

不完全な三面図からの立体モデルの生成

Solid Generation from Incorrect Orthographic Projections.

増田 宏* 沼尾 雅之*
Hiroshi Masuda Masayuki Numao

* 日本アイ・ピー・エム 東京基礎研究所
IBM Research, Tokyo Research Laboratory

1997年3月10日 受理

Keywords: ATMS, orthographic projections, solid models

Summary

This paper proposes methods and a technique for converting incorrect or ambiguous orthographic projections to solid models. First two methods are proposed for handling incorrect draftings that have redundant lines or missing lines, and the third one is a technique to reduce the number of candidates from ambiguous draftings. The first method is for detecting inconsistent lines, which must not appear in orthographic projections. For generating solid models, the system searches for 2D lines that cause inconsistency, and automatically removes them from the original draftings. Such inconsistent lines can be efficiently detected using nogood environments maintained by an ATMS. The second method is for the recovery of missing lines. If the proper wireframe model cannot be obtained from orthographic projections, it is impossible to search for solid shapes that match the draftings. In such a case, candidate missing lines are added in draftings to satisfy the condition that all 2D lines correspond to at least one edge in a 3D model. Boolean equations are re-calculated based on the modified draftings, and are solved using ATMS. Finally, to reduce the number of candidate solids that are generated from ambiguous orthographic projections, we use a technique to introduce additional constraints of cells. Our experiments show that our methods and technique are efficient enough for practical use. We observe that an ATMS mechanism is adequate to solving these equations, because sufficient constraints are derived by examining relationships among 3D models and projections, and described as Boolean equations.

1. はじめに

機械部品の設計においては、2次元 CAD による図面データが多く利用されている。しかし近年では、計算機による干渉チェックや工学的シミュレーションによってリードタイムを短縮しようとする動きが加速しており、立体モデルを必要とする機会が増大している。そのような場合において、図面から自動的に立体モデルが作成できれば、形状モデリングを支援する一手法として非常に有用であると思われる。

三面図から立体モデルを自動的に合成する手法についてはこれまで多くの研究がなされてきた [沼尾 96]。これらの方法を用いることにより、処理効率や冗長性

排除の問題は依然として残るものの、線分が過不足なく書き込まれた三面図からの立体合成問題はほぼ解決したといつてよい。

しかしながら、これらの方法は完全な三面図を前提にしているため、図面に些細な誤りがあっても立体化処理が破綻してしまうという問題があった。現実には設計者が誤りのない完全な図面を作成することは困難である。そのため、立体合成に際しては、試行錯誤的に図面を修正しては立体化処理を行なうという作業が必要となり、作業の効率化には必ずしも結び付かなかった。このような問題を解決するためには、ある程度の図面の誤りを前提とした立体化手法が必要となる。誤りを含んだ三面図からの立体化の研究では、実線/破線

の区別の誤りに対処する研究 [梅澤 93] はあるものの、線分の過不足がある不完全な三面図に対応できる手法はこれまで提案されてこなかった。

筆者らは、三面図からセル分割モデルを生成して立体合成をセルの探索問題に帰着させ、ATMS (Assumption-based Truth Maintenance System) を適用することで高速に立体合成を行なう方法を提案した [増田 94]。この手法も完全な三面図を前提しているが、本論文では誤りを含んだ不完全な三面図からの立体合成問題について論じ、我々の手法が不完全な三面図にも対応できるように拡張できることを示す。まず、2章で筆者らが提案した立体合成法を説明し、3章で ATMS が保持する *nogood* 環境を利用して三面図の矛盾線分を検出、除去する方法を示す。次に、4章で不足している線分を自動的に補うための手法を示し、また5章で三面図が曖昧で極めて多数の立体候補ができてしまう場合の解の絞り込み方法についても示す。

2. セル探索に基づく立体合成

三面図からの立体合成問題では、解となる立体を面の組み合わせ探索として解く方法 [出沢 72] と、基本立体の組み合わせ探索として解く方法 [Wesley 81] が比較的良好に知られていたが、筆者らは本問題をセル分割モデルのセルの組み合わせ探索に帰着させ、ATMSを用いて効率的に解く方法を提案している。まず、この手法について説明しておく。

2.1 セル分割モデルの生成

三面図からセル分割モデルの作成手順を図1に示す。まず、三面図間で対応する線分を調べて、ワイヤフレームを作成し (a)、次にワイヤフレームに面を張っていくことでサーフェスモデルを生成する (b)。最後に面に囲まれた閉空間を探索し、それらをセルとして管理する (c)。

