

制約充足に基づく三面図理解システム

山田 弘展^{*} 若林 明子^{**} 内野 寛治^{*} 狩野 均^{*} 西原 清一^{*}

^{*} 筑波大学 電子・情報工学系

^{**} 日本電気(株) 第二パーソナルC&C事業本部

本稿では三面図理解問題の新たな手法を提案する。この手法は、三面図からルールベース化した復元知識を用いて制約充足問題(CSP)を生成し、得られたCSPを制約ソルバーで解くことによって3次元物体を復元するという2段構成になっている。CSPを生成するために、「3次元物体の存在公理」に基づく3次元物体を復元するための知識から幾何学的知識を導出した。本手法は、復元知識をルールベース化し、図面理解問題をCSPによって等価表現するので、(1)ルールベース上での復元知識の追加削除などのチューニングおよび有効な復元知識の解明などができる、(2)システムの構造が簡明なため他者に理解し易い、(3)CSPの並列処理など汎用的な高速化手法が適用できる、という特徴がある。

Understanding System for Three Orthographic Views Using Constraint Satisfaction

Hironobu Yamada^{*} Akiko Wakabayashi^{**} Kanji Uchino^{*} Hitoshi Kanoh^{*} Seiichi Nishihara^{*}

^{*} Institute of Information Sciences and Electronics
University of Tsukuba

^{**} 2nd Personal C&C Operations Unit
NEC Corporaion

We propose a new approach to understanding engineering drawings, which is the problem to recover the 3D data of the legal solid objects from the 2D data of a given three-view orthographic drawing. Our method essentially consists of two major processes: translation of the original 2D data to a constraint satisfaction problem (CSP) by using a constraint knowledge base, and solving the CSP to get the final 3D data by using a general CSP solver. To realize the translation process, we develop geometrical rules derived from the restoration knowledge based on the axiom of existence of 3D objects in the usual Euclidean space. Since a CSP, an intermediate expression of an original drawing, is actually a knowledge base independent from both the given 2D data and the final 3D data, our approach enables us (1) to tune up the knowledge base easily, (2) to customize the restoration system with high modularity, and (3) to make use of existing general purpose constraint solvers adopting many kinds of efficient search techniques as well.

1 はじめに

製図なかでも三面図は、工業製品の形状設計に広く用いられてきた。この三面図を計算機によって理解し、自動的に3次元データを復元する手法の開発は、知的CADへ至る重要な技術として種々研究されている [出澤72、西原87、伊藤90、西田91、井上93]。三面図理解の手続きの第一歩は、通常、まず図面を構成している2次元要素すなわち点や線から、3次元要素すなわち頂点や稜線を生成し、さらに面の全ての候補を復元することから始まる。この前処理で得られた候補面集合を用いて、最終的な3次元立体を構成するのであるが、この構成法に関していくつかの方法が提案されている。例えば、ワイヤフレームの位相的性質を用いる方法 [出澤72]、局所的制約条件を用いる方法 [西原87]、擬似ブール代数式を用いる方法 [佐々木89] をはじめ多くの報告がある。

筆者らは先に、「3次元物体の存在公理」を基にして、候補面の組合せ規則を複数個導出し、これらを復元知識として組み込んだ理解システムを開発した [井上93]。これは、各候補面についてそれが最終的に復元立体に含まれるか否かを‘真偽値’として順次決定して行く探索手続きとなっており、バックトラックの発生率を低く抑えることにより高速処理が達成された。しかし、復元知識がプログラム内に手続きとして実現されているため、システムの改良や変更が困難であった。

本論文では、三面図理解を制約充足問題 (CSP, constraint satisfaction problem) [西原90] として表現し実行する手法を提案する。このような試みはいくつかあるが [Haralick79、西原87、伊藤93]、本論文では、先の復元知識をルールベース化し、それを用いて具体的なCSPを生成し、得られたCSPを制約ソルバーによって解くという2段構成の方法を提案する。従来システムおよび本システムの処理の概略を図1に示す。本手法は復元知識をルールベース化し、図面理解問題をCSPによって等価表現するので、(1)ルールベース上での復元知識の追加削除などのチューニングおよび有効な復元知識の解明などができる、(2)システムの構造が簡明なため他者に理解し易い、(3)CSPの並列処理など汎用的な高速化手法が適用できる、などの効果が得られる。

