

# GPU上での一般化調和解析・ハーモニッククラスタリングによる音高推定の並列処理

藁谷 拓也<sup>1</sup> 岡村 瑞穂<sup>1</sup> 山内 長承<sup>1</sup> 吉田 明正<sup>2</sup>

概要：本稿では、デジタル音響信号に対する一般化調和解析とハーモニッククラスタリングによる基本周波数解析システムの GPU 上での並列処理手法を提案する。多重音の音響信号から音響の特徴量を推定するためには、高い性能の多重音解析システムの開発が必要である。また、近年マルチコア及び GPU を搭載した計算機システムの普及により、演算処理能力は飛躍的に向上した。並列処理は解析精度の高いシステムを実用化させるために必要不可欠になる。本研究で用いる一般化調和解析は周波数分解能と時間分解能の両立が可能である。この一般化調和解析の結果に対して、ハーモニッククラスタリングを適用し、音高推定を行う。本稿では、NVIDIA Tesla S2050 上で CUDA を用いて音高推定システムの並列処理を実現し、ショパン等のピアノ曲を用いて性能評価を行ったところ、1CUDA コアと比べて 2933 倍の速度向上が得られた。

キーワード：GPGPU、一般化調和解析、ハーモニッククラスタリング、音高推定、並列処理

## 1. はじめに

近年、デジタル音響信号に対する自動採譜のための技術が盛んに研究されている。自動採譜には、ピッチ抽出や音高推定、同時発音数推定など様々な技術が必要とされる [1]。周波数スペクトル解析には高速フーリエ変換 (FFT)、DCT、Wavelet 変換などの手法があるが、これらの手法には不確定性原理を考慮に入れなくてはならないという問題が存在する。

一方、一般化調和解析 [2] では周波数分解能を任意に選択することが可能である。基本周波数解析に関しては、ハーモニッククラスタリング [3], [4] や Specmurt 法 [5] が提案されているが、ハーモニッククラスタリングは、倍音比の情報を必要としない特徴がある。この方法は、基本周波数とそれぞれの倍音周波数を中心とするクラスタ帯域、クラスタ重心を決定し、それぞれのクラスタに倍音構造としての拘束を与えてクラスタリングを行うものである。

本稿では、精度の高い音高推定を実現するために、一般化調和解析とハーモニッククラスタリングを組み合わせた音高推定システムを開発し、GPU 上での並列処理手法を提案する。基本周波数解析システムの並列化においては、NVIDIA Tesla S2050 上で CUDA を用いて実装しており、

複数の GPU を用いて並列処理による性能評価を行った。

本稿の構成は、2 章で一般化調和解析を述べ、3 章でハーモニッククラスタリングを述べる。4 章では GPU 上の音高推定システムの並列処理について述べ、5 章で GPU 上で CUDA により実装した音高推定システムの性能評価を行い、6 章でまとめを述べる。

## 2. 一般化調和解析

本章では、音高推定システムの周波数解析に使用した一般化調和解析 [2] について述べる。

### 2.1 一般化調和解析の概要

一般化調和解析 (GHA: Generalized Harmonic Analysis) は N.Wiener が提案した周波数解析法で、観測区間内において原波形から残差が最小となる正弦波を逐次減算処理するという単純明快なアルゴリズムである。後に平田らによってピアノの音などの非定常信号の解析に導入され [6]、大内らによって雑音と信号の分離が行われる [7] などの研究がされてきた。

楽音を含めて多くの音響信号は厳密には周期関数ではなく、およそ周期関数に近い構造を持つ概周期関数である。ゆえに、周期関数を対象としている FFT に比べ被分析信号を非調和な周波数成分よりなる概周期関数で表す一般化調和解析は音響信号を解析するのに適していると言える。

<sup>1</sup> 東邦大学理学部情報科学科

<sup>2</sup> 明治大学総合数理学部ネットワークデザイン学科

また、FFT を用いて解析をする際に問題となる不確定性原理の制約がなく、時間分解能と周波数分解能を任意に設定できるのが一般化調和解析の強みである。

しかし、一般化調和解析は膨大な計算量を必要とし、その解析時間の長さから広く使われることはなかった。現在では計算機性能の向上とともに、ニュートン法を用いたアルゴリズムが導入されるなどの計算量を減らす研究が行われ [8]、以前よりも実用的な解析方法になっている。