セル分割モデルには、多数の立体の解候補を単一の形状モデルで表現できるという利点がある。図2は、3つのセル C_1, C_2, C_3 からなるセル分割モデルであるが、これらのセルを適当に組み合わせることによって、立体候補が生成できる。図2は $\{C_1, C_2, C_3\}$ と $\{C_1, C_3\}$ を選ぶことで生成できる形状を示している。

2.2 正当化式による立体構成規則

一般に、 n 個のセルからは $2^n - 1$ 通りの立体が生成できるので、この中から三面図に適合した立体を探索すればよい。探索を効率的に行なうために、セルの

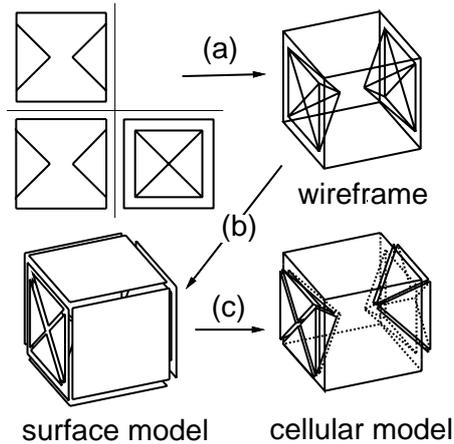


図1 三面図からのセル分割モデル生成

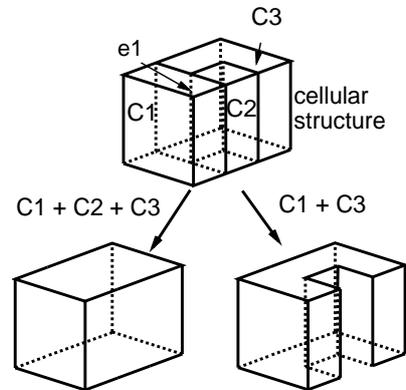


図2 セル分割モデル

存在を仮説 C_i , セルが存在しないという仮説を $\neg C_i$, また、ワイヤの存在をノード e_j , 三面図の線分の存在をノード d_k として以下のような正当化式を ATMS を与え、ラベル更新処理を行なう。

正当化式 1 セルとワイヤの関係: 図2のワイヤ e_1 の回りには3個のセルがあるが、どのようなセルの組み合わせのときにワイヤが立体の稜線として現れるかを、 $C_1, \neg C_2, C_3 \rightarrow e_1$ のように記述する。

正当化式 2 ワイヤと線分の関係: ワイヤ e_i が三面図上の線分 d_j に投影されるとき、 $e_i \rightarrow d_j$ のように記述する。

正当化式 3 線分と三面図の関係: 各面図に含まれる線分を以下のように記述する。

$$d_1, d_2, \dots, d_{n1} \rightarrow \text{top.}$$

$$d_{n1+1}, \dots, d_{n2} \rightarrow \text{front.}$$

$$d_{n2+1}, \dots, d_n \rightarrow \text{side.}$$

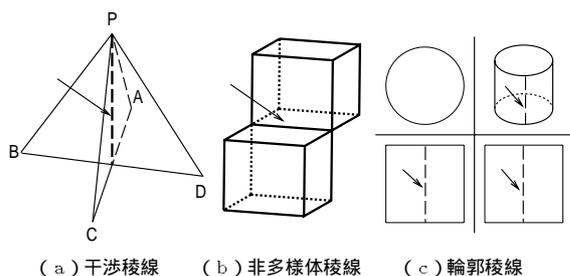


図3 edgeに関するnogood条件

正当化式 4 三面図と立体の関係.

$$top, front, side \rightarrow solid.$$

一方、これらの正当化式に加えて、以下のnogood条件をATMSに与え、nogoodデータベースを計算する。

nogood条件1 干渉稜線: 図3(a)の面PACと面PBDのようにワイヤ以外で干渉線 e が生じるとき、 e が立体に現れるのを禁止する条件 $nogood\{e\}$ を加える。

nogood条件2 非多様体稜線: 図3(b)のような非多様体稜線(3個以上の面に隣接する稜線)を生じるセルの組み合わせをnogoodとする。

nogood条件3 滑らかに連結する曲面: C_1 連続で連結する曲面を含む図面の場合、図3(c)の破線で示したような補助線分がワイヤ生成の過程で図面に追加される。このような補助線分から作られたワイヤが曲面上になかったり、 C_1 連続でないようなセルの組み合わせは元の図面に反するのでnogoodとする。

