スペースZ-バッファ：安定で高速な隠線・隠面消去二次元スペースモデルの応用

大沢 晃†

従来隠面消去方式としてはピクセル対応のZ-バッファ法が定着している．しかし，ピクセル単位のため面や辺の取扱いには適せず，ペンプロッタで線画が描けない，マウスによる図形ピックが困難等の問題があった．一方，ペンプロッタ用には，別プログラムで稜線と全物体面の組合せについて不可視数等の隠蔽関係を調べる方法があったが，図形数 N に対して処理時間が $O(N^2)$ となり，大規模データではバケット法など画面分割が必要であった．また計算誤差で可視判定が混乱して不正な表示をしたり，図形のトポロジーが矛盾して不安定化する問題もあった．今回スクリーン図形の内部表現として，スクリーン表示形式とは独立のスクリーン・バッファをスペースモデルのデータ構造で構成し，可視データだけを面と辺の形で保持する隠面・隠線消去両用方式を考案し，問題点を解決した．試作結果，従来の専用ハード化したピクセル対応Z-バッファよりは遅いが，道しるべ法により画面分割なしで $O(N)$ の速度が確認できた．不正表示については，本方式では入力面が重なり部分ごとの小区画に分割されることを利用して，誤りを実用範囲まで削減できた．またトポロジーの矛盾については，ポインターでトポロジーを表現し管理する方式で解決した．その他，本方式はピクセルやスキャンラインと独立のため解像度の問題がない，データ構造が図形集合演算等，各種の図形処理に適した汎用性を持つ等の特徴も備えている．

Space-Z-Buffer: A Fast and Robust Hidden Line/surface Removal by Using Two Dimensional "Space Model"

AKIRA OHSAWA†

A new method for both hidden line and surface removal is developed. Because conventional Z-buffer holds only pixel data, it is inapplicable for a pen-plotter, picking a figure up by mouse, and such. So far, for pen-plotter, the other method examining quantitative invisibility is used. It's deficiency, however, is too large processing time as $O(N^2)$ unless dividing the screen. In the new method, where only visible surfaces and edges are held in the "Space Z-Buffer," $O(N)$ processing time is realized without dividing the screen. Conventional hidden line removal systems are unstable if the topology of the figures is inconsistent caused by calculation errors. In this method, the system is stabilized by controlling the consistency using pointer representation of the topology. Another problem brought by calculation errors in depth comparison is erroneous display of invisible parts. This can be eased by dividing the figures into regions each of them overlaps with a different surface and by comparing the depth at the central portion of the each region.

1. はじめに

立体図形の隠面消去表示は，ピクセル対応のメモリに視点から最も近い面の情報を保持し，これを1面入力ごとに更新するZ-バッファ法が主流となりつつある．この方法はアルゴリズムが簡単でハードウェア化による高速化にも適するが，ピクセル単位の処理のため，面や辺を扱う処理には適しない．たとえばペンプロッタの線画出力，マウスによる面のピック，表示図形の集合演算等には別プログラムが必要であった．一方，隠線消去には，古くAppel[8]以来，すべての稜線と物体面の組合せにつき隠蔽関係を調べて不可視部を消去する方法[4]が多く用いられてきた．しかし面数 N に対し処理時間が $O(N^2)$ と遅く，大規模データではバケット法など画面分割による高速化が必要であった[7]．

本スペースZ-バッファ（以下，SPZB）法では，スクリーン図形の内部表現として，スクリーン表示形式とは独立のスクリーン・バッファを二次元スペースモ

† 中部大学工学部工業物理学科
Department of Engineering Physics, Chubu University

デル（以下，SPM）のデータ構造（文献1)の改良）で構成し，可視データを面と辺の形で保持して1面入力ごとに逐次更新する方式で上記問題点を解決した．処理時間はハードウェア化した従来のZ-バッファには及ばないが，画面分割なしで$O(N)$を実現できた．

また図形処理アルゴリズムは，杉原ら[5),6)]が図形集合演算の例で指摘しているように，計算誤差で図形のトポロジーが矛盾すると暴走する．実際，従来の隠線消去法は，計算誤差で不可視数の算定を誤ると暴走することがある．この問題は，本研究ではSPMのデータ構造がトポロジーを表現することを利用し，どんな場合にも矛盾したトポロジーを作らない方式[3)]で解決した．また奥行き判定の計算誤差による不正表示も実用範囲で回避できた．

