

三面図からの立体モデル自動復元システムの構築

柴宮 理, 富山 健, 中庭 憲一, 横田 貴之, 肥塚 隆*, 森内 直樹*

青山学院大学理工学部

*新日本製鐵エレクトロニクス研究所

三面図から三次元ソリッドモデル(以降、立体モデル)を自動復元するシステムについて述べる。立体モデル復元の手法としてはCSG法を用いたが、その基本的な考え方は「基本立体(プリミティブ)の集合演算を行うことで立体モデルを復元する」ことである。基本立体は各面図(必要であれば補助線を引くことによって分割した図)に現れている図形と基本立体の投影図が一致しているものを選択する。この時点で基本立体の大きさ、位置、回転、集合演算のための演算子を決定する。選択された基本立体は木構造を基に集合演算を行う。この木構造は基本立体の大きさ、包含関係を基に作成する。このシステムの実験を行った結果、様々なサンプルを復元することができた。一方いくつかの問題点を確認することもできたが、それらについての対処法も提案する。

Automatic Reconstruction of 3D Solid Model from Three Orthographic Views

Tohru SHIBAMIYA, Ken TOMIYAMA, Ken-ichi NAKANIWA, Takayuki YOKOTA,
Takashi KOEZUKA*, Naoki MORIUCHI*

College of Science and Engineering, Aoyama Gakuin University

*Electronics Research Laboratories, Nippon Steel Corporation

This paper discusses an attempt to develop a system that automatically reconstructs a three dimensional (3D) solid model from three orthographic views. The CSG method was adopted for 3D reconstruction where the basic idea is the following: a 3D solid model is constructed by geometric combinations of simple 3D solids, called primitives. Appropriate primitives are chosen by comparing their projection with the original three views that are augmented and segmented by a rule proposed here. The chosen primitives are combined according to a structure tree. It represents primitives' size and inclusion relationships and is constructed from the original views. The proposed system was successfully tested using several simple but practical examples. With some examples, however, some difficulties were observed to which we propose alternate approaches.

1 はじめに

三面図は、三次元の形状を二次元で表現する有効な手段である。そのため、生産現場などで広く利用されている。しかし、図面を見慣れない者にとって、三面図による二次元表示から、三次元の物体をイメージすることは難しいという欠点がある。このため、図面を見慣れている者と見慣れていない者の間でしばしば意志疎通の欠如から問題が起こる。そこで図面から立体に復元することをコンピュータで自動的に行わせることができれば、図面を見慣れないより多くの人が図面に描かれている立体を理解する助けになる。そのため三面図に描かれている二次元情報を、我々人間の視点からみた三次元立体として表示させるシステムが求められている。

三面図あるいは機械図面から三次元立体を復元する手法に関する研究は、初期は出沢[1]によって行われ、現在までに様々な研究者により行われている。ここ数年でも、[2][3][4][5][6][7][8][9][10][11]等の研究成果が報告されている。

ここでは我々の行っている、三面図から自動的に三次元立体モデルを復元するシステムについて説明する。

2 立体モデル自動復元システム

本システムの立体復元の手法としてはCSG法を用いたが、その基本的な考え方は、「基本立体（プリミティブ）の集合演算（和・差）を行うことで立体モデルを復元する」と「中間部品（プリミティブを部分的に集合演算したもの）を先に生成し、その部品を組み合わせて目的の立体モデルを得る」である。この考えを用いて、以下の方法で立体モデルを復元する。

a) システム内に持っているプリミティブのデータ

（プリミティブを三面図に投影した図形：シェイプと呼ぶ）を参照し、三面図（面図により必要があれば補助線を引いて分割した図）の外形線、隠れ線、補助線の情報から、立体モデルを構成するプリミティブ（構成プリミティブ）を生成し、大きさ、位置、回転、集合演算のための演算子等を決定する。

b) 構成プリミティブの大きさ、包含関係を基に構成プリミティブの木構造を生成し、この木構造を基に集合演算を行う。この木構造は幹の部分と枝の部分から成り、先に枝の部分の集合演算を行い、最終的に幹の部分の集合演算を行う。

本システムで用いたプリミティブは直角三角柱、二等辺三角柱、直方体、台形柱、円柱、円錐、正六角柱、直角三角錐、三角柱である。またシェイプはこのペーパーの最後にある付図1に示す。

