

多重相貫図形が表現可能なスペースモデル用 頂点テーブルの検討†

大 沢 晃††

スペースモデリング^{1),2)}は、図形の相互位置関係を含む図形空間全体をポイントによって構造化表現する、二次元・三次元一貫した図形データベースのコンセプトである。この方式によれば、隣接空間・重なり図形の直接的検索が可能となるため、AND・OR等図形集合演算、運動図形の衝突検出、隠面・隠線消去表示、等高度の図形処理が高速に実行できる。本研究はこの方式を実現するために必要な、データ表現のために図形頂点に設置する頂点テーブルの構成法に関するものである。本方式は二次元・三次元スペースモデルのポイント表現および隣接空間検索の基本となる回転サーチをサポートすることで、上記各種図形処理の効率的な実行を可能とするものである。本方式の特徴は図形の交叉点を通常の頂点と同様の方法で陽に表現する点にある。これにより図形集合演算等相貫図形の高速処理が実現する。また本方式によれば従来一般に行われている^{3),4)}点、折れ線、凹凸・穴を含む多角形・多面体、等のほか、界面等表裏の定義されない面、モザイク状複合多角形・多面体、多重相貫体、等のあいまいさのない、かつコンパクトな表現が可能になる。そのほか本方式では、辺、面、空間をすべてポイントで表現すると共に、テーブルの固定長化について配慮しているため、図形追加・削除・加工等に伴うメンテナンスも容易であり、オンライン用汎用図形データベースのデータ構造として有用と考えられる。

1. ま え が き

ワイヤフレーム、ソリッドモデル等従来^{3),4)}の図形表現手法が個々の図形自体の表現に留まっていたのに対し、スペースモデル^{1),2)}は図形内外を対等な空間として扱い、部分空間を表現するポイントの組によって図形空間全体を構造化する、二次元・三次元一貫した図形データの表現法である。この方式ではポイント検索によって隣接空間・重なり図形の直接的検索が可能となるため、図形集合演算、運動図形の衝突検出、隠面消去表示等、広範囲の図形処理が、必要部分だけの局所的処理で実行できる。このため別論文^{1),2)}で示した例外的な場合を除き、多くの応用において全体図形数に依存しない $O(1)$ の処理速度を実現することができる。

スペースモデルでは図形集合演算等、図形重なり処理を高速化するため、図形の交叉点をデータとして陽に表現する手法をとる。図形の交叉点とは、二次元では図形辺間の交叉点、三次元では面を貫通する辺の貫通点のことである。本論文の主要ポイントは、このような図形交叉点や凹凸・回転を含む一般的な図形頂点の表現が可能で、かつスペースモデルの特徴である局所的図形処理の効率的な実行を可能とする、二次元・三次元一貫思想による頂点テーブルの構成法を提案することである。

る。局所的図形処理のためには、本文で説明するが、注目した部分空間の外形や隣接部分空間を高速に検索する回転サーチと呼ぶ手法^{1),2)}を可能にするようなテーブル構造にする必要がある。本方式はこの要求にこたえる一つの解である。

交叉点を表現することにより、表現可能な図形の範囲も広がる。すなわち二次元では従来の点、折れ線、凹凸・穴を含む多角形等に加えて、重なりを含むステンドグラス状多色図形の表現が可能になる。また三次元では上記および凹凸多面体等に加えて、物体間の界面等表裏の定義されない面や、穴・多重相貫を含む多材質モザイク状多面体等の表現が実現する。

本論文第二章ではスペースモデルの概念と、二次元図形を表現する頂点テーブルの基本的考え方を述べ、第三章ではリスト構造を併用した三次元頂点テーブルへの拡張について説明する。本論文で一貫して注意したことは、同一点で交叉する線の本数や、同一線で交叉する面の数等に実用上支障のない程度の制限をつけ、かつ多少の空欄を許容すれば、全データを基本的に固定長テーブルで表現できるようにしたこと。および、辺・面・空間をすべてポイントで表現したことである。これらは処理の効率化と図形部分修正の容易化に役立つと考えられる。

2. 二次元スペースモデル用頂点テーブル

2.1 スペースモデルの基本原則

図1に示すように、頂点の周りの凹な（辺に挟まれ

† Study on Vertex Table for Space Model, Capable to Represent Multiple Penetration of Figures by AKIRA OHSAWA (Musashi Works, Hitachi Ltd.).

†† (株)日立製作所武蔵工場

た角度が 180° より大な) 空間を, 頂点を通り Y 軸に平行な仮想的境界線 B_x (図中 1 点鎖線) で区切り*, これと各図形の辺 B_y によって全体を部分空間 $\omega_0, \dots, \omega_i, \dots$ に分割する. この場合 B_x が B_y を突き抜けることはないとする. 空間を頂点を通る B_x と B_y とで区切ったのであるから, 各 ω_i はその X^+ , X^- 端 (X 座標値最大および最小の位置) に必ず頂点を持つ (辺と辺の交点も頂点とする). その頂点にそれぞれ空間** に対応するポイント SP (Space Pointer) を設置し, ω_i の両端の SP が相互に相手の SP のアドレスを指すようにする. この SP の対を SPP (Space Pointer Pair: 図中矢印付細線) と呼び, SPP を設置した全体の図形データ・ファイルのことを SPF (Space Pointer File) と名付ける.

X 座標値の等しい 2 個以上の頂点が存在する時には, それらが X 方向に相互にごくわずかずれており, それぞれの頂点を通る B_x の間に無限小幅の空間が存在すると考えれば, どの ω_i の左右端もそれぞれ 1 個で, かつ 1 個に限る頂点が存在すると言える. これにより, 各 ω_i には必ず 1 組の SPP _{i} が 1 対 1 で対応することになる. 1 対 1 なので, SPP _{i} が与えられたとき, SPP _{i} から対応する ω_i の持つ情報を求めるのが容易なら, SPP _{i} は ω_i を表現していると言ってよい. SPP _{i} が ω_i を表現しているなら, 各種図形処理における図形 (空間) 検索問題は SPF の中から対応する SPP _{i} を検索する問題に置換できる. ここで ω_i の持つ情報とは, ω_i の外形線となる B_x と B_y , これらに接する X, Y 側隣接空間, 色, 図形名等である. 例えば図 1 で SPP₁ が与えられたとき, ω_1 の X^+ 側外形 B_{x1} や, 隣接空間を表現する SPP₀ が, ポインタをたどって容易に検索できることは明らかであろう. また Y^+ 側外形 B_{y1} は, SPP₁ から図中白矢印 $\curvearrowright$ で示すようにポインタをたどる回転サーチと名付けた手法により求められる. 回転サーチの B_{y1} への到着は, サーチ途上で通過する頂点の X 座標値を ω_1 の範囲と比

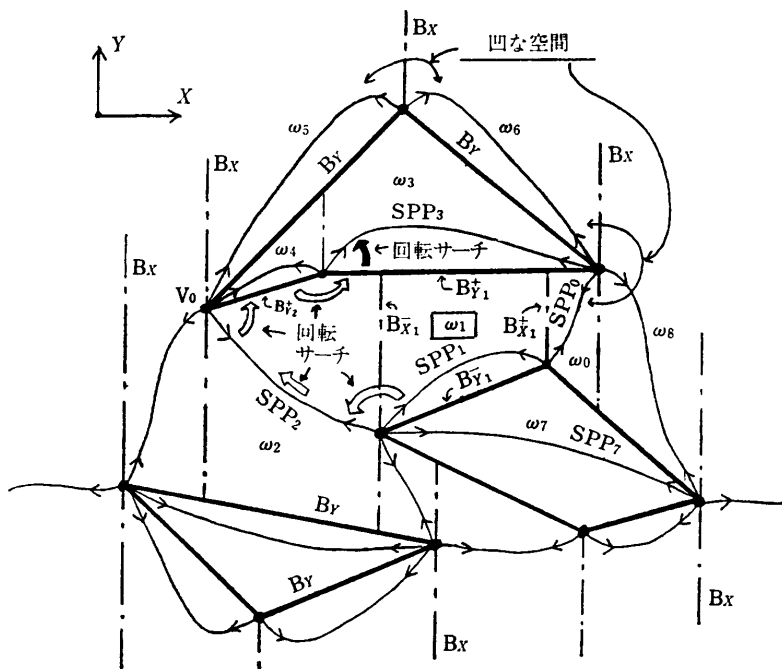


図 1 SPF のデータ構造 (空間 ω_i の SPP _{i} による表現) と隣接空間の検索
Fig. 1 Representation of ω_i by SPP _{i} and retrieval of adjacent subspace.

較していれば判定できる. 回転サーチは SPP の接続する頂点 (同図 V_0) が極端に遠方にあるような例外的図形を除けば局所的であり, 全体図形数に無関係な $O(1)$ の処理となる. なお, 有限な図形なら V_0 は必ず存在する. Y^+ 側隣接空間を表現する SPP₃ も, B_{y1} からの回転サーチで検索できる. X, Y 共に隣接空間が検索できるから, 隣接空間伝いに任意の場所へ注目点を移動させることもできる.

