

# 三次元 スペース・モデリングの考察

## Study on 3D Space Modeling

大沢 晃 (株)日立製作所武蔵工場  
Akira OHSAWA (Musashi Works HITACHI Ltd).

A new concept of 3D geometrical data structure is introduced. Since adjacent and/or piled subspace can be retrieved locally, very high-speed local processing capability is expected, similarly to 2D case.

By means of this method, 3D local geometric Boolean operations (AND, OR, etc.), 3D collision detection, hidden surface removal, and many other useful operations will be possible effectively and consistently.

### 1. はじめに

CAD, CAM等の分野で三次元図形情報をあいまいさなく表現する手法として、ソリッドモデルが提案され、各方面で開発、実用が進みつつあるが、処理速度が遅いことが機械、建築等大規模CAD実用化の為の大きな制限となっていた。ここで提案する三次元スペース・モデルは別途報告した二次元モデルの拡張版で、隣接、重なり図形の検索が部分処理で効率的に実行できるよう空間の構造をモデル化することにより、極端な特殊ケースが無ければ処理速度が全体頂点数 $N$ に無関係となる大巾高速化と汎用性を狙った新しい図形処理方式である。

三次元部分図形論理操作(部分AND,OR等)により、図形加工、干渉チェック等が部分処理で高速にできるほか、図形表示においては、隣接空間検索機能を活用した可視空間追跡法により、隠れ面消去処理を画面に表示される部分に限定して実行することができ、同じく $N$ に無関係な大巾高速化が可能となる。また二次元と同様に運動図形の衝突検出、回避処理が高速に出来るのでNCカッターパス、ロボット、装置組立のシミュレーション等に有用と思われる。二次元と同様に本方式は、点、折れ線、多種材質モザイク状多面体、曲線、曲面等にも拡張可能性を持つ。データ・ベース生成時間は特殊ケースが無視できれば $N$ ヶの頂点に対して $O(N)$ 、また、メンテナンスで図形1個を修正する時間も同じく $O(N^0)$ ( $N$ 無関係)と高速であり、強力な汎用図形処理システムとなることが期待される。

### 2. ポインタ方式空間表現の概念

#### 2.1 三次元空間の分割と三次元SPFの性質

三次元SPFは二次元SPFの拡張であり、三次元空間を部分空間に分割してSPPで表現する。

以下図1を参照しながら空間分割ルールを示す。

#### (1) 部分空間のX境界面 $B_x$

図形面をZ方向正負に動かして出来る角柱状空間において、頂点を中心とするZ方向軸の周りに凹角空間(Z軸に垂直な断面が $180^\circ$ 以上の空間)が存在するとき、その頂点を通りX軸に垂直な面 $B_x$ でその空間を区分する。 $B_x$ の拡がりはY、Z方向に夫々下記 $B_y, B_z$ 面と交わるまで伸びて、そこで終わる。

#### (2) Y境界面 $B_y$

各図形辺(Edge)を含み、Z軸に平行な面をY方向境界面 $B_y$ とする。 $B_y$ のZ軸方向への拡がりは、Z方向(上、下)にある隣接面すなわち下記 $B_z$ に交わるまで伸びてそこで終わる。

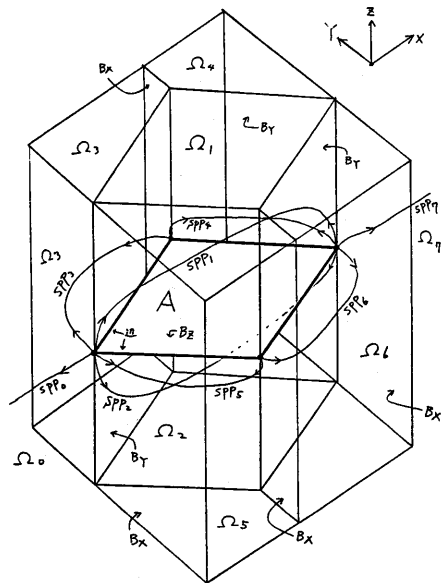


図1. 三次元空間の面Aによる分割と対応するSPP

(3) Z境界面Bz

部分空間の上底および下底Bzは、当該空間のZ方向にある図形面そのものである。

これら(1)~(3)のルールによって分割された部分空間は、一般に上底と下底を図形面とし、X-Y面に垂直な角柱の形になっている。図2は少し複雑で、四面体による空間分割の例である。

さて、三次元SPFでも二次元と同様、各部分空間 $\Omega_i$ を、 $\Omega_i$ の $X^+$ 、 $X^-$ 端の頂点間に相互に指し合う形で張られるポイントの組SPPiで表現する。三次元の場合には、図形の辺自体は直接には交叉しなくても、辺の蔭よりなる $B_y$ が他の辺と交叉する図3(i)  $c_1c_1'$ 、 $c_2c_2'$ のような蔭の交叉があり、これも部分空間のX頂点を構成しているから、各部分空間に夫々SPPが対応している。

同図(ii)は(i)の上図面であり、二次元SPFと殆ど同形となる。

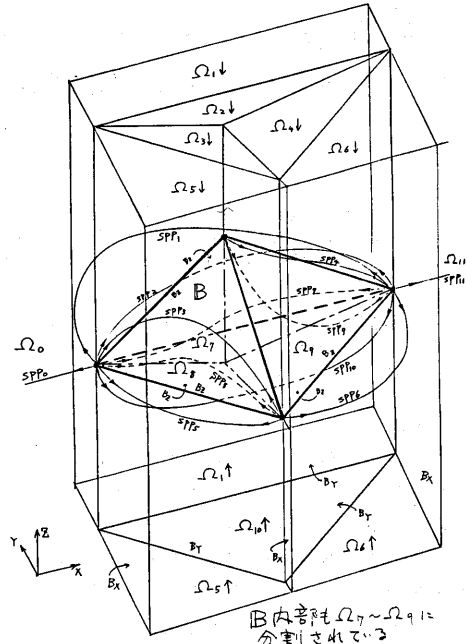


図2. 四面体Bによる空間分割と対応するSPP

2. 2 部分空間外形および隣接空間の検索

部分空間 $\Omega_i$ とポイントの組SPPiとは1対1で対応するため $\Omega_i$ の検索はSPPiの検索で代用される。以下そのための前提として、SPPiが求まったときの対応する空間 $\Omega_i$ の外形面および隣接空間の検索手法を説明する。

(1) 外形面 $Bx^+$ 、 $Bx^-$ 、およびX方向隣接空間の検索。

二次元と同じく直接検索できる。

(2) 外形面 $By^+$ 、 $By^-$ 、およびY方向隣接空間の検索。

二次元の拡張として、Z軸方向を回転軸とするZ軸廻り回転サーチにより $By^+$ 、 $By^-$ を求めることができる(図4)。回転サーチは求めるべき $By$ 、またはその延長面(辺)に密着した空間を辿る白矢印の検索であり、図形々状に拘らず必ず $By$ を見つげられる。

