

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

後藤 正之[†] 中村 一博[†] 浅田 啓一[†] 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,yoshi,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一つに、2値画像から直線や円を検出するハフ変換がある。本稿では、高速化と低消費電力化が求められる組み込み機器に適した直線検出のための並列ハフ変換回路の構成法を提案する。提案法により、ハフ変換の投票処理におけるメモリアクセスを局所化した省メモリな並列ハフ変換回路が構成できる。新たに提案するパッチを用いた画像探索順に基づいて投票処理を行う。この探索順により、回路内の投票メモリのインデックス数と画像サイズの依存関係をなくすることができる。局所化を考慮しない回路構成に比べ回路内の必要メモリ量をQVGA画像サイズで約97%、SVGA画像サイズでは約99%削減することができる。

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, メモリアクセス局所化, 並列ハフ変換回路

A VLSI Architecture of Parallel Hough Transform with Localized Voting Space Access

Masayuki GOTO[†], Kazuhiro NAKAMURA[†], Yoshikazu ASADA[†], Kazuyoshi TAKAGI[†], and
Naofumi TAKAGI[†]

[†] Graduate School of Information Science, Nagoya University

Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan

E-mail: †{gotoh,nakamura,yoshi,ktakagi,ntakagi}@takagi.i.is.nagoya-u.ac.jp

Abstract Hough transform is a computer vision algorithm, which detects lines and circles in a binary image. We propose a VLSI architecture for Hough transform. The architecture is memory efficient and suitable for embedded systems that require real-time performance and low-power. The architecture implements parallel Hough computation with localized voting space access. The voting process is based on proposing patch-based image access order. This image access order enables the voting space access to be localized. This causes the number of indexes of voting memory in the circuit to be independent of image size. The necessary voting memory in the circuit can be reduced by about 97% on QVGA, and about 99% on SVGA compared with the circuit architecture in which localization is not considered.

Key words Hough transform, ASIC, image processing, memory access localization, 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用機器等に拡大しており、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、画像から直線などの図形検出を行うハフ変換 [1] は、入力画像のノイズや欠落の影響を受けること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、指紋認識や道路標識・車線認識等に用い

られている。

ハフ変換は、 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 である特徴点を通りうる直線をパラメータで表現する。この表現式を投票式と呼び、投票式を満たすすべてのパラメータに対して投票する。このハフ変換処理は、扱う画素数が多くメモリアクセス量も大きい。そこでハードウェア化による高速化の研究がおこなわれてきている [3]。しかし、高速化するために並列度を高めると必要な演算器数の増加を招き、回

路面積の増加につながっていた。また、処理する画像のサイズが大きくなるにつれて投票メモリ量が膨大となり、回路に占める割合が増大する。投票計算では多くのメモリを必要とし、回路面積と消費電力がおおきくなるため、高速かつ小面積、低消費電力であるハードウェア構成が求められている。

本稿では、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を提案する。提案法は一般的に画像処理で用いられる画像ストリーミング順とは異なる、ハフ変換に適した画像探索順に基づいている。この探索順は局所的な窓領域単位であるパッチでの画像探索順であり、パラメータ ρ が持つ分布の規則性に着目している。

直線検出のためのハフ変換はパラメータ θ と画素 (x, y) よりパラメータ ρ が一意に決まる。ある θ において画素 (x, y) とパラメータ ρ との関係をもとに画像を探索することでパラメータ ρ の出現を局所化できる。提案画像探索順は画像サイズが大きくなるにつれて投票空間が膨大になるのを軽減できる。これにより投票空間として用意しなければならないメモリ量を少なくすることができ、小面積化が可能である。さらに並列に ρ を計算し投票するため、高速処理も実現できる。

以降、2 ではハフ変換および以前に提案されたハフ変換回路について述べる。3 では画像平面に関する並列化によるハフ変換回路構成を提案し、4 ではこの構成法をさらに効率化した、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構成を提案する。5 では評価を行う。6 ではまとめを述べる。

