多面体の再構成

丹羽敏行・渡辺凡夫
電気情報工学科

（1986年９月６日受理）

Reconstruction of Polyhedric Object

Toshiyuki NIWA, Tsuneo WATANABE
Departmemt of Electrical and Computer Engineering

(Received September 6, 1986)

The data input of 3-D structure has been increasingly important. Unfortunately, the special and expensive devices are usually required to get 3-D coordinates of points.

Then we present a stereo vision method of using only one camera and a turntable on which 3-D object must be fixed, such that the device cost and the arithmetic operations are reduced.

In this method, we use Hough transform to get a more reliable coordinates of vertexes, and the matching algorythm between the vertexes on two images to achieve a quick processing.

まえがき

近年，コンピュータに人間の持つような視覚機能を与えようとする研究が，人工知能の１分野として研究されている。これらはコンピュータ・ビジョンと呼ばれ，２次元画像からもとの３次元世界を再構成することが１つの研究目標である。

この再構成には，大きく分けて２つのアプローチがある。１つは，ある視点から撮られた１枚の写真から，もとの３次元世界を再構成しようとするもので，単眼視処理と呼ばれる。他のアプローチは，我々に目が２つあることから分かるように，２つあるいはそれ以上の視点から撮られた複数枚の画像から３次元世界を認識しようとするものである。これは複眼視処理と呼ばれ，２台のカメラを用いて，その画眼視差から３次元位置を計測するステレオ法は良く研究されてきた。ただ，２つの画面の間での対応づけをいかに行うかが課題となっている。

そこで，我々も，対象を多面体に限定し，その形状パラメータの計測とそれに基づく再構成を試みてきた[1),2)]。

今回，より信頼性の高い頂点の検出と，頂点間での対応づけによる処理の高速化を計ったので，その概要を報告する。

1．概　　要

ステレオ法では，通常２台のカメラを用いて，その両眼視差から３次元位置を計算する。しかし，カメラが２台になると，費用が増し取扱いも煩雑になるばかりでなく，なによりも計算量と誤差の増大を招くことになる。

そこで，Fig. 1 に示すように，カメラは１台で固定し，カメラの前方に設定されたターンテーブル上に対象を置いて，ターンテーブルを回転させて複数の画像を入力することにした。

