

代数曲面を含む三面図の解釈

井上正博 金昌憲 西原清一
筑波大学 電子情報工学系

代数曲面を含む3次元シーンを三面図から復元する手法を開発した。本手法は、曲線や曲面を多角形や多面体で近似することなく、直接復元するものである。曲面部に起因する接線シルエットなどを表す補助線を導入することにより、既存の多面体復元法を拡張して適用することが可能となった。また、面図間の単純な対応によって得られた候補中の虚要素を排除する面の組合せ探索に、知識を導入して処理速度の効率化を図った。実験によって、基本的な代数曲面は効率よく復元できることを示した。

UNDERSTANDING THREE ORTHOGRAPHIC VIEWS INCLUDING ALGEBRAIC CURVED SURFACES

Masahiro Inoue Kim Changhun Seiichi Nishihara

Inst. of Inf. Science & Electronics, Univ. of Tsukuba

Tennodai 1-1-1, Tsukuba, Ibaraki 305, Japan

A system that reconstructs solid models with algebraic curved surfaces from three orthographic views is described. In this paper, we presents two features. First, by introducing two kinds of additional lines, silhouette lines and tangency lines, we can directly reconstruct solid models with algebraic curved surfaces without using a method of polyhedron approximation. Second, we develop new rules and heuristics to handle curved surfaces used by the combinatorial search of faces. Rules work to reduce the processing time, and heuristics are aiming at simulation of human's understanding way of interpretation. Soundness and efficiency of the method is also proved experimentally.

1 はじめに

物体の形状モデリングを行う伝統的な手段として、三面図が最も広く用いられてきた。そこで、三面図を用いて計算機内に構造化した3次元データを生成し、ソリッドモデルを構築する手法が望まれるようになった。

三面図からソリッドモデルの復元に関する従来の研究[1-3]では、平面だけから構成される多面体が中心的に扱われたが、最近、実用上重要な曲面を含む立体モデルを復元する研究[4-7]が、いくつか行われている。曲面を含む立体モデルの復元手法は、曲面を多面体で近似する手法と近似しない手法の2つに大きく分けられる。多面体近似による手法は、復元過程において、処理が簡単になることや、すでに完成された多面体復元手法をそのまま使えるという利点があるが、近似による誤差が大きく生じる(図1参照)。また、誤算を小さくするために分割数を増やすと、3次元要素が増え処理効率が悪化する。一方、多面体近似しない手法、すなわち、曲線をそのまま扱う手法は、曲線や曲面の取り扱いが、直線や平面より煩雑になることや、幾何学的な場合分けや、演算量が増えることなどの問題点がある。

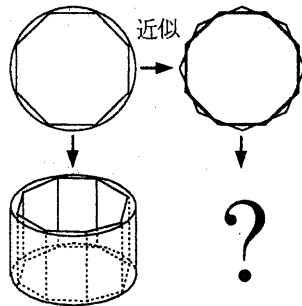


図1 円の8分割近似

本論では、代数曲面を含む物体を対象とし、三面図を解釈してそれに矛盾しない立体モデルを復元する手法について述べる。本手法の特徴は、

- ・多面体手法、すなわち平面の組合せ探索を行う方法を拡張して、曲線、曲面をそのまま取り扱えるようにした点
- ・幾何学的規則に加えて、図面解釈時に人間が無意識に適用している経験知識を導入した点

である。

以降2章では本問題の定義を与え、3章では、入

力データから物体を構成する3次元面の全候補の復元、すなわち、前処理について述べる。4章では、虚物体を排除するフィルタリングプロセス、すなわち、復元の主処理について述べる。最後に、5章では実験結果を述べる。

