

## ダイナミック・リコンフィギュラブル技術を用いたハフ変換回路の設計 Design of a Hough Transformation Circuit Using Dynamic Reconfigurable Technology

望月 要人<sup>†</sup>  
Kanato Mochizuki

中野 秀洋<sup>‡</sup>  
Hidehiro Nakano

宮内 新<sup>†</sup>  
Arata Miyauchi

### 1. はじめに

コンピュータビジョンアルゴリズムの1つに2値化された入力画像から直線を抽出するハフ変換がある。ハフ変換は入力画像のノイズや欠落による影響を受けにくく、安定した直線の抽出を行うことができるという特徴を持つため、道路標識の認識や生体認証などへ適用する研究が行われている。しかし計算量と使用メモリ量が多く、携帯機器などの小型回路への実装が困難である。そのため、回路の小規模化が課題となっている。

近年、ハードウェアの小型化と高速化を両立できる技術として、回路の構成を処理の実行中に再構成することができるダイナミック・リコンフィギュラブル技術が注目されている。

本研究では、ダイナミック・リコンフィギュラブル技術を用いてハフ変換による直線抽出回路を設計することで、処理の高速化と回路の小型化を図ることを目的とする。

### 2. 準備

#### 2.1 ハフ変換

ハフ変換は、入力された2値画像から直線を抽出する手法である。ハフ変換は投票処理と多数決処理によって直線を抽出する。

図1にハフ変換による直線抽出の様子を示す。x-y平面上の直線から原点に下した長さ $\rho$ の垂線とx軸の成す角を $\theta$ とすると、x-y平面上のある点を通る任意の直線は式(1)で表現できる。投票処理では、入力画像中の画素値が1の画素(x,y)において式(1)による $\rho$ - $\theta$ 空間への変換を行う。 $\rho$ - $\theta$ 空間の各座標は値(投票値)を持っており、変換して得られた座標( $\rho, \theta$ )における投票値を+1する。

$$\rho = x \cos \theta + y \sin \theta \quad (0^\circ \leq \theta < 180^\circ) \quad (1)$$

多数決処理では、投票処理によって得られた $\rho$ - $\theta$ 空間の各座標における投票値を評価し、投票値が予め設定した閾値を上回る点を直線として抽出する。

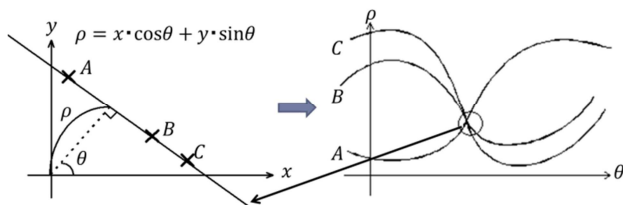


図1 ハフ変換による直線抽出

ハフ変換は外乱にロバストなアルゴリズムとして生体認証や道路標識の認証などに応用されることが期待されているが、計算量が多くソフトウェアでのリアルタイム処理が困難である。そのためハードウェア化による並列化の研究が行われている[1][2]。専用の計算を行うハードウェアはできるだけ小型であることが望ましいが、処理を高速化するために並列数を上げれば当然回路規模も増大するため、小型化は困難になる。

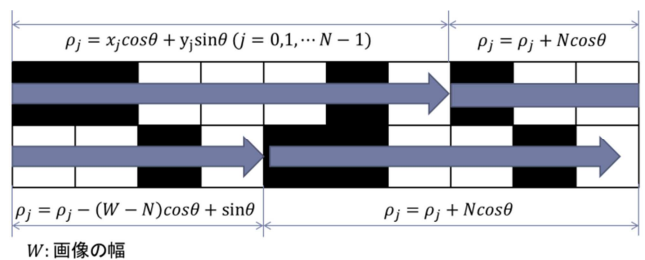
#### 2.2 ダイナミック・リコンフィギュラブル

FPGAのような回路構成をプログラムによって変更可能なデバイスにおいて、回路構成を処理の実行中に再構成することができる技術をダイナミック・リコンフィギュラブルという。ある処理において、逐次的に実行される処理ステップ毎に、その処理ステップに応じた最適な回路構成に回路を切り替えることで、処理を高速化することができる。また、回路構成を動的に切り替えることで、1つの処理要素に複数の機能を割り当てることができるために、回路の小型化も同時に図ることができる。

近年ではこのような動的な再構成を高速に(数 ns~数十 ns)実行できるようなデバイスが登場してきており、実際にアプリケーションを実装する研究が盛んである[3]。本研究では、ダイナミック・リコンフィギュラブル技術を用いてハフ変換の並列化回路を複数の回路構成に分割することで、処理の高速化と回路の小型化を図ることを目的とする。

### 3. 画素並列ハフ変換回路の構成分割

本研究では入力画像の画素について並列化した回路[2]を分割することを提案する。画素並列ハフ変換回路では、角度 $\theta$ を固定し、N個の画素(x,y)について並列に $\rho$ 計算を行うことで処理を高速化している。図2にこの並列化手法による $\rho$ 計算の様子を示す。式(1)より、画素(x,y)と(x+N,y)の $\rho$ の値の差は $N \cos \theta$ である。このことから、画素(x+N,y)における $\rho$ の値は、画素(x,y)における $\rho$ の値に $N \cos \theta$ を加算することで容易に求めることができる。また、画像探索においてy行目からy+1行目に移動するときは、画素(x,y)と(x+N-W,y+1)における $\rho$ の値の差である $-(W-N) \cos \theta + \sin \theta$ を加算する。これによって乗算器を用いずに回路を構成することができ、回路規模を小さくすることができる。