この結果，費用はもとよりカメラ・パラメータが簡単になり計算量も減少できた。

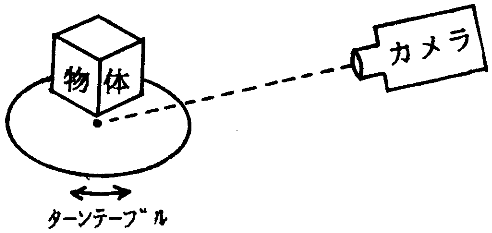

Fig. 1 Device configuration

1.1　処理概要

処理とデータの流れの概要を Fig. 2 に示す。

ここで使用した画像メモリは，ドット数が512×512で，１ドットあたり256階調のもので，パソコンとはＧＰＩＢで接続されている。

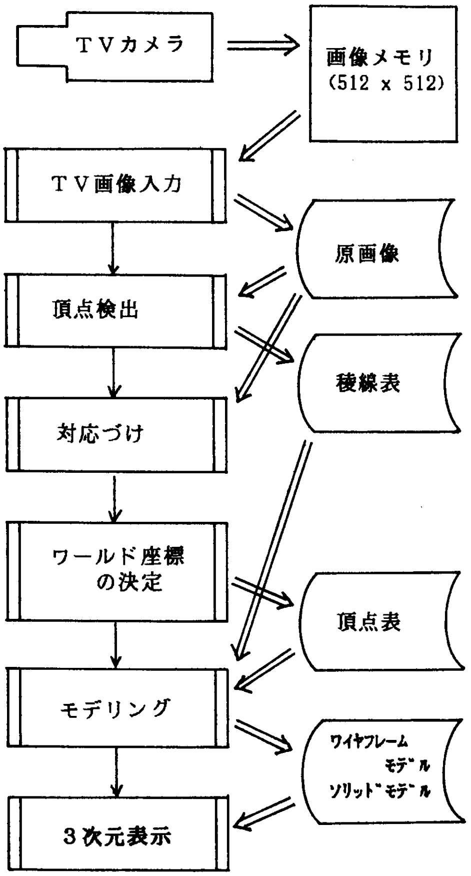

Fig. 2 Data and process flow

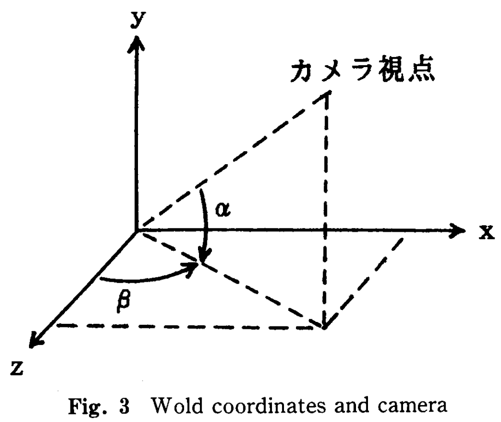

Fig. 3 Wold coordinates and camera

1.2 カメラ・パラメータ

ここで，カメラ・パラメータについて考察してみる。先ずカメラは次のように設定する。

(1)ターンテーブルの回転中心をワールド座標の原点とする。

(2)カメラの光軸をその原点に一致させる。

カメラの光軸がz軸上にくるように，y軸回りにβ，x軸回りにαだけ回転し，さらに軸がフィルム面上で水平になるようにz軸回りにγだけ回転したときの変換マトリクスT_1は次のようになる。

さらに，変換後の座標系のx y平面に対する透視変換をT_2，変換後の画像のスケーリングをおこなうCRT平面への変換をT_3とすれば，それぞれ次のように表される。

$$T_1=\begin{pmatrix}\cos\beta & 0 & -\sin\beta & 0\\ 0 & 1 & 0 & 0\\ \sin\beta & 0 & \cos\beta & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & \cos\alpha & \sin\alpha & 0\\ 0 & -\sin\alpha & \cos\alpha & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\cos\gamma & \sin\gamma & 0 & 0\\ -\sin\gamma & \cos\gamma & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$T_2=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & -1/d\\ 0 & 0 & 0 & 1\end{pmatrix}\qquad T_3=\begin{pmatrix}k & 0 & 0 & 0\\ 0 & k & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

ここで，dはカメラ視点と原点の距離で，カメラ視点とスクリーン平面の距離を1として，$k=l/d$である。

これより，CRT座標への変換マトリスクは次のようになる。

$$T=T_1\ T_2\ T_3=\begin{pmatrix}T_{11} & T_{12} & 0 & T_{14}\\ T_{21} & T_{22} & 0 & T_{24}\\ T_{31} & T_{32} & 0 & T_{34}\\ 0 & 0 & 0 & 1\end{pmatrix}$$

ここで，

$$T_{11}=k\,(\cos\beta\ \cos\gamma-\sin\alpha\ \sin\beta\ \sin\gamma)$$
$$T_{21}=-k\ \cos\alpha\ \sin\gamma$$
$$T_{31}=k\ (\sin\beta\ \cos\gamma+\sin\alpha\ \cos\beta\ \sin\gamma)$$
$$T_{12}=k\ (\cos\beta\ \sin\gamma+\sin\alpha\ \sin\beta\ \cos\gamma)$$
$$T_{22}=k\ \cos\alpha\ \cos\gamma$$
$$T_{32}=k\ (\sin\beta\ \sin\gamma-\sin\alpha\ \cos\beta\ \cos\gamma)$$
$$T_{14}=\cos\alpha\ \sin\beta/\mathrm{d}$$
$$T_{24}=-\sin\alpha/\mathrm{d}$$
$$T_{34}=-\cos\alpha\ \cos\beta/\mathrm{d}$$

これより，ワールド座標（x, y, z）とCRT座標（x^*, y^*）との関係は，

$$[x\ y\ z\ 1]\mathrm{T}=h\,[x^*\ y^*\ 0\ 1]$$
$$T_{11}x+T_{21}y+T_{31}z=hx^*$$
$$T_{12}x+T_{22}y+T_{32}z=hy^*$$
$$T_{14}x+T_{24}y+T_{34}z+1=h$$

