

制約知識ベースに基づく三面図理解

Understanding Three Orthographic Views by Using Constraint Knowledge Base

内野 寛治* 狩野 均* 西原 清一*
 Kanji Uchino Hitoshi Kanoh Seiichi Nishihara

* 筑波大学 電子・情報工学系
 Institute of Information Sciences & Electronics University of Tsukuba, Tsukuba 305, Japan.

1995年10月18日 受理

Keywords: understanding three orthographic views, CSPs, geometric Knowledge, rulebase.

Summary

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

機械製図で用いる三面図から自動的に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。三面図理解の手続きは、まず図面を構成している2次元要素である点や線分から、3次元要素である頂点や稜線を生成し、さらに面のすべ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, 西田 91]、擬似プ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」をもとにして、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い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押さ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、三面図理解をルールベースを用いて制約充足問題(CPS: constraint satisfaction problem)[西原 90]に変換し表現する手法を提案する。CSPに変換する試みはいくつかあるが[Haralick 79, 伊藤 93, 西原 87]、本論文では、先の復元知識を完全にルールベース化し、それを用いて具体的なCSPを生成するという特徴を持つ。したがって、制約生成のためのルールベースを備えた図面理解システムの提案が本論文の主題である。本手法は復元知識をルールベース化

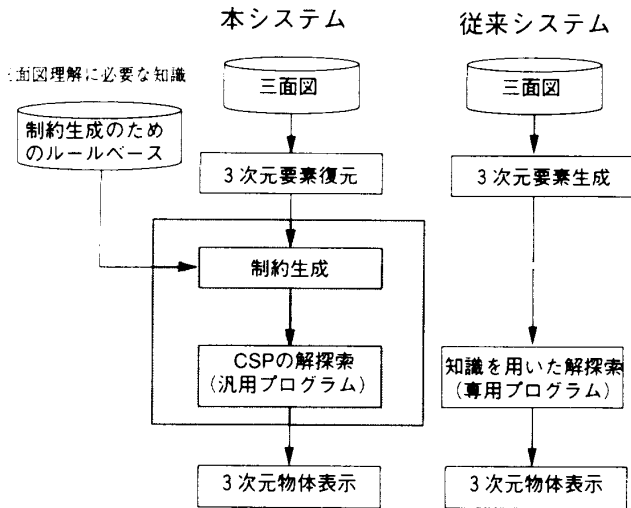


図1 従来システムとの比較

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やすい、(3)CSP の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。図1に本システムと従来システムの処理の流れの違いを示す。

以下、2章で三面図理解問題およびCSPの概説を行った後、3章において三面図理解に必要な復元知識とそのルールベース化、およびCSPへの等価変換手法を述べる。4章では、実際に開発したシステムによる実験とその効率解析を行う。5章では、本方式の特徴と拡張性について考察する。

2. 三面図理解と制約充足の基本概念

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すデータであり、直線と円弧からなるものとする。また、扱う3次元物体は平面と2次代数曲面*1から構成される[井上93]。

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、実線および破線からなる‘線分’によって表されている。本システムにおいて線分は物体の外形を表す‘外形線’、面によって隠された外形を表す‘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す

*1 ここでいう2次代数曲面とは球面、円柱面、円錐面、円環面であり、これらの曲面を任意の座標軸に平行な平面でカットしてできる曲面も含む[井上93]。

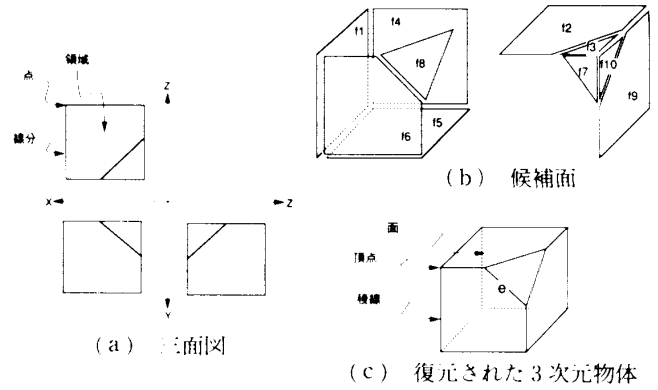


図2 三面図と復元された3次元物体の例

‘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。

一方、本システムで扱う‘シーン’は3次元空間に置かれた多面体であり、複数個の‘面’の集合によって境界づけられた3次元の連結部分である。面は、同一平面上に存在する一続きの閉じた‘稜線’によって境界づけられた有向平面である。稜線の両端点は‘頂点’で定義する。3次元物体の頂点は、三つ以上の稜線が出合う点である。