以下では、まず三面図理解問題およびCSPの概説を行なう。次に、本システムで取り入れた、復

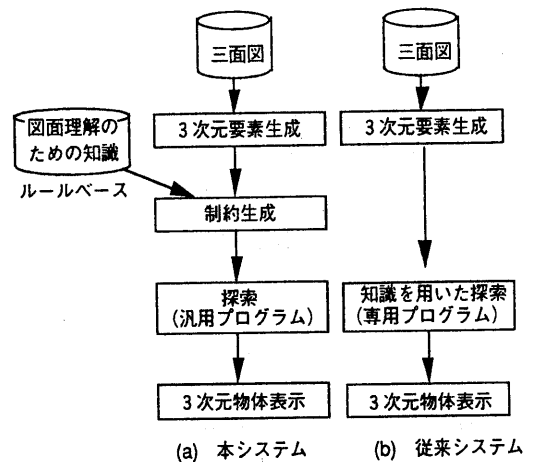


図1 従来システムおよび本システムの処理の流れ

元知識のルール化、生成する制約の記述方法および参照方法、解探索の高速化手法について述べる。最後に、実際に開発したシステムによる実験を行ない、従来システムとの比較、検討を行なう。

2 諸定義

2.1 三面図理解に関する諸定義

本システムの入力は、平面図、正面図、右側面図が第三角法によって描かれた三面図を表わすデータであり、直線と円弧から成るものとする。また、扱う3次元物体は平面と2次代数曲面から構成される。三面図ならびに三面図から復元された3次元物体の例を図2に示す。

三面図は、実線及び破線からなる‘線分’によって表わされている。本システムにおいて線分は物体の外形を表わす‘外形線’、面によって隠された外形を表わす‘隠れ線’、面種の異なる面が滑らかに接している部分や曲面を面図に投影したときにできるシルエット部分を表わす‘補助線’の3種類に分類されており、それぞれ実線、点線、一点鎖線によって三面図上では表現している。各線分の両端点を‘点’、線分の閉じた並びによって囲まれた部分を‘領域’という。領域のうち、分割が不可能なものを特に‘単純領域’という。

三面図理解問題とは、入力された点や線分のデータから頂点、稜線、面といった3次元要素を復元し、それらの要素を組み合わせで意図する3次元物体を構成する問題である。頂点、稜線、面は、それぞれ点、線分/点、領域/線分として2次元

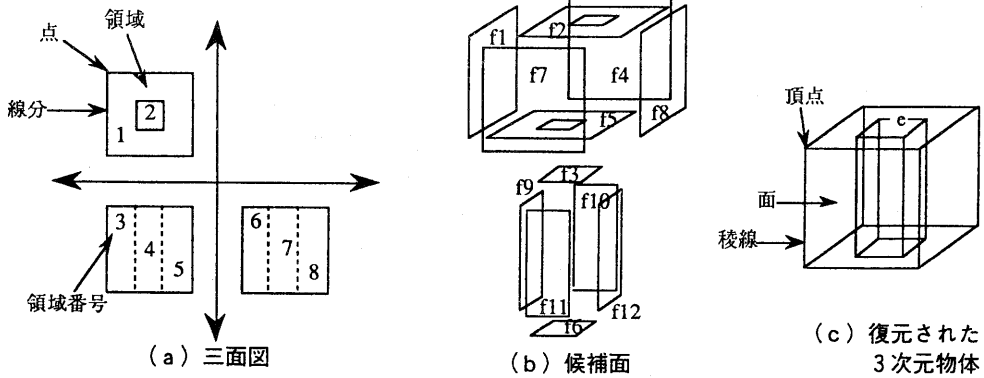


図2 三面図と復元された3次元物体の例

投影図面に変換されてしまうので復元過程には一般的にあいまい性が存在する。つまり、三面図データから復元される3次元要素は3次元物体を構築するのに必要な要素を全て含んでいるが、それら全ての要素が3次元物体を構築するわけではない。そこで本論文では、3次元要素の復元過程で復元された全ての面を‘候補面’と呼び、三面図理解問題を「復元された全ての候補面の中から意図する3次元物体を構成する面を選び出す問題」として定義する。図2(a)の三面図の例では、まず図2(b)の候補面 (f_1, \dots, f_{12}) が復元される。次に、これら候補面の中から図2(c)に示すように $f_1, f_2, f_4, f_5, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}$ が3次元物体を構成する面として選ばれる。