と表され，hを消去して次の式が得られる。

$$(T_{11}-T_{14}x^*)x+(T_{21}-T_{24}x^*)y+(T_{31}-T_{34}x^*)z=x^*$$
$$(T_{12}-T_{14}y^*)x+(T_{22}-T_{24}y^*)y+(T_{32}-T_{34}y^*)z=y^*$$

上式より，5点の座標（x, y, z）と（x^*, y^*）が与えられれば，9個のカメラ・パラメータが求まることにな

る。

なお，一方のカメラ・パラメータが求まれば，他方はy軸回りの回転角度が異なるだけであるから簡単に求めることが出来る。

結局，カメラ・パラメータが求まれば，2つの透視投影T^1，T^2による画像上の点（x^{*1}, y^{*1}）と（x^{*2}, y^{*2}）から，対象の点の座標(x, y, z)の近似値は次式で表される。

$$X=(A^{\mathrm{T}}A)^{-1}(A^{\mathrm{T}}B)$$

ここで，

$$X=\begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad A=\begin{pmatrix} T^1_{11}-T^1_{14}x^{*1} & T^1_{21}-T^1_{24}x^{*1} & T^1_{31}-T^1_{34}x^{*1} \\ T^1_{12}-T^1_{14}y^{*1} & T^1_{22}-T^1_{24}y^{*1} & T^1_{32}-T^1_{34}y^{*1} \\ T^2_{11}-T^2_{14}x^{*2} & T^2_{21}-T^2_{24}x^{*2} & T^2_{31}-T^2_{34}x^{*2} \\ T^2_{12}-T^2_{14}y^{*2} & T^2_{22}-T^2_{24}y^{*2} & T^2_{32}-T^2_{34}y^{*2} \end{pmatrix}$$

$$\mathrm{B}=\begin{pmatrix} x^{*1} \\ y^{*1} \\ x^{*2} \\ y^{*2} \end{pmatrix}$$

2．前　処　理

入力された画像からエッジ検出を容易にするために，雑音除去から始まり，画像における微分処理，微分画像における非極大点の抑制，しきい値処理を行う。

2.1　画像の平滑化

平滑化処理による雑音の除去法は，近傍画素の平均濃度を採用すると，2つの領域の境界付近に中間的な濃度の部分が生じて，2つの領域の境界(辺縁)がぼけてしまう。そこで，辺縁をぼかすことのないように辺縁を含まない平滑化領域をFig. 4のように9箇所定め，これらの中で濃度の分散が最小になる領域を平滑化領域とした。

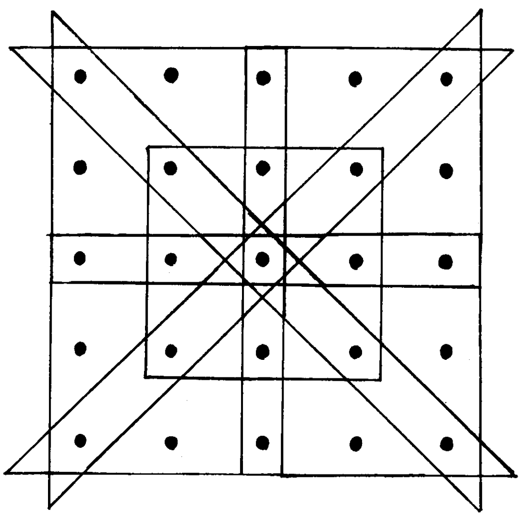

Fig. 4　Noise reduction method

2.2　エッジの検出

画像の分割のための重要の方法は，濃度やテクスチャーが急激に変化していて，1つの領域が終り他の領域が始まっている様な不連続性を検出することに基ずく方法である。この不連続性をエッジと呼ぶ。

濃度の急激な変化には，いくつか違った形があるが，重要な種類の濃度の不連続は線(LINE)であり，これはその両側の領域とは性質が異なった細い帯である。線はエッジに伴って起こることが多い。こうしたエッジを検出するには画像を微分すればよく，本研究で扱うディジタル画像では微分の代りに差分を用いる。

