グラフィクスとＣＡＤ　37−6
（1989. 2. 22）

パラメータ平面を使用しない高精度Ｈｏｕｇｈ変換を用いた近接２直線の識別

Discrimination of close two lines by high precision Hough transform without using a parameter plane

塩野　充
Mitsuru SHIONO

岡山理科大学　工学部電子工学科
Okayama University of Science

あらまし　Ｈｏｕｇｈ変換は画像中に存在する直線等を安定に検出する有力な特徴抽出手法として近年その応用が活発化している。Ｈｏｕｇｈ変換の問題点の１つとしてθ−ρパラメータ平面に必要とするメモリの問題がある。検出しようとする直線の傾斜や間隔の精度を高めるにはθ、ρを細かく計算する必要があり、それだけパラメータ平面を大きく設定しなければならず、大きな配列用のメモリが必要となる。本研究ではＨｏｕｇｈ曲線の全ての交点を代数的に求め、その座標値を指定桁で丸めてから併合する方法で、パラメータ平面を全く使用せずに任意の高精度で直線検出が可能な方法（丸め併合法）を提案し、極めて近接した２本の直線を識別する基礎実験を行った。

ABSTRACT　Hough transform is one of the most stabilized methods for feature extraction from digital images. One of the problems of Hough transform for practical use is that it requires very large memory for θ - ρ parameter plane. If very high precision about θ or ρ is required, the parameter plane grows very large. In this report, a method for Hough transform computation without using a parameter plane is proposed and the result of its experiment is shown. By this method, high precision computation can be executed with so small memory.

１．まえがき

Ｈｏｕｇｈ（ハフ）変換は画像中に存在する直線等を安定に検出する有力な特徴抽出手法として近年その応用が活発化している[1]。ハフ変換を用いれば、細線化追跡法等の局所的方法では検出困難な低品質画像等における断続（途切れ）の多い不安定な直線成分をも安定に検出することができる。さらに、直線だけではなく、円や楕円、その他の一般的な図形を抽出しようとする一般化ハフ変換法の研究も行われている。

ハフ変換の問題点としては２つが考えられる。第１は計算量の多さに起因する処理速度の問題であり、種々の高速化手法が提案されている[2]−[5]。第２はθ−ρパラメータ平面に必要とするメモリの問題である。検出しようとする直線の傾斜や間隔の精度を高めるにはθ、ρを細かく計算する必要があり、パラメータ平面を非常に大きく設定しなければならず、そのための大きな２次元配列用のメモリが必要となる。本研究ではハフ曲線の全ての交点を代数的に求め、その座標値を指定桁で丸めてから併合する方法で、パラメータ平面を全く使用せずに任意の高精度で直線検出が可能な方法（丸め併合法と呼ぶ）を示し、極めて近接した２本の直線の識別実験を行った。

２．ハフ変換

通常のハフ変換による直線の検出方法を以下に示す。画面サイズＸ×Ｙの２値原画像をＦ（ｘ，ｙ）（ｘ＝０，１，２，・・・，Ｘ−１；ｙ＝０，１，２，・・・，Ｙ−１）とし

、その中のN個の黒点（エッジ点）をP_i（x_i，y_i）（$i=1, 2, \cdots, N$）とすると、1個の黒点P_iに関するハフ曲線（$\theta-\rho$パラメータ平面上の、「～」状の形をした曲線のことで、原画面上の1個の黒点がパラメータ平面上の1本のハフ曲線に対応する）は次式で表される。

$$\rho = x_i \cos\theta + y_i \sin\theta \qquad (1)$$

このハフ曲線を2次元配列G（k，ℓ）（$k=0, 1, 2, \cdots, K$；$\ell=0, 1, 2, \cdots, L$）で表された$\theta-\rho$パラメータ平面（$0 \leqq \theta \leqq 2\pi$）に写像する。それには、$\theta$を0から逐次、$\Delta\theta$、$2\Delta\theta$、$3\Delta\theta$、$\cdots$、$2\pi$、と増加させていって、その各々の場合の$\rho$の値を式（1）で計算して、配列G上の対応するG（k，ℓ）の要素値を1だけ増加させる。なお、配列Gの初期値は全て0とする。この場合、検出精度をあげるためにθ軸の刻み$\Delta\theta$を小さくすれば、配列Gのθ方向の要素数Kは大きくなり、大きなメモリが必要となる。