2.3 ATMSによる解の算出

求めるべき解は、ノード $solid$ のラベルとnogood環境から得られる極大無矛盾環境である。図4は、セルが C_1, C_2 の二つのみの場合の探索空間を示す(この図では $\neg C$ を \bar{C} と記す)。ここで、 $solid$ のラベルが $\{C_1\}$ 、また、環境 $\{C_1, C_2\}$ がnogoodであるとする。このとき、図の破線より上が矛盾した環境となるので、ラベル $\{C_1\}$ を含んだ太線で示した環境のうち、極大無矛盾環境である $\{C_1, \neg C_2\}$ のみが解となる。すなわち、三面図に一致するのは、セル C_1 のみから構成される立体である。

3. ATMSによる矛盾線分の検出

以上述べた手法は、三面図面に矛盾がないことを前提としている。しかしながら、現実には過不足のない完

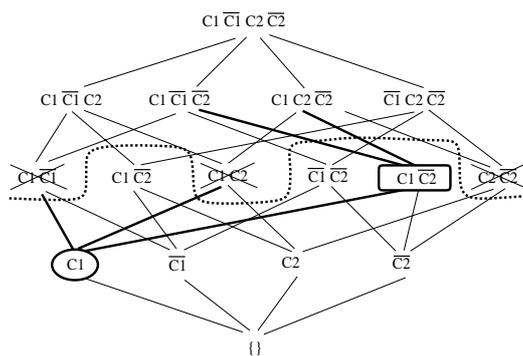


図4 ラティス構造と極大無矛盾環境

全な図面を作成することは簡単な作業ではない。そこで以下において、三面図が誤りを含んでいる場合の立体化手法について論じる。

まず、三面図に過剰な線分が書かれているときに、矛盾線分を検出する方法について考える。三面図に誤った線分が含まれるとき、すべての線分を満たす立体が存在しないことはわかるが、従来の手法では、どこに矛盾が生じたかを調べることは容易ではなかった。ATMSを用いた本手法では、三面図のすべての線分は、立体の投影図として現れるための条件を、セルを仮説とするラベルとして保持している。そのため、線分間の整合性を調べることで、矛盾が検出ができるという利点を持っている。

ある線分が矛盾するかどうかは何を基準に矛盾判定をするかに依存するが、本研究においては、以下の節で述べるような基準を用いることにする。矛盾と判断された線分は自動的に除外され、立体合成が行なわれる。

3.1 面図を基準にした矛盾検出

三面図のうち少なくとも一つが正しければ、正当化式3を解くことによって、三面図を満たす解の必要条件を算出することができる。ここで、正当化式4を $top \rightarrow solid', front \rightarrow solid', side \rightarrow solid'$ と書き換えれば、少なくとも一つの面図を満たす立体の条件を求めることができる。ただしこの式は元の式に比べ、条件が緩いので非常に多くの解が算出されてしまう恐れがある。そこで、このようにして求めた条件を基準にし、 $solid' \cap d$ が空になる線分 d を矛盾線分と見做し、それらを除外した上で、正当化式3, 4を適用する。

この方法は一つの面図が正しければ解を求められるので非常に有効であるが、図面がある程度複雑になるとすべての面図に誤りが含まれることも少なくない。

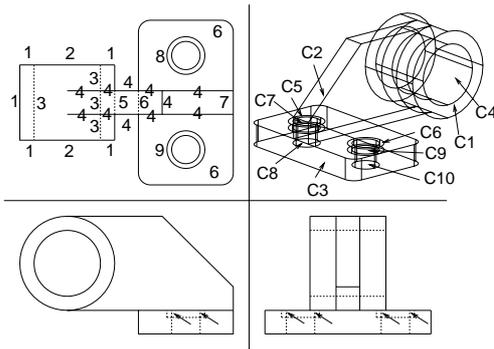


図5 矛盾する三面図とセル分割モデル

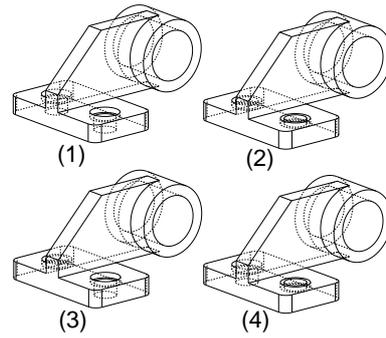


図6 立体の解候補

そのときには $3 \cdot 2, 3 \cdot 3$ で述べるような方法で解探索を行なうことが必要である。

3・2 自明なセルを基準にした矛盾検出

ある線分の存在が唯一つのセルの存在によって支持されるとき、その線分を基準にして矛盾線分を検出することを考える。そのために、次のような条件を加える。

nogood 条件 4 線分 d のラベルがただ一つのセル $\{C_i\}$ であるとき、 $\text{nogood}\{\neg C_i\}$ を ATMS に与える。