処理の流れを図1に示し、次章で各処理について説明する。

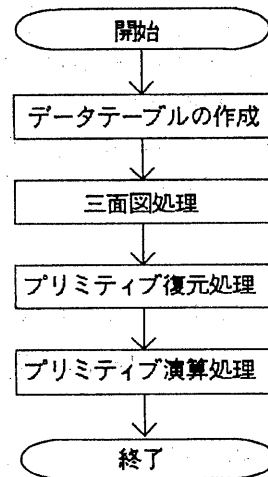


図1 処理の流れ

3 各処理の概要

3.1 データテーブルの作成

本システムに必要なプリミティブデータと線分データの各データテーブルを作成する。プリミティブデータはシェイプが3個1組で1個のプリミティブを表している（表1）。線分データは三面図から得られる外形線、隠れ線のデータである。

3.2 三面図処理

ここでは、構成プリミティブを復元するのに必要な二次元図形（閉ループ：閉じた線分列）を各面図に対して探し、得られた閉ループに対して、プリミティブの演算子決定に必要な閉ループ属性を与える。以下に示す処理を各面図に対して行う。

表1 プリミティブのデータ

プリミティブ名	シェイプデータ*
直角三角柱	7-1-1
二等辺三角柱	8-1-1
立方体	1-1-1
台形柱	9-1-1
円柱	10-1-1
円錐	10-8-8
正六角柱	17-1-1
直角三角錐	7-7-7 7-7-1 8
三角柱	18-1-1

*表中の数字はプリミティブシェイプ番号

a) 孤立部分分割

個々の面図において孤立部分(図2)を分類し、各々を別なグループに属する線分とする。特に一番外側のグループを親グループとし、その他のグループを子グループ1, 2, ...とする。

このグループは孤立部分内で探索された閉ループにも受け継がれる。これは閉ループの属性決定時に参照するほか、構成プリミティブの木構造で幹や枝の部分にプリミティブを分類するのに利用する。

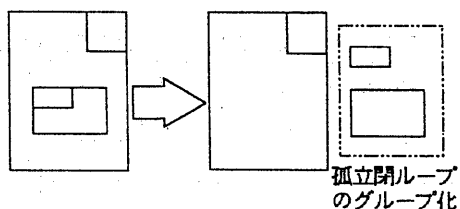


図2 孤立部分分割

ブの組合わせである組合わせ閉ループを探索、形状判別を行う。判別できる形状はb)と同じ。

形状判別できない単純閉ループに対しては、形状判別が可能になるまで外接する最小の長方形を被せる形で補助線を設定し、閉ループを探索する。この補助線設定の回数を階層的に表す変数として、depthを導入する(図3)。depth導入の理由は、補助線を設定した回数を閉ループ属性に反映させるため、構成プリミティブの演算を行う時に利用するためである。

また補助線設定により生成された閉ループはまとめて子グループ化し、a)の子グループと同等に扱う(図3)。これはa)の理由と同じである。ただしグループ化するのは、対象の単純閉ループに初めて補助線を設定する時だけであり、これ以降補助線設定(形状判別可能になるまで)を行っても別なグループになる事は無い。

なお組合わせ閉ループは形状判別できるものだけを扱う。

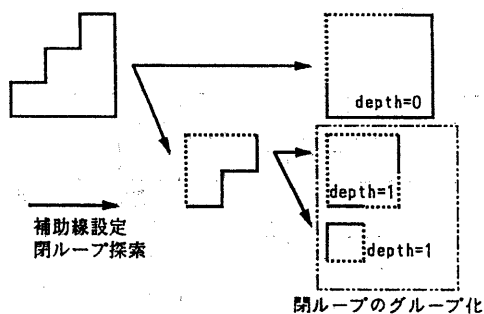


図3 補助線設定

b) 外周閉ループ探索

各孤立部分で一番外側の閉ループ(外周閉ループ)を探索し、形状を判別する。判別できる形状は付図1のシェイプである。形状判別できない時は、閉ループに外接する最小の長方形を被せる形で補助線を設定する(図3)。

