論　文

Hough変換を用いた線分検出の高精度化

正　員　大和　淳二[†]　　正　員　稲葉　稔智[††]
正　員　石井　郁夫[†]　　正　員　牧野　秀夫[†]

Highly Accurate Segment Detection Using Hough Transform

Junji YAMATO[†], Toshinori INABA[††], Ikuo ISHII[†] *and* Hideo MAKINO[†], *Members*

あらまし　Hough変換は画像処理における直線(線分)検出の有力な一手段であるが，要求される検出精度の高度化に伴い，所要メモリ量と処理時間の激増を招く点に実用上の問題がある．本論文では，Hough変換を使用して，メモリ量の増大を伴うことなく線分を高精度で検出するための方法を提案する．この方法は，従来のθ-ρパラメータ平面だけに頼る線分検出手法から脱脚し，線分検出の主な処理をパラメータ平面からx-y平面に戻して行うこと，検出を終了した線分を構成する画素に対応するHough曲線をパラメータ平面から除去することなどにその特徴がある．実験により，多数の長・短線分を含むノイズのある図面から，高い精度で線分を検出し得ることを確かめた．

1. まえがき

Hough変換[(1),(2)]は画像処理における線分(直線)検出の有力な一方法で，ノイズのある図面からも安定して線分を検出できる特長があり，種々の分野で応用が試みられている[(3),(4)]．しかしこの方法には，要求される検出精度の高度化に伴ってθ-ρパラメータ空間用メモリ量と処理時間の激増を招く欠点があり，また得られる検出精度にも限界がある．更に，長・短多数の線分を含む図面から線分を検出する際の困難性にも問題がある．

本論文は，このような点の改善を目的とする新しい線分検出方法(Hough変換を使用)を提案するもので，**2.**にその基本原理を示し，**3.**と**4.**でこの方法の特徴であるθ-ρパラメータ平面からx-y平面への処理の帰還と，パラメータ平面からのHough曲線の除去について述べる．更に**5.**ではこれらを組み合わせて多数の長・短線分を含む図面から線分を検出するための方法について述べ，**6.**で数種の図形による実験結果を示す．

† 新潟大学工学部情報工学科，新潟市
Faculty of Engineering, Niigata University, Niigata-shi, 950-21 Japan
†† 沖電気工業株式会社，東京都
Oki Electric Industry Company Ltd., Tokyo, 108 Japan

2. 線分検出高精度化の基本原理

ここに提案する線分検出方法の第1の特長は，線分検出の処理をθ-ρパラメータ平面だけに頼る従来の手法から脱脚してパラメータ平面からx-y平面に戻し(ここではこれを帰還と呼ぶ)，主要な検出処理をx-y平面で行うことである．すなわちこの方法ではHough変換をx-y平面での線分の位置の大略の把握に用い，その詳細な解析はx-y平面上で原図形を対象に行うため，従来の方法では大きな誤差を生じる短い線分の検出も，高精度で行うことが可能となる．

第2の特長は，1本の線分の検出を終えるつどその線分を構成するすべての画素のHough曲線をパラメータ平面から減算除去することであって，これにより短い線分の検出が容易になり，長・短線分が混在する複雑な図面からの線分の検出が可能となる．

3. 帰還による沿線直線の検出

ここでは1本の線分(L)で構成された図面を例にとり，パラメータ平面からx-y平面への帰還によってこの線分(L)に最も近い所を通過する直線(沿線直線と名づける)を求める方法について述べる．

3.1 第1候補直線の設定と帰還直線

在来の方法によってパラメータ平面からその累積度

数のピーク点 (θ_i, ρ_j) を求め，式(1)

$$\rho_j = x \cdot \cos(\theta_i) + y \cdot \sin(\theta_i) \quad (1)$$

で表される直線 (l) を x-y 平面上に描く．以後このようにしてパラメータ平面から求めた x-y 平面上の直線 (l) を，帰還直線と呼ぶことにする．

この帰還直線 (l) は，検出しようとする線分 (L) の近傍を通過してはいるが，それに最も近いところを通過するとは限らないので，ここではこれを第1候補直線 (l_1) と呼ぶことにする．

