多面体認識の並列処理

田中弘美，辻三郎，Dana H. Ballard，Matthew Curtiss
大阪大学基礎工学部　University of Rochester

A parallel algorithm that uses the description of a 3-d scene to detect instances of known object models is described. Parallelism of algorithms is achieved by the use of the Hough method. The 3-d scene contains stacked objects each of which is a rigid polyhedron and may be partially occluded. The geometric structure of the scene and object models is described in the polyhedral representation that represents faces, edges, and vertices explicitly. Objects at arbitrarily different orientation and position in the scene are recognized by finding transformation parameters which specify changes in orientation and translation between parts of the scene and object models by matching edges and vertices.

1. 序

コンピュータビジョンやロボット工学の多くの研究は、画像と実世界の重要なパラメータ間の拘束について説明することに、大きな努力をはらってきた。しかしこのような拘束された環境を想定して計算を行うことについての問題点は、あまり議論されていない。また一方では物体の適切な表現法(representation)は、まずはじめに、これらの拘束下にある限定された環境下で考案され、その表現法の特徴もそこで記述されるべきであると考えられてきた。

従来のシーケンシャルな処理に適したノイマン型のコンピュータはこの様な制約された環境における問題の解決には有効であるが、一般の複雑な環境を想定する場合には、並列処理機構が使われるべきであるとする見方が一般的になって来ている。それに伴いコンピュータビジョン用の並列処理機、並列アルゴリズムも急速に注目されて来ている。

本論文は複数の多面体を含む三次元のシーンにおいて、既知の多面体(群)を一時に識別する並列アルゴリズムについて述べる。アルゴリズムは並列であるために並列処理機を用いれば、計算の複雑さは、シーンに含まれる多面体の個数に対して独立である。

シーンの複雑さの次に挙げられる問題点としてオクルージョンがある。空間の固定した一点から見たシーンの解析には物体の多くの部分が他の物体によって隠されているという大きな問題点を含んでいる。ここで述べるアルゴリズムはオクルージョンに対してあまり影響を受けず、また物体の記述が完全でなく、一部分の記述しかない場合にも有効であるという特徴を持っている。

三次元物体の形状認識問題は論理的には次の二部分に分けられる。(1) データから三次元の幾何構造の作成，(2) 得られた幾何構造から既知物体のライブラリへのマッピング，である。

本論文では第一問題は解決済み つまり多面体の形状は、得られたデータを基に、面、稜、頂点、を用いた多面表現(Polyhedral Representation)によって記述されているものと仮定する。このような幾何構造は距離画像や濃淡画像やステレオ法から諸手法を[1]用いて作成することが可能である。

第二の問題点である画像に基いた幾何構造とモデルデーターとのマッチングは、従来は、サーチ問題として扱われて来た[2,3,4,5]。しかし、二次元における画像とモデルデータとマッチングを並列処理する手法は[6]は、三次元の場合にも拡張可能であることが示唆された[7]。この並列処理法は、画像の局所的な特徴などを手掛りにして、画像データから各モデルデータへの変換を並行して算出できるという事実に基づいている。本論文では、これらを発展させ 1) 変換の回転部分検出がより効果的に処理される、2) 領域間の隣接に関する拘束も考慮されている，3) 変換の回転部分と平行移動部分の両方が算出できる，4) データーの希薄性を考慮し、パラメターテーブルと呼ばれるデータ構造を採用しているなどの特色を持つ並列処理法を述べる。

2. 概略

アルゴリズムは三次元シーンの面稜と頂点による多面表現の一部分と、既知物体のモデルの多面表現を比較し、その間の変換を第一処理の回転部分，第二処理の平行移動部分に分けて検出する。変換が正しく求められれば、物体はシーンにて確認され、変換パラメータ値がその位置を示唆する。システムの概略を図1に示す。

