

立体要素連立式による三面図からの自動立体作成法

田 中 雅 次†

CAD で描かれた三面図から自動的に3次元の立体を作成する方法は、これまで多く研究されているが、これらは解は得られるものの、実際の処理時間が立体形状の複雑さに対して急激に増大する傾向があり、機械部品などの設計現場での実用化には問題がある。本論文では、Wesley と Markowsky のようにサーフェスモデルより面を境界とする3次元の閉領域を立体要素として作成し、これらの中には実際には存在しない偽の立体要素の存在可能性があるが、そのような立体要素間の真偽の関係を、三面図との直接的な対応関係より定式化し、立体要素連立式として解く手法を紹介する。三面図からの自動立体作成は、基本的には面や立体要素など図形要素の組合せ探索の問題であるが、本手法では、この問題を立体要素連立式の解法に変換させることで、実際の処理量を立体形状の複雑さと比例させる程度に押さえた高速な処理を可能とした。また、実際にプログラムを開発してその有効性を検証した。

Construction of Solid Models from Orthographic Views by Solid Element Equations

MASAJI TANAKA†

Much study has been conducted on reconstructing solid models from orthographic views drawn by a 2-D CAD system. However, previously proposed algorithms require a large amount of computational time when the solid model becomes complex. For a real application of the reconstruction of the solid model, efforts still need to be put on developing an algorithm that requires less computational time. This paper describes an algorithm that requires computational time linear to the complexity of a solid. First, the algorithm computes a surface model as in Wesley and Markowsky from orthographic views, computes solid elements out of the surfaces, and then establish a set of equations that represent true and false relationships among the solid elements by comparing the solid elements with the orthographic views. Since the algorithm solves the set of the equations, it does not employ a mechanism of searching suitable combinations of surfaces or solid elements. Therefore the algorithm runs faster than those which employ search mechanisms. The computational time of this algorithm is linear to the complexity of a solid.

1. はじめに

CAD で描かれた三面図から自動的に3次元の立体を作成する方法は、これまで多く研究されてきた³⁾。その主要な方法は、三面図より立体の稜線となるワイヤフレームモデル、立体の表面となるサーフェスモデルを順次作成する方法である。この方法は初期に出沢より始められたが、ここではワイヤフレームモデルやサーフェスモデルにおいて、実際には存在しない稜線や面が発生することが明らかにされており、このような虚図形の除去方法が課題とされた¹⁾。

この流れに属する主な研究として、佐々木らの疑似ブール代数を用いる方法²⁾、西田らの人間の知識を導

入する方法³⁾、千田や増田らの組合せ探索の効率を計る方法^{7),8)}などがある。

一方、Wesley らは、サーフェスモデルより面を境界とする3次元の閉領域を立体のブロックとして作成し、これらの組合せ探索による方法を提案している²⁾。同様の方法はGujar も行っている⁴⁾。

本論文で紹介する手法は、基本的には稜線や面のように立体の構成要素となるような、立体形状の構成要素を作成し、これらの真偽（実際に存在するものを真、存在しないものを偽とする）の関係を、三面図との直接的な対応関係より定式化する方法であり、この立体形状の構成要素として先述の立体のブロックを採用している。なお、本研究では立体のブロックを、立体形状の構成要素という意味で立体要素とする。

先に挙げた研究はすべて、解は得られるものの、実際の処理時間が立体形状の複雑さに対して急激に増大

† 財団法人 京都高度技術研究所
Advanced Software Technology & Mechatronics
Research Institute of KYOTO

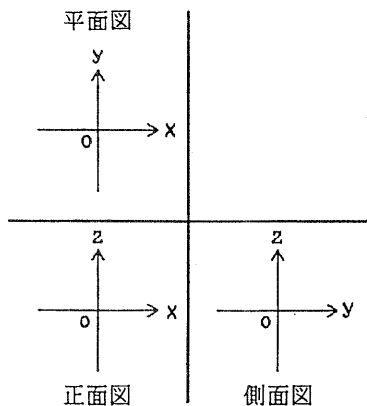


図 1 三面図の座標系

Fig. 1 Coordinate system of orthographic views.

する傾向があり、機械部品などの設計現場での実用化には問題がある。

三面図からの自動立体作成は、基本的には面や立体要素など図形要素の組合せ探索の問題であるが、本手法では、この問題を立体要素連立式の解法に変換させることで、実際の処理量を立体形状の複雑さと比例させる程度に押さえた高速な処理を可能とした。また、実際にプログラムを開発してその有効性を検証した。

本手法の適用対象となる立体は、現段階では多面体に限定し、非多様体は除外する。本研究での三面図の座標系を図1に示す。本文中において、正面方向、平面方向、側面方向とは、おのこの y の正方向、 z の負方向、 x の負方向を示す。以下、2章では、本手法のもととなるアイデアを示し、3章において本手法を述べる。

2. 直方体要素連立式による解法

2.1 直方体要素の作成

図2のように、 x 、 y 、 z いずれかの座標軸と平行となる線分のみで構成される三面図の場合、描かれた線を延長すると、図3のように外形が長方形となる格子状の面が作成できる。これを格子面とする。格子面の個々の格子の目の三面間の対応関係を考えると、すべての対応関係より3次元の直方体の集合ができる。図4にそのワイヤフレームモデルを示す。

x 、 y 、 z いずれかの座標軸と平行となる線分のみで構成される三面図の場合、解である立体の表面は、稜線を境界として x - y 平面、 y - z 平面、 x - z 平面のいずれかと平行となるので、解はこれらの直方体より構成できる。

