

曲面を含む三面図の矛盾の検出と理解

梅澤 顕 菊池浩司 狩野 均 西原清一
筑波大学 電子情報工学系

三面図を解釈し立体モデルを自動復元するシステムにおいて、矛盾を含む三面図が入力された場合に、その矛盾を解消するための一手法を開発した。立体モデルの復元処理過程で、実線/破線/鎖線といった線分属性に関する入力エラーが原因で矛盾が生じた場合、探索に用いる制約を減らすことにより、探索空間を拡大して探索を続行し、線分属性を訂正しながら3次元物体を復元する。本手法を曲面を含む三面図に適用し、その有効性を確認した。

CHECKING AND RECOGNITION OF INCONSISTENCY
IN THREE ORTHOGRAPHIC VIEWS
INCLUDING ALGEBRAIC CURVED SURFACES

Ken Umezawa Koji Kikuchi Hitoshi Kanoh Seiichi Nishihara
Inst. of Inf. Science & Electronics, Univ. of Tsukuba
Tennodai 1-1-1, Tsukuba, Ibaraki 305, Japan

In this paper we described a method to correct errors on a three orthographic views including contradictions in a system that reconstructs 3-D models of polyhedrons. When a contradiction with respect to attribute of segments (solid line, dotted line and chain line) is detected in the process of 3-D model reconstruction, search space is expanded by reducing constraints that are used in searching process in order to continue searching. 3-D model is reconstructed with attribute of segments corrected. The function of this method is confirmed by applying it to three orthographic views including curved surfaces.

1 はじめに

三面図から3次元モデルを自動復元する方法は、CADの入力手段として多くの機関で研究されている[1~7]。著者らは以前、代数曲面を含む立体モデルの復元に関する研究として、3次元物体が3次元空間内に存在するための制約条件や、人間が用いていると思われる知識を用いて三面図を解釈し、立体モデルを効率良く復元するシステムを開発実用化した[10]。ここでは、自然な解の優先復元、探索速度向上に有効なヒューリスティクスを示した。

ただし、この三面図解釈では、矛盾を含まない図面が入力されることを前提としていた。しかし、三面図は人間が何らかの方法で入力するものであるため、実際には間違っただけの図面が描かれていることもありうる。こういった場合、復元過程で矛盾が検出され、立体モデルの復元は途中で中止される。そこで、この問題に対処するため、矛盾を含む図面が入力された場合に、これを修正する方法を検討した。図面における矛盾としては多々挙げられるが、本稿では、その中でも描かれている図面の線分属性(後述)が間違っている場合を対象とし、矛盾の指摘と図面の修正を行うための手法を述べる。

本手法は、立体モデルの復元過程で、線分属性の間違いが原因で矛盾が生じた場合、探索に用いる制約を減らすことにより、探索空間を拡大して探索を続行し、線分属性を訂正しながら3次元物体を復元するものである。

以下、2章ではエラーが無い場合の三面図解釈処理の概要を説明する。3章では、解釈処理における矛盾を分類し、矛盾解消の方法について述べる。4章では、本手法の評価として実施例を示す。5章では、図面上のエラーを分類する。

2 三面図の解釈と制約・知識

本章では、エラーが無い場合の三面図解釈について概要を説明する[7]。

2.1 諸定義

入力は三面図を表す‘点’、‘線分’のデータである。線分には、直線分と曲線分があり、線分属性と

して、実線、隠れ線を表す破線、補助線を表す一点鎖線がある。各面図で、閉じた線分列が示す区域を‘領域’、背景と接する最も外側の領域を構成する線分を‘輪郭’と呼ぶ。また、面と面とが滑らかに接すること(1階微分連続)を‘滑接’と呼ぶ。

曲面を扱うために、2種類の‘補助線’を導入する。1つは、違う面種が滑接する‘接稜線’、もう1つは、回転体を面図に投影したとき‘シルエット’の部分に生成する‘シルエット稜線’である。これらの‘補助線’を面図に投影して新たにできる線分を‘補助線分’と呼ぶ。図1に補助線の例を示す。