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て意図する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曖昧性が存在する。つまり、三面図データから復元される3次元要素はシーンを構築するのに必要な要素をすべて含んでいるが、それらすべての要素がシーン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すべての面を‘候補面’と呼び、三面図理解問題を「復元されたすべての候補面のなかから意図する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 $\{f_1, \dots, f_{10}\}$ が復元される。次に、これら候補面のなかから図2(c)に示すように $\{f_1, f_2, f_4, f_5, f_6, f_8, f_9\}$ 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、与えられたすべての制約を同時に満足するような局所解の組合せを求めるような問題の総称である。本論文では、[Haralick 79]に沿ってCSPを四つ組 $(U, L,$

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 によって具体的な制約を構成し、 T は相互に制約し合っているユニットの組(ユニットの制約関係)、 R は T に含まれる各ユニット組に対する可能な部分解(ラベル制約関係)を表す。図2の三面図理解問題の例を CSP に対応させる場合、表1のようになる。

稜線接続などを考慮しない一般的な多面体では、一つの稜線には二つの面が接続している。したがって、ある候補稜線が存在するならば、その稜線に接続する

二つの候補稜線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」という知識として捉えることができる。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_6, f_7, f_8 であるので、ユニット組は $t_1 = (f_6 f_7 f_8)$ で表される。そして、3枚の候補面のうち、「0枚もしくは2枚が真となる」という条件から対応する部分解は $R_1 = \{(F F F), (F T T), (T F T), (T T F)\}$ となる。 $(F F F)$ は稜線 e 自体が存在しないことを表す。すべ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。

表1 三面図問題からCSPへの変換

制約充足問題	三面図理解問題
ユニット(U)	候補面
ラベル(L)	真偽の値
ユニット制約関係(T)	知識を構成する候補面組
ラベル制約関係(R)	候補面組の取り得る状態

$U = \{f_1, \dots, f_{10}\}$
 $L = \{T, F\}$
 $T = \{t_1, \dots, t_{18}\}$
 $t_1 = (f_6 f_7 f_8), t_2 = (f_8 f_9 f_{10}), t_3 = (f_2 f_3 f_8),$
 $t_4 = (f_1 f_5), t_5 = (f_4 f_5), t_6 = (f_1 f_6), t_7 = (f_2 f_6),$
 $t_8 = (f_5 f_6), t_9 = (f_3 f_7), t_{10} = (f_6 f_9), t_{11} = (f_7 f_{10}),$
 $t_{12} = (f_4 f_9), t_{13} = (f_5 f_9), t_{14} = (f_2 f_9), t_{15} = (f_3 f_{10}),$
 $t_{16} = (f_1 f_4), t_{17} = (f_1 f_2), t_{18} = (f_2 f_4),$
 $R = \{R_1, \dots, R_{18}\}$
 $R_1 = R_2 = R_3 = \{(F F F), (F T T), (T F T), (T T F)\}$
 $R_4 = R_5 = \dots = R_{17} = R_{18} = \{(F F), (T T)\}.$