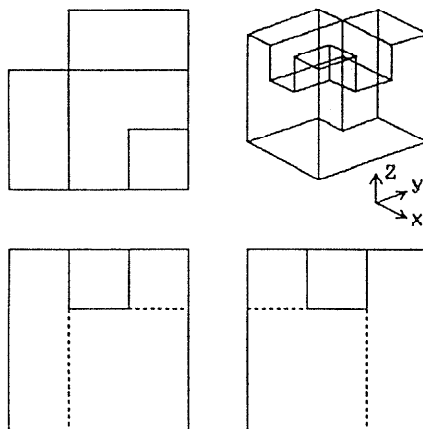


図 2 例題1と解

Fig. 2 Example 1 and solution.

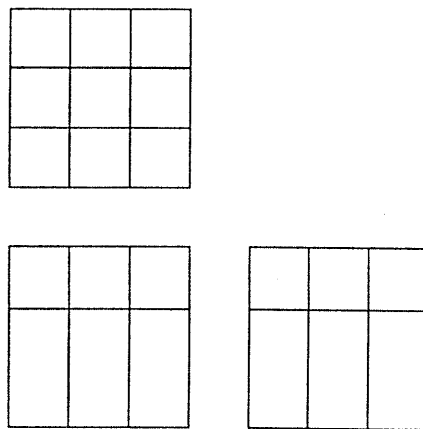


図 3 格子面

Fig. 3 Lattice faces.

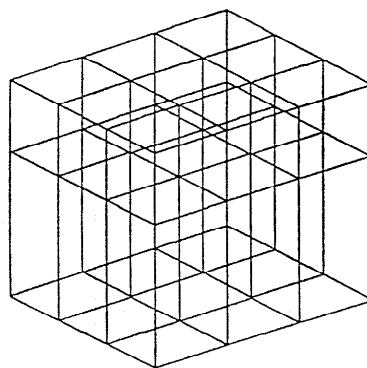


図 4 直方体の集合のワイヤフレームモデル

Fig. 4 Wire frame model of cuboid elements.

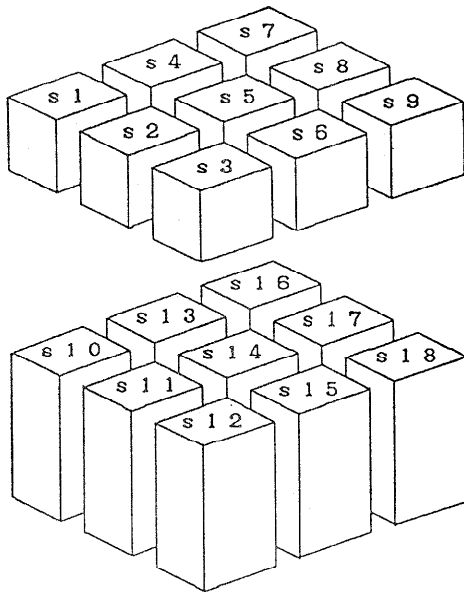


図 5 直方体要素
Fig. 5 Cuboid elements.

個々の直方体を解を構成する直方体要素として、図 5 のように番号を付ける。直方体要素の集合には、図 5 より明らかなように、解を構成しない偽の直方体要素が含まれるので、解は各直方体要素の真偽を判別することで求める。次にその判別方法を述べる。

2.2 直方体要素間の真偽の関係

直方体要素の集合において、格子面上に現れる直方体要素を図 6 に示す。直方体要素の集合は、その外形が直方体となる 3次元の規則的な配列であるから、この性質を利用して図 6 と三面図とを比較すると、直方

s7	s8	s9
s4	s5	s6
s1	s2	s3

s1	s2	s3
s10	s11	s12

s3	s6	s9
s12	s15	s18

図 6 格子面上の直方体要素
Fig. 6 Cuboid elements on the lattice faces.

体要素間の真偽の関係が次のように明らかになる。

○実線条件

例えば、正面図において s1 と s2 の間には三面図では実線がある。したがって、s1 と s2 は共に真ではない。この関係を次式で表す。

$$s1 \times s2.$$

○仮線条件

例えば、正面図において s11 と s12 の間には三面図では線がない。したがって、s11 と s12 の真偽は同じである。この関係を次式で表す。

$$s11 - s12.$$

正面方向に見て、s11 と s12 の背後に位置する s14 と s15 間および s17 と s18 間にも、同様に s14-s15, s17-s18 が成り立つ。

○破線条件

例えば、正面図において s10 と s11 の間には三面図では破線がある。したがって s10-s11 が成り立つ。破線が存在するためには、正面方向に見て s10 と s11 の背後に位置する s13 と s14 または s16 と s17 の対において、一方が真で他方が偽でなければならない。

○存在条件

例えば、正面図において s1 の輪郭を表す線の中で、左側の縦の線分は三面図では実線である。したがって s1 と、正面方向に見て s1 の背後に位置する s4, s7 の中には真が存在する。逆に、例えば平面図において s7 の輪郭を表す線の中で、左側の縦の線分は三面図では線がない。したがって s7 と、平面方向に見て s7 の背後に位置する s16 は偽である。