色や図形名等は図 2 図 2 のごとく SPP に空間属

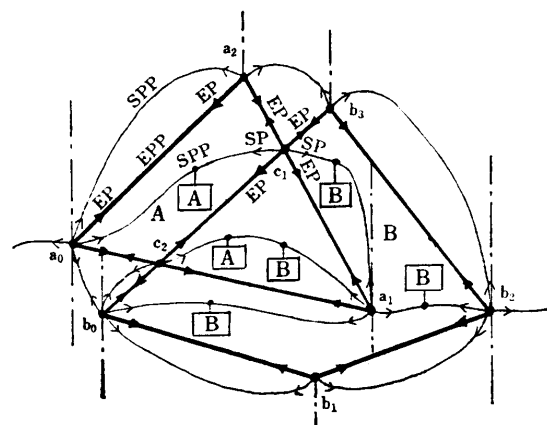


図 2 空間属性 図 2 のごとく SPP に空間属
Fig. 2 Concept of overlap representation by Space Attributes 図 2.

* ここでは図の煩雑化を避けるために B_x を凹な空間に限定したが, 凸な (180° 以下の) 空間にもすべて B_x を設置する方式もある. どちらにしても原理上問題はない (文献 1 の付録 1).

** 以下部分空間のことを単に空間と呼ぶこともある.

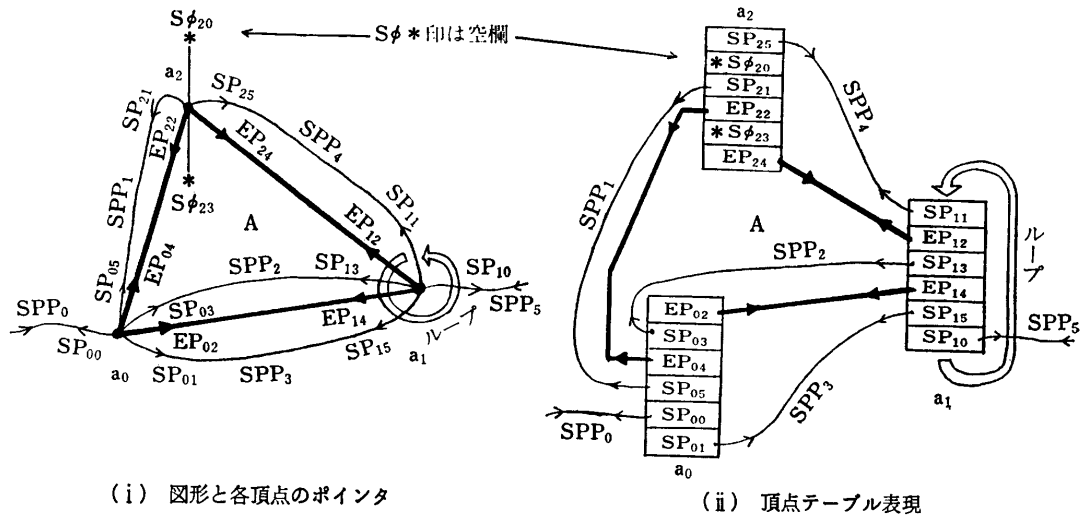


図 3 回転サーチの効率化を実現するループ状頂点テーブル
Fig. 3 Loop style Vertex Table realizes effective Rotary-Search.

性として付加する。図形の重なりは空間属性の重なりで認識できる。SPF では図形辺もポインタ EP (Edge Pointer) の組 EPP (EP Pair) で表現し、変更を容易化している。SPF 構造の生成時間は例外的な場合を除き、総頂点数 N に対し $O(N)$ となる^{1),2)}。

2.2 二次元 SPF におけるループ状頂点テーブル

図 3(i), (ii) に各図形頂点に設置して図形データを表現する頂点テーブルを示す。同図(ii)の頂点テーブルには、同図(i)に対応して辺を表現する EP と、空間を表現する SP とが頂点まわりの並び順に合わせてループ状に設置してある。各 EP, SP はそれぞれ他の頂点の EP, SP と接続（相互に相手のアドレスを指す）して EPP, SPP を形成している。図中 $S\phi$ はテーブルを固定長化してメモリ管理を容易化するためにダミーとして設置した空欄である。この方式によれば、図 1 に白矢印で示した回転サーチに必要な、頂点周りで隣り合う EP, SP の検索は、隣りの欄を見る ($S\phi$ は読みとばす) ことで高速に実行できる。また辺や空間がすべてポインタで表現されているので、図形追加・削除・変形等に伴う更新が簡単な利点もある。図 4 は図形回転時の部分空間の変化に伴うポインタの更新が、一部の SP および $S\phi$ の指し先の変更だけで容易に可能であることを示す例である。なお

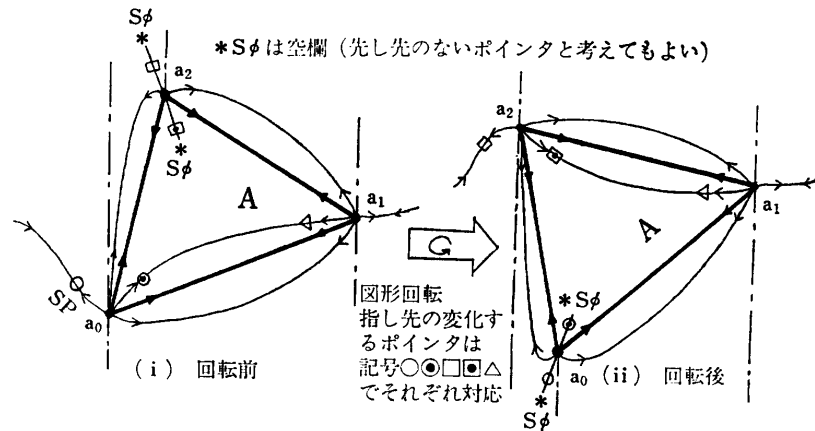


図 4 図形回転時のポインタ修正
Fig. 4 Pointer update corresponding with figure rotation.

頂点テーブルにはこれ以外に頂点座標値 x, y や, EP, SP に付加する空間属性、辺属性（色、図形名など）等も設置するが、図 3(i), (ii) では図示を略してある。

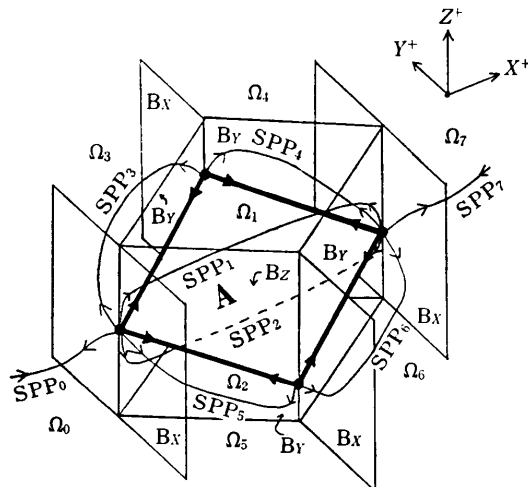
前出図 2 のごとく図形に交叉がある場合、辺の交点 c_1, c_2 も一種の頂点として頂点テーブルで表現する。図 2 に示すように交点に設置すべきポインタは 2 個の SP と 4 個の EP, 合計 6 個である。同一点で三本以上の辺が交叉する「複雑な交点」は、二本の辺による交点が複数個近接したものとして取扱う。この方法によれば、交点を含むすべての多角形頂点は、ポインタ 6 個の固定長頂点テーブルで表現できることになる。

3. 三次元スペース・モデル用頂点テーブル

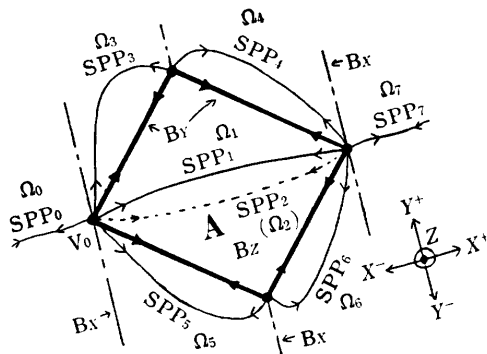
3.1 原理と頂点テーブルへの要求事項

図5(i)に三次元空間分割と部分空間 Ω_i を表現するSPP $_i$ を示す。三次元空間は図形頂点を通り X 軸に垂直な仮想面 B_x 、図形辺を含む Z 軸に平行な仮想面 B_y 、および図形面自体 B_z によって部分空間 Ω_i ($i=0, 1, 2, \dots$)に分割され、 Ω_i と1対1でこれを表現するSPP $_i$ が設置される。これを Z 軸+側から眺めたのが図5(ii)であり、二次元の空間分割と類似した形となる。この方式によれば二次元の場合と同じく隣接空間や重なり図形の検索が局所的処理で可能となり、図形集合演算をはじめ各種図形処理の高速化が実現する²⁾。

以下三次元図形ファイルSPFにおいて、図形を具体的に表現する頂点テーブルへの要求事項を列挙する。



(i) 斜視図



(ii) Z^+ 方向視図

図5 三次元空間の面Aによる分割と対応するSPP
Fig. 5 3-D space partition by plane A, and corresponding SPPs.