2 諸定義

この章では、復元するシーンの構造やそれに対応する三面図の性質を明らかにしたのち、三面図解釈門持阿の定義を与える。

2.1 3次元シーン

復元シーンは、3次元空間内に置かれた1つ以上の物体である(図2参照)。各物体は複数の‘面’の集合によって境界付けられる。面は、同一平面上あるいは同一代数曲面上の一続きの‘稜線’によって境界付けられた有向平面である。稜線の両端点は‘頂点’と呼ぶ。各物体は、頂点、稜線を共有しても良い。

また、復元シーンは平面または、表1の制限下の2次曲面を含む物体で構成される。これらの2次曲面のカットは平面のみである。2次曲面の‘配置’で、その物体の回転軸が、任意の座標軸に平行な場合を‘正立’、任意の座標平面に平行な場合を‘半正立’と呼ぶ。

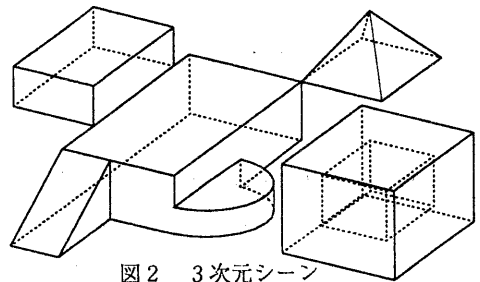


図2 3次元シーン

曲面種	配置	カット(平面)
球面	自由	任意の座標軸に平行
円柱面	半正立 正立	回転軸に平行な座標平面に垂直 任意の座標軸に平行
円錐面	正立	任意の座標軸に平行
円環面	正立	任意の座標軸に平行

* 球面以外は、回転軸を含む平面によるカットも可

表1 曲面種と条件

2. 2 三面図

本復元アルゴリズムは三面図を対象とし、入力には三面図を表す‘点’、‘線分’のデータである。線分には、直線分と曲線分（円、楕円、円弧、楕円弧）があり、隠れ線を破線、補助線を一点鎖線で表す。表1による曲面の制限によって、三面図上に現れる曲線分は、円、楕円、円弧、楕円弧で、傾いた楕円は描かれない。各面図で、閉じた線分列が示す区域を‘領域’という。一般に、これ以上分割不能な状態を‘単純’と呼び、逆に分割可能な状態を‘併合’と呼ぶ。背景と接する最も外側の領域を構成する線分を‘輪郭’と呼ぶ。

曲面を扱うために、2種類の‘補助線’を導入する。1つは、違う面種が滑らかに接する‘接稜線’もう1つは、回転体を面図に投影したとき‘シルエット’の部分に生成する‘シルエット線分’である。これらの‘補助線’を面図に投影して新たにできる線分を‘補助線分’と呼ぶ。補助線分は、入力データに描かれているものとする。図3に三面図の例と単純な対応で復元される候補稜線の例を示す。

2. 3 復元問題

三面図を構成する線分の集合をSとし、すべての候補面の集合を Φ とすると、復元される物体は、集合 Φ の部分集合T ($T \subseteq \Phi$)によって表現される。復元問題は、虚要素を排除するフィルタリング問題であると同時に、次の2つの条件を満足するすべての解を見つけだす面の組合せ探索問題である。

【条件1】 Tは、1つ以上の物体からなる幾何学的に正しいシーンを形成する。各物体は、体積を持ち、稜線を解さず交差する‘交差面’などは存在しない。

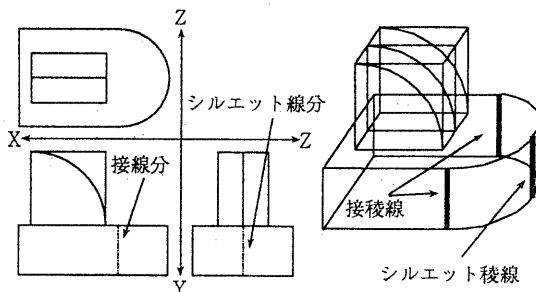


図3 三面図の例と補助線

【条件2】 Tによって形成されるシーンから得られる投影面図中に含まれるすべての集合をS'とするとき、 $S' = S$ が成り立つ。

2. 4 復元処理

復元処理手順は、図4に示すように、頂点の復元、稜線の復元、候補面の復元の順に考えられるすべての‘3次元要素’を復元した後、フィルタリングプロセスによって虚要素を排除する。