2.3 直方体要素連立式による解の探索

図 6 に現れた直方体要素間の真偽の関係を、各条件で示した式で表すと図 7 (a) のようになる。直方体要素は 3次元の立体なので、図 7 (a) において s1, s2, s3, s6, s9, s12 は三面中二面以上に現れている。これらの直方体要素より、図 7 (a) は図 7 (b) のように一つにまとめられる。

図 7 (b) に他の仮線条件: s10-s13, s11-s14, s17-s18 を加え、各条件の定義に従って式を簡略化すると図 7 (c) のようにできる。これを直方体要素連立式とする。

直方体要素連立式は、定式化された個々の直方体要素間の真偽の関係を一つの連立式として表したものであり、本手法のアイデアの中核となるものである。直方体要素連立式より、真偽の同じ要素からなる二つの集合: $A = \{s1-s3-\dots-s18\}$, $B = \{s2-s5-s6, s7\}$ が

$$\begin{array}{ccc}
 s7 \times s8 - s9 & & \\
 \times & \times & \times \\
 s4 \times s5 - s6 & & \\
 | & | & \times \\
 s1 \times s2 \times s3 & &
 \end{array}$$

$$\begin{array}{ccc}
 s1 \times s2 \times s3 & & s3 \times s6 \times s9 \\
 | & \times & | & \times & | \\
 s10 - s11 - s12 & & s12 - s15 - s18
 \end{array}$$

(a) 三面各々の直方体要素の連立式
(a) Equations of cuboid elements in 3 views.

$$\begin{array}{cccc}
 s7 \times s8 - s9 - s18 & & & \\
 \times & \times & \times & | \\
 s4 \times s5 - s6 \times s15 & & & \\
 | & | & \times & | \\
 s1 \times s2 \times s3 & & & | \\
 | & \times & & | \\
 s10 - s11 - s12 & & & s12
 \end{array}$$

(b) 一つにまとめられた連立式
(b) Combined equations.

$$\begin{array}{c}
 s2 - s5 - s6 \\
 \times \\
 s1 - s3 - s4 - s8 \\
 | \\
 s12 - s11 - s10 - s9 \\
 | \\
 s13 - s14 - s15 - s17 \\
 | \\
 s7 \times s18
 \end{array}$$

(c) 直方体要素連立式
(c) Cuboid element equations.

図7 直方体要素連立式
Fig. 7 Cuboid element equations.

できることがわかる。

このように直方体要素連立式によって、個々の直方体要素を真偽の同じいくつかの集合にまとめることができる。したがって、解の探索はこれらの集合の真偽の組合せ探索となり、個々の直方体要素の真偽の組合せ探索から比べると、はるかに効率良く処理することができる。

解を得るには、例えば、先に示した存在条件の例より $s7, s16$ は偽となり、 $s1, s4, s7$ の中には真が存在するから、集合 A は真、集合 B は偽となる。破線条件はこの段階ですべて満たされるので、解は集合 A となる。

2.4 立体要素へのアイデアの適用

これまで述べた方法の適用は、図2のように三面図が x, y, z いずれかの座標軸と平行となる線分のみで構成されることを前提とする。そこで、この方法をあらゆる三面図にも適用できるように、直方体要素を

立体要素に置き換えて、立体要素間にアイデアである真偽の関係式および連立式を適用させたのが本手法である。

3. 立体要素連立式による解法

3.1 立体要素の作成

図8を例題として、始めに立体要素の作成方法の概要を示す。

(1) 2D 頂点と 2D 線分

三面図に描かれた直線の交点を 2D 頂点 ($fv1, fv2, \dots$) とし、2D 頂点を両端とする線分を 2D 線分 ($fe1, fe2, \dots$) とする (図9)。

(2) ワイヤフレームモデル

三面間の 2D 頂点の位置関係よりできる 3次元の点を頂点 ($v1, v2, \dots$) とする (図10)。

例えば、 $v5$ と $v6$ を両端とする 3次元の直線 ($e7$) は、正面図では $fe1$ と $fe2$ で表され、平面図では $te1$ と $te2$ で表され、側面図では $sv3$ で表される。したがって $e7$ は立体の稜線の候補となる。

このように各頂点を両端とする直線の中で、立体の稜線の候補となるものを稜線 ($e1, e2, \dots$) として作成し、ワイヤフレームモデルができる (図10)。ワイヤフレームモデルには、図10より明らかのように実際には存在しない偽の稜線が含まれている。

(3) サーフェスモデル

稜線を境界とする閉じた平面領域を面 ($f1, f2, \dots$)

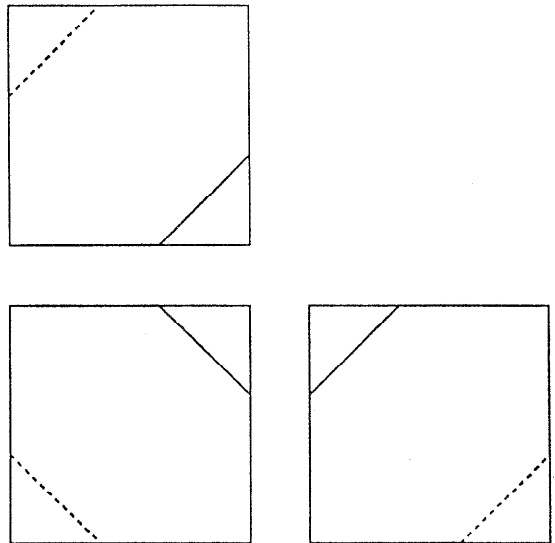


図8 例題2
Fig. 8 Example 2.