2.2 制約充足問題 (CSP) に関する諸定義

制約充足問題 (CSP) とは、問題全体を問題の構成要素間に成立する局所的な制約の集合として定義し、与えられた全ての制約を同時に満足するような局所解の組み合わせを求めるような問題の総称である。本論文では、文献 [Haralick 79] に沿って CSP を4つ組 (U, L, T, R) で定義する。Uはユニットの集合で、各要素は対象とする問題の構成要素に対応する。Lはラベルの集合で、各要素はユニットに与えるべき解釈や値の候補を表す。そしてTとRで具体的な制約を表わす。Tは相互に制約し合っているユニットの組、RはTに含まれる各ユニット組に対する可能な部分解を表わす。

図2の三面図の例をCSPとして記述する場合、ユニットは $f_1 \sim f_{12}$ の候補面、ラベルは候補面が意図する3次元物体を構成するのに必要であるか、不必要であるかを表わすために真(T)と偽(F)の二

$$U = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}\}$$

$$L = \{T, F\}$$

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21}, t_{22}, t_{23}, t_{24}\}$$

$$\begin{aligned} t_1 &= (f_{10}, f_{12}), t_2 = (f_5, f_6, f_{12}), t_3 = (f_2, f_3, f_{12}), t_4 = (f_{11}, f_{12}), \\ t_5 &= (f_2, f_3, f_{11}), t_6 = (f_5, f_6, f_{11}), t_7 = (f_9, f_{11}), t_8 = (f_2, f_3, f_{10}), \\ t_9 &= (f_2, f_3, f_9), t_{10} = (f_5, f_6, f_{10}), t_{11} = (f_5, f_6, f_9), t_{12} = (f_9, f_{10}), \\ t_{13} &= (f_4, f_8), t_{14} = (f_5, f_8), t_{15} = (f_2, f_8), t_{16} = (f_7, f_8), \\ t_{17} &= (f_5, f_7), t_{18} = (f_2, f_7), t_{19} = (f_1, f_7), t_{20} = (f_4, f_5), \\ t_{21} &= (f_1, f_5), t_{22} = (f_2, f_4), t_{23} = (f_1, f_2), t_{24} = (f_1, f_4). \end{aligned}$$

$$R = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}, R_{20}, R_{21}, R_{22}, R_{23}, R_{24}\}$$

$$\begin{aligned} R_2 = R_3 = R_5 = \dots = R_{11} &= \{(F, F, F)(F, T, T)(T, F, T)(T, T, F)\} \\ R_1 = R_4 = R_{12} = \dots = R_{24} &= \{(F, F)(T, T)\}. \end{aligned}$$

図3 CSPの例

値を取るものとする。

稜線接続などを考慮しない一般的な多面体では、1つの稜には2つの面が接続している。すなわち、ある候補稜線が存在するならばその稜線に接続する2つの候補面が存在しなければならない。このような必ず成り立つべき幾何学的性質は「稜線を共有する面数は0または2である」という知識としてとらえることができる。この知識から具体的なTとRを生成する。例えば図2(c)中の稜線eを共有する候補面は f_2, f_3, f_{10} であるので $t_8 = (f_2, f_3, f_{10})$ で表わされ、3枚の候補面の内で「0枚もしくは2枚が真となる」はずなので対応する部分解は

$$R_8 = \{(F, F, F), (F, T, T), (T, F, T), (T, T, F)\}$$

となる。(F, F, F)は稜線e自体が存在しないことを表わす。全ての候補稜線について同じようにTとRを生成した場合のCSPを図3に示す。

3 制約充足パラダイムによる 理解手法

3.1 三面図理解のための幾何学的知識

本論文で扱う復元知識は

「3次元空間に一つ以上の物体が存在するための必要十分条件は、任意の閉曲線が物体の表面と0または偶数回交差することである」

という「3次元物体の存在公理」を多面体に適用した幾何学的な知識である。稜線の周りを一周する閉曲線、領域に対応する面を貫くような閉曲線を考慮すると次の2つの基本知識が導かれる。