3.2 線分構成画素の概数の推定

次節で使用する必要上，線分 (L) を構成する画素(以後これを黒画素と呼ぶことにする)の総計の概数 N を求めることにする．それは，図1に示すパラメータ平面の度数ピークセル (θ_i, ρ_j) と，それを中心として ρ 軸方向に隣接する2個のセル (θ_i, ρ_{j+1})，(θ_i, ρ_{j-1}) の累積度数 A，B，C の和として，式(2)により求めることができる．

$$N = A + B + C \quad (2)$$

その理由は，Hough 曲線を描く際に θ_i をインクリメントしつつそれに対応する ρ のセルに1度数ずつ累積するので，線分 (L) の各黒画素に対応する Hough 曲線の画素は，パラメータ平面の θ 軸の各列中にそれぞれただ1回だけ必ず累積されるからである．なおここでは2個の隣接セルの累積度数を加算したが，その個数は線分 (L) の黒画素の並び方の斎一性(一直線上にいかに正確に並んでいるか)に依存し，その低下に伴って増加する必要がある．

3.3 第2候補直線の推定

前節の方法によって求めた第1候補直線 (l_1) と線分 (L) の方向および位置は，パラメータ平面の θ 軸と ρ 軸の分割数が有限であることや x-y 平面内のノイズなどの影響を受けて，一般には一致しない．そしてこの傾向は線分が短くなるほど顕著になる．そこで第1候補直線 (l_1) を参照しつつ，線分 (L) と更によく一致する第2候補直線 (l_2) を次の方法で求める．

すなわちまず第1候補直線 (l_1) に沿って2点 $(x_l + M, y_l + M)$，$(x_l - M, y_l - M)$ を対角線とする正方形ウィンドウを，図2に示すごとく互いに重ならないように移動させる．但しここで (x_l, y_l) は第1候補直線 (l_1) 上の座標であって，次式で表される．

$$y_l = [(\rho_j - x_l \cdot \cos(\theta_i)) / \sin(\theta_i)] \quad (3)$$

ここで $x_l = 2Mn$，$(n = 1 \sim [255/(2M)])$，また [ ] はガウス記号である．

実際にウィンドウを移動するにあたっては，まず最初に $M = 1$ (単位は画素)として行う．移動を終了した時点でこれらのウィンドウ内に入った黒画素の総計が前節で求めた N の70％以下の場合には，70％以上になるまで M の値を順次1ずつ大きくしながらこの操作を繰り返す．このようにして得られた最後の M の値を M_1 とする．

この処理を終了した際，最後のウィンドウ群の内に含まれていた黒画素の大部分が線分 (L) のものと考えられる．従ってそれらの座標値を最小2乗誤差近似する直線を下記の(1)または(2)の方法で求め，これを第2候補直線 (l_2) とする．

(1) $45 \leqq \theta_i < 135$ あるいは $225 \leqq \theta_i < 315$ の場合．
求める第2候補直線を

$$y = m \cdot x + c \quad (4)$$

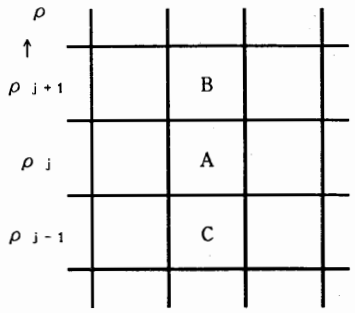


図1 直線を構成する画素数の推定法
Fig. 1 Estimation for the total pixels making up a segment.

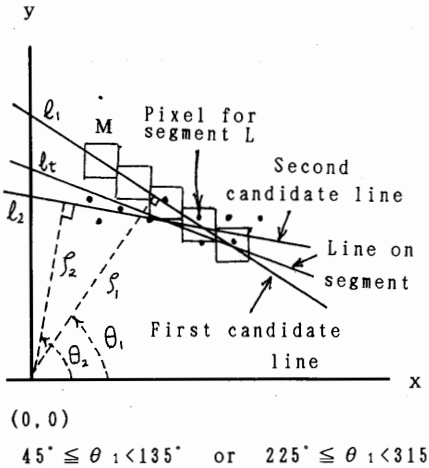


図2 第2候補直線の推定法
Fig. 2 Estimation for more desirable line.

