間違い指摘機能を含む立体復元処理*

内山幹雄** 増田 健*** 新谷裕之† 山口富士夫††

Reconstructing a Polyhedron System with a Mistake-checking Mechanism

Mikio UCHIYAMA, Takashi MASUDA, Hiroshi NIIYA and Fujio YAMAGUCHI

This paper describes the method of reconstructing a polyhedron system with the function of checking mistakes. This function is aimed to find and check mistakes on the three orthographic views, and to inform the mistakes to a user. At the same time, this mistake-checking function has been united to the polyhedron reconstruction system. As a result, we can use this system efficiently. Finally the effectiveness of the mistake-checking mechanism has been confirmed with respect to many types of polyhedra.

Key words: mistake-checking function, polyhedron reconstruction, three orthographic views

1. はじめに

三面図からの立体復元方法は、大きく分けて2つに分類される。1つはプリミティブの組合せとして立体認識する方法、もう1つはワイヤフレームモデルの段階を経て、立体を復元する方法である。この両者において共通して言えることは、“入力された三面図に間違いを含まない”という前提条件が存在することである。しかし、人間が三面図を入力する以上、作成段階で何らかのミスをおこすことが考えられる。こういった三面図への入力ミスに対する対処法として、実際に立体復元処理の段階で、ユーザに間違いをフィードバックするシステムとして、間違い指摘機能を提案する。

この分野において坂田ら²⁾は検図処理として、線分と立体の稜線との拘束条件を用いることで、線分の過不足を認識するようにしている。しかし、検図システムが立体復元処理と別であること、また、判断要素による段階的処理であることにより、処理時間の増大につながっている。増田ら³⁾は少なくとも1つの投影図は正しいという条件の下、セル分割による論理和を取ることで、間違い検出を行っている。しかしこの際、論理和を求めることに起因して、本来設計者が意図しなかったであろうと考えられる間違いをも指摘することになる。またすべての投影図に間違いが含まれる場合には適応できないという問題を抱えている。

本論文では、従来の立体復元処理に間違い指摘機能を導入し、処理の統一化を図ることにより、ユーザフレンドリなシステムの構築と処理の能率化について述べる。処理の概要としては、間違いを含んだ三面図から実際に入力者が意図したであろう立体を復元し、それを三面図に再投影することで、入力三面図の間違いをユーザにフィードバックできるようにした。論文の流れとしては、2章で立体復元機能の説明、3章で間違い指摘機能の意義、4章で間違いの分類、5章で間違い指摘処理の流れ、6章で仮定順序制御、7章で実験結果、8章で考察を述べる。

2. 立体復元機能

本論文で提案する間違い指摘機能の手法は、立体復元処理を用いて行う。そこで我々の研究における自動立体復元処理について簡単に説明する(図1参照)。

Fig.1 System of reconstructing a polyhedron

自動立体復元処理は大きく2つの処理段階に分けることができる^{***}。第1は、2つの投影図のすべての組合せにおいて線分をそれぞれ奥行き方向にスウィープさせることにより、3次元頂点、稜線を生成し(これを線分スウィープ法と呼ぶ)、この3

^{***} 本論文に用いる用語を次のように定義する
 線分: 三面図上の直線要素
 領域: 線分に囲まれた部分
 稜線: 3次元空間の直線要素
 面分: 3次元空間の面要素
 実: 正解立体の構成要素
 虚: 正解立体以外の候補立体の構成要素

* 原稿受付 平成11年2月8日
 ** 学生会員 早稲田大学大学院(東京都新宿区大久保3-4-1)
 *** NTTアクセス網研究所(茨城県那珂郡東海村白方白根162)
 † セイコー精機(習志野市屋敷4-3-1)
 †† 正会員 早稲田大学理工学部

次元稜線を用いて3次元面分の生成を行う。これが中間図形生成と呼ぶ処理である。第2は中間図形の面要素を組み合わせるとともに、入力三面図との対応を取ることで、外形線、隠れ線の存在を確認しながら物体を見出す。これを虚実判定処理と呼ぶ。

我々は、以上のような自動立体復元処理に、矛盾の検出や状態値の局所伝播を効率的に利用した探索手法を導入し、効率的な処理を実現している。詳細に関しては文献1)を参照されたい。

3. 間違い指摘機能の意義

自動立体復元処理における問題は、その処理の始めもしくはその処理の途中でごく一部の不都合が生じたならば、それが後の処理に進むにしたがって全体に拡大してしまうことである。例えば立体復元処理において三面図情報に何らかの問題があるならば、処理途中で不都合が生じ処理が破綻する。