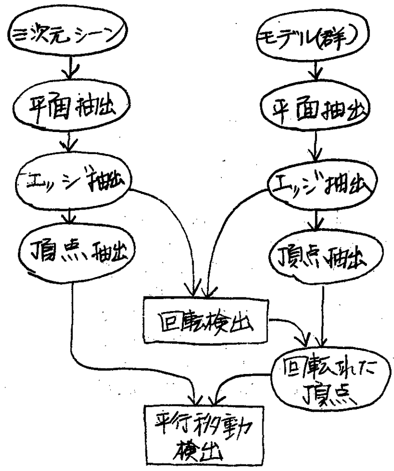


図1. システム概略

3.表現法

物体の形状の幾何学的構造を簡潔にとらえ、その拘束を表現するために、各三次元物体は、平面、稜、頂点の集合として表現されている。

3.1. 平面表現 (P representation)

物体の形状をその表面を構成する平面 P_i の集合 $\mathbf{P}$ として表現する。各平面 P_i は次の様に定義される。

$\mathbf{P}=\{P_i\}$　平面 P_i の集合 $i=1,2,\cdots,k$

$P_i=\{\vec{n_i},\rho_i\}$

ただし、$\vec{n_i}=(\alpha_i,\beta_i,\gamma_i)$ 平面 P_i の法線ベクトル、$\alpha_i,\beta_i,\gamma_i$ は x,y,z 軸と $\vec{n_i}$ との方向余弦

ρ_i = 平面 P_i から原点までの距離

3.2. エッジ(稜)表現 (E representation)

物体の形状を、隣接する二平面の境界線(エッジ) E_i の集合 $\mathbf{E}$ として表現する。各エッジ E_i は次の様に定義される。

$\mathbf{E}=\{E_i\}$　エッジ E_i の集合 $i=1,2,\cdots,l$

$E_i=(\alpha_i,\vec{e_{\alpha i}},\vec{e_{ni}},\vec{e_{Ii}},\rho_i,l_i,m_i)$

ただし、α_i = 隣接する二平面の交差角

$\vec{e_{\alpha i}}$ = E_i と x,y,z 軸との方向余弦

$\vec{e_{ni}}$ = E_i の傾き

$\vec{e_{Ii}}$ = 原点から E_i への方向

ρ_i = E_i から原点への距離

l_i = E_i の長さ

m_i = モデルインデクス、E_i が属する物体モデル名

図2に、平面 P_i と P_j が隣接し、その境界線上にエッジ E_{ij} を定義した図を示す。二平面の方程式を $P_i=(\vec{n_i},\rho_i)$, $P_j=(\vec{n_j},\rho_j)$ とすると、二平面の交差角 α は次式により求められる。

$$\alpha=\pi-\theta \quad \text{ただし} \quad \cos\theta=(\vec{n_i}\cdot\vec{n_j})/\|\vec{n_i}\cdot\vec{n_j}\| \qquad (1)$$

エッジ方向 $\vec{e_\alpha}$ は両平面の法線ベクトル $\vec{n_i},\vec{n_j}$ に垂直であることから、

$$\vec{e_\alpha}=\pm(\vec{n_i}\times\vec{n_j})/\|\vec{n_i}\times\vec{n_j}\|=(l_\alpha,m_\alpha,n_\alpha) \qquad (2)$$

ただし $l_\alpha,m_\alpha,n_\alpha$ は $\vec{e_\alpha}$ の方向余弦。

エッジの傾き $\vec{e_n}$ は二平面の法線ベクトルの平均として定義される。$\vec{e_n}$ は $\vec{e_\alpha}$ に垂直である。

$$\vec{e_n}=(\vec{n_i}+\vec{n_j})/2=(l_n,m_n,n_n) \qquad (3)$$

エッジ E_i への方向を示すベクトル $\vec{e_I}$ と ρ は、原点から E_i への最短距離を与えるベクトル $\vec{D}$ $(=\rho\cdot\vec{e_I})$ を使い次式から求められる。

$$\vec{n_1}\cdot\vec{D}=\rho_1,\ \vec{n_2}\cdot\vec{D}=\rho_2,\ \vec{e_\alpha}\cdot\vec{D}=0 \qquad (4)$$

