社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2009－SLDM－139　(14)
2009－ＥＭＢ－ 12　(14)
2009／3／6

投票無衝突化手法を用いた小面積画素並列ハフ変換回路

後藤　正之†　　中村　一博†　　高木　一義†　　高木　直史†

† 名古屋大学大学院情報科学研究科
〒 464-8603 愛知県名古屋市千種区不老町
E-mail: †{gotoh,nakamura,ktakagi,ntakagi}@takagi.i.is.nagoya-u.ac.jp

あらまし　コンピュータビジョンアルゴリズムの中で画像から直線の検出を行うハフ変換は，携帯機器や車載機器などの組込システムに用いられている．ハフ変換は処理する画像サイズが大きくなるにつれリアルタイムで処理することが難しくなる．そこで並列化による計算高速化の研究が行われてきている．本報告では画素について並列に計算するハフ変換回路において投票を無衝突化する手法を提案する．画素について並列に計算する際に発生する投票の衝突を解消する手法は，画像探索方向と丸め精度の工夫により投票の無衝突化を達成する．本手法に基づく画素並列ハフ変換回路は投票無衝突であり，本手法と以前提案した投票空間アクセス局所化を用いた画素並列ハフ変換回路は省メモリ，小面積で，投票の衝突を解消するための回路も不要である．提案手法と投票空間アクセス局所化手法を用いた画素並列ハフ変換回路について評価したところ，画像サイズを VGA，並列数を 16 としたとき，回路内に必要なメモリ量が θ 並列ハフ変換回路の回路内メモリ量の約 1 %，必要クロックサイクル数はほぼ同じであることが確認できた．
キーワード　ハフ変換, 特定用途向けアーキテクチャ, 画像処理, 投票無衝突, 並列ハフ変換回路

Memory Efficient Pixel-parallel Hough Transform Circuit Using Collision-free Voting Memory Access

Masayuki GOTO†, Kazuhiro NAKAMURA†, Kazuyoshi TAKAGI†, and Naofumi TAKAGI†

† Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
E-mail: †{gotoh,nakamura,ktakagi,ntakagi}@takagi.i.is.nagoya-u.ac.jp

Abstract　The Hough transform in computer vision algorithms is a method which detects lines from an image and is used in embedded systems such as cellular phones and vehicles because of the robustness. The more the number of pixel, the more difficult it is to process Hough transform in real-time. Thus, researchs on hardware acceleration which parallelize voting operations on the angular parameter θ have been done. In this report, we propose a method for eliminating vote collisions. Collision-free voting memory access is accomplished by selecting the direction of parallelism and changing accuracy of rounding. The voting memory accesses are collision-free. A pixel-parallel Hough transform circuit using collision-free and localized voting memory access is more memory efficient than a θ-parallel circuit and a pixel-parallel circuit, and collision-free. We have evaluated the circuit by comparing clock cycles and amount of memory in the circuit of the circuit and conventional circuits. We compared 16-parallel circuits for VGA image size, and the result shows that clock cycles are almost the same and the circuit has approximately 99% less amount of memory than a θ-parallel circuit.
Key words　Hough transform, ASIC, image processing, collision-free voting, parallel Hough transform circuit

1. はじめに

近年，コンピュータビジョンアルゴリズムの適用範囲は，ロボットだけでなく携帯機器や車載機器等に拡大しており，外乱にロバストな処理がリアルタイムで行われることが求められている．コンピュータビジョンアルゴリズムの中で，画像から直線の検出を行うハフ変換 [1], [2] は，入力画像のノイズや欠落による検出結果への影響が少なく，安定した検出を行うことができるという特徴を持つため，道路標識認識や車線認識，虹彩認証などの生体認証等に用いる研究が行われている．

一般化ハフ変換 [2] は，x-y 座標の画像平面から，直線を定める傾き θ と原点からの距離 ρ をパラメータとする，ハフ空間への変換である．画素値が 1 である画素を通りうる直線をパラメータで表現する．このパラメータ ρ と θ を用いた表現式を投票式と呼び，投票式を満たす全てのパラメータの組に対して投票する．画像サイズを $W \times H$ ピクセル，角度分解能を Θ とすると，投票処理を $W \times H \times \Theta$ 回行う必要があり，画像サイズが大きくなるにつれリアルタイムで処理することが難しくなる．そこで並列化による計算高速化の研究が行われてきている [3]〜[5], [**?**]〜[6].