なお，本論文の内容は紙面の都合で隠面・隠線消去に限定した．このデータ構造による集合演算その他の図形処理機能については文献1)，2)を参照されたい．

以下，2章でSPMの原理を，3章でSPZBによる隠面・隠線消去手法の説明と性能を，4章で計算誤差に起因する問題点，対策および実験結果を示す．

2. SPMの原理

2.1 SPM（SPZB）のデータ構造

隠面・隠線消去用スクリーン・バッファとしてSPMデータ構造を持つSPZBを採用した．SPMの原理は文献1)と同じであるが，今回は空間分割の方法を一部改良してプログラムを簡単化した（付録参照）．

SPMでは頂点間（以下，交点も頂点の一種とする）で相互に指しあうポインタEP（Edge Pointer）の組EPP（EP Pair）で辺を表現する（図**1**）．次に各頂点を通り上下隣の辺まで伸びる鉛直仮想境界線B_xを想定し，図形内外の全空間をB_xと辺とで小空間に分割する．各小空間には左右端の頂点間（X軸＋－無限遠にも頂点あり）で相互に指しあうポインタSP（Space Pointer）の組SPP（SP Pair：図1細矢印）を設置する．具体的にはSPMは各頂点に設置した**図2**の頂点テーブルの集合である．頂点テーブルには頂点周りの辺と空間の反時計回り順にEPとSPを並べる．EPPとSPPには両端頂点の座標値の辞書順を表すX方向フラグ（同図＋印）を付加する．辞書順ではYにかかわらずXの大きい方を＋（右）側，Xが等しいときはYが大きい方を＋側とする．X，Yともに等しい頂点や，他の辺に重なる頂点，3辺以上の同一点での交差は，座標値は同じでもトポロジカルには重ならず，相互に（任意の固定方向に）ずれているとする．これで鉛直辺はわずかに右傾した斜辺とされ，X値の等しい

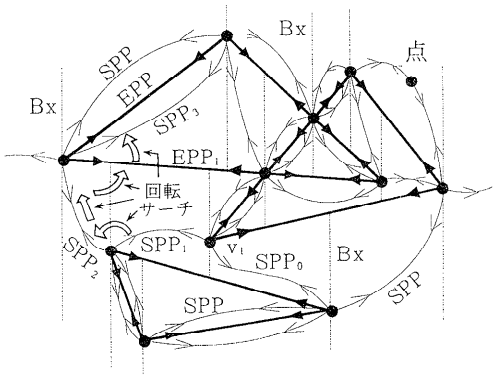
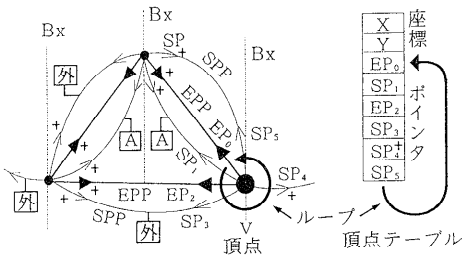


図1 SPMのデータ構造と回転サーチ
Fig. 1 The data structure of SPM, and a rotary search on it.



SPPの属性でAの面番号[A]や外部[外]を表現

図2 頂点テーブルによるSPMデータ構造の実現
Fig. 2 Realization of SPM structure by vertex tables.

2頂点のそれぞれを通るB_xは重ならず，間に無限小幅の空間ができる．以上ですべての小空間の左右端にはSPPの接続する頂点が必ず1個だけ存在し，SPPは小空間を1対1で表現するといえる．物体面はSPPに属性として付加した面番号で識別できる．面の外部では，後ろに重なる他の面がなければ，バックグランドとして面番号をゼロとする．穴は面の外部である．SPMは折れ線や点も表現できる．図1の右上には頂点の一種としての点を示す．

SPMはEPPとSPPをブランチ，頂点をノードとする有向グラフで，図形要素（頂点，辺，小空間）の並び順（トポロジー）を表現しているともいえる．

2.2 図形要素の高速検索と道しるべの利用

SPMでは隣接図形要素の高速検索が可能になる．たとえば図1でSPP_1を与えれば，右隣のSPP_0は頂点V_1から発するプラスX方向のSPPとして検索できる．また＋Y側のSPP_3は，白抜き矢印の順にSPPをたどる回転サーチと呼ぶ手法で辺EPP_1を求め，さらに上に黒矢印の回転サーチで求められる．