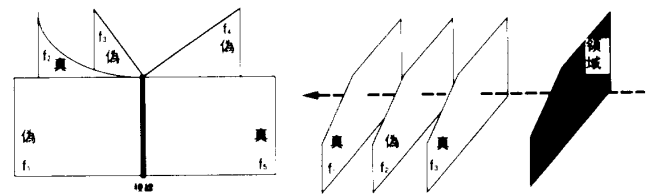
図3 (U, L, T, R)形式で記述したCSPの例

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



(a) 知識1 稜線を共有する候補面のなかで偶数枚が真となる
 (b) 知識2 領域に対応する候補面のなかで偶数枚が真となる

図4 幾何学的知識の例

知識1	稜線eを共有する全ての面(f_1, \dots, f_n)のうち真面の数は0または2である。
知識1.1a	稜線eを共有する2つの最前面(f_1, f_2)が滑らかに接しており、eの対応線の線分種が外形線ならば、真面の数は高々1である。
知識1.1b	稜線eを共有する3面(f_1, f_2, f_3)のうち1組(f_1, f_2)が滑らかに接しており、eの対応線の線分種が外形線であつその線分sの対応稜線が以外に存在しないならば、滑らかに接していない面 f_3 は真である。
知識1.2a	稜線eを共有する2つの最前面(f_1, f_2)が滑らかに接しており、eの対応線の線分種が隠れ線ならば、最前面(f_1, f_2)の状態は同じである。
知識1.2b	稜線eを共有する3面(f_1, f_2, f_3)のうち1組(f_1, f_2)が滑らかに接した最前面であり、eの対応線の線分種が隠れ線ならば、滑らかに接していない面 f_3 は偽である。
知識1.3a	稜線eを共有する2つの最前面(f_1, f_2)が滑らかに接しており、eの対応線の線分種が補助線ならば、最前面(f_1, f_2)の状態は同じである。
知識1.3b	稜線eを共有する3面(f_1, f_2, f_3)のうち1組(f_1, f_2)が滑らかに接しており、eの対応線の線分種が補助線であつその線分sの対応稜線が以外に存在しないならば、滑らかに接している面(f_1, f_2)が真で滑らかに接していない面 f_3 が偽である。
知識1.4	稜線eを共有する面が2面(f_1, f_2)のみで線分sの対応稜線が以外に存在しないならば、 f_1, f_2 はいずれも真である。
知識2	各単純領域に対応する全ての面(f_1, \dots, f_n)のうち真面の数は偶数個である。ただし、その領域が輪郭を境界線分として含む場合は2以上の偶数個である。
知識3	交差面(f_1, \dots, f_n)のうち、真面の数は高々1である。

図5 幾何学的知識の分類

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、次の二つの基本知識が導かれる(図4参照)。

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、まずこれらの知識を分類・整理し、各知識の関係を図5のようにまとめた。

図5において、**知識1**は稜線に関するもの、**知識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面のみの場合の知識(**知識1.4**)を包含している。

本手法では、これら知識を制約条件として利用することで三面図理解問題をCSPパラダイムに変換している。

3.2節では、このような知識を制約条件に変換するための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

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得る真偽値の組合せを表している。図4(a)の例では($f_1 f_2 f_3 f_4 f_5$)という候補面組が(F T F F T)という真偽値の組合せを取っている。そこで本論文では、知識から具体的な制約条件を生成するためのルールを以下の書式を用いて記述することとした。

rule ID
 if $Pred_1, Pred_2, \dots, Pred_n$
 then R

IDはルールの識別番号で、図5の対応する知識の番号を示している。

またif部に記述されているPredは述語で、ユニット組(候補面組)の満たすべき条件を表している。列記された述語はそれぞれand結合されており、ユニット組の集合はそれぞれの述語によって限定される。すなわち、if部の述語はユニット組を以下に示すような共

通集合CSに絞り込むフィルタとして機能している。

$$CS = S(Pred_1) \cap S(Pred_2) \cap \dots \cap S(Pred_n)$$

ただし、 $S(Pred)$ は述語Predを満たすユニット組の集合を表すものとする。

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(ゴール)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処理の違いによるものである。

また、then部のRは、ユニット組が取り得る具体的な真偽値の組合せ(部分解)、またはそれを生成するための手続きである。手続きが必要となる理由は、ユニット組を構成するユニットの個数が限定されないと具体的なラベル組を記述することができないためである。例えば、知識1の「真面の個数は0または2」という条件から($f_1 f_2$)という要素数2のユニット組が得られた場合、部分解Rは $R = \{(F F), (T, T)\}$ であり、($f_1 f_2 f_3$)のように3要素からなるユニット組が得られた場合の部分解Rは $R = \{(F F F), (F T T), (T F T), (T T F)\}$ となる。手続きは、‘制約生成部’において解

表2 本システムで用いた述語と手続き

[述語]	
述語名	述語の意味
same_area	対応する領域が等しい
outline	対応する領域が輪郭を含む
not_outline	対応する領域が輪郭を含まない
cross_face	交差面である
segment_crosspnd_1_edge	共有する稜線に対応する線分が1本のみである
plane_face	面種が平面である
KASSETSU	候補面組の中の最初の2面が滑接している
nearest_faces	候補面組の中の最初の2面が最近面である
joint	ある稜線を共有する面がある
joint(2)	ある稜線を共有する面が2面である
joint(3)	ある稜線を共有する面が3面である
line_type(GAIKEI)	共有する稜線に対応する線分種が外形線である
line_type(HOJO)	共有する稜線に対応する線分種が補助線である
line_type(KAKURE)	共有する稜線に対応する線分種が隠れ線である

[手続き]	
手続き名	手続きの意味
labels (Lbl end)	ラベルLblを条件endの数だけ含むリストを生成する
[条件endの種類]	
even0	0以上の偶数
even2	2以上の偶数
!	リストに含まれる全ての要素数
append_list ((Proc.1)(Proc.2)...(Proc.n))	n個の手続きProc.によって生成されたラベル組をつなぎ合わせて部分解を生成する

