

画像のB-REPのためのアルゴリズム

Algorithms for a B-REP of an Image as Its Intermediate Description

富田 文明

Fumiaki TOMITA

電子技術総合研究所

Electrotechnical Laboratory

高橋 裕信

Hironobu TAKAHASHI

三洋電機筑波研究所

Tsukuba Res. Cen. SANYO Ltd.

Abstract: This paper presents algorithms to get a B-REP of an image as its low-level intermediate description. In the algorithms are image segmentation, region labeling, boundary segmentation, line and curve fitting, and virtual-line drawing. As a result, an image is represented by a set of regions, each region by a list of boundaries to a background and some holes, and each boundary by a list of boundary segments. This data structure will serve for such a higher level processing as object recognition.

1. まえがき

画像の中間表現にはMarrのPrimal Sketch^[1]などがあるが、本報告では、CAD/CAMが立体の表現法としてB-REP(Boundary REPresentation)を用いているのに対応して、画像のB-REPを画像の中間表現として利用することを提案する。図1に、画像のB-REPによる表現例を示す。すなわち、画像は領域の集合であり、各領域Rには領域の外側に対する境界(B-境界)と領域の内側の穴に対する境界(H-境界)がある。各境界は、それを構成するセグメントSのリストによって表現される。このような画像のB-REPは、画像を再構成できる程度に情報圧縮を行っている。このようなデータ構造を得るための画像処理は、画像の分割、領域のラベル付け、境界の分割、直線または2次曲線のあてはめなどである。本報告では、われわれが開発した方法を技術的なノウハウを含めて紹介する。また、画像のB-REPの応用例として、仮想線と単眼による3次元物体の認識の問題を考える。

2. 画像の分割

画像の分割の方法には大別して、エッジ検出法、領域分割法、領域拡張法がある。エッジを検出して領域を求める方法に二次微分オペレータによりゼロ・クロッシングを求める方法がある

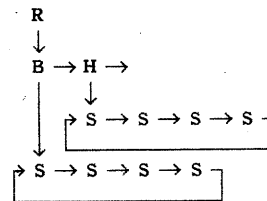
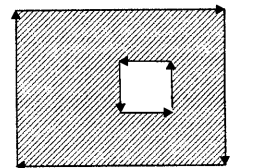


図1 画像のB-REPのデータ構造

が、そのままではほぼ一様な部分が細分化されてしまう。領域分割法の代表としてしきい値処理がある。しきい値処理は大局的過ぎて、ヒストグラムに分離できるクラスターが存在する場合か、予めしきい値がわかっている場合にしか適用できない。また、クラスターが存在してもそのクラスターを求めること自体難しい問題であり、その方法はむしろ対象に依存するので、一般的な解法はない。領域拡張法は、最も確実な方法であるが計算コストが高い。ここでは、画像の一次微分から領域を求める方法を紹介する^[11]。

まず、一次微分オペレータにより画像の各点についてエッジの値と方向を求める。

$$|G(x,y)| = \sqrt{f_x(x,y)^2 + f_y(x,y)^2}$$

$$\angle G(x,y) = \tan^{-1} \frac{f_y(x,y)}{f_x(x,y)} + \frac{\pi}{2}$$

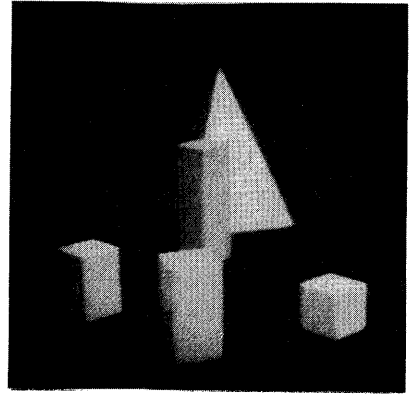
但し、 f_x と f_y は x 軸方向と y 軸方向の差分である。次に、非極大値消去処理によりエッジの値が局所的に極大でないエッジ点を消去し、エッジを細線化する。また、しきい値処理によりエッジの値の小さいエッジ点を消去する。この処理の結果残る“強い”エッジは、図2(b)のように、必ずしも連続ではない。そこで、“強い”エッジの端点を延長することにより、領域を囲む閉じた境界を求める。