計算の高速化のための並列化手法として，画素並列化とパラメータ θ 並列化が挙げられれ [5]，パラメータ θ について並列化し，専用回路化する研究が行われている [3], [4]．高速化するために並列数を大きくすると必要な演算器数，プロセッシングエレメント (PE) 数，回路内メモリ量の増加を招き，回路面積の増加につながる．また，画素並列のハフ変換では投票の衝突が発生する．投票の衝突とは複数の PE が同じアドレスの投票メモリに同時に投票することである．これは投票用のメモリに同時書き込み可能メモリを用いるか，専用回路化する場合，投票衝突解消回路を投票メモリの前に挿入するかしなければならない．画素並列の一般化ハフ変換はパラメータ θ 並列の一般化ハフ変換に比べ，回路内に必要な投票用メモリが少ないため，回路面積や消費電力の点で低コストであるという利点がある．

本報告では画素について並列に計算するハフ変換回路において投票を無衝突化する手法を提案する．θ の値を固定し，画素について並列に計算する際に発生する投票の衝突を解消する本手法は，θ の値による画像の探索方向の選択と丸め精度の 0.5 距離精度化により投票の無衝突化を達成する．提案手法を用いた回路は画素並列でありながら投票無衝突であるため投票衝突解消回路を必要としない．そのため画素並列ハフ変換の特長である小さい投票メモリの回路を構成することが可能である．

投票無衝突化手法と，以前提案した投票空間アクセス局所化手法 [7] を用いて回路を構成し，評価した．結果として従来の θ 並列，画素並列両方のハフ変換回路に比べ投票メモリを大きく削減することができる．提案手法と投票空間アクセス局所化手法を用いた回路を，必要クロックサイクル数と回路内メモリのサイズについて評価し，従来の並列ハフ変換回路と比較を行った．その結果，画像サイズを VGA，並列数を 16 としたとき，回路内に必要なメモリ量をパラメータ θ 並列ハフ変換回路に比べ約 99 %削減できることが分かった．必要クロックサイクル数の増加は約 0.31 %に抑えられることがわかった．約 0.31 %のクロックサイクル数の増加はリアルタイム実行に必要な回路駆動周波数を大きく上げるものではなく，並列化の本来の目的である計算高速化を妨げるものではない．

以降，2. では，ハフ変換とハフ変換専用回路，以前提案した投票空間アクセス局所化手法について述べる．3. では，画素並列ハフ変換の投票無衝突化を提案する．4. では，投票無衝突化手法と投票空間アクセス局所化手法を用いた回路構成の必要クロックサイクル数と回路内メモリ量とレジスタ量を見積もり，評価を行う．最後に 5. でまとめを述べる．

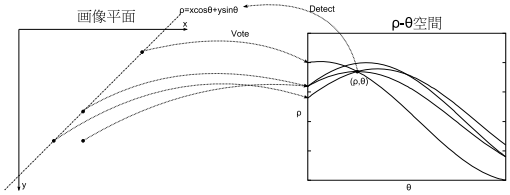

図 1　ρ-θ 投票空間を用いた一般化ハフ変換

2. 準　　備

2.1 ハフ変換

ハフ変換は入力 2 値画像から直線を検出する手法である [1], [2]．ハフ変換は投票処理と多数決処理によって直線を検出する．

直線の検出のための投票処理では，画像平面上の画素値が 1 である画素について処理を行う．各画素について，その画素を通る，切片と傾きをパラメータとしたすべての直線に投票を行う．ここで，画素を通る直線はパラメータによって一意に表現される．投票が終了した後に投票値を多数決処理することで直線を検出する．投票処理によって検出することからハフ変換はノイズなど外乱の影響が少なく，ロバストである．