積され、対応するユニット組の要素数に応じたラベル組を生成する。幾何学的知識をルールベース化するために用いた述語と手続きを表2に示す。

以上のようなルールの書式を用いると3.1節で述べた知識1, 知識2は次のルールK1, K2.1, K2.2のように簡潔に表現できる(知識2は記述の便宜上、ルールK2.1とK2.2に分けている)。

```

rule K1
  if joint
  then append_list(labels('F!')labels('T2'))
rule K2.1
  if same_area, outline
  then labels('T even2)
rule K2.2
  if same_area, not_outline
  then labels('T enen0)
    
```

また、図5に示した幾何学的知識の全体のルールベースは付録1に示す。

3.3 ルールベースを用いた制約の生成

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6に示す。このなか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すで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

条件の重複が起こる。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。そこで重複する二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(ラベル組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の三面図の例にルールK2.1を適用した結果を表3に示す。また、すべてのルールを適用して生成されたCSPを付録2に示す。

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7~図9は実験に用いた三面図とそれから復元された3次元物体の例の一部で、(c), (e), (f)は後述する表4のデータ記号に対応している。

図7~図9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。さらに表4は、実験に用いた例の立体の面数、復元された立体の数(正解数)ならびに復元までに要した処理時間を示したものである。探索部には単純な木探索と5.1節で述べる高速化手法を取り入れた木探索法の2種類を用いた。

表4から、本手法によって実用的な時間(候補面数97, 正解数1の高速解法の場合で6.3秒)で立体を復元

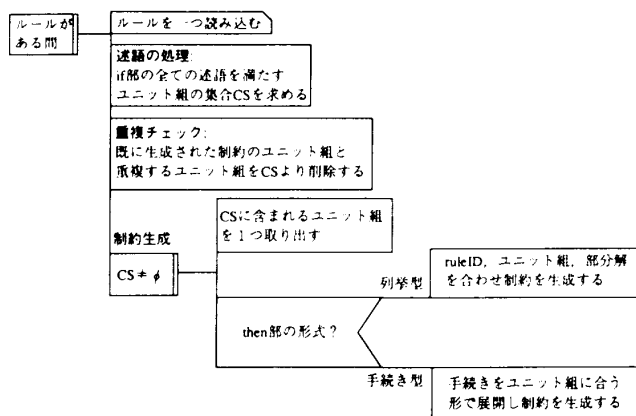


図6 制約生成部のPAD

表3 ルールの適用結果

ルールID	ユニット組	部分解
K2.1_1	(f ₁ f ₂)	((T T))
K2.1_2	(f ₁ f ₇ f ₁₀)	((F T T)(T F T)(T T F))
K2.1_3	(f ₄ f ₆ f ₇)	((F T T)(T F T)(T T F))
K2.1_4	(f ₅ f ₂)	((T T))
K2.1_5	(f ₂ f ₇ f ₉)	((F T T)(T F T)(T T F))
K2.1_6	(f ₂ f ₂)	((T T))

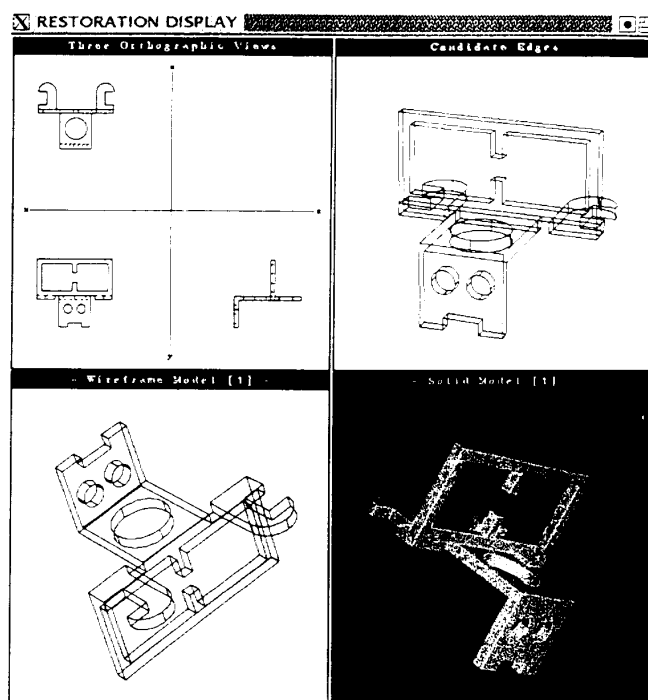


図7 実験に用いた三面図の例(c)