全ての黒点に対してこの計算を行うと、配列G上には全てのハフ曲線が描かれ、複数のハフ曲線が交わっている部分ではその要素値が、交点の度数分布を表していることになる。これより、度数分布の最も大きい交点の（θ，ρ）を求めると、式（1）から検出すべき直線の方程式が得られる。

上記の方法はハフ変換による直線抽出の最も基本的な方法であるが、上述したようにθの検出精度を上げようとして、θの刻み$\Delta\theta$を小さくすると、パラメータ平面を表す配列Gのk方向の要素数Kが多きくなり、それだけ大きなメモリが必要となってしまう。このことはρの検出精度を上げようとする場合でも同じである。

本研究では、θやρの検出精度を上げるために、θやρの刻みをいくら小さくしても大きなメモリを必要としない方法として、パラメータ平面を全く使わない方法を提案し、基礎的な実験を行った。次章以下にその説明を行う。

3．丸め併合法

（3－1）処理の流れ

図1に本方式（丸め併合法）の処理の流れを示すジェネラルフローチャートを示す。

（a）ではまず画面サイズX×Yの画像格納用2次元配列F（0：X－1，0：Y－1）上に逆正接の数表を作成する。この数表に作成される逆正接の値は後述するように、arctan（x／y）である。

（b）では、処理対象となる2値画像をFに入力する。この場合、2値の画素値と（a）で作成した数表とを重畳させて記憶するために後述するような方法を用いている。

（c）では、画面Fを走査し、対象とする全ての黒点（エッジ点）を検出して、そのxy座標値を配列Pに記録する。Pは黒点位置表である。

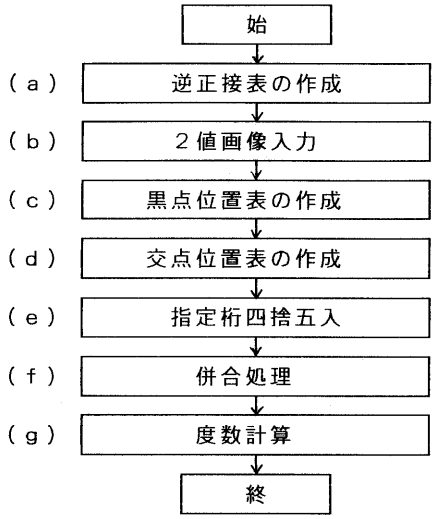

図1　本方式のジェネラルフロー

（d）では任意の2つの黒点に対して、ハフ曲線の交点座標（θ，ρ）を求めて、それを配列Cに記録する。全ての黒点の2個の組み合わせに対して計算を行う。Cは交点位置表である。

（e）では配列Cに格納されたθとρに関して、必要な精度で指定された桁で四捨五入を行う。

（f）では四捨五入した結果、θ、ρとも等しくなった交点を順次併合してゆく。この時、併合回数も記録してゆく。これが度数となる。

（g）では併合回数、すなわち度数の最大のものを抽出して、その時のθとρを求め、抽出すべき直線の方程式を求める。抽出したい直線の数だけこの操作を繰り返す。

（3－2）逆正接表の作成

2値原画像Fの中のN個の黒点（エッジ点）をP_i（x_i，y_i）（$i=1, 2, \cdots, N$）とすると、任意の2個の黒点P_iとP_jに関するハフ曲線は次式で表される。

$$\rho = x_i \cos\theta + y_i \sin\theta \qquad (2)$$

$$\rho = x_j \cos\theta + y_j \sin\theta \qquad (3)$$

式（2）、（3）を解くと、

$$\left.\begin{aligned} \theta &= -\arctan(\Delta x_{ij} / \Delta y_{ij}) \\ \Delta x_{ij} &= x_i - x_j \\ \Delta y_{ij} &= y_i - y_j \end{aligned}\right\} \qquad (4)$$