3 3次元要素の復元

入力された三面図から、面図間の対応をとりながら各要素を復元する。3次元空間内だけでなく、場合によっては面図に投影し、2次元の処理にして単純化を図った。

3. 1 頂点の復元

頂点は、三面図のすべての面図に点として現れる。頂点とその投影点は、その共通する座標系において同じ座標値を持つ。この関係を利用して面図間で対応をとり、頂点を復元する。ここで、頂点に対応する点が線分の中に埋もれてしまう（点の縮退）場合は後の処理で復元する。

3. 2 稜線の復元

三面図上で、直稜線は直線分あるいは点として現れる。また、曲稜線は曲線分あるいは直線分として現れる。三面図の各面図において、頂点に対応する点間に線分が存在していれば、この頂点間を結んで稜線を復元する。多面体の場合、頂点間を結ぶ稜線はただか1つであったが、曲線の導入によって複数の可能性がある。対応のとれるすべての稜線を復元する。

3. 3 平面、曲面の設定

平面は、同一直線上にない3つの点から設定される。曲稜線の両端点と中心点、同一直線でない2つの直稜線からすべての平面を設定する。その後、同

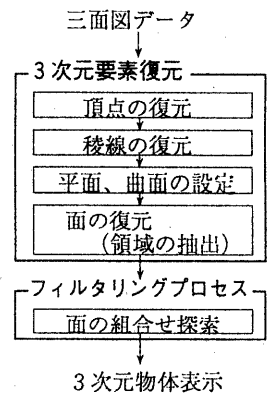


図4 復元処理の流れ

一平面上の稜線を集めて平面の要素として登録する。点の縮退で復元できなかった頂点を復元するため、その平面上で交差する稜線があれば分割し、その交点を頂点に登録する。

曲面は、平面の設定の手法を拡張して、2つの稜線から表2のように設定する。2つの稜線からの曲面の設定例を図5に示す。曲面も平面と同様に、同一曲面上の稜線を要素として登録する。

3. 4 候補面の復元

先に求めた平面、曲面において、同一平面、同一曲面上にある稜線の集合を用いて、面を復元する。この閉じた稜線列を求める際に、三面図上の閉じた線分列である領域を用いて2次元の処理として、簡単化を図った。

平面の場合、最も平行に近い面図に投影し、曲面の場合は表3のように投影した。ここで、平面における稜線と投影線分とは1対1である。しかし、曲面の場合、これが保証されない。よって、投影処理を曲面体を上下2つに分割して行う(図6参照)。また、回転軸(中心点)の位置を考慮し、同じ側にある領域を求める。

線種 \ 面種	平面	球面	円柱面	円錐面	円環面
直稜線 & 直稜線	○	×	×	×	×
直稜線 & 曲稜線	×	×	○	○	×
曲稜線 & 曲稜線	×	○	○	×	○

表2 稜線種と設定曲面

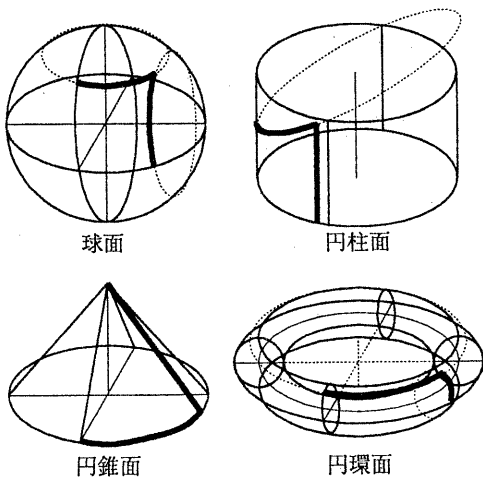


図5 曲面設定の稜線組の例

面種	投影面図	上下2分割処理
平面	最も平行に近い面図	必要なし
球面	任意の面図	必要
円柱面	回転軸に平行な面図	必要
円錐面	回転軸に垂直な面図	必要なし
円環面	回転軸に垂直な面図	必要