と置き，ウィンドウ群に入った n 個の黒画素の座標を (x_1, y_1), (x_2, y_2), …, (x_n, y_n) とすると，それらの式(4)に対する誤差 e は

$$e_s = y_s - m \cdot x_s - c, \quad (s=1 \sim n) \qquad (5)$$

となるので，e_s の 2 乗和を最小にする条件から m および c を求めると，次のようになる．

$$m = (n \cdot \Sigma x_s \cdot y_s - \Sigma x_s \cdot \Sigma y_s)/D \qquad (6)$$

$$c = (\Sigma x_s^2 \cdot \Sigma y_s^2 - \Sigma x_s \cdot \Sigma x_s \cdot y_s)/D \qquad (7)$$

但し $D = n \cdot \Sigma x_s^2 - (\Sigma x_s)^2$ で，また Σ は $s=1 \sim n$ の全画素について加算する．

(2) $-45 < \theta_i < 45$ あるいは $135 < \theta_i < 225$ の場合．

求める第 2 候補直線を

$$x = m \cdot y + c \qquad (8)$$

として，m と c を前回同様に求める．

このようにして求めた第 2 候補直線 (l_2) は，以後極座標形式 (パラメータは θ_2 と ρ_2) に変換して取り扱うことにする．

3.4 第 2 候補直線の検定

多量のノイズなどを含む図面では，前節で得た第 2 候補直線 (l_2) も必ずしも線分 (L) に最も近いところを通過するとは限らない．従ってその検定が必要になる．それは次のようにして行う．すなわち第 2 候補直線 (l_2) に沿って，前節で得た M_1 の大きさのウィンドウをもう一度移動させ，その中に入る黒画素数の総数を求める．それが N の 95 % に満たないならば，この第 2 候補直線 (l_2) を前述の第 1 候補直線に置き換えて **3.3** からの操作を繰り返す．

但しこの繰返しは最大 5 回程度とし，もしその間に黒画素の総計が N の 95 % を超えなかった場合には，黒画素の総計が最大になった際の m と c の値を採用する．この繰返し回数は，図面中のノイズの量，第 1 候補直線の線分 (L) への近似の程度等によって異なるが，後述の実験例では 2～3 回であった．このようにして得られた最終の直線が，求める沿線直線 (l_t) である．

4. Hough 曲線のパラメータ平面からの除去

任意の方向を向いた長・短多数の線分を含む図面からすべての線分を検出することは，従来の方法では困難である．それはすべての線分の検出をパラメータ平面に手を加えることなく行っているので，短い線分を検出する際にそのピーク位置が長い線分の Hough 曲線によってかく乱され，不明りょうになるからである．

しかしこの問題は，1 本の線分の検出を終了するつど，その線分の黒画素の Hough 曲線をパラメータ平面から除去することによって解決できる．なぜならば線分は度数ピークの大きいもの，すなわち長いものから順次検出されるので，このようにすれば短い線分でもそれが検出される時にはパラメータ平面では最大の度数ピークをもつ線分となっており，その検出が容易になるからである．

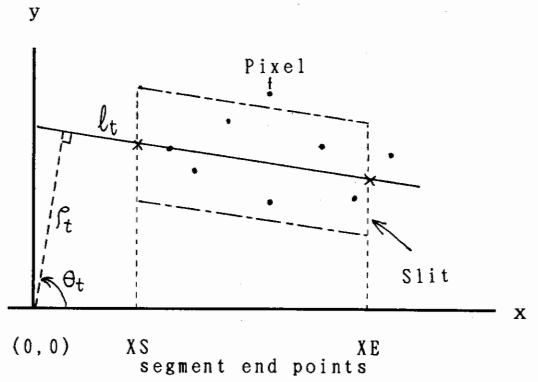


図 3 Hough 平面から減算除去するスリット領域
Fig. 3 Slit area including pixels whose Hough curves are removed from Hough plane.

次にその具体的な処理方法を述べる．**3.** の方法で求めた沿線直線 (l_t) を

$$\rho_t = x \cdot \cos(\theta_t) + y \cdot \sin(\theta_t) \qquad (9)$$

とし，$XS < X < XE$ (または $YS < Y < YE$) の範囲を区切って図 3 のような，数画素幅 (画素幅は線分 (L) の斎一性に依存する) の平行四辺形のスリットを設ける．そしてこのスリット内の黒画素はすべてこの線分 (L) を構成するものとみなし，それらの黒画素に対応する Hough 曲線をパラメータ平面から除去 (累積減算による) する．なおここで XS, XE (または YS, YE) は線分 (L) の両端点の $X(Y)$ 座標値であるが，その求め方については後に **5.2** で述べる．