ρは式（4）を式（2）へ代入すると得られる。画面サイズはX×Yゆえ、

$$\left.\begin{aligned} 0 \leqq |\triangle x_{ij}| \leqq X-1 \\ 0 \leqq |\triangle y_{ij}| \leqq Y-1 \end{aligned}\right\} \quad (5)$$

となる。従って、

$$\arctan\left(|\triangle x_{ij}| / |\triangle y_{ij}|\right)$$

の表はサイズがX×Yとなり、原画像Fと同じサイズである。本方式では時間の掛かるarctanの計算を表参照形式で行う。更にメモリ節減の工夫として、arctanの表を原画像の画面に重畳して格納しておく。即ち、

$$\text{if } F(x,y) = \begin{cases} 0 & \text{then } F(x,y) \leftarrow \arctan(X/Y) \\ 1 & \text{then } F(x,y) \leftarrow \arctan(X/Y) + G \end{cases} \quad (6)$$

とする。ここで、←は代入を意味する。G（＞2π）は適当な正数であり、FがG以上の場合は黒画素であることを表し、又、Gを取り除いたFの値が逆正接表を表している。即ち、

$$\text{if } F(x,y) \geqq G \text{ then } \arctan(X/Y) = F(x,y) - G$$
$$\text{else } \arctan(X/Y) = F(x,y) \quad (7)$$

となる。このようにして、逆正接表のためだけの特別なメモリ領域を使用せずに表参照計算を可能にしている。

（3－3）交点位置表の作成

N本のハフ曲線の交点数は、

$$N_H = {}_N C_2 = N(N-1)/2 \quad (8)$$

となる。交点位置格納用の2次元配列、

$$C(1:3,\ 1:N_H)$$

を用意し、式（3）によって求めたN_H個の交点位置のθをC（1，＊）に、ρをC（2，＊）に格納する。ここで、＊は全ての添字を表す。すなわち配列の断面を表す。C（3，＊）は後述するように度数を格納するための場所である。

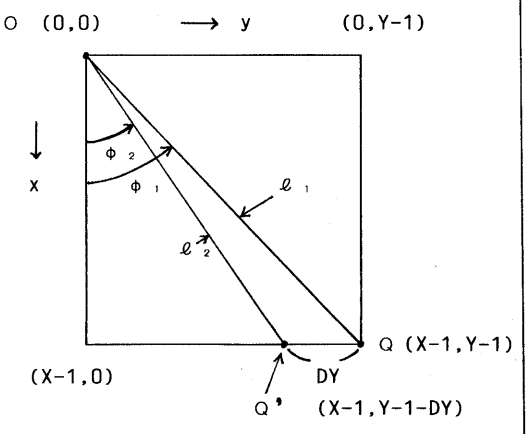

図２　検出対象とする２本の直線

（3－4）丸めと併合

全てのC（1，＊）、C（2，＊）をそれぞれ小数第μ位と小数第ν位で四捨五入する。すなわち、θは小数第μ位で、ρは小数第ν位で四捨五入する。μとνは必要とするθ、ρの精度によって決める四捨五入の桁数で、精度が細かくなる程大きくなる。例えば、C（1，n）が1．234567のとき、μ＝4で四捨五入を行うと、C（1，n）＝1．2346となる。

この後、交点の度数を計算してC（3，＊）へ格納する。即ち、ある2つの交点のθとρがともに一致すれば、その2つの交点は併合可能と見なし、C（3，＊）の値を1だけ増加する。なお、C（3，＊）の初期値は全て1としておく。度数計算のアルゴリズムを以下に示す。∧はａｎｄを表し、→は代入を表す。

$$\left.\begin{aligned} &\text{if} \quad C(1,i) = C(1,j) \\ &\quad \wedge \ C(2,i) = C(2,j) \\ &\quad \wedge \ C(1,i) \neq -999 \\ &(i = 1 \sim N_H - 1,\ j = i+1 \sim N_H) \\ &\text{then} \ C(3,i) \leftarrow C(3,i) + 1 \\ &\qquad C(1,i) \leftarrow -999 \end{aligned}\right\} \quad (9)$$