まず、ラスタ走査により分岐数が1であるエッジの端点を探索する。エッジの端点に対して以下の処理を行なう。いま、図3のように、エッジの端点 e_0 を中心とする 3×3 のウィンドウを考える。そして、中心点のエッジの方向に最も近い3点(図3では、 e_5 、 e_6 、 e_7)の中で次の評価関数を最大にする点をこの端点の延長点とする。

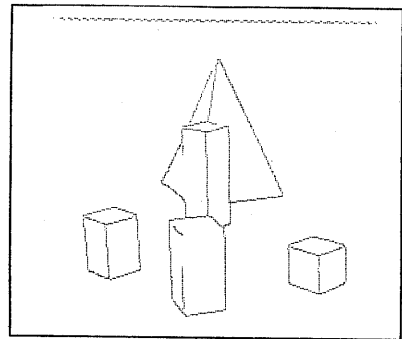
$$f(e_0, e_i) = |e_i| \cos(e_i - e_0)$$

この評価関数は、エッジの値が大きく、エッジの方向が中心点の方向と同じ点に大きな値を与える。延長点がまた端点となる場合には、この延長点を中心点として以上の延長処理を繰り返す。このエッジの延長が停止する場合には次の2通りある。

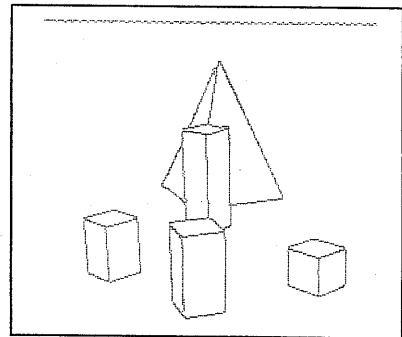
① 延長点が既に存在するエッジ点である場合は



(a)



(b)



(c)

図2 エッジ延長法による領域の抽出

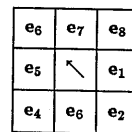


図3 エッジの端点の 3×3 のウィンドウ

延長を停止する。

②一定値以下のエッジ点が一定の長さ以上続く場合には、その延長の開始点から延長した点をすべて削除し、延長を停止する（この場合にはクラックとなる）。

延長が停止すると延長の開始点に戻り、今度は中心点のエッジの方向とは逆の方向に延長を開始し、上記と同様に延長処理を繰り返す。そして、逆方向の延長も停止すると、開始点に戻り、ラスタ走査により次のエッジの端点を捜し、以上の処理を繰り返す。その結果、図2(c)のように、領域を囲む閉じたエッジを得ることができる。

このアルゴリズムはハードウェア化が可能で、実際に部品の検査に利用されている^[3]。

3. 領域のラベル付け

エッジで囲まれた各領域を区別するために領域のラベル付けを行なう。すなわち、任意の領域Rに対してRのすべての点に同じ番号を持たせ、Rでない点はこの番号を持たせないようにする。従来のラスタ走査型のラベル付けアルゴリズムでは、2回の走査+ α を必要とする。1回目の走査で、横方向の各ラン（同じ領域に属する連続点）に仮のラベル付けを行なうと同時に、各ランの縦方向の連結関係を記録する。この記録に基づいて縦方向に連結するランを統合し（ α の部分）、2回目の走査で同じ番号に付け直す。ここでは、ラスタ走査と領域の境界追跡の併用により、ランの連結関係を記録することなく、1回のラスタ走査で領域に通し番号のラベルを付けると同時に、領域のトポロジーすなわち穴を見つけるアルゴリズムについて述べる。

いま、図4(a)のように、エッジで囲まれた（4-連結）領域はブランクb、エッジと画像の外側は0とする。そして、ラベルは1から番号順に付けるとする。まず、ラベルの最大値をN=0に初期化し、画像の一行目からラスタ走査を開始する。そして、走査点が0または既にラベルが付いている場合は素通りする。ブランクbの場合は、図5のように、走査点の1つ前と1つ上の点のラベルに基づいて3通りに分類し、次のようにしてラベル付けを行なう。

