

二分決定グラフの適用による三面図の効率的理解

正 木 寛 人^{†,☆} 斎 藤 逸 郎[†]

石 塚 満[†] 奥 乃 博^{††}

三面図は3次元モデルを表現するために最もよく使用されるが、それから3次元モデルを復元するときにはしばしば複数の解釈が生ずるという曖昧性の問題があり、従来はヒューリスティクスに基づき妥当な解釈を生成する方法がとられてきた。我々はこの問題に対し、適切な方向から見た面図を新たに1枚追加することで曖昧性を除去することができると考え、その追加面図に含まれることが要求される物体要素を計算する手法を考案した。本手法は、複数解釈された3次元モデルの中から、ユーザとのインタラクション（所望のモデルの指定）を通して曖昧性の所在を探るものであり、すべての人間が常識的に感じるとは限らないモデルが対象である場合、あるいは機械製図などの分野において用いる有効な補助面図の視線方向を決定する場合に特に有効である。復元過程におけるデータ保持を効率よく行うために、内部処理には論理関数のコンパクトな表現法である二分決定グラフ（BDD）を使用し、曖昧性解消のために用いる組合せ集合の管理には集合のコンパクトな表現であるZero-Suppressed BDD（ZBDD）を使用している。本手法に基づく三面図解釈の試作システムTOVIN（Three Orthographic Views INterpreter）についても記す。

Efficient Understanding of Three Orthographic Views by Employing Binary Decision Diagrams

HIROTO MASAKI,^{†,☆} ITSURO SAITO,[†] MITSURU ISHIZUKA[†]
and HIROSHI G. OKUNO^{††}

Although a set of three orthographic views is one of the most common representations of three-dimensional mechanical objects, it potentially has ambiguities which result in multiple interpretations of solids to be reconstructed. This paper presents a new method for eliminating the ambiguities by adding a new mapping plane, and describes its implementation in a prototype system TOVIN (Three Orthographic Views INterpreter). The processing of TOVIN consists of two parts. First, it encodes given views into pseudo Boolean formulas by applying reconstruction rules. By employing Binary Decision Diagrams (BDDs), TOVIN computes all possible interpretations efficiently and represents them compactly. Next, TOVIN requests a user to specify one from them on a display, and searches a set of critical object elements (edges and/or faces) to determine an additional mapping plane. This part employs Zero-suppressed BDDs (ZBDDs) which allow a compact representation of combinatorial sets and efficient set operations. The TOVIN runs quite efficiently thanks to the BDDs and the ZBDDs; this efficiency is the most salient feature of the system.

1. ま え が き

三面図は直行する3軸方向からの平行投影図（陰線は破線で描く）の組で3次元物体を表現したものであり、3次元物体の2次元平面上での表現として機械製図などの分野を中心として、古くから現在に至るまで

広く用いられている。コンピュータによる三面図理解のこれまでの研究の多くは、表現された3次元モデルの効率的復元、および三面図に誤りや過不足があるときにいかにしてその不都合を乗り越えるかを目標としてきた¹⁾。

三面図は本来3次元空間中の物体を2次元平面上へ投影した表現であるため、解釈が曖昧になる可能性が存在し、対応する3次元モデルが一意に定まらない場合があるという問題が内在している。特に、三面図は製図を行う人が曖昧性が少ない方向ではなく、書きやすくなる方向を座標軸に設定することが多くあるために、多義解釈の問題がしばしば生じることになる。こ

[†] 東京大学工学部電子情報工学科
Faculty of Engineering, University of Tokyo

[☆] 現在、KDD 研究所
Presently with KDD Laboratories

^{††} NTT 基礎研究所
NTT Basic Research Laboratories

のような三面図解釈の曖昧性の問題に対し、これまでではたとえば物体には対称性がある、平行性や垂直性が高い、凹凸が少ないなどのヒューリスティクスを用いて、人間が常識的に感じる3次元モデルを一意的に合成する研究が行われてきた²⁾。

一方、3次元物体を表現する三面図以外の手法として、位相的に異なるすべての可能なアスペクトを節点とし、それらの間の可能な遷移関係に対して枝を張った Aspect Graph が提案されている³⁾。ここでは、Aspect Graph が有する表現力の完全性を、情報量が冗長になることをできるだけ避けながら三面図に反映させることを目指す。その一手法として、ユーザとのインタラクションにより三面図の曖昧性の所在を稜線あるいは面の組合せの形で求め、その組合せを含む適切な面図を三面図に加えることで曖昧性の解消を可能にする。本手法は、機械製図などの分野で補助面図を加える際にその効果的な視線方向を示唆できる点において、また三面図があまり常識的でない3次元物体を対象にして描かれたものである場合にも、曖昧性を除去可能とする点において有効である。三面図は3次元モデル生成のための表現ととらえることができるが、我々の研究は、この生成のための表現に完全性を付与するための一手法ととらえることができる。

解釈可能な3次元モデルの生成と曖昧性の解消の効率化のために、二分決定グラフ (BDD) を導入し、処理速度の向上を図っていることも大きな特徴である。なお、ここで扱う3次元物体のクラスは多面体のうち複数に分離していないもの (多様体) に限るものとする。また、端点・線分・領域はいずれも面図上の物体要素に対しての用語として用い、頂点・稜線・面は3次元空間中の物体要素に対して用いることにする。

以下、2章で三面図から対応する3次元モデルを復元する手法の概要と、擬似ブール代数を用いる解法を紹介する。次に、3章で三面図を表現する論理式のデータ構造として用いる二分決定グラフ (BDD) について、4章で探索過程と曖昧性除去への BDD の適用について説明し、試作システム TOVIN について記す。

2. 三面図の解釈と対応物体の復元

入力された三面図から対応する3次元モデルを復元する方法には様々なものが存在するが、それらの共通点は図1のようにまとめられる。以下で、候補物体要素列挙と探索過程について本研究に関連する事項を説明する。

2.1 候補物体要素列挙

所与の三面図から、候補頂点、候補稜線、候補面、そ

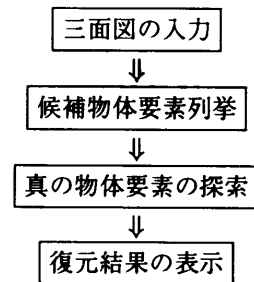


図1 三面図理解の概略

Fig. 1 Outline of understanding methods of three orthographic views.

