

二分決定グラフを用いた三面図の効率的な理解

正木 寛人† 斎藤 逸郎† 石塚 満† 奥乃 博†

†東京大学工学部電子情報工学科 †NTT 基礎研究所

masaki@miv.t.u-tokyo.ac.jp

三面図は、3次元モデルを表現するために盛んに使用されているが、それから3次元モデルを復元する時にはしばしば複数の解釈が生ずるといった曖昧性の問題があった。三面図に関連したこれまでの研究の多くはその効率的な復元に焦点を当てており、曖昧性の解消はヒューリスティクスに頼ってきた。我々はこの問題に対し、新たに面図を追加することで曖昧性を除去するだけでなく、さらに三面図をユニークな表現法にすることができると考え、そのための手法を本稿で提案する。具体的には、複数解釈できる3次元モデルの中からユーザが指定した所望のモデルだけを表現するために必須な物体要素を同定する。この内部処理には、復元過程におけるデータ保持を効率よく行うために、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を使用する。また、試作したTOVIN (Three Orthographic Views INterpreter) についても報告する。

Efficient Understanding of Three Orthographic Views Using Binary Decision Diagram

Hiroto Masaki† Itsuro Saito† Mitsuru Ishizuka† Hiroshi G. Okuno†

†Dept. of Info. & Commun. Eng., Univ. of Tokyo

†NTT Basic Research Laboratories

Although a set of three orthographic views is one of the most common representations of three-dimensional objects, it has been pointed out that it may have ambiguities that it allows multiple interpretations of solids to be reconstructed. In this paper, we present a new method for ambiguity elimination by adding a new mapping plane and report TOVIN (Three Orthographic Views INterpreter). The processing of TOVIN consists of two parts. First, it encodes three orthographic views in pseudo boolean functions by using the rules of reconstruction by Sasaki et al. By using Binary Decision Diagrams, all the possible interpretations are enumerated. Next, TOVIN requests the user to specify one from them. Then, it searches a set of critical object elements (edges or faces) to eliminate ambiguities. This process uses a set of combinations and employs Zero-Suppressed BDDs as a compact representation of sets. TOVIN runs quite efficiently thanks to BDDs and Zero-Suppressed BDDs.

1 はじめに

三面図は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で歴史的に広く用いられてきた。三面図に関連したこれまでの研究では、対応する3次元モデルを効率的に復元する研究が主流であった。また、三面図は本来3次元空間に存在する物体を2次元平面上へ投影した表現であるため、三面図が曖昧になる可能性が存在し、そのために対応する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た。とくに、三面図は製図をする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おおようにしてあるために、復元すべき3次元モデルの解釈に曖昧性が残り易くなる。

このような三面図の曖昧性の問題に対し、ヒューリスティクスを用いて対応する3次元モデルを一意的に生成する研究がさかに行なわれてきた。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が提案されている。一方、我々はこれまでに三面図の曖昧性の除去に物体表現の観点から取り組んできた [12]。

3次元物体の三面図以外の表現法として、位相的に異なるすべての可能な面図を節点とし、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]。我々は、*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、そして復元過程を効率的にすることを考えてきた。*aspect graph* を認識のための表現法と見なすと、三面図は3次元モデル生成のための表現にとらえることができよう。我々の研究は、生成のための表現に完全性を付与するための一方法と考えられる。本稿では、論理関数のコンパクトな表現である二分決定グラフを復元過程に適用することと、論理式で表された復元結果に対して組合せ集合演算を施すことで、追加面図において注目すべき物体要素を同定することを提案する。後者は、機械製図などの分野で補助面図を加える際に、どの方向から見ればよいかに対する指針となろう。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、曖昧性を除去することができるという利点を有していると考えられる。なお本稿で扱う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。

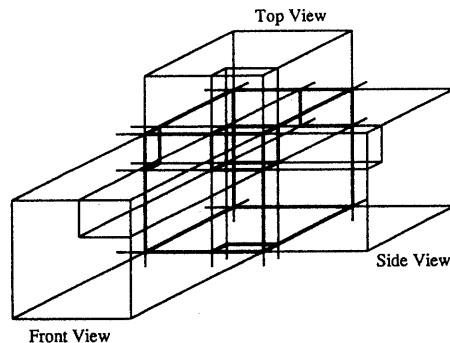
本稿の構成は以下のとおりである。第2節で三面図から対応する3次元モデルを復元する手法についてその概要を記す。第3節では擬似ブール代数で定