また $By$ の向う側隣接空間の検索は、一旦 $By$ を求めてから更に同図印の回転サーチをすればよい。

なお、三次元においても二次元の場合と類似の特殊ケースで回転サーチの手数が増大する問題があるが、対策等も二次元から容易に類推する事ができるので本報では省略する。

また、これら回転サーチを三次元図形々状に拘らず正しく且つ効率よく行うには、二次元の場合の頂点テーブルを拡張した三次元頂点テーブルで表現する等、具体データ構造に工夫を要するが、これについては第5章で述べる。

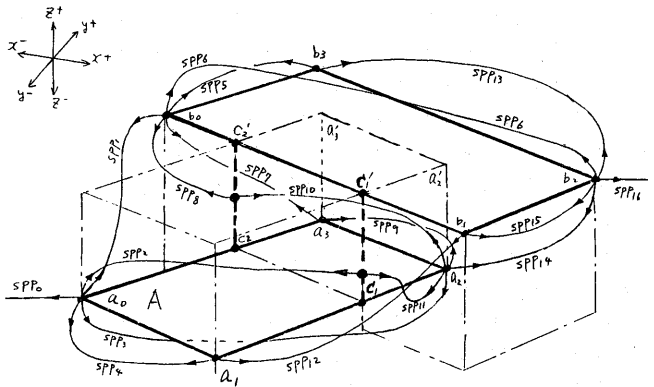


図3.(i) 蔭の交叉点 $C_1, C_2, C_1', C_2, C_2'$

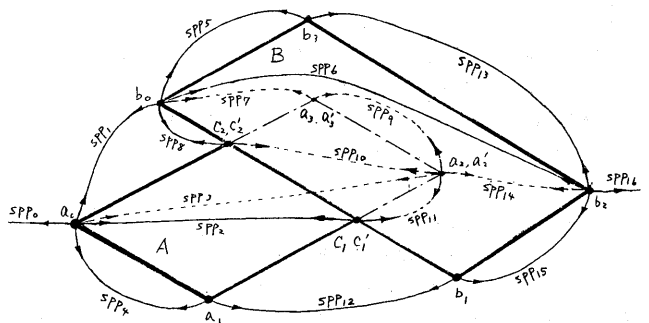


図3.(ii) 上記(i)の面図、二次元SPPと類似

(3) 外形面  $Bz^+$ ,  $Bz^-$  および Z 方向隣接空間の検索

図5で部分空間  $SPP_1$  の Z 方向境界面  $Bz^+$  の周辺エッジを求める。この検索は Y 軸廻りの回転サーチといってもよいが、むしろ求める  $Bz^+$  に下側から密着した空間の SPP を一方向(ここでは  $X^+$  方向)にどこまででも進めれば、そのうち SPP は必ず  $Bz^+$  面の外周辺上の頂点に出会うと考えたら分かり易い。このルートは  $X^+$  方向で  $Bz^+$  に密着するという条件を満たせば良いので、同図白、または破線矢印のごとく複数の可能なルートがあり、どれを選んでも良い。 $Bz^+$  への到着は辿り着いた SPP が面の下側である事で判断できる。

$Bz^+$  の Z 方向、向う側の隣接空間は、求めた  $Bz^+$  の頂点からスタートして  $Bz^+$  に上側から密着している空間の SPP を検索し、その中から必要な空間を頂点座標値によって選択すれば良い。

以上(1)~(3)の空間外形および隣接空間検索法は、いずれも基本的には遠方の図形とは無関係なローカル処理のため特殊ケースの対策をすれば  $O(N^2)$  の高速が期待される。

2.3 SPP への空間属性付加による重なり検索

図6に示すように SPF 生成時に、重なり部分を表現する SPP に複数個の空間属性を付加しておけば、検索時にはこれを調べるだけで重なるかの判定が出来ることは二次元の場合と同様で、外形々状や外側図形の数によらず超高速で可能である。SPF 生成時にこのような空間属性を付加するには、図形追加時に、常に追加前のバック・グラウンド空間の属性に、追加図形の属性を付加することで簡単に実現できる。SPF の高速生成手法については第4章で述べる。

3. 三次元 SPF の応用

3.1 応用の広がり

隣接空間や図形の重なりが直接的に検索できるから、三次元における図形論理操作(AND, OR等), 従って図形加工, 干渉

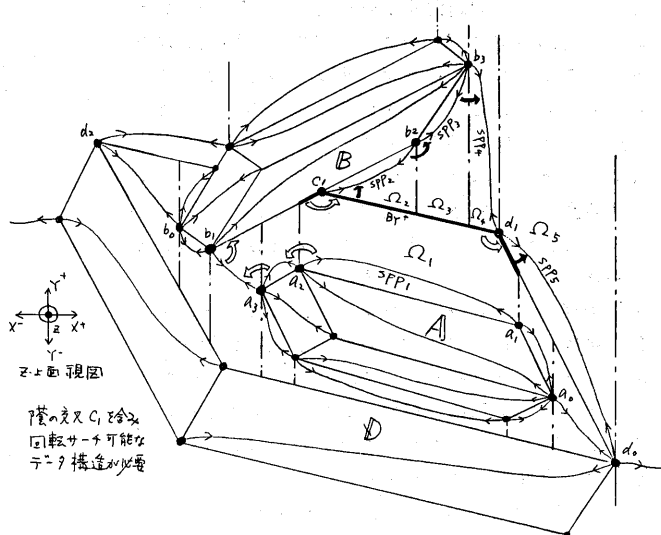


図4.  $SPP_1$  から  $Bz^+$  を求める回転サーチ(白矢印)と隣接空間  $\Omega_2 \sim \Omega_5$  を求める回転サーチ(黒矢印)

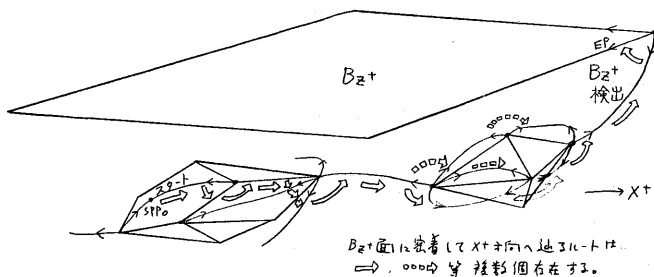


図5.  $SPP_0$  の Z 側外形面  $Bz^+$  を求める Y 軸廻り回転サーチ

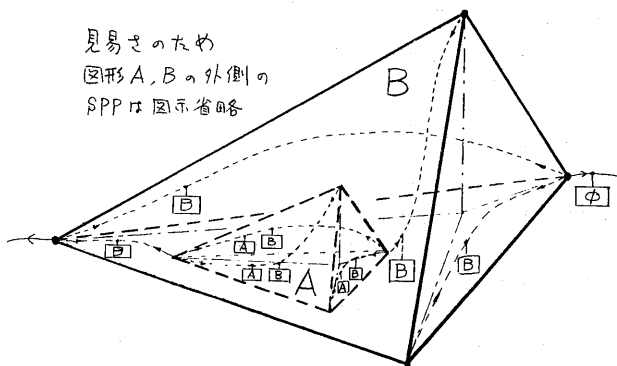


図6. 空間属性  $A, B$  による重なるの表現

チェック等が部分処理で高速に実行可能である。また、三次元空間で運動する物体をSPFで表現し、運動による空間トポロジーの変化に伴って間欠的にSPPをメンテナンスしてゆけば、衝突の検出がローカルに可能である。衝突は必ず隣接空間で発生するから、調査すべき空間を特殊ケースを除き近傍に限定できるからである。衝突検出はNC、ロボット、ゲーム等への応用に極めて重要な意味を持つが、従来、ステイックな干渉チェックよりも一段と困難とされ大きな課題になっていたものである。