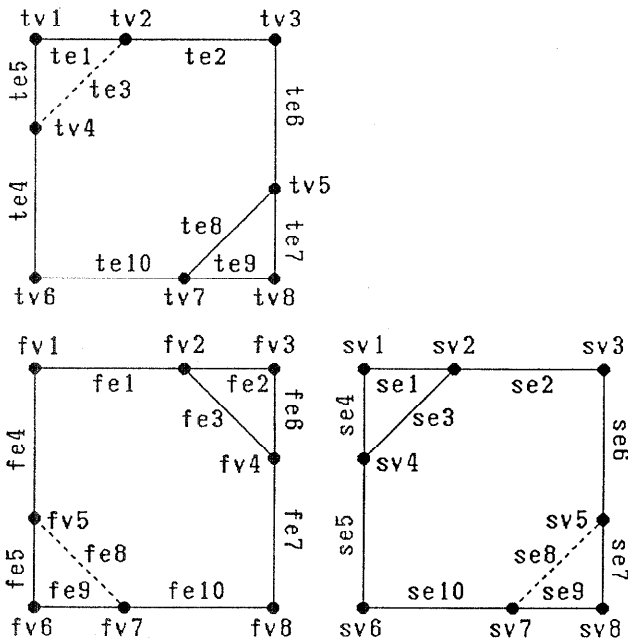


図9 2D 頂点と2D 線分
Fig. 9 2D vertices and 2D lines.

として作成し、サーフェスモデルができる (図 11). サルフェスモデルには、図 11 より明らかなように実際には存在しない偽の面が含まれている.

(4) 立体要素

面を境界とする閉じた立体領域を立体要素 (s1, s2, s3) として作成する (図 12). 立体要素には、図 12 より明らかなように実際には存在しない偽の立体要素が含まれている.

3.2 立体要素間の真偽の関係

図 9 において、2D 線分を境界とする閉領域を 2D 面 (ff1, ff2, ...) として作成する (図 13). 2D 面は、2 章で述べた格子面の格子の目に対応するものであり、2D 面間の境界となる 2D 線分をもとに、2 章と同様の実線条件、破線条件、存在条件を定義する.

ただし、立体要素の集合は直方体要素の集合のような規則的な配列ではないので、立体要素の表面の法線方向などを調べて、真偽の関係を明らかにする必要がある.

○実線条件

実線 fe3 を例として、図 14 より説明する. 正面方向に見て ff1 と ff2 に対応する面は、おのおの f4, f5, f8 と f2, f8 である. これらを構成要素とする立体要素はおのおの s1, s2 と s2 であり、これらが fe3 を形成しうる立体要素となる.

もし、s1 と s2 が共に真ならば、ff1 と ff2 にはおのおの f5 と f2 が対応する. ところが、f5 と f2 は同一平面上にあるので、fe3 に対応する稜線が存在しなくなる. したがって s1 と s2 は共に真ではない. この関係を次式で表す.

$$s1 \times s2.$$

もし、s1 が偽で s2 が真ならば、正面方向に見て s1 の背後に位置する s2 の f4 が ff1 に対応する. f4 と f2 は同一平面上にないので、fe3 に対応する稜線 (e9) ができる.

もし、s2 が偽ならば、ff2 に対応する面が存在しなくなるので、s2 は真であることがわかる.

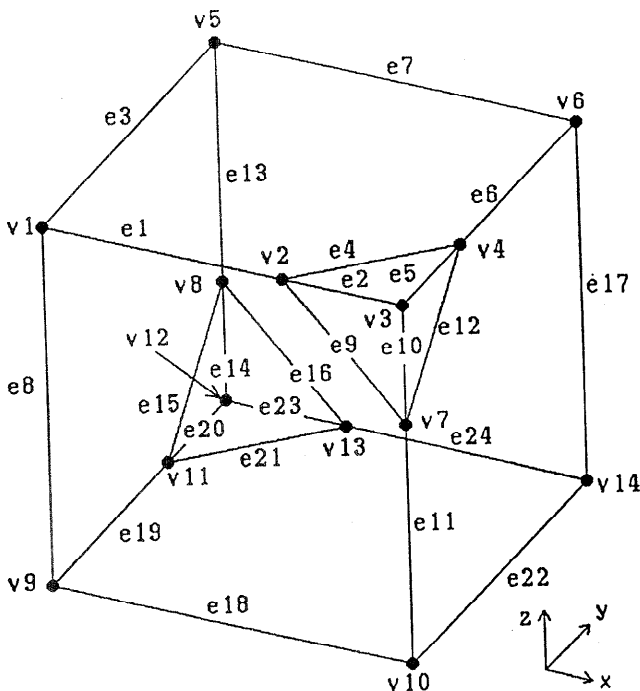


図 10 ワイヤフレームモデル
Fig. 10 Wire frame model.

