

探索手法による三面図からの多面体復元

西原 清一[†], 西田 淳[†], 池田 克夫^{††}
[†]筑波大学 電子・情報工学系 ^{††}京都大学 情報工学科

本論文では、三面図をファクシミリ装置により入力し、それに矛盾しない多面体を復元するシステムについて述べる。ここでは、線分抽出と多面体復元についての新しい手法を提案する。線分抽出は、二値画像によって表された各線分の中心軸に、プローブという微小線分を当てはめることによって行う。これにより、二値画像のノイズの除去あるいは細線化といった前処理を必要としなくなる。多面体の復元は、従来多く用いられてきたワイヤーフレームを基にしたアプローチとは異なる、面を基本にしたアプローチを用いている。この多面体復元の方法は、可能な面の中から解となる組合せを全て求める組合せ探索手続きである。

Restoring Polyhedral Scenes from Engineering Drawings by Using Combinatorial Search Method

Seiichi NISHIHARA[†], Jun NISHIDA[†], Katsuo IKEDA^{††}

[†] Inst. Inf. Sci. & Electr., Univ. of Tsukuba, Tsukuba, Ibaraki.

^{††} Dept. Information Science, Kyoto University, Sakyo, Kyoto.

This paper deals with a system that restores solid models, or a set of polyhedrons, consistent with a given set of three orthographic views which are input by using a facsimile device. We lay main stress on the description of two novel techniques: the line segment extraction method and the polyhedron restoration method. The line segment extraction is performed by using a probe, a small segment of line, which moves along the central axis of each line segment represented by a binary image. Therefore, the present method do not require any preprocessing such as noise elimination or thinning of the given binary image. The polyhedron restoration method employs a face oriented approach, rather than the conventional wire-frame oriented approach. The present method is basically a combinatorial search procedure that finds all of the legal combinations of possible faces.

1. はじめに

コンピュータグラフィックスの分野において、三次元モデルを計算機内に生成することは、基本的かつ重要な問題である。三次元物体のモデリングに三面図を利用することは、この問題に対する有効なアプローチである。今日まで、三面図から三次元シーンを復元するために、ワイヤーフレームの位相的性質を用いる方法 [1] [2], 虚物体排除法 [3], 対話処理による方法 [4], 拘束関係を用いる方法 [5] など数多くの研究がなされてきた。最後の拘束関係を用いる方法は、完全なものとはいえないが、入力画像を構成する各線分が、相互に拘束し合う他の線分と局所的拘束関係を形成するという性質を用いているという点で重要である。この局所的拘束関係は、整合ラベリング問題 (CLP: Consistent Labeling Problem) [6] によって表され、与えられた三面図に一致する全ての可能な三次元シーンは、この CLP を解くことによって得られる。

本稿では、二値化データとして表された三面図を、それに合致するソリッドモデルに変換するシステムについて述べる。ここでは特に、復元過程で用いる 2 つの主なアルゴリズムについて報告する。まず、線分抽出アルゴリズムは、与えられた二値画像からその画像中に現れる全ての線分を抽出する手続きである。次に、多面体復元アルゴリズムは、三次元物体を

構成する複数の面の組合せを探索するという組合せ探索アルゴリズムである。このアルゴリズムでは、三面図の解釈に際し、ワイヤーフレームよりも面が多面体を定義し、処理するのに主要な役割を果たすという考えを基礎に置いている。

2. 問題

ここでは、三面図の解釈における問題の具体的な定義を、解釈する画像や復元するシーンの構造を明確にすることによって与える。

2.1 三面図

図 1 に示すように三面図は平面図、正面図、(右)側面図の 3 つの投影図からなるものとする。また、各面図は基線によって分けられている。各面図に含まれる幾何学的要素は直線分のみであり、中心軸を示す鎖線や数字のような補助要素は含まれないものとする。線分は、実線と破線の 2 種類のみを扱い、隠線は破線によって表すものとする。各線分は両端点によって定義する。また、連続する線分の並びによって囲まれた部分を領域という。特に、境界線分がすべて実線で、かつ、その領域中に実線を含まないとき、その領域を単純領域と呼ぶ。

2.2 三次元シーン

本システムによって復元することのできる三次元シーンは、1 つ以上の多面体である。各多面体は、複数の面によって境界付けられた三次元空間の連結部分である。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付けられた有向平面である。頂点は、3 つ以上の稜線が会う点である。シーンには、他に接する面を持たない浮遊面や、2 枚の面が背中合わせに密着しているなどといった体積 0 の部分