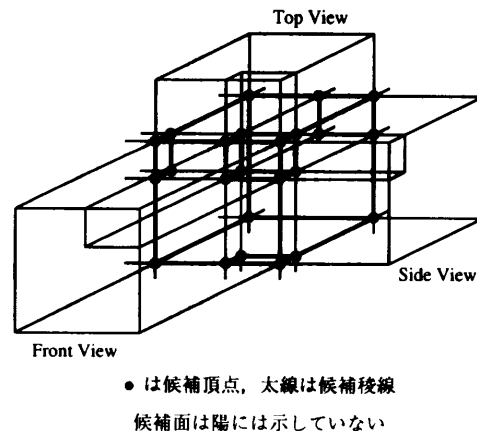


図2 候補物体要素列挙 (文献4) より引用)

Fig. 2 Candidate enumeration of object elements (from Ref. 4)).

して候補物体をこの順に列挙する。候補物体要素 (頂点、稜線、面) 列挙には伊藤の方法を用いる⁴⁾。以下はその要素であり、図2にその概観を示す。

候補頂点の列挙は、三面図上の端点と端点との対応、あるいは端点と線分との対応を基にして行う。後者の必要性は即座には理解できないが、頂点が3つの面図すべてにおいて端点として投影されない場合を扱うために必要となる。

候補稜線は、原理的には任意の2つの候補頂点間に張ることができるが、三面図上に対応する端点あるいは線分が存在しなくてはならない。

候補面とは、候補稜線による閉ループのうち、その内部に候補稜線を含まないものである。したがって、内部に穴を持つ候補面も存在しうる。

一般に候補物体要素列挙では、三面図には合致するものの所望の物体には存在しない物体要素 (虚物体要素と呼ばれる) や、虚ではないが物理的に同時には存在しえない複数の物体要素 (非両立物体要素と呼ばれる) が含まれる可能性があり、2.2節で述べる探索過程においてこのような不適切な候補物体要素を除去し

なくてはならない。

2.2 真の物体要素の探索

従来の研究事例における真の物体要素の探索過程は、以下の2つの方法に大別される。

逐次型探索 ワイヤフレームモデルやサーフェスモデルの物体要素のつながりや、幾何学的な情報に関する規則を手続きとして組み込んだ探索プログラムにより、真の物体要素の判定と偽の物体要素の除去を行い、3次元モデルを復元する手法である。探索プログラムは研究者ごとに微妙に異なっているが、虚物体要素の除去、非両立物体要素の選択、三面図との適合チェックなどを行っている点では大体共通している⁴⁾。縦型の探索過程と見なすことができる。

擬似ブール代数解法 候補稜線と候補面にそれぞれ e_i, f_j という論理変数を割り当て（所望の物体中に存在する場合に1、存在しない場合に0をとる）、擬似ブール代数で可能な制約関係を定式化する。これを汎用解法（Hammerの解法⁵⁾）で解いて、無矛盾な物体モデルを求める手法である⁶⁾。手続き記述でなく宣言的（あるいは制約）記述に基づく解法であり、横型の探索過程と見なすことができる。

本研究では宣言的記述に基づくことにより汎用性が高い後者の手法をとる。一般に宣言的な記述に基づく処理は汎用性は高いものの処理効率が十分でないという問題を生じるが、ここではBDDの導入によりこの問題の解決を図る。

2.3 擬似ブール代数解法

基礎となっている、佐々木ら⁶⁾により提案された擬似ブール代数解法における復元規則と擬似ブール代数による定式化を説明しておく。まず用いる記号を下記に示す。

- F_e : e を構成する（辺として含む）候補面群。
- $\langle f_i, f_j \rangle$: 同一平面上にある2つの候補面の組。
- F_x : 非両立物体要素である候補面の組。
- F_{cross} : F_x の集合。
- s : 三面図上の（任意の）線分。
- S_{real} : 三面図上の実線分の集合。
- S_{hidden} : 三面図上の破線分の集合。
- S : S_{real} と S_{hidden} の和集合。
- $E(s)$: ある面図上で線分 s として投影される候補稜線群。
- $E_{\text{pre}}(s, e)$: $E(s)$ 中の候補稜線 e よりも前にある $E(s)$ 中の他の候補稜線群。
- $F(s)$: 線分 s を内部に含む領域として投影される

候補面群。

- $F_{\text{pre}}(s, e)$: $E_{\text{pre}}(s, e)$ のすべての候補稜線を構成する候補面群と、 e より前にある $F(s)$ 中の候補面群の和集合。
- $F_{\text{post}}(s, e)$: e と e の直後にある候補稜線の間にある $F(s)$ 中の候補面群。

2.3.1 Rule 1

候補稜線 e が真ならば、 e を構成する候補面群 F_e のうち、coplanar でない2つの候補面が真である。候補稜線 e が偽ならば、 e を構成する候補面群 F_e のうち、すべての候補面が偽であるか、coplanar な2つの候補面が真である。

すべての e に対して、

$$\sum_{f \in F_e} f = 2 \cdot e + 2 \cdot \bar{e} \cdot \left(\sum_{\langle f_i, f_j \rangle \in F_e} (f_i \cdot f_j) \right), \quad (1a)$$

$$\left(\sum_{\langle f_i, f_j \rangle \in F_e} (f_i \cdot f_j) \right) \leq \bar{e}. \quad (1b)$$

