

GPU を利用した一般化 Hough 変換アルゴリズムの実装 Generalized Hough Transform on GPUs

吉田 光寿†

宮永 直樹‡

Teruhisa Yoshida Naoki Miyanaga

1 背景と目的

近年, GPGPU[1] という研究が注目を浴びている。これは安価でありながら高い演算性能を持つ GPU に, CPU で行われるような一般的な計算をさせて処理の高速化を図るという試みである。GPU はもともと画像表示に必要な演算を行う事を目的としたハードウェアである。そのため, 画像処理に関連した汎用計算処理を GPU に行わせる研究が特に盛んである。

本研究では, 画像認識アルゴリズムの一種である一般化 Hough 変換 [2] を GPU 上で実装するための手法を提案する。すでに Strzodka らによって類似した研究が行われている [3] が, 本研究で提案する手法は, 処理時間を更に短縮するものである。

2 Strzodka らの手法

2.1 概要

Strzodka らによる提案で GPU に実装される一般化 Hough 変換は, GPU 上での実装に適した形に改変されている [3]。次に, アルゴリズムの概要を示す。

1. テンプレート画像のエッジを構成するすべてのエッジ点について, エッジ点からテンプレート画像中央への距離 l^r , それぞれのエッジ点とテンプレート画像中央を結ぶ線分と水平成分との成す角度 l^θ を格納したリスト $l = \langle (l_i^r, l_i^\theta) \rangle_{i=1}^{|l|}$ を準備する。ここで, $|l|$ はリスト l の要素数を表す。
2. 入力画像の各点 (x, y) についてそれがエッジであるか否かを示す値 $I^{edge}(x, y) = v(d)$ を準備する。ここに, $0 \leq d$ は直近のエッジ点までの距離, $v(d)$ は $v(d) = 1$ で任意固定の非負単調減少関数とする。
3. 適当に量子化した三次元投票空間 $Vote$ を用意し, 0 で初期化する。 $Vote$ の大きさは $W \times H \times D$ とする。ここで, W, H はそれぞれ入力画像の幅と高さ, D は求める角度の検出精度によって定める任意の値である。
4. $Vote$ のすべての要素について, $Vote(x, y, \omega)$ の得票率を次の式により求める。

$$Vote(x, y, \omega) = \frac{\sum_{i=1}^{|l|} I^{edge}(V_x, V_y)}{|l|} \quad (1)$$

ここで,

$$V_x = x + l_i^r \cos(l_i^\theta + \frac{2\pi\omega}{D}) \quad (2)$$

$$V_y = y + l_i^r \sin(l_i^\theta + \frac{2\pi\omega}{D}) \quad (3)$$

である。

5. 最も得票率の大きい要素 $Vote(x_{MAX}, y_{MAX}, \omega_{MAX})$ を探し, これをテンプレートと入力画像の一致率とする。

2.2 問題点

この手法では投票空間 $Vote$ の要素すべてについて式 (1) の計算を行う事になる。これは W, H, D の値が大きくなると共に膨大な計算量となる。

Strzodka らはこの問題に対してバウンディングボックスによる投票範囲の制限等で対処しているが, 根本的な解決にはなっていない。

3 提案手法

3.1 概要

提案手法では, 通常的一般化 Hough 変換にできるだけ沿った形のアルゴリズムを実装する。次に, アルゴリズムの概要を示す。

1. Strzodka らの手法で準備したリスト l の情報に加え, エッジ点におけるエッジとの接線と水平成分との成す角度 l^ω を格納したリスト $l = \langle (l_i^r, l_i^\theta, l_i^\omega) \rangle_{i=1}^{|l|}$ を準備する。
2. 入力画像のエッジを構成するすべてのエッジ点について, それぞれの位置 (m^x, m^y) , エッジ点におけるエッジの接線と水平成分との成す角度 m^ω を格納したリスト $m = \langle (m_j^x, m_j^y, m_j^\omega) \rangle_{j=1}^{|m|}$ を準備する。
3. Strzodka らの手法と同様に投票空間 $Vote$ を用意し, 0 で初期化する。
4. 投票処理を行う。すべての i, j について

$$V_\omega = m_j^\omega - l_i^\omega \quad (4)$$

$$V_x = m_j^x + l_i^r \cos(l_i^\theta + V_\omega) \quad (5)$$

$$V_y = m_j^y + l_i^r \sin(l_i^\theta + V_\omega) \quad (6)$$

を求め, $Vote(V_x, V_y, [\frac{V_\omega D}{2\pi}])$ の値を $1/|l||m|$ だけ増加させる。

5. 最も得票率の大きい要素 $Vote(x_{MAX}, y_{MAX}, \omega_{MAX})$ を探し, これをテンプレートと入力画像の一致率とする。