2. 準備

本節では、P.V.C.Hough により提案されたハフ変換 [1] と、それを計算機で扱いやすいように改良した Duda らのハフ変換 [2] について述べる。そして、ハフ変換の専用回路を FPGA に実装した平井らの並列ハフ変換回路 [3] について述べる。

2.1 ハフ変換

ハフ変換は、入力 2 値画像から直線や円を検出する手法である [1]。ハフ変換は、投票処理と多数決処理によって直線や円を検出する。

直線 (円) の検出のための投票処理では、画像平面上の画素値が 1 である特徴点について処理を行う。各特徴点について、その特徴点を通る、切片と傾き (半径と中心座標) をパラメータとしたすべての直線 (円) に投票を行う。ここで、特徴点を通る直線 (円) はパラメータによって一意に表現される。すべての特徴点について投票が行われる。

多数決処理では、パラメータによって表現された直線 (円) の中で、投票値が多い直線 (円) を実際に存在する直線 (円) として検出する。以下では、本研究で対象とする直線検出のためのハフ変換について述べる。

入力画像は、2 値化された白黒画像とし、画像平面に $(x_1, y_1), \dots, (x_n, y_n)$ の n 個の特徴点があるとす。画像平面における傾き a 、 y 切片 b の直線は、式 (1) のように表現される。

$$y = ax + b \quad (1)$$

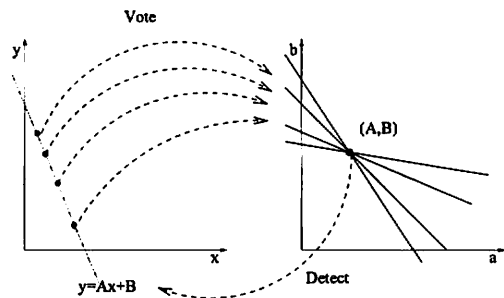


図1 a-bパラメータ空間を用いたハフ変換

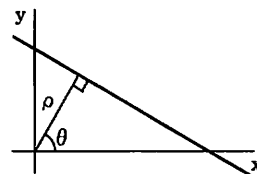


図2 Duda らによる直線の表現法

図1は画像中に存在する直線 $y = Ax + B$ がハフ変換により検出される様子を示している。1つの特徴点 (x_i, y_i) が与えられたとき、 $y_i = ax_i + b$ を満たすすべてのパラメータの組 (a, b) に対して投票される (図1の Vote)。 n 個の特徴点すべてについて投票を行い、パラメータによって表現される直線のうち、投票値が最多のパラメータの組 (A, B) を、直線として検出する (図1の Detect)。

パラメータによって直線が表現される空間を投票空間と呼ぶ。計算機上でこのハフ変換を行う場合は、投票空間を2次元の累算値配列で実現する。入力画像として使用される画像平面は、通常有限平面であり、 x, y ともに有限である。しかし、上で述べた方法によってハフ変換を行うと、 $a-b$ 投票空間が有限にならない。計算機上では有限のメモリしか持たないため、有限ではないパラメータ空間を扱うことはできない。次小節において、計算機上で扱うための直線検出用ハフ変換を示す。

2.2 Duda らによるハフ変換

2.1 で述べたように、Hough が提案した直線検出のためのハフ変換は、画像平面が有限であるにもかかわらず、投票空間が有限にならない。

そこで Duda らは、画像平面上の直線を図2のように、 $x-y$ 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 を用いた。直線を表わす式は式 (2) のように表現し、投票空間として $\rho-\theta$ 投票空間を用いることを提案した [2]。

$$\rho = x \cos \theta + y \sin \theta, (0^\circ \leq \theta < 180^\circ) \quad (2)$$

式 (2) 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、 $\rho-\theta$ 投票空間も有限となる。このため計算機上で取り扱うことが可能となる。計算機上で $\rho-\theta$ 空間を用いたハフ変換を行うときは、 ρ, θ をそれぞれ量子化し、2次元の累算値配列 $H(\rho, \theta)$ で投票空間を実現する。