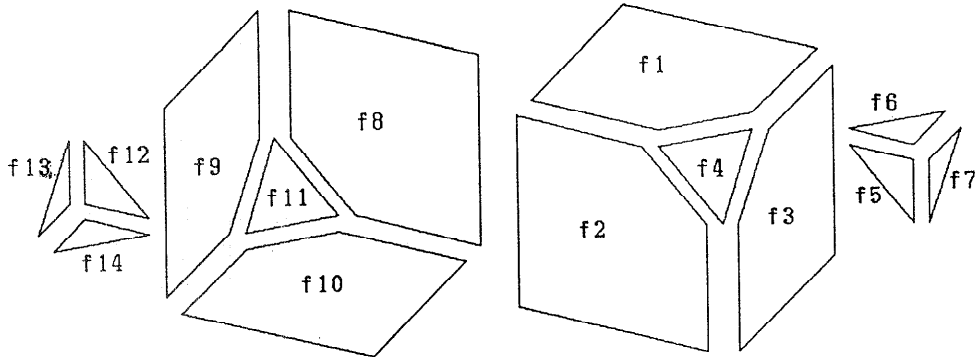


図 11 サーフェスモデル
Fig. 11 Surface model.

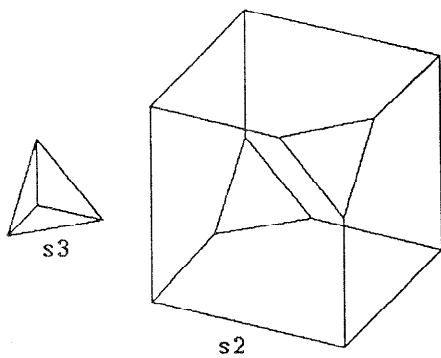


図 12 立体要素
Fig. 12 Solid elements.

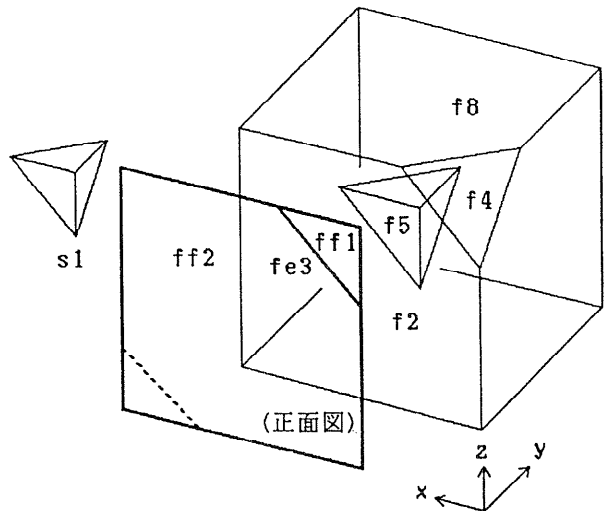


図 14 実線 fe3 に関する立体要素
Fig. 14 Solid elements relate to fe3.

ここで、仮に fe3 を破線とすれば、s1 と s2 の真偽は同じとなる。この関係を次式で表す。

$$(s1-s2.)$$

○破線条件

破線 fe8 を例として、図 15 より説明する。正面方向に見て ff2 と ff3 に対応する面は、おのおの f2, f8 と f2, f11, f12 である。これらを構成要素とする立体要素はおのおの s2 と s2, s3 であり、これらが fe8 を形成する立体要素となる。

もし、s2 が真ならば、ff2 と ff3 には共に f2 が対応するので、s2 は fe8 に対応する稜線を覆う立体要素となりうる。もし、s2 と s3 が共に真ならば、fe8 に対応する稜線 (e16) が存在しなくなるので、fe8 が存在するためには、s2 と s3 において一方が真で他方が偽でなければならない。もし、s2 が偽ならば、

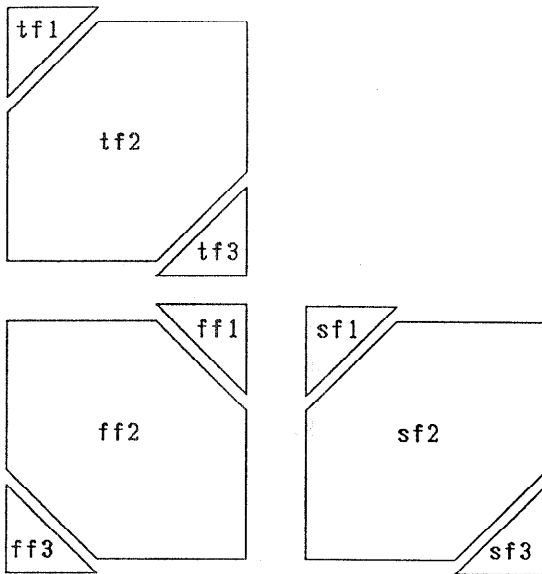


図 13 2D 面
Fig. 13 2D faces.

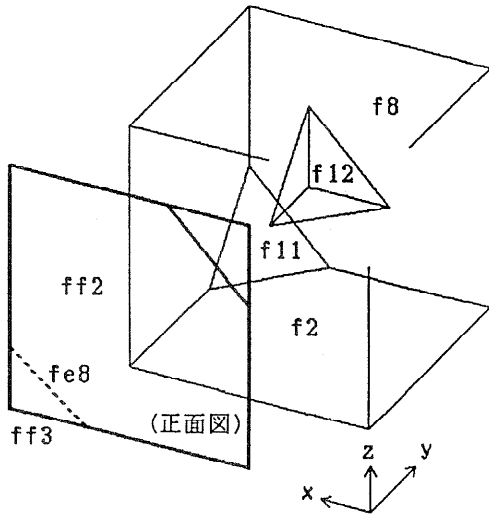


図 15 破線 fe8 に関係する立体要素
Fig. 15 Solid elements relate to fe8.

f2 に対応する立体要素が存在しなくなるので, s2 は真, s3 は偽となることがわかる.