表3 投影面図と分割処理

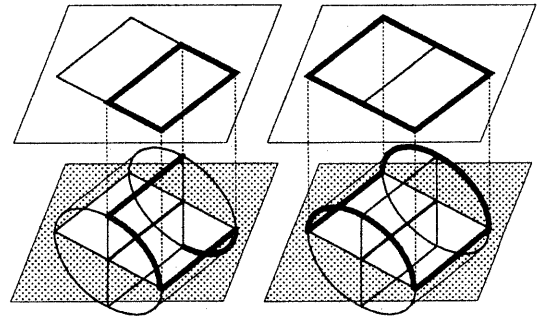


図6 曲面の面図投影

領域は、点を共有する複数の線分から最左・最右線分を選び出し、半時計回りの線分列で抽出される。直線分の場合、最左・最右線分の設定は、共有する点からもう1つの端点への単位ベクトルを用いて行った。しかし、曲線分では、例えば図7の⑤と⑥のようにこのベクトルは使えない。そこで、その点位置での接線ベクトルと、曲率を用いて問題を解決した。

得られた領域を3次元にもどして、面を復元する。図3からは図8の候補面が復元される。

面を面図に投影したときの領域が、ある単純領域を内部に持つ場合、この面をその単純領域の‘対応面’と呼び、視点に最も近い面を‘最前面’という。また、稜線を介して隣接する2つの面が滑らかに接続するときは‘滑接’しているという(図8参照)。

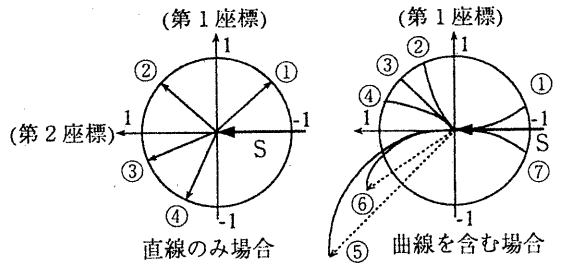


図7 最左、最右線分

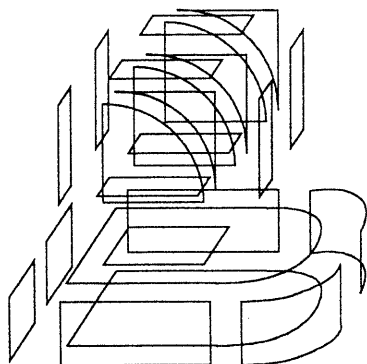


図8 候補面

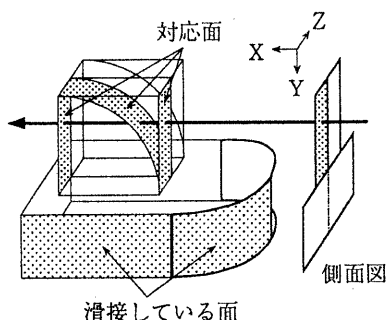


図9 対応面と滑接面の例

4 フィルタリングプロセス

この章では、3章で得られた候補面の中から虚要素を排除する方法を述べる。単純な面の組合せ探索でなく、知識を導入して処理の高速化を図った。

4.1 面の組合せ探索

候補面全体の集合中を中心と考えると、復元は、解を構築する‘真’の面と、構築に参加しない‘偽’の面を決定する操作で表される。この真偽を面の‘状態’と呼ぶ。よって、候補面数 n のとき、面の組合せ数は 2^n である。この組合せの中に、条件1、条件2に合う三面図に矛盾しない解が、一般には複数個含まれている。