セルの存在に直接関与するこのような線分は三面図の特徴的な部分であるので、誤りが生じにくいという仮定に基づいている。また、この条件は、解の探索空間を小さくするので、処理の高速化にも有効である。

図5にこの方法を適用した例を示す。この三面図はすべての面図に誤りを含んでいる。各線分は、図の C_1, \dots, C_{10} を仮説とするラベルを持つ。ここでは、上面図の線分におけるラベルを以下に示す。ただし、同一のラベルをもつ線分には同じ番号を記してある。

- 1 : $\{C_1\}$, 2 : $\{C_1, \neg C_1, C_4\}$,
- 3 : $\{C_1, \neg C_4\}, \{\neg C_1, C_4\}$,
- 4 : $\{C_2\}$, 5 : $\{C_1, \neg C_2\}$, 6 : $\{C_3\}$,
- 7 : $\{C_2\}, \{C_3\}$, 8 : $\{\neg C_3, C_5\}, \{C_3, \neg C_5\}$,
- 9 : $\{\neg C_3, C_6\}, \{C_3, \neg C_6\}$.

nogood 条件 4 に該当する、ただ一つのセルをラベルに持つ線分は、1, 4, 6 であり、 $\{\neg C_1\}, \{\neg C_2\}, \{\neg C_6\}$ が **nogood** となる。その結果、ラベル $\{C_1, \neg C_2\}$ を持つ線分5が矛盾と判定される。実際、線分5は誤りであり、存在してはならない線分である。

矛盾と判定された線分5を除外すると、ノード top のラベルは、

- 環境 (1) : $\{\neg C_4, \neg C_5, \neg C_6, C_7, C_9\}$,
- 環境 (2) : $\{\neg C_4, \neg C_5, \neg C_6, C_7, \neg C_{10}\}$,

環境 (3) : $\{\neg C_4, \neg C_5, \neg C_6, \neg C_8, C_9\}$,

環境 (4) : $\{\neg C_4, \neg C_5, \neg C_6, \neg C_8, \neg C_{10}\}$

の4個になる。ここで、正面図と側面図の線分で環境(1)と矛盾するものを調べると、図5で矢印で示した6個の線分が得られる。これらを除外した上で正当化式3, 4を適用することで、図6(1)の形状を求めることができる。同様にして、環境(2)(3)(4)と矛盾する線分を除去して立体生成を行なうと、それぞれ、図6(2)(3)(4)の形状となる。この例題では矛盾線分の候補が4通りあるが、立体候補からユーザが正しいものを選択することで、どの線分が矛盾しているかが確定する。

3・3 図面の形状特徴を基準にした矛盾検出

次に、形状特徴と呼ばれる、図面の特徴的な部分を基準にして矛盾検出を行なう。形状特徴として何が適当であるかは設計対象に依存するので一概にいうことはできないが、ここでは典型的な形状特徴として、外郭線分と孤立ループを考える。外郭線分とは、図7(a)の太線のように面図の輪郭に相当する部分であり、孤立ループとは図7(b)の太線で示した、すべての頂点に2個の線分が連結する閉ループを指す。外郭線分は比較的投影図が把握しやすく、また孤立ループは工学的に意味を持った形状単位であることが多いので、誤りが混入しにくいと考えられる。実際に筆者らはある設計部門で約100個の三面図を作成してもらいテストを行なった結果、これらの部分には誤り線分は含まれていなかった。外郭ループと孤立ループは、線分間の連結性が保持されていれば容易に探索することができる。これらを構成する線分のラベルの積を求め、それに矛盾する他の線分を検出、除外することで立体候補を算出することができる。

なお、以上述べた方法でも解が求まらないときは、画

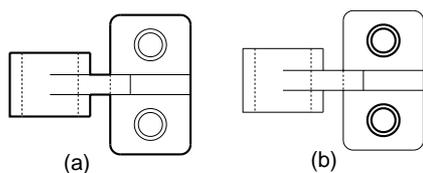


図7 外郭ループ(a)と孤立ループ(b)

面に表示された三面図を介して対話的に矛盾検出を行なうことができる。ユーザが誤りが生じにくいと想定される部分を指定すると、その部分に含まれる線分のラベルの積が計算され、それに矛盾する線分が誤り線分の候補としてユーザに提示される。

