ステレオ法による凸多面体の再構成

丹羽敏行・渡辺凡夫
電気情報工学科
（1987年9月5日受理）

Reconstruction of Polyhedric Object by Stereo method

Toshiyuki NIWA and Tsuneo WATANABE
Department of Electrical and Computer Engineering
(Received September 5, 1987)

In these days, the research that gives computers visual recognition faculity, just like a human, is making progress as one section of A. I. It's called Computer Vision. The reconstruction of 3D-world from 2D-image is one of its theme for us. In general, there is one approach to reconstruct 3D-world and it's called Stereo Vision. This approach makes use of more than two pictures which have different view points and it is considered because human has two eyes.

This Stereo Vision has been developped very well, so we experiment on reconstruction of 3D-world basing to measure shape parameters of Polyhedric object, and show our trial in this report.

はじめに

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・ヴィジョンと呼ばれ，2次元画像から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1つのアプローチとして我々に目が2つあるように，2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複眼視処理と呼ばれる方法がある。2台のカメラを用いて，その両眼視差から3次元位置を計測するステレオ法は，良く研究されてきた。

今回，我々は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，高い信頼性の特徴点抽出と，高速の対応付けを基調として試みたので，その概要を報告する。

1．概　　要

ステレオ法では通常，2個のカメラを用いて，その両眼視差から3次元位置を計測する。今回，費用の節約とカメラ・パラメータの簡単化による計算量の節減のために，前回同様 Fig. 1 に示すように，カメラは1台で固定し，カメラ前方に設定されたターンテーブルを回転させて複数の画像を入力した。

今回も図形の直線抽出には Hough 変換を採用したが，大規模な記憶域を必要とする欠点を解消するために，1次元アキュムレイタ配列による Hough 変換を試行してみた。

また，悪状況での多面体の画像撮影においては，しばしばエッジの欠損が生じ，その結果，不自然な2次元図形を得ることがある。その状況に対し，よく知られている Huffman の辺縁と頂点の線画パターンに拘束条件を付け加えて，2次元図形補完を行った。

対応付けは頂点間で行うことにより，処理の高速化を計った。

扱う画像はカラー画像としたので，各画像につきR，G，B 3回の処理を要するが，多面体の各面に彩色したり，色のついた照明を用いることにより，エッジの検出を容易にすることも可能とした。尚，これら一連の処理は全てパソコンで処理した。

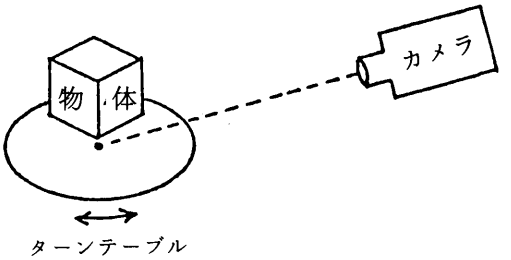

Fig. 1 Device configuration

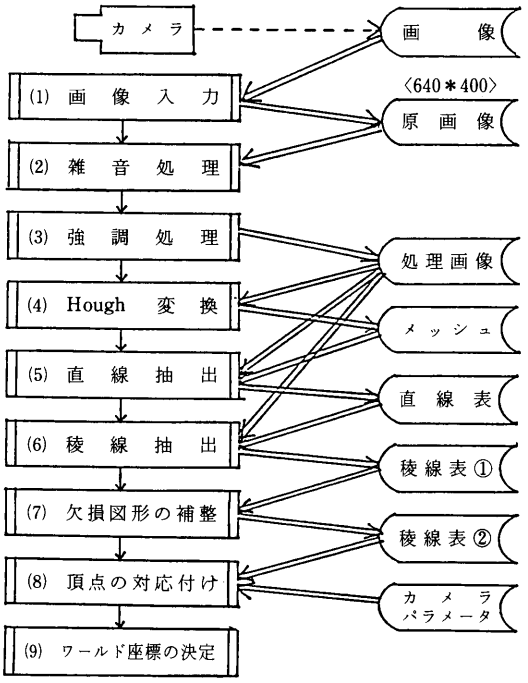

Fig. 2 Data and process flow

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の雑音除去，画像強調処理を行う。対象はR，G，Bを基調としたカラー画像なので，微分処理までは1枚の画像につき3プレーン分の処理を逐次施し，それらを再び1枚の画像に構成する処理を経て，非極大点抑制等を行う。