W: 画像の幅

図2 差分値の加算による $\rho$ 計算

このように画素並列の回路における $\rho$ の値の計算は2つ

<sup>†</sup> 東京都市大学大学院工学研究科

Graduate School of Engineering, Tokyo City University

<sup>‡</sup> 東京都市大学知識工学部情報科学科

Department of Computer Science, Faculty of Knowledge Engineering, Tokyo City University

のステップに分かれている。すなわち、最初の N 並列分の  $\rho$  の値を通常通り式(1)で計算し、それ以降は  $\rho$  の値が既知である画素からの差分をとって計算する。この 2つのステップ毎に回路構成を切り替えることで回路の小型化を図る。

図 3 に提案する回路構成を示す。Config1 で式(1)による  $\rho$  計算を行い、Config2 でそれ以降の差分を加算する計算を行う。予備実験において回路量の見積もりを行った結果、Config2 では並列数を Config1 の 3 倍にすることで回路規模の平準化を図る。

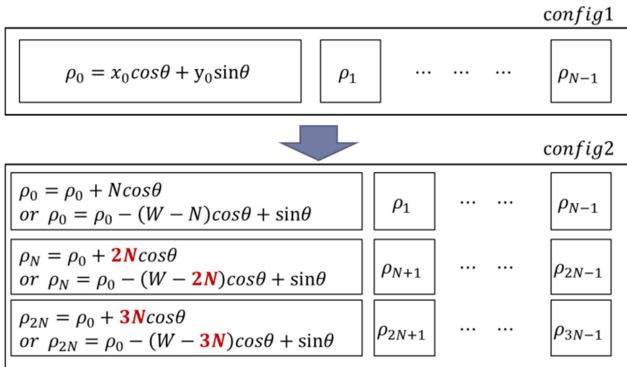


図 3 画素並列回路の分割構成

4. 評価

提案した回路構成をそれぞれ Verilog-HDL で作成し、回路規模と処理速度の評価を行った。回路規模の評価は Altera Quartus-II ver11.1 を使用し、FPGA をターゲットデバイスとした場合の LE 数で評価した。処理速度の評価は Altera Modelsim Starter Edition 10.0c を用いた HDL シミュレーションを行い、画像サイズ 640×480 の画像について、すべての画素の変換が終了するまでのクロック数を計測した。ここで、回路の再構成は 1クロックで行えるものとし、使用するコンフィギュレーションを切り替えるタイミングで 1クロック分の遅延を挿入することでオーバーヘッドを考慮する。

表 1 に提案回路構成の各コンフィギュレーションにおける回路量を示す。ここでは並列数を 3 倍にした Config2 の方が回路量が大きくなっており、これが実際に必要な回路量となる。いずれの並列数においても、全体の回路量において、Config1 が占める割合が約 48%、Config2 が占める割合が約 52% となり、二つのコンフィギュレーションの回路量はほぼ平準化できているといえる。

表 2 に回路を分割しない従来の画素並列回路と提案回路の回路量の比較を示す。表 2 の結果から、提案回路は従来の画素並列回路よりも平均で約 20% 回路量を削減できていることが分かる。

表 3 に処理クロック数の比較結果を示す。回路の再構成にかかるオーバーヘッドを考慮しても、提案回路の方が約 3 倍高速に実行できていることが分かる。1 度の画像探索において、Config1 の処理が行われるのは 1クロックのみであり、それ以降の計算は全て Config2 で行っているため、回路の再構成はほとんど行われない。これによって再構成のオーバーヘッドを非常に小さくすることができている。また、提案回路では Config2 の計算を Config1 の 3 倍の並列数で行っているため、実際にはほとんどの計算を非並列時の 3 倍の速度で行うことができる。

表 1 分割した回路構成の各回路量

並列数	回路量(LE 数)		
	config1	config2	合計
10	7333(48%)	<b>7816(52%)</b>	15149
20	12574(48%)	<b>13426(52%)</b>	26000
30	17816(48%)	<b>19390(52%)</b>	37206
40	23054(48%)	<b>24590(52%)</b>	47644
50	28294(48%)	<b>31199(52%)</b>	59493
60	33535(48%)	<b>36274(52%)</b>	69809

表 2 従来の画素並列回路との回路量の比較

並列数	回路量(LE 数)	
	従来回路	提案回路
10	9943	7816(79%)
20	17058	13426(79%)
30	24247	19390(80%)
40	31267	24590(79%)
50	38689	31199(81%)
60	45589	36274(80%)

表 3 並列数に対する処理クロック数

並列数	処理クロック数	
	従来回路	提案回路
10	5529599	1843738(33%)
20	2764799	922318(33%)
30	1843199	615398(33%)
40	1382399	461878(33%)
50	1105919	369898(33%)
60	921599	308638(33%)

5. 結論

ハフ変換の画素並列回路にダイナミック・リコンフィギュラブル技術を適用し、回路構成を分割することを提案した。評価の結果、提案した回路構成では回路規模を約 20% 削減することができ、さらに 3 倍高速に実行することができた。

今回は回路量の評価を FPGA の LE 数で行ったため、実際のダイナミック・リコンフィギュラブルプロセッサを用いた提案手法の評価が今後の課題として挙げられる。

参考文献

- [1] 平井慎一, 座光寺正和, 増淵章洋, 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= Journal of Robotics Society of Japan 22(7), 873-880, 2004-10-15
- [2] 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”, 電子情報通信学会技術研究報告. CPSY, コンピュータシステム 107(558), 79-84, 2008-03-2
- [3] 木原 拓, 清水 翔, 高山, 荒川 豊, 山中 直明, 渡辺 昭文, “並列プロセッサ DAPDNA-2 を用いたリンクディスジョイント経路計算の高速解法”, 電子情報通信学会技術研究報告. VLD, VLSI 設計技術 108(412), 201-206, 2009-01-22