（3－5）直線の検出

度数のしきい値をτとして、τ以上の度数の直線を検出するとすれば、検出される直線の方程式は次のようになる。

$$\left.\begin{aligned} &\rho_m = x\cos\theta_m + y\sin\theta_m \\ &\quad \text{但し、} \theta_m = C(1,m) \\ &\quad \rho_m = C(2,m) \\ &\quad C(3,m) \geqq \tau \end{aligned}\right\} \quad (10)$$

式（10）は次のようにも書ける。

$$\left.\begin{aligned} &y = -\cot\theta_m \cdot x + \rho_m / \sin\theta_m \\ &\quad (\theta_m \neq 0,\ \pi \text{のとき}) \\ &y = \rho_m / \cos\theta_m \\ &\quad (\theta_m = 0,\ \pi \text{のとき}) \end{aligned}\right\} \quad (11)$$

４．識別実験

（4－1）原画像の作成

画面サイズX×Yの原画像をF（0：X－1，0：Y－1）とし、その対角線ℓ_1と、その対角線に非常に近接した直線ℓ_2とを画面F上にN個の黒点として表し、本方式によってそれら2本の直線を識別して検出する実験を行った。

直線ℓ_1は原点O（0，0）およびQ（X－1，Y－1）を通る直線であり、直線ℓ_2は原点Oと、点Qからy方向にDYだけ変位した点Q'（X－1，Y－1－DY）を通る直線である。この設定を図2に示す。ℓ_1、ℓ_2の方程式は各々次式のようになる。

$\ell_1 : y = \{(Y-1)/(X-1)\}x$ （12）

$\ell_2 : y = \{(Y-1-DY)/(X-1)\}x$ （13）

実際の実験では、X＝Yの正方形としたので、ℓ_1は傾き45°、ℓ_2は傾き45°弱の直線である。変位DYは、最小値の1として、直線ℓ_2が直線ℓ_1に最も近接した状況とした。なお、ℓ_1、ℓ_2とも原点を通るので、y切片はいずれも0である。　N個（偶数）の黒点を画面F上に配置するには乱数を用いた。Nの内の半分は直線ℓ_1上に、あとの半分は直線ℓ_2上に位置するように設定した。即ち、rを区間［0，1］の一様乱数とするとき、黒点P_i（x，y）（i＝1～N／2）は、

$$\left.\begin{aligned} x &= \text{round}\ \{(X-1)\,r\} \\ y &= \text{round}\ [\{(Y-1)/(X-1)\}x] \\ &= x \end{aligned}\right\} \quad (14)$$

として直線ℓ_1上に分布し、黒点P_j（x，y）（j＝N／2＋1～N）は、

$$\left.\begin{aligned} x &= \text{round}\ \{(X-1)\,r\} \\ y &= \text{round}\ [\{(Y-1-DY)/(X-1)\}x] \\ &= \text{round}\ [\{(X-1-DY)/(x-1)\}x] \\ &= \text{round}\ [\{1-DY/(X-1)\}x] \end{aligned}\right\} \quad (15)$$

として直線ℓ_2上に分布するようにした。ここで、roundは小数第1位を四捨五入して整数化する操作を表す。これはディジタル画面が、整数による格子点の表現であるために行う操作であり、この時点で量子化誤差が発生するがやむを得ない。

（4－2）実験結果と検討

実験における画像Fの画面サイズX×Y（X＝Y）は、X＝32、64、128、256、512、1024の6つの場合について行った。又、各々の場合について、黒点数Nを16、32、64、・・・・・と変化させて実験を行った（N≦X）。いずれの場合も直線ℓ_2の変位DYは最小の1として行った。従って、画面サイズが大きくなる程、直線ℓ_2は直線ℓ_1に接近することになる。実験は東京大学大型計算機センターのHITAC／M682を使用し、プログラムはFORTRANで記述した。