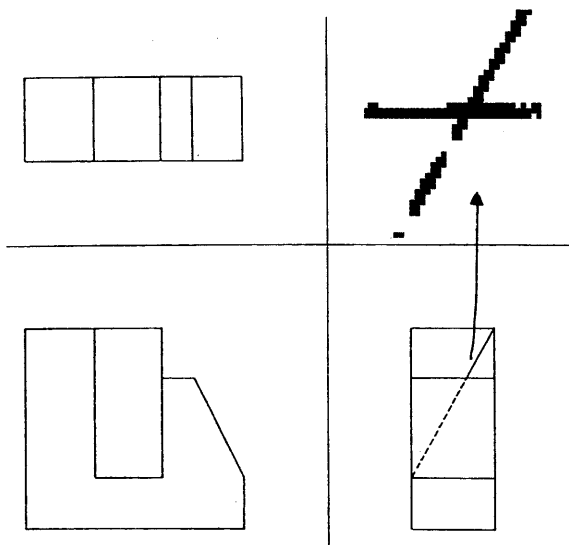


図 1. 入力三面図

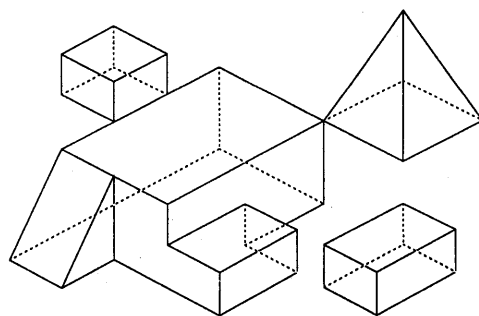


図 2. 三次元シーン

は存在しない。但し、各多面体は稜線や頂点を共有してもよい。つまり、‘無重力の三次元空間に実現し得る多面体世界’であればよい。図2は、可能なシーンの例を示している。なお、正射影により多面体の三次元要素、つまり、頂点、稜線、面は、それぞれ、点、線分または点、領域または線分、として三面図上に現れる。

2.3 復元方法

図3に処理の概要を示す。本稿で詳述するフェイズ2とフェイズ3は、その各々がいくつかのステップによって成り立っている。

[フェイズ2]

フェイズ2は、4つのステップから成り立っている。ステップ1では、全ての可能な線分の候補が、3章で述べるアルゴリズムを使って抽出される。ステップ2では、ステップ1からの線分データの集合を受け取り、端点の補正を行う。つまり、互いに近くに存在する端点を1つの点に統合するという操作を繰り返す。これにより補正の行われた端点は、最終的に、三面図に関する位相的性質を保証するように修正される。ステップ3では、入力画像の正立化を行う。ここでは、ファクシミリ装置の機械的な性質により起こる縦と横の標準化ピッチの差を補正し、さらに、基線の傾きも矯正する。最後にステップ4では、各面図間で一致する点を調べることで各点の座標値

を調整する。

[フェイズ3]

フェイズ3は、主として2つのステップからなる。第1のステップでは、可能な頂点、稜線、面をすべて復元する。第2のステップでは、もとの三面図に一致する多面体の集合を形成し得る全ての面の組合せ、つまり、全ての解を探索する。第1のステップでは、まず頂点の復元を行う。平面図上の点 $u(z_u, x_u)$ 、正面図上の点 $v(x_v, y_v)$ 、側面図上の点 $w(y_w, z_w)$ の3点が頂点 $p(x_p, y_p, z_p)$ の投影点だとする。すると、 $x_u=x_v, y_v=y_w, z_w=z_u$ が成り立つ。従って、頂点の復元は、各面図間で点の対応をとり、上記の3つの等式を同時に満たす点を頂点とする。同様な面図間の性質が、稜線や面を復元するときにも適用される。ここでは、詳細は省略する。なお、このステップは、上記の性質をCLPに置き換える方法[6]によっている。第2のステップは、本質的には組合せ探索アルゴリズムである。4章で詳述するが、我々の提起する復元方法は、ワイヤーフレームを基にしたアプローチではなく、面を基本にしたアプローチであるという点が特徴的である。

3. 線分抽出

ここでは、二値画像から線分を抽出する方法について述べる。2.2節で触れたが、線分は、両端点と、実線/破線という直線の種類によって決定される。しかし今後は、線分とは端点以外に点を含まない、つまり、端点以外の場所において分岐する点が存在