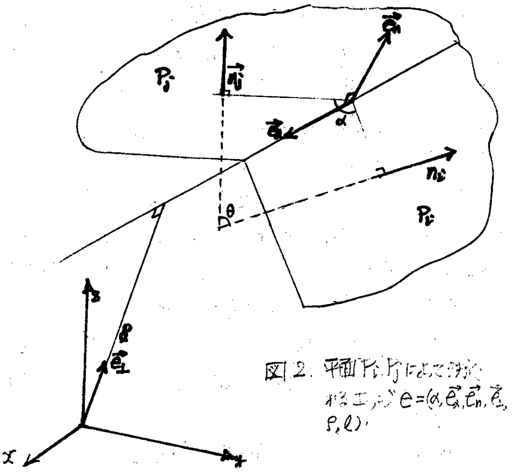


図2. 平面 P_i, P_j によって決定されるエッジ $e=(\alpha,\vec{e_\alpha},\vec{e_n},\vec{e_I},\rho,l)$

3.4 頂点表現 (V representation)

物体の形状を、エッジが交差している頂点 V_i の集合 $\mathbf{V}$ として表現する。各頂点 V_i は次の様に定義される。

$\mathbf{V}=\{V_i\}$　頂点 V_i の集合 $i=1,2,\cdots,j$

$V_i=(\vec{P_v}=(x_v,y_v,z_v),\ n_i,\ A_i=\{\beta\})$

ただし $\vec{P_v}=(x_v,y_v,z_v)$ は V_i の座標

n_i = V_i で交差しているエッジの数

$A_i=\{\beta\}$ は V_i で交差している n 本のエッジから任意に選んだ二本のエッジが交差する角度 α の集合

図3に、E_1,E_2,E_3,E_4 の4本のエッジにより定義される頂点 V を示す。

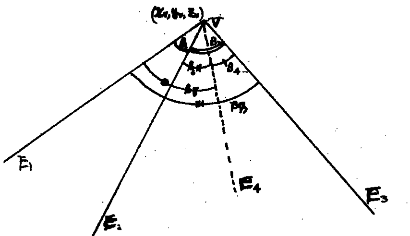


図3. エッジ E_1, E_2, E_3, E_4 によって決定される頂点 V
V=($\vec{P_V}=(x_V, y_V, z_V)$, n=4, A={$\beta_1, \beta_2, \beta_3, \beta_4$ ...})

4. 並列形状認識

三次元シーン中の各物体は、その物体のモデルからの変換を知ることにより認識される。シーンは観測者中心座標系で、又物体は物体中心座標系で記述されていることから、三次元シーン中の各物体を認識する問題は、この異なる二座標系間の変換パラメータ値を決定する問題に置き換えることができる。

以下に述べる並列認識処理は、シーンに含まれる各物体の回転パラメータを決定する第一段階、得られた回転パラメータ値を基に平行移動パラメータ値を決定する第二段階に分けて それぞれを別処理にし、順次決定する方法を採っている[7]。各段階共、キャッシュアキュムレーターを用いた一般ハフ変換が用いられ[8,9,10]、シーンに含まれるすべての物体の回転・平行移動の検出は並列処理される。

4.1. 回転(傾き)検出

シーン中の物体を認識するために、そのモデルからの変換を求める手法が、Ballardらによって提案された[7]。シーンとモデルとの間で、共通する特徴を持つ部分間で対応をとり、両者の見かけの違いから変換パラメータ値を、特に回転パラメータ値を推定した。しかし、その手法では、対応の手がかりされる特徴として面の大きさが用いられているために、オクルージョンの影響を受け易く、多数の物体を含む複雑なシーンでは正しい面間の対応を見つけ出す事は困難である。又一面の対応は、物体座標系を、対応している面の法線ベクトルに対してある角度に拘束するだけであるので、物体座標系を一意に決めるためには、少くとも傾きの異なる三面の対応が必要である。

本文では、対応をとる部分として、稜にあたるエッジを、共通する特徴としては、エッジの角度、つまり隣接する二平面が交差する角度を用い、そのエッジ間の相対姿勢の違いから回転パラメータを算出する。

対応の手がかりにされるエッジの角度αは、交差する二平面の方程式から計算されるために、それぞれの平面の一部分がシーンに現われていれば、角度は正しく計算される。そのためにオクルージョンが起こっていても、エッジ間の正しい対応関係を見つけることが可能である。又、一面の対応は回転パラメータ空間上では一円周に射影されるのに対して、一エッジの対応は二点、すなわちそのエッジの両側の二平面が描く二円周の二交点に射影されている。これはシーン中のあるエッジと物体モデル上のあるエッジとの対応関係が判れば、そのエッジを含む物体の回転の可能性は二通りに限定されることを意味する。図4に、シーン中のエッジ E_s と物体モデル中の E_m が対応した場合の、物体 $\mathfrak{M}$、つまり物体 $\mathfrak{M}$ のモデルが記述されている物体座標系の二通りの回転可能性を示す。