よく知られているディジタル勾配の近似にRobertsによるものがあり，今回はこの"Robertsの勾配"による近似を用いた微分処理を行った。

画像の勾配は任意の互いに直交する2方向とも隣接する2画素間の濃度差分から求められ，次に示すような斜め方向の4画素を利用して画像の勾配の大きさ$M(i, j)$を次式から求める。

$$M(i, j)=\{(f(i, j)-f(i+1, j+1))^2+(f(i, j+1)-f(i+1, j))^2\}^{1/2} \quad (1)$$

勾配の大きさ$M(i,j)$は画像$f(i, j)$において辺縁付近で大きな値となり，濃度がほぼ一定の位置では小さな値になる。したがって，濃淡画像に対して差分処理を行って得られる画像$M(i, j)$は辺縁が強調された画像となる。

原画像を微分することにより，図形のエッジが検出されるわけであるが，微分処理しただけの画像では，しきい値処理を行うと1本のエッジが2箇所以上で検出されることがあり，結果としてエッジが太くなり，直線の抽出が困難となる。そこで，エッジの水平，垂直方向についてスキャンラインを作り，現行の点の濃度値$D(P)$とスキャンラインの上で1つ後の点の濃度値$D(P-1)$を比較して，$D(P)$が$D(P-1)$よりも大きければ，$P-1$の点情報を削除する。

こうして水平，垂直方向についての極大点のみを残してゆき，微分画像のつながりを明確に細線化する。

次に，非極大点の抑制を施した画像に対して今度はしきい値処理を行う。つまり，雑音と思われるような点を削除してしまい，次のHough変換での無駄をなくし，精度の良い直線を得るためである。会話形式により何回でもしきい値の設定を可能にし，エッジよりも高い濃度値を持つ雑音除去も可能にした。

微分画像と非極大点の抑制画像をFig. 5，Fig. 6に示すので参照されたい。

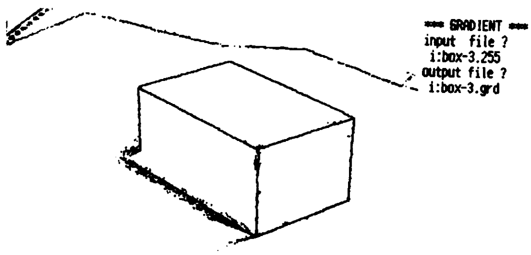

Fig. 5 Image of gradient

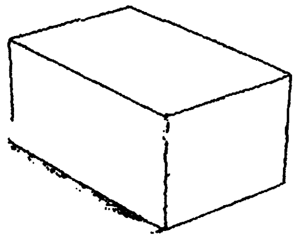

Fig. 6 Non-maximum suppressed image

3．画像の直線抽出

ここでは直線抽出までの処理を述べる。

3．1 Hough変換

画像処理でよく使われる直線のあてはめ（直線近似）に Hough 変換による方法がある。これは Fig. 7 に示すように，幾つかの点が直線(2)式の上に存在していると仮定すれば，直線(3)式となる。