## 2.2 一般化調和解析の計算アルゴリズム

以下に一般化調和解析の計算式を示す。観測区間長  $L(t)$  で観測された信号  $x_0(t)$  に対し  $x_n(t)=x_0(t)$  として次の処理を行う。(1) 式より信号  $x_n(t)$  に対し、任意の周波数  $f[\text{Hz}]$  の予測波形を求める。

$$W(t, f) = S(f) \sin 2\pi ft + C(f) \cos 2\pi ft \quad (1)$$

振幅  $S(f), C(f)$  は (2) 式, (3) 式で示される。また,  $S(f), C(f)$  によって位相情報も保存される。

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

$$C(f) = \frac{2z}{L} \cdot \frac{c(z - \sin z) - s(1 - \cos z)}{z^2 - 2(1 - \cos z)} \quad (3)$$

ただし,  $z, s, c$  は (4) 式で示される。

$$z = 4\pi fL,$$

$$s = \int_0^L x_n(t) \sin 2\pi ftdt, c = \int_0^L x_n(t) \cos 2\pi ftdt \quad (4)$$

信号  $x_n(t)$  から予測波形  $W(t, f)$  を減算し二乗することで残差エネルギー  $E(f)$  を得る。

$$E(f) = \int_0^L \{x_n(t) - W(t, f)\}^2 dt \quad (5)$$

$f$  の値を変え、(1) 式 ~ (5) 式を任意の回数繰り返す。その中で最終的に  $E(f)$  を最小とする  $f$  を信号  $x_0(t)$  に含まれる一音  $f_n$  として抽出する。

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

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

## 3. ハーモニッククラスタリング

楽器音音響信号によるパワースペクトルでは、その楽器音特有の倍音が同時に観測される。このような倍音の構造を調波構造として利用することで、基本周波数を推定する手法の一つにハーモニッククラスタリング [3], [4] がある。

### 3.1 ハーモニッククラスタリングの概要

ハーモニッククラスタリング (Harmonic Clustering) とは、基本周波数とそれぞれの倍音周波数を中心とするクラスタ帯域、クラスタ重心を決定し、それぞれのクラスタに倍音構造としての拘束を与えて自由度 1 のクラスタリングを行うものである。このような拘束を与えられたクラスタ群を倍音クラスタ群と呼ぶ。クラスタリングを行う際は EM アルゴリズムを用いて最尤パラメータを求める。音源数を判定するには、AIC (赤池情報量基準) [9] を導入し最適なモデルの選択基準とする。

### 3.2 FFT を用いたハーモニッククラスタリング

ハーモニッククラスタリングでは、フーリエ変換によって得られたスペクトルを利用している。多重音のハーモニッククラスタリングは複数 ( $K$  個) の倍音クラスタ群を用いて EM アルゴリズムにより解く。その際、倍音クラスタ群  $k$  の  $n$  倍音に対する重心を  $\mu_k + \log n$  とし、クラスタ群  $k$  において上限 (ナイキスト周波数の対数) まで取りえるクラスタ重心数を  $N(k)$  とする。また、クラスタごとの重みを  $w_n^k$  とし、クラスタ帰属度を  $p_n^k(x)$ 、尤度関数を  $\rho(x, \mu_k + \log n)$ 、観測スペクトル密度を  $f(x)$  と定義することにより、クラスタリング評価関数 (7) 式を得る。

$$D_{multi}(\mu_k, w) = \sum_{k=1}^K \sum_{n=1}^{N(k)} \int_{-\infty}^{\infty} w_n^k \cdot \rho(x, \mu_k + \log n) \cdot p_n^k(x) f(x) dx \quad (7)$$

$p, \rho$  は (8) 式, (9) 式で示される。

$$p_n^k(x) = \frac{w_n^k \cdot g(x|\mu_k + \log n, \rho^2)}{\sum_k \sum_n w_n^k \cdot g(x|\mu_k + \log n, \rho^2)} \quad (8)$$

$$\rho(x|\mu_k + \log n, \rho^2) = \log \frac{1}{\sqrt{2\pi\rho^2}} - \frac{\{x - (\mu_k + \log n)\}^2}{2\rho^2} \quad (9)$$

ただし,  $g$  は (10) 式で示される。

$$g(x|\mu_k + \log n, \rho^2) = \frac{1}{\sqrt{2\pi\rho^2}} - \exp\left(-\frac{x - (\mu_k + \log n)^2}{2\rho^2}\right) \quad (10)$$