c) 閉ループ探索

各孤立部分で、単純閉ループ(閉ループを分割する線分がないもの)を探索し、次に単純閉ル

d) 閉ループ属性決定

補助線や外周閉ループの情報から閉ループ属性決定する。この属性を基に構成プリミティブの集合演算の演算子を決める。属性の決定方法は以下のように3つに分類することができる。

i) 親グループに属する閉ループについて

- ・外周閉ループ：+
- ・補助線を一辺に持つ単純閉ループ：-
- ・補助線を一部に持つ組合わせ閉ループ：-

- ・補助線を一辺に持つ単純閉ループと隣合う単純・組合せ閉ループ：+ (図4)
- ・補助線を一辺に持つ単純閉ループと隣合わない単純・組合せ閉ループ：0 (図4)

ii) 子グループに属する閉ループについて

depth = 0 の場合

- ・外周閉ループ：⊕
- ・補助線を一辺に持つ単純閉ループ：⊖
- ・補助線を一部に持つ組合せ閉ループ：⊖
- ・補助線を一辺に持つ単純閉ループと隣合う単純・組合せ閉ループ：⊕ (図4)
- ・補助線を一辺に持つ単純閉ループと隣合わない単純・組合せ閉ループ：0 (図4)

depth ≠ 0 の場合 (補助線設定が行われた)

- ・外周閉ループ：+
- ・外周閉ループ以外：-

この⊕⊖は、子グループに属する閉ループは一面図だけでは属性を決定できないため、一時的な属性を付与することを示している。後で三面図間で閉ループの対応を取った時に正式な属性を付与する。

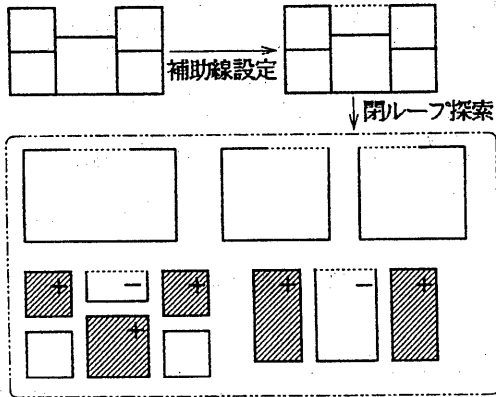


図4 補助線を一辺に持つ単純閉ループと隣合う閉ループの例

3.3 構成プリミティブ復元処理

次に3.2の処理で得られた閉ループを使用して、構成プリミティブを復元する。

a) 子グループに属する閉ループの属性決定

子グループに属する外周閉ループの属性決定を、三面図間で対応の取れる(閉ループの端点座標が隣合う面図間で一致する: 図5) 他の二面図の閉ループ属性から決定する(表2)。同じ子グループ内の他の閉ループ属性は、外周ループ属性が決定した時点で元の⊕⊖に沿って決定する。つまり外周閉ループが"+"なら"⊕"⇒"+", "⊖"⇒"-", 外周閉ループが"-なら逆になる。

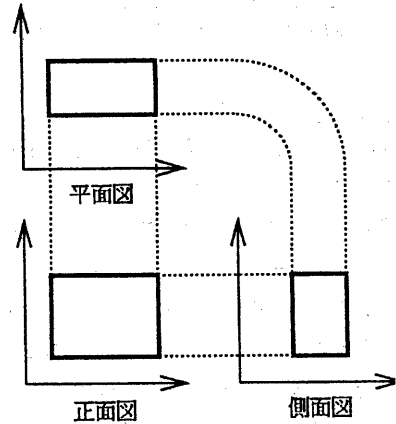


図5 三面図対応

表2 子グループに属する閉ループの演算子決定表

対応の取れる二面図の閉ループ属性	子グループの外周閉ループ
+ +	+
0 +	+
0 0	-
- +	-
- 0	-
- -	-

b) 構成プリミティブ復元

三面図全体で閉ループの+, -, 0が決定したら改めて三面図間対応を取り、対応が取れた閉ループの形状の組合せとシステムが持っているプリミティブデータが一致する時、構成プリミティブとして復元する。