なお、シルエット稜線は、通常三面図には描かれないので、解釈の前処理として自動生成している。この方法を付録に示す。

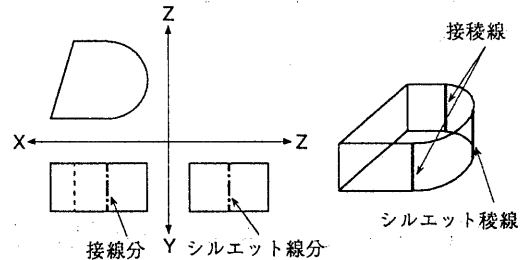


図1 補助線の例

2.2 三面図解釈

入力された三面図から、面図間の対応をとりながら各3次元要素(頂点、稜線、候補面)を復元する。

次に、面の組合せ探索によって、候補面の集合から虚要素を排除し、三面図に矛盾しない立体モデルを構築する。ただし、立体モデルは、1つ以上の物体からなる幾何学的に正しいシーンを形成し、体積を持っているものとする。

2.3 面の組合せ探索

復元は、全ての候補面を、解を構築する‘真’の面と、しない‘偽’の面に状態決定する二分木探索によって行われる。よって、候補面数 n のとき、面の組合せ数は 2^n になる。

単純なバックトラックによる方法では、 2^n をすべて調べる完全二分木探索となり、無駄な探索が多く処理効率が悪い。そこで、以下の制約と知識を用いて無駄な探索を抑え、処理効率の向上を図る。

【制約】 幾何学的に3次元物体が3次元空間内

に存在するための拘束条件「3次元空間における任意の閉曲線は、3次元物体の表面と偶数回交差する」、立体モデルを三面図に投影したときの線分属性や輪郭についての整合性、等。

【知識1】 人間は三面図を解釈する場合、視点に最も近い面（可視部）から経験的に正しいと思われる面を決定する。次に、その面の決定状態から派生する拘束条件によって、他の面の真偽決定を行う。

【知識2】 人間は、見えないところは単純なものを想像する傾向がある。

制約は効率的な候補面の真偽の決定、知識1は処理の高速化、知識2は多義解釈時、最も自然な解を最初に復元するために用いる。

3 線分属性訂正による矛盾解消

三面図に線分属性の間違った線分が描かれている場合について、これを修正して正しいと思われる解を復元する。また、全体の処理の概要を図2に示す。なお本研究では、復元方法、及び矛盾の性質から「要素の復元時に候補面の枚数が増減したり、構成稜線が変更したりすることはない」という条件を得ることができる。

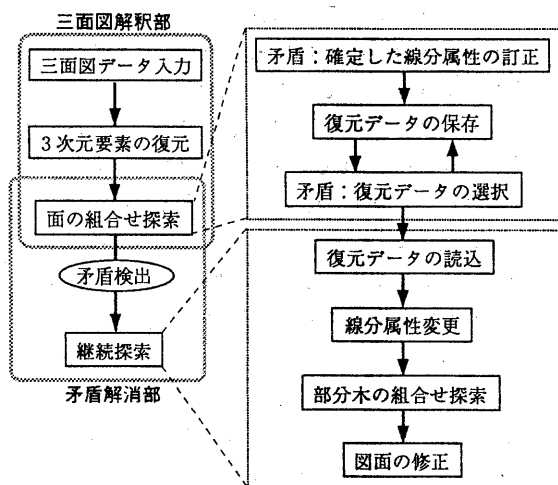


図2 処理の概要

3.1 矛盾の検出

面の組合せ探索時、候補面の真偽が決定される度に、制約に矛盾しないかがチェックされる。このチェックで矛盾が検出された場合は、間違えた

分岐なのでバックトラックし、別の分岐へ進む。バックトラックの結果、探索木のルートまで戻ってきたときに、他に分岐がなければ、矛盾を含む図面と判断し矛盾の解消を行う。