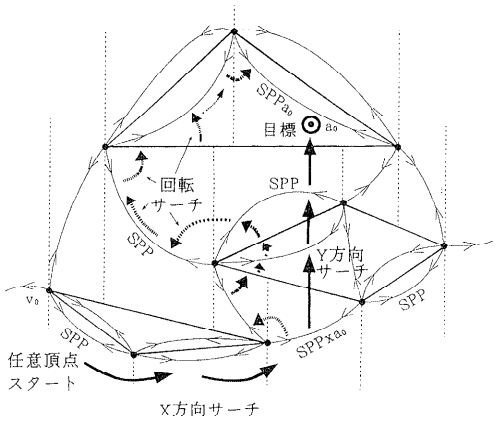


図3 目標点 a_0 を含む空間 SPP_{a_0} の探索
Fig. 3 Retrieval of the space SPP_{a_0} that include target a_0.

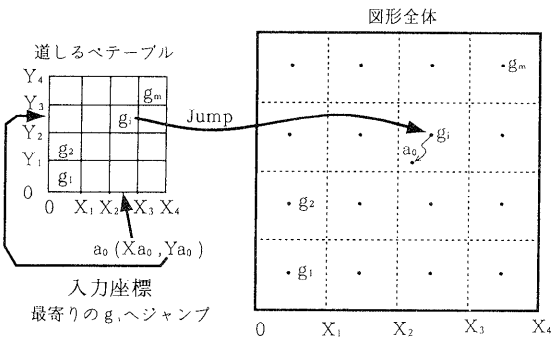


図4 全体を $m = N/k$ 個に区分，最寄りの道しるべ g_i へジャンプ
Fig. 4 Jump to the nearest guide post g_i dividing the entire space into $m = N/k$.

この応用で，マウスによる面のピックでは，座標値で示された点 $a_0(x_{a0}, y_{a0})$ を含む空間の SPP_{a_0} を検索し，属性の図形番号から面を特定する．検索は**図3**の任意頂点 V_0 から出発し，まずX方向へ，両端のX座標が x_{a0} を挟む $SPP_{x_{a0}}$ までSPPをたどって進み（太矢印：最適道筋でなくともやむをえない），次にY側で x_{a0} を含む小空間を回転サーチでたどりY方向に進む．SPP_{a_0} への到着は，a_0 の座標値が到着小空間の上下の辺の間にあることで判定する．検索の道筋はX，Yともに一方向的で戻りがないため効率が良い．

次章の多角形 N 個の入力では，各最初の頂点 a_0 の位置する空間 SPP_{a_0} の検索をさらに高速化するため，事前に N に比例する $m = N/k$ 個（k は定数）の道しるべ g_i $(i = 1, 2, \cdots, m)$ を設置する（**図4**：波線は概念的区分を示すだけで実在しない）．各入力多角形ごとに，その最初の頂点 a_0 の座標値から同図の道しるべテーブルで最寄りの g_i へジャンプして，これを上記 SPP_{a_0} 検索の出発点 V_0 とすれば，図示のように V_0 から a_0 までの検索距離をつねに一定以下として処理時間を $O(1)$ にできる（多角形の均一分布を仮定）．この方法の長所は，画面分割と異なり複数バケットにまたがる図形の重複管理の問題がないことである．

3. SPZBによる隠面・隠線消去処理の手順

SPZBに多角形面 N 個を逐次入力しながら奥行き（視点からの距離）を比較し，既設面の不可視部の新入力面への置換を繰り返して隠面・隠線消去を実現する．以下，1面あたりの頂点・交点数，重なり数の平均値が定数 k より小で，かつ図形がほぼ均一に分布するとし，処理手順と計算時間のオーダーを示す．