5. 複数の線分の検出

3. および **4.** で述べた方法を組み合わせ，更に線分とノイズの判別，線分の端点検出などのアルゴリズムを付加すると，多数の長・短線分を含む複雑な図面から線分を検出することが可能となる．本章ではそのアルゴリズムを図 4 の流れ図によって説明する．

5.1 線分画素とノイズ画素の判別

パラメータ平面における累積度数ピーク点は線分画素によるものとは限らず，ノイズ画素による場合もある．従ってそれがいずれによるものであるかをまず最初に判別する必要がある．この判別は，度数ピーク点 (θ_i, ρ_j) の帰還直線，すなわち第 1 候補直線 (l_1) の近傍

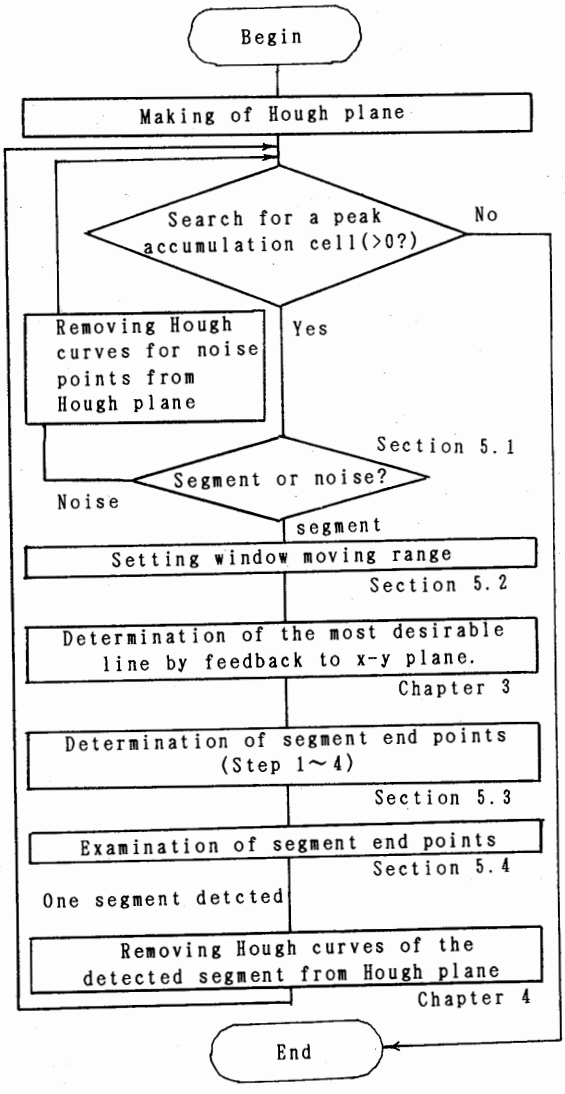


図4　線分検出の流れ図
Fig. 4　A flow chart for segment detection.

の黒画素の密度の大小によって行う．

具体的には図5に示すように第1候補直線(l_1)を中心に数画素幅のスリットを設け，その内にある黒画素のx軸($-45\leqq\theta_i<45$あるいは$135\leqq\theta_i<225$の場合にはy軸)への投影を配列Iに作る．その際黒画素を'1'，背景画素を'0'に対応させ，また第1候補直線(l_1)が図面から外れるところは，例えば'9'としておく．次にこの配列Iから式(10)に示すBの

$$B=\frac{1}{255-m_0}\sum_{k=1}^{M}\left(\frac{1}{m_k+1}\right)^n,\quad (n=3,255\geqq M)\tag{10}$$

値を求める．ここでm_iは隣接する'1'の間に介在する'0'の数で，その計算例を図6に示す．

このBの値が大きければ黒画素の列は密であり，第1候補直線(l_1)に沿って線分を構成するものと判断してよい．逆にBの値が小さいときにはその密度は低く，ノイズと判断される．この処理によってノイズと判断された黒画素のHough曲線を，パラメータ平面から除去する．

この判別方法は，線分が図面のほんの一部分にしかない短い場合にも効果的であり，後述する実験例等では，$n=3$，$B>0.05$として良い結果を得た．

5.2　ウィンドウ端点の検出

線分の数が多いときには，第2候補直線を求める際にウィンドウ幅を絞ることによってその位置精度を上げる必要がある．これを行うため，次のような方法によってウィンドウの両端点を決める．