本報告では計算機上での実行が容易な一般化ハフ変換 [2] を扱う．文献 [2] では，画像平面上の直線を，x-y 座標系における原点から直線に下ろした垂線と x 軸の成す角 θ と，垂線の符号付き長さ ρ を用い，式 (1) のように表現し，投票空間として ρ-θ 投票空間を用いる．

$$\rho = x\cos\theta + y\sin\theta, (0^\circ \leqq \theta < 180^\circ) \tag{1}$$

式 (1) により，直線を表わすパラメータ (ρ, θ) の組は一意に決まる．さらに画像平面が有限である場合，ρ-θ 投票空間も量子化することで有限となる．このため計算機上で取り扱うことが容易であり，計算機上で ρ-θ 空間を用いた一般化ハフ変換を行うときは，ρ, θ をそれぞれ整数値に量子化し，2 次元の累算値配列 $Vote(\rho, \theta)$ で投票空間を実現する (図 1)．パラメータ ρ, θ の量子化精度は一般に整数精度である．画像はピクセル単位で離散的に表現されている．そのため整数精度は，検出された直線を認識する際に適した精度である．また，精度が高すぎると正しい投票先に投票が集中せず，投票先がばらけてしまいノイズによる投票の影響が大きくなってしまう．精度が低すぎると，投票先が近いが，異なる 2 本の直線を誤って 1 本の直線と誤検出してしまう可能性が高くなる．

ρ-θ 投票空間を用いた一般化ハフ変換のフローチャートを図 2 に示す．ここで，W は画像平面の幅，H は画像平面の高さである．また，$G(x, y)$ は入力画像上の座標 (x, y) の画素値である．LoopA,B,C はそれぞれ θ, x, y に関するループである．投票処理は，各画素 (x_i, y_i) に対して，その画素の画素値が 1 の場合のみ行われる．θ の値を 1° ずつ変えながら式 (1) の ρ を計算し，累算値配列の要素 $Vote(\rho, \theta)$ をインクリメントする．

ハフ変換は計算量が多く，ソフトウェアではリアルタイム処

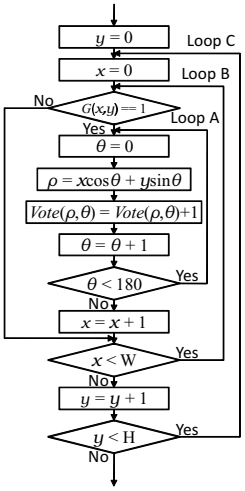

図 2　一般化ハフ変換のフローチャート

理を行うことが困難である．そのため専用回路として実装することで高速化を実現する研究が行われている [3], [4]．専用回路として実装する際に，計算を並列に行うことで高速化を達成している．専用回路実現向きの並列処理は画素についてと，パラメータ θ についての 2 種類である [5].

2.2　パラメータ θ について並列化した回路

式 (1) の x, y の値を固定し，パラメータ θ について並列処理を行う並列化について述べる．図 2 のハフ変換のフローチャート中の LoopA の処理を n 並列に実行する並列ハフ変換回路として，並列数 n を $n = 180$ とした回路が提案されている [3]．n 個の PE が $x\cos\theta + y\sin\theta, x\cos(\theta+1) + y\sin(\theta+1), \cdots, x\cos(\theta+n-1) + y\sin(\theta+n-1)$ を計算する．

パラメータ θ についての並列化の最大の利点は，投票の衝突が起こらないことである．各 PE で計算された ρ の値をインデックスとして投票回路に投票する．投票先はそれぞれ $Vote(\rho_0, \theta_0)$, $Vote(\rho_1, \theta_1), \cdots, Vote(\rho_{n-1}, \theta_{n-1})$ であるため投票の衝突が起こることはないものの，回路内に必要な投票回路のレジスタ量は並列数に対し線形に増加する．

2.3　画素について並列化した回路

式 (1) 中の θ の値を固定し画像平面の画素についての計算の並列化を行い，図 2 の LoopB の処理を m 並列に実行する並列ハフ変換回路について述べる．m 個の PE により $x\cos\theta + y\sin\theta, (x+1)\cos\theta + y\sin\theta, \cdots, (x+m-1)\cos\theta + y\sin\theta$ が計算される．

画素についての並列化の最大の利点は，回路内の投票回路の量を大幅に減らすことができる点である．各 PE で計算された ρ の値をインデックスとして投票回路に投票が行われる．投票先はそれぞれ $Vote(\rho_0, \theta), Vote(\rho_1, \theta), \cdots, Vote(\rho_{m-1}, \theta)$ であるため ρ が同じ値であったとき，投票の衝突が発生する．そのため衝突解消回路で投票の衝突を解消する必要がある．