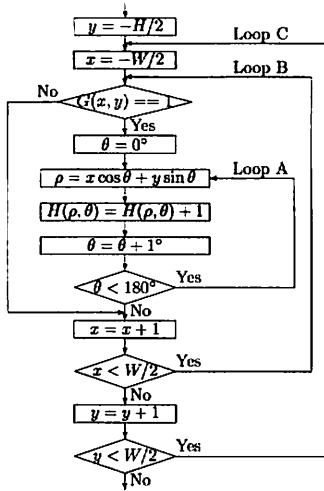


図3 ハフ変換のフローチャート

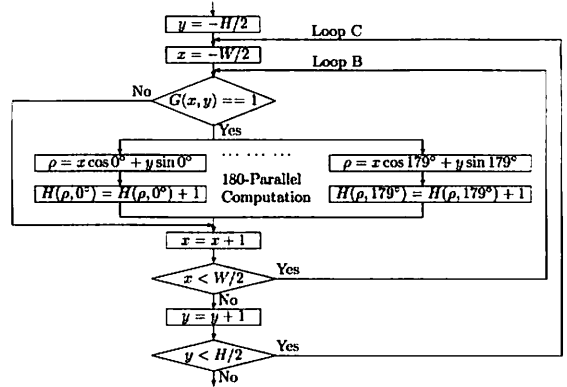


図4 平井らの並列ハフ変換のフローチャート

投票処理は、各特徴点 (x_i, y_i) に対して、その特徴点の画素値が1の場合のみ行われる。 θ の値を 1° ずつずらしながら式(2)の ρ を計算し、累算値配列の要素 $H(\rho, \theta)$ を加算する。

ρ - θ 投票空間を用いたハフ変換のフローチャートを図3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。

図3が示すように、処理が3重ループになっていることが計算量を増加させている要因である。入力画像のサイズが大きくなるほど、 x と y の繰り返し回数が増え、計算量が増大する。

Duda らの方法によるハフ変換は、

- 画像平面の1点が、投票空間のひとつの正弦曲線に対応する
- 投票空間の1点が、画像平面のひとつの直線に対応する
- 画像平面で同一直線上にある特徴点群は、投票空間では共通する点を通る曲線群に対応する
- 投票空間の同一曲線上にある点群は、画像平面で同じ点を通る直線群に対応するという特徴を持っている。

2.3 並列ハフ変換回路

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理を行うことが困難である。そのためこれまでに平井らにより並列ハフ変換回路が提案された [3]。

平井らは Duda らのハフ変換のパラメータ θ に対する並列性に着目し、パラメータ θ に関する並列処理を行う回路を提案した。図4に平井らの並列処理のフローチャートを示す。図4において、図3のパラメータ θ についてのループである Loop A を展開して並列計算を行っている。

平井らはこの回路を FPGA 上で実現し評価を行った。 θ に対して 180 並列で ρ の計算をすることで大幅に計算時間を短縮することに成功した。しかし 180 並列であるため回路規模が大きくなってしまふ。計算して得られる ρ の範囲は、

$$-\frac{\sqrt{W^2 + H^2}}{2} \leq \rho \leq \frac{\sqrt{W^2 + H^2}}{2} \quad (3)$$

であるため、投票空間として用意する投票メモリのインデックス数は $180 \times \lceil \sqrt{W^2 + H^2} \rceil$ である。そのため回路内に膨大なメモリを持つことが必要であった。

次節に、この欠点を回避する省メモリ型の回路構成手法を提案する。

3. 画像平面に関する並列処理によるハフ変換回路構成

3.1 画像平面に関する並列処理によるハフ変換

本節では画像平面に関して並列化などの効率化を行った並列ハフ変換回路構成を提案する。図5に提案するハフ変換処理のフローチャートを示す。図3の画像平面に関するループである Loop B, C を並列に展開している。画像平面に関して N 並列化することにより高速化を実現する。

ハフ変換の式(2)より、座標 (x, y) と $(x + N, y)$ における ρ の値の差は $N \cos \theta$ である。このことから、 $(x + N, y)$ における ρ の値は (x, y) における ρ の値に $N \cos \theta$ を加算することで容易に計算を行うことができる。また、画像探索において y 行目から $y + 1$ 行目に移動する際には、座標 (x, y) と $(x + N - W, y + 1)$ における ρ の値の差である $-(W - N) \cos \theta + \sin \theta$ を加算している。これにより乗算器を用いずに回路を構成でき、回路規模を小さくすることができる。