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（エッジ）がぼけてしまう。そこで，今回もエッジ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. 3 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。

2.2 強調処理（エッジの検出）

面像の分割のための重要な方法は，濃度やテクスチャーが急激に変化していて，1つの領域が終わり他の領域が始まっている様な不連続性（エッジ）を検出することである。濃度の急激な変化に伴って起こる線は，その両側の領域とは性質が異なった細い帯である。

こうしたエッジを検出するために画像の微分を行うのだが，扱う画像がディジタル画像なので微分の代わりに差分を用いる。

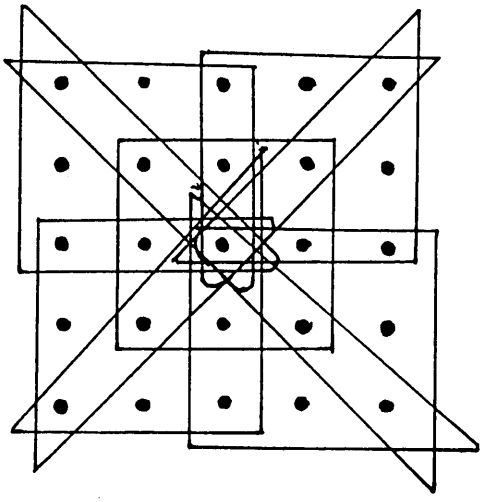

Fig. 3 Noise reduction mask

よく知られているディジタル勾配の近似に Roberts による方法があり，今回もこの “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。$M(i,j)$ が64階調を越す場合は64，アンダー・シュートの場合はその絶対値とする。

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

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。

1枚のカラー画像はR，G，B 3プレーンで構成されているので，微分画像を1枚のカラー画像に再構成する必要がある。今回は各プレーン6ビットの64階調を1バイトで処理を行ったため，各濃度値を計算速度とデータのコンパクト性を考慮して，R，G，Bの最大値を再構成の際の濃度値とした。Fig. 4 に微分画像と最構成画像を示す。Rで検出されなかったエッジがBやGで検出されていることがわかる。