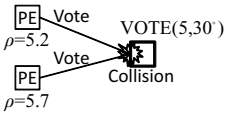

図 3　投票の衝突の例

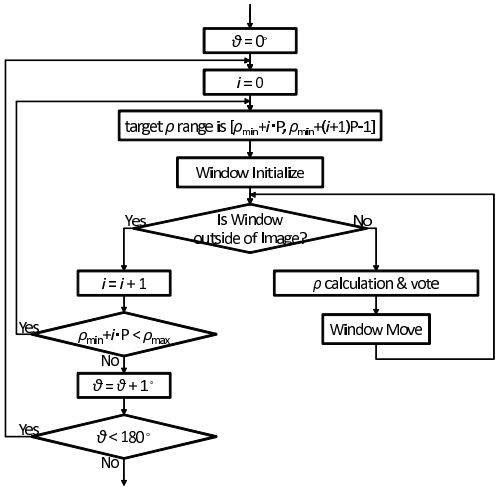

図 4　局所窓単位の画像探索順を用いたハフ変換のフローチャート

2.3.1　投票の衝突

一般にハフ変換はパラメータ ρ と θ の値を量子化して投票を行う．そのため，θ の値を固定し，複数の画素について ρ の計算をする画素並列ハフ変換では，複数の PE が同時に同じインデックス (ρ, θ) のメモリセルに投票することがあり，その際に投票の衝突が起こる．図 3 に投票の衝突の一例を示す．

投票の衝突が発生すると PE 間で書き込みの競合が発生する．投票を正しく行うためにロックを用いると実行時間が長くなるため，処理量が多いハフ変換には向いていない．投票メモリを同時書き込み可能なメモリで実装すること [6] や，同時投票を解消するための回路を付加することは一般にコストが大きくなってしまうため，その他の解決法が必要である．

2.4　投票空間アクセス局所化 [7]

一般的な画像処理アルゴリズムで用いられる画像探索順を用いたハフ変換においては，ρ-θ 投票空間へのアクセスは同じ行の画素について計算しているときは連続的に近いアクセスであるが，行が変わる際には不連続なアクセスになってしまう．この ρ-θ 投票空間の離れた領域への不連続なアクセスをなくし，アクセスを局所化することができれば，回路内に必要な累算値配列を小さくすることができる．局所窓単位の画像探索順を用いたハフ変換のフローチャートを図 4 に示す．図中の P は ρ の値域を分割するための分割単位である．

このハフ変換では $\theta = 0°$ から $180°$ までの局所窓単位の ρ 計算と投票 (ρ calculation & vote) が行われる．局所窓は，ρ 計

算の結果が着目している ρ の値域の分割範囲 (target ρ range) に入るように，θ の値に応じて移動 (Window Move) される．

図 4 の Window Initialize では局所窓の位置の初期化を，“Is Window outside of Image?” では局所窓が画像の外に出ているか否かのチェックを行っている．局所窓が画像内にあれば局所窓内の画素について ρ を計算し，得られた ρ をインデックスとして投票を行う．局所窓内のすべての画素の投票が終わったら Window Move を行う．Window Move は局所窓の移動であり，着目している範囲内の ρ の値を持つ画素を含む局所窓に移動する．局所窓が画像の外に出ている場合，着目している範囲内の ρ の値について投票が確定したため，次の ρ の範囲に着目する．

3. 画素並列ハフ変換における投票の無衝突化

2.3 で述べたように画素ついて並列化するためには，投票の衝突解消回路を用いて投票の衝突を解消する必要がある．このような回路は複雑になりやすく，回路規模を大きくする要因となりうる．各 PE 毎に局所メモリを用い，画像を探索し終えてから局所メモリの内容を足し合わせることで投票の衝突がない画素並列ハフ変換を達成する構成も考えられるが，回路内に必要なメモリ量が少ないという画素並列ハフ変換の利点が損なわれてしまう．

そこで，投票衝突解消回路や PE 毎の局所メモリを用ない，投票無衝突な画素並列ハフ変換を提案する．投票の衝突がない画素並列ハフ変換は，パラメータ ρ の丸め精度を細かくし 0.5 距離精度で投票メモリを確保することと，パラメータ θ の値により画像の探索方向を選択することで達成する．