る。

- (1) 任意枚数の面の交叉により成る辺が任意本数集まった頂点の表現ができること。これにより図6の多重相貫体、図7のような凹空間・凹面を含む多種材質モザイク図形等の表現が可能になる。注意すべきことは、これらの図形では物体の界面のような、表裏の定義できない面の表現が必要なことである。曲面も重要であるが今回の検討からは外した。
- (2) 頂点座標値のほか辺EP、面FP、空間SP等の各ポイントが設置できること。ここで初めて出てきたFP (surFace Pointer) は、図8に示すように模様や色を含む物体表面を、物体内部の空間とは別のもので表現するためのポイントである。
- (3) 部分空間の外形や隣接空間を局所的処理で検索するための回転サーチが効率的にできること。三次元の場合には、与えられた空間の Y 側外形や隣接空間を求める Z 軸まわり回転サーチと、 Z 側のそれを求める Y 軸まわり回転サーチの二種がある。

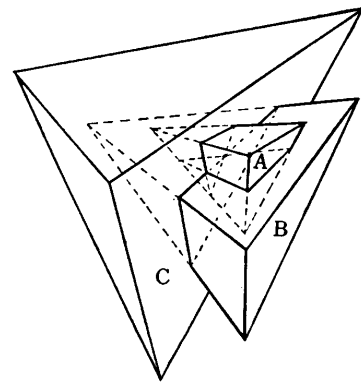


図6 多重(三重)相貫図形の例
Fig. 6 Multiple (Triple) mutually penetrated figures.

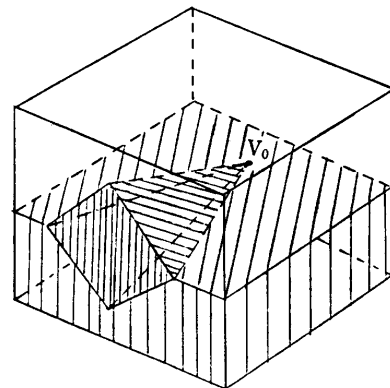


図7 凹空間を含む多種材質頂点 V_0
Fig. 7 Multi-material vertex V_0 with concave spaces.

(4) 三次元 SPF 空間分割に特有の影の交叉線の表現が可能のこと。3.8 節で説明する。

(5) 空間 (SP), 面 (FP), 辺 (EP) 等にはそれぞれ空間属性, 面属性, 辺属性が設置できること。これは二次元 (図 2) と同様である。図 9 に原理的に示した重なり表現を図 6, 図 7 の表現と組合せれば多重包含を含む多重相貫体の表現が可能になる。

(6) 図形追加・削除・加工・移動・回転等に伴うデータの更新が容易なこと。これは辺・面・空間をすべてポイント EP・FP・SP で表現し, かつ頂点テーブルを可能な限り固定長とすることで達成される。

(7) コンパクトなこと。

等である。これらのうち特に重要なのは(3)回転サーチの効率的サポートである。以下これを重点に, 最初にリスト構造による三次元頂点テーブルの原理を説明し, そこから実用的なテーブルの構成法を誘導する。

3.2 リスト構造による頂点テーブルの原理説明

図 10 に SPP_1 から Y^+ 側境界面 Br_1^+ および隣接空間の SPP_4 を求める, 三次元 Z 軸まわり回転サーチを示す。必要なことは同図白矢印 $\curvearrowright$ や黒矢印 $\curvearrowleft$ の, 頂点において回転的に隣り合うポイント間の検索が容易な頂点テーブルの構造をつくることである。2.2 節で示した二次元の場合には, 各 EP や SP の (回転的) 左右隣には, SP や EP が必ず 1 個ずつしか存在しない。このため, 図 3 に示した 1 列ループ状の頂点テーブルにより, 隣りの項目を検索することで回転サーチの要求を実現できた。しかし三次元では EP で表現される辺の左右*に複数の SP, FP が Z 方向上下に重なって隣接する形となるため同じ方式はとれない。例えば図 11(i) は凸な三面のつき合わせによる凸三面頂点を Z^+ 側から見た図であるが, EP_0 の辺の左側には SP_{01}^+ , FP_{01}^+ , SP_{01}^+ , FP_{02}^+ , SP_{02}^+ , で表現される合計 5 個の空間と

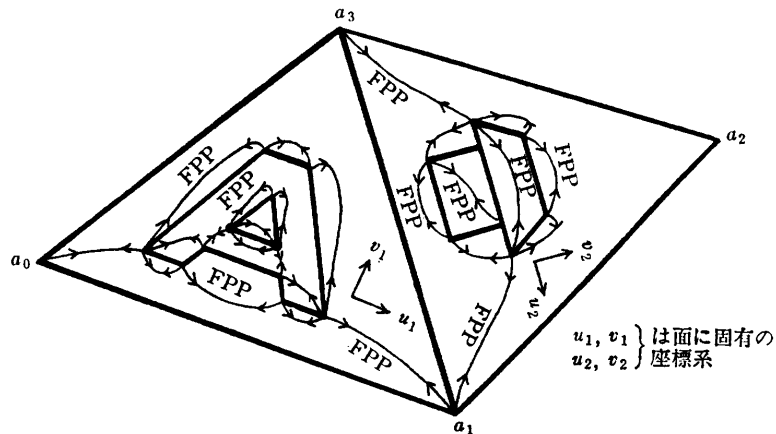


図 8 物体表面の図柄の FPP による表現
Fig. 8 Representation of the figures on the surface of solid by FPPs.

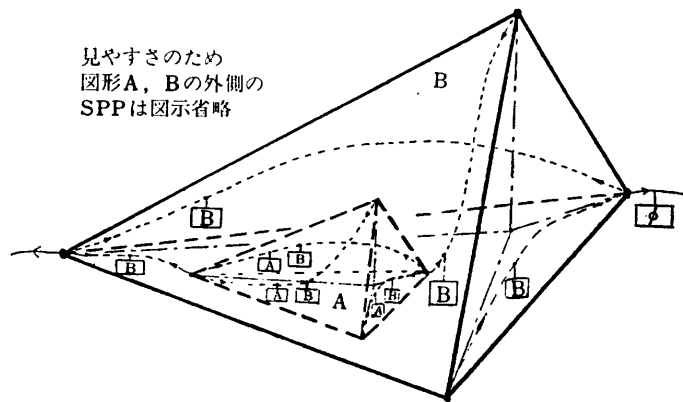


図 9 空間属性 (図面) の SPP への付加による重なり図形の表現
Fig. 9 Representation of overlapped figures by attaching space attribute (図面) to SPPs.

面が直接に隣接している。三次元頂点テーブルの主眼点は, このような 1 対複数の隣接関係をリスト構造を利用して検索に便利な形で表現することである。なおここでポイントの記号について触れておく。 FP_{ij}^z , SP_{ij}^z の肩の添字 z は, EP_i で表現される辺の左側に隣接している面 (FP) または空間 (SP) を Z^+ 側から数えた重なり順を表す番号 ($z=0, 1, 2, \dots$) である。また右下の添字 i と j は, それが辺 EP_i を含む仮想面 Br_i と EP_j を含む Br_j の間に挟まれた位置にあることを示している。例えば SP_{01}^+ は EP_0 を含む Br_0 と EP_1 を含む Br_1 の間にあって Z^+ 側から 3 番目に (最上位は SP_{01}^+) ある空間を表現する。また同じく図 11(i) には SP_{02}^+ , SP_{03}^+ , SP_{04}^+ のみしか出てきていないが, 一般には SP_{ij}^z , SP_{ij}^z , SP_{ij}^z , FP_{ij}^z , FP_{ij}^z , FP_{ij}^z と表記されて