I. 面の法線方向を調べ全裏向き面を削除：$O(N)$

II. $m = N/k$ 個の道しるべ g_i を設定：$O(N)$

II 1. 入力面数 N を数える：$O(N)$

II 2. 図4の道しるべ g_i を入力する：$O(N)$

III. 以下 $O(1)$ の処理を N 回繰り返す：$O(N)$

III 1. 1個の入力面Aの外形を入力：$O(1)$

① 面Aの最初の頂点 $a_0(x_{a0}, y_{a0})$ を含む SPP_{a_0} を検索．検索法は図4，図3：$O(1)$

② SPP_{a_0} を切断し，短い辺 a_0t（EPP）を挿入（ポインタ付け替え）**図5**(a)：$O(1)$

③ Aの外形沿いに第 i 頂点から入力辺 a_it $(i = 0, 1, 2, \cdots)$ を伸張し空間分割の変化点（イベント $E_j(j = 1, 2, \cdots)$）でEPP，SPPを更新する（図5(b)）．E_j は a_it が前方の小空間 SPP_t の前方を向いての左右側辺または B_x と交差する点，またはAの次の頂点 a_{i+1} のうち最も近い点である（4.2節の表1）．E_j 以外では小空間の形が変化してもトポロジーは不変なので更新不要．E_j だけの間欠処理となる．辺伸張がAの外形を一巡したら辺を閉じ，外形辺入力を終了する．A近傍の局所処理なので：$O(1)$

III 2. Aと重なる面の不可視部をAと置換：$O(1)$

① 重なり検索がA外に行かぬよう，Aの外形に乗り越え禁止フラグを付加し，Aの内側へEPP，SPPをたどり，EPPで囲まれた小区画（図5(c)の pa_1q，$b_0b_1b_2$，rsa_2）を重なりとして検出する．小区画ごとにSPPに付加した面番号から既設面の方程式の係数を調べてAと奥行き比較し，Aが可視の場合に

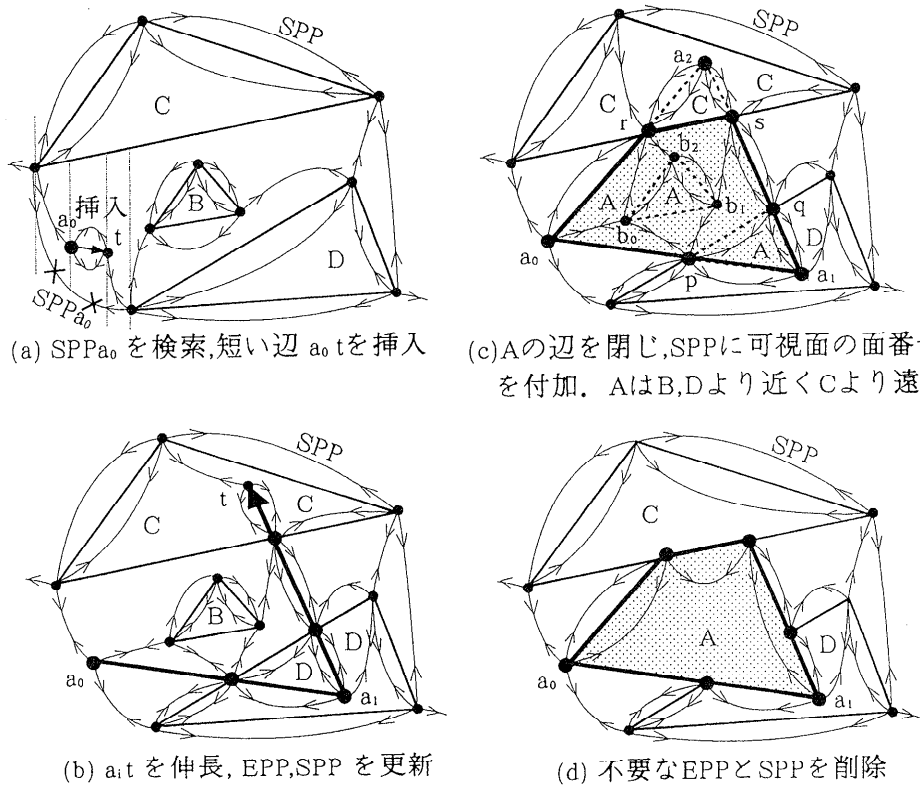


図5 面Aの入力と隠線処理手順
Fig. 5 Hidden line processes for an input face A.