6つの場合の識別実験結果を表1～表6に示す。各表の上部に2本の直線ℓ_1、ℓ_2の真の勾配（傾き）を示す。各項の上段はℓ_1、下段はℓ_2である。前述したように、直線の方程式を画面F上でディジタル化する際の量子化誤差があるので、ℓ_1、ℓ_2の勾配が完全に正確には抽出されていないが、2本の直線の識別にはいずれの場合も成功している。y切片ξはいずれの場合も正しく0となっている。度数百分率ψとは、その直線（ハフ曲線の交点）に対して行われた併合回数を全併合回数で除した割合である。ℓ_1とℓ_2のψを加えても100%にならないのはℓ_1、ℓ_2以外にも雑音的な直線が多数検出されるからである。しかし、常にℓ_1は第1位、ℓ_2は第2位に入っている。

表7に逆正接表を使用した場合と使用せずにそのつど逆正接を計算した場合の処理時間の比較を示す。この表では余り大きな差は表われていないが、画面サイズがもっと大きくなって計算量が増加すると逆正接表の効果が表われるものと考える。

表8、表9に通常のハフ変換の計算方法で同じ直線検出を行った場合の結果を示す。表8は画面サイズX＝256の場合、表9はX＝512の場合である。θ軸の刻み数Kは360（1．0°間隔）、720（0．5°間隔）、1800（0．2°間隔）、3600（0．1°間隔）の4つの場合について行った（但し、X＝512の場合にはKを3600にするとパラメータ平面が標準の仮想記憶空間16MBを越えるので実験していない）。ρ軸の刻みは全て1．0間隔とした。X＝256の場合では、表4（本方式）と表8（通常方式）の比較になる。ℓ_2の勾配の真値は44．88744°に対して、本方式では44．79362（誤差は－0．09382）又は44．85092（同－0．03652）であり、通常方式では刻み数Kが360のとき44．49992（同－0．38752）、720のとき44．74991（同－0．13753）と誤差が大きく、Kが1800のとき、44．89995（同－0．01251）、Kが3600のとき、44．84995（同－0．03749）と少なくなっている。しかし、処理時間は本方式が、例えばN＝256のとき463（ミリ秒）であるのに対し、通常方式では刻み数Kが720のときに405（ミリ秒）とコンパラブルであるが、Kが1800では1090（ミリ秒）、3600では2409（ミリ秒）と大変長くなっている。又、原画像Fの格納用以外に必要なメモリとしては、本方式では黒点数Nに依存するが、この場合の最大のN＝256の場合で比較すると、黒点位置表Pとして256×2×4＝2048B、交点位置表CとしてはNH＝32640ゆえ、32640×3×4＝391680B、合計約394KBとなるのに対し、通常方式ではパラメータ平面はρの区間を［$-\sqrt{2}X$，$+\sqrt{2}X$］とすると、ρ軸の刻み数Lは刻み間隔を1として、$2\sqrt{2}X$となるが、このとき、θ軸の刻み数Kが360のときに約1MB、720のときに約2MB、1800のときに約5MB、3600のときに約10MBと大変多くなる。

X＝512の場合の表5と表9の比較に関しても同じようなことがいえる。この場合、表9では勾配に＊印を付けてあ

表１　直線検出結果（Ｘ＝３２）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.06081 （度））

黒点数Ｎ（交点数N_H）	直線	勾配 φ（度）	γ切片 ξ	度数百分率 ψ (%)	併合回数	処理時間（ｍｓ） 表作成	計算時間
１６	ℓ_1	45.02280	0.0	73.03	89	2	2
(120)	ℓ_2	43.24664	0.0	5.62			
３２	ℓ_1	45.02280	0.0	68.31	445	2	6
(496)	ℓ_2	44.13205	0.0	1.80			