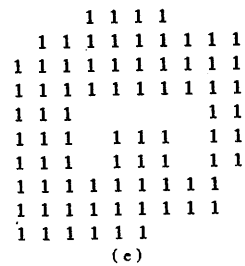
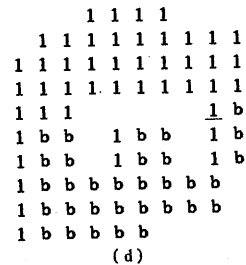
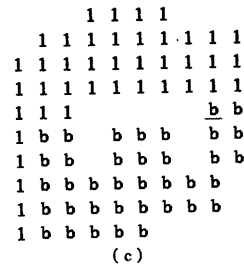
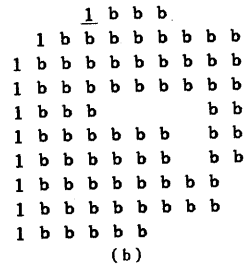
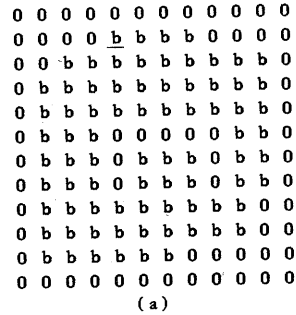


図4 領域のラベル付け

① B-境界追跡

図5(a)のような走査点(アンダーライン)は、ラベル付けされていない領域の最初の点で、領域の背景に隣接するB-内境界上の点でもある。そこで、 $N \leftarrow N+1$ とし、走査点を最初の境界点として領域のB-内境界を次のように右回りに追跡し、同時にラベル付けも行なう。いま、次のように、境界点☆から4-隣接する各点への方向コードを設定する。

	3	
2	*	0
	1	

そして、直前の境界点から現在の境界点への進行方向が*i*であるとすると、次に進むべき点を $i-1 \pmod{4}$ の方向から時計回りに探索する。そして、最初に**b**または**N**のラベルを持つ点が次の境界点となる。このとき、境界点☆にラベルを付ける条件を表1に示す。この追跡は、B-内境界を一周して最初の境界点に戻るまで繰り返される。その結果、図4(b)のように、少なくとも左側にエッジ(0)のあるB-内境界上の点にはラベル**N**が付くことになる。

② ラスター走査

図5(b)のような走査点は、領域**K**(>0)内の点であり、走査点にラベル**K**を付ける。図4(b)から図4(c)まではこの方法によりラベル付けされる。

③ H-境界追跡

図5(b)のような走査点は、領域**K**(>0)内の穴に隣接するH-内境界上の点である。(領域**K**のB-内境界上の点であるならば、①で既にラベル付けされていなければならない。)そこで、走査点を最初の境界点として領域のH-内境界を①と同様に追跡し、ラベル付けを行なう。その結果、図4(d)のように、左側にエッジ(0)のあるH-内境界上の点にだけラベル**K**が付くことになる。

以上の処理を画像の最終行まで繰り返すと、図4(e)のように、1回のラスター走査で画像内の各領域に1から**N**(領域の数)の通し番号を

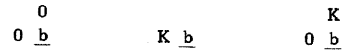


図5 走査点の近傍パターン(4-連結)

表1 ラベル付けの条件(4-連結)

- : ラベルを付ける
- ×: ラベルを付けない
- 空: どちらでもよい

		出力方向			
		→	↓	←	↑
入力方向	→				
	↓		× ×	○	
	←		○ ×	○	
	↑		○ ○	○	

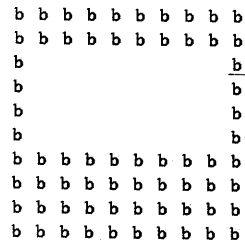


図6 幅1の部分を持つ領域

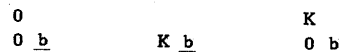


図7 走査点の近傍パターン(8-連結)

表2 ラベル付けの条件(8-連結)

- : ラベルを付ける
- ×: ラベルを付けない
- 空: どちらでもよい
- : 存在しない

		出力方向							
		→	↘	↓	↙	←	↗	↑	↖
入力方向	→								- -
	↘		× × × ×	○	-				
	↓		- × × ×	○ ○	-				
	↙		- × × ×	○ ○ ○					
	←		○ - - ×	○ ○ ○					
	↗		○ ○ - ×	○ ○ ○					
	↑		○ ○ ○ - -	○ ○ ○					
	↖		○ ○ ○ ○ -	○ ○ ○					

