

片側ラドン変換ビジョンアルゴリズムのFPGA実装

平井慎一 座光寺正和 坪井辰彦 (立命館大学)

Implementation of Vision Algorithm based on One-sided Radon Transform on FPGA's

*Shinichi HIRAI, Masakazu ZAKOUJI, and Tatsuhiko TSUBOI (Ritsumeikan University)

Abstract— In this paper we will implement one-sided Radon transform on FPGA's. One-sided Radon transform requires much computation time. It is necessary to process within 33ms for real-time computation. Here, we will construct a parallel computation of the one-sided Radon transform and will implement the one-sided Radon transform on FPGA's.

Key Words: vision, realtime, Radon transform, FPGA's, parallel processing

1 はじめに

リアルタイムビジョンを実現しようとする場合、入力信号がビデオ信号であれば、ビデオフレームレート(33ms)以内に画像処理を完了する必要がある。一般に画像処理アルゴリズムの性能と演算量はトレードオフの関係にあるため、PCによる処理では高性能なビジョンアルゴリズムのリアルタイム性を実現することは困難である。これは、大量の画素に対し、同一の処理を繰り返すことに起因する。一方、処理を並列に行うことができれば、高性能なアルゴリズムのリアルタイム処理が実現できると考えられる。柔軟性を考慮すると、FPGAを用いて並列処理を実現することが考えられる。本研究では、片側ラドン変換を使った物体認識アルゴリズム [1] をFPGAに実装し、リアルタイムビジョンシステムを構築することを目標としている。本報告では、片側ラドン変換の実装について述べる。

2 FPGA ビジョン

2.1 FPGA ビジョンシステム

本研究で使用しているFPGAビジョンシステムをFig. 1に示す。このビジョンシステムは、Xilinx社のVertexE2000を搭載したFPGAボードに、ビデオデコーダ、ビデオエンコーダ、SRAM、PCとの通信用に評価用FPGAボードを接続してある。入力ビデオ信号は順次A/D変換され中心のFPGAボードに送られるので、回路をFPGAに実装すれば、FPGAによる画像処理が実現できる。なお、すべての回路は入力信号に同期した24MHzのクロックで動作させている。

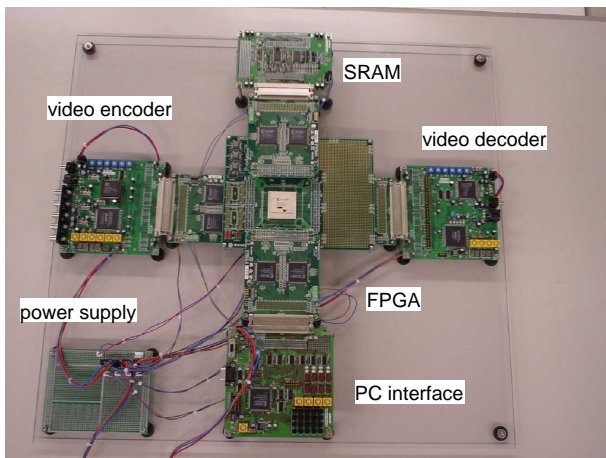


Fig. 1: FPGA-based vision system

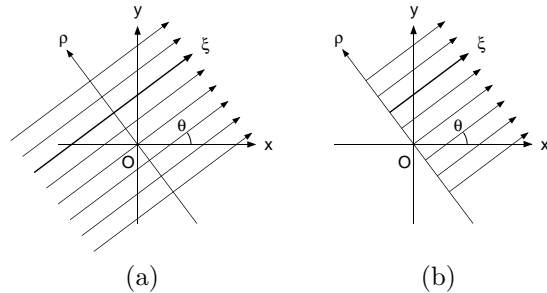


Fig. 2: Radon and one-sided Radon transforms

2.2 設計環境

FPGAへ実装する回路は、HDLで設計を行うことが多い。しかしながら、ビジョンシステムの開発では、設計と評価を繰り返すことが多く、また回路規模が大きいため、論理合成に時間を要する。そこで、C++言語を用いた設計を導入する。本研究では、C++で記述したソースをHDLへ変換するソフトであるSystemCompilerを使用している。このC++言語ベースの設計のメリットとして、高速なシミュレーションが可能であることが挙げられる。画像処理は計算量が多いため、C++レベルでシミュレーションが可能である点は、シミュレーション時間の短縮とテストパターンを容易に生成できるため、特に有効である。