3.2 画像平面に関する並列処理によるハフ変換回路の構成

提案回路構成は高速化だけではなく演算器やメモリ量の削減も考慮した回路構成である。3.1 で述べたように、乗算器を用いずに計算ができるため、回路規模を小さくすることができる。また、外部メモリに格納するために書き込み用バッファを用意することで、書き込みと次の投票処理を並列に行うことが可能である。図6に回路構成を示す。図6は並列に N 個の ρ の値を計算し、投票メモリで投票を行う。 ρ を計算する式は、[3] と同じく式(2)を用いているため、計算して得られる ρ の範囲は [3] と等しい。投票空間として用意する投票メモリのインデックス数は式(3)より $2 \lceil \sqrt{W^2 + H^2} \rceil$ である。

メモリの各セルのビット数は最大投票数により決定される。

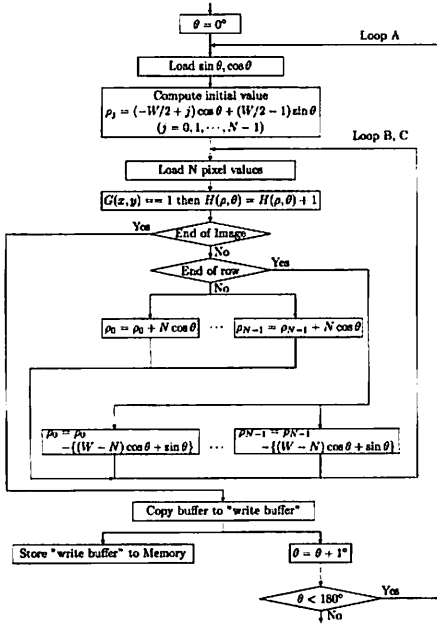


図5 提案する並列ハフ変換処理のフローチャート

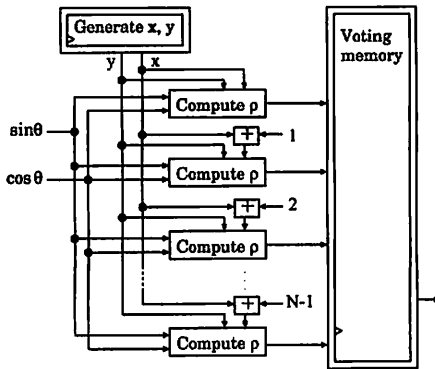


図6 提案する並列ハフ変換回路の構成

表1 必要投票メモリビット数

画像サイズ	[3]	提案回路
QVGA(320 × 240)	648,000	7,200
CIF(352 × 288)	737,100	8,190
VGA(640 × 480)	1,440,000	16,000
SVGA(800 × 600)	1,800,000	20,000

最大投票数は $\lceil \sqrt{W^2 + H^2} \rceil$ であるため、各セルのビット数は $\lceil \log_2 \lceil \sqrt{W^2 + H^2} \rceil \rceil$ となる。そのため回路内に必要な投票メモリ量は $2 \lceil \sqrt{W^2 + H^2} \rceil \times \lceil \log_2 \lceil \sqrt{W^2 + H^2} \rceil \rceil$ [bit] となる。各画像サイズに対して必要になる回路内の投票メモリ量を表1に示す。

[3]は θ について180並列で処理しているため、この回路構成よりも180/N倍高速であるが、提案回路構成では回路内の投票メモリ量を[3]に比べ、約99%削減できた。

次節において、さらに投票メモリ量を削減する構成手法を提

案する。

4. 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構成

本節では ρ - θ 投票空間へのアクセスを局所化させる画像探索順を用いた並列ハフ変換回路構成を提案する。 $N \times N$ 画素をひと固まりとした局所的な窓領域単位をパッチと呼ぶ。この探索順はパッチ単位での画像探索順であり、パラメータ ρ が持つ分布の規則性に注目している。