○存在条件

例えば, fe2 をもとに作られた稜線は e2, e4, e7 であり, これらを構成要素とする立体要素は s1, s2 である. したがって, s1, s2 の中には真が存在する.

3.3 立体要素連立式による解の探索

2D 面間の境界となるすべての 2D 線分より得られる立体要素間の真偽の関係は, 三面共に $s1 \times s2$ となる. これは 2 章の直方体要素連立式に対応するものであり, ここでは立体要素連立式とする.

先に示した実線条件の例より s2 は真, s1 は偽となり, 破線条件の例より s3 は偽となるので, 解は s2 となる.

3.4 例題

立体要素連立式の効果を示す例題として図16(a)を解く. 図16(b)にワイヤフレームモデル, 図16(c)に立体要素, 図16(d)に三面図上に現れる立体要素を示す. 立体要素の集合は 3 次元の規則的な配列となるので, この配列と図16(d)を比較すると, 立体要素間の真偽の関係は図16(e)のようになることが容易にわかる. 図16(e)は図16(f)のように簡略化され, これが立体要素連立式となる.

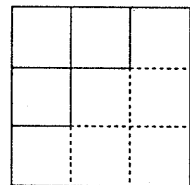
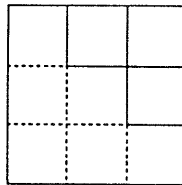
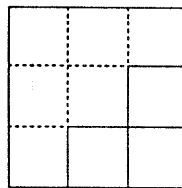
解を得るには, 例えば, 図16(d)の正面図において, s1 の輪郭を表す線の中の左側の縦の線分の存在条件より s1, s4, s7 の中には真が存在するから, s1-s4-...-s27 は真, s2, s6, s12 は偽となる.

s3 は真でも偽でもよいが, s3 は稜線のみで他の真の立体要素と接するので本手法では偽とする.

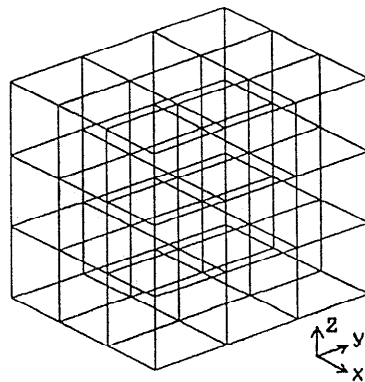
立体要素連立式に現れない立体要素は s13, s14, s16, s17, s22, s23, s25, s26 であり, これらの真偽の組合せよりすべての破線条件を満たすようにする. これらの真偽を判別する要因は各条件からは得られないので, 個々の真偽を組合せて各破線条件と照合させる. 全組合せ数は 2^8 通りとなり, 12 通りの解が得られる. 図16(g)はその一例である.

4. 実際の処理例

本手法の有効性を検証するために, 実際にプログラムを開発 (PC 9801 ES 上に C 言語で実装) して種々の例題を試みた. 図17(a)~(f)にその一部の三面図

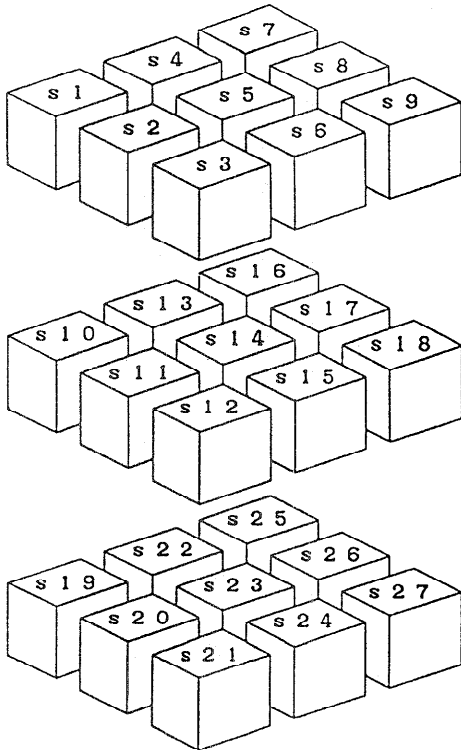


(a) 三面図
(a) Orthographic views.



(b) ワイヤフレームモデル
(b) Wire frame model.

図 16 例題 3
Fig. 16 Example 3.