表２　直線検出結果（Ｘ＝６４）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.54164 （度））

黒点数Ｎ（交点数N_H）	直線	勾配 φ（度）	γ切片 ξ	度数百分率 ψ (%)	併合回数	処理時間（ｍｓ） 表作成	計算時間
１６	ℓ_1	45.02280	0.0	74.71	87	6	2
(120)	ℓ_2	44.50714	0.0	8.05			
３２	ℓ_1	45.02280	0.0	70.21	433	6	6
(496)	ℓ_2	44.27797	0.0	3.23			
６４	ℓ_1	45.02280	0.0	62.99	1932	6	25
(2016)	ℓ_2	44.33527	0.0	2.59			

表３　直線検出結果（Ｘ＝１２８）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.77354 （度））

黒点数Ｎ（交点数N_H）	直線	勾配 φ（度）	γ切片 ξ	度数百分率 ψ (%)	併合回数	処理時間（ｍｓ） 表作成	計算時間
１６	ℓ_1	45.02280	0.0	68.42	95	26	5
(120)	ℓ_2	44.62175	0.0	13.68			
３２	ℓ_1	45.02280	0.0	69.89	435	26	8
(496)	ℓ_2	44.62175	0.0	6.67			
６４	ℓ_1	45.02280	0.0	63.48	1917	26	27
(2016)	ℓ_2	44.62175	0.0	5.01			
１２８	ℓ_1	45.02280	0.0	59.45	8009	26	113
(8128)	ℓ_2	44.62175	0.0	4.27			

表４　直線検出結果（Ｘ＝２５６）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.88744 （度））

黒点数Ｎ（交点数N_H）	直線	勾配 φ（度）	γ切片 ξ	度数百分率 ψ (%)	併合回数	処理時間（ｍｓ） 表作成	計算時間
１６	ℓ_1	45.02280	0.0	65.66	99	104	26
(120)	ℓ_2	44.79362	0.0	15.15			
３２	ℓ_1	45.02280	0.0	64.65	447	105	29
(496)	ℓ_2	44.85092	0.0	9.84			
６４	ℓ_1	45.02280	0.0	63.72	1910	104	46
(2016)	ℓ_2	44.85092	0.0	9.06			
１２８	ℓ_1	45.02280	0.0	59.68	7978	104	135
(8128)	ℓ_2	44.79362	0.0	7.08			
２５６	ℓ_1	45.02280	0.0	61.90	32471	104	463
(32640)	ℓ_2	44.79362	0.0	6.49			

表５　直線検出結果（Ｘ＝５１２）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.94388 （度））

黒点数Ｎ（交点数N_H）	直線	勾配　φ（度）	γ切片 ξ	度数百分率 ψ (%)	併合回数	処理時間（ｍｓ） 表作成	計算時間
１６	ℓ_1	45.02280	0.0	61.90	105	418	154
(120)	ℓ_2	44.90822	0.0	20.95			
３２	ℓ_1	45.02280	0.0	63.24	457	416	158
(496)	ℓ_2	44.90822	0.0	16.63			
６４	ℓ_1	45.02280	0.0	62.93	1934	426	175
(2016)	ℓ_2	44.90822	0.0	16.86			
１２８	ℓ_1	45.02280	0.0	59.80	7962	421	256
(8128)	ℓ_2	44.90822	0.0	14.18			
２５６	ℓ_1	45.02280	0.0	64.83	32420	416	571
(32640)	ℓ_2	44.90822	0.0	12.85			
５１２	ℓ_1	45.02280	0.0	59.44	130556	418	2223
(130186)	ℓ_2	44.90822	0.0	13.49			

表６　直線検出結果（Ｘ＝１０２４）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.97198 （度））