2.3.2 Rule 2

交線となる稜線なしで交差する候補面群 F_x のうちただか1つが真である。

すべての $F_x \in F_{\text{cross}}$ に対して、

$$\left(\sum_{f \in F_x} f \right) \leq 1. \quad (2)$$

2.3.3 Rule 3

ある面図上の同じ線分から得られた候補稜線群はそのうち1本以上が真である。

すべての $s \in S$ に対して、

$$\left(\sum_{s \in E(s)} e \right) \geq 1. \quad (3)$$

2.3.4 Rule 4a (実線規則)

ある面図上の実線分 S_{real} から得られた候補稜線群のうち、最も投影面に近い（最も前にある）候補稜線について、この候補稜線よりも投影面に近く、この候補稜線を覆い隠す候補面群中のすべての候補面は偽である。

すべての $s \in S_{\text{real}}$ に対して、また $E_{\text{pre}}(s, e) = \emptyset$ である $e \in E(s)$ に対して、

```

# Rule 1
C1 = (f1 + f5 == 2 * e1)
...
# Rule 2
D1 = (f8 + f11 <= 1)
...
# Rule 3
E1 = (e5 + e18 >= 1)
...
# Total Constraints
F = C1 & C2 & C3 & ... & D1 & D2 & D3 & ...

```

図3 擬似ブール代数式によるコーディングの例

Fig. 3 A coding example using pseudo Boolean equations.

$$\left(\sum_{f \in F_{pre}(s,e)} f \right) = 0. \quad (4a)$$

2.3.5 Rule 4b (実線規則)

ある面図上の実線分から得られた候補稜線群のうち、1本の候補稜線 α が偽であり、かつ α より前にある他の候補稜線もすべて偽であるならば、 α を構成するすべての候補面が偽であり、かつ α と α の直後にある他の候補稜線の間にある候補稜線を覆い隠す候補面も偽である。

すべての $s \in S_{real}$ に対して、

$$\bar{e} \cdot \left(\prod_{e \in E_{pre}(s,e)} \bar{e} \right) \cdot \left(\prod_{f \in F_e} \bar{f} \right) \cdot \left(\prod_{f \in F_{post}(s,e)} \bar{f} \right) = \bar{e} \cdot \left(\prod_{e \in E_{pre}(s,e)} \bar{e} \right). \quad (4b)$$

2.3.6 Rule 5 (破線規則)

ある面図上の破線分 S_{hidden} から得られた候補稜線が真であるならば、この稜線よりも投影面に近く、この稜線を覆い隠す候補面群中の1つ以上の面が真である。

すべての $s \in S_{hidden}$ に対して、

$$e \cdot \left(\prod_{f \in F_{pre}(s,e)} \bar{f} \right) = 0. \quad (5)$$

図3に図7(a)を例とした擬似ブール代数式によるコーディングの一部を示す。

3. 二分決定グラフ (BDD)

二分決定グラフ (Binary Decision Diagram, 以下 BDD と略記する) は 1978 年に Akers が提案した論理

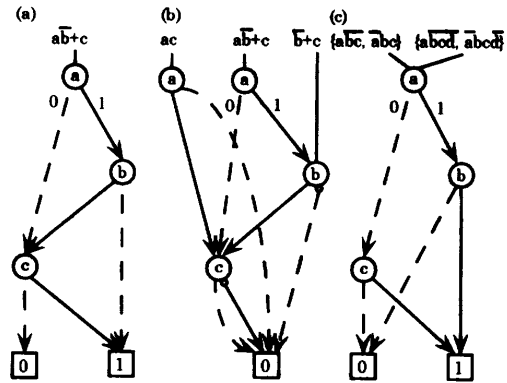


図4 二分決定グラフの例: (a) ROBDD, (b) SBDD, (c) ZBDD.

Fig. 4 Examples of BDD: (a) ROBDD, (b) SBDD and (c) ZBDD.

関数のコンパクトな表現法であり⁷⁾, 1986年に Bryant がその効率的な演算法を考案したことによって一躍脚光を浴びるようになった⁸⁾. 図4はBDDの一例であるが, 入力変数と最終的な論理関数値を節点とし, 入力変数のとる値を枝とすることで, 論理関数の Shannon 展開 ($f = \neg x \wedge f_{x=0} \vee x \wedge f_{x=1}$) を有向グラフ表現したものであるといえる. ある論理関数を表現する BDD は一般に複数存在するが, これらに対して既約化 (冗長な節点の削除, 等価な節点の共有) を施したものはみな標準形になることが知られており⁸⁾, これを ROBDD (Reduced Orderd BDD) と呼ぶ. ここで “Orderd” とは, BDD の根節点から終端節点へ向かうすべてのパスにおいて遭遇する変数の順序が, あらかじめ決められている全順序に従うことを意味する.

湊らは複数の ROBDD 間で等価なサブグラフを共有する SBDD (Shared BDD) を提案しており, 記憶効率の点などで優れている⁹⁾.

さらに, 湊は集合ではリテラルが選択されないことがデフォルトとしての意味を有することに着目し, 1枝が直接0定数節点 (図4の $\boxed{0}$) へ向かっているような節点を削除するという既約ルールを導入した Zero-Suppressed BDD (以下 ZBDD と略記する) も提案している¹⁰⁾. ZBDD は BDD と既約ルールの点では異なるものの, グラフが標準形になる点では共通しており, また解密度^{*}の希薄な集合を扱う場合には BDD よりもコンパクトになり, また効率的な集合演算が可能であることが知られている.

^{*} n 個の入力変数に対して 2^n 通りの入力割当てが考えられるが, ここで充足解の個数を m としたとき, $m/2^n$ を解密度と呼ぶ. 組合せ集合に関しても同様である.