付けることができる。この処理は、①または③の境界追跡によりランの左端に確定したラベルを付けた後、②のラスタ走査で横方向のランにラベル付けしているのである。境界追跡中に3つの場合にラベルを付けない理由は、常にラベルを付けるようにすると、図6のように、幅が1の部分があると誤ったラベルを付れたり、穴を見つけることができない場合があるためである。

上記のアルゴリズムは領域が4-連結の場合であるが、領域が8-連結の場合も同様にラベル付けができ、図7にB-境界追跡、ラスタ走査、H-境界追跡の場合のパターンと、表2に境界追跡中に境界点にラベルを付ける条件を示す。また、このアルゴリズムもハードウェア化が可能であり、現在検討中である。

この段階で、領域のトポロジーを表わすリスト構造（領域→B-境界→H-境界→…）が生成される。

4 境界の分割

各境界を簡単な形状のセグメントに分割しそのリストを作成する。領域の外境界（エッジの部分）を領域を右側に見ながら追跡する。境界は4-連結と8-連結が混在しているので、その追跡は4-連結を優先することから図8のようになる。このようにして得られた境界の点列を分割する手順を図9に示す。

4.1 分岐点での分割

まず、境界を分岐点でセグメントに分割する。分岐点は、エッジの分岐数が2以外の点である。

4.2 屈曲点での分割

さらに、各セグメントを傾きが急変する点で分割する。通常、セグメントの局所的な傾きの変化は k -曲率によって測定し、 k -曲率の絶対値が一定値以上で局所的に極大あるいは極小な点で分割する。しかし、1つの k -曲率では良い結果が得られない場合がある。すなわち、 k が小さ過ぎるとノイズの影響を受け、 k が大き過ぎると図10のように正しい位置で分割されない場合がある。そこで、本報告では k -曲率 C_k と nk -曲率 C_{nk} の複数の曲率から曲率を次の

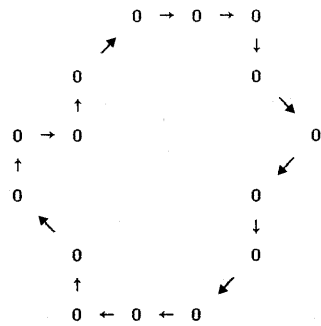


図8 境界追跡

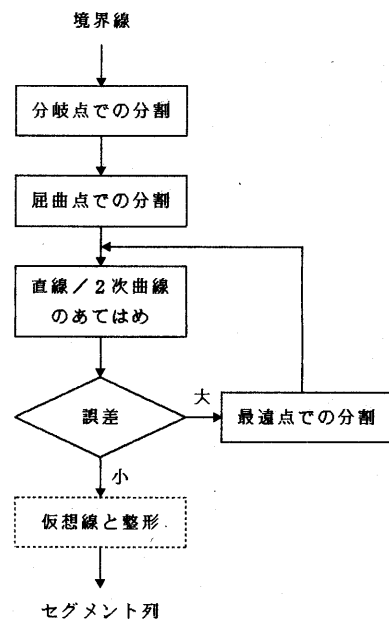


図9 境界分割処理の流れ

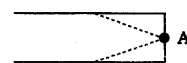


図10 K-曲率の極大点

ように定義する。

① $|C_{2,1}| < |C_{1,2}|$ の場合 :

$$C^* = \frac{n \cdot |C_{2,1}| - |C_{1,2}|}{n-1}$$

但し、 $C^* < 0$ の場合は $C^* = 0$ とする。

② その他の場合 :

$$C^* = |C_{1,2}|$$

その結果、図 11(a) のように十分大きな屈曲点では、 $C_{1,2}$ も $C_{2,1}$ も同じ値を持つことから $C^* = |C_{1,2}| = |C_{2,1}|$ となる。図 11(b) のような曲線上の点では、曲率は k に比例することから $C^* = 0$ となる。図 11(c) のように小さく屈曲する部分では、 $|C_{1,2}| < |C_{2,1}|$ となるが、A のような点では特に $C^* < 0$ となり B 点では $C^* > 0$ となる。ノイズ点では、図 11(d) のように $|C_{1,2}| > |C_{2,1}|$ となるので $C^* = |C_{1,2}|$ と小さな値になる。