知識1 任意の稜線を共有する候補面(f_1, \dots, f_n)

のうち真となる面の数は0または2である。

知識2 三面図上のある領域に対応する候補面

(f_1, \dots, f_n)のうち真となる面の数は偶数枚である。

ただし、その領域が輪郭を境界線分として含む場合は2以上の偶数枚である。

このような幾何学的知識は、著者らが以前開発したシステム [井上93] において候補面の真偽を

決定する手続きとして用いられていた。本論文ではまずこれら知識を分類・整理し、各知識の関係を図4のようにまとめた。

図4において、知識1は稜線に関するもの、知識2.1および知識2.2は領域に関するもの、また、知識3は交差している面に関するものである。特に知識1は、外形線(知識1.1)、隠れ線(知識1.2)、補助線(知識1.3)ならびに稜線を共有する面が2面の場合の知識(知識1.4)を包含している。3.2節では、このような知識の分類・包含関係に基づいたルールベースの構成手法について述べる。

3.2 幾何学的知識のルールベース化

(1) ルールベース

前節で述べた幾何学的知識は、ある局所的な条件を満たす候補面組に対して取りうる真偽値の組み合わせを表わしている。そこで本論文では以下の書式を用いて、幾何学的知識を具体的な制約条件を生成するためのルールとして記述する。

<一般形>

```
rule ID no_of_values = N
  if Pred1, Pred2, ..., Predn
  then (R)
```

注: ID:知識ID, N: ruleが適用される候補面数, 0は‘不定’を示す。
Pred_i:述語, R:部分解

ここで注意しなければならないのは、一般のプロダクションシステムではif部の述語は真偽値を返しているのに対し、本システムのif部では各述語を満たすユニット組の集合を返している点である。これはプロダクションシステムがルールを用いた‘推論’によって解を求めているのに対し、本手法ではルールを用いてまず‘制約の生成’を行い、その後CSPの解探索によって解を求めるという違いに因るものである。本システムでは、上記の書式により記述された、図4の各知識に対応している11のルールを用いている。

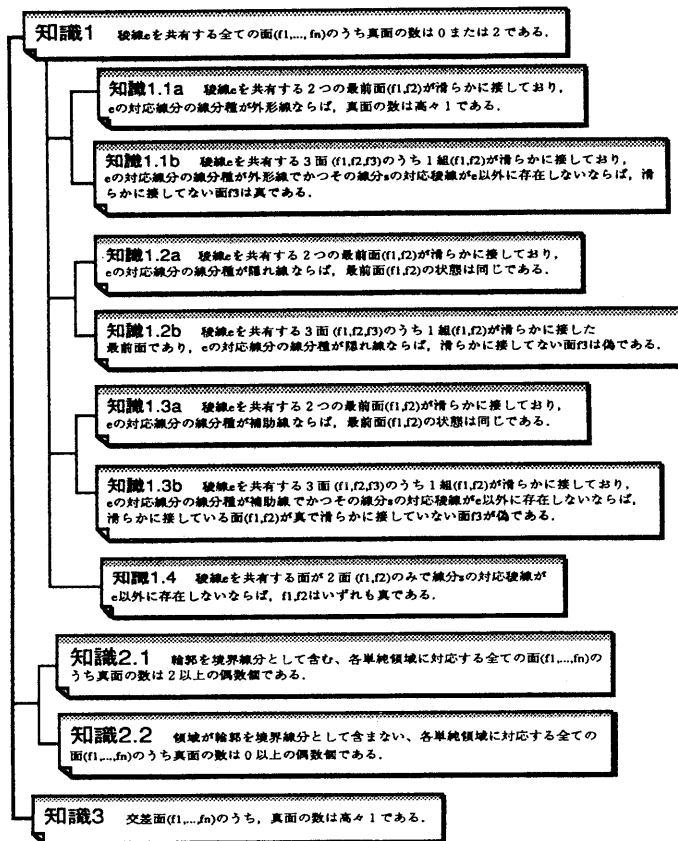


図4 幾何学的知識の分類

if部に列記された述語はそれぞれand結合されており、ユニット組集合はそれぞれの述語によって限定される。すなわち、if部は次のようなユニット組集合を絞り込むフィルタとして機能している。