BDD の主な特徴は、以下の 3 点にまとめられる。

- (1) 入力変数の順序を固定すると、ある論理関数に対して一意の標準形の BDD が得られる。
- (2) 多くの実用的に重要な論理関数に対して、コンパクトなサイズとなる。
- (3) 論理関数に対する演算がグラフのサイズにほぼ比例する時間で、あるいは論理関数間の演算がそれぞれのグラフのサイズの積にほぼ比例する時間で可能。

矛盾関係を含む宣言的記述のもと、あるいは制約条件下で可能解を求める問題は一般に NP 完全問題となり、最悪ケースの計算量は問題サイズに対して指数オーダーで増加することは避けられない。しかし上記 (2), (3) により、多くの実用的な場合に対してこのような計算量の増大を回避できる可能性を持つことになる。

また、BDD が全解を同時に表現していることに起因して、複数の解の比較、内包表現と外延表現の比較、制約条件の付加などを容易に行うことができる。これらの機能は、解を求める際にバックトラックを引き起こすタイプの探索プログラムでは効率的になしえず、BDD を効果的に応用する際の重要な指針となる¹¹⁾。

4. BDD を用いた三面図の効率的な理解

本章では、二分決定グラフを用いた三面図理解について、復元過程の効率化と曖昧性除去の 2 点から説明する。図 5 はこの原理に基づいて開発した曖昧性を除去する適切な面図の追加機能を有する三面図理解システム TOVIN (Three Orthographic Views INterpreter) の処理の流れを示している。

4.1 BDD の適用による復元過程の効率化

擬似ブール代数解法における連立擬似ブール代数式の処理の部分には、従来 Hammer の解法⁵⁾と呼ばれる汎用解法が用いられてきた。Hammer の解法の要点を以下に記しておく。まず、単一式については、以下のようなになる。

- (1) 係数を自然数のみにする変換、項の順序を係数に関する降順にする変換を行い、所与の式を標準形にする。
- (2) 判定ステップ：あらかじめ解析されている係数間の関係について場合分けされた表に従って、値の固定できる変数とそうでない変数を区別する。固定できる変数についてはその値を求め、固定できない変数についてはその変数を含む(規模の小さくなった)新たな式を導出する。
- (3) 新たに得られた式に対して、上記の判定ステッ

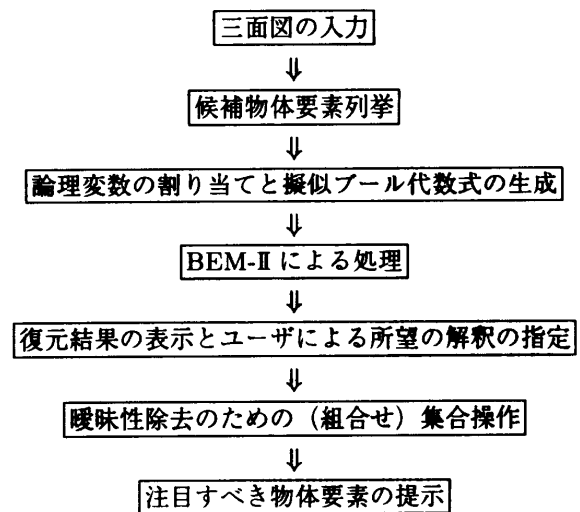


図 5 三面図理解システム TOVIN の処理の流れ

Fig. 5 Processing flow of the understanding system for three orthographic views: TOVIN.

プを繰り返す。最後に値の固定できない変数が残った場合、その変数は任意の値をとる。

連立式となった場合も基本的には同じであり、また非線形項が出現した場合には、新たな変数で置き換える。この置き換えにより式の本数が増え、問題の規模を大きくすることになる。

我々はこの Hammer の解法を、BDD によるものに置き換える。具体的には、湊の開発した算術論理式計算プログラムである BEM-II¹²⁾を BDD の処理系として用いた。これは、2.3 節に記した擬似ブール代数による復元ルールの定式化には算術演算(四則演算・大小比較演算)が含まれており、BEM-II との相性が良いためである。Hammer の解法による処理では三面図を表現する連立擬似ブール代数式を汎用解法により解くのにに対し、BDD による処理では与えられた連立擬似ブール代数式を基にしてコンパクトにデータ保持を行う BDD 構築フェーズと、構築された(問題を表している) BDD 操作フェーズ(たとえば、全充足解やコスト最小の入力割当てを求める)とに分けて解を求める。上述の全充足解を求める操作は、BDD 上では根節点から 1 定数節点(図 4 の [1])へのすべてのパスを求めることに相当する。

4.2 三面図の曖昧性除去

1 章で述べたように、三面図は解釈が曖昧になる可能性がある。したがって、まさに実用的なシステムを開発する場合、復元過程の効率化や誤り等のある入力を処理する機構の開発とともに、曖昧性を解消することが重要な役割を果たす。我々は、ヒューリスティクスを用いて最もありうる解釈を得るというアプローチ