$$y = a \cdot x + b \tag{2}$$

$$y_i = a \cdot x_i + b \quad i = 1, 2, \cdots, M \tag{3}$$

これらの式で (a, b) を変数とみなして (a, b) 平面を考えると，そこでの直線の式は式(4)となる。

$$b = -x_i \cdot a + y_i \tag{4}$$

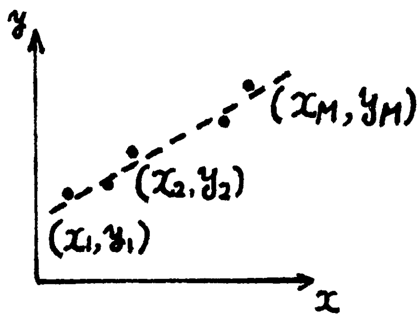

Fig. 7 Points on the line

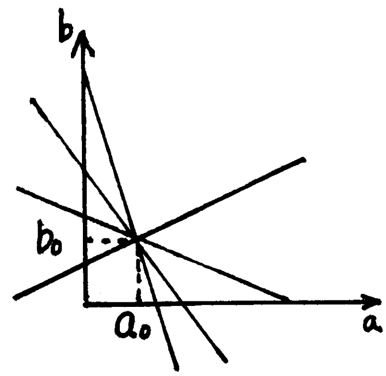

Fig. 8 Lines on the (a, b) plane

この直線は (x_i, y_i) によって全て異なるが，(x_i, y_i) は全て同一直線上に乗っていると仮定したのであるから，この直線を $y = a_0 \cdot x + b_0$ であるとすれば，直線(4)式は (x_i, y_i) のいかんにかかわらず，(a, b) 平面上の点 (a_0, b_0) を通らねばならない。これを Fig. 8 に示す。そこで (a, b) 平面を十分細かに区切って，(x_i, y_i) に対応する直線があるメッシュを通過するものであれば，そのメッシュの計数に (x_i, y_i) の濃度値を累積してゆく。これを画像上の全ての (x_i, y_i) について行えば，当然のことながら (a_0, b_0) に対応するメッシュの計数値が最大となるはずである。そこで (a, b) 平面上でメッシュの計数値が最大のものを求めればそれによって (a_0, b_0) が決定したことになる。そして直線の方程式は $y = a_0 \cdot x + b_0$ となる。

この方法を Hough 変換による直線の決定法という。

Fig. 9 のように幾つかの直線・線分が存在するような図形に対して Hough変換を行うと，(a_0, b_0) に当たる極大値を取るメッシュが幾つか発見されることになる。このような場合には，十分に大きな局所的極大を取るメッシュを発見する必要がある。

(a, b) 平面上への変換では a, b の十分に大きな所ま

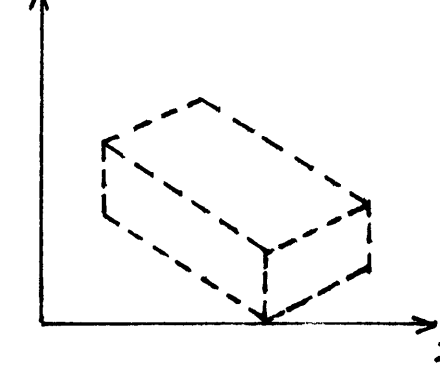

Fig. 9 Picture composed of plural lines

でメッシュを区切って取らなければならないという面倒さがある。そこで，(2)式の代りに

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \tag{5}$$

という形で（ρ，θ）平面に変換する方が良い。

こうすれば，少なくともθは0からπまでの有限の値を細かく分割すればよく，またρもある程度の大きさまで取れば実際上は十分である。

さて，メッシュのテーブルを作らねばならないので，θは$0-\pi$までとして，あらかじめρの範囲（ρ_{max}，ρ_{min}）を調べなくてはいけない。

今回は最初の入力画像が512＊480だったので，

$0 \leqq x_i \leqq 511,\quad 0 \leqq y_i \leqq 479$

となる。

(1)ρ_{max}

$0 < \cos\theta$, $0 < \sin\theta$ で$\cos\theta$, $\sin\theta$の計数が最大の時のθの変化を見ればよい。つまり，$0 \leqq \theta \leqq \pi/2$ の範囲を$x=511$，$y=479$としてρ_{max}を調べればよい。

$$\rho = 511 * \cos\theta + 479 * \sin\theta$$

$$\rho' = -511 * \sin\theta + 479 * \cos\theta$$

$\rho' = 0$ とすると，$\tan\theta = 479/511$ となり，

$$\theta = \tan^{-1}(479/511)$$

$\theta = 43.1°$ で $\rho_{max} = 700.4014$ という値になる。

(2)ρ_{min}

$0 > \cos\theta$, $0 < \sin\theta$で$\cos\theta$, $\sin\theta$の係数が最大の時のθの変化を見ればよい。つまり，$\pi/2 \leqq \theta \leqq \pi$ の範囲を$x=511$，$y=479$としてのρ_{min}を調べればよいが，明らかに，$\theta=\pi$で，$\rho_{min}=-511$である。

これで$-511 \leqq \rho \leqq 700$とわかったが，整数値で$\rho$を区切ると，1211となり，メッシュが細かすぎて処理時間もメモリも莫大になる。ρメッシュの大きさとして，600が適当なので，(5)式から導びかれるρの値の大きさを600に圧縮することにした。尚，θは$0 \leqq \theta \leqq \pi$の範囲を0.8995刻みで区切った。

データ構造はρ（600）＊θ（200）で，1メッシュ4バイトとして約480キロバイトを要する2次元配列にした。配列の第1引数からθを，第2引数からρを導びけるようにしてある。