4.2 知識の導入

2^n をすべて調べる探索木は完全二分木になる。しかし、単純なバックトラックによる方法では、無駄な探索が多く処理効率が悪い。そこで、次の3つの知識を用いて処理効率の向上を図る。

【知識1】 幾何学的に3次元物体が存在するための局所的拘束条件や特徴を利用して、候補面の真偽

を決定する。

【知識2】 人間が三面図を解釈するとき、視点に最も近い面、つまり、可視部から経験的に正しいと思われる面を決定し、その後、その面の決定状態から派生する拘束条件（‘波及効果’）によって、他の面の真偽決定を行っている。

【知識3】 人間は、見えないところは単純なものを想像する傾向がある。

[1] Rule_1

知識1は、次の2条件からなる。

立体の存在条件：

- (1) 任意の稜線に対し、その稜線を共有する面の個数は、2以上の偶数個である。
- (2) 稜線を介さず存在する複数個の候補面、つまり、交差面のうち、それらのたかだか1つの面のみが存在し得る。

立体と三面図の一致条件：

- (1) 三面図を構成するすべての線分について対応する稜線が必ず存在する。
- (2) 求めた解を三面図に投影したとき、投影面図が元の三面図と同じである。

上記の条件をより現実の場合に合うように具体化して次の条件を得た。

(Rule_1_1) 3次元物体の投影図における単純領域の対応面は、その領域が輪郭を境界成分として

- ・含まない場合は0以上の偶数個である。
- ・含む場合は2以上の偶数個であり、対応面が2つのみのときはその何れもが真である。

(Rule_1_2) 任意の隣合う2つの単純領域において、その2つの領域の最前面が滑接するとき、2領域が

- ・実線を介して隣接している場合、2つの最前面のうちたかだか1つが真である。最前面が同じ場合は偽である。
- ・破線を介して隣接している場合、最前面の状態は同じである。
- ・補助線を介して隣接し、滑接面が1組の場合、最前面の状態は同じである。

(Rule_1_3) 破線を含む領域に対応する面のうち、破線に対応する稜線を介して併合されない面

が少なくとも1つ、あるいは、その稜線をシルエットとする曲面が存在する。

(Rule_1_4) 破線を境界成分とする領域に対応する面は、その破線位置において、ある面を隠す視点により近い他の面が存在する。つまり、破線に対応する稜線を覆い隠す面が存在する。

(Rule_1_5) 補助線に対応する稜線に接続する面のうち、滑接するたかだか1組の面が真である。

(Rule_1_6) 任意の2つの単純領域の最前面が滑接するとき、2領域が

- ・破線を介して接続する場合、滑接する稜線に接続する他の面が1面のときは、その1面は偽である。
- ・補助線を介して隣接する場合、滑接でない面は偽である。

(Rule_1_7) 線分に対応する稜線が1つで、その稜線に接続する面が

- ・2面の場合、その何れもが真である。
- ・3面で1組のみが滑接していれば、残りの滑接でない1面は真である。

(Rule_1_8) ある線分に対応する稜線が2つで、両方の稜線を境界とする面が存在し、稜線位置において、稜線に接続するその面以外の面で、滑接でない組ができない場合、その面は真である。

これらのルールを単一または、複数個組合せて面の真偽決定を行う。例えば、対応面の中で、真の面が偶数個で、状態未決定の面が2個の場合、この2つの面が交差面であれば、立体の存在条件と(Rule_1_1)より、2つとも偽に決定される。

[2] Rule_2

知識2より得られる高速化の規則を述べる。

(Rule_2_1) 対応面の少ない単純領域の方が、Rule_1を適応できる。なぜなら、Rule_1で真偽決定できる面は、状態未決定な対応面が1~3個の場合がほとんどであるからである。