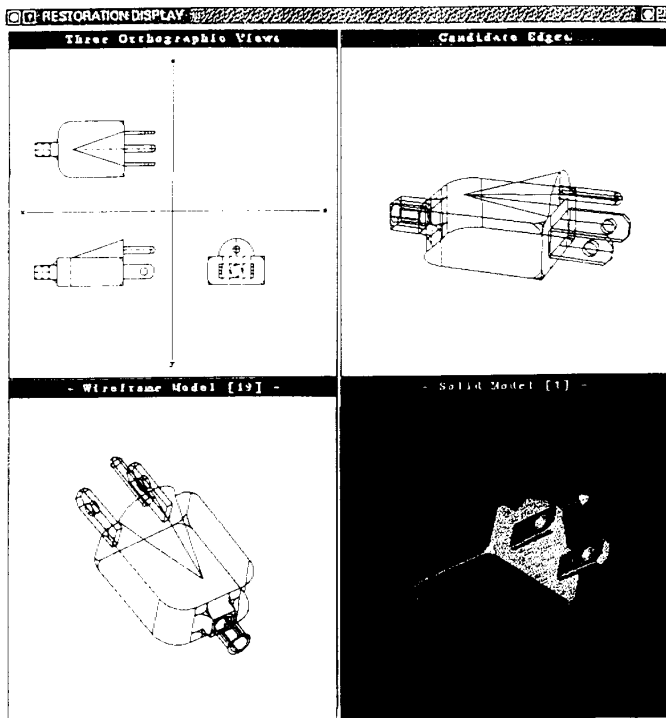


図8 実験に用いた三面図の例(e)

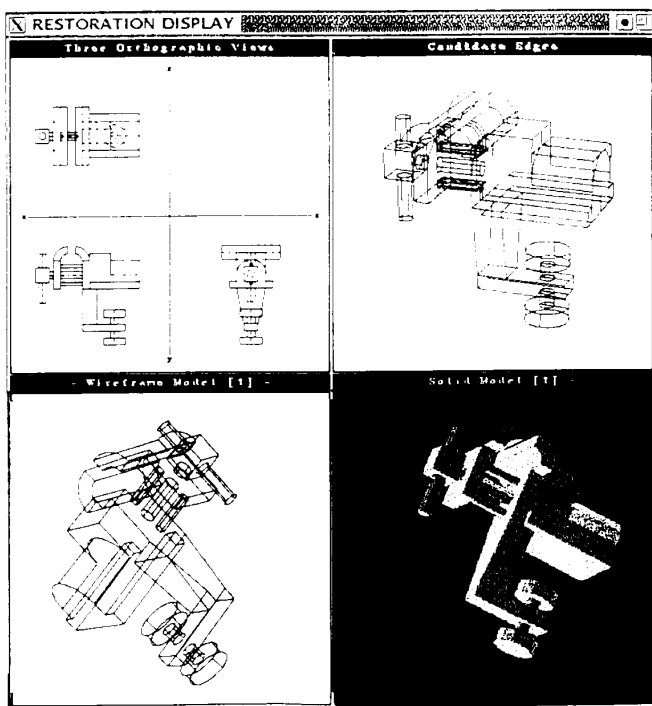


図9 実験に用いた三面図の例(f)

できることがわかる。また、表4の最後の列に示されている従来手法とは[井上 93]による手法であり、本手法と同じ、'3次元要素の復元'処理を行った後、ヒューリスティクスを用いた解探索を行っている。探索部の処理時間は0.1~0.3秒と、与えられた図面にかかわらず高速であるのに対し、本手法では、候補面数や解の個数によって探索時間が大きく異なっていることがわかる。

表4 実験結果

データ記号	面および解の個数		処理時間 (sec)					従来手法 合計
	面数	正解数	本手法				合計(探1/探2)	
			要素復元 ¹⁾	生成 ²⁾	探索1 ³⁾	探索2 ⁴⁾		
(a)	67	1	3.3	0.5	0.4	1.2	4.2 / 5.0	3.5
(b)	89	1	4.1	0.5	0.5	0.7	5.1 / 5.3	4.2
(c)	97	1	5.1	0.5	2.1	0.7	7.7 / 6.3	5.4
(d)	71	8	1.8	0.5	3.0	0.9	5.3 / 3.2	1.9
(e)	139	19	7.4	0.7	9.4	2.3	17.5 / 10.4	7.5
(f)	197	18	20.6	1.3	25.3	4.3	47.2 / 26.2	20.7