理論的には200＊600本の直線が存在するのだが，濃度値が累積されていったメッシュの内，大きい計数をもつものが画像の直線候補と推定されるメッシュである。そこで，適当な値でしきい値処理を行い，おおきいメッシュを配列に格納し，後でρ，θからそのメッシュが表す直線の傾きと切片が参照できるようにしておく。

次に，各メッシュが表す直線の式に画像上の全点を当てはめて，誤差評価を(6)式する。

評価式が真となる点を直線の構成要素とし，最終的に最小二乗法をこれらの構成要素について適用し，精度の高い傾きと切片を持った直線を検出する。

更に，その直線候補群から同一直線と考えられるもの同士を構成要素の数の大きさで1つに絞ると，画像のエッジの数と抽出された直線の数とが一致する。

$$|\ y_i - (x_i * a + b)\ | < \mathrm{SEIDO} \tag{6}$$

最小二乗法により得られる傾きと切片の式を示す。

$$a = \frac{(n\Sigma x_1 y_1 - \Sigma x_1 \Sigma y_1)}{(n\Sigma x_1^2 - (\Sigma x_1)^2)} \tag{7}$$

$$b = (\Sigma y_1 - b\Sigma x_1)/n \tag{8}$$

このa, bを$y=ax+b$に代入して回帰直線の式となる。

ここで，傾き絶対値が1よりも大きいと実際には直線の近傍の点でも評価式の誤差が大きくなり，その直線の構成要素から外されてしまう点がでてくる。そこで，傾き絶対値が1よりも大きい$|a|>1$直線に対しては

$$x = y/a - b/a \tag{9}$$

評価式 $|x_1 - (y_1 - b)/a\ | < \mathrm{SEIDO}$ として調べる。

3.2　稜線決定

前の段階で抽出された直線は傾きと切片だけなので，これは不完全である。Fig. 10参照。そこで次に頂点の検出をする。抽出された直線どうしの組み合わせから，全交点を求める。但し，$0 \leqq x_1 \leqq 510$, $0 \leqq y_1 \leqq 479$にない（x_1, y_1）については画面外ということでその交点は無視する。また，画像上では交点座標は四捨五入されて表示さ

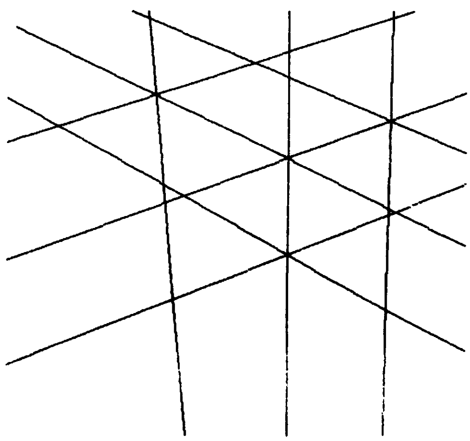

Fig.10　Selected lines by Hough transform

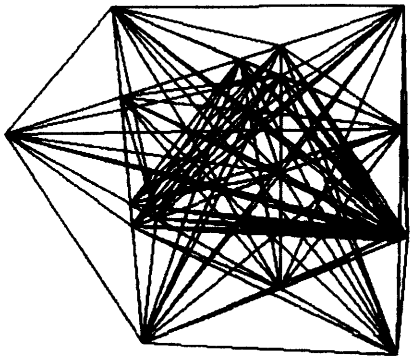

Fig.11 1'st candidates for edge

れるので，格納する交点は四捨五入しておく。そして，後でその交点がどの2直線の交点なのかを参照できるようにそれらの情報も格納しておく。全ての交点どうしの組み合わせによって出来た稜線候補群を Fig. 11 に示す。画像の稜線は稜線候補群の中に存在する。

次に求められた交点から頂点候補を選抜する。まず画像を表示し，各交点中心に5＊5ドットの枠内に画像上に点が存在するかどうかを調べ，1つも点が存在しなければ，その交点を頂点候補から外す。前処理においてかなり面密に雑音除去をしてあるので，この方法により画像エッジ上か，かなり近傍の点以外は頂点候補として認められない。