このことから立体復元を行う研究は三次元立体を完全に表す、間違いのない三面図の入力が前提条件として必要となる。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ことから、単純なミスや線分の線種の違いなどの図学的なミスが伴う可能性がある。

従来の立体復元処理では、三面図が正しいかどうか分からない場合には、正解立体が出力されるか何も出力されないかの対極的な結果のどちらかに至り、その中間がない。そこで、立体復元処理をモデリングシステムとして応用するためには、間違いがあった場合に修正のヒントとなる何らかの情報をフィードバックするシステムとすることが望まれる。機械部品などの複雑な三面図の入力において、もし間違いがあるならば、ヒントを参考にすることで修正の際の無駄な時間を省略することができる。

4. 間違いの分類

処理途中で不都合を起こす三面図について述べる。三面図上に現れる間違いを大別すると、以下の4種類に分類することができる。

間違い①：線分の対応のずれ

投影図上の点の座標値の対応が、他の投影図との間で正しく取れていないことによる間違い。

間違い②：必要な線分の不足

対応する立体を表すために本来必要な線分が描かれていないことによる間違い。

間違い③：線分の過剰

対応する立体を表すために本来必要でない線分が描かれていることによる間違い。

間違い④：線分の線種の違い

本研究では、投影図上に描かれている線分が隠れ線であるかないかの間違いに限定する。

これら4種類の間違いの相違点を考察する。①の間違いは、入力レベルの間違いである。②～④は図学レベルの間違いであり、そのうち、③と④が入力された三面図データの修正が必要であるのに対し、②は入力された三面図データ以外の線分を自動的に補う必要がある。

5. 間違い指摘の流れ

本論文で提案する間違い指摘処理の流れを各段階ごとに説明する。

図2から分かるように、間違い指摘処理は、自動立体復元処理をそのまま用いて行っている。これにより、処理の簡潔化、効率化を達成している。

Fig.2 The mistake-checking function within a polyhedron reconstruction system

(1) 線分スウィープ法による不足線分の予想

これは、間違い①、②に対応するために、三面図上に不足していると思われる線分を予想する処理である。線分スウィープ法を応用して、2つの投影図から中間図形を生成し、これを残りの投影面に再投影することにより不足線分を予想する。

スウィープを行う2投影面に共通する軸を垂直軸と考えた場合、次のような理由から、通常は水平でない線分どうしに限って線分スウィープを行う。

- ・ 水平な線分同士は、スウィープにより稜線候補を生成しない。
- ・ すべての稜線候補は、非水平な線分の組合せのみで生成できる。水平線を用いたスウィープにより生成された稜線候補は、他の稜線候補に重複するか三面図に対応しないかのどちらかである。

三面図上の線分に欠落があった場合、非水平線同士のスウィープでは生成できない稜線候補が現れるが、このような稜線候補を水平線と非水平線とのスウィープを行うことで発見できる場合がある。そこで間違い指摘を行う際には、水平線と非水平線とのスウィープによる稜線候補生成をあわせて行うことにする。

三面図に再投影の際、元の線分に重ならない線分を新たに書き加える。普通の線分は「実線分」「隠れ線分」の属性を持つが、書き加えられた線分には「仮定線分」の属性を与える。「仮定線分」とは“入力された投影図には対応する実線分が存在しない線分”である。

(2) 中間図形生成

修正された投影図をもとに、通常の方法（非水平線同士のスウィープ）で中間図形を生成する。

(3) 矛盾箇所を見つけながらの虚実判定

通常、虚実判定は「立体として正しいか」「三面図と対応が取れているか」の2条件をもとに正解立体を探し出す。間違い指摘を行う場合、三面図が間違っている以上「三面図と対応が取れているか」という条件は使えないので、「立体として正しいか」という条件のみで虚実判定を行う。このとき三面図との矛盾箇所をカウントし、記録することで、一定数以上の間違いを持つ立体を排除する。

仮定線分に対応する稜線がある場合、これも矛盾としてカウントする。またその仮定線分を矛盾箇所として表示することにより、間違い①、②を指摘することができる。

(4) 間違い箇所の少ない順に正解立体を表示

複数の間違いがあり、入力者の意図した間違いが指摘されるとは限らない場合もある。そこで一定数以下の間違いを持つパターンが複数ある場合、間違い箇所の少ないものがより正解に近いものと考え、間違い箇所の少ない順に表示することにする。ここで、もし間違い箇所が多すぎて正解立体が得られない場合は中間図形を表示して間違い指摘に替える。