(c) 立体要素
(c) Solid elements.

$$\begin{array}{ccc}
 s7 - s8 - s9 & & \\
 | & | & \times \\
 s4 - s5 \times s6 & & \\
 | & \times & \times \\
 s1 \times s2 \times s3 & &
 \end{array}$$

$$\begin{array}{ccc}
 s1 \times s2 \times s3 & s3 \times s6 \times s9 & \\
 | & \times & \times & \times & \times & | \\
 s10 - s11 \times s12 & s12 \times s15 - s18 & & & & \\
 | & | & \times & \times & | & | \\
 s19 - s20 - s21 & s21 - s24 - s27 & & & &
 \end{array}$$

(e) 三面のおおのの立体要素の連立式
(e) Equations of solid elements in 3 views.

$$\begin{array}{ccc}
 & & s3 & & & \\
 & \times & \times & \times & & \\
 & s2 & s6 & s12 & & \\
 & \times & \times & \times & & \\
 s1 - s4 - s5 - s7 - s8 & & & & & \\
 & & & & | & \\
 s18 - s15 - s11 - s10 - s9 & & & & & \\
 | & & & & & \\
 s19 - s20 - s21 - s24 - s27 & & & & &
 \end{array}$$

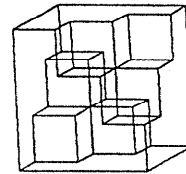
(f) 立体要素連立式
(f) Solid element equations.

s7	s8	s9
s4	s5	s6
s1	s2	s3

s1	s2	s3
s10	s11	s12
s19	s20	s21

s3	s6	s9
s12	s15	s18
s21	s24	s27

(d) 三面図上の立体要素
(d) Solid elements on the orthographic views.



(g) 解の一例
(g) An example of solutions.

図 16 (つづき)
Fig. 16 (Continued)

と解を示し、処理時間について表1にまとめる。
表1において、立体要素作成時間は立体表面の凹凸の多さにほぼ比例し、連立式作成時間は三面図上に表しにくい形状（三面図において他の稜線に重なる稜線

や破線となる稜線など）の多さにほぼ比例すると考えられる。
立体形状の複雑さがこの二つの要因に代表されるとすれば、解の探索時間はこの二つの時間に対して極力押さえることができるので、全体として処理時間は立体形状の複雑さにほぼ比例するとみなせる。

表 1 例題の処理時間
Table 1 Processing time of examples.

例題 No.	立体要素作成時間	連立式作成時間	解の探索時間	合計時間
3	11.5	26.5	38.0	76.0
4	4.0	5.0	瞬時	9.0
5	7.0	4.0	瞬時	11.0
6	32.5	17.5	瞬時	50.0
7	6.5	25.0	瞬時	31.5
8	16.0	57.0	13.5	86.5
9	12.0	25.0	1.0	38.0

(単位はすべて sec)

5. 考 察

本手法では、三面図に描かれた実線に矛盾する偽の立体要素は、立体要素連立式より容易に判別できる

が、破線に矛盾する偽の立体要素は、図 16 の例題のように各条件だけでは判別できない場合がある。このような偽の立体要素は、形状のほとんどが三面図において破線で示されるような立体要素が、多く固まった場合に発生しやすいと考えられる。

したがって、実線よりも破線が多いような三面図では、図 16 の例題のように単純な組合せ探索を行う分だけ、処理量が増大する可能性がある。ところが、実際の機械部品などの三面図では、実線よりも破線が多くなるような立体では断面図を描くなど、設計者は理解しやすさを優先すると考えられるので、このような三面図は例外とみなせる。

本手法は、外観としては三面図と合致するまで偽の立体要素を取り除いていく方法なので、合致した時点

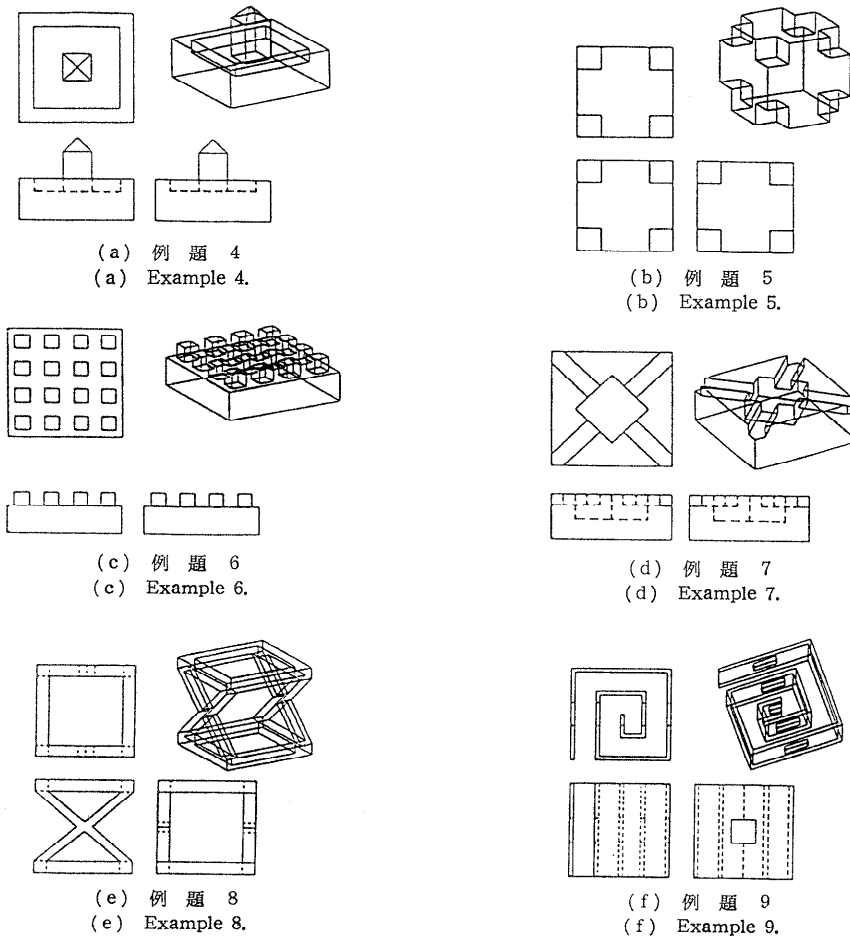


図 17 例題と解
Fig. 17 Example and solutions.