更に衝突検出が部分処理で高速に出来ることから、試行錯誤による衝突フリー経路の自動探索等も可能になるだろう。将来は、SPFをロボットの視覚の内部モデルとすれば、視覚、知識ベース等と組み合わせた高度な展開もあり得よう。曲線、曲面、階層構造化等も検討を要する。

SPF中の各図形を点にまで縮小した極限では、SPFは点をX座標値でソートしSPPで接続した形になっている。その意味でSPFは従来のソーティングの拡張形と云える。各種部分処理が高速で可能なもデータがきっちりソートされているからである。

N次元ソートと考えると応用展開も拡がる。

時間を含めた四次元等も

興味深い。図形以外、OR(オペレーションズ・リサーチ)等への応用も可能性がある。

### 3.2 可視空間追跡法による高速隠れ面消去表示

図7に三次元図形表示の概念を示す。

隠れ面消去処理では、スクリーン面に投影された二次元図形の重なりチェックと表示優先度判定を、二次元スクリーンSPFの上で行うが、SPFでは処理が一図形づつ行えるので、これを三次元ワールドSPFと組み合わせると以下に述べる可視空間追跡法によって処理速度を画期的に上げることが可能になる。図8にその原理を示す。最初に三次元SPFにおいて視点を含む部分空間 $\Omega_0$ からスタートし、視点から発した視線が通過する(可視)部分空間の壁面を順次透視変換し、スクリーンSPF上で重なりチェックにより可視性を判定しながら、SPFの隣接空間検索機能を使って表示すべき空間を手前の側から逐次的に探索する手法をとる。この場合、可視空間は必ず可視透明空間の向う側に隣接し、不透明または不可視壁面

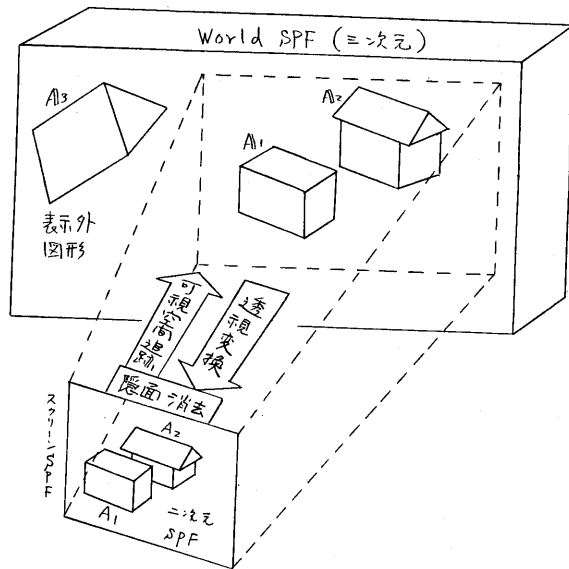


図7. 三次元図形表示システム概念図

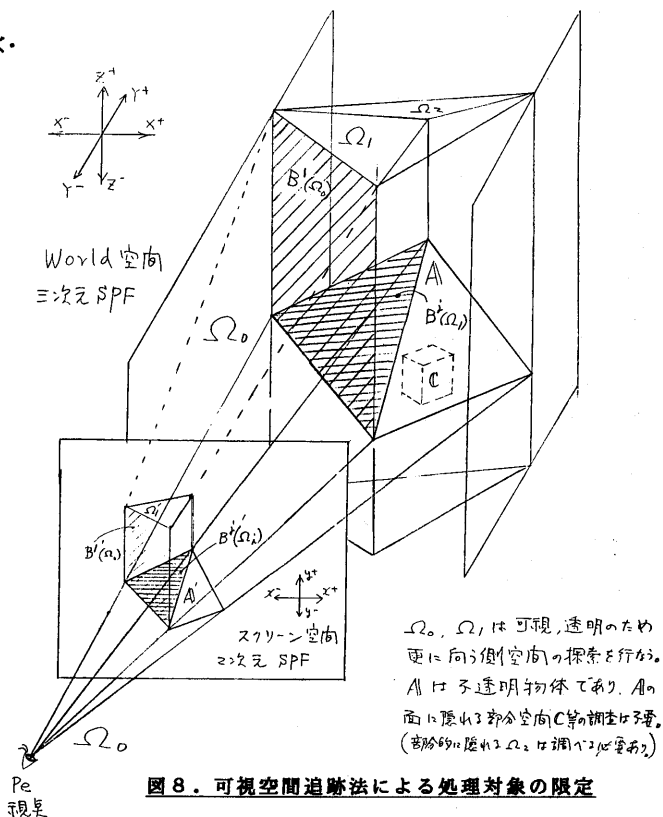


図8. 可視空間追跡法による処理対象の限定

の向う側の空間は不可視であるから、可視空間のみを検索し、不可視空間は検索しないことで、従来手法に較べオーダー違いの効率向上が可能になる。以下本方式の特徴を列挙する。

(1) 三次元SPF上で一貫処理が可能。

図形論理、衝突チェック等の結果を直接表示可能

(2) 可視空間のみを処理するので高速。

図8の家を外側から見るとき、家の中の間取り、家具等は検索しない。スクリーン外の不可視空間も調べないので、従来のクリッピングより効率が高い。処理時間は特殊ケースを除き $O(N^0)$ 、可視空間数を $n$ とすれば $O(n)$ となる。

(3) 視点変更が高速で自由。

従来の透視変換後にクリッピングを行なう方式では、視点変更時には全体図形の再処理を要したが本方式では探索した可視空間のみ処理すれば良いので高速。

(4) 三次元図形のピックアップが高速容易

スタイラス/ペン、マウス等により指示されたスクリーン図形は、二次元SPFで高速探索できる。三次元ワールドSPFの図形は、スクリーンSPFの図形とポインタで接続しておけば、すぐに判明する。