3 片側ラドン変換

本節では、片側ラドン変換を用いた画像処理を簡単に説明する。Fig. 2に示すように、片側ラドン変換はラドン変換の積分路を $[0, \infty]$ に変更したものである。入力画像を $g(x, y)$ とすると、片側ラドン変換 $R(\rho, \theta)$ は次式で与えられる。

$$R(\rho, \theta) = \int_0^{\infty} g(\xi \cos \theta - \rho \sin \theta, \xi \sin \theta + \rho \cos \theta) d\xi. \quad (1)$$

ここで、 θ は x 軸からの角度であり、 ρ は原点からの距離である。すなわち、片側ラドン変換では、 ρ と θ で決まる半直線上で画素値を積分する。この片側ラドン変換の変換結果を用いるビジョンアルゴリズムは、物体の位置・姿勢をロバストに検出可能である。しかし、現在この変換に要する時間は、サイズ 256×256 [pixel]の画像に対して、1.7GHzのCPUで約1400msであり、リアルタイム処理は実現されていない。一方、片側ラドン変換は、各画素に対し独立に行われるため並列性が高い。そこで、片側ラドン変換をFPGAに実装することにより、処理の高速化を図る。

4 片側ラドン変換モジュールの設計

片側ラドン変換モジュールの設計とは、(1) 式の計算を行う回路を設計することである。直接 (1) 式を用いて計算すると、入力画像の画素へのアクセスがランダムになり、結果としてアクセス時間が増加する、画像のキャプチャと並行に処理が行うことができないという問題が生じる。そこで、並列処理に適するように、(1) 式を変形した次式を用いて、等価な変換を行う [2]。

$$\begin{aligned}\rho &= y \cos \theta - x \sin \theta, \\ \xi &= x \cos \theta + y \sin \theta.\end{aligned}\quad (2)$$

(2) 式を用いて x, y, θ の値から対応する ρ, ξ の値を計算する。計算した ξ の値が正または 0 ならば、片側ラドン変換 $R(\rho, \theta)$ の値を画素値 $g(x, y)$ だけ増加させる。本報告では、角度 θ を定数としてモジュールを設計した。(2) 式を用いて ρ, ξ を計算するモジュールを、 θ の分解能に応じて並列に配置することにより、片側ラドン変換の並列計算を実現できる。(2) 式の計算は組み合わせ回路として設計した。各モジュールにはそれぞれ一次元のメモリ $R(\rho, \theta_k)$ が対応する。以上の設計を Fig. 3 に示す。ここで、 g_{out} は増加させる値を表す。すなわち、条件 $\xi \geq 0$ が成り立つときは $g_{out} = g(x, y)$ 、 $\xi < 0$ のときは $g_{out} = 0$ である。

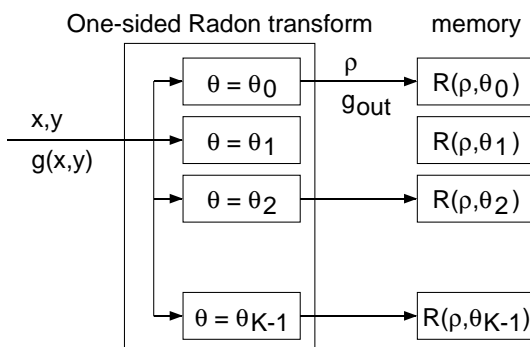


Fig. 3: One-sided Radon transform module

5 評価

5.1 シミュレーション

設計した片側ラドン変換モジュールを使って PC 上でシミュレーションを行う。設計したモジュールの演算はすべて固定小数点による演算である。本研究の最終的な目的はこの変換結果を使った物体認識アルゴリズムを FPGA に実装することであり、変換結果の精度は物体認識に要する精度で決定される。したがって、必要な精度が保証し、かつ回路規模を押えるためには、ビット幅をあらかじめ求めておくことが望まれる。