3.2 矛盾の分類

探索過程で検出される制約との矛盾を、以下の8つに分類する。

- 【矛盾1】 稜線を共有する真の面が奇数枚存在する。
- 【矛盾2】 単純領域に対応する真の面が奇数枚存在する。
- 【矛盾3】 輪郭を境界線分として含む単純領域に対応する真の面が2枚未満になる。
- 【矛盾4】 線分に対応する稜線が存在しない。
- 【矛盾5】 補助線分が補助稜線以外の稜線に対応する（補助稜線を介して接続する面が滑接していない）。
- 【矛盾6】 滑接する2枚の真の最前面の投影図面における境界線分が実線である。
- 【矛盾7】 滑接しない2枚の真の最前面の投影図面における境界線分が破線である。
- 【矛盾8】 滑接する真と偽の最前面の投影図面における境界線分が破線である。

矛盾1,2は、3次元物体が3次元空間内に存在するための条件に対する矛盾である。矛盾3~8は、三面図との整合性に関する矛盾である。特に、矛盾5~8は、線分属性に関する矛盾である。

3.3 矛盾を考慮した探索

ここでは、上記の制約を線分属性には関係しないものと、するものに分類して考える。前者を「制約1」、後者を「制約2」とする。制約1の例としては、「視線に平行な無限直線と面が偶数回交差し、面の表の向きが交互に入れ替わること」、「稜線上の任意の点において接する面は偶数枚であること」などがある。制約2の例としては、「滑接しない最前面の境界を投影した場合、実線である」、「破線に対応する稜線を隠す面が存在する」などがある。

ここで、制約2を用いずに最初から探索すれば、必ず何らかの解が得られる。しかし、これでは分岐

数が増し探索空間が広がりすぎてしまうため、無駄な探索が増え、余分な解まで得ることになってしまう。

そこで、2章で述べた手法の組合せ探索部において、制約1で真偽決定できる面は可能な限り決定し、必要に応じて制約2を用いて決定することにした。制約1でも制約2でも決定できない場合は、真の場合、偽の場合について分岐する。したがって処理としては、次の順に優先して探索を進めるようにした。

- ① 制約1を用いた探索
- ② 制約2を用いた探索
- ③ 通常の2分木探索

これにより、制約2の適用を最小限に抑えられるので、線分属性によって真偽が決定される面を少なく抑えることができる。したがって、線分に関する矛盾が生じたとき、その線分の属性が間違っている可能性が高くなる。また、矛盾1～3の発生も回避することができ、矛盾を矛盾4～8に集中させることができる。

3.4 線分属性の訂正

<第1段階>

探索開始時より、制約1を用いた探索のみで決定した面の状態は、必ず正しい。したがって、この間に矛盾が発生した場合には、矛盾の発生した箇所の線分属性が間違っているといえる。この場合は、次の条件によって線分属性を訂正し、探索を続行する。

(各線分属性の決定条件)

実線：最前面の滑接しない稜線に対応する
破線：・最前面以外の滑接しない稜線に対応する
・面が4枚以上滑接する稜線に対応する
補助線：対応稜線の接続面が全て2枚でかつ滑接しており、対応稜線に補助稜線を1本以上含む

この条件を満たせず属性が決定できない場合は、属性を絞り込んで保留しておき随時決定していく。また、後の探索でやむおえず制約2を用いることになった場合には、どちらにも解釈できるようにしておく。

三面図に矛盾が含まれている場合でも、ここまでで矛盾が解消されれば、バックトラックすることなしに線分属性の訂正と物体の復元が可能である。

<第2段階>

探索開始時より、制約1を用いた探索のみでは、解が得られなかった場合の処理である。ここでは、矛盾の解消が複数考えられるので、矛盾が発生せずに探索可能なところまで探索する。探索終了後、解が存在しなければ、各矛盾に対して矛盾を解消して探索を継続する。この処理を実現するために、組合せ探索における探索途中の復元データを保存し、矛盾発生時に適当なデータを選択しておく。そして、矛盾の解消を実行する時に、この復元データを基に探索を継続する。