4. 不足線分の復元

これまで三面図に矛盾を引き起こす余分な線分が含まれる場合について述べてきたが、三面図に線分が不足する場合には別の手法が必要である。線分が不足していれば必要なセルが作れないため、どのようにセルを選択しても正しい立体を得ることはできない。当然のことながら、線分の足りない三面図には必要な情報が欠落しているため、誤りに対応するにはなんらかの前提条件が必要である。ここでは面図間で対応する線分を書き落とした場合を想定し、ある仮定を設けることで不足線分を自動的に追加することを考える。

4・1 未対応線分の検出

正しく書かれた三面図であれば、三面図のすべての線分は、図1のワイヤフレームの少なくとも一つのワイヤに対応する。もし、ワイヤに対応しない線分があれば、既にこの段階で、三面図を満たす立体が生成できないことがわかる。ここでは、ワイヤに対応しない三面図の線分を未対応線分とよぶことにする。

図1の(c)(d)の3次元化処理では、面やセルの生成に関与しないワイヤは刈り込まれる。従って、処理が進むにしたがってワイヤの個数が減少するので、未対応線分の個数は増えていくことになる。未対応線分の検出と修正は、以下のような手順で行なう。

(1)ワイヤフレームの生成後、ワイヤの生成に寄与しなかった三面図の線分を未対応線分とする。検出された未対応線分は4.2で述べる方法で修復する。

(2)サーフェスが張れないワイヤを消去した後に(1)と同様の処理を行ない、未対応線分があれば面図の修復を行なって、サーフェスモデルを作り直す。

(3)セルを構成できないワイヤを消去した後に(1)

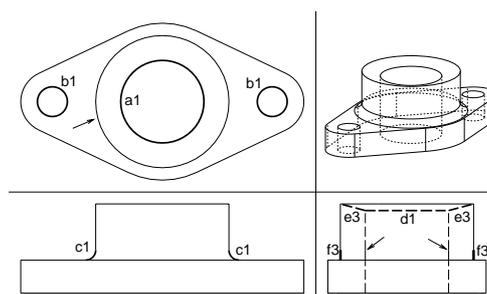


図8 線分の不足した三面図(太線は未対応線分)

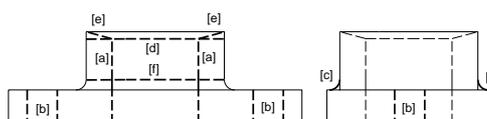


図9 算出された不足線分の候補(太線)

と同様の処理を行ない、未対応線分があれば修復を行なってセル分割モデルを作り直す。

(4)この段階でも未対応線分が存在すれば、それをユーザに提示して修正を促す。

図8は線分が不足している三面図であるが、未対応線分として検出されたものを太線で示している。まず、照合(1)で a_1, b_1, c_1, d_1 が検出され、これらを修正した後は、照合(3)で e_3, f_3 が検出される。

4・2 復元のための規則

ここでは、他の面図に対応する線分がなかったためにワイヤが生成できなかったと仮定して線分を補う。線分を復元する規則は以下の通りである。

(1)孤立ループ：未対応線分が図8の a_1 や b_1 のような孤立ループで、他の面図の一つに、側面図の矢印のような深さ情報があれば、立ちあげ形状と見做して、それに応じた線分を図9[a]や[b]のように補う。

(2)曲面の輪郭線：円弧の未対応線分(図8の c_1)があるとき、それ自身と他の面図の円弧(図8上面図矢印)を輪郭線とする球またはトーラスが定義できるならば、その曲面が存在すると仮定して輪郭線を図9[c]のように追加する。同様に、斜めの直線の未対応線分(図8の e_3)が存在するとき、それ自身と他の面図の同心円(図8の a_1 と矢印の円)から円錐台が定義できるならば、円錐の輪郭線を他の面図に図9[e]のように追加する。

(3)垂直面：未対応線分(図8の d_1)を通り投影

面に垂直な面が存在すると仮定し、両端点から他の二つの面図に水平線または垂直線を図 9 [d] のように追加する。

以上の方法によって補われる線分は、図 9 で太線で示されたものである。これらの線分はあくまでも不足線分の候補であって、必ずしも立体の稜線に現れる必要はない。そのため、正当化式 3 には含めない。その上で正当化式 4 を適用することで立体候補を求めることができる。

なお、不足線分の復元と前章の矛盾検出法は組み合わせて使用される。この例題では図 8 の f_3 が解探索の過程で矛盾線分と判断され、その結果、図 8 に示すソリッドモデルが生成される。