その後、構成プリミティブの大きさ、回転角度、原点位置を決定する。また集合演算のための演算子を閉ループ属性の組み合わせ（表3参照）から決定する。また対応している3個の閉ループの一番大きい depth 値をプリミティブの level とする。

level 導入の理由は「三面図処理でシェイプに合うように補助線設定を行う事は、三次元で考えた時、その部分で先に中間部品を作って置く事に相当する」と考えたことによる。level の導入により閉ループの depth の考え方をプリミティブ演算に反映させることができる。level の扱いは3.4で述べる。

表3 プリミティブ演算子決定表

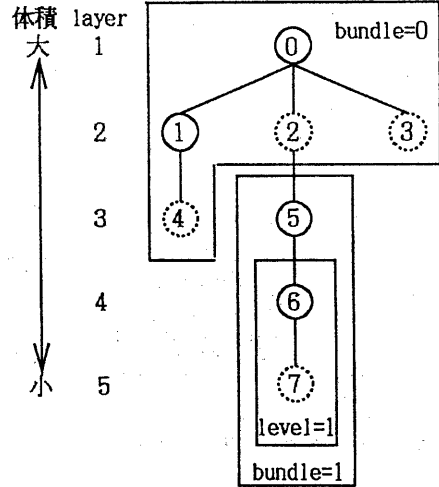
閉ループ属性	プリミティブ演算子
0 0 0	-
- - -	-
+ + +	+
- 0 0	-
0 0 +	+
- - 0	-
- - +	-
0 + +	+
- + +	-
- 0 +	-

3.4 プリミティブ演算処理

目的の立体を復元するために、構成プリミティブの演算順序を決定する必要がある。本システムでは演算順序決定法に、プリミティブの体積の降順と包含関係および幹と枝の関係を利用した木構造（図6）を導入する。

この幹と枝の関係は2で述べたように、「先に枝となる部分に含まれるプリミティブで中間部品を作成しておいて、それを幹の部分に繋げて処理を行い立体を復元する事を示している。

木構造の階層を表す指標として layer、幹と枝を区別する指標として bundle を用いる。以下に述べる方法で木構造を作成し、この木構造を基に演算順序を決定する。



円の中は、構成プリミティブの識別番号

○：演算子が“+”のプリミティブ

○：演算子が“-”のプリミティブ

図6 ツリーの概念図

a) 木構造作成

i) プリミティブの bundle を決定する。平面図・正面図・側面図の各面図で共通のグループに属する閉ループから構成されるプリミティブは、同じ bundle とする。特に各面図とも親グループに属する閉ループから構成されるプリミティブは bundle=0 とする（表4）。これが木構造の幹になり、bundle≠0 が枝になる。

表4 bundle 決定の例

プリミティブ	閉ループのグループ			プリミティブの bundle
	平	正	側	
0	親	親	親	0
1	親	親	子1	1
2	親	親	親	0
3	親	親	親	0
4	子1	親	親	2
5	親	親	子1	1
6	子1	親	子1	3
7	子2	親	親	4

- ii)各 bundle 内で構成プリミティブの体積の降順と包含関係から木構造を生成。またこの段階で、後の演算処理を難しくする共有プリミティブ(図7)を処理の対象から除外する。

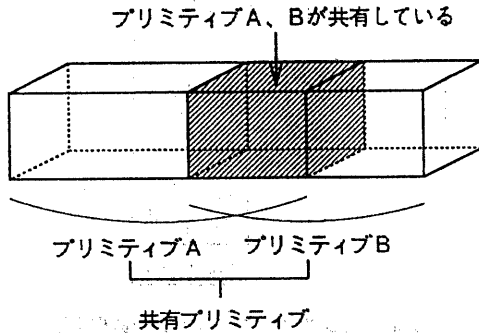


図7 共有プリミティブについて

- iii)bundle 毎の木構造を結合する。枝の木構造の一番上のプリミティブが幹の木構造のどのプリミティブと包含関係にあるかを調べ、位置が確定すると結合される。この際、枝の木構造が幹のどのプリミティブにも包含されない場合や、2個以上のプリミティブと包含関係にある場合には、幹と結合しない。

b)各種パラメータの扱い(演算の進め方)