すなわち，図7の配列Iの$I(1)$から順にその値を追って行き，要素'1'を配列$I(t_1)$で見つけたとき，そこをウィンドウの開始端点の候補とする．次に4個以上連続する要素'0'を見つけると，その直前の要素'1'の位置$I(t_2)$をウィンドウの終了端点の候補とする．次に両候補端点の中間にある要素'1'の数を調べ，それが幅(t_2-t_1)の4割以下ならその間の黒画素はノイズであるとみなし，この候補端点の組を無視する．もし配列Iの中に複数の候補端点の組が存在する場合には，その中で最も幅の広い1組をウィンドウの両端点として採用し，他は無視する．このようにして得られた端点の間の黒画素を対象に，**2.**で述べた方法によって沿線直線(l_t)を作るわけである．

長さの異なる至近の距離にある2線分の場合からも予測されるように，ウィンドウの端点の組と線分の端点の組は一致しない場合もある．

5.3　最終的な線分端点の決定

前節で求めた沿線直線(l_t)に沿う黒画素の配置を分析して最終的な線分の端点を求めるが，その方法について以下に説明する．

[Step 1]　沿線直線(l_t)をたどることによってx軸上(またはy軸上)に線分の画素の投影を作り，配列Iに収容する．その作成方法は**5.1**の場合とほぼ同じであるが，相異点は検出済みの線分の画素にはその検出を終えた時点で例えば'7'を入れておき，沿線直線(l_t)をたどる際に'7'を見つけたらその画素に対応する配列Iの要素に例えば'2'を入れることである(図8参照)．

[Step 2]　配列Iの'2'の要素の部分については，下記の判断基準(1)，(2)により，真に交差している部分，す

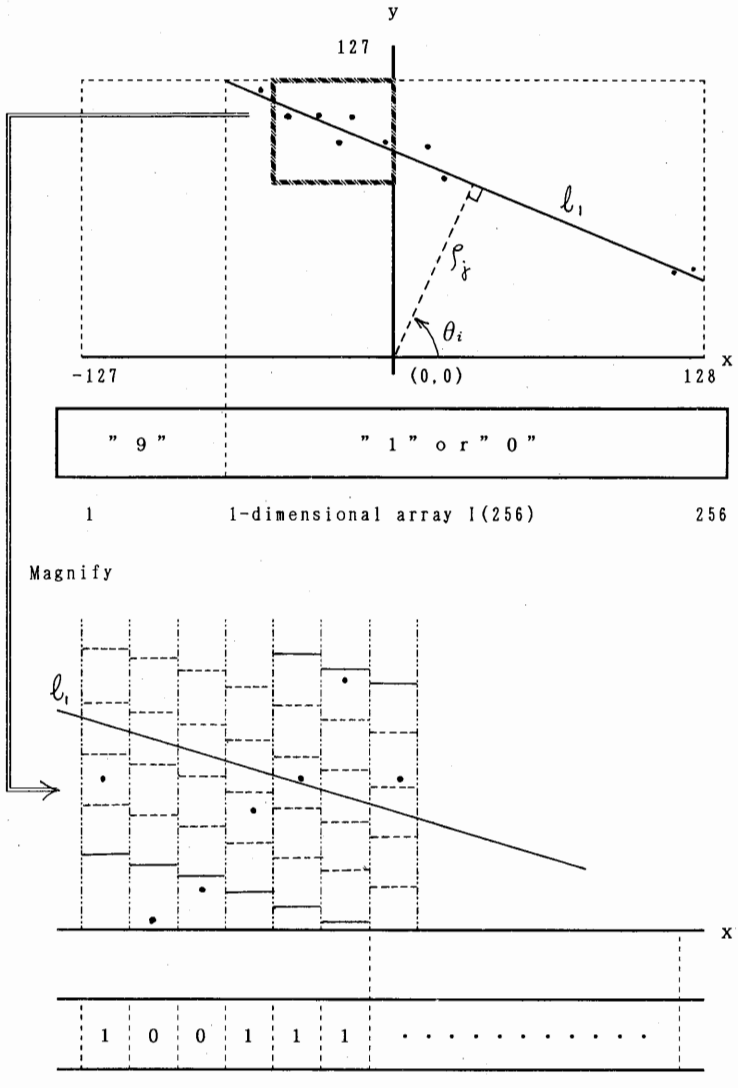
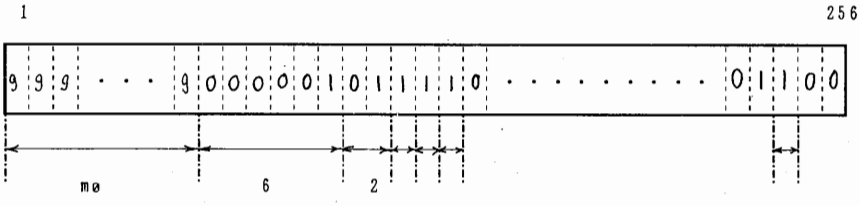