Phase I: 状態集合の計算

各解釈における変数の真偽の値を列挙する。



Phase II: 反転要素集合の計算

ユーザの指定する解釈における変数の値と異なっている変数を、各解釈について列挙する。



Phase III: 注目要素集合の計算

反転要素集合の直積をとることで、追加面図において注目すべき物体要素を計算する。

図 6 曖昧性除去のための追加面図の条件の計算法

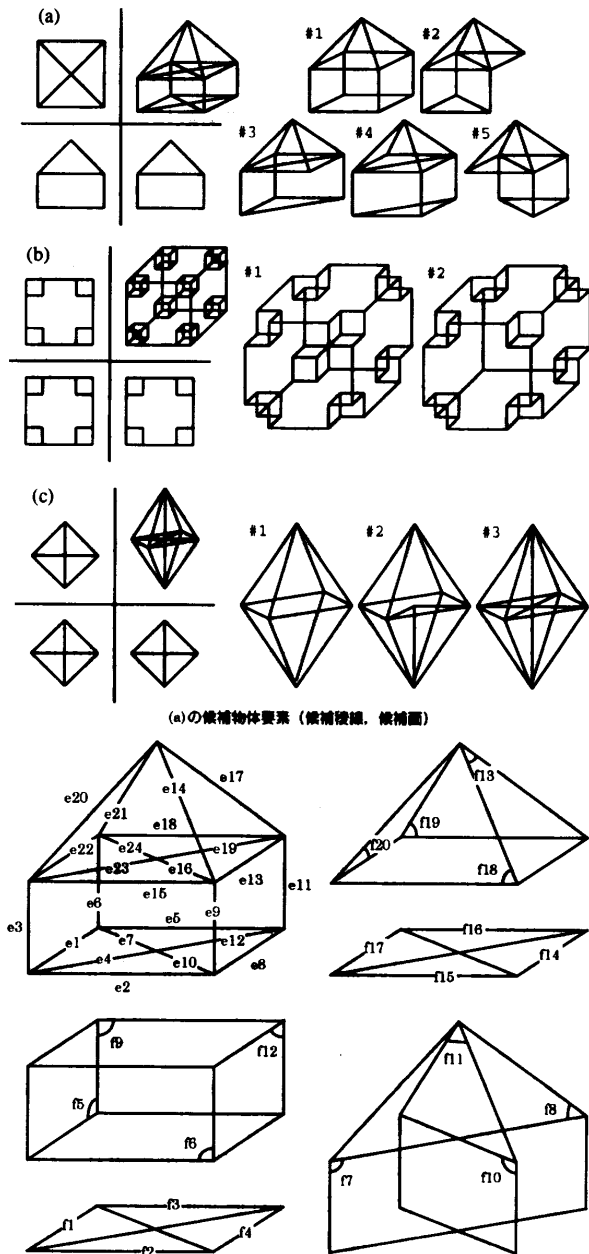
Fig. 6 Computation of the condition of an additional view for ambiguity elimination.

でなく、BDD による効率的処理能力を活用し、意図する 3 次元モデルが一意に解釈されるために必要となる、適切な方向から見た 1 枚の面図を追加するシステムを開発した。システムは可能な解釈をすべて計算、提示し、ユーザが意図する解釈を指定すると、曖昧性を解消する追加の面図を自動生成する。適切な追加面図生成は以下の原理に基づいて行う。

“三面図の解釈がユーザの所望する 3 次元モデルに一意に対応するためには、その他の解釈可能なモデルとの違いに注目しなくてはならない。そのためには、最終的な解釈において真偽値が反転しているような変数に対応する物体要素（以下では注目要素と呼ぶ）が含まれる面図追加が適切である。”

この原理に基づく曖昧性除去の具体的な処理の流れを図 6 に示す。本手法では、情報（データ）を 2 種類の方法で管理する。まず論理式を用いて三面図を表現し、これにより 3 次元モデルの可能な解釈すべてを論理式で保持する。論理式の効率的な表現法として BDD を使用する。次に注目要素を求めるときには、物体要素の組合せを集合で表現する。組合せ集合の効率的な表現法として ZBDD を使用する。集合を表現するのに BDD ではなく ZBDD を用いる理由は、通常の論理式や算術論理式では曖昧性除去のコーディングが容易でないこと、そして変数の数に対して指数オーダーのサイズになる問題空間に対して、得られる結果は非常に希薄になることによる^{*}。以後、理解を容易に

^{*} 2 種類の BDD の使い分けは本稿が最初ではなく、論理式の prime implicants を求めるときにも使用されている¹³⁾。BDD で論理関数を表現し、その BDD の 1 パスをたどりながら得られた変数の組合せを管理するのに ZBDD を使用すると、従来よりも効率よく膨大な prime implicants を求めることができる。



三面図の右上部は候補物体要素を示す。(c)については #2 のタイプが 8 個、#3 のタイプが 4 個得られる。

図 7 三面図および対応モデルの例 ((a), (b) は文献 6) より, (c) は文献 14) より引用)

Fig. 7 Examples of views and their corresponding solid objects.

するために図 7 (a) の例を参照しながら説明を行う。

- (1) まず、三面図を表す BDD (これを F とする) に対して全充足解を求める操作を施す。これは、BEM-II においてコマンド `print F` を実行すればよく、その結果は積和標準形で出力される。同時に、コマンド `print /count F` で充足解

表1 可能な解釈結果の真理値表

Table 1 A truth table of possible interpretations.

変数 \ 解釈	#1	#2	#3	#4	#5
e1	1	1	1	0	0
e2	1	1	0	1	0
e3	1	1	1	1	0
e4	0	0	1	1	0
e5	1	0	1	0	1
e6	1	1	1	0	1
e7	0	1	0	0	1
e8	1	0	0	1	1
e9	1	1	0	1	1
e10	0	1	0	0	1
e11	1	0	1	1	1
e12	0	0	1	1	0
e13	1	1	1	1	1
e14	1	1	1	1	1
e15	1	1	1	1	1
e16	0	1	0	0	1
e17	1	1	1	1	1
e18	1	1	1	1	1
e19	0	0	1	1	0
e20	1	1	1	1	1
e21	1	1	1	1	1
e22	1	1	1	1	1
e23	0	0	1	1	0
e24	0	1	0	0	1

の個数を求めておき、ここでは $n+1$ であるとする。

- (2) 次に、得られた各解釈における変数の真偽値を BDD 処理の外側で求める (BDD と ZBDD との変換をスムーズに行うことが可能な処理系を用いた場合には本ステップは不要になるが、ここでは BDD と ZBDD の処理系として別々のものを用いたため、このような中間的なステップを置いた)。表 1 は、図 7(a) の例に対して得られる 5 つの解釈における各稜線の真理値を例示している。