先の微分処理により，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

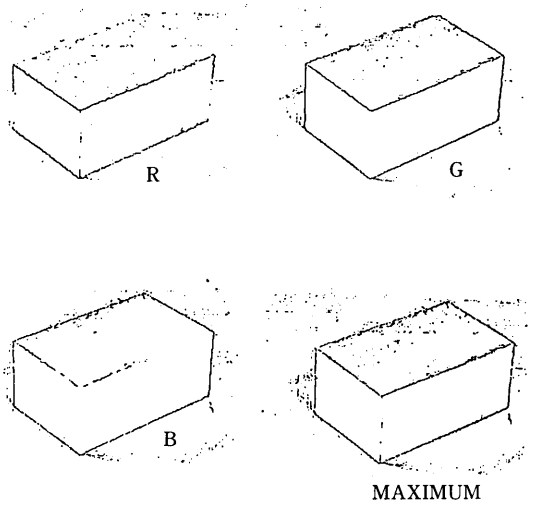

Fig. 4 Enhanced Image by gradient

ンラインを作り，以下の操作を行う。

IF　$D(P) \geqq D(P-1)$

THEN DELETE $D(P-1)$

P：スキャンライン上の点

$D(P)$：画素の濃度値

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。

非極大点の抑制を施した画像に次にしきい値処理を行い，エッジ以外の雑音と思われるような点を削除する。

非極大点の抑制及びしきい値処理画像を Fig. 5 に示す。

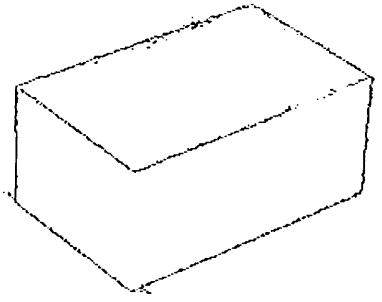

Fig. 5 Enhanced Image by suppression non-maximum

3．画像の直線抽出

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。

一般的には，直線の方程式(2)を(3)式の曲線に置き換えて量子化された（ρ，θ）空間での軌跡を追跡するのが Hough 変換の方法である。

$$y_i = a \cdot x_i + b \tag{2}$$

$$\rho = x_i \cdot \cos\theta + y_i \cdot \sin\theta \tag{3}$$

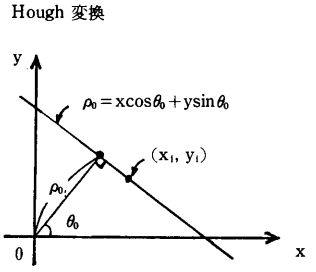

Fig. 6　(x－y) coordinates

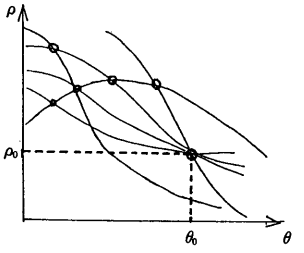

Fig. 7　($\rho-\theta$) space

この様に ρ と θ を新しい変数と考えた場合，(3)式は点（x_i，y_i）を通るように制約された直線のパラメータ間の関係を与え，(ρ，θ)空間における正弦曲線に対応する（Fig. 7)。画像平面における同一直線上の点の集合は,(3)式によって(ρ，θ)空間で，各曲線は(画像点を通る直線の実際パラメータに対応する）1つの共通点ですべて交わらねばならない。(($\rho-\theta$) 空間の（ρ_0，θ_0))

この様に同一直線上の点のクラスタを検出するため，その直線の各点について（ρ，θ）空間で変換曲線を作図し，3つあるいはそれ以上のこような曲線同時発生に対応する点（交点）を選ぶようにする。

ディジタル計算機でこれを実行するため，ρ，θ パラメータを量子化する。θ は0，2π の間で変化し，ρ は画像平面の大きさまでで十分である。(ρ，θ)空間における曲線の作図は，曲線に沿う量子化された(ρ，θ)空間の各座標点のカウントを増やすことに対応し，曲線の同時発生点を見つけるには，高いカウントの座標点を選べばよい。アルゴリズムの実現は（ρ，θ）2つの方向を記す2次元のアキュムレイタ配列を用いることに相当する。

性能は選んだ量子化に影響される。粗いディジタル化は近接している線の区別を困難にし，非常に細かい分解能では，典雅直線からほんの少し外れても，それを同一

直線とみなさなくなる。

故に，画像により適当な量子化レベルを選定する訳だが，粗いメッシュを選定しても必要とする記憶域がかなり大規模になり，メモリの不足が生じる。そこで，1次元配列による Hough 変換を行い，メモリ節約を試みた。

3．2　1次元 Hough 変換

($\rho-\theta$) 空間のために要する配列はかなりの規模になるが，Fig. 7 を見てもわかるように (x−y) 座標上の点の軌跡が ($\rho-\theta$) 空間において占める割合は極めて小さい。また，その軌跡の中でも必要とされるのは，画像の数ドット分の軌跡が局所的に集中している特定の(ρ_0, θ_0)座標であり，多少のずれはあるものの量子化された軌跡の交点である。

そこで θ を 0 から π までを量子化した1次元の配列を2つ (ROH [ ], SUM [ ]) と極大値を記録するための配列を2つ (MAXR [ ], MAXS [ ]) を用意する。

画像上のドット数を N として，(x−y) 座標上の任意の2点 (x_i, y_i)，(x_j, y_j) につき ($\rho-\theta$) 空間での軌跡の交点 (ρ_l, θ_l) を求め，量子化した θ を添字として ρ_l の値を ROH [(θ_l)] に格納し，SUM [(θ_l)] をインクリメントする。

ある点 (x_i, y_i) に対して N 回 (j=1, …N) 処理したあと SUM [ ] の極大点を MAXS [ ] に，その極大点の ρ_l の値を MAXR [ ] に各々格納する。

この処理を N 回 (i=1, …N) 行い，最終的にもう一度 MAXS [ ] の極大点を拾い，これを求める値 (θ_0, ρ_0) とする。MAXR [ ], MAXS [ ] に格納する際には前のラウンドの記録を更新することになるので，MAXS [(θ_1)] の値の大きい場合だけ書き換えることにする。

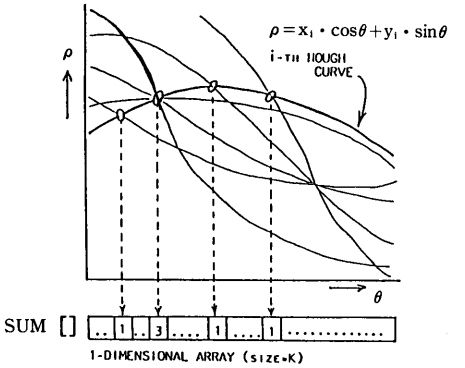

Fig. 8　1-D Hough Transform

今回扱った画像は同じ傾きを持つ直線が複数存在するのだが，線分の長さがほぼ等しく，θ を細かく量子化したことにより検出されない直線はなかった。

3．2　最小二乗法の適用

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，直線の傾きの大きさごとの誤差評価を行う。評価式(4)を満たす点を直線の構成要素とし，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求める。