以上の処理は局所的であるので、大局的には屈曲点である場合を見過ごす場合があり、それを以下の処理では補うことになる。

4.3 直線のあてはめ

まず、通常の主成分解析によりセグメントに直線をあてはめる。すなわち、セグメントを構成する点列の重心 (\bar{x}, \bar{y}) のまわりのモーメントから主軸の傾きが

$$\theta = \frac{1}{2} \tan^{-1} \frac{2M_{11}}{M_{20} - M_{02}}$$

で求めることができることから、直線の方程式は

$$y - \bar{y} = \tan \theta (x - \bar{x})$$

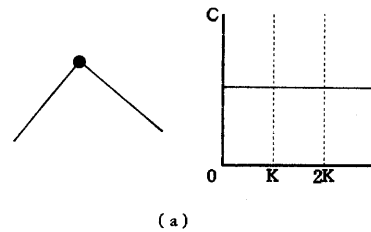
となる。エラーは、この直線から最も遠い点の距離とする。エラーが大きい場合は、次に 2 次曲線をあてはめてみる。

4.4 2 次曲線のあてはめ

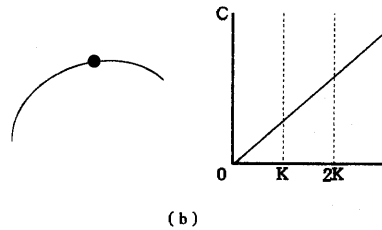
セグメントの点列に対して、2 次曲線

$$ax^2 + bxy + cy^2 + dx + ey + 1 = 0$$

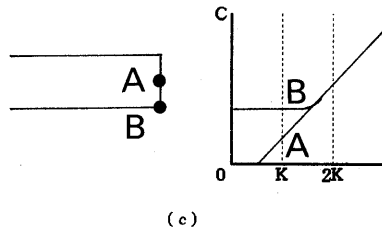
を最小 2 乗法によりあてはめる。但し、セグメントが原点を通らないように点列の重心が原点となるように点列の座標を並行移動する。2 次曲線には楕円、双曲線、放物線があるが、楕円以外の 2 次曲線があてはめられた場合には、セ



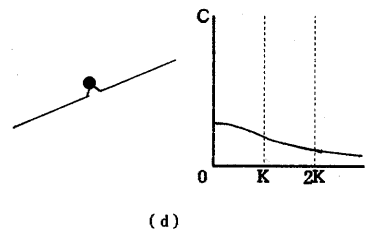
(a)



(b)



(c)



(d)

図 11 K - 曲率の変化

グメントの両端点を結ぶ直線から最も遠い点でセグメントを分割し、それぞれのセグメントに対して再び直線または 2 次曲線をあてはめる。これは、曲線はすべて楕円であるとする方が取り扱い易いことと、特に双曲線は 2 つの直線で近似する方が実際とよく一致するからである。また、放物線は実際上ほとんど現れない。

あてはめのエラーは、楕円の 2 つの焦点と楕円上の点との距離の和が一定であることを利用する。すなわち、その値と 2 つの焦点とセグメント上の点との距離の和との差の最大値とす

る。エラーが大きい場合は、セグメントの両端点を結ぶ直線から最も遠い点でセグメントを分割し、それぞれのセグメントに対して再び直線または2次曲線をあてはめる処理をエラーが小さくなるまで繰り返す。その結果、図12のような境界線表現を得る。

4.5 データ構造

以上の処理の結果、図1のように、画像は領域の集合によって記述され、各領域はその境界のセグメントのリストによって表現される。領域は、それを囲む外接長方形と明るさをその値に持つ。セグメントは、始点、終点、そして直線または曲線の方程式をその値に持つ。

5. 仮想線

人間の視覚には、図13のように、実際には存在しない線でも存在するように見える場合がある。このような錯視は、単なる現象としてではなく、物を認識する場合に必要な機能として積極的に利用することができる。