(Rule_2_2) R_1C_1を満足する、輪郭を境界に持ち対応面が2面だけの単純領域に隣接する領域に対応する面は、波及効果がある場合が多い。なぜなら、この領域の対応面は確実に存在するため、稜線の状態未決定な接続面の個数が減少し、立体の存在

条件が適応されることが期待できるからである。

(Rule_2_3) 座標原点に近い最前面の方が、波及効果が大きい傾向がある。なぜなら、最前面が各面図への投影図を端的に表していること、さらに、原点に近い領域の最前面は、他の面図にある輪郭を境界とする領域にも投影される場合が多く、接続関係が分かりやすいからである。

(Rule_2_4) 三面図を描く場合、経験的に正面図、平面図、側面図の順で、各面図にその物体の特徴をよく表す形を配置することが多い。

[3] Rule_3

知識3を基にした、多義解釈における規則を述べる。この規則は、本来処理速度の向上には寄与するものではない。よって、対称性などはあまり考慮せず、簡単かつ高速に行えるようにした。

(Rule_3) 分岐候補面が以下の条件のときは、偽の方を優先的に探索する。

(1)交差面のとき

(2)任意の座標平面に平行でない面の1つ奥の面のとき

(3)最前面で、その1つ奥の面がすべての面図に対応する領域を持つとき

(4)最前面でなく、

- ・その面がすべての面図に対応する領域を持つとき
- ・その面を構成するすべての稜線を介して、同一平面上の面が存在するとき
- ・別の単純領域の真の最前面と、接続する面のとき
- ・対応面の真の数が奇数で、状態未決定の候補面が2つのとき
- ・対応面の奥からn番目の面ので、手前からn番目の面が偽で、2つの面の対応領域が同じとき

4. 3 探索アルゴリズム

2分木探索に知識を適応した探索アルゴリズムを図10に示す。関数Decision-Rule(DR)は、Rule_1を用いて、関数heuristics1はRule_2を用いて、関数heuristics2はRule_3を用いた。探索木において各ノードでは、そのレベルにおける可能なかぎりの面の真偽決定を、DRを用いて行う。

```

begin
  T:=F:={}; U:=Φ; _____ (1)
  heuristics 1; _____ set the first node _____ (a)
  search(T, F, U);
end.
      (i) Main program

```

```

procedure search(T, F, U)
[ while (a face  $f \in U$ ) matching with DR exists) do _____ (2)
  [ decide the state of  $f$  using the DR,
    and move  $f \in U$  to T or F;
    if (any contradiction is found) then return; _____ (3)
  ]
  if (U={}) then _____ (4)
    T gives a solution;
  else
  [ create stack P;
    heuristics 1; _____ get a face  $f \in U$  _____ (b)
    heuristics 2; _____ choose  $\tau$  of  $f$ , _____ (c)
    and push( $\neg\tau$  and  $\tau$ ) to P
    while (P={}) do
      branch (T, F, U,  $f$ , pop-up(P)); _____ (5)
    ]
  ]
]
      (ii) Decision and expansion procedure

```

```

procedure branch(T, F, U,  $f$ , state)
[ move  $f \in U$  to T or F using state; _____ (6)
  search(T, F, U); _____ (7)
]
      (iii) Branch operator

```

図10 探索アルゴリズム

この探索の特徴は、真の決定だけでなく積極的に偽の面を決定している点である。

メインプログラムでは、(1)で、まず、集合に初期値を設定する。Tは真の面の集合、Fは偽の面の集合、Uは状態未決定な面の集合で初期値はΦである。まず、(a)heuristics1で探索開始ノードを選択し、手続きsearchを呼ぶ。

手続きsearchは探索の中心で、面の真偽を決定する。DRで可能なかぎり面の真偽決定を行い、その面の状態にあわせて、集合Uから集合TまたはFへ移動させる。もしここで矛盾を発見したら(3)幾何学的に間違った分岐である。DRで決定できなくなった場合、Uが空なら(4)Tが解の1つである。そうでなければ、(b)heuristics1で分岐のための面をUから1個決定し、(c)heuristics2でその面の優先状態 τ を決定し、(5)優先状態 τ 、 $\neg\tau$ の順に分岐の枝を探索する。