3.1 においてパラメータ ρ の丸め精度の変更について述べ，3.2 においてパラメータ θ の値により画像探索方向を選択する手法を述べる．

3.1 パラメータ ρ の丸め精度の 0.5 距離精度化

画素 (x, y) の ρ の値を $\rho_{(x,y)}$ とすると式 (1) により

$$\rho_{(x,y)} = x\cos\theta + y\sin\theta$$

と表すことができる．ここで，画素 (x, y) に隣接する画素 $(x \pm 1, y)$ の ρ の値は，

$$\rho_{(x\pm1,y)} = (x \pm 1)\cos\theta + y\sin\theta$$

となる．画素 (x, y) と $(x \pm 1, y)$ の ρ の値の差の絶対値を $\rho_{diff(x\pm1,y)}$ とすると，$\rho_{diff(x\pm1,y)}$ は

$$\rho_{diff(x\pm1,y)} = |\rho_{(x,y)} - \rho_{(x\pm1,y)}| = |\cos\theta| \qquad (2)$$

となる．θ の取りうる範囲は $0^\circ \leqq \theta < 180^\circ$ であることから，式 (2) より $\rho_{diff(x\pm1,y)}$ の取りうる範囲は $0 \leqq \rho_{diff(x\pm1,y)} \leqq 1$ である．

一般にハフ変換では投票の際に ρ の値を整数値に丸めることと，$0 \leqq \rho_{diff(x\pm1,y)} \leqq 1$ であることから，隣接画素において丸め後の ρ の値が等しくなってしまうことがある．そのため画素並列の投票では衝突が発生していた．

そこで，投票の衝突を解消するために丸め精度を一般的な整数距離精度から 2 倍の精度を持つ 0.5 距離精度へと変更する．この変更により投票空間 $\rho \times \theta$ は $2\rho \times \theta$ となる．これにより $0.5 \leqq \rho_{diff} \leqq 1$ となる場合は投票の衝突を解消することができる．しかし，$0 \leqq \rho_{diff} < 0.5$ である場合は依然として投票の衝突が発生してしまう．次節で $0 \leqq \rho_{diff} < 0.5$ とならないようにする手法を述べる．

3.2 パラメータ θ の値による画像の探索方向の選択

3.1 で述べたように，隣接画素間の ρ の値の差 ρ_{diff} が 0.5 未満であるとき，0.5 距離精度の投票空間を用意しても投票の衝突を解消することはできなかった．そこで投票を無衝突化するため，パラメータ θ の値によって画像の探索方向を変更する．

一般的な画像処理における画像探索方向は行方向であるが，行方向に探索していくと θ の値によっては近接画素との ρ の差 ρ_{diff} が $0 \leqq \rho_{diff(x\pm1,y\pm1)} < 0.5$ となる場合が発生してしまう．

ここで，画素 (x, y) とその 4 近傍画素との ρ_{diff} の値を式で表すとと式 (3) のようになる．

$$\rho_{diff(x\pm1,y\pm1)} = \begin{cases} \rho_{diff(x\pm1,y)} = |\cos\theta| & (\rho_{(x,y)} - \rho_{(x\pm1,y)}) \\ \rho_{diff(x,y\pm1)} = |\sin\theta| & (\rho_{(x,y)} - \rho_{(x,y\pm1)}) \end{cases} \qquad (3)$$

式 (3) より，ρ_{diff} の値は画像探索方向により $|\cos\theta|$ と $|\sin\theta|$ の 2 通りあることがわかる．これらの大小を式で表すとと式 (4) のようになる．

$$\begin{cases} |\cos\theta| \leqq \frac{1}{\sqrt{2}} \leqq |\sin\theta| \leqq 1 & (45^\circ \leqq \theta < 135^\circ) \\ |\sin\theta| \leqq \frac{1}{\sqrt{2}} \leqq |\cos\theta| \leqq 1 & otherwize \end{cases} \qquad (4)$$

また，$0.5 < \frac{1}{\sqrt{2}}$ であることから以下の式 (5) が得られる．

$$\begin{cases} 0.5 \leqq |\sin\theta| < 1 & (45^\circ \leqq \theta < 135^\circ) \\ 0.5 \leqq |\cos\theta| < 1 & otherwize \end{cases} \qquad (5)$$

式 (5) より，$0^\circ \leqq \theta < 45^\circ, 135^\circ \leqq \theta < 180^\circ$ のときには $\rho_{diff(x\pm1,y)} = |\cos\theta|$，すなわち行方向の探索順にし，$45^\circ \leqq \theta < 135^\circ$ のときには $\rho_{diff(x,y\pm1)} = |\sin\theta|$，すなわち列方向の探索順にすることで，$0.5 < \rho_{diff(x\pm1,y\pm1)}$ を達成することができることがわかる．