これまで回路内のメモリにすべての ρ の値域を静的にマッピングしていたが、投票空間へのアクセスの局所化を行うことで、投票空間を部分的かつ動的に順次マッピングすることで投票処理を行うことができる。これにより回路内のメモリを削減することが可能になる。

4.1 投票メモリへのアクセスの局所化

画像平面上にどのように ρ の値が分布するかを解析した。その分布傾向に基づいて探索順を提案する。2.2で述べたハフ変換の計算式(2)のパラメータ ρ は、画像平面上の座標 (x, y) とパラメータ θ の関数として分布する。この ρ の値の分布は連続的であり、ある画素の ρ の値と隣接する座標の ρ の値は近い値となる。

$\theta = 45^\circ$ のときの画像平面上の ρ の分布を図7に示す。図7ではVGA画像サイズの画像平面上に ρ がどのように分布するかを示している。

図7において、 ρ の最小値は画像の左下の画素において出現し、最大値は右上の画素において出現している。 $0^\circ \leq \theta < 180^\circ$ について調査したところ、 $0^\circ \leq \theta < 90^\circ$ では図7と同様に左下の画素が最小値で右上の画素が最大値であった。そして $90^\circ \leq \theta < 180^\circ$ では、右下の画素が最小値で左上の画素が最大値であった。この傾向を基に投票メモリへのアクセスを局所化する。

提案手法では次の3つの手法により投票メモリへのアクセスの局所化を行う。

- (1) ρ - θ 投票空間の分割
- (2) パッチベースの画像探索順
- (3) 閾値に基づく投票メモリへのアクセス

以下、各手法について述べる。

4.2 ρ - θ 投票空間の分割

2.2で述べたハフ変換の計算式(2)のパラメータ ρ は、図7のように連続的に分布している。この ρ の値により探索する画素を決定していく。そこで ρ の値域を分割し、範囲内の ρ になる画素を範囲ごと探索することで、局所化を行うことができる。

定理1. 分割単位は式(4)のように θ の値により異なるように設定することで、一つのパッチにおいて出現する ρ の範囲を最大で3にすることができる。

$$\begin{cases} N \cos \theta & (0^\circ \leq \theta \leq 45^\circ, 135^\circ \leq \theta \leq 180^\circ) \\ N \sin \theta & (45^\circ \leq \theta \leq 135^\circ) \end{cases} \quad (4)$$

証明1. パッチ内における ρ の最大値と最小値の差は

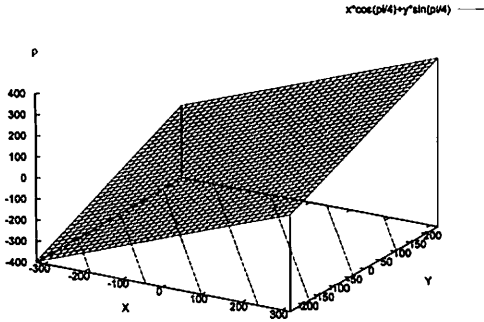


図7 $\theta=45^\circ$ のときの画像平面上における ρ の分布

$N|\cos\theta| + N|\sin\theta|$ と表わされる。パッチ内に分割された ρ の範囲が3より多く出現しないためには、最大値と最小値の差が分割単位の2倍以下であることが必要十分条件である。

i) $0^\circ \leq \theta < 45^\circ, 135^\circ \leq \theta < 180^\circ$ のとき式(4)より

$$N|\cos\theta| + N|\sin\theta| \leq 2N|\cos\theta|$$

この式は成立する。

ii) $45^\circ \leq \theta < 135^\circ$ のとき式(4)より

$$N|\cos\theta| + N|\sin\theta| \leq 2N|\sin\theta|$$

この式は成立する。

i,iiより、式(4)のように分割単位を設定することによりパッチ内には最大で3つの範囲までしか出現することはない。□

4.3 パッチベースの画像探索順

一般的なストリーム順とは異なった、範囲内の ρ の値を持つ画素を範囲ごとに探索する画像の探索順を提案する。この探索順はパッチ単位での画像探索順であり、パラメータ ρ が持っている分布の規則性に着目している。パッチベースの画像探索順のフローチャートを図8に示す。 θ の値によってパッチの変更手続きを4パターンに分けている。手続きのフローチャートを図9に示す。図9において、 $\text{Img}[LB]$ と $\text{Img}[LT]$ はそれぞれ画像中の最左下と最左上にあるパッチのことである。 T_h は ρ の最大値と最小値、そして式(4)によって決定される。 T_h によって区切られる ρ の範囲を基に画像探索を行っていく。