$$S(\text{Pred}_1) \cap S(\text{Pred}_2) \cap \dots \cap S(\text{Pred}_n)$$

注：S()は述語Pred_nをみたすユニット組の集合を表わす

本システムでは、全てのルールを記述するために14種類の述語を用いている（付録A参照）。

さらにthen部には述語を満たすユニット組に対する可能な部分解を記述する。部分解には宣言型と手続き型の2つの形式がある。宣言型は候補面組に対応する部分解を具体的に列挙する。手続き型は候補面組のなかの真または偽となるべき面の数を示す。この部分解を宣言型で表記する方法は、例えば「10個の候補面のうち真となる面の数は0以上の偶数個である（部分解は $\sum_{i=0,2,4,\dots,10} 5 \cdot 1 \cdot 2$ 個できる）」というように、候補面組に対する部分解の数が組み合わせ爆発を起こしてしまうような知識には適していない。本システムでは、そのような知識の部分解を手続き記述で表現することとした。本システムでは以下の2つの手続きを設定した。

・ labels 'L Num

Lはラベル、Numは整数を表し、ユニット組のうちラベルLを割り当てられる面がNum個であることを示す。本システムでは、LはTまたはF、Numは整数、even0またはeven2である。ここで、even0は0以上の偶数個、even2は2以上の偶数個を表すものである。

・ append_list (labels ...)(labels ...)...

この手続きで表現された部分解は、それぞれの (labels ...) が示す部分解をすべて合わせたものであることを表している。

例えば、図4の知識2.1、知識2.2は次のように記述される。

<知識 2.1 >

```
rule 2.1 no_of_values = 0
  if same_area, outline
  then ( labels 'T even2 )
```

<知識 2.2 >

```
rule 2.2 no_of_values = 0
  if same_area, not outline
  then ( labels 'T even0 )
```

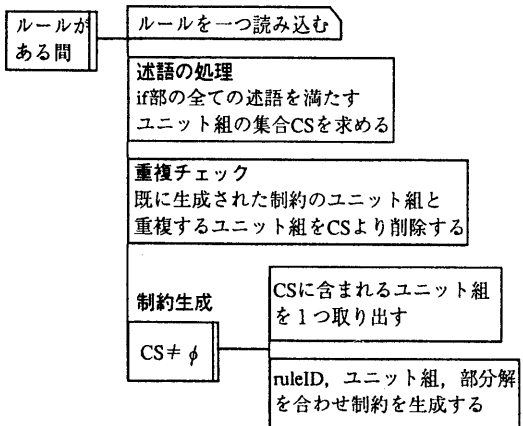


図5 制約生成部のPAD

(2) 制約生成手順

前節で説明したルールベースから制約を生成するアルゴリズムを図5に示す。この中で特徴的な処理は制約条件の重複チェックであり、これは知識1.1a~1.4が知識1の特殊形になっているために必要となる処理である。すなわち、新しく生成された制約条件のユニット組が別のルールから既に生成された制約条件のユニット組の部分集合になっている場合、制約条件の重複がおこる。制約条件が冗長に定義されていると解探索時に余分な制約チェックが必要となり処理効率の低下を招く。そこで重複する2つの制約条件が存在する場合、それらの部分解をチェックし、そのサイズが小さい（部分解の個数が少なくより限定されている）制約条件のみを対応するユニット組の制約として定義する。

図2の三面図の例に rule 2.1 を適用することにより、以下の5つの制約ができる。

- 2.1_1 (8 1)(labels 'T even2)
- 2.1_2 (8 12 9 1)(labels 'T even2)
- 2.1_3 (7 4)(labels 'T even2)
- 2.1_4 (7 11 10 4)(labels 'T even2)
- 2.1_5 (2 5)(labels 'T even2)

3.3 制約の参照と展開

探索手続きのなかでは、制約は、探索するユニットの候補ラベルを決定するときに参照される。例えば、3.2節の(2)の例で生成された

2.1_4 (f7 f11 f10 f4)(labels 'T even2)