また、以下のような反復計算により最尤パラメータを求める。

- (1) 初期値の設定初期のクラスタ群の数  $K$  と、重み  $w_n^k$ 、初期基本クラスタ重心  $\mu_k$  を設定する。
- (2)  $p_n^k(x)$  の更新  
 $p_n^k(x)$  を算出し、評価関数を求める。
- (3)  $\mu_k$  と  $w_n^k$  の更新  
パラメータを (11) 式, (12) 式のように更新し、(2) に戻る。

$$\bar{\mu} = \frac{\sum_{n=1}^{N(k)} \int_{-\infty}^{\infty} (x - \log n) \cdot p_n^k(x) \cdot f(x) dx}{\sum_{n=1}^{N(k)} \int_{-\infty}^{\infty} p_n^k(x) \cdot f(x) dx} \quad (11)$$

$$\bar{w}_n^k = \frac{1}{f} \int_{-\infty}^{\infty} p_n^k(x) dx \quad (12)$$

ただし、 $F$  は (13) 式で示される。

$$F = \int_{-\infty}^{\infty} f(x) dx \quad (13)$$

最尤パラメータが求まったら (4) へ進む。

(4) AIC の算出

(5) 倍音クラスタ群数の削減

重みの最も小さい倍音クラスタ群を削除する。また、2つの基本クラスタ重心の距離が  $1/12 \cdot \log 2$  より小さい時、重みの小さいほうの倍音クラスタ群を削除する。

$K=0$  となったとき処理を終了する。残った  $K$  を新たな  $K$  とし、(2) に戻る。

以上の反復計算を  $K=1$  まで行い、最も AIC が低いものを最尤のパラメータとする。

### 3.3 一般化調和解析を用いたハーモニッククラスタリング

ハーモニッククラスタリングはフーリエ変換によって得られるスペクトルの特性を利用したものである。しかし、調波構造を利用したクラスタリングは一般化調和解析によって得られるスペクトルでも利用が可能である。フーリエ変換で得られるスペクトルのデータ数は解析区間長に依存する。また、データ間の周波数距離はサンプリング周波数/解析区間長によって決まる。一方、一般化調和解析ではデータ数は任意でデータ間の周波数距離は可変である。よってハーモニッククラスタリングに用いる (7) 式を以下のように変形した。

$$D_{multi}(\mu_k, w) = \sum_{k=1}^K \sum_{n=1}^{N(k)} \sum_{i=0}^I w_n^k \cdot \rho(x_i, \mu_k + \log n) \cdot p_n^k(x_i) \cdot x_i$$

ここで  $x_i$  は一般化調和解析によって得られた  $i$  番目の周波数のパワーの対数である。一般化調和解析によって得られるスペクトルデータは FFT とは周波数の順番も周波数ごとの距離も各々異なる。しかし、得られたスペクトルデータは同様に調波構造を持つためハーモニッククラスタリングを行うことは可能であると考えられる。

## 4. GPU 上の音高推定システムの並列処理

本章では、GPU 上で音高推定システムの並列処理を行うための処理の流れを示す。

### 4.1 音高推定システム

本システムの構成を図 1 に示す。まず、WAVE データ



図 1 音高解析の手順

を取り込み、その波形をもとにして一般化調和解析を行う。その際のフレーム長とフレームシフトは任意に設定可能である。ここでのフレーム長とは一度に解析するデータの短時間区域であり、フレームシフトは解析区間の開始地点間の距離である。一般化調和解析によって得られた周波数パワースペクトルのデータは、GPU 上のメモリに保持され基本周波数解析に用いられる。

ハーモニッククラスタリングではフレームごとの周波数パワースペクトルデータを用いて、各々のフレームにおける基本周波数推定を行う。すべてのフレームに対して基本周波数解析を終えたら、その周波数ロールデータを CPU 側に転送し、出力し終了する。

WAVE データの GPU 側への受け渡しから、周波数ロールの結果を CPU 側に受け渡すまでが GPU 上での並列処理部分である。なお、本稿で提案する GPU 上での並列処理については 4.3 節で述べる。

### 4.2 CUDA による並列処理コード

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 開発環境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る。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。