注目画素の 4 近傍画素について投票無衝突化が達成されたが，8 近傍画素については投票に衝突が起こる可能性がある．

PE_i が画素 (x, y) について計算しているとき PE_{i+1} が 8 近傍画素 $(x + 1, y + 1)$ を計算すると，画素 (x, y) と 8 近傍画素 $(x + 1, y + 1)$ の ρ の値の差は $|\cos\theta + \sin\theta|$ となる．この値は θ の値によっては $|\cos\theta + \sin\theta| > 0.5$ とならない．そのため，提案する 0.5 距離精度を用いた画素並列ハフ変換は全ての PE が同じ行 (列) について計算している必要がある．

提案する m 個の PE を用いた画素並列ハフ変換のアルゴリズムを Algorithm1 に示す．$W \times H$ サイズの画像 IMG$[H][W]$ を入力とし，投票結果配列 VOTE$[\theta][\rho]$ を出力とする．1 行目では，投票結果配列 VOTE と 0.5 距離精度のための投票配列 VOTE’ を初期化する．$0^\circ \leqq \theta < 180^\circ$ について繰返し計算を行う際に，$45^\circ \leqq \theta < 135^\circ$ の場合は列方向の探索を行い (5 行目から 17 行目)，$0^\circ \leqq \theta < 45^\circ, 135^\circ \leqq \theta < 180^\circ$ の場合は行方向の探索を行う (19 行目から 31 行目)．

0.5 距離精度での投票を行うため，ρ の値を 2 倍し，ρ' としている (9 行目と 23 行目)．画素値が 1 の場合，ρ' の値をインデックスとして投票配列 VOTE' に投票を行う (11 行目と 25 行目)．ρ の計算と投票はそれぞれ m 並列で行われる．画像を探索し終えたら投票結果を 0.5 距離精度から整数精度に変換し，投票結果配列 VOTE に格納する (34 行目)．

Algorithm 1 投票無衝突な画素並列ハフ変換

Input: IMG$[H][W]$
Output: VOTE$[\theta][\rho]$

1: VOTE := {0}; VOTE'$[\theta][2\rho]$:= {0};
2: **for** $\theta := 0°$ **to** $179°$ **do**
3: $x := 0; y := 0;$
4: **if** $45° \leqq \theta < 135°$ **then**/* 列方向探索*/
5: **while** $x < W$ **do**
6: $y := 0;$
7: **while** $y < H$ **do**
8: **for** $i = 0$ **to** $m - 1$ **para do**
9: $\rho := x\cos\theta + (y + i)\sin\theta;\ \rho' := 2\rho;$
10: **if** IMG$[y + i][x] = 1$ **then**
11: VOTE'$[\theta][\rho']$:= VOTE'$[\theta][\rho']$ + 1;
12: **end if**
13: **end for**
14: $y := y + m;$
15: **end while**
16: $x := x + 1;$
17: **end while**
18: **else**/* 行方向探索*/
19: **while** $y < H$ **do**
20: $x := 0;$
21: **while** $x < W$ **do**
22: **for** $i = 0$ **to** $m - 1$ **para do**
23: $\rho := (x + i)\cos\theta + y\sin\theta;\ \rho' := 2\rho;$
24: **if** IMG$[y][x + i] = 1$ **then**
25: VOTE'$[\theta][\rho']$:= VOTE'$[\theta][\rho']$ + 1;
26: **end if**
27: **end for**
28: $x := x + m;$
29: **end while**
30: $y := y + 1;$
31: **end while**
32: **end if**
33: **for all** $i \in [\rho_{min}, \rho_{max}]$ **do**
34: VOTE$[\theta][i]$:= VOTE'$[\theta][2i]$ + VOTE'$[\theta][2i + 1]$;
35: **end for**
36: **end for**
37: outputs VOTE as the result.