式化された復元規則を紹介し、第4節でその復元規則の表現に用いる二分決定グラフについて説明する。そして、第5節で探索過程と曖昧性除去に対して二分決定グラフを用いることを提案し、最後に第6節でまとめを行う。

2 三面図の解釈と対応物体の復元

三面図を解釈し、対応する3次元モデルを復元する方法にはさまざまなものがあるが、本稿で参考にしているものは以下のようにまとめることができる。

候補物体要素列挙 三面図から候補頂点と候補稜線を列挙する。この中には、三面図には合致するものの所望の物体には存在しない虚物体要素や、虚ではないが同時には存在し得ないはずの非両立物体要素が含まれる可能性がある。候補稜線が得られた後、候補面を列挙する。本稿では、候補物体要素列挙の方法として、出澤の方法を基にした伊藤の方法 [1] を用いた。図1にその概観を示す。



端点間の対応、端点と線分の対応から候補頂点を求める。候補頂点間に候補稜線(三面図へ投影可能なもののみ)を張る(太線)。可能な候補面を求める。

図1: 候補物体要素列挙(文献[2]より引用)

真の物体要素の探索 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、真の物体要素を判定し、それらを合成

する。探索プログラムは研究者ごとに微妙に違っているものの、虚物体要素の除去、非面立物体要素の選択、三面図との適合のチェックなどを行っている点では大体共通しているものと考えられる [1]。本稿で参考にする探索過程は佐々木らの方法 [2] であり、詳細は第 3 節に記す。

3 擬似ブール代数による復元規則の定式化

佐々木らは、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 を、存在しない場合に 0 をとる)、上記の探索プログラムを代数的に定式化したものに従って連立擬似ブール代数式を構築し、これを汎用の代数解法 (Hammer の解法) で解いて、真のサーフェスモデルを求めるという方法を提案した。以下に、それらの規則の意味と擬似ブール代数による定式化を示す。なお、下記の記号を用いることとする。

- s : 三面図中の線分
- $\langle f_i, f_j \rangle$: coplanar な二つの面の組
- $E(s)$: ある面図上の線分 s として投影される候補稜線群
- $F(s)$: 線分 s を内部に含む面として投影される候補面群
- $E_{pre}(s, e)$: $E(s)$ 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
- $F_{pre}(s, e)$: $E_{pre}(s, e)$ 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
- $F_{post}(s, e)$: 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

規則 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 でない二つの面が真である。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、すべての面が偽であるか、coplanar な二つの面が真である」

すべての e に対して、

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

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

規則 2 : 「交線となる稜線なしで交差する候補面群 F_x のうち高々一つが真である」
すべての $F_x \in F_{cross}$ に対して、

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

ただし、 F_{cross} は F_x の集合とする。

規則 3 : 「ある面図上の同じ線分から得られた候補稜線はそのうち 1 本以上が真である」
すべての $s \in S$ に対して、

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

規則 4a: 実線規則 : 「ある面図上の実線の線分 (この集合を S_{real} と表記する) 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) 稜線について、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すべての面は偽である」
すべての $s \in S_{real}$ に対して、 $E_{pre}(s, e) = \emptyset$ である $e \in E(s)$ に対して、

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

規則 4b: 実線規則 : 「ある面図上の実線の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、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、

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

規則 5: 破線規則 : 「ある面図上の破線の線分(この集合を S_{hidden} と表記する) 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一つ以上の面が真である」

すべての $s \in S_{\text{hidden}}$ に対して,

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

佐々木らの方針は, 位相的・幾何学的規則を何らかのスケジューリングの下で逐次的に適用していく従来の方法とは異なり, 立式した連立式を解くことにより, さまざまな探索を分離しないで行うところが特徴となっている. 図 2 に図 6 (a) を例としたコーディングの一部を示す.

```
# Regulation1
C1 = (f1 + f5 == 2 * e1)
...
# Regulation2
D1 = (f8 + f11 <= 1)
...
# Regulation3
E1 = (e5 + e18 >= 1)
...
# Total Constraint
F = C1 & C2 & C3 & ... & D1 & D2 & D3 & ...
# Number of Solutions
print /count F
# Solutions in Sum-of-products Form
print F
```