という制約は、ユニットf7、f11、f10、f4が探

索される際に参照される。

また、探索するユニットの候補ラベルは、参照した制約の部分解の探索するユニットに対応する部分解を見て決定する。したがって、手続き記述で表現されたままでは候補ラベルを決定できない。そこで、参照された制約の部分解が手続き記述で表現されている場合は、宣言型記述に展開しなければならない。本システムでは、手続き記述の制約を宣言型記述に展開するとき、探索済みのユニットを参照することにより、無駄な部分解を作らないようにし、探索空間の縮小化を図っている。

つまり、手続き記述制約が宣言型に展開される時、ユニット組のなかに既に探索済みのユニットがある場合、その結果を考慮して展開する。

例えば、上で例に挙げた制約 2.1_4 が参照されたときに、f7、f11 のラベルは T である、と決定しているとすると、展開結果は以下のようになる。

(f7 f11 f10 f4)((TTTT)(TTFF))

4つのユニットがすべて未探索である場合は ${}^4C_2 + {}^4C_4 = 7$ つの部分解が生成されることから、部分解の数が減少していることがわかる。

3.4 探索順序の決定

本システムでは、与えられたCSPからユニットの探索順序を決定し、探索順序が決定したユニットのリストを高速化の知識として、解探索部に与えるという方法をとった。ユニットの探索順序は以下のような基準により決定している。

- (1) ユニット組中のユニット数が少ない制約に関係するユニットから優先的に探索する。

- (2) 一つの制約でラベルが決定するユニットから優先的に探索する。
- (3) (2)のユニットを探索した後、(2)のユニットが関係しているような制約に含まれるユニットを探索する。
- (4) (2)、(3)のユニットを探索しても未探索ユニットが多数存在する制約のみに含まれるようなユニットは遅い段階で探索する。
→(2)、(3)のユニットを探索すると、未探索ユニットが小数となる制約に関係するユニットを先に探索する。

生成されたCSPから上記の基準によりユニットの探索順序を決定する手順を以下に示す。

- step1: 一つの制約でラベルが決定するユニットを探索順序リストに加える。
- step2: 探索順序リストにあるユニットを探索すると一意にラベルが決定するユニットを探索順序リストに加える。
- step3: 探索順序リストに無いユニットのうちで評価関数 $h(i)$

$$h(i) = a \times (1000 - |R_i|) + b \times (100 - |t_i|)$$

の値の大きい制約に含まれるユニットから探索順序リストに加える。ただし i は制約番号、 a, b は定数、 $|R_i|$ は局所解の数、 $|t_i|$ は制約中のユニットの数とする。