* 文中, 上下, 左右という時は, 視点を当該頂点に置き, Z^+ 方向を上にして対象となる辺や空間を眺める。

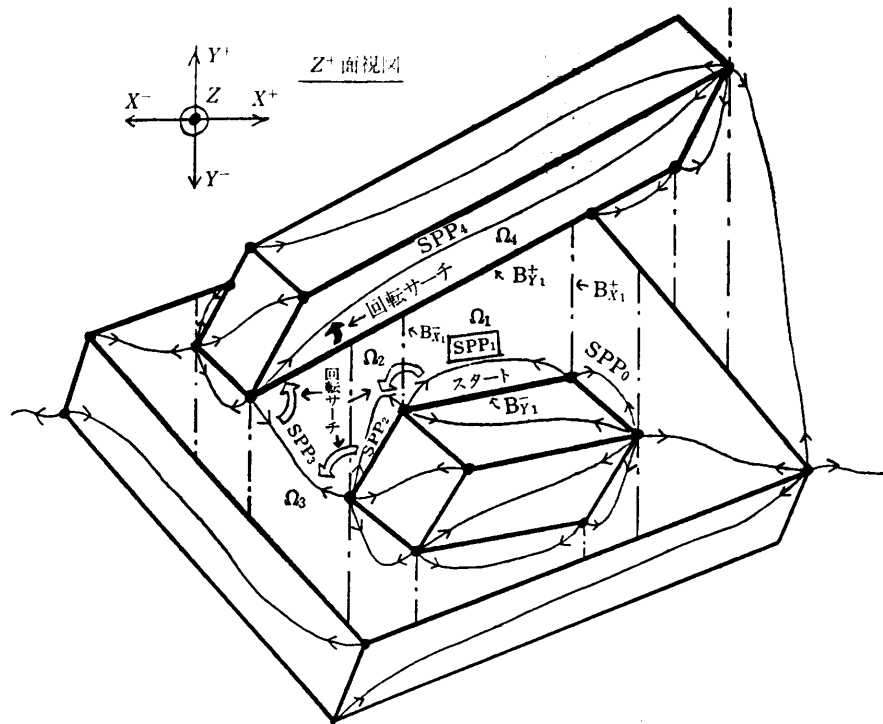


図 10 Y 側外形と隣接空間を求める Z 軸まわり回転サーチ
Fig. 10 Z-axis rotary-search to retrieve Y boundary & abutting subspace.

凹空間・凹面を表現するポイントが存在する。この場合 z および i, j は上記と同じ意味であり、さらに付加されている肩の添字 r, c, l はそれぞれ頂点から凹空間(面)に向かって右側(right), 中央(center), 左側(left)にある部分空間(面)を表現している。例えば図 11(i)で SP_{10} は EP_2 と EP_0 の二辺に挟まれて、 Z^+ 側最上位にある空間($z > 0$ はこの図の場合には存在しない)のうち、頂点 V から凹空間に向かって右側(図では下側)にある部分空間を表現している。なお以下では煩雑さを避けるため、特に注目しない添字は適宜省略する。

さて、前述の 1 対複数の隣接関係をリスト構造で表現する方法について述べる。図 11(ii) は同図(i)の AA' 断面図であり、各 EP, FP, SP の隣接関係をその間を結ぶ矢印で示したものである。ここでこれら矢印を EP, FP, SP 相互間を接続する頂点テーブル内の付加的ポイントと考えれば、これはリスト構造の頂点テーブルそのものである。図中 LP^* (Left edge Pointer) は SP^*, FP^* に付加してその左側の EP を指し、 RP^* (Right space/surface Pointer) は EP に付加してその右側の SP^*, FP^* を指すテーブル内ポイントである。この方式によれば、隣接 EP, FP, SP 間の相互検索は、 z 値を与えることで容易に実行できる。 z

値は回転サーチを今どの z 値の面または空間で行っているかにより決まる。なお本件は 3.9 節で補足する。

リスト構造はフレキシブルであり、1つの頂点に任意個数の辺が集まり、1つの辺が任意枚数の面の交叉より成る一般的な頂点に拡張することも可能である。この方式の頂点テーブルを各頂点に対応して設置すれば原理的には三次元 SPF は完成する。しかしこの方式のリストでは、付加的ポイントのためにメモリ所要量が多く、また検索時にポイントをたどる回数が多いなど、効率上の問題点がある。次節ではこの問題を緩和する手法につき述べる。説明は最初に基本的な凸三面頂点について行い、その後一般化する。

3.3 凸三面頂点の頂点テーブル

図形が回転すると SPF の空間分割が変化し、それに伴って EP, FP, SP の隣接関係も変わってくる。図 11(ii) のリスト構造では、回転への対応はポイントの追加・削除と指し先変更で可能であるが効率が悪い。一方頂点テーブルを固定フォーマットのテーブル構造にすれば効率は良いが、変化への対応には無理がある。このため破線で囲った EP_i ブロックごとに部分的にテーブル化し、ブロック間をポイントで接続したテーブルとリストの折衷方式を検討する。 EP_i ブロックをテーブル化したものが同図(iii)の EP_i テーブル

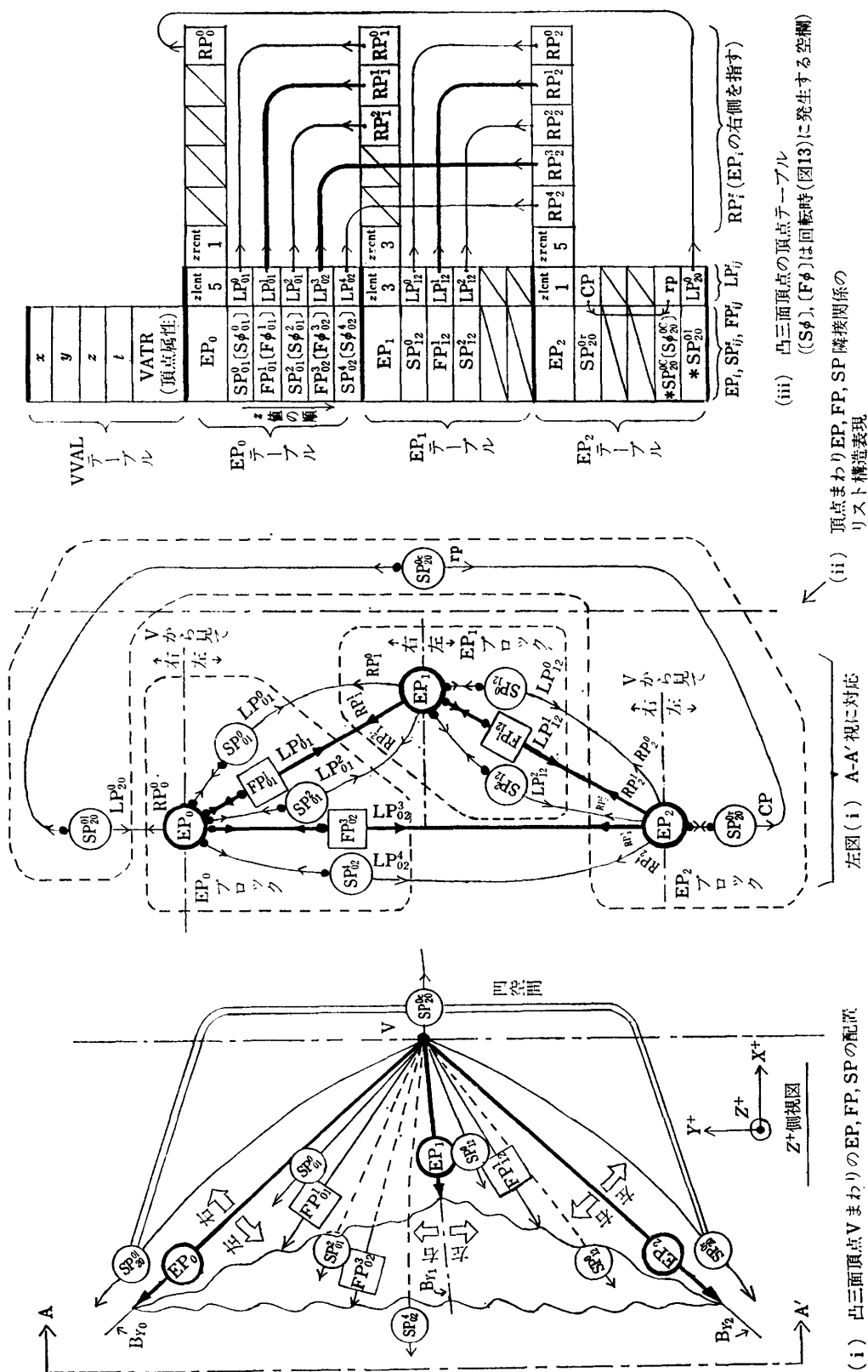


図 11 凸三面頂点における EP, FP, SP 隣接関係のリスト表現と頂点テーブル

Fig. 11 List expression of EP, FP, SP neighboring relationship & Vertex Table.