図 2: 擬似ブール代数式によるコーディングの例

4 二分決定グラフ

二分決定グラフ (Binary Decision Diagram, 以下 BDD と略記する) は 1978 年に Akers が提案した論理関数のコンパクトな表現法であり [5], 1986 年に Bryant がその効率的な演算手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 [6]. 図 3 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言えよう. ある

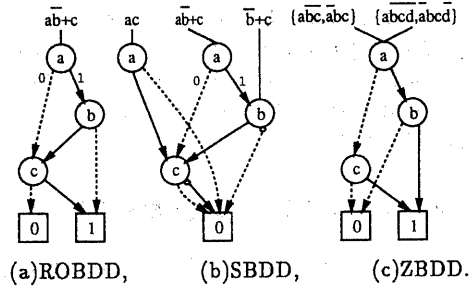


図 3: 二分決定グラフ

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カノニカル形になることが知られており, これを ROBDD と呼ぶ. また, さらには複数の ROBDD 間で等価なサブグラフを共有した SBDD を提案しており, 記憶効率の点などで優れている [7].

さらに, 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もつことに着目し, 1 枝が直接 false 節点へ向かっているような節点を削除するという既約規則を導入した Zero-Suppressed BDD (以下 ZBDD と略記する) も提案されている [10]. ZBDD は BDD と規約化の規則の点では異なるものの, グラフがカノニカル形になる点では共通しており, また希薄な集合を扱う場合には BDD よりもコンパクトになることが知られている.

BDD の主な特徴は以下の 3 つにまとめられる.

- (1) 入力変数の順序を固定したときにその論理関数の標準形となる.
- (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はほぼ比例する時間で行える.

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, などが容易に行える. 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得ないと考えられるため, BDD を効果的に応用する際の重要な指針となっている [8].

5 BDD を用いた三面図の効率的理解

本節では、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4は本稿で提案するシステムの構成を示している。

5.1 BDD の適用による復元過程の効率化

従来、連立擬似ブール代数式の処理の部分には Hammer の解法が用いられていた。Hammer の解法についての詳細は文献 [4] に譲るが、以下にその概要を示す。まず、1本の式(方程式・不等式)については、以下ようになる。

(1) 以下の変換を行い、標準形にする。

- 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

(2) 判定ステップ:

予め解析されている、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判定する。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)新たな式を得る。

(3) 新たに得られた式に対して、上記の判定ステップを繰り返す。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るが、値の固定具合で3通りに分類されており、それらは以下のようなになる。

- **Determinate**
解なし、すべての変数が0か1に固定可能、変数の一部が0か1かに固定可能。
- **Partially determinate**
条件分岐が可能。その際、変数の一部が0に固定可能。
- **Undeterminate**
ある変数について、0の場合と1の場合の2本の部分式に分割。