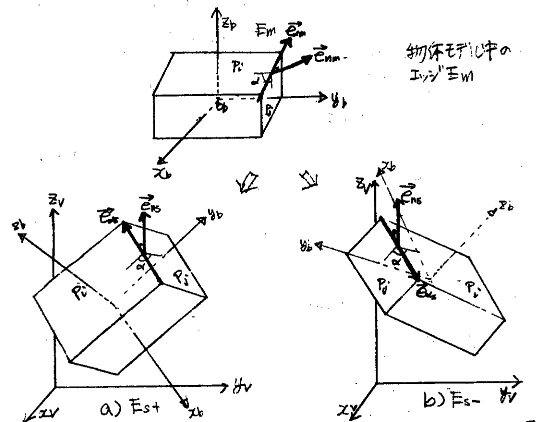


図4. 物体モデルの二通りの回転可能性 a) $E_m \to E_{s+}$ b) $E_m \to E_{s-}$

4.1.1. 回転算出原理

物体の回転 R は、回転軸 W とその回りの回転角 θ で表現され、対応しているエッジ E_s, E_m の E 表現の一部の二単位ベクトル ($\vec{e_{\alpha s}}, \vec{e_{ns}}$), ($\vec{e_{\alpha m}}, \vec{e_{nm}}$) を対として比較することにより算出される。

シーン中のエッジ E_s がモデル $\mathfrak{M}$ のエッジ E_m に対応する場合を考える。E_s, E_m の E 表現は、

$$E_s = (\alpha_s, \vec{e_{\alpha s}}, \vec{e_{ns}}, \vec{e_{1s}}, \rho_s, l_s) \quad (5)$$
$$E_m = (\alpha_m, \vec{e_{\alpha m}}, \vec{e_{nm}}, \vec{e_{1m}}, \rho_m, l_m)$$

エッジ方向ベクトルの差分ベクトルを $\vec{D_\alpha}$、エッジ法線ベクトルの差分ベクトルを $\vec{D_n}$ とする。求める回転軸 $\vec{W}$ は $\vec{D_\alpha}$, $\vec{D_n}$ の両方に対して垂直であるから、

$$\vec{D_\alpha} \cdot \vec{W} = 0 \qquad (6)$$
$$\vec{D_n} \cdot \vec{W} = 0$$

ただし $\vec{D_\alpha} = \vec{e_{\alpha s}} - \vec{e_{\alpha m}}$
$\vec{D_n} = \vec{e_{ns}} - \vec{e_{nm}}$

(6)より回転軸 W は二差分ベクトル $\vec{D_\alpha}$, $\vec{D_n}$ の

外積から次の様に計算される。

$$\vec{W} = (\vec{D_\alpha} \times \vec{D_n}) / \| \vec{D_\alpha} \times \vec{D_n} \| \qquad (7)$$

右図(図5)に示す様に、得られたWを極ベクトル、gを赤道とするガウス球を考える。二本の方向ベクトル$\vec{e}_{\alpha s}$, $\vec{e}_{\alpha m}$と、法線ベクトル$\vec{e}_{ns}$, $\vec{e}_{nm}$をガウス球に射影すると、$\vec{W}\times\vec{e}_{ns}$と$\vec{W}\times\vec{e}_{nm}$は赤道上の二ベクトルに対応し、両ベクトルの成す角度は求める回転角θに等しくなっている。回転角θは(　)を解くことによって得られる。

$$\cos\theta = (\vec{W}\times\vec{e}_{ns})\cdot(\vec{W}\times\vec{e}_{nm}) \qquad (8)$$
$$\vec{W}\cdot\sin\theta = (\vec{W}\times\vec{e}_{ns})\times(W\times\vec{e}_{nM})$$