ここで、全ての復元データを保存しておくのは効率的ではないので、探索途中において、制約2で状態が決定した場合は復元データを保存し、制約1で決定した場合には保存しないことにする。これは、制約2を用いた真偽決定に関しては、異なる状態の探索が考えられるが、制約1を用いた真偽決定に関しては、再び探索しても同じ探索経路をとるためである。これによって、同じ探索を2度行うことは避けることができる。また、線分属性に関する矛盾が生じた場合にも、復元データを保存する。

矛盾が発生した場合、適当な復元データを選択するが、その時点の復元データを採用して良いとは限らない。なぜなら矛盾の原因は、①矛盾の発生した箇所の線分属性にある場合と、②それ以前の面を真偽決定するのに用いた線分属性にあり、間接的に矛盾が発生する場合があるからである。このことから、2つの選択を考える。

- (1) 矛盾の発生した時点のデータ
- (2) 間違った状態と考えられる面が未決定となる時点までバックトラックしたデータ

各矛盾に対しては、次のように用いる。

[矛盾4]

矛盾はその時点の線分属性とは関係ないので、(2)を用いる。

[矛盾5～8]

(1)を最初に用いる。3.3の探索順序変更から、生じる確率が高くなっており、探索の進んだ状態のままであるので探索効率もよい。これで解が得られなかったときは、(2)を用いる。

このようにして、復元データを保存しながらの面の組合せ探索が終了した時点で、解が存在しなければ、矛盾を含む図面である。

矛盾を含む場合、各矛盾に対して選択した復元データを基にして線分属性を訂正し、組合せ探索を継続する。(2)の場合は、今とは逆の状態の分岐のみを探索してゆく。これにより、同じ枝を探索することは避けられる。