図10に各手続きにおけるパッチ探索順の例を挙げる。 T_h に基づき、注目している ρ の範囲を含むパッチを全て探索してから次に注目する ρ の範囲に変更する。

4.4 閾値に基づく投票メモリへのアクセス

処理する画像のサイズが大きくなるにつれて投票メモリ量がハードウェアに占める割合が増大する。平井らの回路構成の投票メモリ量は ρ の値域と最大投票数によって決定される。平井らの回路構成では、式(3)で表わされる ρ の値域を被覆できる投票メモリを回路内に用意している。そのため投票メモリが大きくなっていた。

それに対し、提案回路構成では回路内に用意する投票メモリ量は ρ の分割単位最大値と、一つのセルに必要なビット数

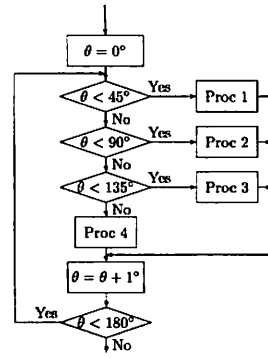


図8 パッチベースの画像探索順を用いたハフ変換のフローチャート

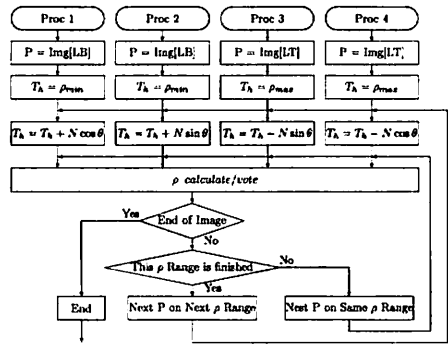


図9 パッチベースの画像探索順を用いたハフ変換の手続きのフローチャート

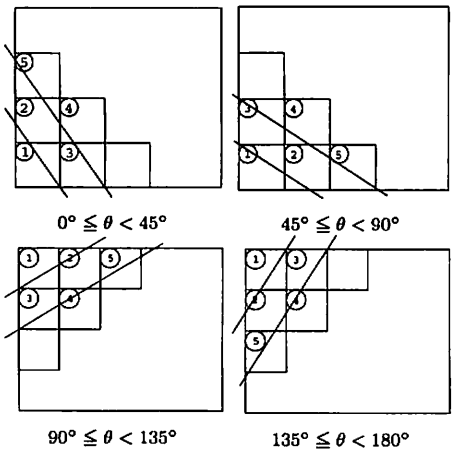


図10 提案パッチ探索順

によって決定される。しかし ρ の分割単位の最大値は画像サイズが変化しても一定であるため、一つのセルに必要なビット数によってのみ決定される。一つのセルに必要なビット数は $\lceil \log_2(\sqrt{W^2 + H^2}) \rceil$ で表わされ、画像サイズが大きくなっても増加分は式(3)で表わされる ρ の値域に比べ、大きく変化しない。そのため回路内に用意する投票メモリを抑えることができる。

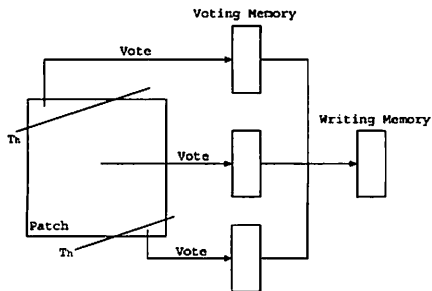


図 11 パッチ単位での投票処理

4.5 投票回路の構成

パッチベースの画像探索順に基づいた投票回路を構成する。投票空間へのアクセスの局所化を行うことで、それまで回路内のメモリに横俵していた投票空間を局所的に用意することで代替できる。これにより回路内のメモリを削減を実現した。