CASE $|a| \leq 1$ THEN $|y_i-(x_i*a+b)|<\triangle$

CASE $|a|>1$ THEN $|x_i-(y_i-b)/a|<\triangle$

($i=1, \cdots\cdots, N$)　(4)

△：誤差許容値

a：Hough 変換より得られた直線の傾き

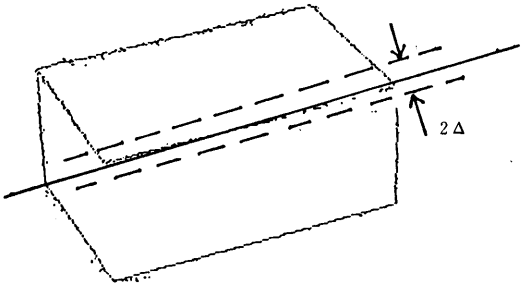

Fig. 9　Error range

得られる傾き切片は以下の良く知られた式となり，(5)式の a と(6)式の b を $y=ax+b$ に代入して回帰直線の式となる。

$$a=\frac{(n\Sigma xy-\Sigma x\Sigma y)}{(n\Sigma x^2-(\Sigma x)^2)} \quad (5)$$

$$b=(\Sigma y-b\Sigma x)/n \quad (6)$$

3．2　稜線決定

前段階で抽出された傾きと切片だけの直線から頂点の検出を行う。抽出された直線どうしの組み合わせから，画像範囲内 ($0 \leq x \leq 510$, $0 \leq y \leq 479$) の全交点を求める。後でその交点がどの2種類の交点なのかを参照できるよう，それらの情報を格納しておく。全ての交点の組み合わせによって得られた稜線候補群を Fig. 11 に示す。画像の稜線は稜線候補群の中に必ず存在する。

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綿密に雑音除去を施してあるので，この方法により

画像のエッジ上か，かなり近傍の点以外は頂点候補として認められない（Fig. 12)。

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。

SET Line $\{L_1, \cdots\cdots, L_i, L_j, \cdots\cdots, L_N\}$

(L_i, L_j：各直線，得られた直線 N本)

SET Vertices $\{P_1, \cdots\cdots, P_h, P_k, \cdots\cdots, P_M\}$

(P_h, P_k：各頂点，頂点候補 M 個)

SET Edge $\{L_{i1}, L_{i2}, \cdots\cdots, L_{ic}\}$ *for* L_i

(L_{i1}, L_{i2}：直線 L_iより得られた線分 C 本)

FUNC. Attrib : どの2直線による交点かの属性

EX *Attrib* $(P_k, 1) = L_i$,
Attrib $(P_k, 2) = L_j$

IF *Attrib* $(P_k, 1) = L_i$ *OR*
Attrib $(P_k, 2) = L_1$

THEN {*Line* L_i*has Vertex* P_k}
(L_i*has at least 2 Vertices !*)

FUNK. Number : *Attrib* $(P_k, 1$ *or* $2) = L_i$となる頂点の数

EX Number $(L_i) \geqq 2$

FUNC. Revers : どの2頂点による線分かの属性

EX Revers $(L_{i1}) = \{P_h, P_k\}$

関数値が L_iとなる全て P_kは L_iに属すると考える。尚，少なくとも，L_iに属する頂点は2個以上は必ず存在する。稜線は以下の方法で決定する。

```
/*for Line Li*/
IF    Attrib (Ph, 1 or 2)=Li AND
      Attrib (Pk, 1 or 2)=Li AND
      Number (Li)= 2
THEN  Revers (Li){Ph, Pk} :
      /*Line Li's Vertices are Pk & Ph*/
ELSE   {
       MAX= 0 ;
       C=Number (Li)C2 ; /*C2 : Combination */
     FOR I= 1 TO C {
            IF (Si/ | Li1 | )>MAX THEN {
              MAX=Si/ | Li1 | ;
              R=I :
            }
     }
       Revers (LiR){Ph, Pk} ;  /*={Ph, Pk} */
       /*Line L1's Vertices are Pk & Ph*/
}
```