4. 投票無衝突化手法と投票空間アクセス局所化手法 [7] による画素並列ハフ変換回路の評価

4.1 無衝突化と局所化された投票メモリへのアクセス

2.4 で述べた投票空間アクセス局所化は回路内の投票メモリを大きく削減することが可能であが，投票の衝突が発生してしまうため投票衝突解消回路が必要であった．そこで，局所窓単位の投票処理において 3. で提案した投票無衝突化手法を適用する．

局所窓単位の投票処理において投票無衝突化手法を適用するために，パラメータ ρ の 0.5 距離精度化とパラメータ θ の値による画像探索方向の選択を行う．投票空間へのアクセスを局所化するための局所窓単位画像探索順に，0.5 距離精度と画像探索方向の選択を適用する．

0.5 距離精度化は投票メモリを 2 倍用意することで達成することができる．画像探索方向の変更は，局所窓内での探索方向を選択することで行い，これにより無衝突な投票を実現することができる．

4.2 回路の評価

N 並列ハフ変換回路が $W \times H$ サイズの画像に対してハフ変換の投票処理を行うときに必要なクロックサイクル数とメモリ量について評価を行った．評価を行うに当たり回路の構成を以下のように定めた．

- 1 クロックで N 画素を画像メモリから取得する
- 外部メモリへの書き込みをパイプライン化するためのバッファを有する
- 回路内に局所窓 2 つ分の画素値を一時保存する
- 分割単位 P を式 (6) のように定め，一つの局所窓中に現れる分割された ρ の値域の範囲の数をを最大 3 にする

$$P = \begin{cases} (N-1)|\sin\theta| & (45° \leqq \theta < 135°) \\ (N-1)|\cos\theta| & otherwize \end{cases} \tag{6}$$

回路構成を図 5 に示す．このような回路構成の場合の必要クロックサイクル数 clock は，式 (7) のようになる．

$$\text{clock} = (N + \frac{W \cdot H}{N} + 3N) \cdot 180 \text{ [cycles]} \tag{7}$$

回路内に必要な画像探索回路内の一時保存レジスタ量 temporary_reg は式 (8)，

$$\text{temporary_reg} = 2N^2 \text{ [bit]} \tag{8}$$

投票メモリのレジスタ量 voting_reg は式 (9)，

$$\text{voting_reg} = 3(N + 1) \cdot B \text{ [bit]} \tag{9}$$

バッファのレジスタ量 buffer_reg は式 (10)，

$$\text{buffer_reg} = (N + 1) \cdot B \text{ [bit]} \tag{10}$$

三角関数テーブルのメモリ量 table_mem は式 (11)

$$\text{table_mem} = 90 \cdot B \text{ [bit]} \tag{11}$$

のようになる．式中の B は $B = \left\lceil \log_2 \left\lceil \sqrt{W^2 + H^2} \right\rceil \right\rceil$ とする．

4.3 提案手法を用いない回路との比較

2.2 で述べたパラメータ θ 並列ハフ変換回路 (θ-para) と，2.3 で述べた提案手法を用いていない画素並列ハフ変換回路 (pix-para) と，4.1 で述べた投票無衝突化と投票空間アクセス局所化による画素並列ハフ変換回路 (col-free+localized) の回路内に必要なメモリ量と必要クロックサイクル数について比較

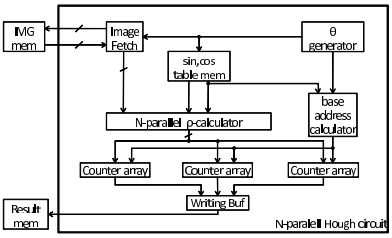

図 5　回 路 構 成

を行った．必要メモリ量として画像の一時保存レジスタ量，投票メモリのレジスタ量，バッファのレジスタ量，三角関数テーブルのメモリ量について比較を行った．

また各回路の構成については，バッファを有し外部メモリへの書き込みをパイプライン化する構成とした．局所窓単位画像探索順を用いる回路 (col-free+localized) においては，局所窓 2 つ分の画素値を一時保存するものとした．入力画像のサイズ 256×256，QVGA，CIF，512×512，VGA，XGA について，並列数 N を $N = 16$ として比較を行った．必要メモリ量を表 1 に，必要クロックサイクル数を表 2 にそれぞれ示す．

表 1　画像サイズに対する必要回路内メモリビット数

画像サイズ	θ-para	pix-para	col-free+localized
256×256	105,930	7,380	2,366
QVGA	116,586	8,046	2,366
CIF	132,426	9,036	2,366
512×512	233,540	15,440	2,572
VGA	257,540	16,940	2,572
XGA	452,254	29,194	2,778

表 2　画像サイズに対する必要クロックサイクル数

画像サイズ	θ-para	pix-para	col-free+localized
256×256	792,256	737,644	748,800
QVGA	928,016	864,401	875,520
CIF	1,223,808	1,140,936	1,152,000
512×512	3,169,024	2,949,846	2,960,640
VGA	3,702,032	3,456,801	3,467,520
XGA	9,457,680	8,848,641	8,858,880