(注) (1) 3次元要素の復元、(2) ルールベースを用いたCSPへの変換、
(3) CSPの解探索 (単純な木探索)、(4) CSPの解探索 (高速化手法)

なお、探索時間を含め処理時間の内訳については5・1節で考察する。

5. 本手法の拡張性

5・1 処理の高速化

表4の処理時間の内訳を見ると、3次元要素の復元とCSPの解探索が大きな比率を占めていることがわかる。このうち、前者は、従来手法[井上 93]にも共通の部分であり、後者は、本手法独自のものである。ここでは後者を高速化することを考える。

まず第一に、本手法では三面図理解問題をCSPで表現しているので、既存のCSPの解法をそのまま適用することができる。これには、木探索法[Haralick 79]や併合法[内野 94]などの厳密解法、MCHC法[Minton 92]やGA[松本 95]などの確率的解法が提案されており、それぞれ並列処理などの高速解法が報告されている。

第二に、CSPの特徴を生かした高速化が考えられる。この高速化を施したのが実験で用いた探索2であり、これは「部分解Rのサイズが小さい制約を優先的に探索したほうが効率が良い」という基本的なCSPのヒューリスティクスに基づいており、具体的には次のような手順で処理される。

まず、(1)ルール K 1.3 b, K 1.4 から生成された列挙制約は部分解がただ一つなので、このような制約を構成する候補面組は最初に探索する。次に、(2)ルール K 1.1, K 1.2, K 1.3 a, K 3 から、得られた制約は部分解の個数が少ない(2または3個)ので、このような候補面組を次に探索する。さらに、(3)探索がある程度進むとすでに探索された候補面を多く含む制約が増え、このような制約は部分解Rを絞り込むことができると考えられるので、上記の処理で決定した探索順序の次に探索するようにする。

このようなCSPの単純なヒューリスティクスを用いた高速化でも表4の探索2で示した効果が得られた。さらには、探索の分岐に関するヒューリスティクスの導入なども今後考えられる。なお、(a)、(b)の例で高速化手法が単純な木探索より処理が遅いのは、生成されたCSPの制約順序がたまたま最適な並びであり、最適な探索順序を決定する前処理を行わない単純な木探索のほうが速いためである。候補面数の多い他の例ではこのような逆転は見られなかった。

5・2 ルールベースの調整

本方式によると、ルールベースを調整することによって多様な図面に適した処理を容易に実現することができる。本節ではこのことを(1)三面図中にエラー(描き間違い)がある場合、および(2)限定された3次元物体の例として板金部品の三面図の場合について、ルールベースの調整による対応が可能であることを示す。

(1) エラー図面への対応

三面図中にエラーが存在すると各面図間の対応に矛盾が生じ、通常立体を復元することができない。本節では、このような三面図に対しても本手法によって、正しい立体を復元できる可能性があることを示す。ここでは、三面図中に現れる線分種に関するエラーを対象とする。

線分種は図面をスキャナなどを用いて計算機に読み込む場合、解像度の度合いから隠れ線や補助線が互いに間違っ取り込まれることが多い。例として、図2の三面図を図10のように一部の外形線を補助線に描き間違えた場合についてエラーのない場合と比較して考察する。

図10を本手法でCSPに変換した結果、補助線に関するルール **K 1.2 a**, **K 1.2 b** によって次のような制約が生成された。また、ルール **K 1**, **K 2** から生成される制約については付録2と同じであるが、外形線によるルール **K 1.1 a**, **K 1.1 b** による制約は生成されない。

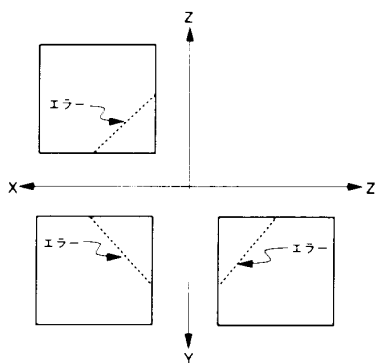


図10 エラーを含む三面図の例

K 1.2 a より:

$K1.2a_1 (f_2 f_3) ((T T)(F F))$

$K1.2a_2 (f_6 f_7) ((T T)(F F))$

$K1.2a_3 (f_9 f_{10}) ((T T)(F F))$

K 1.2 b より:

$K1.2b_1 (f_2 f_3 f_8) ((T T F)(F F F))$

$K1.2b_2 (f_6 f_7 f_8) ((T T F)(F F F))$

$K1.2b_3 (f_6 f_{10} f_8) ((T T F)(F F F))$

ここで、候補面組の種類に関しては付録2と同じだが、部分解 R が異なっており、矛盾のある制約が生成されている。したがって、このままでは f_8 が必ず偽となり、意図する解を得ることはできない。そこで、線分種が関係するルール **K 1.1 a**~**K 1.4** をルールベースから削除し、CSPを生成するとルール **K 1** と **K 2.1** による制約のみが生成される。これを解くと二つの解 $\{(T T F T T T F T T F), (T T T T T T T F T T)\}$ が得られ、このうち最初の解が意図する3次元物体となっていることがわかる。

以上のようにエラーを含む三面図に対しても本手法では、(i)エラーに関する述語を含むルールをルールベースから削除してCSPを生成し、(ii)CSPを解いて得られた解のなかからユーザが対話的に正解を選ぶ、という手順によって、意図する立体を復元することができる。

(2) 板金物体に限定した場合

板金物体とは、厚さが均一の連続した1枚の平板に切断、穴あけ、曲げなどの加工を施して生成される3次元物体である。図7の実験に用いた三面図の例(c)はその一例である。板金物体の最大の特徴は板厚が一定である点なので、この特徴を生かしたルールをつくり、ルールベース中に加えることを考える。すなわち、図5の知識2をさらに特殊化した「輪郭を境界線分として含む領域に対応する候補面のうち、面間の距離が板厚と等しい候補組は真である」という知識を次のようにルール化し、ルールベース中に加える。

rule **K 2.3**

if same_area, outline,

(distance == plate_thickness)

then labels ('T!')

このルールにより真と決定できる候補面が増え、その結果、探索空間を縮小し処理の効率化を図ることができる。このように本システムでは、図面の特徴を制約生成のためのルールとして記述することによって、効率良く制約を生成し、システム全体の処理効率を上げることが可能である。

6. 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、実用的な時間内で解を求められることを実験で確認した。さらに、本手法の拡張性として、CSP の特徴を利用した処理の高速化、エラーを含んだ三面図からの復元可能

性、さらに、板金物体に限定したときのルールベースのカスタマイズについて述べた。

今後の課題としては、本手法で対応できるエラーの範囲を明らかにし、より知的な概略理解の手法の開発がある。

謝 辞

本研究に御協力いただいた筑波大学第三学群情報学類、蒔田尚子氏、若林明子氏(現 日本電気(株))、山田弘展氏(筑波大学理工学研究科)に感謝致します。

◇ 参 考 文 献 ◇

- [Haralick 79] Haralick, R. M. and Shapiro, L. G.: The Consistent Labeling Problem, Part I, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 1, No. 2, pp. 173-184 (1979).
- [出澤 72] 出澤: 三面図からの立体形成のためのシステム, *機械学論*, Vol. 38, No. 310, pp. 1267-1276 (1972).
- [井上 93] 井上, 金, 西原: 代数曲面を含む三面図の解釈, *情処学研報*, Vol. 93, No. 14, pp. 9-16 (1993).
- [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, *情処学研報*, Vol. 93, No. 62, pp. 107-114 (1993).
- [松本 95] 松本, 内野, 西原: 遺伝的アルゴリズムによる制約充足問題の解法, *情処学研報*, Vol. 95, No. 86, pp. 33-40 (1995).
- [Minton 92] Minton, S., Johnston, M. D., Philips, A. B. and Laird, P.: Minimizing conflict: a heuristic repair method for constraint satisfaction and scheduling problem, *Artif. Intell.*, Vol. 58, pp. 161-205 (1992).
- [西田 92] 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. 96-103 (1991).
- [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).
- [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).
- [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).
- [内野 94] 内野, 窪田, 狩野, 西原: 制約充足問題の並列化効率に基づく分類, *情処学論*, Vol. 35, No. 12, pp. 2676-2684 (1994).