しない直線であると定義する。従って図1の側面図を例にとって考えると、図4に見られるようにa,b,...,mという13個の線分に分割される。図4では、1~9の9個の点も示されている。これらの点の次数は全て2以上である。

今日までに線分抽出のための数多くの手法が提案されてきた。本稿の方法は、ノイズ除去や細線化といった前処理を必要とし

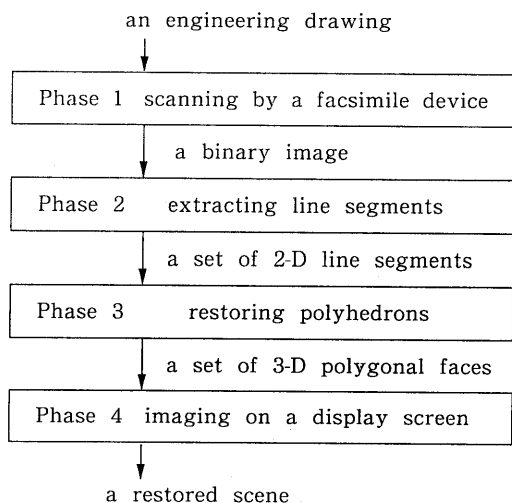


図3. 処理の概要

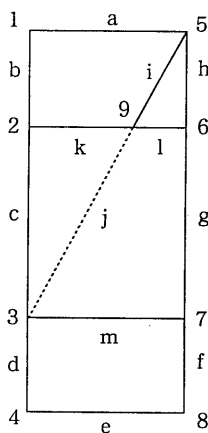


図4. 点および線分

ないので、その分全体の処理時間は短縮される。図1に見られるように、線分は数ピクセルの幅を持っているものとする。

線分抽出のアルゴリズムを図5に示す。①では、未処理の線分候補上にプローブつまり微小線分を正しい位置に置くという操作を行う。つまり、それが線分をトレースしていく上での開始位置となる。

図6は、開始位置に置かれた初期プローブを示している。ここでは、プローブcとプローブdだけが正しく置かれている。他のプローブは正しく置かれていない。プローブaとプローブbに関していえば、次数が2ならびに3である点P、Qに近づくすぎている。プローブeに関していえば、線分の中心軸上に位置していない。

次に、正しいプローブ位置がどのようにして決定されるかについて述べる。まずプローブが正しく置かれなかったとする。しかし、そのプローブが少なくとも線分の一部であることは確かである。①では、正しく置かれていないプローブを手がかりとして、代替りの新たなより正しいプローブを求めている。図7において、 $\overline{p_1 p_2}$ は長さrの不適切プローブである。そこで p_2 を中心に半径rの円を描く。さらに'l'なるピクセルからなり、 p_1 から最も遠くにある弧 $\widehat{st}$ を求める。弧 $\widehat{st}$ の中心ピクセルを p_3 とすると、正しい位置にあるプローブ $\overline{p_2 p_3}$ が得られる。

次に図5の②では、一線分の抽出を行っている。それには、まず線分候補に沿って1つずつプローブを延ばしていく。従って、プローブは画像の変動に直接影響される。この種の局所的なふらつきを吸収するために、我々は、連続する2つのプローブの符号付き角度の差の総和である集積誤差という概念を導入した。これにより、一続きのプローブが直線を形成することを保証するための必要条件として

(1) 任意の連続するプローブのなす角は、前もって与えられた閾値 α_0 以下である。

(2) 集積誤差は、常に0に充分近い値を保つ。

$$\sum \alpha_i \approx 0$$

という2つの条件を与えた。

```

begin
  while (other segments not yet visited are remaining) do
    begin
      place a probe at an appropriate position ;      --- ①
      in both longitudinal directions of the probe do
        repeat
          make the probe move a step incrementally --- ②
        until (an end point is encountered, or linearity
              of the probe movement is violated)
    end
  end.