表 1 から，投票無衝突化と投票空間アクセス局所化による画素並列ハフ変換回路 (col-free+localized) は，θ 並列ハフ変換回路や提案手法を用いていない画素並列ハフ変換回路に比べてメモリ量が大幅に少ないのがわかる．また，画像サイズが大きくなっても投票メモリのビット数は他のハフ変換回路と比べて増加量が緩やかである．これは，投票空間アクセス局所化を用いていないハフ変換回路は，画像サイズ $W \times H$ に対して，投票メモリの増加率が $\sqrt{W^2+H^2} \times \left\lceil \log_2 \sqrt{W^2+H^2} \right\rceil$ であるのに対し，投票空間アクセス局所化を用いたハフ変換回路では増加率が $\left\lceil \log_2 \sqrt{W^2+H^2} \right\rceil$ であるためである．そのため，画像サイズが大きくなっても投票メモリ量を小さく抑えることができる．

表 2 において，θ 並列ハフ変換回路 (θ-para) の必要クロックサイクル数が他の回路よりも多いのは，$N = 16$ が 180 の約数でないため計算に無駄が生じているためである．並列数が 180 の約数である場合は画素並列ハフ変換回路と同じクロックサイクル数で処理することができる．

また，投票空間アクセス局所化を用いたハフ変換回路 (localized, col-free localized) の必要クロックサイクル数が投票空間アクセス局所化を行っていない画素並列ハフ変換回路に比べ約 0.12 %から約 1.5 %多いことがわかるがこの必要クロックサイクル数の増加は高速化にほとんど影響を与えるものではない．

5. ま と め

本報告では，画素並列ハフ変換回路を構成するための手法として投票無衝突化を提案した．そして，以前提案した投票空間アクセス局所化手法とともに用いることで小面積な画素並列ハフ変換回路を構成することができることを示した．

投票無衝突化では，画素並列ハフ変換において発生する投票の衝突をなくす，0.5 距離精度での投票とパラメータ θ に応じて並列化の方向を変更する手法を提案した．提案手法を用いた回路は提案手法を用いていない画素並列ハフ変換回路に比べ約 2 倍の投票メモリを有するが，投票の衝突を解消するために必要な回路をなくすことができた．

提案手法と投票空間アクセス局所化手法を用いた画素並列ハフ変換は，投票無衝突であるため投票衝突解消回路が必要ではない．必要クロックサイクル数は約 0.12 %から約 1.5 %増加するが，投票メモリを大きく削減することができる．

この提案により，画素並列ハフ変換回路において投票の衝突を解消することができ，投票空間アクセス局所化手法とともに用いることで画像サイズが大きくなっても小面積、投票無衝突な画素並列ハフ変換回路を構成することができることが明らかとなった．

文　　献

[1] P.V.C Hough, “Method and Means for Recognizing Complex Patterns,” U.S.Patent 3,069,654, Dec.18, 1962.

[2] R.O.Duda and P.E.Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures,” Communications of the Association for Computing Machinery, Vol. 15, pp.11-15, 1972.

[3] 平井 慎一, 座光寺 正和, 増渕 章洋, 坪井 辰彦, “FPGA ベースリアルタイムビジョン,” 日本ロボット学会誌, Vol.22, No.7, pp.873-880, 2004.

[4] E.K. Jolly and M. Fleury, “Multi-sector algorithm for hardware acceleration of general Hough transform,” Image and Vision Computing 24, No. 9, pp.970-976, September, 2006.

[5] Yen-Kuang Chen, Wenlong Li, Jianguo Li, and Tao Wang, “Novel Parallel Hough Transform on Multi-core Processors,” Acoustics, Speech and Signal Processing, 2008., pp.1457-1460, March 31-April 4, 2008.

[6] H. Y. H. Chuang and L. Chen, “An Efficient Hough Transform on SIMD Hypercube,” Proc. Intern. Conf. on Parallel and Distributed Systems, pp.236-241, Dec., 1994.

[7] 後藤正之, 中村一博, 浅田啓一, 高木一義, 高木直史, “投票メモリへのアクセスを局所化した並列ハフ変換回路の構成法,” 信学技報 Vol.107 No.558, pp.79-84, March, 2008.