黒点数Ｎ（交点数N_H）	直線	勾配　φ（度）	γ切片 ξ	度数百分率 ψ (%)	併合回数	処理時間（ｍｓ） 表作成	計算時間
１６	ℓ_1	45.02280	0.0	60.19	108	1731	796
(120)	ℓ_2	44.96552	0.0	25.93			
３２	ℓ_1	45.02280	0.0	62.15	465	1727	807
(496)	ℓ_2	44.96552	0.0	21.29			
６４	ℓ_1	45.02280	0.0	62.47	1948	1716	804
(2016)	ℓ_2	44.96552	0.0	18.89			
１２８	ℓ_1	45.02280	0.0	59.72	7972	1753	888
(8128)	ℓ_2	44.96552	0.0	16.12			
２５６	ℓ_1	45.02280	0.0	64.90	32384	1742	1208
(32640)	ℓ_2	44.96552	0.0	14.63			
５１２	ℓ_1	45.02280	0.0	61.45	130456	1732	3000
(130186)	ℓ_2	44.96552	0.0	16.23			
１０２４	ℓ_1	45.02280	0.0	59.41	523338	1737	11203
(523776)	ℓ_2	44.96552	0.0	15.29			

表７　逆正接表を使用した場合と、使用しなかった場合の処理時間（単位ミリ秒）の比較（Ｘ＝２５６）

黒点数Ｎ	使用した場合 表作成時間	使用した場合 計算時間	使用しない場合
１６	１０４	２６	２６
３２	１０５	２９	２９
６４	１０４	４６	４８
１２８	１０４	１３５	１４４
２５６	１０４	４６３	５０６

表８　通常のハフ変換計算法による直線検出結果（Ｘ＝２５６）
（真の勾配値　ℓ_1 :45.00000（度）；ℓ_2 :44.88744（度））（時間はミリ秒）

θ軸の刻み数＝３６０の場合					θ軸の刻み数＝７２０の場合				
黒点数 N	直線	勾配 φ（度）	度数	計算時間	黒点数 N	直線	勾配 φ（度）	度数	計算時間
１６	ℓ_1	44.99989	11	147	１６	ℓ_1	44.99989	11	263
	ℓ_2	44.49992	8			ℓ_2	44.74991	9	
３２	ℓ_1	44.99989	23	152	３２	ℓ_1	44.99989	23	271
	ℓ_2	44.49992	11			ℓ_2	44.74991	16	
６４	ℓ_1	44.99989	47	160	６４	ℓ_1	44.99989	47	289
	ℓ_2	44.49992	34			ℓ_2	44.74991	37	
１２８	ℓ_1	44.99989	91	180	１２８	ℓ_1	44.99989	91	328
	ℓ_2	44.49992	50			ℓ_2	44.74991	73	
２５６	ℓ_1	44.99989	190	213	２５６	ℓ_1	44.99989	190	405
	ℓ_2	44.49992	111			ℓ_2	44.74991	125	
θ軸の刻み数＝１８００の場合					θ軸の刻み数＝３６００の場合				
黒点数 N	直線	勾配 φ（度）	度数	計算時間	黒点数 N	直線	勾配 φ（度）	度数	計算時間
１６	ℓ_1	44.99994	11	602	１６	ℓ_1	44.94995	16	1231
	ℓ_2	44.89995	14			ℓ_2	44.84995	14	
３２	ℓ_1	44.99994	23	641	３２	ℓ_1	44.94995	32	1312
	ℓ_2	44.89995	26			ℓ_2	44.84995	26	
６４	ℓ_1	44.99994	47	710	６４	ℓ_1	44.94995	63	1475
	ℓ_2	44.89995	55			ℓ_2	44.84995	55	
１２８	ℓ_1	44.99994	91	842	１２８	ℓ_1	44.94995	125	1779
	ℓ_2	44.89995	111			ℓ_2	44.84995	109	
２５６	ℓ_1	44.99994	190	1090	２５６	ℓ_1	44.94995	241	2409
	ℓ_2	44.89995	221			ℓ_2	44.84995	220	

る項目は直線検出に失敗した場合であり、Ｘ＝５１２の場合、刻み数Ｋが３６０や７２０では失敗が発生している。この場合、必要メモリは、刻み数Ｋが３６００の場合には約２０．８５ＭＢとなり、大型計算機の標準の仮想記憶空間の１６ＭＢを越えるのに対して、本方式ではＮ＝５１２のときに約１．６ＭＢであり、パソコンでも実行可能なメモリの大きさである。