である。EP_iテーブルの基本フォーマットは、EP_iの左側のSP^{*}, FP^{*}, および凹空間・凹面がある場合にはSP^{rr}, FP^{rr}, をzの順に並べて設置することである。図中[Sφ], [Fφ]は次節で説明する。EPテーブル内では項目間の相対位置が固定されるから、図11(ii)でブロック内項目間を結合していた黒丸矢印 $\bullet \rightarrow \bullet$ のポインタは不要になる。EPテーブルのサイズは辺の左側と面のZ方向重なりが最大となるEP₀ブロックに合わせ、余りが出れば空欄にしておく。図のごとくEPテーブルを辺の数だけ並べ、頂点座標値等を含むVVAL (Vertex VALue) テーブルを加えたものが頂点テーブルである。凹空間凹面がある場合には、これ表現するSP^{rr}, SP^{rl}, FP^{rr}, FP^{rl}は頂点テーブルの中で上記空欄を探して設置する。同図(i) EP₂の左側の凹空間については、SP^{rr}, SP^{rl}は同図(ii)*印の位置が(この設置前は)空欄であったので、そこに

設置してある。この場合SP^{rr}とSP^{rl}, SP^{rr}との間はポインタ cp, rp によって相互に検索できる。空欄を探して設置する手法は、この図の場合には煩雑だけで意味がないが、3.5節の凹三面頂点の場合に役立つ。

EPブロック間(EPテーブル間)の接続は、図示のごとくLP^{*}, RP^{*}の指し合いで行われる。同図(iii)ではRP^{*}はEPテーブルから突出した形で図示したが、見やすさのため以上の意味はない。なお、LP, RP, cp, rpはテーブル内アドレス差分として、それぞれ1バイト以下でコンパクトに記述できる。EPに付加したzlcntは辺の左側の空間と面のZ方向への重なり数、zrcntは右側の空間と面の重なり数である。辺や空間の属性、その他詳細項目は図示を略した。

3.4 凸三面頂点の回転

図12は、図11(i)の頂点をZ軸まわりに回転

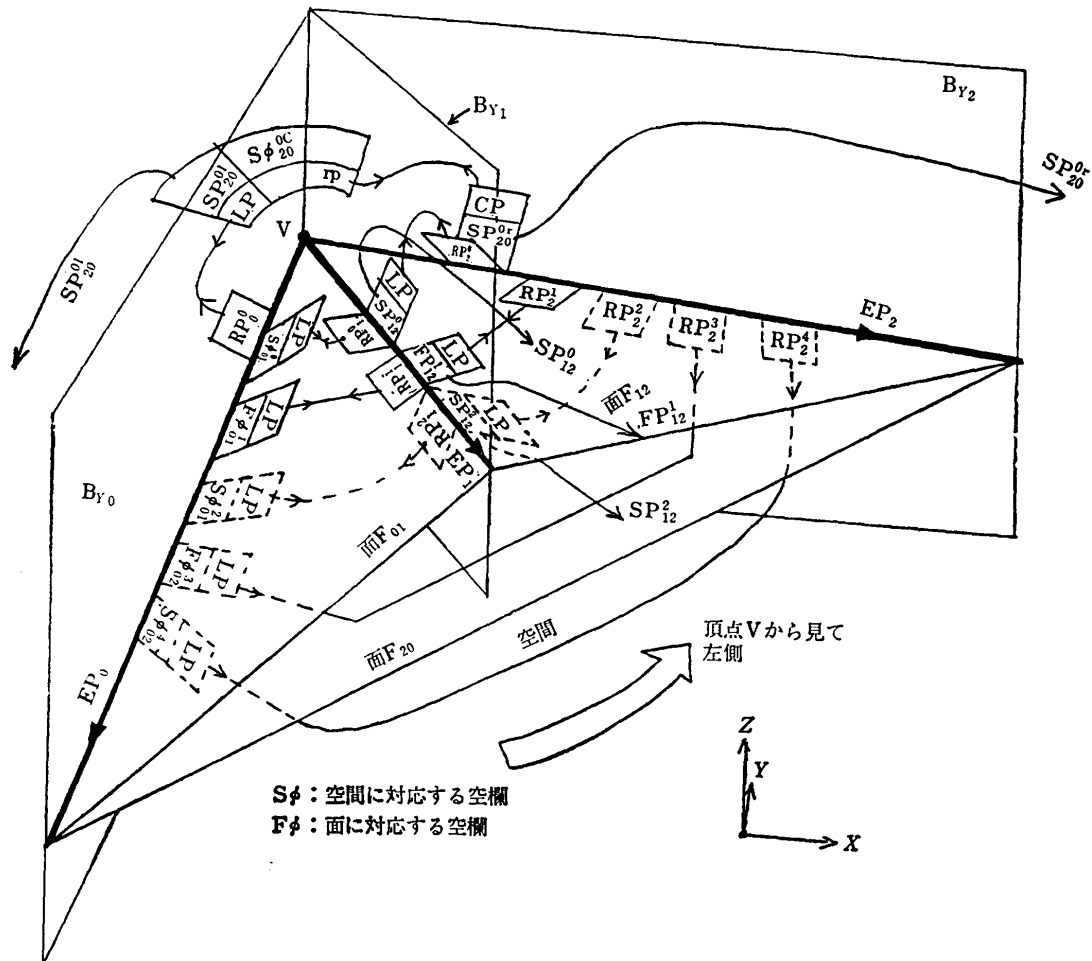


図12 図11(i)をZ軸まわりに回転
Fig. 12 Rotate Fig. 11 (i) around Z-axis.

し、各種ポイントを名前で表示したものである。Z 軸まわりの回転では、二次元の場合（図 4）と同じく、SP や FP の一部を空欄 $S\phi$, $F\phi$ に置換する。これは図 11(ii)において対応する SP, FP を [] 内の $S\phi$, $F\phi$ に替えることで実現できる。 $S\phi$, $F\phi$ は空欄ではあるが、頂点周りにある空間や面の存在を表現していると考えられる。その意味で、 $S\phi$, $F\phi$ を SP, FP と等価であると考え、Z 軸まわりの回転に関しては頂点テーブルの構造は不変（SP の指し先は変化する）と言える。

図形の一般的な回転では EP テーブルの構造は不変にはならないが、プログラム効率からは、テーブルサイズが一定であるのが望ましい。図 12 の凸頂点を三次元的に回転させると、各辺の左右にある空間と面の Z 方向重なり数が増える。しかし上述のように $S\phi$, $F\phi$ を SP, FP と等価と考えると、頂点が Z 方向に向く場合を含み、どのような回転に対しても辺の左右の SP, FP ($S\phi$, $F\phi$) の数は同図の EP_0 , EP_1 , EP_2 のどれかと必ず等しくなる。したがって多少の空欄を許せば EP テーブルサイズは一定でよく、頂点テーブル全体も固定サイズにできる。

3.5 凹三面頂点の頂点テーブル

最初に図 13(i) に示す凹三面頂点が、前述の凸三面頂点と同一コンセプトで、同一サイズの図 13(ii) に示す頂点テーブルにより表現できることを示す。図 13(i) の図 11(i) との相異は、 EP_0 の辺の左側に凹空間・凹面を表現する SP^{0r} , SP^{0e} , SP^{0l} , FP^{1r} , FP^{1e} , ... 等合計 9 個のポイントが存在することである。このため EP テーブルのサイズを図 11(ii) の凸頂点と同じにすると EP_0 テーブルは溢れてしまう。しかし、この場合には EP_1 テーブルと EP_2 テーブルに空欄ができるので、3.3 節で説明したように凹空間、凹面に対応する SP^{*r} , SP^{*l} , FP^{*r} , FP^{*l} を、これらの空欄に設置できる（図 13(ii)*印）。以上により図 13(i) の頂点テーブルが凸三面頂点の場合と同一サイズになることが分かった。