このような仮想線が生じる場合には2通りある。1つは、図14(a)のように、1つの境界線が他の面によって隠れて見えない場合であり、他の場合は、図14(b)のように、1つの面の背景にその面と同じ明るさの面があり、その面の境界線でその背景と重なっている部分の線が見えない場合である。一般的に、前者の仮想線は画像上で他の領域を通過し、後者の仮想線は画像上で同一の領域を通過することから、両者ある程度識別することができる。

連結していない2つの直線セグメントに直線をあてはめた場合に、そのエラーが小さい場合にはその間に仮想直線を引くことができる。曲線から仮想線を引く場合には、2つの曲線セグメントに曲線をあてはめた場合に、そのエラーが小さい場合である。

このような仮想線は、図15のように、多数引くことができる場合があるが、一意的に1つの組み合わせを決定することはできない。

この仮想線を引く過程で、セグメントを整形することができる。連続するセグメントや仮想線が引ける一対のセグメントに対して直線または2次曲線を新たにあてはめることができる。特に、短い線はノイズの影響を受け易いのでそ

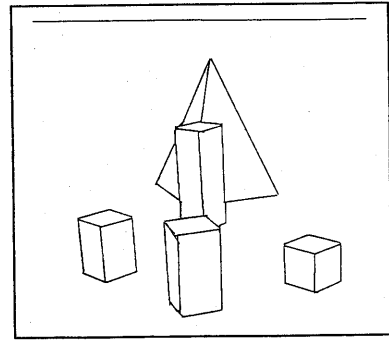


図12 直線・2次曲線のあてはめ

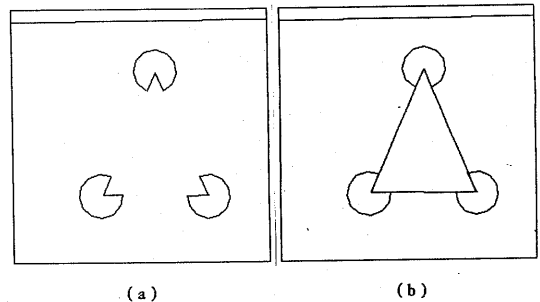


図13 仮想線

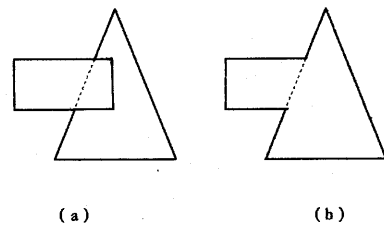


図14 仮想線の種類

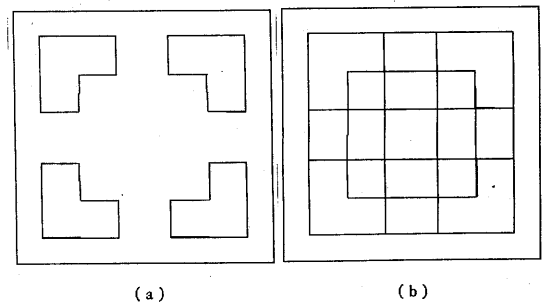


図15 仮想線

の信頼性は低く、この整形処理は有効である。例として、図2の画像から得られる仮想線を図16に示す。

6 3次元物体の認識

3次元物体の形状モデルが3次元B-REPで表現されていると、画像の2次元B-REPとモデルの3次元B-REPを照合することにより、画像中の物体の3次元の形状、位置、姿勢を推定することができる(詳細は準備中)。照合は、モデルの記述が画像の記述と一致するように、モデルの3次元の回転と並行移動を含む座標変換を求める問題である。

いま、遠くの物を見ている場合には、物体の画像への投影を図17のような疑似正射影で近似できるとする。物体のモデルの一平面上の点の集合の重心からの相対座標を (x, y, z) 、それに対応する画像中の点の集合の重心からの相対座標を (X, Y) とすると、両者の間には次のようなアフィン変換の関係がある¹⁴⁾。

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \sigma R \begin{pmatrix} x \\ y \end{pmatrix}$$

但し、 σ は拡大縮小、 R は回転を表わす。行列 R からは、 z 軸回りの回転 α 、 x 軸回りの回転 θ 、 y 軸回りの回転 ϕ を求めることができる。