で真偽の判別ができていない立体要素はすべて真としている。したがって、完全解を得ようとする場合は、これら真偽の判別ができていない立体要素について、さらに真偽の判別を進めていくことになる。

ところが、実際には設計者はなるべく三面図と立体が1対1となるように描くと考えられ、また、識別が困難な除去すべき部分がある場合は、副面図を描くなど配慮されると考えられるので、実用上完全解を求める必要はないと考えた。また、もし解に設計者の意図しない部分がある場合は、立体要素の編集により容易に修正を加えることが可能である。

6. 他の研究との比較

三面図より立体を作成する方法は、基本的には稜線や面、立体要素など図形要素の組合せを三面図と合致させるような探索的な方法となる。したがって完全解を求めることは、どのような方法であれ、原理的には全探索を行うことになる。

ところが、1章で挙げた研究はすべて立体復元の完全な理解の構築が中心であり、完全解を必要条件とするので、立体形状の複雑さに対して実際の処理量が急激に増大する傾向がある。

本研究では実用化を目的として、2章で示した効率良く解を求められるアイデアを考案し、これを立体要素において具体化することで、完全解は求めなくても設計者の要求に沿う解を、立体形状の複雑さと比例させる程度に押さえた処理量で求めることができる手法を開発した。

手法の違いとしては、立体要素による方法は Wesley らや Gujar らが行っているが^{2),4)}、これらは立体要素の組合せ探索による方法であり、定式化は行われていない。

図形要素の定式化より連立式を立てて解くという点では、佐々木らも行っているが⁵⁾、ここでは稜線と面を式の構成要素としている。サーフェスモデルより解を求める方法は他にもいくつか行っているが⁶⁾⁻⁸⁾、立体要素を用いる利点として次の点が挙げられる。

- サーフェスモデルより解を求める場合、三面図と合致させる条件だけでなく、「一つの稜線は必ず二つの面に接する」といった立体を形成するための条件が必要となるため、それだけ処理が複雑となるが、立体要素はそれ自体がソリッドモデルとして形成されるので、先述のような立体形成の条件が不必要となる。

- 面より作成された立体では、後でソリッドモデルへの変換が必要となるが、立体要素からは、これらを貼り合わせる際の不要な面や稜線を除去するだけでソリッドモデルができる。

- 面の数に比べると立体要素の数はかなり少なくできるので、探索的処理を行う上ではるかに効率が良くなる。

これらの点より、立体要素連立式による本手法は、サーフェスモデルより解を求める方法よりも、かなり有効であると考えられる。さらに今後研究を進めて定量的に有効性を確かめたい。

7. おわりに

本研究では、立体要素間の真偽の関係を定式化し、立体要素連立式として解く手法を確立した。曲面を含む立体に対しても、立体要素が作成できれば容易に適用できると推定される。

謝辞 研究を進める上で数々の助言をいただいた京都高度技術研究所・岩間憲三研究室長、および研究をまとめる上でご指導いただいた京都大学・沖野教郎教授に心より感謝いたします。

参考文献

- 1) 出沢：三面図から立体形成のためのシステム，日本機械学会論文誌，Vol. 38, No. 310, pp. 1267-1276 (1972)。
- 2) Wesley, M. A. and Markowsky, G.: Fleshing Out Projections, *IBM J. Res. Dev.*, Vol. 25, No. 6, pp. 934-954 (1981)。
- 3) Nagendra, I. V. and Gujar, U. G.: 3-D Objects from 2-D Orthographic Views—A Survey, *Comp. & Graphic.*, Vol. 12, No. 1, pp. 111-114 (1988)。
- 4) Gujar, U. G. and Nagendra, I. V.: Construction of 3D Solid Objects from Orthographic Views, *Comp. & Graphic.*, Vol. 13, No. 4, pp. 505-520 (1989)。
- 5) 佐々木, 伊藤, 鈴木: 非線形疑似ブール代数解法による三面図からの物体の自動合成, 情報処理学会論文誌, Vol. 30, No. 6, pp. 699-708 (1989)。
- 6) 西田, 張, 西原: 面の組合せ探索による三面図の解釈, 人工知能学会誌, Vol. 6, No. 1, pp. 96-104 (1991)。
- 7) 千田: 三面図からもとの立体の自動復元, 情報処理学会論文誌, Vol. 31, No. 9, pp. 1312-1320 (1990)。
- 8) 増田, 梶家, 清水, 我妻, 山口: 三面図からの自動立体復元, 1991年度精密工学会春季大会講演論文集, pp. 745-746 (1991)。

(平成5年2月3日受付)

(平成5年6月17日採録)

**田中 雅次 (正会員)**

1963年生。1986年3月佐賀大学
理工学部機械工学科卒業。同年4月
(株)リコー入社。1990年4月(財)京
都高度技術研究所(ASTEM)入所。
現在に至る。CAD/CAM を中心と
した生産技術の研究に従事。日本ロボット学会会員。