- i)layer の扱い:layer の小さい方から演算を行う。同じ layer 内では演算は順不同。但しこれは、同一の bundle 内で行うものとする。
- ii)bundle の扱い:まず bundle≠0 (枝)で演算を行う。枝でできた部品を bundle=0 (幹)に付ける。部品の演算子は枝の一番上のプリミティブ演算子とする。ただし、枝のプリミティブの level が0以外の時は、level を0にした時の演算子とする。
- iii)level の扱い:一番大きい level から演算を始める。同一 level を持つプリミティブの中で演算を行う。この演算で得た部品の演算子を包含の上位に当たるプリミティブと逆符号にし、level を1小さくする。以上の処理を繰り返して、最終的に level=0 になるまで演算を行う。

c)具体例(図6)

図6を演算式にすると、以下のようになる。

$$S = P_0 + P_1 - P_2 - P_3 - P_4 + P_5'$$

$$(P_5' = P_5 - (P_6 - P_7))$$

ここで、Sは目的の立体モデル、P₀~P₇は図6でのプリミティブを示す。P_{5'}はP₅~P₇の演算を行った結果の「中間部品」に相当する。また+は前後のプリミティブを結合する演算、-は前のプリミティブから後ろのプリミティブを引く演算を表す。

4 実験

4.1 実験条件

- a)復元する立体は前述のプリミティブ(9種類)の組み合わせでできるものとする。
- b)より機械図面に近いものを使用する。ただし曲線情報は、真円のみとする。
- c)処理の対象図面は三面図で部分図や断面図は含まない。扱う線分は外形線と隠れ線とし、線分の省略は無いものとする。

4.2 実験結果

実験結果を図8、図9に示す。各々の立体モデルの図は本システムの出力結果を EusLisp[12] によって表示したものである。

5 おわりに

ここまで孤立部分の子グループとして扱う事、形状判別できない閉ループに対する補助線の設定、閉ループ属性の与え方、プリミティブ演算子の決定方法、プリミティブ演算処理での layer, bundle, level の扱い方を提案した。結果として図8、図9に示すようなサンプルを復元する事が可能になった。

一方、復元できない(復元結果が元の立体と違う)サンプルも存在する。この原因の代表的なものは、

- a)構成プリミティブが復元できない(三面図間で閉ループの対応が取れない。
- b)プリミティブ演算子が正しく決定できない。
- c)不要なプリミティブが存在する。
- がある。a)の対処法は二面図間での対応からプリミ

タイプを復元する事がある。プリミティブ復元処理部の改良が必要である。b)は閉ループ属性が正しくない事が原因であり、新しい情報(外形線・隠れ線に含まれる)の利用、新しい補助線設定の方法を考える必要がある。また木構造によるプリミティブ演算子の検証(包含情報の利用)も重要になる。c)は閉ループ探索で不必要な閉ループを探索するため、必要な閉ループのみを探索するアルゴリズムが重要になる。

また現在のシステムでは、どのサンプル(三面図)が復元でき、どれができないかというルール化ができていない。今後、探索する閉ループの種類、実線・隠れ線の情報の利用、補助線設定の方法、ツリーによるプリミティブ演算子の検証などの考えを用いることで、より有効な復元システムになると考える。

6 謝辞

本研究の立体復元結果の表示に使用した EusLisp は、電子技術総合研究所の松井俊浩氏によって開発されたものです。またこの研究の一部は青山学院大学総合研究所の援助によるものです。