L_iに属するのが P_h，と 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

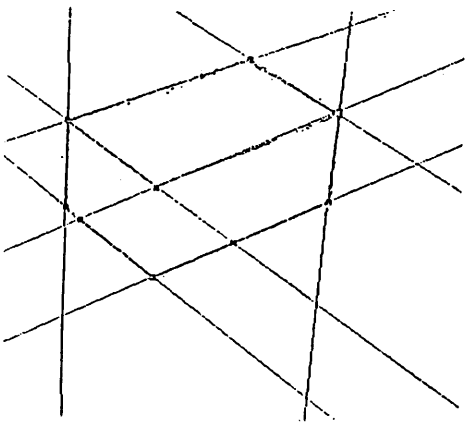

Fig.10 Selected lines by Hough Transform

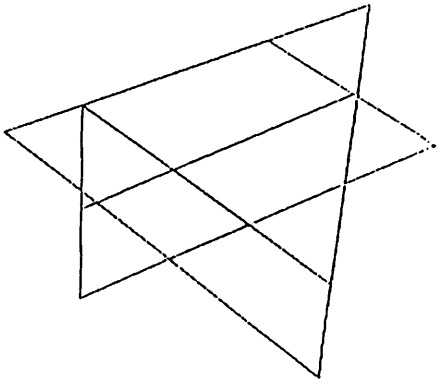

Fig.11 1'st candidates for lines

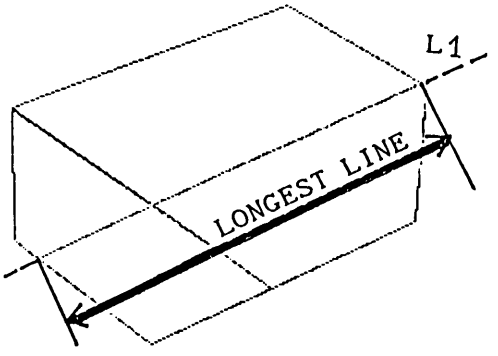

Fig.12 2'nd candidates for lines

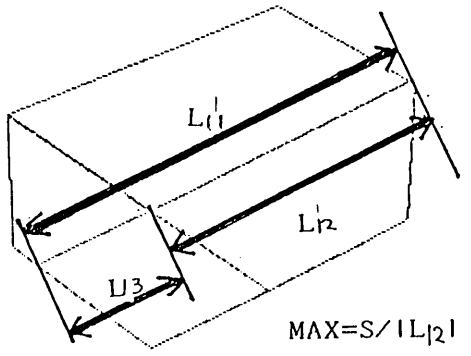

Fig.13 Appreciation of lines

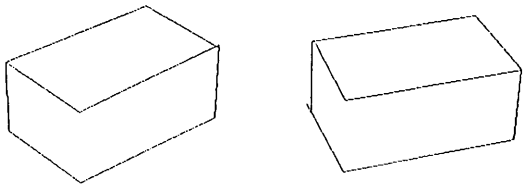

Fig.14 Reconstructed figurs

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，L_i上に存在する画像上の点の合計を S_lとする。L_iに属する全ての頂点間で線分 L_{il}（P_hP_k）を引き，$S_l/|L_{il}|$が最大であるものを稜線と決定する（Fig. 13)。

図形は閉じているので，かなり近傍の頂点は平均（複数の点の重心）を求めることにより1つの頂点と見なすことにする。こうして稜線表を作成し，得られた図形がFig. 14 である。

3.3 図形補完

Fig. 14 を見ると頂点の状況，線のつながり，稜線の欠損など人間の目から見て明らかに不自然な図形となっている。そこで，(1)頂点の修正，(2)稜線の補完，という順序で図形の補完をおこなう。予め各頂点のつながりを示す頂点リストを作成しておく。

(1) 頂点の修正

1. 図形の輪郭線を探索し，記録する。
2. 輪郭線上の頂点を検出する（Fig. 15)。
3. 稜線図形より輪郭線以外の稜線を削除する。(Fig. 16)
4. もう1度輪郭線探索をして，消えた輪郭線上の頂点があれば，その近傍の頂点が本来の頂点と見なす。
5. 頂点リストと輪郭線上の頂点より本来存在しない稜線を検出する（Fig. 17)。
6. Fig. 16 の頂点AかBのどちらが矛盾するのかを調べる。AをBに，またはBをAに重ね，輪郭線の一筆画きのできなかった場合，重ねられた側の頂点が矛盾する点とみなし，重ねた側の頂点に置き換える。