図5　線分点列の x 軸への射影
Fig. 5　Projection of a segment on x-axis.



$$B = \frac{\left(\frac{1}{6}\right)^n+\left(\frac{1}{2}\right)^n+\left(\frac{1}{1}\right)^n+\left(\frac{1}{1}\right)^n+\left(\frac{1}{1}\right)^n+\cdots+\left(\frac{1}{1}\right)^n}{255-m_\emptyset} \qquad (n=3)$$

図6　点列の密度の計算式
Fig. 6　Calculation of pixel density.

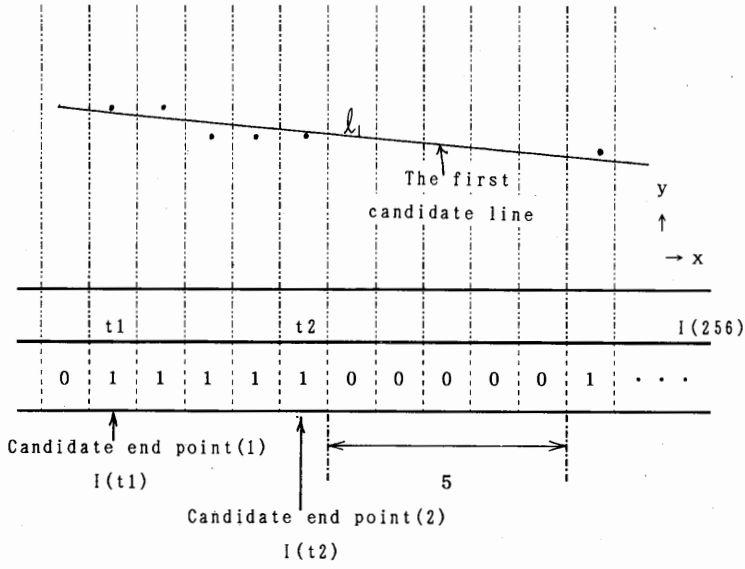


図7　近似範囲の両端点の決定法
Fig. 7　Decision of range end points for approximating the points to a line.

なわち沿線直線 (l_t) に沿う線分の画素列と判断される部分は，'2' を '1' に変更する．またその線分の画素列ではないと判断される部分は，これを '0' に変更する．

（1） 配列 I の '2' の要素列の前後の長さが 4 個以上の '0' の要素列がある場合には，要素 '2' の部分を線分の一部分とみなさない．

（2） 配列 I の '2' の要素列の前後各 2 要素中に要素 '1' が半分以上ある場合には，要素 '2' の部分は線分を構成する要素とみなす．

[Step 3]　先に **5.2** で述べた方法を再度用いて，配列 I から近似範囲の両端点の組を見つける．この操作によって一般には複数の端点の組が得られる．

[Step 4]　端点の組が複数ある場合，それらの中で最も幅の広いものを選択する．これにより図 9 の③の部分のような未検出の線分との交差部を誤って線分と検出することを防止できる．

5.4　検出線分の補正

前節のアルゴリズムによってかなり精度良く線分の端点を検出できる．しかし線分群があまりにも密な場合には，その影響を受けて線分が多少長目に検出される場合もある．図 10 (a)のオーバシュート部がその例で，前節のアルゴリズムではこれを防ぎ得ない．ここではこれを次のような方法で補正した．