次に任意形状・方向を持つ凹三面頂点が、すべて同一サイズの頂点テーブルで表現できることを示す。そのためには次の二つの命題が正しければよい。

- (1) 図 14(i)~(vi) で示す凹三面頂点、またはこれらと XY 平面に関し面対称な形状の頂点はすべて前出の凸三面頂点と同一サイズの頂点テ

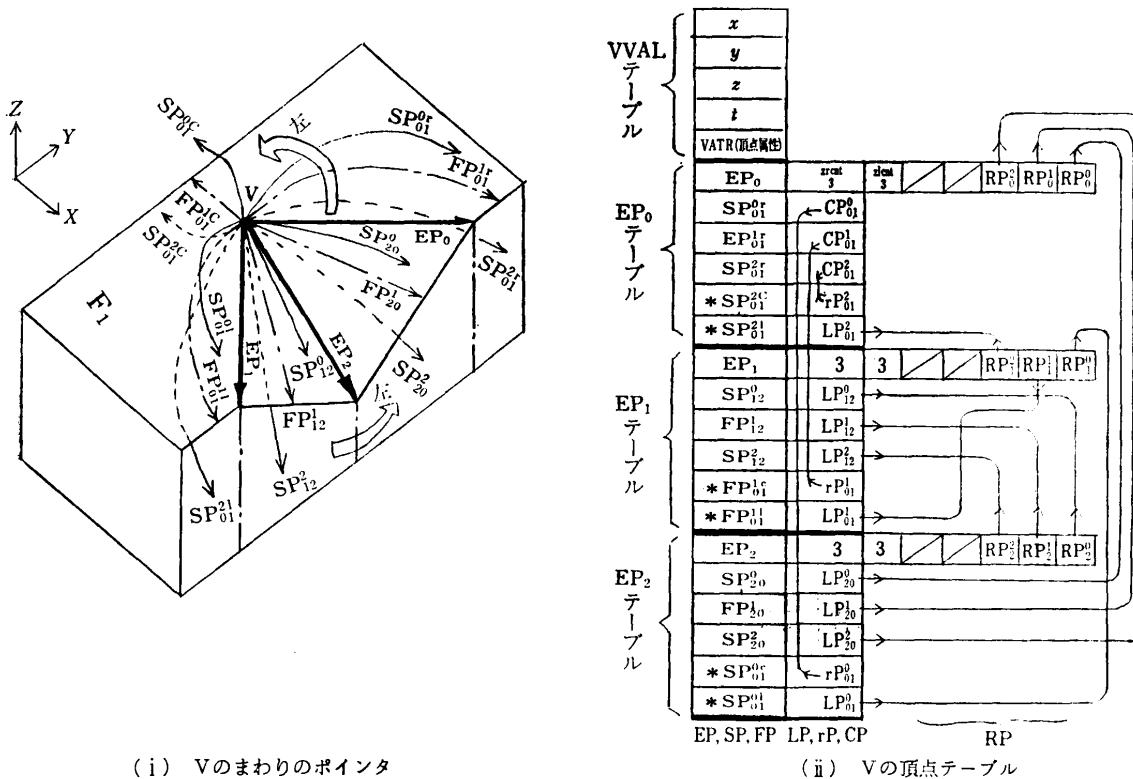


図 13 凹三面頂点 V のまわりのポイントと頂点テーブル

Fig. 13 Pointers around concave vertex V of polyhedron, & Vertex Table.

ブルで表現できる。

- (2) 任意の凹三面頂点は、これら図14(i)~(vi)またはそのXY平面に関する面对称形のどれかと、必ず同一構造の頂点テーブルで表現できる。

図14(i)~(vi)は図中一点鎖線で示す凹面の中心線 c がY軸に垂直な面の X^- 側半平面内にある凹三面頂点を、 Z^+ 側から見た各種形状の代表例である。代表例というのは、各 EP_i の左側のSP,FPの並び方が同じ形の頂点では頂点テーブルが同一になるので、違うもののみを挙げたということである。頂点テーブルは凹面の中心線 c が上記 X^- 半平面内にある限り、凹面の傾き、凹角($\angle EP_0 \cdot EP_1$)によらず同様に、すべて同図(i)~(vi)のどれかで代表される。またXY平面に関する面对称形も原理同様なので、以下の説明は省略する。

さて、上述の命題(1)を説明しよう。これらの図には EP_0, EP_1, EP_2 の左側に存在するSP,FP, $S\phi, S\phi$ の合計数が⑨、③、①等で表示されている。このうち図14(i)は図13と同一形状を示したものである。図13は本節前半で説明したように EP_0 の辺の左側に計⑨個のSP,FPがあり、 EP_0 テーブルは溢れるがその溢れ分は EP_1, EP_2 テーブルの空欄に設置された。その様子を図14(i)細線矢印で示した。他の同図(ii)~(vi)についても EP_0 の左側のSPとFPを数えると必ず⑨であり、 EP_0 テーブルは溢れるが、どの場合にも EP_1, EP_2 テーブルには空欄があり溢れ分が収容できる(各EPテーブルには合計5個までのSP,FPが設置できる)。以上で命題(1)の説明が終った。

次に命題(2)について述べる。与えられた任意の凹三面頂点は、Z軸まわりの回転により凹面の中心線をY軸に垂直な面の X^- 半平面まで動かせば、図14(i)~(vi)のいずれかと合わせることができる。この回転に際し、前述のごとく $S\phi, F\phi$ をSP,FPと等価と考えれば、頂点テーブルの構造は不変である。したがって、与えられた凹三面頂点は、図14(i)~(vi)のいずれかと同じ構造の頂点テーブルで表現できると言える。

以上、命題(1)、(2)により、任意の凹三面頂点は凸三面頂点と同一サイズの頂点テーブルで表現できることが分かった。

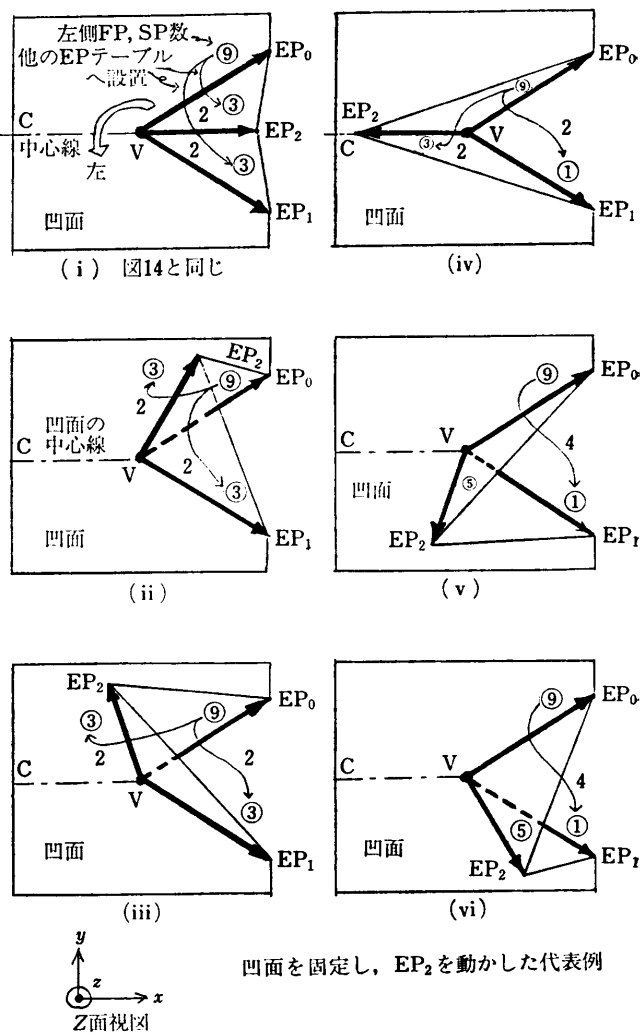


図14 凹面を持つ頂点の全基本形状と溢れた SP^{ac}, SP^{af} 等の他のEPテーブルへの設置

Fig. 14 All basic forms of vertices which include a concave surface.

3.6 貫通点の頂点テーブル

図15は図形のコーナー(辺)が面を貫通する貫通点、図16は三面の交叉点であるが、これらも一種の頂点として頂点テーブルで表現する。この場合には頂点に集まる辺の数がそれぞれ4個、6個なので、各辺に対応してEPテーブルの数を増やす必要がある。しかし、すべての辺は2面の交叉線であるから、辺の左右の面と空間のZ方向重なり(最大値が2面と上中下3空間、合計5を越えることはない。このためEPテーブルのサイズは前出の凸凹三面頂点と同じでよい。これに対し、EPテーブル数の増加は、これを並べた頂点テーブル全体が凸凹三面頂点と同じ固定サイズに収まらない問題を起こす。1つの対策は、メモ

リ・アロケーション・プログラムの工夫で、標準的な凸凹三面頂点のサイズと共にその2倍サイズのテーブルも許すようにする方法である。この場合、貫通点の頂点テーブルは図15(ii)、図16(ii)のごとく2倍サイズに統一し、余りは空欄にする。貫通点の出現頻度が低ければ多少の無駄は許容できよう。その他、次節に示す任意頂点として処理する方法もある。なお図中PEP (Penetration Edge Pointer) は貫通辺EP間の相互検索を容易にするテーブル内ポインタである。貫通点の表現に空間属性による重なり表現を加えれば、多重包含を含む多重相貫体の表現が可能になる。

3.7 任意頂点の頂点テーブル

頂点テーブルはZ方向最大重なり数によってサイズを決めたEPテーブルを、辺の数だけ並べてVVALテーブルを加えたものであるから、任意頂点の表現は本来可変長テーブルになる。しかし、可変長テーブルはメモリ管理が複雑化して効率が悪いから、実用的には固定長化を図るのが望ましい。三次元空間における頂点は一般に三面の交点で、辺は二面の交叉による基本形で表現できる。すなわち、四面以上が1点で交叉する「複雑な頂点」や三面以上が交線を共有する「複雑な辺」等は基本形が接近隣接したものと考えてよい。このため三面頂点への分解アルゴリズムを適切に採用すれば凸・凹三面頂点の組合せで任意頂点が表現でき、固定長化が可能になる。もちろん、曲面の四辺形パッチ等で四面頂点を多用するような時には、専用テーブルを同じ考えで作成するのがよい。