これらはこの順に優先度をもっており、例えばある式が解なしの場合には、残りの式を吟味する必要がなく

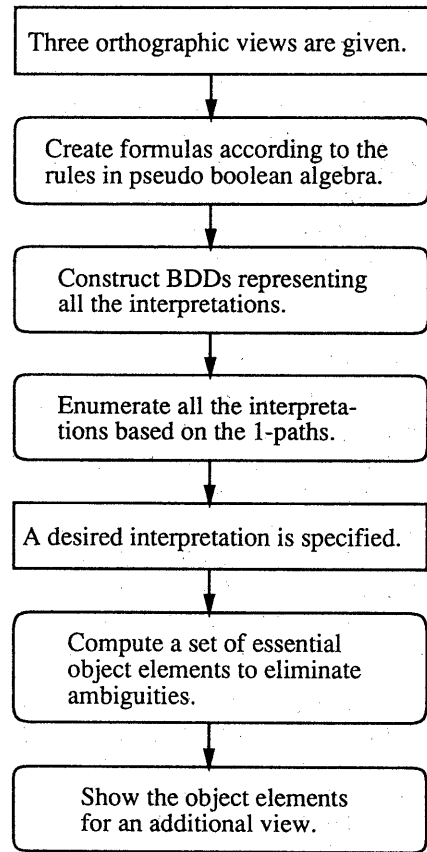


図4: 三面図理解システムの構成

なる。さらに非線形項が現れた場合には、その項を新たな変数で置き換え、線形にする。非線形項を置き換える必要があるため、結果的に連立式の本数が増大する。

我々はこの Hammer の解法による内部処理を 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-II [9] を BDD による処理形として用いた。これは、佐々木らの提案した擬似ブール代数による復元規則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 との相性が良いと考えられるためである。両者の違いは、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与えることがそのまま式を解くことに直結するのに対し、BDDによる処理では連立擬似ブール代数式を与えることはその時点までのすべての式(制約)を表すBDDを構築することにしかならないという点である。したがって、連立式全体を与え終わった時点でのBDDは候補物体要素の状態(存在/非存在)の可能な組合せすべてを表現しており、例えば全解を求めたい場合にはroot節点からtrue節点(図3の①)へのすべてのパスを求めるコマンドをBEM-IIに与えるなどの、BDDを操作する演算が後に続くことになる。

5.2 三面図の曖昧性の除去

三面図は3次元物体を2次元平面上へ投影した表現法であるため、それから対応する3次元モデルを復元するときに生ずる解釈の曖昧性を解消しなければならない場合が多い。我々は、ヒューリスティクスを用いるのではなく、aspect graph [3]が表現力の点で完全であることに着目し、曖昧な三面図に対して面図を追加することで、この曖昧性を除去できるものと考えた。提案する手法の概要を図5に示す。本手法を直観的にとらえるならば以下のようになる。

三面図が所望の3次元モデルに一意に対応するためには、その他の3次元モデルとの違いに注目しなくてはならず、そのためには、最終的な解において真偽値が反転しているような物体要素(以下注目要素と呼ぶ)を中心に面図が適当である。

本手法では、情報を2種類の方法で表現している。まず、論理式を用いて三面図を表現し、これいんより3次元モデルの可能な解釈をすべて論理式で保持する。論理式の効率的な表現法としてBDDを使用する。次に、注目要素を求める時には、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Zero-Suppressed BDD (ZBDD)を使用する。集合を表現するのにBDDではなくZBDDを用いた理由は、通常の論理式や算術論理式では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が挙げられる*。集合によるコーディングとし

*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[11]。BDDで論理関数を表現し、そのBDDの1-pathを辿りながら、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。

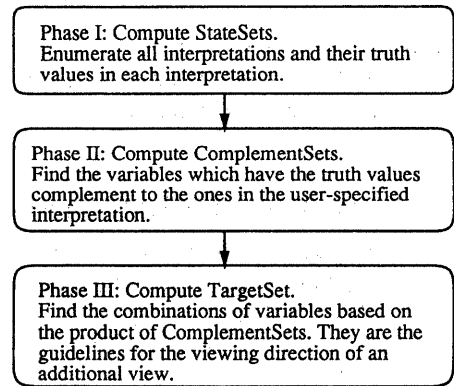


図5: 曖昧性除去の手法

て、具体的な実装時にはUCBで開発されたZBDDのライブラリパッケージを用いた。以降、集合の範囲では変数を単にリテラルと呼ぶこともあり、また理解の簡単化のため例(図6(a))を交えて説明を行う。

- (1) まず、三面図を表すBDDに対して全解を求める操作を施す。これは、全制約(連立式全体)をFとした場合、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cubeの論理和の形で出力される。同時に、print /count Fで何個の解釈(3次元モデル)が得られたかを求めておき、その個数をn+1であるとする。以下はその出力例の一部である。

```

count
  5
sop
e1 & e2 & e3 & !e4 & e5 & e6 & !e7 & e8 & e9 &
!e10 & e11 & !e12 & e13 & e14 & e15 & !e16 &
e17 & e18 & !e19 & e20 & e21 & e22 & !e23 & !e24
& f1 & f2 & f3 & f4 & f5 & f6 & !f7 & !f8 & f9 &
!f10 & !f11 & f12 & f13 & !f14 & !f15 & !f16 &
!f17 & f18 & f19 & f20
| e1 & e2 & e3 & !e4 & !e5 & e6 & e7 & !e8 & ...
  