５．まとめ

パラメータ平面を用いない丸め併合法という方法で、極めて近接した２本の直線を識別できる高精度なハフ変換の計算法を提案し、基礎的な実験を行った。その結果、ディジタル画面上で最小限である１画素の差で近接させた２本の直線を、画面サイズが３２×３２から１０２４×１０２４のいずれの場合でも正しく検出することができた。又、従来の通常の計算方法との比較実験も行った。その結果、精度、処理時間、使用メモリともに本方式の方に良い結果が得られた。特に必要メモリには大きな差が表われ、従来の方法では計算が困難な巨大なパラメータ平面を必要とする場合でも本方式ではわずかなメモリで計算を行うことが出来ることが分かった。今後は本方式を、直線以外の検出を目的とする一般化ハフ変換に応用する方法を検討したいと考える。

文　献

（１）奥水大和：”Ｈｏｕｇｈ変換に関する最近の研究動向”，情処学研資，ＣＶ５１－１，ｐｐ．１－８，１９８７．
（２）奥水大和：”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．１７６９－１７７６，１９８５．
（３）恩田邦夫，青木由直：”三角関数の周期性を利用したＨｏｕｇｈ変換の高速計算法”，信学論Ｄ，Ｊ７０－Ｄ，１０，ｐｐ．２００９－２０１１，１９８７．
（４）安居院猛，崔亨振，中嶋正之：”ピラミッド階層を利用した高速ハフ変換について”，信学技報ＩＥ８６－６７，１９８６．
（５）花原啓至，丸山次人，内山隆：”実時間Ｈｏｕｇｈ変換プロセッサ”，昭和６０年度信学情報・システム部門全大，９２，１９８５．

（表９は次ページ ㊧）

表９　通常のハフ変換計算法による直線検出結果（Ｘ＝５１２）
（真の勾配値　ℓ_1 :45.00000 （度）；ℓ_2 :44.94388 （度））（時間はミリ秒）

θ軸の刻み数＝３６０の場合

黒点数 N	直線	勾配 φ（度）	度数	計算時間
１６	ℓ_1	44.99989	11	389
	ℓ_2	44.49992	5	
３２	ℓ_1	44.99989	23	397
	ℓ_2	*44.99989	9	
６４	ℓ_1	44.99989	47	409
	ℓ_2	*44.99989	17	
１２８	ℓ_1	44.99989	91	427
	ℓ_2	*44.99989	37	
２５６	ℓ_1	44.99989	197	466
	ℓ_2	*44.99989	59	
５１２	ℓ_1	44.99989	367	540
	ℓ_2	*44.99989	145	

θ軸の刻み数＝７２０の場合

黒点数 N	直線	勾配 φ（度）	度数	計算時間
１６	ℓ_1	44.99989	11	625
	ℓ_2	44.74991	8	
３２	ℓ_1	44.99989	23	644
	ℓ_2	*45.24986	11	
６４	ℓ_1	44.99989	47	675
	ℓ_2	44.74991	23	
１２８	ℓ_1	44.99989	91	736
	ℓ_2	44.74991	49	
２５６	ℓ_1	44.99989	197	811
	ℓ_2	44.74991	103	
５１２	ℓ_1	44.99989	367	993
	ℓ_2	44.74991	218	

θ軸の刻み数＝１８００の場合

黒点数 N	直線	勾配 φ（度）	度数	計算時間
１６	ℓ_1	44.99994	11	1342
	ℓ_2	44.79996	9	
３２	ℓ_1	44.99994	23	1382
	ℓ_2	44.89995	16	
６４	ℓ_1	44.99994	47	1453
	ℓ_2	44.89995	37	
１２８	ℓ_1	44.99994	91	1622
	ℓ_2	44.89995	73	
２５６	ℓ_1	44.99994	197	1950
	ℓ_2	44.89995	140	
５１２	ℓ_1	44.99994	367	2527
	ℓ_2	44.89995	294	