単一の多面体では Huffman の線画パターンのT型は含まないとすると，常に頂点と頂点は結びつき，稜線上の端点や孤立した頂点は存在し得ない。つまりAかBの

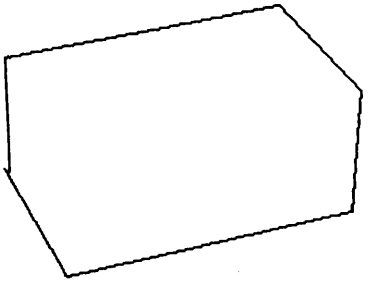

Fig.15 Contour tracing

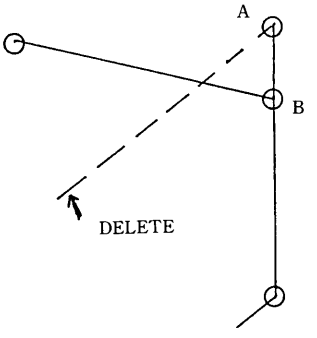

Fig.16 Deleted line

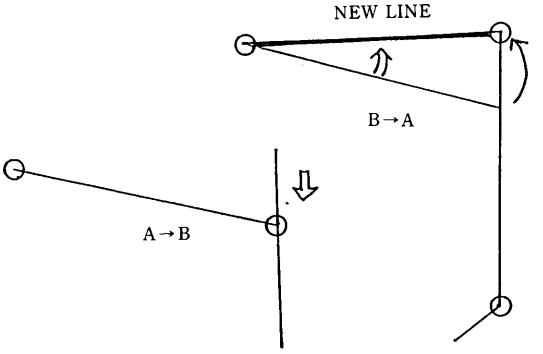

Fig.17 Moved vertex

どちらかがおかしいということになり，これはもともとAとBが同一の頂点であったことが理由といえる。どちらかを1方に頂点座標を重ねて一筆画きをすると矛盾する点に重ねたとすると新しい稜線がディスプレイ上の図形に現れる。

画像を修正する場合，少なくとも輪郭線はあい昧ではないということを前提にしている。実際には Hough 変換後の誤差評価式の条件を緩めて構成要素をふやせば，頂点の矛盾は起こりにくいと思われる。

(2)　図形補完

1．頂点の修正処理(1)後の図形に対して Huffman の線画パターンによる稜線の符号化を行う。輪郭線を構成する稜線を→で，図形内部は凸稜線として，＋の符号を与える(Fig. 18)。これは最初に扱う対象を凸多面体に限定しているためである。

2．地面に接している稜線に対し，凹であるマイナスの符号をつける（Fig. 19)。

3．各頂点において，Huffman の線画パターンを満たさない頂点を摘出する。

4．矛盾する頂点どうしを結び，新たにこの稜線を頂点リストに付け加える（Fig. 20)。

Huffman の線画パターンは実在する図形に対しては

A：TYPE ARROW (−>, +, −>)

C：TYPE Y(+,+,+)

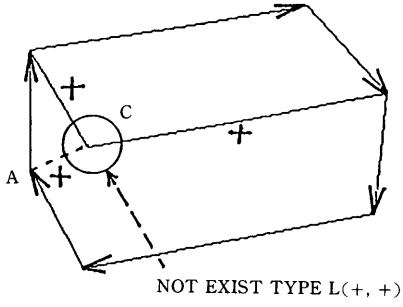

Fig.18　Huffman pattern

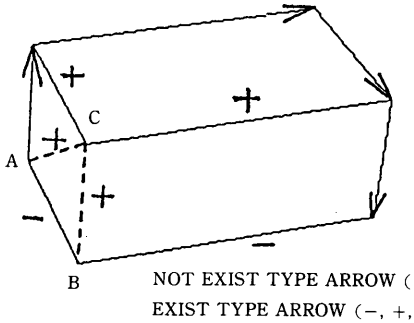

Fig.19　Huffman pattern with other constraints

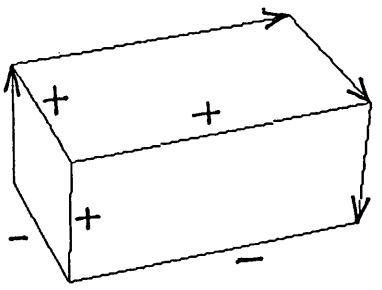

Fig.20　Complemented figure

条件を満たすが，これを満たすからといって必ずしも与えられた図形が実在するとは限らない。

今回の図形（Fig. 18）も同様であり，何等矛盾は生じないが，この図形は不自然である。よって，この必要条件の他に拘束条件として，多面体が空中に浮いているのではなく，地面に接しているという状況（Fig. 19）を与えてやる。するとBは矛盾を呈し，Aは矛盾を生じないので，初めからの矛盾点CとBを凸であるプラスの符号をつける（Fig. 20)。