再び矛盾が生じた場合は、各部分木においてこの処理を繰り返す。結果として、線分属性の変更が少ない物体から復元されることになる。

解が得られた時点で、もとの線分属性と修正後の線分属性を比較し、異なる箇所を表示する。

以上の処理を図3に示す。

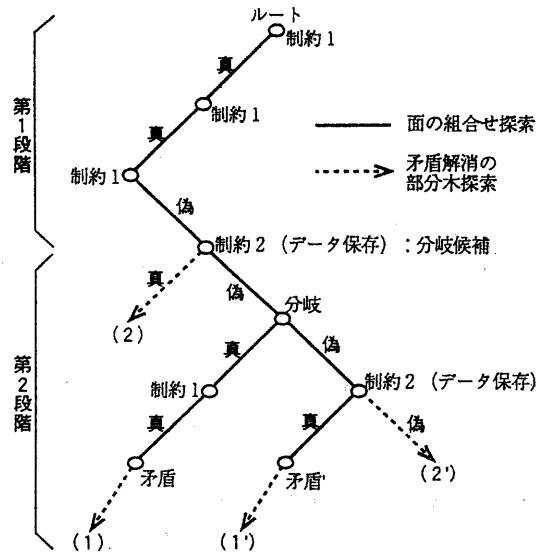


図3 面の組合せ探索課程

4 実施例

本手法を曲面を含む三面図に適用した実施例を図4に示す。

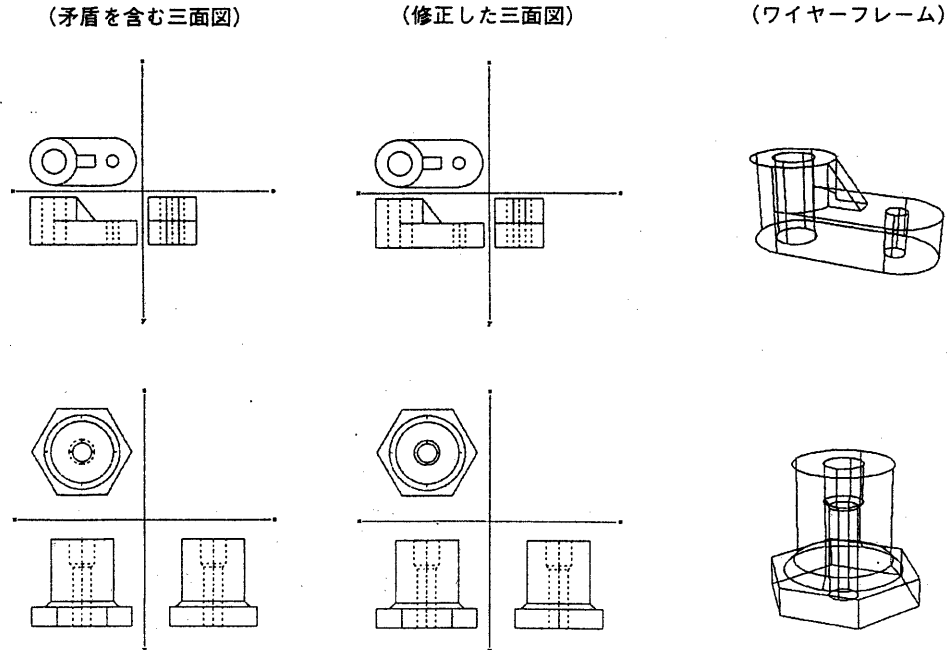


図4 本手法の実施例

5 三面図上のエラーの分類

本稿では、三面図上の線分属性のエラーに起因する矛盾を解消する手法を述べた。しかし、一般には、三面図解釈における矛盾の原因としては、線分属性エラーの他にもいくつかエラーが考えられる。図5は矛盾の検出という観点からエラーを分類したものである。「単純エラー」は、正面図、平面図、側面図の各面図間の対応をとらなくても検出できるもの、「幾何エラー」は、各面図間において点の座標値が一致しないもの、「位相エラー」は、各面図間で線分の対応がとれないものである。

矛盾の解消を考えた場合、単純エラーについては、独立点、線分の閉包などの条件から容易に修正できる。幾何エラーについては、確率的弛緩法[11]が有効と考えられる。位相エラーについては、ほとんど検討されていない。なお、矛盾の生じないエラーが含まれている場合には、設計者の意図したものと異なる立体が復元されるが、システム側での検出は不可能である。このようなケースに対しては、対話形式で立体を修正し、修正後の立体から三面図を自動修正する方法が実用的と考える。

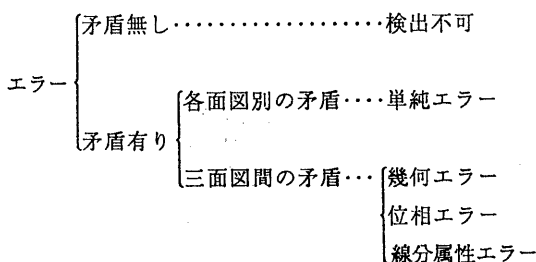


図5 三面図解釈におけるエラーの分類

6 おわりに

三面図中の線分属性エラーに起因する矛盾が生じたとき、エラーの指摘と修正を行う一種法を提案した。また、この手法を曲面を含む三面図に適用し、有効性を確認した。

参考文献

1)西田,張,西原:面の組合せ探索による三面図の解釈,

人工知能学会第3回全国大会,437-440(1989)

2)Kim,Inoue,Nishihara:Heuristic Understanding of Three Orthographic Views,JIP,15,4(1992)

3)千田:三面図から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,1122-1128(1991)

4)宮本,小林,伊藤:外接による三面図からの曲面物体の合成,人工知能学会第4回全国大会,407-410(1990)

5)横山,河上:三面図から曲面を含む立体の自動生成,日本機学会論文誌56巻526号,174-179(1990)

6)Sakurai,H.:Solid Model Input Through Orthographic Views,CG17.3,243-252(1989)

