

GPU 上での Specmurt 分析を用いた基本周波数解析の並列処理

Parallel Processing of Fundamental Frequency Analysis Using Specmurt on GPU

高橋 太士†
Futoshi Takahashi吉田 明正†
Akimasa Yoshida

1 はじめに

近年、デジタル音響信号から基本周波数を抽出する方法としてウェーブレット変換 [1] と Specmurt 分析 [2] を用いた解析手法や、短時間フーリエ変換とハーモニッククラスタリング [3, 4] を用いる解析手法が注目されている。デジタル音響信号から基本周波数解析を行うことは、自動採譜の技術において必要不可欠なものである。本稿では、GPU を使用したウェーブレット変換と Specmurt 分析による基本周波数解析の並列処理手法を提案する。本研究ではウェーブレット変換と Specmurt 分析を用いる手法を取り上げる。Specmurt 分析による解析には多くの時間がかかるため、高速化が望まれている。本手法では、GPU 上で並列処理を行い処理時間の短縮を実現する。具体的には、ウェーブレット変換でピッチ抽出を行い、その後、Specmurt 分析で基本周波数を解析する。GPU 上での並列プログラムは、CUDA [5, 6] を用いて開発しており、NVIDIA Tesla C1060 上での性能評価の結果から有効性を確認した。

2 基本周波数解析によるシステム

本システムの構成を図 1 に示す。まず、入力音響信号をウェーブレット変換 [1] により周波数変換する。次に、得られた周波数成分に対し Specmurt 分析 [2] による基本周波数の抽出を行う。計算量の多い箇所であるウェーブレット解析と Specmurt 分析による解析を行っている部分には、GPU 上で並列処理を行い高速化を実現する。

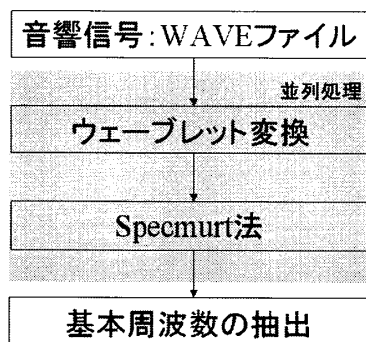


図 1 基本周波数解析の手順.

3 Specmurt 分析による基本周波数の抽出

本研究では、音響信号から周波数を抽出するにあたって Gabor ウェーブレットを利用した連続ウェーブレット変換を行った。以下に、Gabor ウェーブレットと連続ウェーブレット変換の計算式を示す。

$$\psi(t) = \frac{1}{2\sqrt{\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} e^{-it} \quad (1)$$

$$W(a, b) = \langle f(x), \psi_{a,b}(x) \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(x) \psi\left(\frac{x-b}{a}\right) dt \quad (2)$$

また、音響信号は音高に相当する基本周波数以外に倍音と呼ばれる周波数を含んでいる。倍音とは、基本周波数の整数倍である周波数のことである。必要となる音高の情報を得るためには倍音のパワースペクトルを抑制する必要がある。周波数軸を f とし、 $x = \log f$ に変換すると、基本周波数と倍音の距離は一定となる。この時、基本周波数と倍音の距離は基本周波数に依存しなくなる。そして、単一音において倍音の強度比のパターンは一定であると仮定した時、これを共通調波構造と呼び、対数基本周波数を原点にとって $h(x)$ と表すことができる。基本周波数の強度を表す分布を $u(x)$ とした時、多重音の対数周波数スペクトル $v(x)$ はこれらの畳み込み

$$v(x) = f(x) * h(x) \quad (3)$$

として表現する事ができる。これに基づき、基本周波数 $u(x)$ を多重音スペクトル $v(x)$ に対する対数周波数での逆畳み込み

$$u(x) = F \left[\frac{F^{-1}[v(x)]}{F^{-1}[h(x)]} \right] \quad (4)$$

で求める手法を Specmurt 分析と呼ぶ。

4 GPU 上での CUDA による並列処理

GPU (Graphics Processing Unit) は主に 3D グラフィックスの処理を行う半導体チップである。近年 GPU は 3D グラフィックスの処理のみならず、一般的な計算処理にも利用されている。GPU を用いて計算処理を行う事を GPGPU (General-Purpose computing on GPU) と呼ぶ [5]。GPGPU に特化したグラフィックボードとして、本研究では Tesla C1060 を用いて、CUDA 開発環境を用いて、ウェーブレット変換と Specmurt 分析の並列プログラムを開発した。

4.1 NVIDIA Tesla の構造

Tesla C1060 は 8 個の SP (Streaming Processor) からなる MP (Multi Processor) が 30 個搭載された GPU である [6]。SP とは最小単位の演算処理ユニットである。Tesla C1060 に搭載されている SP の総数は 240 である。また、Tesla C1060 のクロック周波数は 1.60GHz で動作しており、GPU 上のデバイスメモリは 4 GB が搭載されている。

4.2 CUDA プログラミング

GPU 上の並列プログラムは、CUDA プログラミングにより実現するのが一般的である。CUDA のプログラ

† 東邦大学理学部情報科学科

Department of Information Science, Toho University

ム構成は、CPU を動作させるホストコードと、GPU を動作させるデバイスコードから構成されている。ホストコードは GPU 上のメモリの確保やカーネルと呼ばれる単位でプログラムの実行を操作でき、デバイスコードにはカーネル関数が記述されている。また、各 SP で実行する GPU は多数のスレッドを並列実行する構造となっている。CUDA では、スレッドのまとまりをブロックと呼ぶ単位で管理することができる。そして、同じサイズのブロックをまとめたものをグリッドと呼び、グリッドは 2 次元ブロックを含むことができる。また、グリッド内の全スレッドは、同一のプログラム (カーネル関数) を実行する。