(5) 画面の部分修正ができる。

スクリーンSPFで変更部分のみの隠れ面処理ができる。図形を一部削除したり移したりしたらその後の部分だけについて可視空間追跡を行なえばよい。

(6) 表面画面のベクター/ラスタ変換が容易。

スクリーンSPFにより、図形とラスタとのANDをとればよい。静電プロッタ等へ直接出力できる。

(7) Z-バッファ等特殊ハード不要。オール・ソフト、汎用性、移植性高く、低システム・コスト、ピクセルによる精度制限がなく、ズームングでも画面が荒れない。プロッタ等大画面でも隠れ面消去出力できる。

#### 4. 三次元SPFの生成

前章まで、三次元空間・モデル原理、応用およびその数々の利点について述べたが、これらは全て三次元SPF構造による図形データベースが生成されていることを前提としたものであった。本章ではこの生成手法について述べ、その生成処理時間が特殊ケースを無視できるならば $O(N)$ のオーダーであり、一部修正等メンテナンス処理は同じく $O(N^0)$ ( $N$ と無関係)と高速であることを示す。

三次元SPFの生成も二次元の場合と同様SPFへの図形の1ヶづつの追加の繰返しで実現する。図9に示すように、図形Aの追加は最初に1つの頂点 $a_0$ を次節に示す手法で入力した後、図形の辺より成るワイ

ヤーフレームを入力し、次にこれに面を張り、最後に内部空間を空間属性で埋める(SPPに属性を付加する)段階的な方法が考え易い。以下その手順に沿って説明する。

#### 4.1 基点 $a_0$ の追加; SPOT ルーチン

SPFに点 $a_0(x_{a_0}, y_{a_0}, z_{a_0})$ を追加する手法は、基本的には二次元と同じで、まず、 $a_0$ を含む空間SPP $a_0$ をSPFの中から検索、切断し、 $a_0$ を継ぎ込めばよい。SPP $a_0$ を見付けるには最初に任意のSPP $^0$ をとり、そこからX座標の範囲が $x_{a_0}$ を含む空間に達するまでX方向に隣接空間を辿り、次は $x_{a_0}$ を含む空間ばかりをY方向に辿って、 $x_{a_0}$ と $y_{a_0}$ 両者を範囲に含む空間に到達し、最後にそこから $x_{a_0}$ 、 $y_{a_0}$ 両者を含むZ方向隣接空間ばかりを辿ってゆけば必ず $a_0$ に到達する(二次元の例参照)。この方法によれば検索ルートにループや振動はなく、検索時間はSPP $^0$ と $a_0$ の間の距離にほぼ比例するから特殊ケースを除き全体頂点数 $N$ に対し、平均 $O(N^{1/3})$ 程度の速度が得られる。

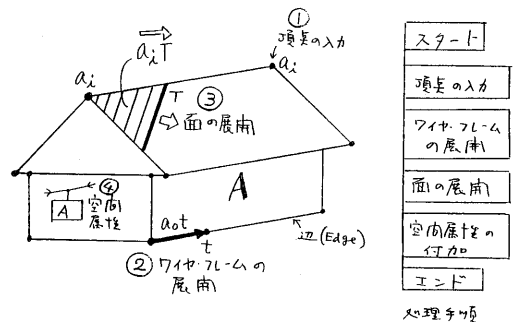


図9. 三次元SPPへの図形の追加手順

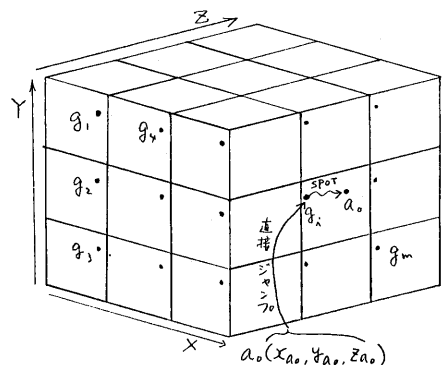


図10. 全体を $m$ ヶに区分し道しるべ $g_i$ を設置

更に高速化するには、図10の如く全体をmヶの区間に分割し、各区間に直接アクセス出来る道しるべ点 $g_i (i=1, 2, \dots, m)$ をあらかじめセットしておく、 $a_0$ の座標値が与えられたら、先ず、最寄りの $g_i$ へ直接飛んで、そこから前記方式で $a_0$ へ接近する、一種のバケット・ソート方式を採用すれば、特殊ケースを除いて $O(N^0)$ のオーダーとすることができる。

#### 4.2 ワイヤ・フレームのSPF展開

前節で点 $a_0$ を入力したから、次は図11の如く $a_i$ から発する線分 $a_i t$ を入力Aの辺に沿って順に伸ばすことにより、Aの辺より成るワイヤ・フレームをSPF上に展開する。 $a_i t$ が伸びて、先端 $t$ が周囲空間上下、左右何処かの壁を突き抜ける時、空間分割のトポロジーが突然変化し、SPFの接続を変える必要が起る。これをイベント $E_j (j=1, 2, \dots)$ と呼ぶ。 $a_i t$ をAの辺に沿って伸長させながら、イベント毎にSPPのメンテナンスを行ない、Aの辺全体をカバーすればワイヤ・フレーム入力完成する。 $E_j$ の種類は三次元では $a_i t$ が $B_z$ (図形面)を貫通する場合(図12)がある他は、二次元(文献1, 表1)と同じになる。

これは図3(ii)で空間分割をZ方向から見ると二次元と同様に見えることから理解出来るよう、ワイヤ・フレーム入力により、Aの周囲の空間は、Aの頂点を通りX軸に垂直な面 $B_x$ とAの辺を含みZ軸に平行な面 $B_y$ によって分割され、夫々の空間にSPPが設置される。

以上の処理は入力図形の近傍空間のみで行なわれるから、特殊ケースの対策をすれば $O(N^0)$ のオーダーになる。

#### 4.3 面の展開

ワイヤ・フレーム入力が終わったので、次は面を張る。面Aの展開処理は図13の如く始点を面Aの $X^-$ 端に取り、A面上にあってX軸に垂直な先端線 $T$ を持つ部分 $a_i t$ を、 $T$ を $X^-$ 端から $X^+$ 方向へ動かしながら、拡げてゆき、これに伴って $a_i t$ の周囲空間のSPPをメンテナンスしてゆく処理である。

$T$ が $X^+$ 方向へ動いて、既設図形によって出来ている空間の外壁 $B_y, B_z$ の折れ線を通過するとき、空間分割のトポロジーが急変する。これをイベント $E_j (j=1, 2, \dots)$ という。 $T_j$ と $B_y, B_z$ との交点 $t_{jk}$ には頂点テーブルが設置されて周囲空間のSPPが接続しており、 $E_j$ 発生の際にメンテナンスする。

イベント $E_j$ における空間トポロジーの変化は、 $E_j$ が発生した $t_{jk}$ のまわりだけだから、 $T$ のメンテナンスは、図14の如く、 $T$ に $E_j$ を通過する部分 $T_j$ を追加して、 $T_{j-1}$ を削除するプログラムを作ればよい。このとき、同図(a), (b)部は $E_j$ 前後で同形だから、そのまま流用できる。 $T_j$ の追加は同図にしめすように、A面上の線分 $t_{j0}$ を $T_j$ にそって伸ばしながら既設空間境界 $b_a, b_b, b_c$ 等との交点に頂点テーブルを設置してゆき、続いてその

