。現在のところ、形状エンジンは、これらの機能に対し、システム・インタフェースを介して、機能するように設計している。しかし、ここでは、形状データ表示部と形状データ管理部を、形状エンジンに対し、従属的なかたちで構築することも考えている。

一方、設計(CAD)の上流(CAE)と下流(CAM)の垂直方向に対しては、設計の機能がインタラクティブに作用することを、システム構築の最大の眼目にしている。ここでのCIMシステムは、インタフェースレスに近いかたちで独立に機能すると同時に、形状エンジンがリアル・タイムに直接作用する形態をシステム設計の基本としている。このため、メニュー方式により、プログラムレスとして、機能するように、種類の固有の技法などを開発しているが、ここでは、部分的画面機能を最後に提示し、その詳細は省略する。

参考文献

瀬戸口、中村：情報処理学会第23、26、27回全国大会講演集。

```
*****
***** DETAILED SHAPE GENERATION *****
***** SPACE WINDING RULER/FRENCH CURVED RULER *****
***** DATA DECISION *****
*****
```

- [ 1 ] NEW DATA CREATION
- [ 2 ] PRESENT DATA MODIFICATION
- [ 3 ] => MAIN MENU

CHOOSE(1- 3):

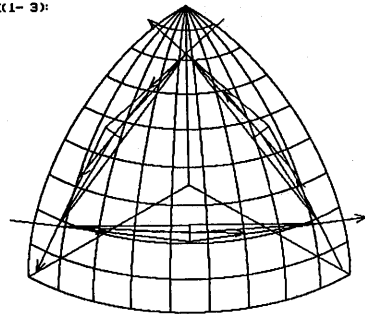


図6 形状生成処理における画面機能