手続きbranchは分岐を行う手続きで、(6)引数stateの状態に合わせて分岐面を集合Uから移動させる。branchとsearchはお互いを呼び合い、(7)間接的な再起手続きを形成している。

5 実験

本手法の復元システムをSparc 1+上にC言語を用いて実装し、実験を行った。複数の三面図において、探索に要するCPU時間を計測し、表4を得た。DRの効果が顕著に現れている。また、解の個数や、候補面の個数が多い複雑な物体に対しては、heuristicsが有効に働いている。多義解釈においても、比較的想像しやすい物体が復元された。また、表2の制限内の曲面を含む物体の復元も行えた。復元システムの出力結果の例を図12に示す。

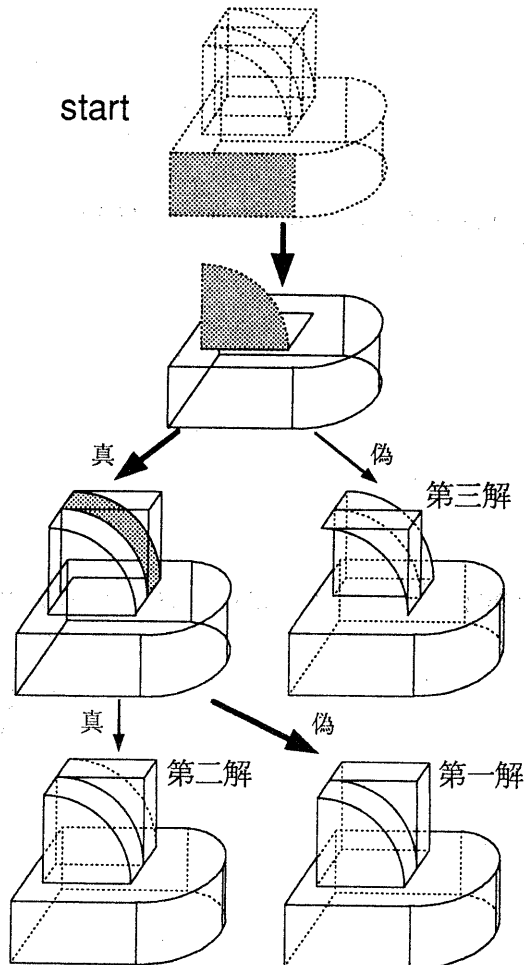


図11 探索過程の例

6 おわりに

三面図を解釈し、曲面を含む物体の復元を行う手法について報告した。多面体ベースの手法を拡張して曲面を比較的簡単に扱うことができた。また、知識の導入により高速処理が、複数解のある場合は、考えられるすべての解のなかで、より自然な解を優先的に復元することが可能となった。

しかし、4章で示した知識は、人間の持つごく一部に過ぎない。また、補助線は実際の三面図では描

かれることは少ない。今後は、補助線の自動発生、あるいは、補助線を利用しない候補面の復元と探索、扱える代数曲面の条件の緩和、知識の充実、さらに、より厳密な曲面同士の交差の判定などを考えている。