```

図5. 線分抽出アルゴリズム

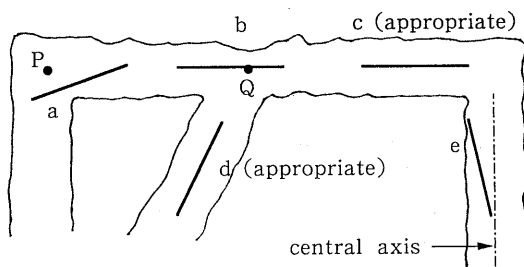


図6. 初期プローブ

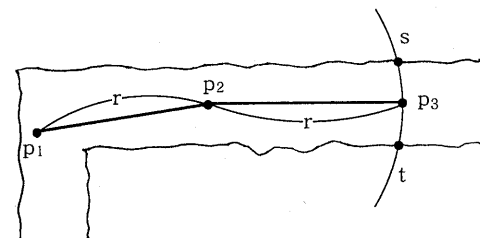


図7. 正しいプローブ位置

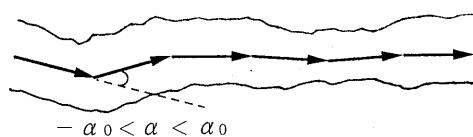


図8. プローブの延長

4. 多面体復元

多面体復元は、多面体を構成する全ての可能な面の組合せを求める一種の組合せ探索問題である。ここでは、以下の三次元物体に関する基本的性質を基に復元アルゴリズムを開発した。

「三次元空間における任意の閉曲線は、三次元物体の表面と偶数回交差する」 ——— (*)

さらに、三面図に一致した三次元多面体シーンを構

築するための必要十分条件は、以下のよう
に要約できる。

条件1. 三次元物体が存在するための
必要十分条件

(Rule1) 投影図に垂直な直線は、面と偶
数回交差する。また、最初に交差する面
の向きは、視線方向を向いている。

(Rule2) 任意の稜線について、その稜全
体または一部を共有する面の個数は、2
以上の偶数個である。また、稜の周囲を
十分小さな閉曲線に沿って進むと、面の
向きは交互に逆向きである。

条件2. 復元された多面体とものの三
面図との一致

(Rule3) 隣接する2つの領域内に設定し
た垂線のうち少なくとも一方は、面と交
差する。

(Rule4) 破線を境界とする領域に対応す
る面は、その破線位置において、面を隠
蔽する視点により近い他の面が存在する。

条件3. その他の条件

(Rule5) 視点に最も近い面、つまり他の面によって
全く遮られていない面に対応する領域が、各面図に
少なくとも1つ以上存在する。

ここで、条件1は、三次元物体に関する基本的性質
(*)を多面体に適用したものである。従って、復元
された多面体は三次元空間内に実際に実現できると
いう意味においてすべて正しい。また、条件2は、も
との三面図と復元されるシーンとが一致することを
保証するための必要十分条件である。条件3は、厳密
に言えば必ずしも正しいというわけではない。しか
しながら、探索空間を縮小させるという意味におい
ては実用的なルールである。

以上3つの条件、5つのルールによって探索は進め
られる。まず、条件3によって探索木の開始ノードを
決定する。条件1は、現ノードから次のノードを見つ
け出すのに使われる。条件2は、現ノードがゴール
ノードであるかどうか、すなわち解であるかどうか
を判定するのに用いられる。なお、探索木において
各ノードは、全ての可能な面の部分集合を表してい
る。

```
begin
  for each (simple area  $\alpha$ ) do
    for each (face  $\phi$  corresponding  $\alpha$ ) do
      [E := {e | e (<  $\phi$ )}; F := {}];
      search (E, F, { $\phi$ })
    ]
end.
```

(a) Main program.

```
procedure search (E, F, C);
  [if E= {} then [if (C forms a valid scene)
    then (C is a solution)] ---③
  else
    [for each  $\varepsilon \in E$  do
      for each (eligible S ( $\subseteq \{f | f (> \varepsilon)\}$ )) do ---④
        branch (E, F, C, S,  $\varepsilon$ )
      ]
    ]
```

(b) Node check and expansion procedure.

```
procedure branch (E, F, C, S,  $\varepsilon$ );
  [F := F  $\cup$  { $\varepsilon$ }; S := S - C;
  C := C  $\cup$  S;
  E := E  $\cup$  {e (< f) | f  $\in$  S} - { $\varepsilon$ }; ---⑤
  search (E, F, C)
  ]
```

(c) Branch operator.