7)井上,金,西原:代数曲面を含む三面図の解釈,グラフィックスとCAD,61-2(1993)

8)菊池:補助線の自動生成機能を備えた対話型三面図入力システム,筑波大学情報学類卒業論文(1993)

9)梅澤:矛盾を含む三面図の修正支援機能に関する研究,筑波大学情報学類卒業論文(1993)

10)Computer Today,7月号(1993)

11)堀内,寅市,山本,山田,大瀧:弛緩整合法による三面図からの多面体復元方法,電子情報通信学会論文誌,J76-D,9-19(1993)

付録 補助線分の自動生成

2.1で述べた2種類の補助線分のうち、シルエット線分の自動生成を行う。自動生成が可能なのは、円柱、円錐、円環、球におけるシルエット線分である。

(1) 第1曲線分の探索

円、楕円については、中心点から各座標軸に平行に縦軸と横軸を引き、4つの区域に分け象限とよぶ。まず、各面図で2つの象限に連続して現れる曲線を探索し、2象限の境界において分割して新たに点(境界点)を設定する。これを第1曲線分と呼び、探索の起点とする。また、第1曲線分の存在する面図を第1面図、境界点が端点として現れる面図を第2面図、線分上の点として現れる面図を第3面図と定義する。第2面図上では、シルエット線分が実線となって現れ、線分を第2線分と呼ぶ。第3面図上では、シルエット線分が生成され、線分を第3線分と呼ぶ。

(2) 第2線分の探索

まず、第1、第2面図の共通座標において、第1曲線分上の境界点と同一の座標値をもつ点を第2面図から探索する。そして、この曲線分の同心円上にある曲線分についても同様の点を見つける。

次に、第2面図上において、第1図面上の境界線に垂直に交わる直線分の端点と共通座標で同一の座標値をもつ点を探索する。

これら第2面図上の点同士を接続して、各曲面種を決定するための第2線分を同定する。

(3) 第3線分の探索

第1、第3面図の共通座標において、第1曲線分の開始点、終了点と同一の座標値を持つ点を第3面図から探索する。これらの点同士を接続して、各曲面種を決定するための第3線分を同定する。

(4) シルエット稜線の復元

まず、第1～第3線分を組み合わせ、曲面の種類を決定する。

次に、第1面図上の境界点と第2面図上の端点、第1面図の両端点と第3面図の両端点からそれぞれ頂点を復元する。そして、これらの頂点を基準に各線分をそれぞれ逆投影し、稜線を復元する。この稜線がシルエット稜線の候補となる。

最後に、各線分との照合の結果、曲面として認識された場合、先に求めたシルエット稜線の候補をシルエット稜線として決定する。

(5) シルエット線分の生成

(4)で得られたシルエット稜線を第1面図、第3面図に投影し、現れた線分がシルエット線分である。

実施例

図6に、シルエット線分の自動生成の実施例を示す。

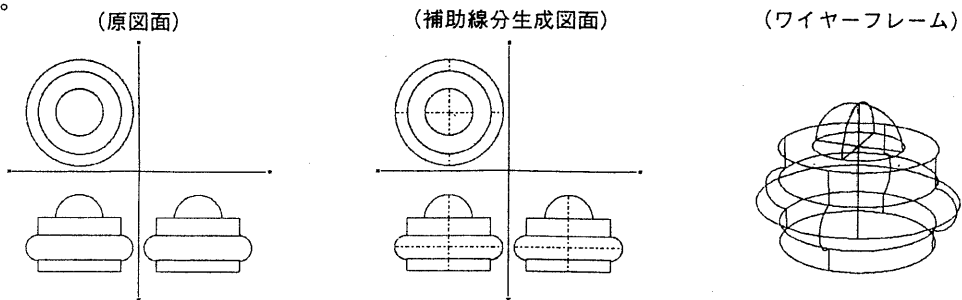


図6 シルエット線分の自動生成の実施例