4.1.2. 並列回転検出アルゴリズム

三次元シーン、又シーンに含まれている物体のモデル群の形状は、3.2で述べられた様に、E表現によってエッジの集合 $\mathbf{E}=\{E_i\}$として記述されている。各エッジは同じ値の交差角を持つエッジ毎にまとめられて、交差角αをインデクスとする。シーンエッジテーブル、モデルエッジテーブルに格納されている。図6にエッジテーブルの様式を示す。アルゴリズムは、この二テーブルを用いて対応関係にあるエッジ対を捜し出し、その相違から物体の回転パラメータ値を算出する。

隣接する平面P_i,P_jの成す交差角 α_i	α_iを交差角として持つエッジE_iの集合 $\mathbf{E}(\alpha_i)=\{E_i=(\alpha,\vec{e}_\alpha,\vec{e}_n,\vec{e}_\perp,\rho,l)\}$
$\Delta\alpha$	$\mathbf{E}(\Delta\alpha)$
$2\Delta\alpha$	$\mathbf{E}(2\Delta\alpha)$
⋮	⋮
$i\Delta\alpha$	$\mathbf{E}(i\Delta\alpha)$
⋮	⋮
$k\Delta\alpha(\fallingdotseq 2\pi)$	$\mathbf{E}(k\Delta\alpha)$

図6. エッジテーブル

複数物体の並列処理を可能にするために、各物体モデルがそれぞれに格納されているエッジテーブルを集めて一つのエッジテーブルに統合する。統合エッジテーブルのエントリは、各物体モデルのエッジテーブルのエントリの和、つまり統合エッジテーブルのインデクスαによるエントリ$\mathbf{IE}_m(\alpha)$は、各モデルm_iが格納されているエッジテーブのインデルスαによるエントリ$\mathbf{E}_i(\alpha)$の和である

$$\mathbf{IE}_m(\alpha) = \mathbf{E}_1(\alpha) + \mathbf{E}_2(\alpha) + \cdots + \mathbf{E}_k(\alpha)$$
ただし kは物体モデル数　　(9)

統合エッジテーブルにおける各エッジのE表現は、そのエッジが属する物体モデル名を示すために、モデルインデルスm^*が追加されている。

$$E' = (\alpha, \vec{e}_\alpha, \vec{e}_n, \vec{e}_\perp, \rho, l, m^*)$$

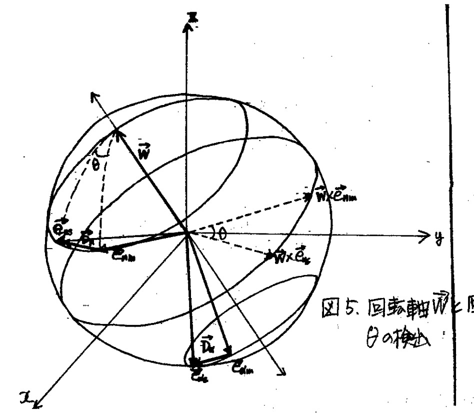


図5. 回転軸$\vec{W}$と回転角θの検出

E_s, E_mをそれぞれシーン、物体モデルmから抽出された任意のエッジとする。もし"両者の持つ交差角αが等しければ、E_sとE_mは対応している"とみなす。これらは交差角αをインデクスとして、シーンエッジテーブルから得られるエントリを$\mathbf{E}_s(\alpha)$とする。同様に統合エッジテーブルから得られるエントリを$\mathbf{IE}(\alpha)$とする。$\mathbf{E}_s(\alpha)$と$\mathbf{IE}_m(\alpha)$に含まれるエッジの交差角はすべてαであるから、それぞれ任意に選んだ二本のエッジの組み合わせは、すべてが対応関係にあるとみなすことができる。$\mathbf{E}_s(\alpha)$と$\mathbf{IE}_m(\alpha)$の大きさをそれぞれk, lとすると$k\times l$個の対応するエッジの組み合わせが得られる。

図7に例を示す。今n個の物体モデルのうち、交差角がαであるエッジはモデルm_iから$\mathbf{E}_i(\alpha)=\{E_{i1},E_{i2}\}$、モデル$m_j$から$\mathbf{E}_j(\alpha)=\{E_{j1},E_{j2},E_{j3}\}$抽出され、その和$\mathbf{E}_i(\alpha)+\mathbf{E}_j(\alpha)$が統合エッジテーブルでインデクス