図9. 復元アルゴリズム

図9は、多面体復元のための探索アルゴリズムを
示している。ここではまず、条件3を用いて探索開始
ノードを決定する。実際、メインプログラムでは、フ
ェイズ3の第1のステップで求められた面の中から
ある候補面を選び出す。アルゴリズム中の表記 e
($< f$) は、稜線 e が面 f の境界の一部を形成してい
ることを表す。逆に f ($> e$) は、面 f が稜線 e に接続
する面であることを示している。また、手続き
'search' のパラメータ E, F, C はそれぞれ未処理稜
線の集合、処理済稜線の集合、現時点での構成面の
集合を示している。従って C は、探索木の現在の状
態を表しているといえる。

③では、現在の状態がゴールであるか否か、言い
換えれば、その時点での構成面の集合 C がもとの三
面図に矛盾しない適格なシーンを形成しているかど
うかを判定する。④では、次に探索を進めるべき状
態を生成している。ここで S は、条件1を満足する適
格な面の集合である。稜線 ε は、 ε に接続する面に
対する適格な面の接合処理が行われた後、手続き
'branch' において集合 F に加えられる。さらに⑤で

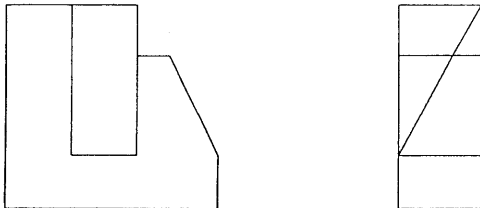
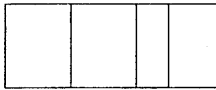


図 10. 中間結果

は、面を接合することにより生じた新たな稜線をEに加えている。なお、このアルゴリズムにおける2つの手続き search と branch は互いに間接的な再帰的手続きになっており、全体として縦型探索による木探索を実現している。

5. 実験

図5のフェイズ1とフェイズ4は、Perkin-Elmer 3220 上に Fortran 77 を用いて実装した。フェイズ2とフェイズ3は、Sequent社の Balance 21000 上にCを用いて実装した。例として、図1の三面図について処理を行った。図3のフェイズ2までの処理、つまり線分抽出を行った後に得られた中間結果を図10に示す。図10において、入力画像側面図中の破線は実線で描かれているが、線分の型はデータ中に正しく破線として求められている。最後に、復元されたシーンを図11に示す。図10に示した中間結果の生成を含む多重プログラミング環境下での全ユーザタイムは4分24秒であった。

6. 結論

手書き三面図を認識、解釈し、三次元ソリッドモデルを復元するシステムを開発した。本稿では、線分抽出法ならびに多面体復元法について主に報告した。一般に、三面図から復元されるモデルは1つとは

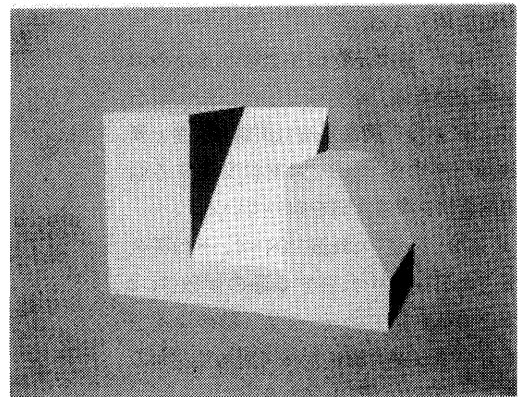


図 11. 実験結果

限らない。多義解釈が存在するような場合、本システムは、順次、全てのモデルを復元し表示する。

[文献]

- [1] Markowsky, G., Wesley, M. A.: Fleshing out wire frames, IBM J. Res. Develop., 24, 5 (1980), 582-597.
- [2] Idesawa, M.: 3-D model reconstruction and processing for CAE, 8th ICPR (1986), 220-225.
- [3] 佐々木, 伊藤, 鈴木: 三面図からの物体自動合成のための線形疑似ブール代数解法, 情報処理学会論文誌, Vol. 28, No. 12, 1288-1297 (1987).
- [4] Yoshiura, H., Fujimura, K., Kunii, T. L.: Top-down construction of 3-D mechanical object shapes from engineering drawings, IEEE Computer, 17, 12 (1984), 32-40.
- [5] Haralick, R. M., Queeney, D.: Understanding engineering drawings, Comput. Graphics and Image Proc., 20 (1982), 244-258.
- [6] Haralick, R. M., Shapiro, L. G.: The consistent labeling problem: Part I, IEEE Trans. Pattern Anal. Mach. Intell. 1, 2 (1979), 173-184.