5. 曖昧性の解消

一般に三面図は曖昧であるので、非常に多くの立体が同一の三面図に対応することがある。特に、不足線分を復元した場合には、復元線分を現れても現れなくてもよい線分と解釈して解探索を行なうので、曖昧性の問題が比較的生じやすくなる。そこで、曖昧性を解消するための知識を導入して、候補の個数を絞り込むことを考える。ここでは、曖昧性を解消するための条件をセルに関する条件として記述し、ATMS を用いて解くことを考える。

5.1 セルのグループ化

通常は三面図においては、同一と見做せる部分には同一の解釈を与えるという暗黙の了解がある。この知識により、図 10 (a) から (c) のような立体を一通りに決めることができる。一方、三面図からの立体合成では、たとえば、図 10 (b) で正面図の対角線上の穴の形状が A であれば、残りの 2 0 個の穴は A, B のどちらの形状であっても三面図は同じなので、少なくとも 2^{20} 個の立体が同一の三面図に対応することがわかる。このような穴やスロットを規則的に配置した形状は機械設計では頻繁に現れる。

そこで、セル C_1, C_2 の三面図の各面図への投影図 1, 2, 3 が以下の条件を満たすとき、 C_1, C_2 に同一の解釈を行なうための拘束を追加する。

- (1) 面図 1 で合同な孤立ループとなる。
- (2) 面図 2 で一致する。
- (3) 面図 3 で合同になる。

以上の条件を満たすセルを求めることで、図 10 の例題では 25 個のセルからなる 3 つのグループが得られる。

C_1, C_2, \dots, C_n が同一の解釈になると判定されると、

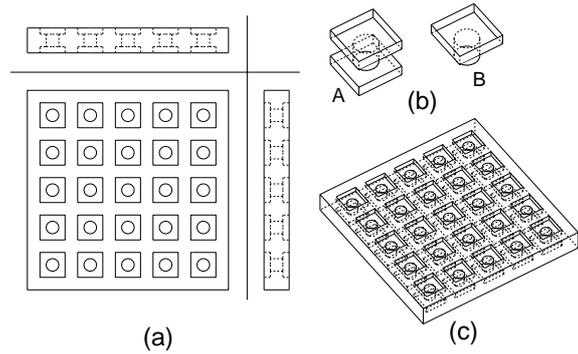


図 10 曖昧な三面図の解釈

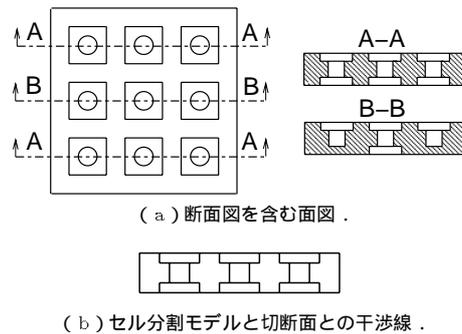


図 11 切断線と断面図の指定された面図

$\text{nogood } \{C_i \neg C_{i+1}\}$, $\text{nogood } \{\neg C_i C_{i+1}\}$ ($i = 1, \dots, n-1$) が ATMS に与えられる。それにより、同一グループのセルがすべて真かすべて偽になり、図 10 の解が (b) の一通りに絞り込まれる。

5.2 断面図による解の絞り込み

実際の図面においては、三面図が曖昧で立体が一意に決まらない場合は、断面図を補助的に用いることがある。断面図は、図 11 (a) の切断線 A-A のように断面図の位置を指定し、ハッチングによって実体部分を表現する。この情報をセルの真偽を判定に反映させることができる。手順は以下に示す通りである。

(1) 投影面と切断線から切断面の方程式を決定する。図 11 (a) では切断線 A-A により、XY 平面に平行な切断面ができる。

(2) 三面図から生成されたセル分割モデルの各面と切断面との交線を求める (図 11 (b))。

(3) 図 11 (b) の各閉領域はセルの境界上かセルの内部に相当する。セルの内部にあるとき、そのセルが実体かそうでないかを与えられた断面図 (図 11 (a)) のハッチングを参照して決める。この結果は、 $\text{nogood}\{\neg C_1\}$

例題	セルの数	幾何計算	推論のみ	合計
図 12	7 個	1.88 秒	0.23 秒	2.11 秒
図 13	7 個	3.18 秒	0.18 秒	3.36 秒
図 14	19 個	1.41 秒	0.46 秒	1.87 秒

表 1 計算時間 (IBM RS/6000-980)