```

- (2) 次に、各解釈における変数の真偽値をBDD処理の外側で求める。表1は、5つの解釈における各稜線の真値表である。
- (3) 真値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

表 1: 復元結果に対する真理値表

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

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計算は、三面図を表す BDD 構築の最終部で、

e.g.) $x \oplus x' = 1$ (\oplus : 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理値に応じて反転した値で新変数を解釈してもよい。本ステップは、後で用いる集合の引算の際に偽であるような変数が演算後に残ることを期待するために置いた。

- (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 (状態集合と呼ぶ) を作成する。例えば解釈 #1 に対して、以下のような S_1 が作成される。

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

- (5) ユーザが解釈 # j を指定したとする。指定された解釈とその他の各解釈との間で、候補物体要素の

状態 (真理値) が反転しているものから構成される集合 C_i (反転要素集合と呼ぶ) を求める。本ステップは、以下のようにして実現する。

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

例えば、ユーザが解釈 #1 を指定したときの解釈 #4 に対する C_4 は以下のようなになる。

e.g.) $C_4 = \{e1', e4, e5', e6', e12, e19, e23\}$

- (6) ユーザの指定した解釈 (# j) はその他のすべての解釈と異なることが必要とされるので、 n 個の反転要素集合の直積をとり、指定された解釈が各リテラルについてどのように異なっている必要があるのかを計算する。得られた集合を注目要素集合 R と呼ぶ (注目要素集合は変数の集合に対して組合せ集合となっている)。

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

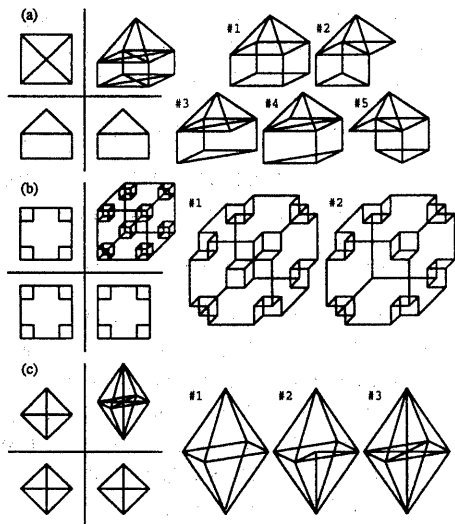
例えば、ユーザが解釈 #1 を指定したときの注目要素集合は以下のようなになる。

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

- (7) 注目要素集合の各要素 (cube) のうち、選択されたリテラルを最小個数含むものを選び出す。これは、注目すべき物体要素として多数の物体要素を見るよりも、できるだけ少ない個数の物体要素を見る方がよいためである。実現方法としては、ZBDD ライブラリパッケージでサポートされている `foreach_zbdd_cube{}` マクロを用いて、注目要素集合の各 cube で選択されたリテラル数をチェックし、現在保持している最小値と同じであれば最小注目要素集合に入れ、それよりも少なければ新たに最小注目要素集合を作成し直す、というものである。例としては、ユーザが解釈 #5 を指定したときの最小注目要素集合は、 $\{\{e3\}\}$ になり、ユーザが解釈 #1 を指定したときの最小注目要素集合は以下のようなになる。 $\{\{e1', e8'\}, \{e2', e5'\}, \dots, \{e19, e24\}\}$

5.3 試作システムの実装

上記の考え方に基づき、Sun Sparc 10 上に試作システム TOVIN (Three Orthographic Views INterpreter) を実装した。上述したとおり、全解を求めるま



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 6: 三面図および対応物体の例 ((a), (b) は文献 [2] より引用)

では擬似ブール代数に相性のよい BEM-II を使用し、そこで一度 BDD 処理の外側に出る。中間処理を行った後、集合を扱うのに都合のよい ZBDD のライブラリパッケージを用いて、我々の提案した手法を C 言語でプログラミングした。以下にその使用例を紹介する。

図 6 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 /usr/bin/time コマンドを用いて測定した。表中、#var は候補要素物体列挙の結果得られた候補稜線と

表 2: BEM-II による処理時間 (単位は [sec])

例	#var	#eqn	#solid	#node	#time
(a)	44	103	5	194(672)	3.1
(b)	162	432	2	265(2038)	54.4
(c)	38	84	13	261(662)	3.1

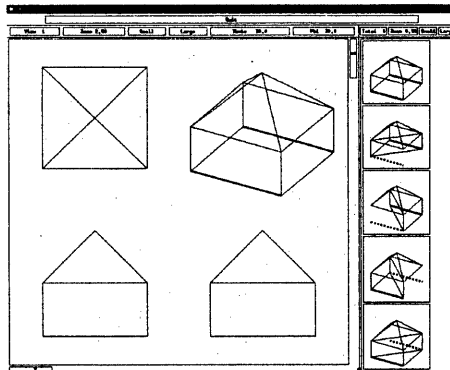


図 7: 試作システム TOVIN

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 は復元される対応モデルの数を、#time は実行時間のうちの user time を示している。