- (3) 真理値の反転する新変数の導入：候補物体要素に割り当てられているすべての論理変数 x に対して、その真偽の意味が逆であるような新たな論理変数 x' (所望の物体中に存在しない場合に真となり、存在する場合に偽となる) を用意する。このような論理変数の導入は、三面図を表す BDD 構築の最終部で、

e.g.)

$$x \oplus x' = 1 \quad (\oplus: \text{排他的論理和})$$

という形の制約を論理積により付加することで実現でき、あるいはそのような制約付加を行わずに、BDD 処理の外側において変数の真偽値に応じて反転した値で新変数を解釈してもよい。どちらにするかは使用する処理系に依存し、こ

こでは前述したように BDD と ZBDD の処理系が別々のものであることから、BDD 処理の外側で解釈する方法で実装している。この新変数の導入は、後述する集合の引算の際に、本来偽であるような変数が演算後に残るようにするために置く。

- (4) 各解釈に対して、その値が真であるような論理変数のみを列挙し、これらで構成される集合 S_i (以下では状態集合と呼ぶ) を計算する。たとえば表 1 の解釈 #1 に対しては、以下のような S_1 が計算される。

e.g.)

$$S_1 = \{e1, e2, e3, e4', e5, e6, e7', e8, \dots\}$$

(ここで $e4'$, $e7'$ は前ステップで導入した $e4$, $e7$ を反転した変数である。)

- (5) ユーザが意図する解釈 # j を指定したとする。指定された解釈とその他の各解釈 # i との間で、# i の状態集合に含まれ # j に含まれない候補物体要素の集合 C_i (以下では反転要素集合と呼ぶ) を計算する。具体的には、

$$C_i = S_i - S_j \quad (1 \leq i \leq n+1, i \neq j)$$

の差集合演算によって計算する。

たとえば表 1 の例で、ユーザが解釈 #1 を指定したときの解釈 #2 に対する C_2 と、解釈 #4 に対する C_4 はそれぞれ以下ようになる。

e.g.)

$$C_2 = \{e5', e7, e8', e10, e11', e16, e24\},$$

$$C_4 = \{e1', e4, e5', e6', e12, e19, e23\}$$

- (6) 注目要素集合 R はユーザの指定した解釈(# j) とその他のすべての解釈とで異なることが必要となる。したがって、全体で n 個ある反転要素集合の直積を以下のようにしてとって求める。

$$R = \bigotimes_{i=1, i \neq j}^{n+1} C_i$$

($\otimes$ は直積を表す)

表 1 の例では、ユーザが解釈 #1 を指定したときの注目要素集合は番号順に整理すると以下のようになる。

$$\{\{e1', e2', e3', e5'\}, \{e1', e2', e3', e7\}, \dots\}$$

- (7) 注目要素集合の各要素 (これを cube と呼ぶ) のうち、選択された変数の個数が最少のものを選び出す。これは曖昧性除去のための追加面図において、多数の物体要素に注目するよりも、できるだけ少ない個数の物体要素に注目する方が自然なためである。本ステップにより計算さ

表2 BEM-IIによる処理結果
Table 2 Results of processings with BEM-II.

例	#var	#eqn	#solid	#node	time [sec]
(a)	44	103	5	194(672)	3.1
(b)	162	432	2	265(2038)	34.7
(c)	38	84	13	261(662)	3.1

れる集合を最少注目要素集合と呼ぶ。

(上記(6), (7))の具体的実現方法としては、ZBDD ライブラリパッケージに機能追加した `foreach_zbdd_cube{}` マクロを用いて、注目要素集合の各 cube で選択された変数の数をチェックし、現在保持している最少値と同じであれば最少注目要素集合に入れ、それよりも少なければ新たに最少注目要素集合を作成し直すというものである。)

表1の例でユーザが解釈 #5 を指定したときの最少注目要素集合の例は $\{e3\}$ になり、ユーザが解釈 #1 を指定したときの最少注目要素集合は以下になる。

$$\{e1', e8'\}, \{e2', e5'\}, \dots, \{e19, e24\}$$

追加面図はこの最少注目要素を表示するものとすれば、曖昧性の除去が図れることになる。

4.3 BDD 適用の効果と試作システムの実装

上述の考えに基づき、Sun SPARC10 上に曖昧性除去の面図追加機能を持つ三面図理解の試作システム TOVIN (Three Orthographic Views INterpreter) を実装した。曖昧性除去のための組合せ集合の管理には UCB で開発された ZBDD のライブラリパッケージを用い、C 言語でプログラミングした。以下に、BDD 適用の効果と TOVIN の使用例を示す。

図7の各三面図から得られた連立擬似ブール代数式を BEM-II で処理し、全解探索を行ったときの処理時間を表2に示す。処理時間は `tcsh` の下で UNIX の `/usr/bin/time` コマンドを用いて測定した。表中、`#var` は候補物体要素列挙の結果得られた候補稜線と候補面の総数を、`#eqn` は復元規則適用の結果立式された擬似ブール代数式の数を、`#solid` は復元される3次元モデル(解釈)の数を、`#node` は全解釈を表す BDD の節点数(括弧内は処理系全体で使用する節点数)を、`time` は実行時間のうちの `user time` を示している。

一般に BDD を使用する際には、変数や制約式適用の順序付けの問題*が生じ得ることが知られているが、

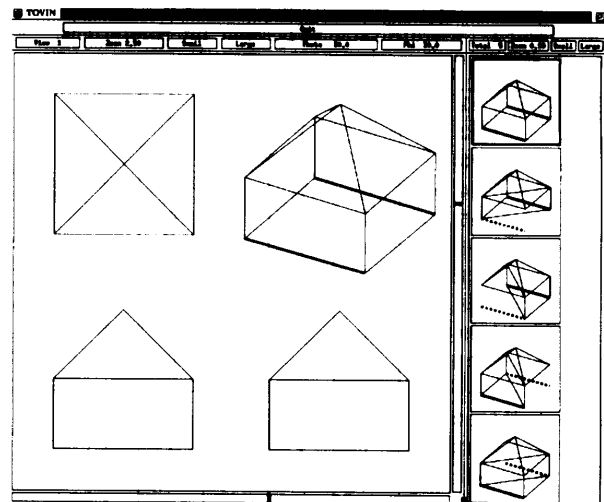


図8 試作システム TOVIN の表示画面

Fig. 8 A display of the prototype system: TOVIN.