もう1つの方法は、頂点に集まる辺を表現するEEテーブルを独立させて別テーブルとし、図17のように表現することである。これによれば頂点へ集まる辺数の制限はなくなる(三面以上が同一線で交叉する複雑な辺の問題は解消しないが、出現頻度が少なく実用上は困らないと思われる)。しかし反面、LP, RP, cp, rpをテーブル間を指すフルサイズのポインタにする必要があるため、メモリ容量が増加することは避けら

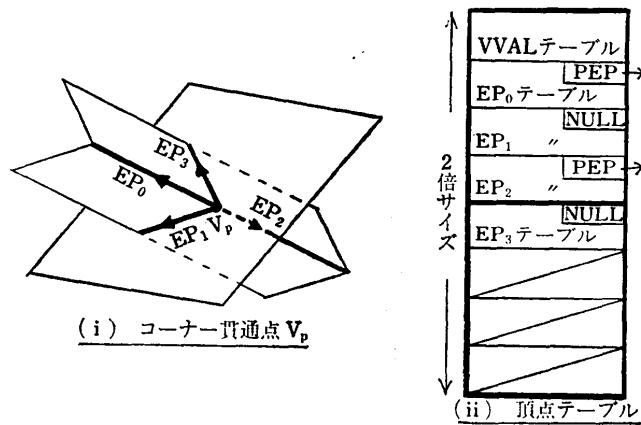


図15 コーナー貫通点 V_p と頂点テーブル
Fig. 15 Corner penetration point V_p & Vertex Table.

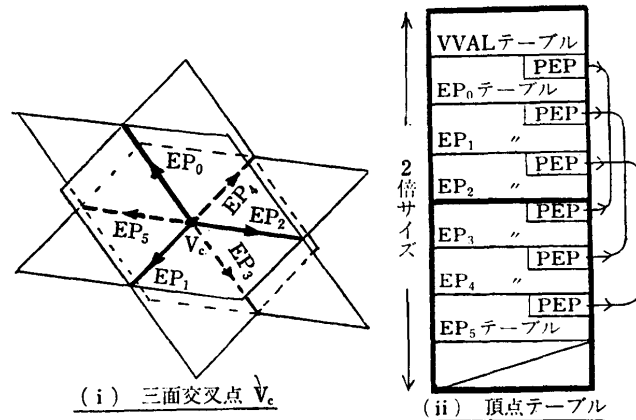


図16 三面交叉点 V_c と頂点テーブル
Fig. 16 Three planes cross-point V_c & Vertex Table.

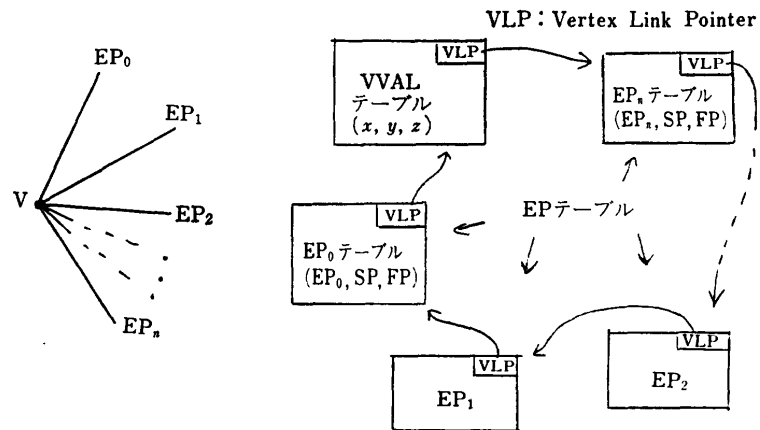


図17 EPテーブルによる頂点の表現
Fig. 17 Representation of vertex by EP table.

れない。

三次元空間における面、折れ線、点、等の頂点テ

ブルも同様に作成できるが説明は省略する。

3.8 影の交叉の頂点テーブル

影の交叉とは、図 18(i) のように Z 方向には離れているが、同図(ii)のごとく Z^+ 側から見れば交叉している形状である。影の交叉を直接に構成する空間は同図影の交叉線 cc' でつぎ合わせになる $\Omega_{01} \sim \Omega_{30}$ のみであり、 A 面またはその上側や B 面またはその下の空間は直接には関係しない。直接に関係する空間は Z^+ 側から見れば同図(ii)のごとく cc' のまわりにループ状に相互に隣接して並んでいるため、二次元の場合と同様に同図(iii)に示すループ状の頂点テーブルで表現できる。なお、同図(iii)には先に図 12(iii)で示したような VVAL テーブルが示していないが、影の交叉の座標値 (x, y) は二辺の両端の座標値から計算できるため、メモリ節約の目的で省略したのである。影の交叉は出現頻度が高いから、メモリ管理は複雑化するが三面頂点と同一サイズの中にこれを複数個設置してメモリを節約するのがよい。もちろん各 EP テーブルを独立させた図 17 の方式では、影の交叉の頂点テーブルを一般の EP テーブルと同一サイズにすることで問題を回避できる。図 18(iii)のテーブルの各 EP_i に付加された項目 z_{ci} は、 EP_i の左側で影の交叉をする空間の Z^+ 側から数えた重なり番号で、この図の場合には z_{c2} のみが 3 である（その他例えば EP_3 の左側空間 Ω_{30} は、 EP_3 の左側隣接面の中では Z 方向最上位なので、 $z_{c3}=0$ である）。 z_{ci} は 3.9 節で示す回転サーチに際して、影の交叉がサーチ中の空間に関するものかどうかをプログラムに知らせる役目をする。影の交叉の頂点テーブルは異なる図形間際に挿入するが、すべてポイント構成であるため追加・削除は容易である。

3.9 Z 軸まわり回転サーチに関する補足

図 19 白矢印 $\rightarrow$ ①~④に、図形 A の内部空間を表現する SPP_1 を与えられて、 Y^+ 側外形 B_{Y1^+} を求める Z 軸まわり回転サーチの経路および頂点テーブル上での検索の様子を示す。このうち①、②のサーチはポイントで指されているとおりにたどればよいから簡単である。しかし頂点 a_0 の EP_1 から、これに対応する頂点 a_1 の EP_0 へ行く EP たどり③においては、その後に続く頂点 a_1 における隣接辺の検索が、④の経路で SPP_1 の外形 B_{Y1^+} の一部である EP_2 へ正しく行われるための情報を保持していなくてはならない。間違えて④'等へ行ってはいけな。そのためには同図で示すように、②でたどり着いた RP_1^2 (これは頂点 a_0 の

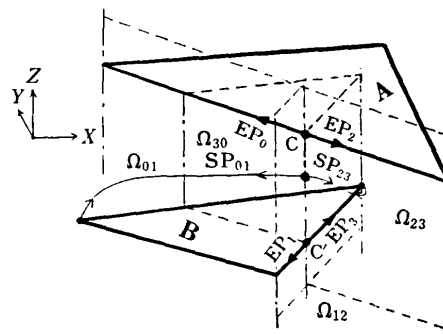


図 18 (i) 影の交叉線 cc' (斜視図)

Fig. 18 (i) Shadow crossing line cc' (perspective).

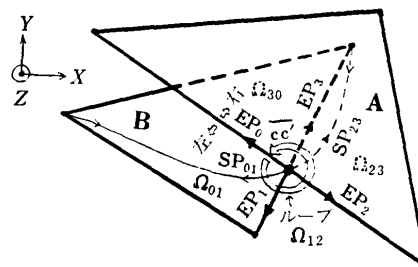


図 18 (ii) 上記(i)の Z^+ 面視図, cc' は点に見える

Fig. 18 (ii) Z^+ view of Fig. (i) above.

cc' looks like a point.

EP ₀	$z_{c0}=0$	PEP ₁	ループ
SP ₀₁			
EP ₁	$z_{c1}=0$	PEP ₂	
EP ₂	$z_{c2}=3$	PEP ₃	
SP ₂₃			
EP ₃	$z_{c3}=0$	PEP ₄	

図 18 (iii) 影の交叉 cc' の頂点テーブル

Fig. 18 (iii) Vertex Table of shadow crossing cc' .