接地している稜線は垂直に近い稜線とつながるY座標の大きい側の頂点間の稜線を摘出する。

4．3次元再構成

4．1　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。(Fig. 21)

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り，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。

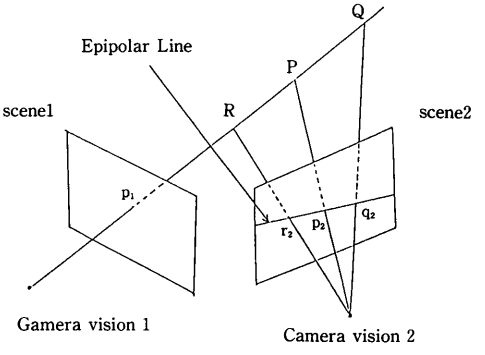

Fig.21　Epipolar line

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 を画像2での同一のものとして対応付ける。

4.2 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。先ずカメラは次の様に設定する。

(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，さらに光軸がフィルム面上で水平に成るようにZ軸回りに γ だけ回転した時の変換マトリクス T_1は次のようになる。

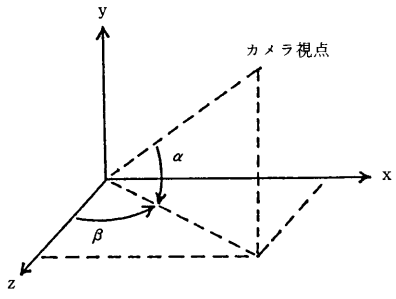

Fig.22 World coordinates and camera

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

$$\begin{pmatrix} cos\gamma & sin\gamma & 0 & 0 \\ -sin\gamma cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

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，$k=1/d$ とする。これより，CRT座標への変換マトリクスは次のようになる。

$$T=T_1T_2T_3=\begin{pmatrix} T_{11} & T_{12} & 0 & T_{14} \\ T_{21} & T_{22} & 0 & T_{24} \\ T_{31} & T_{32} & 0 & T_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$T_{11}=k(\cos\beta\ \cos\gamma-\sin\alpha\ \sin\beta\ \sin\gamma)$

$T_{21}=-k\cos\alpha\ \sin\gamma$

$T_{31}=k(\sin\beta\ \cos\gamma-\sin\alpha\ \cos\beta\ \sin\gamma)$

$T_{12}=k(\cos\beta\ \sin\gamma-\sin\alpha\ \sin\beta\ \cos\gamma)$

$T_{22}=k\cos\alpha\ \cos\gamma$

$T_{32}=k(\sin\beta\ \sin\gamma-\sin\alpha\ \cos\beta\ \cos\gamma)$

$T_{14}=\cos\alpha\ \sin\beta/\mathrm{d}$

$T_{24}=-\sin\alpha/\mathrm{d}$

$T_{34}=-\cos\alpha\ \cos\beta/\mathrm{d}$

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

$[x\ y\ z\ 1]\ T=h\ [x^*y^*\ 0\ 1]$

$T_{11}x+T_{21}y+T_{31}z=hx^*$

$T_{12}x+T_{22}y+T_{32}z=hy^*$

$T_{14}x+T_{24}y+T_{34}z=h-1$

となり，h を消去して次の式が得られる。

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

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

変換マトリクスは未知なので，予め既知のモデルを用いて算出しておかなければならない。

マトリクス要素 T_{ij}は計9個であり，4行4列目の要素（$T_{44}=1$）をそれらに入れて未知数10個とする。

上の式は1点の座標 (x, y, z) と (x^*, y^*) から2方程式を導くので，5点のワールド座標 (x, y, z) とスクリーン座標 (x^*, y^*) が与えられれば，マトリクスの未知の T_{ij} 9個の値が求まることになる。