CUDA によるプログラミングでは、並列処理を適用する計算処理部分をカーネル関数として定義し、そのカーネル関数をブロック\*スレッド数の個数分だけ実行する。CPU と GPU のメモリ間のデータ転送には、`cudaMemcpy()` を使用する。

### 4.3 CUDA による音高推定システムの並列処理

本システムでは、図 1 に示した通り、一般化調和解析

とハーモニッククラスタリングは GPU 上で計算を行い高速化を実現する。さらに、複数の GPU (マルチ GPU) を用いることにより、一般化調和解析の解析時間の短縮を目指す。

#### 4.3.1 一般化調和解析のフレーム内・フレーム間の並列処理

一般化調和解析はその膨大な計算量ゆえに、ナイキスト周波数全ての残差エネルギーを計算すると解析に多くの時間を消費してしまう。そこで本システムには通常の一般化調和解析のみのモードと、FFT による前処理により大まかな周波数スペクトルを計算し、その中で最も大きな周波数成分の近傍を一般化調和解析で詳細に解析する高速化モードを用意した。FFT の解析には、ハミング窓を使用し、CUDA ライブラリ CUFFT を用いて GPU 上でスペクトル計算を行った。

以下に GPU 上での一般化調和解析の並列処理の手順を示す。

CPU から GPU に引き渡された信号の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。

- (1) (FFT を前処理に利用するモード) FFT により周波数の解析区間を限定する。
- (2) 入力信号に対しての残差エネルギーを任意の解析周波数分求める。
- (3) 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とそのパワースペクトルを抽出する。
- (4) 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。

GPU 内での並列処理は、各フレームの各周波数における残差エネルギー計算にスレッドレベルの並列処理を適用し、それら複数フレームの並列処理をブロックレベルで行った。即ち、フレーム内とフレーム間の並列性を利用する階層的並列処理を実現している。

マルチ GPU による並列処理は、CPU 側で Pthreads ライブラリにより複数スレッドを生成し、各スレッドから CUDA のカーネル関数を実行することにより実現する。これにより、複数フレーム間の並列性をさらに引き出すことができる。

#### 4.3.2 ハーモニッククラスタリングのフレーム内・フレーム間の並列処理

ハーモニッククラスタリングによる解析では一つのフレームに関して多重ネストのループが複数存在する。本手法では倍音クラスタ群数  $K$  と倍音数  $N$  についてループアンローリング [10] を行い、スレッドでの並列処理部分とした。これをフレーム内並列処理とする。一度の処理に使うスレッド数は  $\sum_{k=1}^K N(k)$  となる。

フレーム内並列処理では、計算順序制約 (データ依存) [10] が存在する。また、(8) 式の分母のように一度の計算で複数回利用する計算部分が存在する。よって、複数回使

用する計算部分は、GPU 上のメモリに計算結果を保持する。データ依存を満たすため、一度の計算が終了する際に同期をとる。また、フレーム間にはデータ依存関係がないので、各フレームでの計算をブロックごとに分けることによってフレーム間並列処理を実現できる。CUDA では、ブロックとして使うことのできる最大数は  $65535 * 65535$  個である。サンプリング周波数を  $F$ 、元の音響信号の演奏時間を  $T$ 、フレーム長を  $L$ 、フレームシフトを  $S$  とした時、フレーム総数  $Num$  は  $Num = (F * T - L) / S + 1$  として求めることができる。

この  $Num$  が  $65535 * 65535$  を超えない限り CUDA で扱うことができる。仮に扱えないほどの長い信号だった場合は、解析する音響信号を複数に分割し、順に処理することによって解決される。

## 5. GPU を用いた音高推定システムの性能評価

音高推定システムの CUDA プログラムを Tesla S2050 上で並列実行し、性能評価を行った。

### 5.1 性能評価環境

本システムで使用した Tesla S2050 は、Fermi アーキテクチャの GPU を 4 基搭載した GPU サーバであり、各 GPU は 32 個の SP からなる SM が 14 個搭載されている。よって、各 GPU に搭載されている SP の総数 (CUDA コア) は 448 である。各コアはクロック周波数 1.15GHz で動作している。CUDA の処理系はバージョン 3.1 である。

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 44100Hz、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長を約 23ms (標本点数 1024 点)、解析フレームのシフト数をフレーム長と同じ約 23ms (標本点数 1024 点) とした。