†早稲田大学大学院 理工学研究科 寛研究室

‡株式会社マルチタム 経営企画室

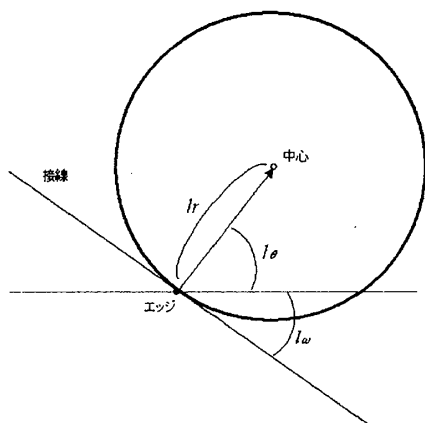


図1: リスト l に格納する情報

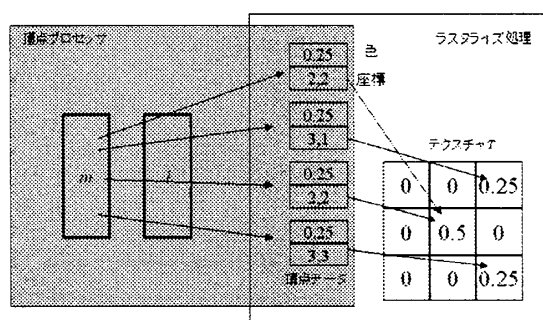


図2: 投票処理

3.2 投票処理の実現方法

投票処理は GPU の頂点プロセッサと固定機能のラスライズ処理を利用して行う。

巨大な三次元配列を GPU で直接扱うことはできないので、投票空間は二次元のテクスチャ上に展開する。例えば $10 \times 10 \times 10$ の三次元配列を用意したい場合は、 10×100 の大きさのテクスチャ T を用意すれば良い。

実際の投票は頂点プロセッサの入力にリスト l と m を与え、投票先座標を格納した頂点データを生成させる。このとき、頂点データの色には $1/|l||m|$ を指定しておく。生成された頂点データは固定機能のラスライズ処理によって投票空間テクスチャ T にラスライズされる。色の合成方法を加算合成として設定しておくことで、すでに描画済みのピクセルに描画される場合も、ピクセル色の値は上書きされずに加算される。この処理が行われる様子を図2に示す。

3.3 利点

提案手法は Strzodka らの手法に比べて少ない計算量で照合を行う事ができる。

提案手法では、テンプレート画像と入力画像の照合1回につき、 $|l||m|$ 回の投票処理が必要となる。一方、Strzodka らの手法において、式(1)で $|l|$ 回の投票処理が纏めて行われていると考えられる。即ち、Strzodka らの手法では投票処理を $WHD|l|$ 回行っている。 $|m| \leq WH$ なので、 $|l||m| \leq WHD|l|$ は明らかである。

4 実験結果

Strzodka らの手法と提案手法を実装し、 32×32 の大きさの入力画像約 900 種とテンプレート約 300 種との照合を行った。テンプレート画像は文字フォントから作成し、入力画像はテンプレート画像を任意の角度だけ回転させて作成した。与えられた入力画像に対し、その画像の元となったテンプレート画像との一致率が他のテンプレート画像との一致率に比べて最も高かった場合を検出成功であるとして、検出精度を算出した。

表1は、それぞれの手法における平均の秒間照合回数と、検出精度である。

表1: 実験結果

	Strzodka らの手法	提案手法
照合回数/秒	7.66	405.68
検出精度 (%)	86.53	83.78

表1からも分かるように、提案手法は大幅な高速化に成功しているといえる。

また、検出精度はどちらの手法も少々低めの結果となった。これは実験に使用した画像が小さく、量子化による誤差が無視できないほど大きくなったためであると考えられる。特に b と q や $<$ と $>$ など、点対象な関係にある文字を誤る例が多く見られた。

5 問題点と今後の課題

本研究では、一般化 Hough 変換を GPU 上で実装するための手法を提案した。

本研究で提案した手法には、次のような2つの問題点がある。まず、本来の一般化 Hough 変換は拡大縮小を考慮するアルゴリズムである事である。今回提案した手法では拡大縮小に対応する事ができない。2つ目の問題は、扱える入力画像は小さなものに限られる事である。これは GPU が利用できるテクスチャのサイズには限りがあるために起きる問題である。実験において 32×32 の画像を用いたのは、このためである。

より柔軟な一般化 Hough 変換を GPU 上で効率良く実装するため、これらの問題に対応する事が必要である。

参考文献

- [1] GPGPU
<http://www.gpgpu.org/>
- [2] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognit., vol.13, no.2, pp.111-122, 1981.
- [3] Robert Strzodka, Ivo Ihrke, Marcus Magnor, "A Graphics Hardware Implementation of the Generalized Hough Transform for fast Object Recognition, Scale, and 3D Pose Detection," International Conference on Image Analysis and Processing (ICIAP 2003), pp.188-193, 2003.