[担当編集委員・査読者: 久野義徳]

◇ 付 録 ◇

1. 三面図理解に用いたルールベース

```
rule K1
  if joint
  then append_list(labels('F !') labels('T 2'))

rule K1.1a
  if joint(2), nearest_faces,
    KASSETSU, line_type(GAIKEI)
  then {(F F)(T F)(F T)}

rule K1.1b
  if joint(3), KASSETSU,
    segment_crrspnd_1_edge, line_type(GAIKEI)
  then {(T F T)(F T T)}
```

```
rule K1.2a
  if joint(2), nearest_faces,
    KASSETSU, line_type(KAKURE)
  then {(F F)(T T)}

rule K1.2b
  if joint(3), nearest_faces,
    KASSETSU, line_type(KAKURE)
  then {(F F F)(T T F)}

rule K1.3a
  if joint, nearest_faces,
    KASSETSU, line_type(HOJO)
  then {(F F)(T T)}

rule K1.3b
  if joint(3), KASSETSU,
    segment_crrspnd_1_edge, line_type(HOJO)
  then {(T T F)}
```



```
rule K1.4
  if joint(2), segment_crrspnd_1_edge
  then {(T T)}
```

```
rule K2.1
  if same_area, outline
  then labels('T even2')
```

```
rule K2.2
  if same_area, not_outline
  then labels('T even0')
```

```
rule K3
  if cross_face
  then {(F F)(T F)(F T)}
```

2. ルールベースより生成された CSP の例

K1_1 (f₆ f₇ f₉) ((F F F) (F T T) (T F T) (T T F)),
 K1_2 (f₈ f₉ f₁₀) ((F F F) (F T T) (T F T) (T T F)),
 K1_3 (f₂ f₃ f₉) ((F F F) (F T T) (T F T) (T T F)),
 K1_4 (f₁ f₃) ((F F) (T T)),
 K1_5 (f₄ f₃) ((F F) (T T)),
 K1_6 (f₁ f₆) ((F F) (T T)),
 K1_7 (f₂ f₆) ((F F) (T T)),
 K1_8 (f₃ f₆) ((F F) (T T)),
 K1_9 (f₃ f₇) ((F F) (T T)),
 K1_10 (f₆ f₉) ((F F) (T T)),
 K1_11 (f₇ f₁₀) ((F F) (T T)),
 K1_12 (f₄ f₉) ((F F) (T T)),
 K1_13 (f₃ f₉) ((F F) (T T)),
 K1_14 (f₂ f₉) ((F F) (T T)),
 K1_15 (f₃ f₁₀) ((F F) (T T)),
 K1_16 (f₁ f₄) ((F F) (T T)),

K1_17 (f₁ f₂) ((F F) (T T)),
 K1_18 (f₂ f₄) ((F F) (T T)),

K1.1a_1 (f₂ f₃) ((F T) (T F) (F F)),
 K1.1a_2 (f₇ f₆) ((F T) (T F) (F F)),
 K1.1a_3 (f₁₀ f₉) ((F T) (T F) (F F)),

K1.1b_1 (f₂ f₃ f₈) ((T F T) (F T T)),
 K1.1b_2 (f₆ f₇ f₈) ((T F T) (F T T)),
 K1.1b_3 (f₉ f₁₀ f₈) ((T F T) (F T T)),

K2.1_1 (f₃ f₈ f₃) ((F T T) (T F T) (T T F)),
 K2.1_2 (f₇ f₈ f₄) ((F T T) (T F T) (T T F)),
 K2.1_3 (f₁₀ f₈ f₁) ((F T T) (T F T) (T T F)),
 K2.1_4 (f₂ f₃) ((T T)),
 K2.1_5 (f₆ f₄) ((T T)),
 K2.1_6 (f₉ f₁) ((T T)).

著者紹介



内野 寛治(学生会員)

1991年筑波大学第三群情報学類卒業。1996年同大学院工学研究科修了。現在、(株)富士通研究所メディア統合研究部勤務。工学博士。人工知能、制約充足問題、自然言語処理の研究に従事。情報処理学会会員。



狩野 均(正会員)

1978年筑波大学第一学群自然科学類卒業。1980年同大学院工学研究科修士課程物理学専攻修了。同年、日立電線(株)入社。同社オプトロシステム研究所において人工知能・神経回路の応用に関する研究に従事。1993年より筑波大学電子・情報工学系。現在、同助教授。制約に基づく知識表現、遺伝的アルゴリズムの研究に従事。工学博士。1992年電気学会論文賞受賞。電気学会、情報処理学会、計測自動制御学会等各会員。



西原 清一(正会員)

1968年京都大学工学部数理工学科卒業。同年、同大学大型計算機センター助手。1975年より筑波大学電子・情報工学系。現在、同教授。工学博士。1982～83年文部省在外研究員として米国ワシントン工科大学。図形画像処理、グラフ構造、組合せ探索、制約充足問題の研究に従事。著書に「グラフ構造」(オーム社)など。情報処理学会、電子情報通信学会、ACM、IEEE各会員。