は同図の pa_1q, $b_0b_1b_2$ のように，面番号をAに変更する．重なり数 $< k$ により：$O(1)$

② 隣り合う小区画が同一面となった境界辺のEPP（図5(c)波線部）および不要なSPPを削除して同図5(d)となる：$O(1)$

◆以上の結果，SPZBには可視面だけが残る．

IV. 道しるべ g_i を削除：$O(N)$

V. 最後に面を表示すれば隠面消去であり，辺を表示すれば隠線消去表示になる：$O(N)$

以上の処理時間はすべて $O(N)$ 以下のため全体も $O(N)$ となる．**図6**(b)は(a)のデータの実測値で，N の小さい辺りでは不可視図形の削除が少なく傾斜がゆるいが，傾斜一定の直線に漸近し，$O(N)$ を示している．

一方，メモリ容量は1頂点平均約60バイトで，不可視データの削除により $O(N^{0.4})$ となった．

4. 計算誤差による誤動作と暴走の対策

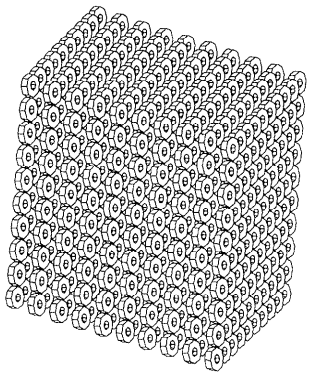
4.1 奥行比較の誤差による不可視面の誤表示防止

交差線なしの貫通面は，事前の三次元対象空間での図形集合演算（本論文の範囲外）で分割しておく．

さてSPZBでは既設面と新入力面の辺はともにEPPで表現され，図5(c)のように全体がEPPで囲まれた小区画に分割されている．入力に貫通面がなければ奥行き比較は小区画ごとに，どこか1点ですればよい．しかし三次元集合演算で貫通の検出ができない計算誤差の深さの貫通面で誤らないよう，次の対策をする．

（対策1） 小区画が表示上無視できない大きさの場合．面は平面でかつ貫通がないから，既設面と新入力面が三次元空間で平行でなければ，小区画の縁（へり）で奥行き差が最小になる．このため縁でなく中心付近で奥行き判定して誤りを減らす．今回は簡単のため小区画の外形頂点の平均位置で判定した．両面がほぼ平行で，中心付近での奥行き差が誤差範囲の場合には誤りは不可避であるが，そんなケースはまれである．

（対策2） 小面積のため不可視としてもよい場合．隣接面を検索し，これに隠された不可視面の形にして削除する．稜を挟む両側の面が，計算誤差でわずかに重なったり隙間が空いて投影された部分がこれに当たる．



(a) 隠線消去：表向きポリゴン10k個，表裏全ポリゴン25k個

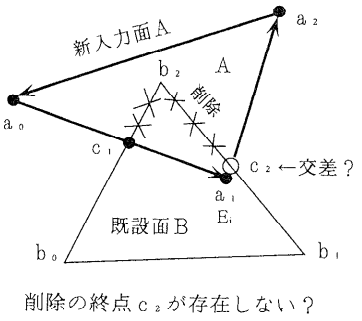


図7 計算誤差による矛盾と暴走
Fig. 7 A failure caused by calculation errors.

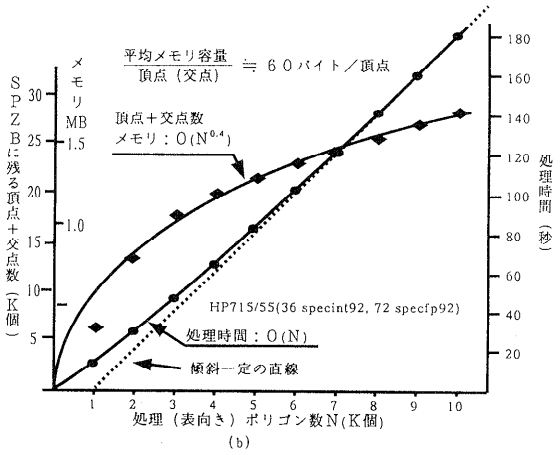


図6 陰線消去データ例 (a)，処理時間とメモリ容量 (b)
Fig. 6 An example of hidden line removal (a), processing time and memory consumption (b).

なお誤差の範囲で視線に平行な面は表示の必要もないから，事前に裏向き面と同時に削除しておく．

4.2 計算誤差によるトポロジー矛盾と暴走対策

4.2.1 外形辺入力時のトポロジー矛盾防止

図7で面Bが入力面Aより遠方なら辺 $c_1b_2c_2$ を削除する．しかし計算誤差で辺 a_0a_1 と a_1a_2 がともに b_1b_2 と交差したり，両者とも交差しないとトポロジーが矛盾して削除プログラムが暴走する．これに対しSPMでは「a_it の先端 t の小空間境界通過（c_2 等）の計算誤差による見逃しや二重通過を防止」して対策する．この原理は文献3)と同じであるが，今回は空間分割法を単純化し（付録参照）プログラムを簡素化した．