EP_1 の右側空間の回転サーチ中の SPP_1 を指している)の z 値 [$z=2$] を保持し、③でたどり着いた頂点 a_0 の EE_0 テーブルの中から、 $z=2$ を使って SPP_1 と同一空間を表現している $S\phi^2$ を選ばばよい。一般に辺の片端の頂点から眺めたとき辺の右側にあつて RP^* で指される空間(面)は、その辺の他端の頂点から眺めるとその辺の左側になって SP^* , $S\phi^*$ (FP^* , $F\phi^*$) で表現され、 z 値は両者等しい。同一空間(面)の z 値が辺の両端で等しい理由は、辺の途中で空間(面)の Z 方向重なりが変化することは(曲面でなければ)あり得ないからである。

次に同図黒矢印 $\leftarrow$ の回転サーチにおいて、⑥の EP たどりの途中で影の交叉 c に出合ったときの処置について述べる。影の交叉の頂点テーブルは図 18(iii)で

示したように、交叉 c に属する EP_0 の左側で影の交叉を構成する空間の EP_0 に関する z 値を z_{c0} (この図の場合には EP_0 の左側の面の上側空間が影の交叉を構成するので $z_{c0}=0$)として保持している。図の場合には EP たどり⑥で保持している z 値が $[z=0]$ のため、 $z=z_{c0}$ となり、⑦の回転サーチを行う。しかし、もし $z \neq 0$ のような別の空間で回転サーチを行っている場合には $z \neq z_{c0}$ となり、影の交叉と無関係であることが分かるから、 c の頂点テーブルは無視して(辺に沿って影の交叉をパスして)⑧の経路で先に進む、これは具体的には図18(Ⅲ)の頂点テーブルで PEP で指される EP へ移ることで実行される。

3.10 Z側境界 Bz の局所的検索(Y軸周回転サーチ)

本節では、これまでに作成した頂点テーブルを使えば、Z側境界 Bz が局所的処理で検索できることを示す。図20で SPP_1 が与えられて Bz_1^+ を求めるには、 SPP_1 を出発して目的の Bz_1^+ にZ-側から密着する空間の SPP のみを一方(ここでは X^+ 方向)的にどこまでもたどればよい。 Bz_1^+ が有限ならばそのような探索ルートは必ず Bz_1^+ の外周上の頂点(交点を含む)に達し Bz_1^+ が求まる。 Bz_1^+ にZ-側から密着した空間は、 SPP をたどる途中の頂点では必ずZ方向最も上側にある空間で $z=0$ である。したがって、頂点テーブルで常に $SP^{z=0}$ を選びながら X^+ 方向へたどればよい。この条件を満たす道順は、同図に白または破線矢印で示すように複数存在するが、どれを選んでもよい。最終的に Bz_1^+ の外周に到達したことは、 X^+ 向きの SP_1^0 に

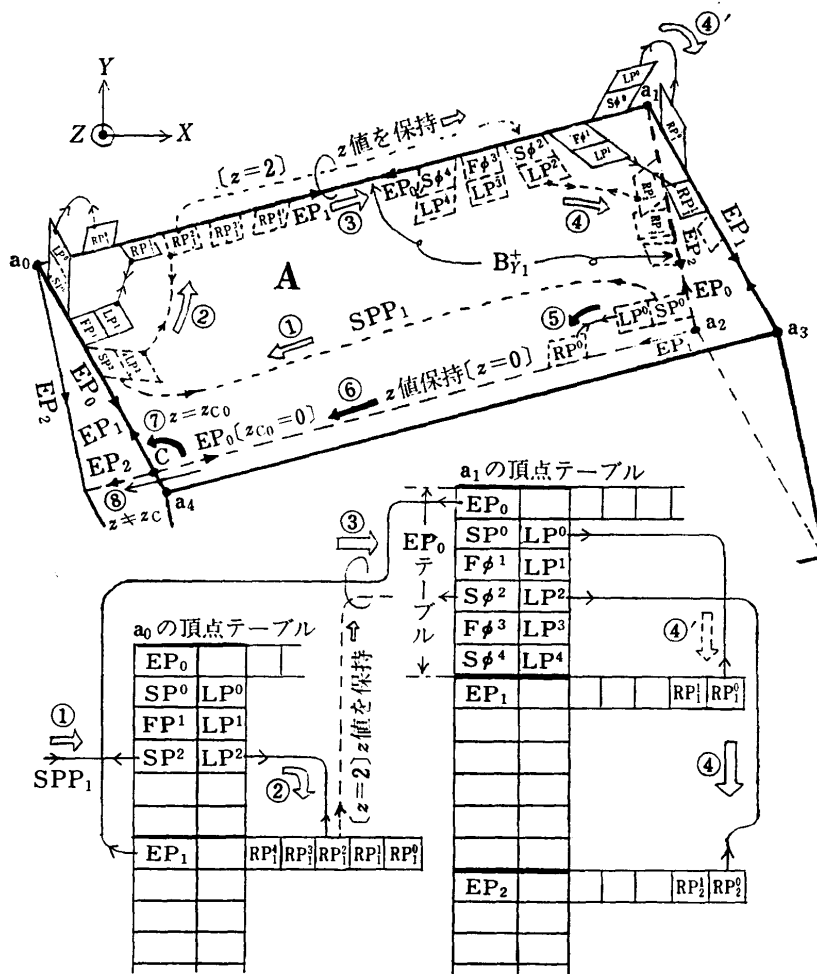


図19 回転サーチ EP たどりにおける z 値の保持

Fig. 19 z value keeping for EP tracking on rotary-search.

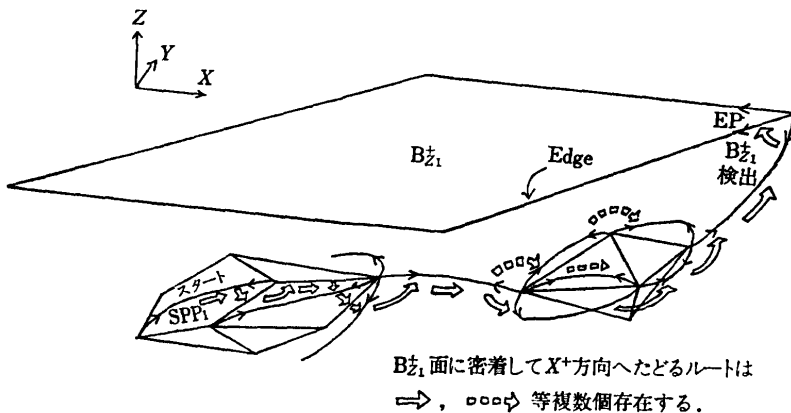


図20 Z側境界面を求めるY軸まわり回転サーチ

Fig. 20 Rotary-search around Y axis, to find out Z side border surface.

よって指されてたどり着いた SP_j^* の z 値が $z > 0$ であることで判定できる。これは SP_j^* より Z^+ 側に辺と頂点を共有する面が存在することを示し、それはすなわち $B_{z_1}^+$ にはかならないからである。また、 Z^- 側の $B_{z_1}^-$ を求めるには、 SP^0 の代りに Z 方向最も下側の SP_{max}^* をたどればよい。なお本節で述べた Z 側の検索は、別の見方から Y 軸まわり回転サーチと呼ぶこともある。

4. む す び

スペースモデリングを具体化するために必要な、頂点テーブルの構成法について提案した。検討に当り留意したことは、スペースモデルの特徴である局所的処理に必要な回転サーチをサポートすること。および、二次元・三次元・凹凸・任意回転・多重相貫等を含む汎用性を持たせることであった。また、辺・面・空間をすべてポインタで表現すると共に、できるだけ固定長テーブル化することにより部分修正の高速化を図った。これはオンライン対話処理のみならず、回転・変形を伴う運動図形のシミュレーション等、将来の高度応用への備えにもなると考えている。

今後の課題としては、曲線・曲面、階層構造の取扱い等が残されている。

謝辞 最後に本研究に当り御支援、御助言くださった関係者の方々に心から感謝の意を表します。

参 考 文 献

- 1) 大沢 晃：二次元スペースモデリングの考察，情報処理学会論文誌，Vol. 27, No. 12, pp. 1174-1185 (1986).
- 2) 大沢 晃：三次元スペースモデリングの考察，情報処理学会論文誌，Vol. 27, No. 12, pp. 1186-1196 (1986).
- 3) Baumgart, B. G.: A Polyhedron Representation for Computer Vision, *AFIPS Conf. Proc.*, Vol. 44, pp. 589-596 (May 1975).
- 4) 齊藤たつき：コンピュータ・グラフィックス工学 (4) 機械の研究，Vol. 36, No. 11, pp. 78-81 (1984).

(昭和 61 年 11 月 12 日受付)

(昭和 62 年 4 月 15 日採録)



大沢 晃 (正会員)

昭和 11 年生。昭和 34 年名古屋大学工学部電気工学科卒業。昭和 36 年同大学大学院応用物理修士課程修了。同年(株)日立製作所入社。日立大みか工場で制御用計算機ハード、主としてプロセス入出力装置、AD コンバータ、磁気ドラム、CRT、電源装置等アナログ関係設計を担当。加速迷路法によるプリント板自動配線アルゴリズムを開発。昭和 50 年より日立武蔵工場で半導体 CAD 等を担当、現在に至る。副技師長。電子情報通信学会会員。