対応の過程は、通常は次のような3段階になると考えられる。

①局所的な特徴が一致するような座標変換の候補を求める。

②その座標変換の候補が残りの特徴についても満足するものであるかを調べることにより、その正当性を確かめる。

③照合の誤差に基づいて座標変換のパラメータの微調整を行ない、座標変換をより正確なものとする。

①の段階では、局所的に対応点を求めることになるが、対応点の数はできるだけ少ない方がよい。なぜなら、オクルージョンやノイズによって多くの対応点を一度に求めるのが難しいからである。基本的には、上記の R を求めるには3点の対応点を求める必要がある。一方、求める対応点の数を少なくすると、探索空間が大

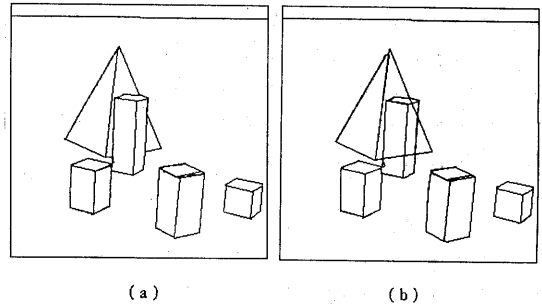


図16 仮想線

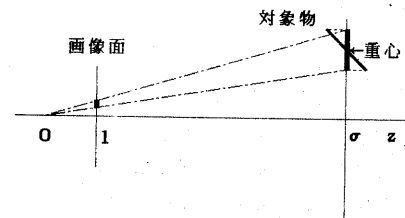


図17 疑似正射影

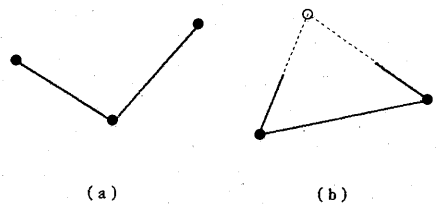


図18 照合点

きくなるので、探索空間を刈り取るなんらかのヒューリスティックな拘束が必要となる。

直線を照合する場合には、図18(a)のように、連結する2辺の3つの端点を利用するか、それがだめなら、図18(b)のように、平面上の2角挟辺を利用する方法が最も簡単であるが、それでも正しい対応が得られない場合が多く、仮想線が重要な役割を果たすことになる。そして、いずれかの方法により3点の対応点の照合を行なって R の成分を求めると、 $(\theta^*, \phi^*, \alpha^*)$ と $(-\theta^*, -\phi^*, \alpha^*)$ の2つの解が得られる。3次元の円と2次元の楕円を照合する場合は、楕円1つで十分である。すなわち、楕円の長軸と短軸から面の傾きが (θ^*, ϕ^*) か $(-\theta$

*、 ϕ *) であることが分かり、その面上で回転することによりいずれかの端点が一致するような α^* を求めればよい。

いずれの場合も2つの解が得られるが、これらはいずれも、②の段階で検証されることになる。また、③の処理は②の処理と同時に行なうこともできる。

7. あとがき

B-REP の応用は広く、例えば、従来のエッジを利用する処理（例えば、Hough 変換など）は B-REP を利用することにより、安定した結果を得ることができると考える。実際の応用例としては、ステレオ画像の対応に利用している^[6]。また、概念形成のような高レベル処理との結合も容易である。

謝辞

日頃有益な討論と助言をいただく白井制御部長、大島視覚システム研究室長、および視覚システム研究室の皆様にご感謝いたします。

参考文献

- [1] Marr, D., Vision, Freeman, 1982.
- [2] 富田, エッジを利用した領域抽出, 情報学会第23回全国大会, 4C-5, 1981.
- [3] 野村ほか, 高速視覚認識・検査システム, 第27回自動制御連合講演会, 4050, 1984.
- [4] 金出, Skewed Symmetryと画像中の規則性を形の拘束条件に変換する方法, 信学会パターン認識と学習研資, PRL79-84, 1979.
- [5] 高橋, 富田, ステレオ画像のB-REP対応による3次元シーンの記述, 情報学会コンピュータビジョン研資, 1987.