表2においてはそのような問題への対処を行っていないので、今後処理時間の絶対値を小さくする余地が残されている。この場合、文献11)で提案されている CCVO 法の適用がその有力な候補となりうる。表2を見る限りにおいては `#var`, `#eqn`, `#node` (括弧内の数字)の間には相互に比例関係が、またこれらと処理時間との間には2乗の関係があるように見えるが、厳密にはさらに検討を行う必要がある。Hammer の解法では非線形項を新変数に置き換えるため、その置き換え自体が新たな制約式の追加となり、結果的に問題の規模を大きくしてしまう性格を帯びている。これに対して(S) BDD を適用する本手法では、制約式自体の表現をコンパクトにし、複数の制約式間の冗長性を共有することで排除しているため、BDD の構築フェーズにおいてより大規模な問題を扱う(表現する)ことが可能となり、そのことが結果的に、操作フェーズに要する処理を相対的に軽くすることにつながっている。

さらに、文献6)では評価のグラフの横軸に「制約式の延べ評価回数 × 式の本数」なる量を置いて対象物体の複雑度としている。我々は BDD の1パスをたどることで解を得るため、そのような複雑度として BDD の節点数を採用することができるといえる。表2の `#node` は、そのような観点から示したものである。`#node` の括弧外の節点数が探索の時間にはほぼ比例する量となり、括弧内の節点数は問題表現(BDD構築)に要する時間に対応する量となる。

次に、曖昧性除去の処理の様子を図8に示す。図8は、図7(a)を例として TOVIN を起動した表示画面である。左側のメインパネルに三面図とユーザの所望する3次元モデル(解釈)が、また右側のサブパネル

* 同じ問題を表現する場合でも、順序付けの良し悪しで構築される BDD の大きさに差異が生じるという問題。

にすべての解釈可能なモデルが表示されている。メインパネルの下側と右側にあるスライダバーは各モデルを回転させるためのものであり、ユーザが所望のモデルを指定するときのために作成した。図8は3次元モデル#1（サブパネル中の一番上のモデル）が指定された場合に対応しており（モデル#1の枠が太線となっている）、上述した最少注目要素集合の要素のうち、1番目の要素である $\{e1', e8'\}$ が太線表示されている。点線は、注目すべき要素がそのモデル中には本来存在しないことを示している。この場合は太線表示の部分が可視となる視線方向である下面図（bottom view）を追加すれば、曖昧性は解消されることになる。

追加面図の決定は一般に、太線表示される最少注目要素集合が可視となる視線方向を、以下のような順序で探せばよいといえる。

- (1) $-x$ 軸方向, $-y$ 軸方向, $-z$ 軸方向。
- (2) $(x, y, z) = (1, 1, 0)$ や $(0, 1, -1)$ など、2本の軸で構成される平面の中で、それらの軸から 45° をなす方向（全部で12方向）。
- (3) $(x, y, z) = (1, 1, 1)$ や $(-1, 1, -1)$ など、原点を中心とする立方体の対角線方向（全部で8方向）。

決定された視線方向からの各解釈の追加面図を図7(a)を例として図9に示す。

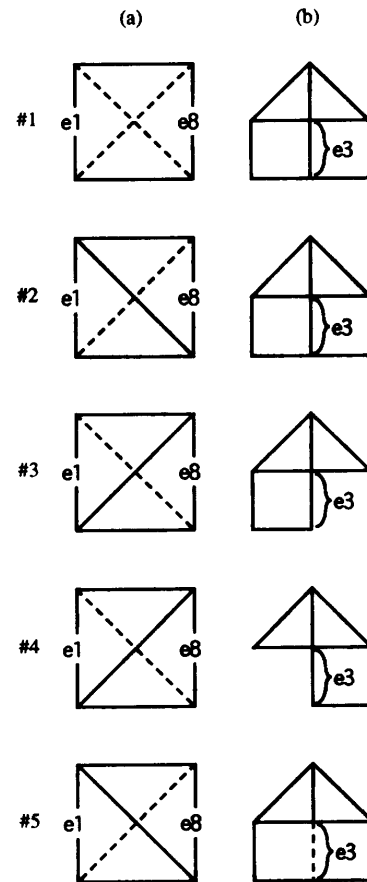
しかし、この機能は現在の TOVIN ではまだ実装されていない。また複雑な物体で、すべての最少注目要素集合を可視とするのに1つの追加面図では不十分で、2枚以上の追加面図を必要とする場合もあり得るかもしれないが、十分には確認していない。

5. む す び

三面図の効率的な復元やその曖昧性除去に、二分決定グラフ（BDD）を用いる手法を示した。ここでの曖昧性除去は、ユーザとのインタラクションを通して達成されるもので、ヒューリスティクスを用いて常識的に最もありうる3次元モデルを一意に復元する手法とは相補的な関係にある。我々の手法は効率的な全充足解探索を含み、必ずしもすべての人間にとって常識ではないモデルに対しても曖昧性除去のメカニズムを保証している。機械製図などの分野で曖昧性を除去するために補助面図を追加する際に、効果的な視線方向の示唆を与えるなどの利点がある。

BDDを用いることのメリットは以下のようにまとめられる。

- ある計算機環境下において、従来では困難だったような大規模な問題も現実的な処理時間で扱うこ



(a)
解釈#1の指定に対して、最少注目要素として例えば $\{e1, e8\}$ を選択し、これらが可視となる視線ベクトル $(0, 1, 0)$ からの各解釈の面図——解釈#1の面図は他と異なっている

(b)
解釈#5の指定に対して、最少注目要素として $\{e3\}$ を選択し、これらが可視となる視線ベクトル $(1, 0, -1)$ からの各解釈の面図——解釈#5の面図は他と異なっている

図9 決定された視線方向からの各解釈の追加面図

Fig. 9 Additional view planes from determined viewing directions.

