

GPU 上での一般化調和解析を用いたピッチ抽出の並列処理

Parallel Processing of Pitch Extraction Using Generalized Harmonic Analysis on GPU

岡村 瑞穂 †
Mizuho Okamura吉田 明正 †
Akimasa Yoshida

1 はじめに

本稿では、単一音源のデジタル音響信号に対して GPU 上で一般化調和解析 [1, 2, 3, 4] を行うことにより、精度の高いピッチ抽出を高速に行う方法を提案する。

一般化調和解析は、ピッチ抽出手法の一つとして用いられ、関数という概念を必要とすることなく高い精度で解析できるが解析に多くの時間を要するのが難点である。本稿では、入力した音響信号を一定時間で分割した標本集合 (フレーム) に対して、フレーム内の並列性とフレーム間の並列性を利用したマルチレベル並列処理を、GPU 上で CUDA [5] により実装した。本性能評価では、デジタル音響信号を入力として、NVIDIA Tesla S2050 を用いて並列実行し有効性を確認した。

2 ピッチ抽出システムの構成

本システムの構成を図 1 に示す。まず、入力音響信号を読み込み、そのデータを短時間毎にフレーム分割する。次に、各フレームに対して一般化調和解析を行い、解析結果を出力する。一般化調和解析は GPU 上で行い高速化を実現する。

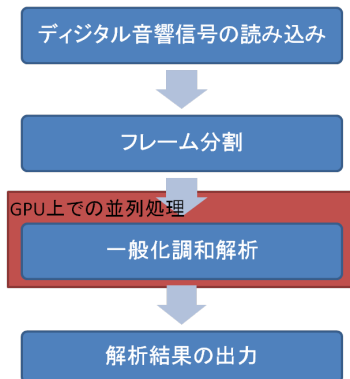


図 1 ピッチ抽出の手順。

3 一般化調和解析によるピッチ抽出

本研究では、音響信号からピッチを抽出するにあたって一般化調和解析を行った。以下に、一般化調和解析の計算式を示す。観測区間長 $L(t)$ で観測された連続信号 $x(t)$ があるとする。 $x_n(t) = x(t)$ として次の処理を行う。ここで n の初期値を 0 とする。連続信号 $x_n(t)$ に対し、周波数 f [Hz] のフーリエ係数を計算する。ここで周期 T は $1/f$ である。また、 f の初期値を 1 とする。

$$S(f) = \frac{2z s(z + \sin z) - c(1 - \cos z)}{L z^2 - 2(1 - \cos z)} \quad (1)$$

$$C(f) = \frac{2z c(z - \sin z) - s(1 - \cos z)}{L z^2 - 2(1 - \cos z)} \quad (2)$$

ただし、

$$z = 4\pi fL, \quad s = \int_0^L x_n \sin 2\pi ftdt, \quad c = \int_0^L x_n \cos 2\pi ftdt$$

(1),(2) で導き出されたフーリエ係数をもとに予測波形 $W(t, f)$ を求める。

$$W(t, f) = S(f) \sin 2\pi ft + C(f) \cos 2\pi ft \quad (3)$$

(4) より、残差波形 $\epsilon(t, f)$ 、残差エネルギー $E(f)$ を求める。

$$\epsilon(t, f) = x_n(t) - W(t, f) \quad (4)$$

$$E(f) = \int_0^L \epsilon(t, f)^2 dt \quad (5)$$

$f=f+1$ として (1) ~ (5) を任意の回数繰り返す、 E を最小とする f を f_n とする。この周波数 f_n [Hz] を $x(t)$ を構成する一音として抽出する。

$$x_{n+1}(t) = x_n(t) - W(t, f_n) \quad (6)$$

最後に、 $x_n(t)$ から $W(t, f_n)$ を引いて次の入力信号 $x_{n+1}(t)$ とし、(1) ~ (6) を任意の回数繰り返すことで一音ずつ取り出す。

4 GPU 上での CUDA による並列処理

GPU は本来グラフィックス処理を目的として設計されたユニットであるが、そのプログラミング環境である CUDA を用いることでその高い演算性能を汎用計算に利用することができ、これを GPGPU という。GPGPU に特化したグラフィックボードとして、本研究では Tesla S2050 を用いて、一般化調和解析の並列プログラムを開発した。GPU の内部は、複数の SP (Streaming Processor) からなる SM (Streaming Multi Processor) が複数搭載されている。SP はスレッド、SM はブロックという処理単位で階層的に並列処理が行われる。

4.1 NVIDIA Tesla S2050 のアーキテクチャ

Tesla S2050 は 32 個の SP からなる SM が 14 個搭載された GPU である。よって、Tesla S2050 に搭載されている SP の総数 (CUDA Core) は 448 である。また、Tesla S2050 のクロック周波数は 1.15GHz で動作しており、2.8GB のデバイス・メモリが搭載されている。

† 東邦大学理学部情報科学科

Department of Information Science, Toho University

4.2 一般化調和解析の並列処理

本研究では、以下に一般化調和解析の並列処理の手順を示す。

1. 入力音響信号を読み込み、短時間毎にフレーム分割する。
2. デバイス・メモリを確保し、CPU 側で作成した計算に必要なデータを引き渡す。
3. GPU 上で一般化調和解析を行う。各フレームに対し以下のステップを任意の回数繰り返す。
 - (a) 入力信号に対しての残差エネルギーを解析周波数分求める。
 - (b) 各周波数の残差エネルギーを比較し、最小となるときの周波数を抽出する。
 - (c) 入力信号からその周波数からなる波形を減算し次の入力信号とする。
4. 解析結果を CPU のメモリに転送する。