再び残った頂点候補どうしの組み合わせによってできた稜線候補群を Fig. 12 に示す。

頂点の選抜の結果は，現在与えられている直線が正しく再現できるものでなくてはならない。そこで今度は，与えられた直線を探索対象とする。与えられた直線N本（各直線をL_i；$1 \leqq i$，$j \leqq N$ で表す)，残った頂点候補M個（各頂点を P_k；$1 \leqq k \leqq M$で表す。)，交点としてのどの2直線による交点なのかという情報が各々2ずつ（頂点 P_k の情報を I（P_k, 1)，I(P_k, 2)；I(P_k, 1)＝L_i,，I(P_k, 2)＝L_j で表す）ある。

全てのI(P_k, 1)＝L_i, またはI(P_k, 2)＝L_iとなるようなP_kはL_iに属すると考える。尚，少なくとも，L_iに属するPは2個以上は必ず存在する。

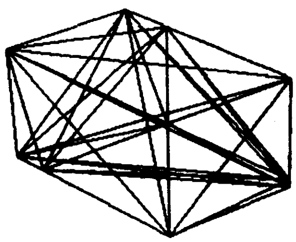

Fig.12 2'nd candidates for edge

各L_iにおいて属する全てのP間で距離が最大にものをひとまずL_iから導びかれる稜線と仮定する。

L_iに属するのがP_h, P_kと2つだけの時には，P_h, P_kを結んだものをL_iから導びかれる稜線と決定する。（こうすることで2次元画像の輪郭線は Hough 変換でしきい値をパスしている限り，かなりエッジが途切れていても再現が可能である。）

頂点を3個以上もっているL_iについては，必ずしも最大のものが真の稜線とはならない。Fig. 13 参照。そこで，L_iに属する全ての頂点間で線分L_{ipk}（P_k；頂点数）を引き，画像を表示し対応させて，L_{ipk}上に画像上の点の合計Sが｜L_{ipk}｜＊9/10以上でなければ稜線候補から外してゆく。最後には1対の頂点が必ず残るので，それがL_iの稜線となる。もしL_iの稜線候補が全てなくなった場合には，S／｜L_{ipk}｜が最大であるものを稜線と決定する。

この状態のままでは，頂点が何回か重複してしまう恐れがある。また同一頂点と見なされるべきものが誤差により2つの頂点と認識されてしまうこともある。従ってある程度の誤差のものは，それらを平均化することによ

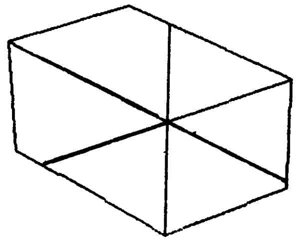

Fig.13 3'rd candidates for edge

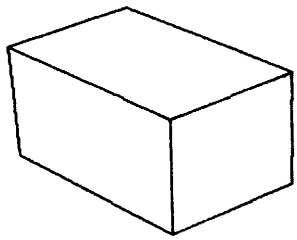

Fig.14 Reconstructed image

Table 1 Last detected edges

直線番号	(XS, YS)	−	(XE, YE)
1	(275, 57)	−	(165, 93)
2	(275, 57)	−	(428, 122)
3	(428, 122)	−	(313, 169)
4	(165, 93)	−	(313, 164)
5	(165, 93)	−	(174, 193)
6	(174, 193)	−	(313, 270)
7	(313, 270)	−	(425, 225)
8	(427, 122)	−	(425, 225)
9	(313, 164)	−	(313, 279)

り（複数の点の重心を求める）１つの頂点と見なすことにする。Table. 1 参照

3．3　頂点の対応付け

ワールド座標を決定するためには，視点の異なる２枚の画像を使用するが，その２枚の画像間の頂点の対応付けが必要である。

カメラ視点から画像１の頂点へ向かう直線はその直線のワールド座標を通る。その直線を別の視点から写した画像２上に投影したものが Epipolar line であり，その line 上に前の画像１上の頂点に対応する頂点が存在すると考えられる。

そこで Epipolar line と画像２上のすべての頂点の距離を計り，許容範囲内に存在する頂点が１つの場合はそれを対応付ける。複数の場合は逆に画像２上のそれらの各頂点から画像１へ Epipolar line を作り，対応付けを行っている画像１の頂点との距離が最小の Epipolar line が画像２での同一のものとして対応付ける。