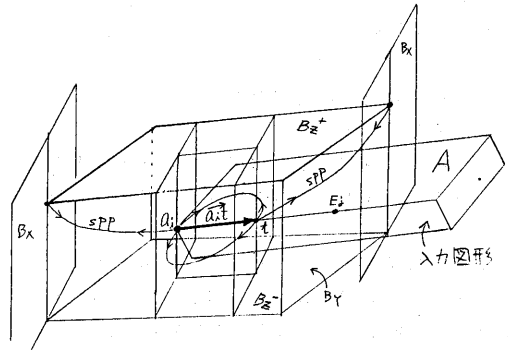


図11. ワイヤフレーム展開  
Aの辺に沿い $a_i t$ を伸ばす

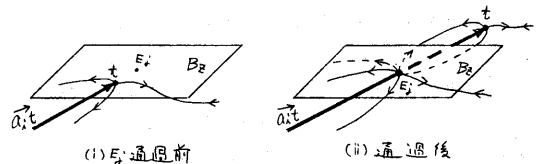


図12. ワイヤフレーム展開における  
 $B_z$ 面貫通のイベント $E_j$

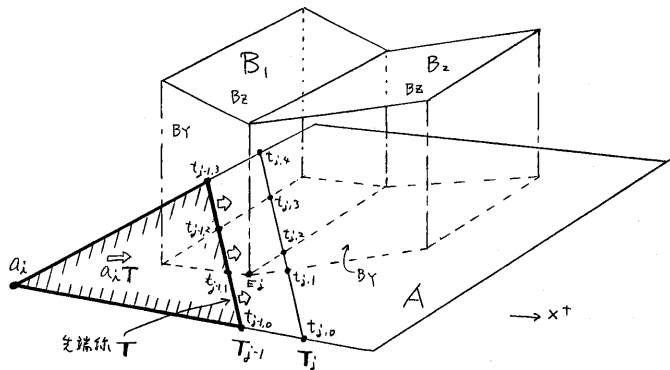


図13. 面Aの展開、 $T$ が $E_j$ に来た時 $t_{jk}$ をメンテする

$X^-$ 側に面を張る(A面 $Z^+$ ,  $Z^-$ 側各空間に別々のSPPを設置することで表現)方式により各種形状の $E_j$ に対処できる。図15は $E_j$ 前後のSPPの接続状況の例である。

$E_j$ 通過の次の $E_{j+1}$ を見つけるには、 $t_{jk}$ をつくっている辺または、 $t_{jk}$ から発しているSPPの $X^+$ 方向にある頂点のうち、最も手前( $X$ 座標の小さい)のものをとればよい。 $T$ をすすめていって $E_{j+1}$ が無くなれば展開は完了する。

以上の面の展開は特殊ケースを除けば全ての処理が入力面Aの周囲近傍空間で閉じ、遠方の図形とは無関係なので $O(N^0)$ オーダーの処理である。

以上頂点入力、ワイヤ・フレーム、面の展開、夫々が $O(N^0)$ の処理なので、全体SPPの初期生成も $O(N)$ の高速性が期待できる。

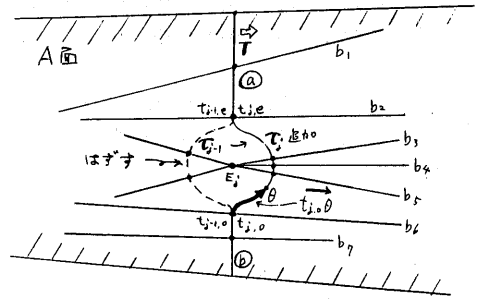
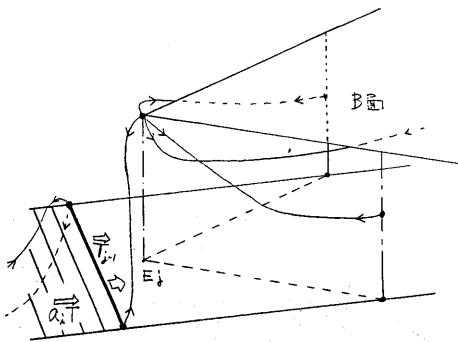
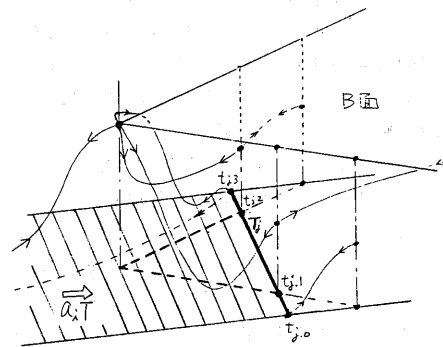


図14.  $E_j$ における $T$ のメンテナンス  
 $T_{j-1}$ を外して $T_j$ を追加



(i)  $E_j$ 通過前



(ii)  $E_j$ 通過後

図15. イベント $E_j$ 前後のSPPの状況

## 5. 三次元空間・モデル用頂点テーブル

### 5.1 三次元図形データ構造への要求事項

(1) 任意角度, 多重相貫, 空洞を含む多種材質モザイク状多面体, 厚さのない面, 太さのない折れ線, 大きさのない点, 等の表現が可能なこと。

図16は多重相貫図形の例, 図17は $\square$ 空間 $\square$ 面を含む多種材質モザイク図形の例である。

(2) 頂点座標値のほか辺FP, 空間SPの各ポイントが記述できること。ここで初めて出てきたFP(sur-face Pointer)は図18に示すように模様や色等を含む物体表面を内部の空間とは別の物として表現するためのポイントである。

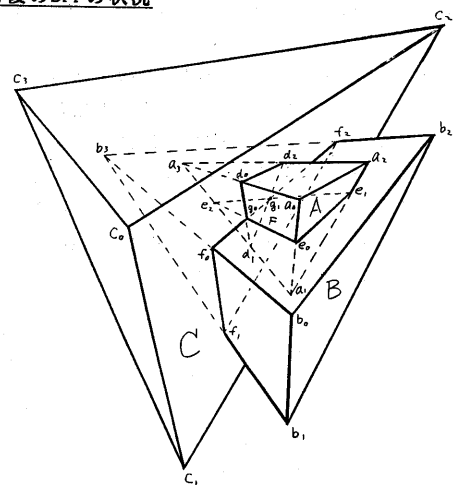


図16. 多重(三重)相貫図形の例

(3) 面や空間には、夫々面属性FA(surFace Attribute),空間属性SA(Space Attribute)が付加できること。図6の重なり表現を、図16の相貫点表現と組み合わせれば、任意多重相貫図形が表現できる。