参考文献

- 1)西田,張,西原:面の組合せ探索による三面図の解釈,人工知能学会第3回全国大会,437-440(1989)
- 2)井上,西原:計算機による三面図解釈に関する研究(3)筑波大学情報学類卒業論文(1991)
- 3)kim,inoue,nishihara:Understanding Three-View Drawings Based on Heuristics,11th IAPR International Conference on Pattern Recognition Vol1.514-517(1992)
- 4)千田:三面図からもとの立体の自動復元-円柱部分を含む立体への適用-,情報処理学会論文誌,1122-1128(1991)
- 5)宮本,小林,伊藤:外接による三面図からの曲面物体の合成,人工知能学会第4回全国大会,407-410(1990)
- 6)横山,河上:三面図から曲面を含む立体の自動生成,日本機械学会論文誌56巻526号,174-179(1990)
- 7)Sakurai.H :SOLID MODEL INPUT THROUGH ORTHOGRAPHIC VIEWS, CG17.3,243-252(1989)

sample id	# solutions	# candidate faces	CPU time($\times 10^{-2}$ sec)		
			present method		old method [1]
			a*	b*	
1	1	12	0.3	0.3	3
2	1	33	0.5	0.5	149
3	1	47	0.8	0.8	136
4	3	36	1.0	1.0	1498
5	3	28	1.2	1.2	138
6	5	34	5.6	3.7	3336
7	6	21	9.7	9.5	254
8	9	34	9.3	7.5	2301
9	35	20	5.0	5.0	566
10	49	42	22.3	16.4	4167

* a is rule-based method, b is heuristics directed method
表4 実験結果 (復元処理時間)

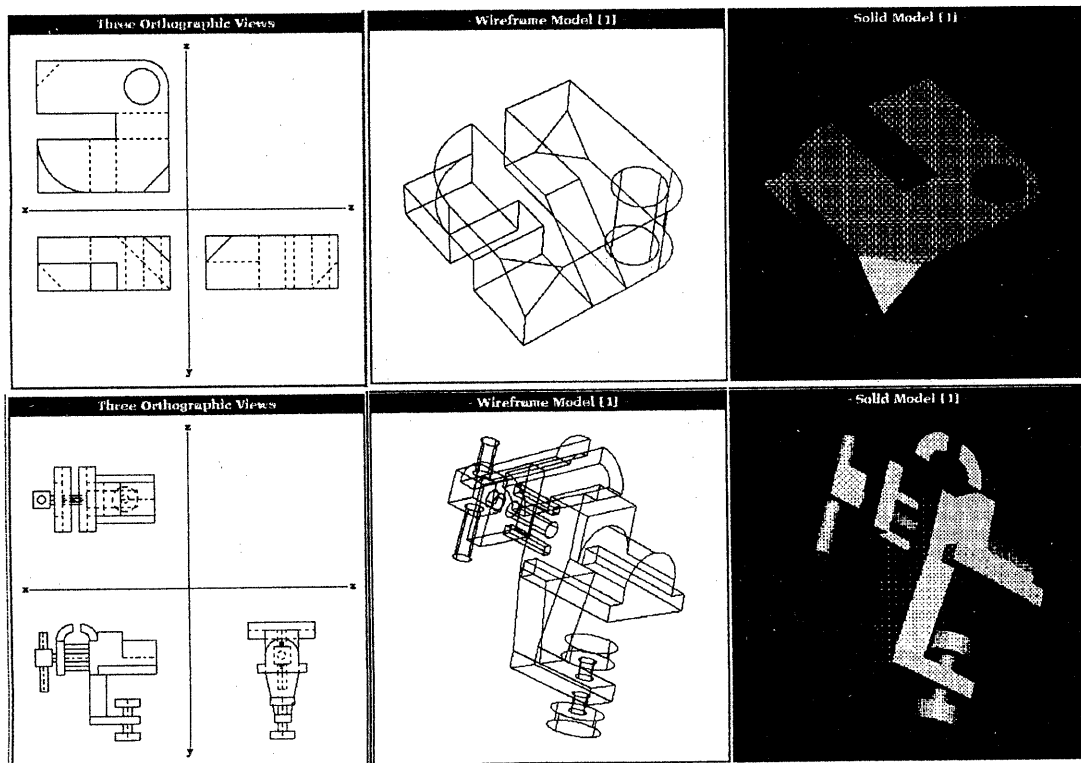


図1.2 復元システムの出力結果の例