表1はこの原理の具体化で，a_it を伸張中，次のイベント E_j の候補を引数とし，X，Yの辞書順で最も手前の候補を正しい E_j として選ぶ関数 nearest() の引数を，t の前方の SPP_t で表す小空間の形状別に示す．プログラムは SPP_t の接続先で表1の横軸を，a_it の左右の辺との交点 C_L, C_R の有無から縦軸を求めて，nearest() の引数を定める．なお引数 C_B は B_x を形成する頂点を，a_{i+1} は A の次の頂点を，∞ は E_j になりえない引数を示す．たとえば表1の⑤では C_R は存在せず，辺の端点 C_B や a_{i+1} は交点 C_L より辞書順で必ず遠方なので ∞ にして，これらが計算誤差で E_j になることを防ぐ．実際図7で a_0a_1 が交差しなければ，b_1b_2 に関して a_2 が a_1 の反対側のため，a_1a_2 上に C_L が存在するとして表1の⑤が選ばれ，E_j は必ず C_L（図7では c_2）で，検出ミスは起きない．なお C_L の座標値が計算誤差で a_1a_2 や b_1b_2 の端点より外側になる場合は，内側の端点の位置に修正して図形歪みを減らす．次に c_2 が a_0a_1 上にあれば，c_2 設置の後は a_it は b_1b_2 の左側になり，a_2 が同じ左側なら決して再交差しない．なおX方向フラグは a_ia_{i+1} の間では，誤差を含む C_L の座標値によらず a_i と a_{i+1} の辞書順方向に固定する．そうしたX方向フラグを使えば，注目点の移動でEPP，SPPをたどる間に交点座標値の誤差で方向を誤ることがない．

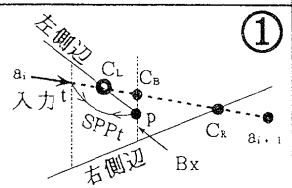
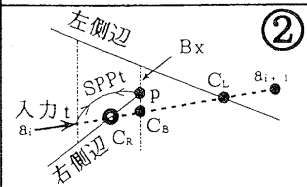
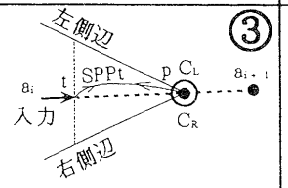
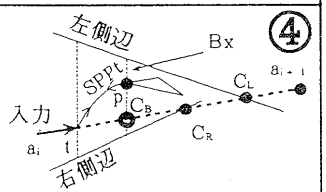
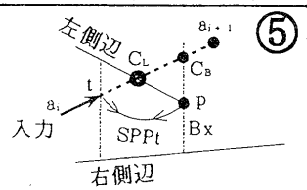
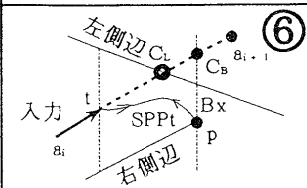
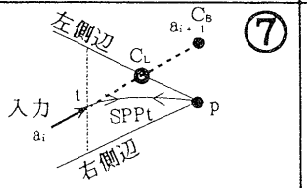
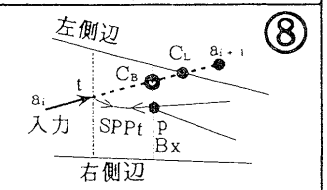
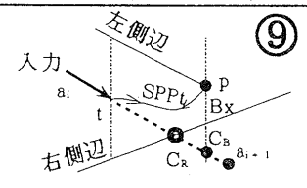
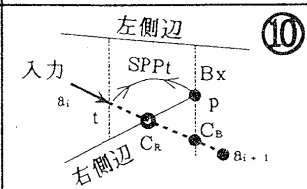
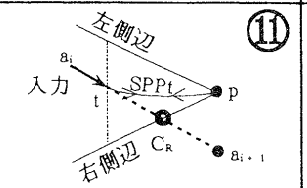
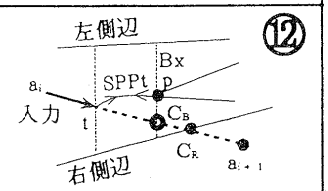
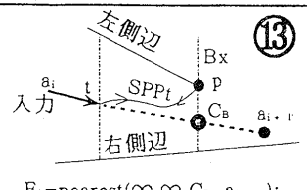
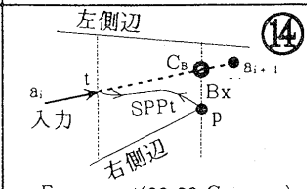
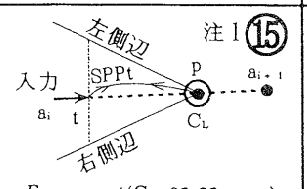
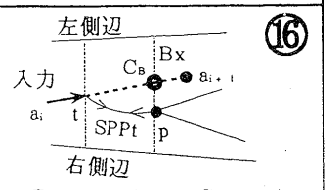
特に同表⑮は左右共に交差の検出がない場合で，もし a_{i+1} が交点pより遠ければpと同じ座標の交点 C_L を強制設置し，交点なしの境界突き抜けを防止する．