カメラパラメータは各 T_{ij}からの逆算で求まり，以下の式で導かれる。

$r=tan^{-1}(T_{21}/T_{22})$

$$\phi=\frac{T_{12}^2+T_{32}^2-T_{11}^2-T_{31}^2}{(T_{12}+T_{32})\cdot sin\gamma-(T_{11}+T_{31})\cdot cos\gamma}$$

$$\alpha=cos^{+1}\left(\frac{|\,T_{22}\cdot cos\gamma\,|\cdot\phi}{-T_{22}\cdot cos\gamma}\right)$$

$$k=\left(\frac{T_{11}+T_{31}}{1-\phi^2\cdot sin^2\gamma}\right)^{1/2}$$

$$\beta=cos^{-1}\left(\frac{T_{31}\cdot sin\gamma-T_{32}\cdot cos\gamma}{k\cdot sin\alpha}\right)$$

$d=-sin\alpha/T_{24}$

$l=d\cdot k$

尚，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できる。Table. 1 に計算された各々のカメラ・パ

Table. 1　Camera parameter

＊＊＊T＊＊＊	camera 1	camera 2
T11(0)	18.02752152	13.59502500
T12(1)	4.60362447	6.62021449
T14(2)	0.00521914	0.00765678
T21(3)	−1.12082858	−0.95532822
T22(4)	18.55617692	18.46143253
T24(5)	−0.00603841	−0.00549451
T31(6)	8.98280522	14.58401575
T32(7)	−7.56510328	−5.51462124
T34(8)	−0.00912176	−0.00698027
T41(9)	0.03771466	−0.01002540
T42(10)	−0.25419385	0.04586298
α = 25.82730°	α = 25.41356°	
β = 22.93818°	β = 43.03520°	
γ = 3.45658°	γ = 2.96226°	
k = 20.171283	k = 19.959615	
d = 72.148125	d = 78.104963	
l = 1455.320258	l = 1558.944968	

ラメータの値を示す。

そしてカメラ・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。カメラ，パラメータを導出するために用いたモデルのワールド座標値を示し，その安定性を確かめた。

$$X=(A^TA)^{-1}(A^TB)$$

$$X=\begin{pmatrix}x\\y\\z\end{pmatrix}\qquad B=[x^{*1}y^{*1}x^{*2}y^{*2}]^T$$

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

5．考　　察

ここで各処理の評価をしてみる。

(1)　全てパソコン上の簡易な処理システムとなった。

(2)　2次元配列の Hough 変換と比較して1次元配列での Hough 変換では，所要メモリがかなり節約できた。しかし，計算量については前者が N・d（d：θの量子化レベル）に対し後者は N^2のオーダーとなり，所要時間も二乗倍となった。

(3)　Hough 変換によるエッジの当てはめは異なる傾きの直線に対しては高精度であるが，平行直線に対しては常に抽出可能とはならなくなるので，より良いアルゴリズムが必要である。

(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遇すれば，不自然な2次元画像が再生されることになる。幸い，Huffman によるパターンのマッチングにより図形の補完が可能となった。

(5)　カメラ・パラメータ（α, β, γ, k, d, l）は理論的に単純な計算で得られるのだが，誤差に敏感なので，今回はかなり複雑な式からの導出を強いられた。

しかし，それだけ誤差の影響は大きくなるので，行列式による処理以外の方法を考案するのが好ましいと思われる。

(6)　対象を彩色したり，色照明などにより，エッジの接続状態やカラー情報を取り込み，より安定した稜線の抽出が今後の課題である。

今回も全ての処理をパソコン上で実現し，コンパクトなシステムとなっているが，それでも一連の処理を全て実行するとかなりの時間といえる。

謝　　辞

日頃より有益な御助言を戴く石井直宏教授（電気情報工学科）に感謝いたします。

参 考 文 献

1）山口富士夫：「コンピュータ・グラフィックス」（日刊工業新聞）。
2）R. Nevatie：「画像認識と画像理解」（啓学出版）。
3）安居院猛，中島正之：「コンピュータ画像処理」（産報出版）。
4）尾崎　弘，谷口慶治，小川秀夫：「画像処理―その基礎から応用まで」（共立出版）。
5）長尾　真：「ディジタル画像処理」（近代科学社）。
6）長尾　真：「パターン認識と図形処理」（岩波書店）。
7）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.
8）輿水大和：「直線群検出のための Hough 曲線追跡型アルゴリズム」　電子通信学会論文誌 Vol. J69-D No. 4.