または $\text{nogood}\{C_1\}$ を ATMS に与えることで反映できる。

5・3 ユーザ指定のセルの拘束

以上の手法によっても、ユーザが選択可能な個数にまで候補立体が絞り込めない場合には、一部のセルに対して、ユーザが対話的に真か偽かを指定し、それを nogood 条件として ATMS に追加することで解の絞り込みを行なう。

6. 評価実験

図 12, 図 13, 図 14 の例題に本手法を適用した場合の計算時間を表 1 に示す。この表で、推論のみ、とあるのは ATMS に値が入力されてから解を出力するまでの時間である。矛盾検出や曖昧性解消処理の時間もここに含まれる。幾何処理は、推論以外のすべての処理を含み、三面図ファイルの入力、セル生成、立体モデル生成、及び不足線分の復元の処理時間を含んでいる。使用した計算機は、RS/6000-980 であり、計算時間は、elapsed time である。

図 12 は、矛盾した線分を含んだ例題である。円内の太線で示された 2 個の破線が誤りとして検出された。矛盾検出は面図を基準に行なわれた。

図 13 は、不足線分と矛盾線分を含んだ例題である。不足線分はワイヤフレームとの照合で検出され、矛盾検出は面図を基準に行なわれた。三面図 (a) は入力された図面で、三面図 (b) は自動修正された三面図である。A, B, C は不足と判断された線分、D は矛盾と判断され除外された線分を示している。算出された立体モデルを (c) に示す。

図 14 は、曖昧な図面の例題である。矢印で示した部分が貫通穴かそうでないかによって 3 5 通りの解が存在する (実線破線を区別をしない場合)。セルのグループ化の手法を用いることにより、矢印で示した部分に相当するセル 1 2 個は 3 グループに分けられ、解は図に示した立体一通りに絞り込まれる。

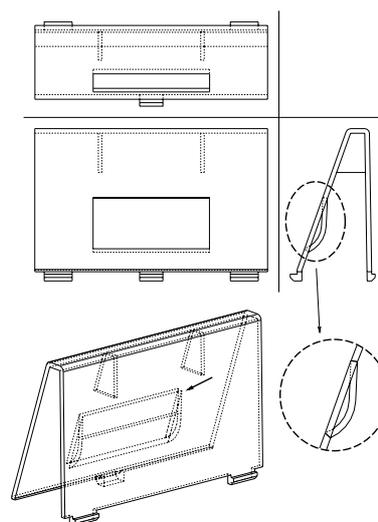


図 12 余分な線分を含んだ三面図と検出された誤り線分

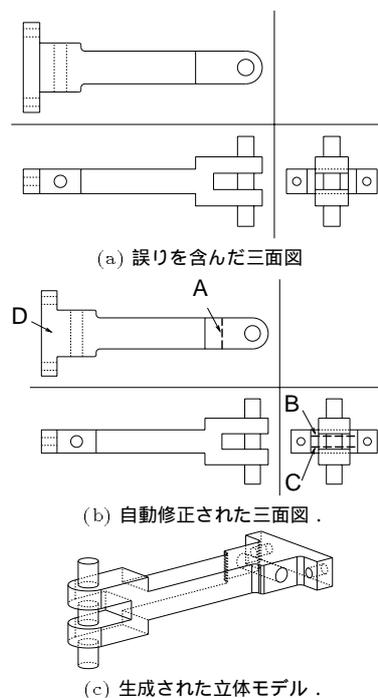


図 13 不足線分と余分な線分のある三面図の例題

7. おわりに

本研究により、設計者の見落としによる単純ミスの多くが吸収でき、図面を完全にするための編集の時間が大幅に低減できることが期待できる。本研究では、以下のことが明らかになった。

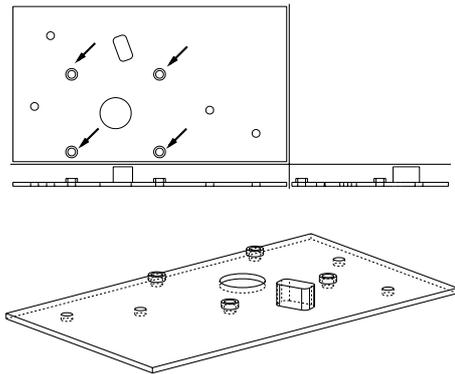


図 14 曖昧な三面図の例題

- (1) 立体の解探索に ATMS を用いた本手法においては、三面図の各線分に、セルの存在を仮説とするラベルを持たせているため、ラベル間の積を計算することで矛盾線分の検出が十分実用的な計算時間でこなせる。
- (2) 三面図での線分の書き落としによる誤りは、主軸に平行か曲面の輪郭に相当する線分であれば、線分の復元規則に基づいて自動的に復元し、立体合成が行なえる。復元できない場合でも不整合が生じた箇所がわかるので、ユーザに変更を促すことができる。
- (3) 三面図が曖昧で、画面での選択が困難なほど多数の候補がある場合、セルに対する拘束条件を ATMS に追加することで十分実用的な計算時間で解の絞り込みが行なえる。