その他，多角形の隣りの辺（$a_{i-1}a_i$ と a_{i+1}）や一度交差した辺（直線）は再交差しないことも制約としてプログラムに組み込む．なお表1は a_it の向き（X軸方向＋－）によらず使用できるため，文献3)より場合分けの数が減り，プログラムが簡単化する．

4.2.2 計算誤差によるその他の問題点

入力多角形を閉じるとき，終点位置判定アルゴリズムが始点と異なるため計算誤差が異なり，終点が始点と同じ小空間に達しないと閉じられず暴走する．また削除は外形辺を切断して長さゼロにまで短縮する方法

表1 正しい次のイベント E_j を求める nearest(C_L, C_R, C_B, a_i+1) の引数決定表
Table 1 Deciding the arguments of nearest(C_L, C_R, C_B, a_i+1) to get the next correct event E_j.

入力と左右辺との交差 C_L, C_R	入力辺a_it前方の入力空間を表現するＳＰＰｔの先端pの接続先による分類			
	SPPtが左側辺の先端pに接続	SPPtが右側辺の先端pに接続	SPPtが左右両辺の交点pに接続	SPPtは左右辺でない頂点に接続
左側辺との交差C_L，右側辺との交差C_Rが共に検出された	①  E_j=nearest(C_L,∞,∞,∞);	②  E_j=nearest(∞,C_R,∞,∞);	③  E_j=nearest(C_L,C_R,∞,∞);	④  E_j=nearest(C_L,C_R,C_B,∞);
左側交差C_Lを検出 右側は交差なし	⑤  E_j=nearest(C_L,∞,∞,∞);	⑥  E_j=nearest(C_L,∞,C_B,∞);	⑦  E_j=nearest(C_L,∞,∞,∞);	⑧  E_j=nearest(C_L,∞,C_B,∞);
左側は交差なし 右側交差C_Rを検出	⑨  E_j=nearest(∞,C_R,C_B,∞);	⑩  E_j=nearest(∞,C_R,∞,∞);	⑪  E_j=nearest(∞,C_R,∞,∞);	⑫  E_j=nearest(∞,C_R,C_B,∞);
左右辺共に交差せず	⑬  E_j=nearest(∞,∞,C_B,a_{i+1});	⑭  E_j=nearest(∞,∞,C_B,a_{i+1});	注Ⅰ ⑮  E_j=nearest(C_L,∞,∞,a_{i+1});	⑯  E_j=nearest(∞,∞,C_B,a_{i+1});

注（全体に）：∞印はＥｊになり得ない引数，◎ 印は上図の場合 nearest()から返る正しい次のイベントＥj.
注Ⅰ：左右辺の先が閉じて，入力 a_itが左右辺共に交差せず p点より先に延びるのはトポロジーの矛盾： 対策として片側辺との交点C_Lを強制設置.

で行うが，短縮辺の端点が左右空間の境界を通る順序の判定を計算誤差で誤ると，EPP，SPP の更新結果が矛盾して暴走する．対策は両者とも文献3) と同じなので説明を略すが，計算誤差があっても矛盾したトポロジーを作り出さぬようにすれば暴走は防止できる．

4.3 誤動作防止のテスト

図8 の図形を少しずつ三次元的に回転させて 10^6 回表示し（CRT 上では滑らかな回転に見える），正常動作を確認した．**図9** は暴走防止テストで，隠面消去 (a) と同じデータの交点に乱数誤差を加えて 10^4 回隠線消去表示した (b)．誤差が大きいので図形歪や不可視面の誤表示は有るが暴走しないことを確認した．