すなわち沿線直線を中心軸とし，前節で得た端点を両端にもつ数画素幅のウィンドウを設定する (図 10 (b))．そしてその両端点からウィンドウの内側に向かって端

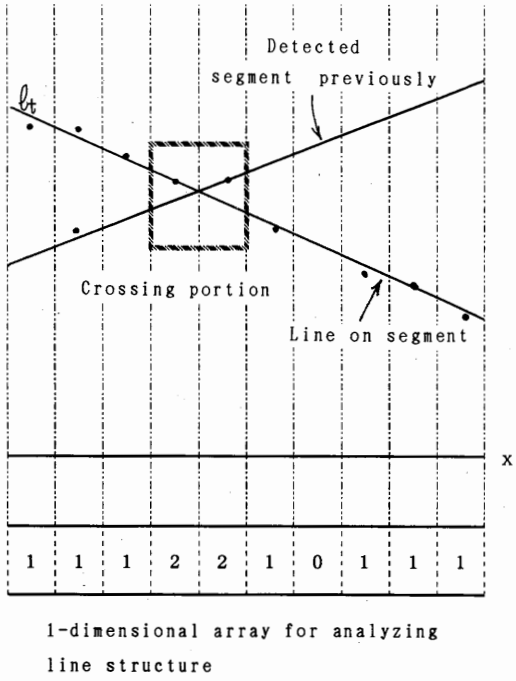


図8　交差部分の x 軸への射影
Fig. 8　Projection of a crossing portion pixels on x-axis.

点間距離の 10 % だけたどり，それぞれ最初に黒画素が現れたところを真の線分の端点の位置と判定する (検出済みの線分の黒画素をもとの数値から変更しておけば，この判定は可能である)．

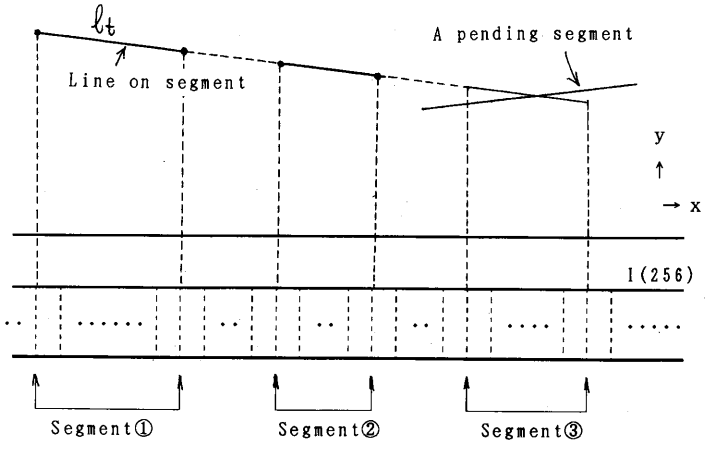
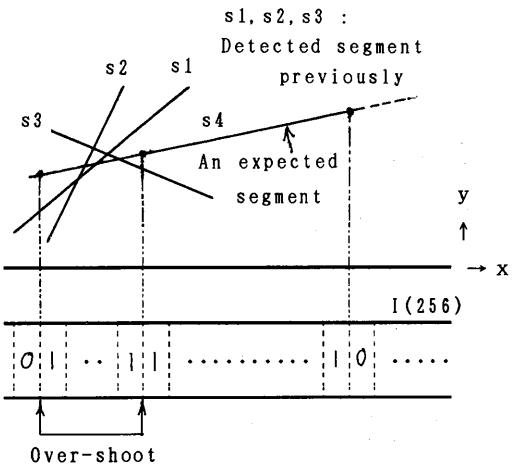
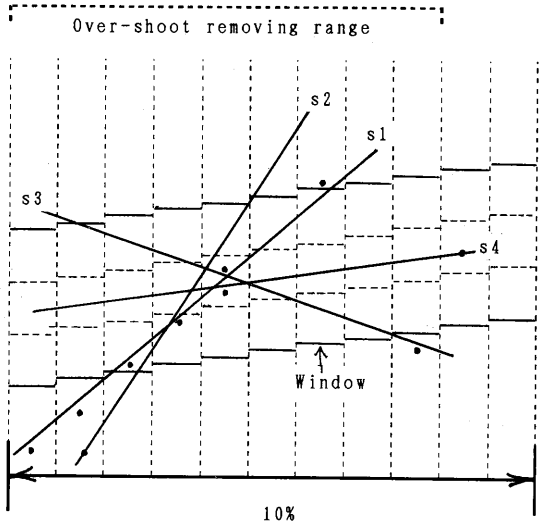


図 9　端点の誤検出例
Fig. 9　An example of fault segment detection.



(a)　Appearance of unexpected over-shoot at one end of expected segment.



(b)　Removing of over-shoot

図 10　オーバシュートの出現とその除去
Fig. 10　Appearance of over-shoot and its removing.