交差角 α_i	$\mathbf{IE}_m(\alpha_i)=\{(\alpha,\vec{e}_\alpha,\vec{e}_n,\vec{e}_\perp,\rho,l,m^*)\}$
⋮	…
α	$\mathbf{IE}_m(\alpha)=\{E_{i1}^*,E_{i2}^*,E_{j1}^*,E_{j2}^*,E_{j3}^*\}$
⋮	…

mはモデルインデクス

統合エッジテーブル

S = {シーンとモデル群間で交差角αを持つ対応エッジ対の集合}

$= \{(E_{i1}^*,E_{s1}),(E_{i1}^*,E_{s2}),(E_{i1}^*,E_{s3})$
$(E_{i2}^*,E_{s1}),(E_{i2}^*,E_{s2}),(E_{i2}^*,E_{s3})$　m_iの回転R_{i1}, R_{i2}の算出
$(E_{j1}^*,E_{s1}),(E_{j1}^*,E_{s2}),(E_{j1}^*,E_{s3})$
$(E_{j2}^*,E_{s1}),(E_{j2}^*,E_{s2}),(E_{j2}^*,E_{s3})$
$(E_{j3}^*,E_{s1}),(E_{j3}^*,E_{s2}),(E_{j3}^*,E_{s3})\}$　m_jの回転R_{j1}, R_{j2}, R_{j3}の算出

交差角 α_i	$\mathbf{E}_s(\alpha_i)=\{(\alpha,\vec{e}_\alpha,\vec{e}_n,\vec{e}_\perp,\rho,m)\}$
⋮	
α	$\mathbf{E}_s(\alpha)=\{E_{s1},E_{s2},E_3\}$
⋮	…

シーンエッジテーブル

図7. 統合エッジ、シーンエッジテーブルから得る対応エッジ対

αのエントリ$\mathbf{IE}_m(\alpha)$として格納されている。一方,シーンからは$\mathbf{E}_s(\alpha)=\{E_{s1},E_{s2},E_{s3}\}$のエッジが交差角$\alpha$を持ち,シーンエッジテーブルのインデクス$\alpha$のエントリとして格納されている。$\mathbf{E}_s(\alpha)\times\mathbf{IE}_m(\alpha)=3\times5$の対応エッジ対が得られる。

(E_{s1},E_{i1}),(E_{s1},E_{i2})はそれぞれシーンのエッジE_{s1}が物体モデル$\mathbf{m_i}$のエッジE_{i1},E_{i2}に対応すると解釈する場合を示し、各解釈に応じた物体m_iの回転パラメータ値を算出する。同様に(E_{s2},E_{i1}),(E_{s2},E_{i2})はシーンのエッジE_{s2}がm_iのエッジE_{i1}又はE_{i2}に対応すると解釈した場合を示し各解釈に応じた物体モデル$\mathbf{m_i}$の回転パラメータ値を算出する。

アルゴリズムは両エッジテーブルから得られた各対応エッジ対に対して,モデルインデクスm^*によって示される物体の,二通りの回転可能性を計算し,その値をアキュムレーターに"投票"(voting)する。その値が既にアキュムレータに在れば,該当の"投票数"(votes)を指定された量だけ増加させる。もし無ければ空領域の先頭に追加する。この様に,キャシュアキュムレータには常に"投票数"の多い順に上位k個のパラメータ値のみ保持されている。物体$\mathbf{m_i}$の回転パラメータ値は,キャシュアキュムレータ上で,モデルインデクスの値が$m^*=i$であるパラメーター値の中から最大の得票をしたものに決定される。

回転検出アルゴリズム

input: シーンエッジテーブル(SETable)
統合エッジテーブル(IETable)