4 評価実験

以上に述べた手法を用いた三面図理解システムをLUNA88K2上に構築して、その機能を確認した。

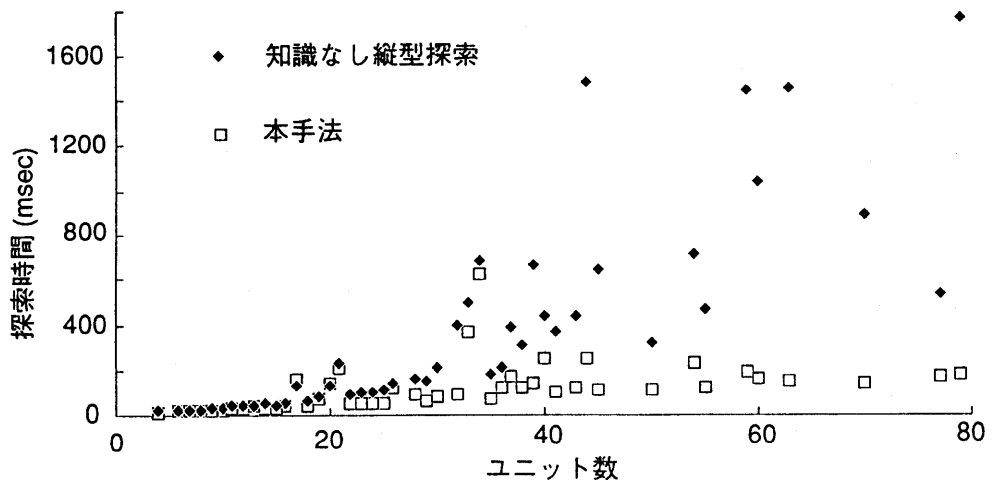


図6 探索にかかる時間

図6は、知識を用いていない縦型探索と本手法において、解探索にかかる時間をグラフで示したものである。横軸は三面図から復元される候補面の数であり、縦軸は解探索にかかった時間である。この実験で比較した知識を用いていない縦型探索では、制約はすべて宣言が記述で表現されている。これに対し、本システムの手法では、3章で述べたように、手続き記述の制約を宣言型記述に展開するとき無駄な部分分解を作らないようにし、知識を用いて探索している。図6のグラフからも、本手法が効率的に解探索をおこなっていることがわかる。

付録Bは実験に用いた三面図とそれから復元された3次元物体の例である。(a)~(f)には曲面(円環面、円柱面、円錐面、球面)からなる図面も含まれており、各種曲面を含む立体が復元されることを確認した。また、(a)~(f)はすべての線分種(外形線、隠れ線、補助線)を含んでおり、線分種に関するルールからの制約生成も正しくおこなわれていることも確認された。

5 おわりに

本論文では、三面図理解に必要な知識を‘制約を生成するためのルールベース’として構築し、三面図理解問題をCSPとして解く手法を提案した。また、本手法が曲面を含んだ三面図に対して有効であること、実用的な時間内でかいを求められることを実験で確認した。

今後は、エラー(書き間違い)を含む三面図についての対策が課題であると考えている。

参考文献

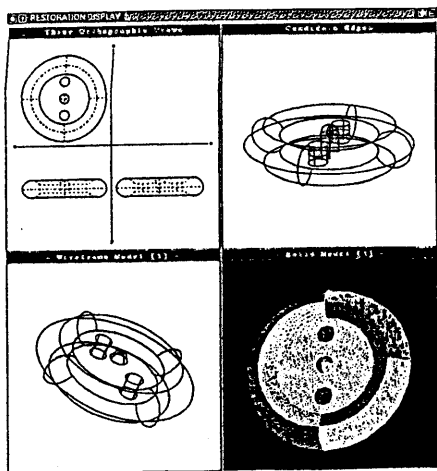
- [出澤 72] 出澤：三面図からの立体形成のためのシステム，日本機械学会論文集，Vol.38, No.310, pp.1267 - 1276 (1972).
- [西原 87] 西原，渡辺，池田：面間の局所的拘束関係を用いた三面図解釈，情処学論，Vol.28, No.5, pp.534 - 537 (1987).
- [伊藤 90] 伊藤：三面図を用いたソリッドモデルの構成—主に多面体を対象として—，情処学会，Vol.31, No.8, pp.1095 - 1106 (1990).
- [西田 91] 西田，張，西原：面の組み合わせ探索による三面図の解釈，人工知能学会誌，Vol.6, No.1, pp.96 - 103 (1991).

- [井上 93] 井上，金，西原：代数曲面を含む三面図の解釈，グラフィクスとCAD61-2, 情処学会 (1991)
- [佐々木 89] 佐々木，伊藤，鈴木：非線形擬似ブール代数法による三面図からの物体の自動合成，情処学論，Vol.30, No.6, pp.699 - 708 (1989).
- [伊藤 93] 伊藤：制約充足問題としての三面図からの物体自動合成，情処研報，93,62 (93-CG-63), pp.107 - 114.
- [西原 90] 西原：整合ラベリング問題と応用，情報処理，Vol.31, No.4, pp.500 - 507 (1990).
- [Haralick 79] Haralick, R.M. and Shapiro.L.G. : The Consistent Labeling Problem, Part 1, IEEE Trans. Pattern Anal. Machine Intel., Vol.1, No.2, pp.173 - 184 (1979).