6. 仮定順序制御

自動立体復元処理に上述の間違い指摘機能を含めたままでは、処理能率が大きく低下する。そこで以下に処理の効率化について、検討する。

虚実判定を行う際、面分候補によって状態値の局所伝播の起こりやすさに差があることが知られている。この性質を利用し、局所伝播を起こしやすい面分候補から虚実判定を行うように並べ替えることにより、虚実判定の能率を上げることができる。これを仮定順序制御と呼ぶ。

仮定順序制御の仕方としては、以下に示す方法が考えられる。すなわち、中間図形において、

- ① 「接続する面分候補が2枚」である稜線候補を多く持つ面分候補から仮定を行う。
- 「接続する面分候補が3枚以上」の場合はその数がより少ない面分候補から仮定を行う。
- ② 「対応する面分候補の数が少ない領域」に対応する面分候補から仮定を行う。
- ③ 「その面分候補に対して稜線を介さずに交差する」面分候補数の多いものから仮定を行う。
- ④ 投影面に近い面分候補から仮定を行う。
- ⑤ 面積の大きい面分候補から仮定を行う。

このような拘束条件の仮定順序制御を組み合わせ、間違い処理時の処理効率の向上を図る。それぞれについて以下に説明する。

6.1 仮定順序制御の拘束条件

- ① 「接続する面分候補が2枚」である稜線候補を多く持つ面分候補から仮定を行う

「接続する面分候補が3枚以上」の場合はその数がより少ない面分候補から仮定を行う

ある面分候補に2枚の面分候補が接続している場合、どちらか片方の面分が“実”に仮定されればもう一方も“実”になり、“虚”に仮定されればもう一方も“虚”になる。これは「実稜線には必ず2枚の面分が接続する」という、立体の拘

束条件によるものである。このような稜線を多く有する面分は、効率よく状態値の局所伝播を行うことができる。

また、接続面分が3枚以上の場合、その数が多いほど、状態値の局所伝播の効率が悪くなることが確認された。そのため、その数の少ない面分候補から仮定を行うことにする。

- ② 「対応する面分候補の数が少ない領域」に対応する面分候補から仮定を行う

「領域に対応する面分候補は偶数枚」という、立体の拘束条件によるものである。ある領域に対応する未定面分の枚数が1枚になったとき、実面分の枚数の奇偶によりその状態値が決定する。対応する面分候補が少ない方が未定面分数を減らしやすいため、局所伝播を起こしやすいと考えられる。

- ③ 「その面分候補に対して稜線を介さずに交差する」面分候補数の多いものから仮定を行う

「対応する線分の存在しない交差面分候補群において、実面分は高々1枚に限られる」という、立体の拘束条件によるものである。このような面分候補から仮定を行うことで広く状態値の局所伝播を行うことができる。

Fig.3 An example of a pair of intersecting candidate faces without a common candidate edge

- ④ 投影面に近い面分候補から仮定を行う

投影面に近い面分候補ほどより多くの稜線を隠す可能性がある。このため、稜線候補の投影状態を調べ、三面図と対応を取る際に、状態値の局所伝播が起こりやすいと考えられる。

- ⑤ 面積の大きい面分候補から仮定を行う

面積の大きな面分ほど「多くの面分と接続する」、「多くの面分と交線なく交わる」、「多くの稜線候補を隠す」ことが多い。そのためこのような仮定順序制御を設ける。

6.2 仮定順序制御の実験

機械部品として使用頻度が高いと考えられる角すい、角柱を用いて実際に仮定順序制御を用いた立体復元処理と用いない場合における処理の所要時間を実験により求めた（表1参照）。

実験において、比較的単純な立体の場合では仮定順序制御を用いた場合、逆に処理時間がかかることがあった。これは仮定順序制御の拘束時間に適応させる際に要する処理時間に起因していると考えられる。逆に立体が複雑になる程、仮定順序制御を用いると、より高速に処理を行うことができた。このことは、さらに処理時間のかかるような立体復元処理において、より効果を挙げることができることを意味すると考えられる。なお、表1には10回の測定平均を示した。