とができる。

- 横型探索の（命題）論理プログラミングを提供するツールと考えることができ、本研究のように複数の解どうしの比較・演算を効率的に行うことができる。

本論文では多面体の物体のみを対象にして記したが、球や円柱、円錐の曲面を含む物体に対する三面図理解の手法についても開発を行っている¹⁵⁾。

謝辞 三面図の擬似ブール代数展開の情報をいただいた上智大学伊藤潔先生、BEM-IIの使用の便宜を図っ

ていただいた NTT LSI 研究所湊真一氏, ZBDD のライブラリパッケージを使用させていただいた UCB の久木元裕治氏に感謝いたします。

参考文献

- 1) 渡辺 崇, 田代明子, 藤井省三: 矛盾を含んだ三面図情報からの物体の推定, 情報処理学会論文誌, Vol.35, No.11, pp.2320-2329 (1994).
- 2) 高橋正充, 佐々木康仁, 伊藤 潔: 非線形擬似ブール計画法による曖昧な三面図からの多面体の一意的合成, 人工知能学会誌, Vol.6, No.6, pp.904-911 (1991).
- 3) Bowyer, K.W. and Dyer, C.R.: Aspect Graphs: An Introduction and Survey of Recent Results, *Int'l Journal of Imaging Systems and Technology*, Vol.2, No.4, pp.315-328 (1990).
- 4) 伊藤 潔: 三面図を用いたソリッドモデルの構成—主に多面体を対象として—, 情報処理学会誌, Vol.31, No.8, pp.1095-1106 (1990).
- 5) Hammer, P.L. and Rudeanu, S.: *Boolean Methods in Operations Research and Related Areas*, Springer-Verlag, New York (1968).
- 6) 佐々木康仁, 伊藤 潔, 鈴木誠道: 非線形擬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).
- 7) Akers, S.B.: Binary Decision Diagrams, *IEEE Trans. Comput.*, Vol.C-27, No.6, pp.509-516 (1978).
- 8) Bryant, R.E.: Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Trans. Comput.*, Vol.C-35, No.8, pp.677-691 (1986).
- 9) 湊 真一, 石浦菜岐佐, 矢島脩三: 論理関数の共有二分決定グラフによる表現とその効率的処理手法, 情報処理学会論文誌, Vol.32, No.1, pp.77-85 (1990).
- 10) Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, *Proc. 30th ACM/IEEE Design Automation Conference*, pp.272-277 (1993).
- 11) 奥乃 博: 二分決定グラフによる探索型組合せ問題の解法での組合せ的爆発抑制法, 情報処理学会論文誌, Vol.35, No.5, pp.739-753 (1994).
- 12) 湊 真一: BEM-II: 二分決定グラフを用いた算術論理式計算プログラム, 電子情報通信学会技術報告, Vol.COMP92-75, pp.15-22 (1993).
- 13) Coudert, O., Madre, J.C. and Fraisse, H.: A New Viewpoint on Two-Level Logic Minimization, *Proc. 30th ACM/IEEE Design Automation Conference*, pp.625-630 (1993).
- 14) Markowsky, G. and Wesley, M.A.: Fleshing Out Wire Frames, Technical Report 5, IBM J. Res. Dev. (1980).

- 15) 斎藤逸郎, 正木寛人, 石塚 満, 奥乃 博: 二分決定グラフによる三面図理解システムの機能拡張, 第 50 回情報処理学会全国大会論文集 (分冊 2), pp.409-410 (1995).

(平成 8 年 1 月 19 日受付)

(平成 8 年 9 月 12 日採録)



正木 寛人

1970 年生。1993 年東京大学工学部電子工学科卒業。1995 年同大学院修士課程修了。同年国際電信電話(株)入社。現在に至る。ネットワークサービスの研究に従事。電子情報通信学会会員。



斎藤 逸郎 (学生会員)

1972 年生。1995 年東京大学工学部電子情報工学科卒業。現在、同大学院修士課程に在学。WWW 上のエージェントの研究に従事。



石塚 満 (正会員)

1948 年生。1971 年東京大学工学部電子工学科卒業。1976 年同大学院博士課程修了。工学博士。1976 年 NTT 横須賀研究所。1978 年東京大学生産技術研究所助教授。1980 年より 1 年半の間、米国パデュー大学客員準教授。1992 年東京大学生産技術研究所教授。1992 年東京大学工学部電子情報工学科教授。現在に至る。人工知能、知識システム、画像処理・生成、擬人化エージェントシステム、知的ヒューマンインタフェースの研究に従事。著書:『知識の表現と高速推論』(丸善, 1996) など。人工知能学会, 電子情報通信学会, 画像電子学会, IEEE, AAAI 各会員。

**奥乃 博 (正会員)**

1950年生。1972年東京大学教養学部基礎科学科卒業。同年電電公社入社。1986～1988年スタンフォード大学客員研究員。1992～1993年東京大学工学部電子工学科客員助教授。現在、NTT基礎研究所勤務。主幹研究員。工学博士。推論機構、音環境理解の研究に従事。著書：『インターネット活用術』（岩波書店、1996）、『知的プログラミング』（共著、オーム社、1993）、『マルチエージェントと協調計算 III』（編、近代科学社、1994）など。本学会英文図書委員、人工知能学会、日本認知科学会、日本ソフトウェア科学会、ACM、AAAI 各会員。