7 参考文献

- [1] 出沢 正徳, 1972, 三面図から物体形成のためのシステム, 日本機械学会論文集, 38-310, 1267-1276.
- [2] 佐々木 康仁, 伊藤 潔, 鈴木 誠道, 1987, 三面図からの物体自動合成のための線形疑似ブール代数解法, 情報処理学会論文誌, 28-12, 1288-1297.
- [3] 原田 毅士, 木村 文彦, 1987, 3面図入力を基本にした3次元形状入力システムの試作, 情報処理学会研究報告, グラフィックスとCAD, 28-2, 15-22.
- [4] 西原 清一, 西田 淳, 池田 克夫, 1988, 探索手法による三面図からの多面体復元, 情報処理学会研究報告, グラフィックスとCAD, 35-4, 1-6.
- [5] 飯塚 厚, 横山 正明, 1990, 機械図面の計算機処理, 電子情報通信学会, 技術研究報告, PRU90-87, 5-11.
- [6] 横山 正明, 河上 恵三, 1990, 三面図から曲面を含む立体の自動生成, 日本機械学会論文集(C編), 38-526, 174-179.
- [7] 今 和生, 久保 洋, 大岩 高雄, 1991, 三面図上の二次元ループ情報からソリッドモデルを生成する一手法, 精密工学会春期大会論文集(1991年度), 2, 749-750.
- [8] 増田 健, 梶尾 秀彦 他, 1991, 三面図データからの立体の自動復元, 精密工学会春期大会論文集(1991年度), 2, 745-746.
- [9] 西田 淳, 張 紹星, 西原 清一, 1991, 面の組合わせ探索による三面図の解釈, 人工知能学会誌, 6-1, 96-104.
- [10] 千田 豊満, 1991, 三面図からの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, 32-9, 1122-1128.
- [11] 北嶋 克寛, 田坂 守康, 1992, プリミティブの直和および差演算のCSGを出力する三面図解釈の一手法, 電子情報通信学会論文誌, J75-DII-9, 1526-1538.
- [12] 松井 俊浩, 1992, EusLisp Reference Manual version 7.27, 電子技術総合研究所.
- [13] 柴宮 理, 吉原 千明, 1992, 機械図面自動認識システム, 青山学院大学理工学部機械工学科卒業論文.
- [14] 中庭 憲一, 横田 貴之, 1993, 三面図からの立体モデル自動復元システムの構築, 青山学院大学理工学部機械工学科卒業論文.

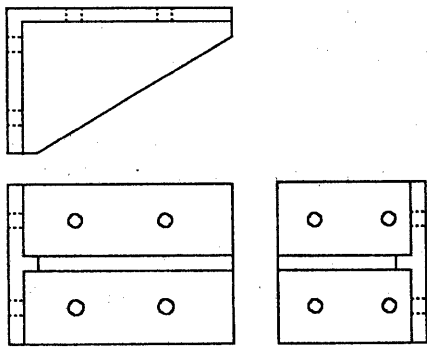


図8 サンプル1

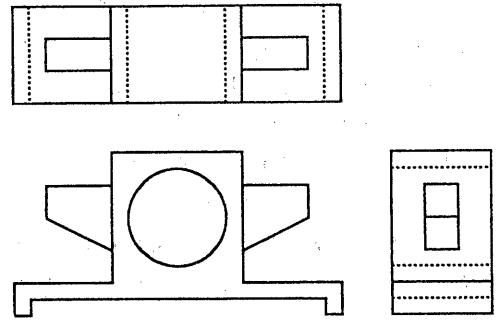
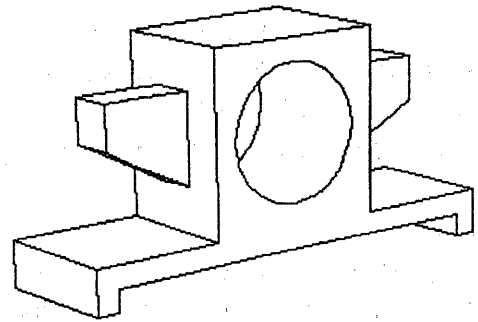
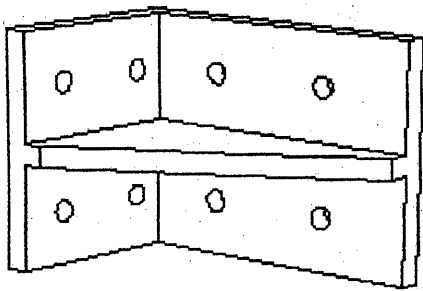


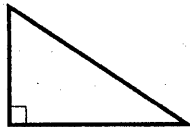
図9 サンプル2



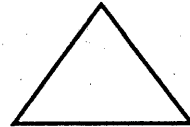
Shape#1 長方形



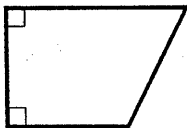
Shape#7 直角三角形



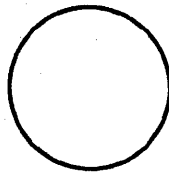
Shape#8 二等辺三角形



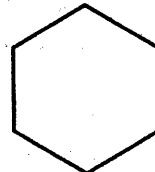
Shape#9 台形



Shape#10 円



Shape#17 六角形



Shape#18 一般的三角形



Shape#20 直角二等辺三角形



付図1 本システムで用いるシェイプ