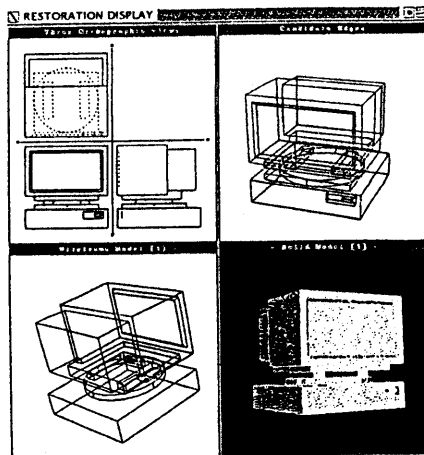
付録A 知識のルール化に使用する述語

- ・ same_area (同じ領域の対応面である候補面組)
- ・ outline (領域が輪郭を境界として含む)
- ・ cross_face (交差面である)
- ・ joint (稜線を共有する)
- ・ joint2 (稜線を共有する候補面が2面である)
- ・ joint3_and_KASSETSU2 (稜線を共有する候補面が3面でそのうち2面が滑接する)
- ・ jointN_and_KASSETSU2 (稜線を共有する候補面のうち2面のみが滑接する)
- ・ nearest_and_KASSETSTU2 (隣接する領域の最前面である2面が滑接する)
- ・ segment_crspnd_ledge (線分に対応する稜線が1つである)
- ・ line_type_GAIKEI (線分種が外形線である)
- ・ line_type_HOJO (線分種が補助線である)
- ・ line_type_KAKURE (線分種が隠れ線である)
- ・ face_type_plane (面種が平面である)
- ・ 論理演算子 not

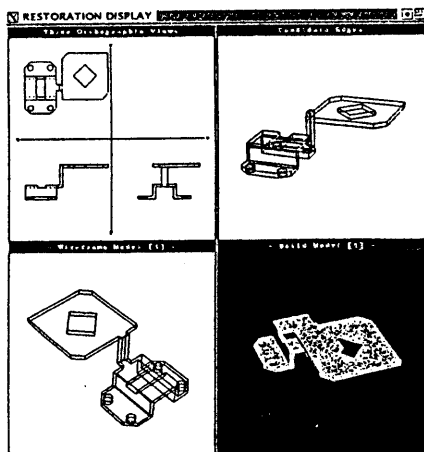
付録B 実験に用いた三面図の例 (a)~(f)



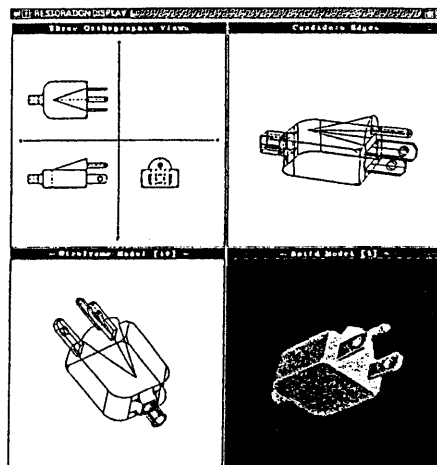
実験に用いた三面図の例 (a)



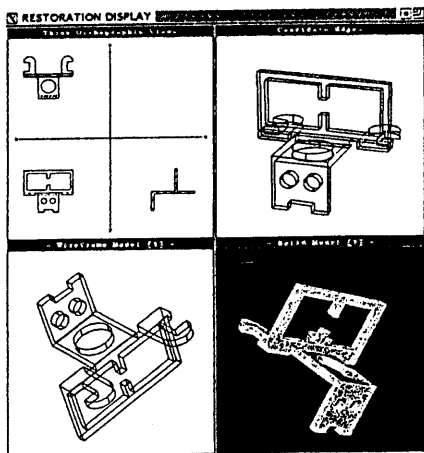
実験に用いた三面図の例 (d)



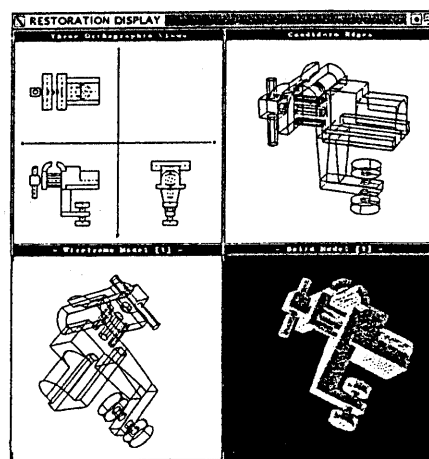
実験に用いた三面図の例 (b)



実験に用いた三面図の例 (e)



実験に用いた三面図の例 (c)



実験に用いた三面図の例 (f)