5. おわりに

スクリーンをスペースモデルで表現する隠面・隠線消去共用の方式を開発した．本方式は処理時間 $O(N)$，メモリ $O(N^{0.4})$（実験値），計算誤差による不可視図形の表示や暴走がない，図形ピックの他図形演算が可能等の特徴がある．処理速度の絶対値の従来方式との比較は今後の課題と考えている．問題点はポインタの多用でプログラムが少し複雑なことであるが，簡単なデバッグルーチンの組込みで対処できた．なお，試作プログラムは C 言語で約 5k 行になった．

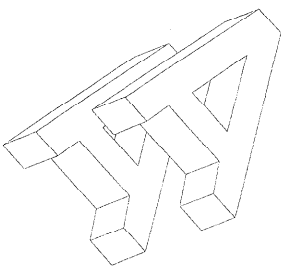


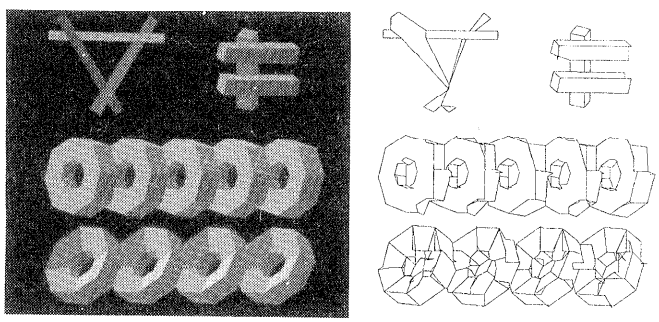
図8 三次元回転図形の隠線消去
Fig. 8 Hidden line removal with three dimensional rotation.

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，川崎，岩本：スペースモデルによる二次元図形処理システムの試作：注目，局所集合演算，運動図形の衝突検出，情報処理学会論文誌，Vol.29, No.10, pp.933–943 (1988).
3) 大沢 晃：計算誤差による暴走のない図形演算アルゴリズム：スペースモデルによる実現，情報処理学会論文誌，Vol.31, No.1, pp.42 55 (1990).
4) 今宮淳美：隠面除去アルゴリズム，情報処理，Vol.24, No.4, pp.539–546 (1983).
5) 伊理正夫，杉原厚吉：計算誤差を考慮した幾何的アルゴリズム，情報処理学会アルゴリズム研究会資料，アルゴリズム 1–1 (1988.5.23).
6) 杉原厚吉，伊理正夫：計算誤差による暴走の心配のないソリッドモデラの提案，情報処理学会論文誌，Vol.28, No.9, pp.962–974 (1987).
7) 太田ほか：配管レイアウトシステムの開発，日本機械学会論文集（C編），Vol.52, No.474, pp.607–615 (1986).
8) Appel, A.: The Notion of Quantitative Invisibility and the Machine Rendering of Solids, *Proc. ACM Nat. Conf.*, p.387, Thompson Books (1967).

付 録

図10 (a) は今回の空間分割法で，文献1) の方法 (b) より SPP の数が増加するが，頂点の凹凸によらず分割ルールが単純でプログラムが単純化できる．



(a) (b)

図9 隠面消去 (a) と乱数誤差印加隠線消去 (b)
Fig. 9 Hidden surface removal (a), hidden line removal with random errors (b).

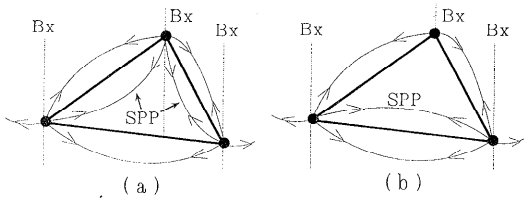
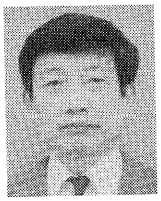


図10 今回の空間分割方式(a)と文献1)の方式(b)
Fig. 10 Space division method in this paper (a), and that in Ref. 1).

(平成8年2月9日受付)
(平成9年2月5日採録)



大沢 晃（正会員）

1936年生．1961年名古屋大学大学院工学研究科修士課程応用物理専攻修了．同年（株）日立製作所入社，制御用計算機回路，半導体用CAD等開発．1992年中部大学工学部工業物理学科教授．現在に至る．スペースモデルによる図形処理の研究を行う．工学博士．情報処理学会，電子情報通信学会会員．