(4) 三次元SPF空間分割に特有の蔭の交叉点の表現が可能なこと。

(5) 三次元回転サーチが高速で可能のこと。

(6) 図形追加,加工,移動,回転等に対するメンテナンスが容易。

(7) コンパクト、その他。

## 5.2 頂点テーブルによる三次元図形表現。

### 5.2.1 三次元頂点テーブルの原理

二次元SPFでは図19で示した如く、反時計廻りループ状の頂点テーブルによって頂点の周りのEPとSPの隣接性を完全に表現でき、これによって回転サーチの高速化が達成された。しかし、三次元では図20(i)の如く、辺 $EP_i$ に隣接する空間が左右夫々複数個存在するため、上記のような一次元ループ状のテーブルでは隣接性の表現は出来ない。

この問題を解決し、頂点を発する辺 $EP_i$ の数によらず、且つ $EP_i$ に隣接する空間SPや面FPの数によらず隣接性を表現できる汎用的な三次元頂点テーブルの方式を同図(ii)に示す。ここでは、頂点テーブルは辺EPの数だけEPブロックを並べたものであり、 $EP_i$ ブロックには頂点VからZ軸+側を上にして $EP_i$ の辺を眺めた時、 $EP_i$ の左側に隣接する $SP^z$ と $FP^z$ (Zはこれから $SP^z$ と $FP^z$ をZ軸上側から数えた順序、 $Z=1,2,\dots$ )を順に設置する。 $EP_i$ ブロックに入れることによって $EP_i$ とその左側の $SP^z, FP^z$ の隣接性が表現される。これら $SP^z, FP^z$ と更にその左側の辺 $EP_j, EP_k$ との隣接関係は図指の如く各 $SP^z, FP^z$ に付属して、その左側を指すテーブル内ポインタLP(Left Edge Pointer)と逆に各 $EP_j, EP_k$ に付属してその右側を指す同じくテーブル内ポインタRP(Right SP or FP)の相互の指し合いで表現する。一個のEPの右側にも一般にはSP,FPがZ方向に複数重なって隣接しているので、これに対応して $RP^z$ についてもEPブロックの中に図のようにZ軸+側から並べれば完全な表現が出来る。この方式により三次元回転サーチがサポートできる。(空間属性は各ポイントに付加する(図示略))

### 5.2.2 各種頂点テーブルの例

図21~図25に示す基本的な頂点テーブルを組み合わせれば、 $\square, \square, \square$ 多重相貫を含む任意多面体を表現できる。曲面の四辺形パッチや、モザイク状多材

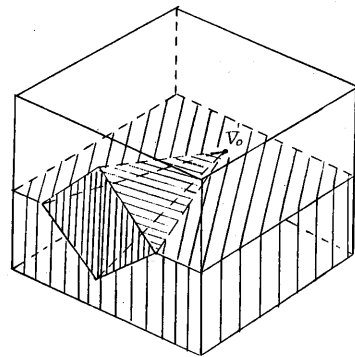


図17.  $\square$ 空間を含む多種材質頂点 $V_0$

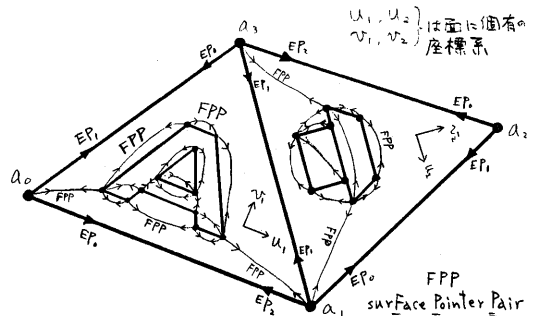


図18. 物体表面の図柄のFPPによる表現

質多面体等で現れる複雑頂点も、これらを接近して組み合わせれば表現できるが、出現頻度が多い場合には、専用テーブルを設定するのがよい。メモリ管理の便のためには頂点テーブル長は全て統一が望ましいが、それも、困難なので整数倍化する等配慮をする。また図23では、空間のポイント $SP^c, SP^i$ 等をテーブル内ポインタ $rp, cp$ で他ブロックの空間へ飛ばして設置する事により、図形の三次元回転があってもEPブロック長を一定に保ち、メンテナンスを容易化している。(空間属性は図示を省略した)



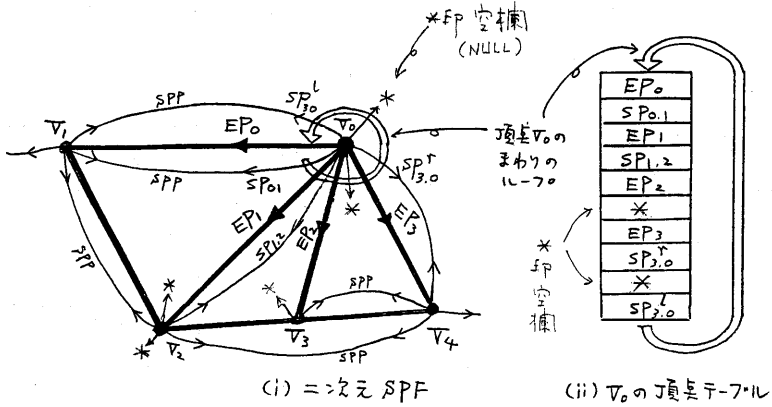
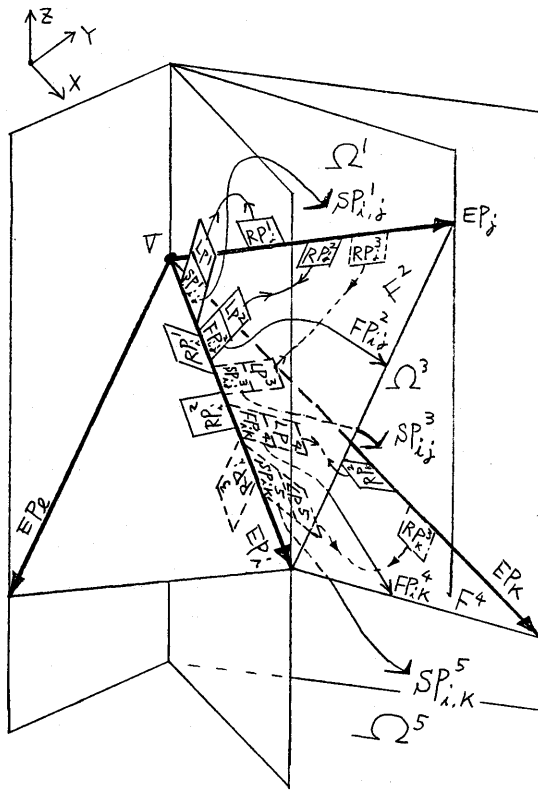