そこで、通常の変換を行った場合と、今回設計した片側ラドン変換モジュールを使って変換を行った場合を比較する。Fig. 4-(a) に示すサイズ 256×256 [pixel] の画像に対して、片側ラドン変換を求め比較する。モジュールの入力信号 $x, y, g(x, y)$ のビット幅は、それぞれ 8 ビットである。これを考慮し、三角関数の計算に対しては、8 ビットと符号ビットの合計 9 ビットの精度でシミュレーションを行った。それぞれの変換結果を、Fig. 4-(b),(c) に示す。結果を見ると、後の物体認識に必要な特徴がしっかり表れており、設計したモジュールによる変換が十分な精度であることが確認できた。

また、タイミングシミュレーションの結果は最大動作周波数が約 40MHz となった。これより処理時間を計算すると、理論上 1.6ms ほどで処理が可能である。

5.2 FPGA 実装

設計した片側ラドン変換モジュールを FPGA に実装し、ビデオフレームレートで動作させる。まず始めに

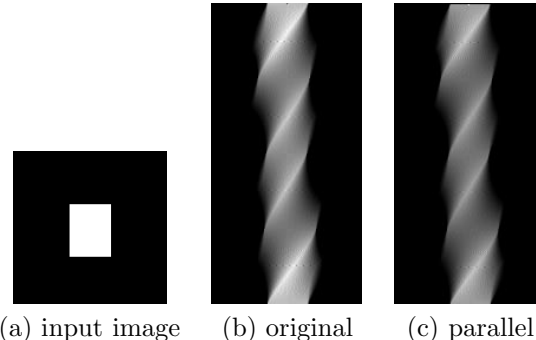


Fig. 4: Simulation result

動作確認のため $\theta = 0$ に対応するモジュールのみを実装し動作させた。その結果を、Fig. 5 に示す。画面中央部の 256×256 [pixel] の領域に対して、片側ラドン変換 $R(\rho, 0)$ を計算する。片側ラドン変換 $R(\rho, 0)$ の値を、画面中央下部に提示する。片側ラドン変換の値は、画素の濃淡で表している。変換結果を画面に出力した結果は、シミュレーションによる結果と同じであり、計算回路が正確に動作していることが示された。また、動作は 24MHz のクロックで行っているため、変換時間は 2.7ms ほどであり、PC での処理に比べ大幅な処理速度の向上を実現できた。角度分解能は、FPGA のゲート数、回路規模や最大遅延等を考慮して決めていく必要がある。

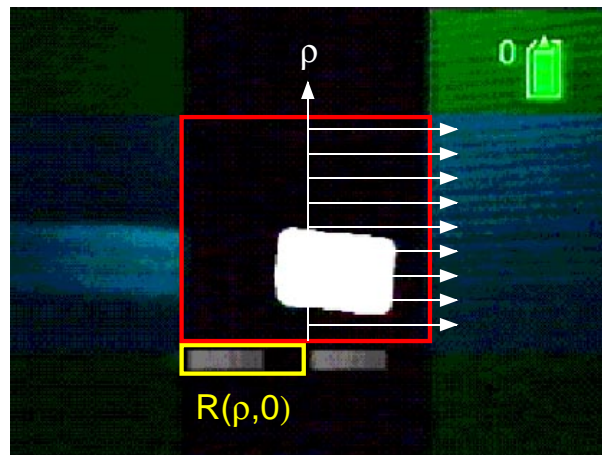


Fig. 5: Output of computed transform

6 おわりに

片側ラドン変換を FPGA に実装することで、大幅な計算時間の向上が達成された。今後は、片側ラドン変換を基にしたビジョンアルゴリズムを完成させ、リアルタイムで位置と姿勢を検出するビジョンシステムを構築する。また、位相限定相関法や一般化ハフ変換法等の実装を試みる。

参考文献

- [1] Tsuboi, T., Masubuchi, A., Hirai, S., Yamamoto, S., Ohnishi, K., and Arimoto, S., "Video-frame Rate Detection of Position and Orientation of Planar Motion Objects using One-sided Radon Transform", Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 2, pp.1233-1238, Seoul, May, 2001
- [2] 座光寺, 平井, "リアルタイムビジョンのための片側ラドン変換法の並列化", 第 19 回日本ロボット学会学術講演会予稿集, pp.301-302, 2001