4.3 ウェーブレット変換の並列処理

本研究ではまず、入力されたデジタル音響信号 (wave ファイル) に対して、連続ウェーブレット変換を行う。ウェーブレット変換の並列処理の手順は、以下の通りである。

1. CPU のホストメモリに保存されている入力音響信号 (wave ファイル) をデバイスメモリに転送する。
2. GPU 上で、連続ウェーブレット変換の計算に使用する Gabor ウェーブレットの値を求める。求める周波数によって Gabor ウェーブレットの値は異なるので、9600 個分の Gabor ウェーブレットをそれぞれスレッドとして並列化して計算を行う。
3. GPU 上で、連続ウェーブレット (式 (2)) の計算を行う。
4. 計算によって得られた周波数分布をホストメモリに転送する。

4.4 Specmurt 分析の並列処理

本研究では、各周波数の処理単位で並列化を行った。GPU の性能を最大限に利用するために、21.5Hz~22050Hz の周波数の中から 9600 個の周波数の値を選び、それぞれの周波数の処理を GPU 上で並列処理した。なお、共通調波構造 $h(x)$ は、ピアノの C4~B4 の 12 音の平均値により事前に求めている。以下に Specmurt 分析の並列処理の手順を示す。

1. CPU のホストメモリに保存されている多重音の音響信号 $v(x)$ 、共通調波構造 $h(x)$ をデバイスメモリに転送する。 $v(x), h(x)$ の要素数は 9600 である。
2. GPU 上で $v(x), h(x)$ に対して逆フーリエ変換を行い、 $F^{-1}v(x), F^{-1}h(x)$ とする。また、GPU 上で行う並列処理は、各周波数の値の処理ごとに行う。
3. 逆フーリエ変換を行った $F^{-1}v(x), F^{-1}h(x)$ に対して除算を行い、波形領域上の基本周波数分布を求める。波形領域上の $F^{-1}(x), F^{-1}(x)$ の要素数は 19200 となる。また、ウェーブレットとの処理時間の比較を行うため、1 スレッドに対し 2 つの周波数の計算を行った。
4. GPU 上で波形領域上の基本周波数分布に対してフーリエ変換を行い、フーリエ領域上の基本周波数分布 $u(x)$ を求める。
5. $u(x)$ を CPU のホストメモリに転送する。

5 Tesla C1060 上での性能評価

本性能評価では Tesla C1060 を使用した。Tesla C1060 の性能を最大限に利用するためにブロック数とスレッド数を変化させ、ウェーブレット変換と Specmurt 分析の

実行時間を測定した。GPU 上で動作するスレッド数を最大限にするためにブロック数を 30 の倍数にし、スレッド数は Warp(1SP 当たり 4 スレッド) の制約を考慮して 32 の倍数とした。また、使用した入力信号はサンプリング周波数 44.1kHz、量子化ビット数 16、モノラル、5 秒間のピアノの和音の音響信号 (C4-E4-G4) を用いた。

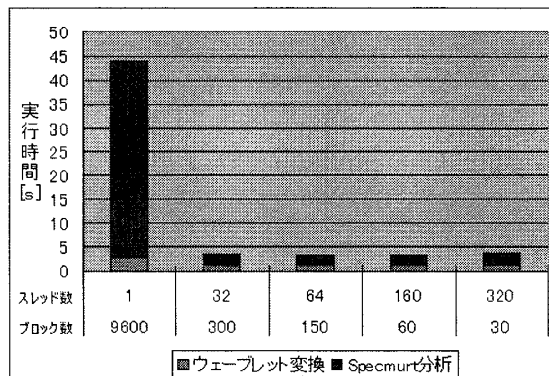


図 2 GPU における Specmurt 分析の並列処理。

実行結果は図 2 の通り、150 ブロック、64 スレッドの時の実行結果が最も速くなった。この場合は、MP を有効利用しており、ブロック数 9600 個、スレッド数 1 個の時と比べて、ウェーブレット変換の処理においては 2.32 倍、Specmurt 分析においては 20.09 倍、総合的に 13.37 倍 (理論値 32 倍) の速度向上が得られた。また、CPU (Core i7 3.20GHz) の 1 コアの実行速度との比較を行った結果、7.37 倍の速度向上を得ることができた。

6 おわりに

本稿では、GPU を使用したウェーブレット変換と Specmurt 分析による基本周波数解析の並列処理手法を提案した。本手法では、フレーム内の並列性を利用して GPU 上で並列処理を行ったことにより、Specmurt 分析の大幅な処理時間短縮を実現した。

今後の課題としては、GPU 上のメモリを有効利用できるように、並列プログラムを最適化することがあげられる。

参考文献

- [1] 榎原進. ウェーブレットビギナーズガイド, 東京電機大学出版局, 2003.
- [2] 高橋圭吾, 西本卓也, 嵯峨山茂樹. 対数周波数逆畳み込みによる多重音の基本周波数解析, 情報処理学会研究報告, 2003-MUS-53, pp.61-66, 2003.
- [3] 亀岡弘和, 西本卓也, 嵯峨山茂樹. ハーモニッククラスタリングによる多重音信号音高抽出における音源数とオクターブ位置推定, 情報処理学会研究報告, 2003-MUS-51-5, pp.29-34, 2003.
- [4] 鈴木涼介, 吉田明正. マルチコアプロセッサ上でのハーモニッククラスタリングを用いた基本周波数解析の並列処理, 情報処理学会第 71 回全国大会, 2D-7, 2009.
- [5] 青木尊之, 額田彰. はじめての CUDA プログラミング, 工学社, 2009.
- [6] NVIDIA CUDA Compute Unified Device Architecture プログラミング・ガイド, 2009.