一回の演算により得られた投票値を毎回メモリに書き込むようにすると書き込みコストによる遅延や書き込みの競合が起こってしまう可能性がある。そのため投票計算用メモリと投票結果出力メモリを持つこととし、投票計算と投票結果出力を同時実行することで遅延や競合を回避する。

一つのパッチにおいて出現する ρ の分割単位は最大で3つであるため、回路内に投票用のメモリを3つの投票メモリと1つの外部メモリへの書き込み用メモリの4つを用意する。そのメモリは1つあたり $\frac{\sqrt{2}N}{2} \lceil \log_2 \lceil \sqrt{W^2 + H^2} \rceil \rceil$ [bit] である。

4.6 回路の動作

N 並列に ρ を計算することができる回路によって ρ を計算し、得られた ρ の値をインデックスとして投票を行っていく。図 8,9,10 に従いパッチの探索を行う。探索しているパッチの画素について N 並列で投票を行う。図 11 にパッチを ρ の値によって分割し、投票を行う場合の投票処理を示す。 T_h によって投票先となる投票メモリに振り分けられ、投票メモリ内にマッピングされたセルに投票を行っていく。ある ρ の範囲の探索がすべて終了したら、その範囲をマッピングしている投票メモリから投票結果を書き込みメモリに退避させる。それにより次の投票処理と外部メモリへの書き込みを並列に行うことができ、高速に動作させることができる。

5. 評価

画像平面に対する並列数 N を $N = 32$ とした回路を対象にして評価を行った。QVGA,CIF,VGA,SVGA の各画像サイズに対する投票空間に必要な投票メモリ量を表 2 に示す。また、必要クロックサイクル数を表 3 に示す。

表 2 から、局所化を行っていない提案回路に対して局所化を行った提案回路の投票メモリは QVGA 画像サイズで約 97% のメモリ量を削減できており、CIF 画像サイズでは約 98%,VGA 画像サイズでは約 99%,SVGA 画像サイズでは約 99% のメモリ量を削減できた。このことから、提案画像探索順を用いることで必要投票メモリ量を大幅に減らすことが可能である。また、

表 2 画像サイズに対する投票メモリビット数

画像サイズ	局所化なし	局所化あり
QVGA(320 × 240)	7,200	204
CIF(352 × 288)	8,190	204
VGA(640 × 480)	16,000	227
SVGA(800 × 600)	20,000	227

表 3 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	[3]	局所化した提案回路
QVGA(320 × 240)	153,600	432,000
CIF(352 × 288)	202,752	570,240
VGA(640 × 480)	614,400	1,728,000
SVGA(800 × 600)	960,000	2,700,000

画像サイズが大きくなって必要投票メモリ量を少なく抑えることが可能である。これにより、専用回路に実装することが可能であるといえる。

また、表 3 で平井らの並列ハフ変換回路と局所化した提案ハフ変換回路の必要クロックサイクル数を比較している。 θ に対して 180 並列を行った平井らの回路に比べ、画像平面に対して 32 並列しか行っていないため、必要クロックサイクル数は平井らの回路より多くなっている。しかし、一秒間に 30 枚の画像を処理するために必要な動作周波数は、QVGA 画像サイズで約 13MHz、SVGA 画像サイズで約 81MHz である。これは、現在の組み込み機器の動作周波数でリアルタイム処理を実現できる。

6. まとめ

ハフ変換の投票空間へのアクセスを局所化させる画像探索順を用いた並列ハフ変換回路構成を提案した。投票空間へのアクセスを局所化させることで回路内に必要とされる投票メモリ量を大幅に減らすことができる。画像サイズが大きくなるにつれて回路内の投票メモリが膨大になることを回避することができ、局所化を考慮しない回路構成に比べ回路内の必要メモリ量を QVGA 画像サイズで約 97%, SVGA 画像サイズでは約 99% 削減することができた。

今後の課題として設計評価することと、投票アクセス局所化と θ に対する並列化を両立させることが挙げられる。

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 ACM, 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.