GPU 内での処理は、フレーム内の並列処理をスレッド単位、フレーム間の並列処理をブロック単位で行うことで、解析をマルチレベル並列処理で行った。そのモデルを図 2 に示す。

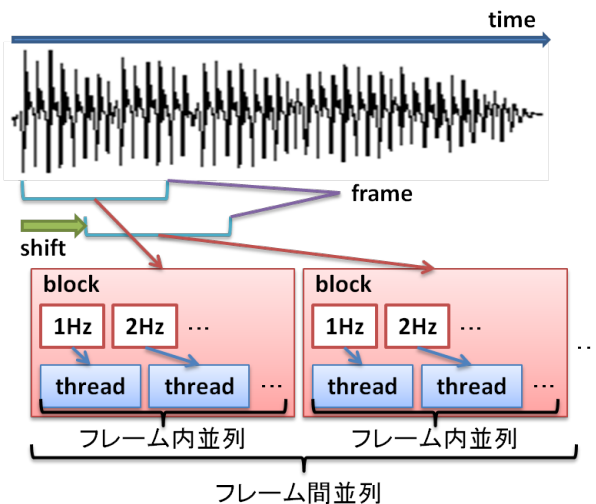


図 2 マルチレベル並列処理のモデル。

5 Tesla S2050 上での性能評価

本研究では、単一音源の音響信号として、WAVE 形式のファイルを用いる。対象の信号は、サンプリング周波数 44,100Hz、量子化ビット数 16bit、モノラル音源とした。各パラメータは解析フレームを約 93ms (標本点数 4096 点)、解析フレームのシフト数を約 3ms (標本点数 128 点) とした。一般化調和解析の基本周波数とみなす周波数を 1~22050Hz とした。これは解析に使用する音源の演奏音域を十分にカバーしている。

今回、性能評価に使用した入力信号は産業技術総合研究所の RWC 研究用音楽データベースにある「3 つのジムノペディ 第 3 番 (サティ,E.)」の冒頭部分である。開始 0.99 秒を第 1 フレームの開始位置とした。これは、C4,E4,A5 の和音が鳴り始めた直後にあたり、その前に鳴った A3 と合わせて 4 和音となっている。よって、GPU では 4 音の解析を行った。

実行時間の測定は、GPU 上での一般化調和解析の並列処理を対象とした。

表 1 1SM 上での 1 フレームの並列処理の結果

フレーム数	SM 数	ブロック数	SP 数	スレッド数	時間 [s]
1	1	1	1	1	915.80
			32	240	4.18
				480	2.90
				960	2.61

まず、解析フレーム数およびブロック数を 1 とし、スレッド数を変化させて実行した。表 1 はその実行結果である。SP 数が 32 の場合は、1SP と比べると演算資源が増えたにより圧倒的に速くなっている。960 スレッドでは 1 スレッドの 450 倍の高速化が達成された。スレッド数が 240 以降は演算資源は変わらないが並列処理性能が上がった。つまり、ブロック内のスレッド数が多いほどフレーム内の並列性が GPU 上で有効利用され、実行速度が向上することがわかった。

表 2 1SM 上でのフレーム内並列処理の結果

フレーム数	SM 数	ブロック数	SP 数	スレッド数	時間 [s]
14	1	1	32	960	36.53
28					73.06
56					146.02
112					292.16

表 3 14SM 上でのフレーム内・間を並列処理の結果

フレーム数	SM 数	ブロック数	SP 数	スレッド数	時間 [s]
14	14	14	32	960	2.61
28		28			5.20
56		56			10.37
112		112			20.73

次に、1SM あたりのスレッド数を 960 に固定し、複数フレームを逐次処理 (1SM) した場合と並列処理 (14SM) した場合に分けて実行した。その実行結果を表 2 と表 3 に示す。表 2 の逐次処理の場合は、表 1 における 960 スレッドの実行時間 2.61[s] のフレーム数倍の時間を要したが、表 3 の並列処理の場合は、(フレーム数/SM 数) 倍つまり、1SM あたりの処理回数倍で処理することができた。例えば、表 2 と表 3 の 14 フレームを比較すると、フレーム間の並列処理により、14 倍の高速化が達成された。

6 おわりに

本稿では、一般化調和解析によるピッチ抽出を GPU 上でマルチレベルに並列処理する手法を提案した。GPU が持つスレッド、ブロックというマルチレベル並列性を利用してフレーム内、フレーム間ともに並列処理することで実行速度の向上が見られた。

今後の展開として、グラフィックスボードに搭載されている複数の GPU を用いた 3 階層での並列処理が挙げられ、更なる速度向上が期待できる。

参考文献

- [1] 山崎芳男. 音響信号の時間周波数分析, 日本音響学会誌 53 巻 2 号, pp.147-153, 1997.
- [2] 安部素嗣. 一般化調和解析の周波数領域での振舞い, 信学技報, pp.13-18, 2005.
- [3] 中沢誠. 聴覚と信号の特徴に着目した音響信号の一般化調和解析, 早稲田大学大学院国際情報通信研究科, 2004.
- [4] Hisayori Noda, Akinori Nishihara. Fast and Accurate Generalized Harmonic Analysis and Its Parallel Computation by GPU, IEICE Trans.Fundamentals,VOL.E92-A,NO.3 MARCH 2009, pp.745-752, 2009.
- [5] Jason Sanders, Edward Kandrot. CUDA by Example 汎用 GPU プログラミング入門, インプレスジャパン, 2011.