これらの手法は拡張が可能である。矛盾線分の検出では扱える形状特徴の種類を増やすことで、不足線分の復元では復元規則を追加することで、また曖昧性の解消ではセルの拘束条件を生成する知識を追加することで扱える三面図の範囲を広げることができる。ただしその場合でも、あらかじめすべての誤りのパターンを想定することは困難である。そのため、実用上は自動修正だけでなく、ユーザの介入する余地も残しておくことが望ましいようである。そこで、本論文では、矛盾線分の検出、不足線分の復元、曖昧性の解消のそれぞれにおいて、対話的にユーザからの指示を立体化処理に反映する方法も示しておいた。

三面図からの立体合成では、三面図で正確に記述できる立体が対象となり、平面、2次曲面、トーラス面の組み合わせで構成されていることが必要である。筆者らの調査によると、そのような形状は機械加工部品に多く、本手法は機械加工品の立体モデル生成に適しているようである。

また三面図からの立体化手法が有望な応用として、

パラメトリックモデラのインタフェースがある。パラメトリックモデリングでは、パラメータで寸法制御ができる単純な立体（フィーチャと呼ばれる）を組み合わせて複雑な立体を作成していく。現状では、寸法をついた一面図に高さを与えることでフィーチャを生成しているが、三面図からの立体化手法を用いることにより、より複雑なフィーチャが容易に作成でき、三面図の寸法を変更することで柔軟な形状制御ができるという利点が生じる。

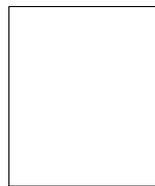
なお、本手法は、筆者らの開発した非多様体形状モデラ [Masuda 93] と ATMS [Shimizu 91] をベースに実装された。ATMS の開発で中心的役割を果たした清水周一氏には感謝の意を表します。

参 考 文 献

- [出沢 72] 出沢正徳: '三面図からの物体生成のためのシステム,' 機械学論, Vol.38, No.310, pp.1267-1276 (1972).
- [Masuda 93] Masuda, H: 'Topological Operators and Boolean Operations for Complex-Based Non-Manifold Geometric Models,' *Computer Aided Design*, Vol.25, No.2, pp.119-129 (1993).
- [増田 94] 増田宏, 沼尾雅之, 清水周一: '非多様体形状モデラと ATMS を用いた三面図からのソリッド合成法,' 情報処理学会誌, Vol.35, No.3, pp.453-460 (1994).
- [沼尾 96] 沼尾雅之, 増田宏: '三面図からのソリッドモデル変換技術の現状,' 人工知能学会誌, Vol.11, No.4, pp.542-549 (1996).
- [Shimizu 91] Shimizu, S., Inoue, K. and Numao, M.: An ATM-Based Geometric Constraint Solver for 3D CAD, *Proc. of Tools For Artificial Intelligence*, pp.282-290 (1991).
- [梅澤 93] 梅澤顕, 菊池浩司, 狩野均, 西原清一: '曲面を含む三面図の矛盾の検出と理解,' グラフィックスと CAD 研究報告 No.63, pp.43-49 (1993).
- [Wesley 81] Wesley, M. A and Markowsky, G: 'Fleshing Out Projections,' *IBM Journal of Research and Development*, Vol.25, No.6, pp.938-954 (1981).

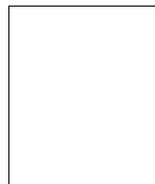
〔担当編集委員: 白井 良明, 査読者: 荒井 英司〕

著 者 紹 介



増田 宏

1985年東京大学工学部精密機械工学科卒業。1987年同大学院修士課程終了。同年、(株)日本アイビーエムに入社、同社東京基礎研究所にて3次元形状処理などの研究に従事。工学博士。情報処理学会会員。 masuda@trl.ibm.co.jp



沼尾 雅之 (正会員)

1981年東京大学工学部電気工学科卒業。1983年同大学院修士課程終了。同年、(株)日本アイビーエムに入社、現在、同社東京基礎研究所にてネットワークセキュリティおよび電子商取引に関する研究に従事。1990年度および1994年度人工知能学会論文賞受賞。現在、人工知能学会誌編集委員。情報処理学会会員。 numao@trl.ibm.co.jp