6. 実験結果

表 1 は計算機で発生した 1 本の線分による実験結果例で，10 画素程度の短い線分まで良好に検出できることを示している．なお同表中の誤差は，線分を生成する際の直線の方程式に対する値である．

図 11，12 はそれぞれテストパターンと霧箱写真の 2 値化像の処理結果で，いずれもかなり良好な検出結果が得られている．また 3 画素間隔の平行線（長さ 20 画素）をも確実に分離検出することができた．

7. むすび

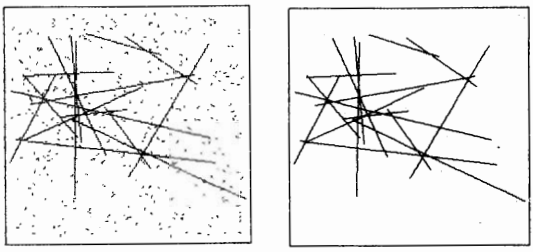
この方法には正負 2 回の Hough 変換を必要とする，度数ピークの検出回数が多くなるなど，処理時間を長くする要因が多い．しかしこれを高速化するための方法もいくつか考えられ，また既に発表されている種々の手法の取入れも可能であって，その実現は今後の課題である．

このアルゴリズムを曲線に適用すると折れ線近似された直線群が得られるので，新しい応用も考えられよう．

表1 短い線分の検出結果（θ の分割間隔は1°，ρ の分割間隔は2画素）

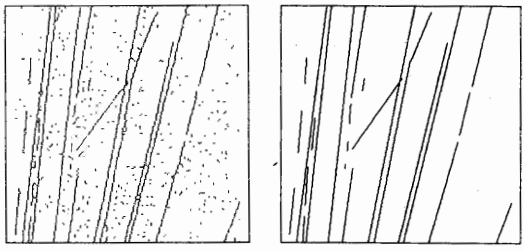
直線の長さ (pixel)	5		10		20	
	θ	ρ	θ	ρ	θ	ρ
検出誤差の平均	0.024	−0.581	−0.142	−0.401	−0.049	−0.118
検出誤差の標準偏差	2.785	6.142	0.859	1.955	0.304	0.718

単位 θ：degree，ρ：pixel



(a) Test pattern (b) Detected segments

図11 テストパターン
Fig. 11 Test pattern generated by Computer.



(a) Digitized original picture (b) Detected segments

図12 霧箱写真
Fig. 12 Cloud chamber photography.

文　献

(1) R. O. Duda and P. E. Hart : "Use of the Hough transformation to detect lines and curves in pictures", Commun. ACM, **15**, 1, pp. 11-15 (Jan. 1972).

(2) 眞水大和："直線パターン検出のための Hough 曲線追跡型アルゴリズムについて"，信学論(D)，**J68-D**，10，pp. 1769-1776 (昭 60-10).

(3) 安居院猛，崔　亨振，中嶋正之："画像処理を用いたナンバープレート領域の抽出に関する研究"，信学論(D)，**J70-D**，3，pp. 560-566 (昭 62-03).

(4) 平子智明，吉田雄二，福村晃夫："市街地図からの道路情報の抽出"，昭 62 信学総全大，1544.

(5) 松山隆司，長尾　真："Hough 変換の幾何学性質と直線群検出への応用"，情処学論，**26**，6，pp. 1069-1078 (昭 60-06).

(昭和 63 年 3 月 22 日受付，8 月 19 日再受付)

大和　淳二

昭 22 東北大・工・電気卒．同年逓信省入省．昭 56 新潟大・工・情報・教授．工博．この間，電子交換方式，画像処理，3 次元情報入出力などの研究に従事．日本 ME 学会，画像電子学会，日本医療情報学会各会員．

稲葉　稔智

昭 61 新潟大・工・情報卒．昭 63 同大大学院修士課程了．同年沖電気工業入社．

石井　郁夫

昭 38 新潟大・工・電気卒．昭 39 新潟大・工・電子助手．昭 42 同講師．昭 46 同助教授．現在同情報助教授．立体画像の表示・入力，音声・画像・図形の符号化などの研究に従事．工博 (東工大)．情報処理学会会員．

牧野　秀夫

昭 51 新潟大・工・電子卒．昭 53 同大大学院修士課程了．昭 54 新潟大・工・情報・助手，現在に至る．この間昭 58 より 1 年間，北大・応電研にて植込み型除細動器等の研究に従事．工博．情報処理学会，日本 ME 学会，日本医療情報学会，日本人工臓器学会，IEEE 各会員．