(step.0) SETable, IETableを作成する。

step.1 キャシュアキュムレーターを初期化する。モデルインデクスm^*,回転軸$\vec{W}=(l_w,m_w,n_w)$ ただしl_w,m_w,n_wはx,y,z軸方向の方向余弦,$l_w^2+m_w^2+n_w^2=1$,回転角θ,投票数N

	m^*	$\vec{W}$		θ	N
		l_w	m_w		
1					
2					
⋮					

キャシュアキュムレーター

step2. 各交差角α_iに対して$(i=1,2,\cdots)$
SETable, IETableから得られるエントリを$\mathbf{E}_s(\alpha_i)$,$\mathbf{IE}_m(\alpha_i)$とする。

step2.1 $\mathbf{E}_s(\alpha_i)$,$\mathbf{IE}_m(\alpha_i)$から得られる各対応エッジ対(E_{sj},E_{mj})に対して(ただし$E_{sj}\in\mathbf{E}_s(\alpha_i)$, $E_{mj}\in\mathbf{IE}_m(\alpha_i)$)

step2.1.1 (7)より回転軸$\vec{W}$を求める

$$\vec{W}=(\vec{D}_\alpha-\vec{D}_n)/\|\vec{D}_\alpha-\vec{D}_n\| =(l_w,m_w,n_w)$$

step2.1.2 (8)より回転角θを求める

step2.1.3 キャシュアキュムレーター上に,m^*,θを"投票"する。

step3. 各物体m_iに対して$(m=1,2,\cdots k)$に対して,
アキュムレータから,モデルインデクス$m^*=i$で"投票数"Nが最大のパラメータ値を選び,物体$\mathbf{m_i}$の回転パラメータ値を決定する。

4.2. 位置検出

第一段階で決定された回転パラメータ値を用いて,各物体モデルを正しく回転させれば,シーン中の各物体とそのモデルとの相違は位置だけとなる。

各物体の位置は,そのモデルが記述されている物体座標系と,三次元シーンが記述されている世界座標系の間の平行移動として定義されている。

位置の検出には物体の頂点が用いられる。三次元シーン中の頂点と各物体モデルの頂点の対応は,等しい交差角βの有無が手掛りにされる。つまりシーン中のある頂点V_sと物体モデルm_iの頂点V_mは,それぞれのV表現に示される交差角集合$A_s=\{\beta_s\}$,$A_m=\{\beta_m\}$(ただし$V_s=(P_{vs}=(x_{vs},y_{vs},z_{vs}),n_s,A_s)$,$V_m=(P_{vm}=(x_{vm},y_{vm},z_{vm}),n_m,A_m)$)に"共通する交差角$\beta$が含まれていれば,$V_s$と$V_m$は対応している"とみなす。求める二座標間の平行移動値はV_sとV_mの三次元座標の差$\vec{P}_{vs}-\vec{P}_{vm}$から計算される。

$$T=(\Delta x,\Delta y,\Delta z) =(\vec{P}_{vs}-\vec{P}_{vm}) =(x_{vs}-x_{vm},\ y_{vs}-y_{vm},\ z_{vs}-z_{vm}) \quad (10)$$

図8は,共通の交差角βをその交差角集合に含む二頂点V_sとV_mが対応し,その三次元座標の差から求める二座標間の平行移動が

算出されることを示している。

n本のエッジで構成される頂点V_iは，${}_nC_2$個の交差角の集合A_i（ただし等しい値のβも重複して数えた場合）として記述されるために、各交差角$\beta \in A_i$毎に，V_iの正しい対応関係が確認されることになり、各頂点の形状の複雑さ（エッジの本数n，交差角βの大きさ等）が正しく対応に反映されている。