図 19. 二次元SPPにおける頂点テーブル

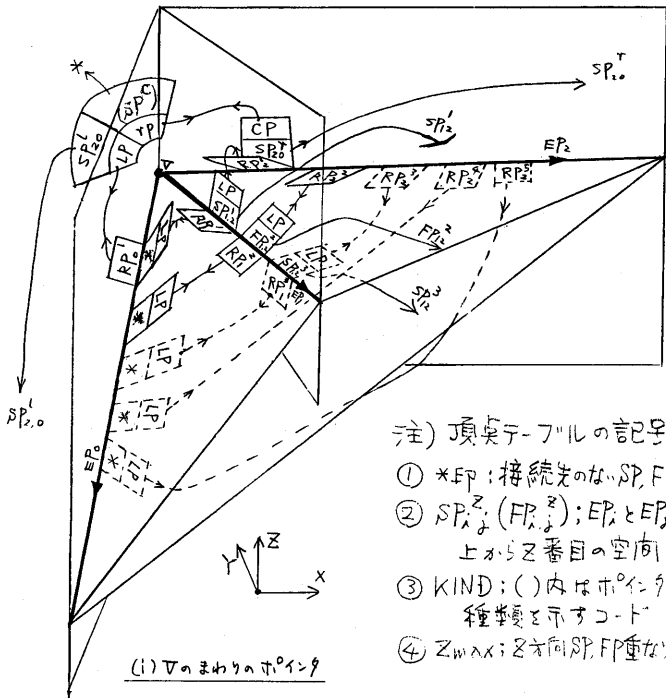


	POINTER	KIND	LP	RP	ADDR
EP <sub>j</sub> ブロック	EP <sub>j</sub>	(+EP)			j0
				RP <sub>j</sub> <sup>1</sup>	j1
				RP <sub>j</sub> <sup>2</sup>	j2
				RP <sub>j</sub> <sup>3</sup>	j3
EP <sub>i</sub> ブロック	EP <sub>i</sub>	(+EP)			i0
	SP <sub>i,j</sub>	(SP)	LP <sub>i</sub> <sup>1</sup>	RP <sub>i</sub> <sup>1</sup>	i1
	FP <sub>i,j</sub>		LP <sub>i</sub> <sup>2</sup>	RP <sub>i</sub> <sup>2</sup>	i2
	SP <sub>i,k</sub>		LP <sub>i</sub> <sup>3</sup>	RP <sub>i</sub> <sup>3</sup>	i3
	FP <sub>i,k</sub>		LP <sub>i</sub> <sup>4</sup>	RP <sub>i</sub> <sup>4</sup>	i4
EP <sub>k</sub> ブロック	EP <sub>k</sub>	(+EP)			k0
				RP <sub>k</sub> <sup>1</sup>	k1
				RP <sub>k</sub> <sup>2</sup>	k2
				RP <sub>k</sub> <sup>3</sup>	k3

辺の左側 | 右側  
(理実不一致)

(ii)  $V$  の頂点テーブル

図 20. 三次元SPPにおける頂点テーブルの原理



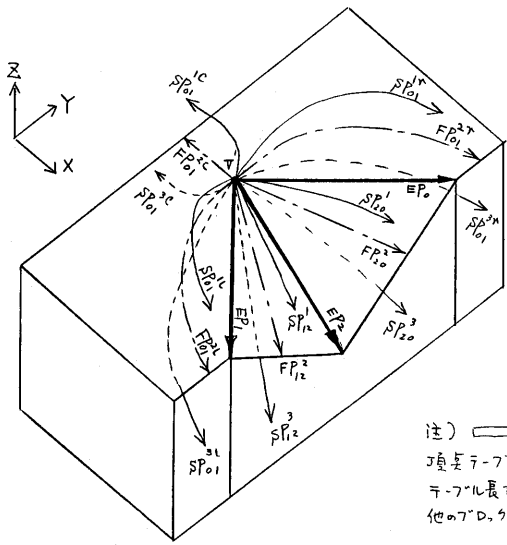
- 注) 頂点テーブルの記号
- ① \*EP: 接続先のないSP, FP(空欄)
  - ②  $SP_{i,j}^z$  ( $FP_{i,j}^z$ ):  $EP_i$ と $EP_j$ の向  
上からz番目の空間(面)
  - ③ KIND: ( )内はポイントの  
種類を示すコード
  - ④  $Z_{max}$ : z方向SP, FP重なり最大値

(i) Vのまわりのポイント

図21. 多面体凹頂点Vの廻りのポイントと頂点テーブル

	x	z		
	$\bar{x}$	$\bar{z}$		
	y	$\bar{y}$		
	z	$\bar{z}$		
	t	$\Delta t$		
EP <sub>0</sub> ブ ロ ック	EP <sub>0</sub>	(-EP)	$Z_{max}=5$	
	* (SP)	LP <sub>1</sub> <sup>1</sup>	RP <sub>1</sub> <sup>1</sup>	
	* (FP)	LP <sub>2</sub> <sup>2</sup>		
	* (SP)	LP <sub>3</sub> <sup>3</sup>		
	* (FP)	LP <sub>4</sub> <sup>4</sup>		
EP <sub>1</sub> ブ ロ ック	EP <sub>1</sub>	(+EP)	$Z_{max}=3$	
	SP <sub>12</sub> <sup>1</sup>	(SP)	LP <sub>1</sub> <sup>1</sup>	RP <sub>1</sub> <sup>1</sup>
	FP <sub>12</sub> <sup>2</sup>	(FP)	LP <sub>2</sub> <sup>2</sup>	RP <sub>2</sub> <sup>2</sup>
	SP <sub>12</sub> <sup>3</sup>	(SP)	LP <sub>3</sub> <sup>3</sup>	RP <sub>3</sub> <sup>3</sup>
EP <sub>2</sub> ブ ロ ック		(NC)		
	EP <sub>2</sub>	(+EP)	$Z_{max}=1$	
	SP <sub>20</sub> <sup>1</sup>	(SP <sup>1</sup> )	CP <sub>1</sub> <sup>1</sup>	RP <sub>1</sub> <sup>1</sup>
	* (SP <sup>0</sup> )	LP <sub>1</sub> <sup>1</sup>	RP <sub>2</sub> <sup>2</sup>	
	SP <sub>20</sub> <sup>L</sup>	(SP <sup>L</sup> )	LP <sub>1</sub> <sup>L</sup>	RP <sub>2</sub> <sup>L</sup>
	(NC)		RP <sub>2</sub> <sup>2</sup>	
	(NC)		RP <sub>2</sub> <sup>L</sup>	