6.3 その他の間違い指摘機能

三面図上の間違いにおいて、今まで述べた間違いとは別に、比較的容易に発見できる間違いがある。そこでそのような間違いを発見し、指摘する機能を設けた。これを「簡易間違い指摘機能」

Table 1 The time required for polyhedron reconstruction

	Unit : s	
	The case with the assuming order method	The case without the assuming order method
4-sided pyramid	0.0225	0.0225
8-sided pyramid	0.0225	0.0250
12-sided pyramid	0.0225	0.0275
16-sided pyramid	0.0325	0.0375
20-sided pyramid	0.0325	0.0400
4-sided prism	0.0225	0.0220
8-sided prism	0.0225	0.0245
12-sided prism	0.0225	0.0265
16-sided prism	0.0230	0.0275
20-sided prism	0.0240	0.0370

(digital 製 CELEBLIS6200 PentiumPro 200MHz 使用)

と呼ぶ。

(1) ぶら下がり線分

一方あるいは両方の端点で他の線分と接続していない線分を「ぶら下がり線分」と呼ぶ。ぶら下がり線分は通常三面図上に現れるはずのない線分であるので、これを発見し警告する機能をつけた。

(2) 外形部分での対応のずれ

外形部分の対応のずれは、各投影図上における点の座標値の最大値、最小値を知ることにより、比較的容易に知ることができる。そこでこのような対応のずれを発見し警告する機能をつけた。

(3) 輪郭線分の線種の違い

正しい三面図なら、輪郭部分はすべて実線により構成されている。もしも輪郭部分に隠れ線が含まれる場合、何らかの間違いを起こしているため、これを発見し警告する機能をつけた。

7. 実 験

実際に、2章で示した4つの間違いを含む三面図から、間違い指摘機能を含んだ立体復元処理を行う。

- ① 必要な線分の不足による間違いを含む場合
- ② 線分の過剰による間違いを含む場合
- ③ 線分の対応のずれを含む場合
- ④ 線分の線種の違いを含む場合

それぞれの実験結果を図4～7に示す。図4では、三面図上で不足していた線分が、間違い指摘機能によって白い太線として指摘されている。同様に図5では、過剰である線分が白い太線として示されている。図6では三面図上の線分の対応を取ることで間違いを発見している様子が示されている。そして最後に、図7では三面図上での線種の違い（実線と隠れ線の違い）を指摘している。

各々の図から明らかなように、これら4つの間違いに対して、ユーザにわかりやすい形で、間違いを指摘することができた。また当然ながら、上記の4つの間違いが混在する場合においても、的確に間違い指摘を行うことを確認した。

8. 考 察

我々は、導入した間違い指摘機能により、間違いの少ない三

Fig.4 A case of a line insufficiency mistake
(the thick white line is added by the function)

Fig.5 A case of an excessive line mistake
(the thick white line is checked by the function)

面図からの立体復元は容易に行えることになった。それと同時に、設計途中の段階で、間違い指摘機能を用いた立体復元を行うことが可能となったため、段階的に確認しながら作図を行うことが可能となった。

また、間違いを含んだ三面図から立体を復元させた場合、最悪でも中間図形が表示されるため、ユーザはその中間図形を参考に、何らかの対応を行うことが可能となった。

今後、本研究における間違い指摘機能に改良を加えるとすれば“間違いを含む三面図から出力された中間図形にユーザが改良点を指摘できるようにする”ことである。出力された中間図形のどの部分が入力しようとしたデータと食い違っているかをユーザが指摘できるようにすることで、間違い箇所の検出が容易にな

Fig.6 A case of a correspondence mistake between views
(the thick red line is written by a user and the white thick line is checked by the function)

ると考えられる。これにより、出力された形状に対して局所操作などの別のモデリング操作を加えることで所望の立体を一意に作り出すことが可能になると考えられる。

参 考 文 献

- 1) 増田 健, 山口富士夫：三面図データからの自動立体復元, 精密工学

Fig.7 A case of a line type mistake
(the thick white line is checked by the function)

- 会誌, (1992) 1713.
- 2) 坂田泰之, 斎藤義夫：三面図による立体認識過程に関する研究, 1993年精密工学会春季大会学術講演会講演論文集, (1993) 275.
- 3) 増田 宏, 沼尾雅之：不完全な三面図からのソリッドモデルの合成, 情報処理学会第47回全国大会論文集 (1993) 295.
- 4) 新谷裕之, 山口富士夫：三面図における間違いの指摘, 1996年精密工学会春季大会学術講演会講演論文集, (1996) 503.