文献 [2] によると、例 (a) では 69[sec] という処理時間が報告されているが、これは規則 3 までしか立式しておらず、また IBM 系の PC 上で実装した結果であることから単純には評価できないため、Hammer の解法の同一計算機環境下での実装を試みた。しかし、文献 [4] による考え方をダイレクトに実装した場合、現実的な時間で処理不可能であった。これより、Hammer の解法には高速化のためのさまざまなチューンアップが必要であると思われるが、本稿ではこれを主眼としていないので、Hammer の解法との単純比較を行うことはしない。ただ、メモリ資源が限られている場合、BDD を適用する手法の方がコンパクトに問題を表現できるので、大規模な問題にも現実的に対応できると言えよう。また、佐々木らのグループは評価のグラフの横軸に「制約式の評価回数×式の本数」なる量を置いて、対象物体の複雑度としているが、我々は BDD の 1 パスを辿ることで解を得るため、BDD の節点数をそのような複雑度として採用することができよう。表 2 の #node は、そのような観点から置いたものである。括弧外は全解釈を表す BDD そのものの節点数であり、括弧内は処理系全体でのそれである。前者が探索の時間に比例する量とすれば、後者は問題表現 (BDD 構築) に要する時間に比例する量と言える。

次に、曖昧性除去に関する様子を図 7 に示す。図 7 は、図 6 (a) を例として TOVIN を起動したものである。左側のメインパネルに三面図とユーザの所望する解釈 (3次元モデル) が、また右側のサブパネルにすべての 3次元モデルが表示されている。メインパネルの左下と右上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。この図は 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、上述した最小注目要素集合の要素のうち、1 番目の要素である $\{e1', e8'\}$ が太く表示されている。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。

6 おわりに

本稿で、三面図の効率的な復元やその曖昧性除去に関して二分決定グラフ (BDD) を用いることを提案した。BDD を用いることのメリットは、以下のようによままとめられる。

- ある計算機環境下において、従来では不可能だったような大規模な問題も現実的に扱うことができる。
- 論理式でコンパクトに表現されている結果を用いてさらなる処理を行うので、追加面図において注目すべき物体要素を効率的に判定することができる。

筆者は、BDD が特定のアプリケーションに依存したものではないにも関わらず、本稿等に見られるようにその威力を多方面で発揮していることを付け加えておきたい。

最後に、BEM-II の使用をお許し頂いた NTT LSI 研究所 湊真一氏、三面図の擬似ブール代数展開の情報を頂いた上智大学 伊藤潔先生、ZBDD のライブラリパッケージを使用させて頂いた UCB の久木元裕治氏に感謝致します。

参考文献

- [1] 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, pp.1095-1106, 1990.
- [2] 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, 情報処理論文誌, pp.699-708, 1989.
- [3] K.W.Bowyer, C.R.Dyer: Aspect Graphs: An Introduction and Survey of Recent Results, *Intl. J. of Imaging Systems and Tech.*, pp.315-328, 1990.
- [4] P.L.Hammer, S.Rudeanu: Boolean Methods in Operations Research and Related Areas, *Springer-Verlag New York*, 1968.
- [5] S.B.Akers: Binary Decision Diagrams, *IEEE Trans. Comput.*, pp.509-516, 1978.
- [6] R.E.Bryant: Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Trans. Comput.*, pp.677-691, 1986.
- [7] S.Minato, N.Ishiura, S.Yajima: Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation, *Proc. of 27th ACM/IEEE DAC*, pp.52-57, 1990.
- [8] 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, 情報処理論文誌, Vol.35, No.5, pp.739-753, 1994.
- [9] 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, 信学技法, COMP92-75, 1993.
- [10] Shin-ichi Minato: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, *Proc. of 30th ACM/IEEE DAC*, pp.272-277, 1993.
- [11] O.Coudert, J.C.Madre, and H.Fraisse: A New Viewpoint on Two-Level Logic Minimization, *Proc. of 30th ACM/IEEE DAC*, pp.625-630, 1993.
- [12] 正木寛人, 奥乃博, 石塚満: 二分決定グラフによる三面図からの 3D モデルの解釈, 第 8 回人工知能学会 全大, pp.693-696, 1994.