(ii) Vの頂点テーブル



- 注) 頂点テーブル\*EPは  
テーブル長調整のため  
他のブロックスに設置

(i) V'のまわりのポイント

図22. 多面体頂点V'の廻りのポイントと頂点テーブル

	x	z		
	$\bar{x}$	$\bar{z}$		
	y	$\bar{y}$		
	z	$\bar{z}$		
	t	$\Delta t$		
EP <sub>0</sub> ブ ロ ック	EP <sub>0</sub>	(+EP)	$Z_{max}=3$	
	SP <sub>01</sub> <sup>1</sup>	(SP <sup>1</sup> )	SP	RP <sub>1</sub> <sup>1</sup>
	FP <sub>01</sub> <sup>2</sup>	(FP <sup>2</sup> )	CP	RP <sub>2</sub> <sup>2</sup>
	SP <sub>01</sub> <sup>3</sup>	(SP <sup>3</sup> )	CP	RP <sub>3</sub> <sup>3</sup>
	SP <sub>01</sub> <sup>L</sup>	(SP <sup>L</sup> )	YP	
EP <sub>1</sub> ブ ロ ック	EP <sub>1</sub>	(+EP)	$Z_{max}=3$	
	SP <sub>12</sub> <sup>1</sup>	(SP)	LP	RP <sub>1</sub> <sup>1</sup>
	FP <sub>12</sub> <sup>2</sup>	(FP)	LP	RP <sub>2</sub> <sup>2</sup>
	SP <sub>12</sub> <sup>3</sup>	(SP)	LP	RP <sub>3</sub> <sup>3</sup>
EP <sub>2</sub> ブ ロ ック	* FP <sub>01</sub> <sup>C</sup>	(FP <sup>C</sup> )	YP	
	* FP <sub>01</sub> <sup>L</sup>	(FP <sup>L</sup> )	LP	
	EP <sub>2</sub>	(+EP)	$Z_{max}=3$	
	SP <sub>20</sub> <sup>1</sup>	(SP)	LP	RP <sub>2</sub> <sup>1</sup>
	FP <sub>20</sub> <sup>2</sup>	(FP)	LP	RP <sub>2</sub> <sup>2</sup>
			RP <sub>2</sub> <sup>3</sup>	
			RP <sub>2</sub> <sup>3</sup>	

(ii) V'の頂点テーブル

6. まとめ

三次元スペース・モデルとその応用について概略を述べた。ポイントのペアで部分空間を表現することで隣接、重なり空間の検索が部分処理で可能になり、この応用として従来困難だった運動図形の衝突検出が実現できるほか、極端な特殊ケースを除き、表2に示す性能が、1つのデータ・ベースの上で一貫して実現できる見通しが得られたと考えている。

本方式はまだ原理考察の段階であり、実用的システム実現までにはまだ相当の距離がある。

しかし、ポテンシャルも大きいと思われるので大方の御批判を仰ぎながら検討を進めたいと考えている。

終りに本研究に当たり御支援、御検討くださいました名古屋大学工学部杉原助教、日立製作所金原半導体事業部長、武蔵工場牧本副工場長、日立研究所谷中部長、徳増主管研究員、ソフト工場越智部長他関係者の方々に感謝の意を表します。

表2. スペース・モデル性能 (\*特殊ケース除き)

NO	項目	スペースモデル	ソリッドモデル
1	隣接・重なり検索	$O(N^2)$ *	$O(N)$
2	図形AND(1対N)	$O(N^2)$ *	$O(N)$
3	隠れ面消去表示	$O(N^2)$ *	$O(N)$
4	運動図形衝突検出	$O(N^2)$ *	? (困難)
5	データベース生成	$O(N)$ *	$O(N)$
6	同上 メンテ	$O(N^2)$ *	$O(N)$
7	所要メモリ	$O(N)$	$O(N)$
8	Zバッファ等	不要(全ソフト)	必要性大

参考文献

- 1) 大沢 晃; 二次元スペース・モデリングの考察  
本情報処理学会グラフィクスとCAD研究会誌
- 2) A.A.G.Requicha, H.B.Voelcker; Solid Modeling  
; Current Status and Reserch Directios.  
IEEE CG & A Oct.1983 PP25-37
- 3) 今宮淳美; 隠面除去アルゴリズム; 情報処理,  
Vol.24, No.4, PP539-546 (Apr.1983)
- 4) D.T.Lee, F.P.Preparata; Computational Geom  
etry-A Survey  
IEEE TRANS. ON COMPUTERS, Vol.C-33, No.12,  
PP1072-1101, Dec.1984
- 5) 重松、嘉敷、沖野 ; 3D形状モデル干渉問題の  
一解法  
精密機械 Vol149, No.11 PP89-94(1983年11月)
- 6) Markku Tamminen, Reijo Sulonen; The Excell  
Method for Efficient Geometric Access to  
Data. ACM IEEE 19th Design Automation Conf.  
Proc. PP 345-351 June 1982

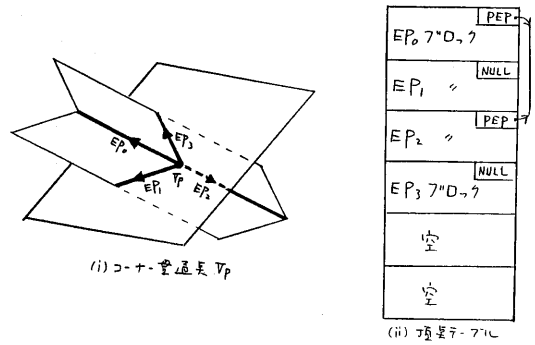


図23. コーナー貫通点Vpと頂点テーブル

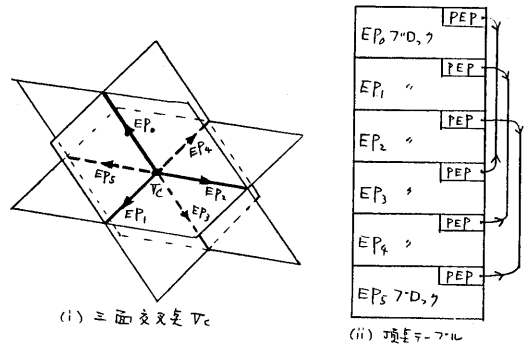


図24. 三面交差点Vcと頂点テーブル

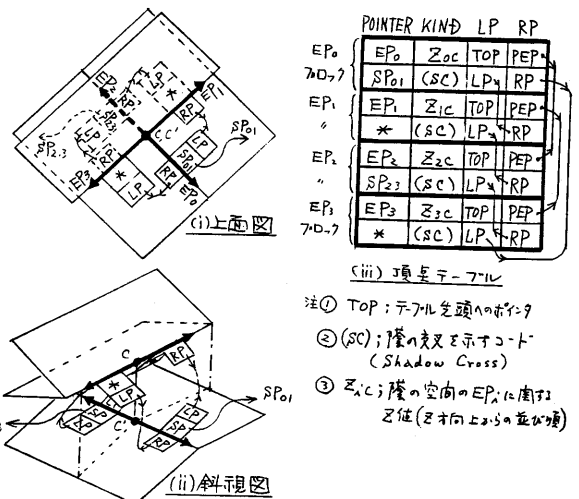


図25. 蔭の交点CC'と頂点テーブル

- 注① Top: テーブル先頭の行  
② (SC): 蔭の交点を示すコード (Shadow Cross)  
③ Zc: 蔭の空間のEPiに属するZ値(Z方向上への並み順)