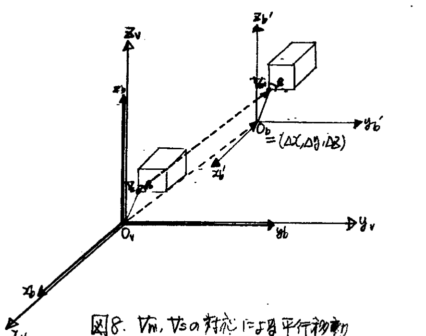


図8. $\vec{V}_m$, $\vec{V}_s$の対応による平行移動 $T=(\Delta x, \Delta y, \Delta z)$の検出

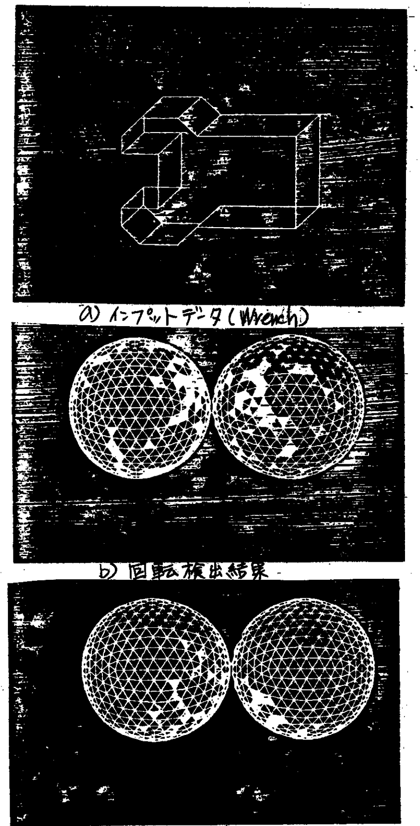
平行移動検出アルゴリズムは回転検出アルゴリズムと同様に、二つのテーブル（シーン頂点テーブル，統合頂点テーブル）を参照し共通のエッジ交差角βを持つ対応頂点対$\{(\vec{V}_s, \vec{V}_m)\}$を得てその位置の差$\vec{P}_s - \vec{P}_m$から平行移動パラメータを算出し、アキュムレータ上に“投票する”。モデルM_iの平行移動パラメータ値はモデルインデクスがm＝iであるものの中から最大の得票を得たものに決定される。

5. 実験結果

現在はシーン，モデルともただ1つの物体を含む場合のみの実験まで終了し、その結果は表1に示す通り、三例ともほぼ正しい結果を得ている。今後は、多数の物体を含むシーンを扱う実験へ発展させ、アルゴリズムの並列性を確認するのが課題である。

	Rotation Computed w	θ	Rotation Desired w	θ	Translation Computed T	Translation Desired T
wrench	0.7000 0.0000 0.7000	1.5080	0.7071 0.0000 0.7071	1.57	16.0 24.0 -4.0	15.0000 -20.0000 -5.0000
hill	0.7000 0.0000 0.7000	1.1310	0.7071 0.0000 0.7071	1.162389	20.0 -12.0 12.0	20.0000 -15.0000 10.0000
polarbear	0.7000 0.0000 0.7000	1.0053	0.7071 0.0000 0.7071	1.036726	-32.0 -20.0 -20.0	-30.0000 -20.0000 -20.0000

表1. 実験結果1.2.3の内容



a) インプットデータ（Wrench）

b) 回転検出結果

c) 平行移動検出結果

図9. Wrenchデータを使った実験結果

参考文献

[1] Ballard D.H. and Tanaka T.H. "Frame-Based Form Perception: Constraints, Algorithms, Implementation" TR , Computer Science Dept., University of Rochester, Submitted to 9th IJCAI, LosAngels, CA

[2] R.O Duda,D. Nitzan and P.Barvet, "Use of Range and Reflectance Data to Find Planar Surface Regions", IEEE Trans. On Pattern Analysis and Machine Intelligence Vol. PAMI-1, No. 3, pp 259-271, July 1979

[3] R.Bajcsy, "Three Dimensional Scene Analysis," Proc. 4th IJCPR Kyoto, Japan, Nov. 1978, PP86-96

[4] Oshima M. and Shirai Y. " A Scene Description Method Using Three-Dimensional Information", Pattern Recognition, Vol. 11, pp.9-17,1979

[5] Oshima M. and Shirai Y., "Object Recognition Using Three-Dimensional Information", In Proc. IJCAI-81, 1981 pp601-616

[6] Hrechanyk, L.M. and Ballard, D.H., "A connectionist model for shape perception," Computer Vision Workshop, Ringe, NH, August 1982; also appeared as "Viewframes: A connectionist model of form perception," *Proc.*, DARPA IU Workshop, Arlington, VA, June 1983.

[7] Ballard H.D. and Sabbah D. " Viewr Independent Shape Recognition" IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol PAMI-5, No6. Nov, 1983

[8] Brown C. "Hierarchical Cache Accumulators For Sequential Mode Estimation," TR 125, Computer Department, University of Rochester

[9] Duda R.O. and Hart P.E., "Use of the Hough Transform to detect lines and curves in pictures," CACM 15, 1. 11-15 Jan. 1972

[10] Ballard D.H., "Generalizing the Hough Transform to Detect Arbitrary Shapes,"