4．考　　察

各処理の所要時間を表に示すとともに，評価する。

(1)　前回に比べ，エッジ検出の処理時間が大幅に向上しパソコン上での処理が簡易なものとなった。

(2)　Hough 変換によるエッジの当はめは高精度であり，分解能も高く，エッジに欠損があっても検出が可能である。

(3)　Hough 変換の精度を上げようとすれば，所要メモリ所要時間が比例して増大する。それゆえ，メモリの小さいパソコン上での処理は難しくなる。一方，メッシュを配列に格納できたので，ファイルとしてデータを扱う場合に比べて６倍速くなっている。よって，画像の解像度にもよるが処理に数時間とまではかからないであろう。

(4)　相対的に短いエッジが存在すると，直線として抽出されない恐れがあり，そのような状況に遭偶すれば，不自然な２次元画像が再生されることになる。微分値の低いエッジは検出しにくいので，エッジの接続状態やカラー情報を取り込んで，より安定した稜線の抽出が今後の課題である。

(5)　対応付けはエッジ上の点と点とで行うのが一般的だが今回は頂点間の対応付けを用いたので，処理の高速化が実現できた。

今回は前回に比べて大幅に時間が短縮され，また，全ての処理をパソコン上で実現しているので，コンパクトなシステムとなっている。

Table2　Process time

	本研究	昨年度
微　分　処　理	40秒(C)	10～15秒
非極大値の抑制 及び値処理	１分(C)	80分
直　線　抽　出	25分(F)	——
稜　線　抽　出	４分(B)	30分
使　用　言　語	C. BASIC FORTRAN	FORTRAN

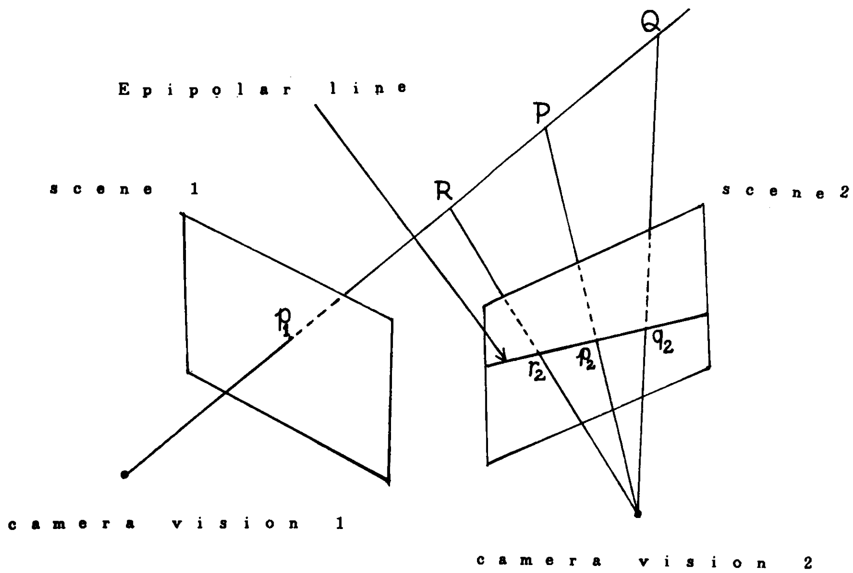

Fig.15　Epipolar line

一連の処理の中にはしきい値や精度などの設定が対話形式になっているので，今後の課題としては濃度やメッシュのヒストグラムから適当な値を決定できるよう画像処理の自動化の実現が望まれる。

参考文献

1）山本公之：凸多面体の再構成，名古屋工業大学大学院工学研究科修士論文，1985年

2）丹羽，山本：多面体の再構成，昭和60年電気関係学会東海支部連合大会，1985年. p. 453

3）山口富士夫：「コンピュータ・グラフィックス」（日刊工業新聞）

4）長尾真，監訳：「ディジタル画像処理」（近代科学社）

5）安居院猛，中島正之：「コンピュータ画像処理」（産報出版）

6）松山隆司，長尾真：「Hough 変換の幾何学的性質と直線群検出への応用」 情報処理学会論文誌 Vol. 26 No. 6