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 [11] にある楽曲を使用している。

### 5.2 一般化調和解析の性能評価

まず、一般化調和解析単体の GPU 上での性能評価として、ショパンのノクターン 第 2 番 op.9-2 の 14 フレーム (0.322 秒) を用いる。一般化調和解析の解析周波数の区間はナイキスト周波数分である  $1 \sim 22050\text{Hz}$  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。また、解析音数はハーモニッククラスタリングで音高推定するのに十分とされる 64 音とした。処理速度の測定は GPU へフレームデータを引き渡してから周波数スペクトルが解析されるまでとした。FFT を前処理として利用した実行モードの場合は FFT の解析も実行時間に含めている。

FFT を前処理とした一般化調和解析の解析区間は、サン

表 1 一般化調和解析とハーモニッククラスタリングの 14 フレームにおける並列処理 ( ショパンノクターン第 2 番 )

実行方式	CUDA コア数 (SM*SP)	ブロック数	スレッド数	一般化調和解析		ハーモニッククラスタリング		合計	
				時間 [s]	逐次比 [倍]	時間 [s]	逐次比 [倍]	時間 [s]	逐次比 [倍]
逐次処理	1*1	1	1	9256.32	1.0	786.83	1.0	10043.15	1.0
フレーム内の並列処理	1*32	1	32	295.54	31.3	30.96	25.4	326.50	30.8
			1024	48.04	192.7	2.85	276.6	50.89	197.4
フレーム間の並列処理	14*1	14	1	859.36	10.8	57.92	13.6	917.28	11.0
フレーム内・間の並列処理	14*32	14	1024	3.22	2874.6	0.20	3872.2	3.42	2933.8
フレーム内・間の並列処理 (FFT による前処理あり)				0.50	18512.6			0.70	14282.1

プリング周波数/標本点数で求められる。評価環境で設定したパラメータの場合 FFT で解析された最もパワーの高いスペクトルに集約されている 44Hz 分の中に求めるべき正確な周波数があると推測される。今回は解析の精度を上げるため FFT で解析されたスペクトル ± 44Hz を一般化調和解析の解析区間とした。また、FFT を利用の有無による一般化調和解析の精度の差はなかった。

GPU 上での並列処理による実行時間は表 1 に示した通りであり、フレーム内・フレーム間を並列化対象とすることにより逐次処理の場合に比べ最大 2874 倍の速度向上を得た。フレーム内の並列処理では、スレッド数が 1024 の場合は、32 スレッドの場合と演算資源 (CUDA コアは 32 個) は変わらないが並列処理性能が上がった。即ち、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確認された。

さらに、FFT を一般化調和解析の前処理として利用した場合には、利用しない場合と比べて 6.5 倍の速度向上が得られた。

### 5.3 ハーモニッククラスタリングの性能評価

一般化調和解析と同様に、ハーモニッククラスタリングの GPU 上での並列処理結果を表 1 に示す。フレーム内の並列処理では、276 倍の速度向上を得た。また、フレーム間並列処理では使用ブロック数にほぼ比例した速度向上比を得ており、スケーラビリティは保たれている。

ここで、1 フレーム内での推定基本周波数の導出過程を図 2 に示す。ここではピアノ音源の単音により解析を行った。解析対象は RWC 研究用音楽データベースの楽器音 No.011PFNOF から 277.18Hz (C#4) 部分を抜き出している。

図 2 では横軸に推定同時発音数をとっており、右に進むにつれて推定同時発音数が絞られていく様子がわかる。この図で、ある間隔ごとに山ができていくのは、解析中に

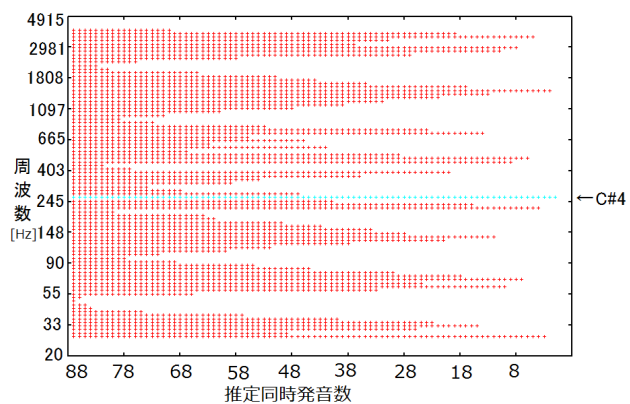


図 2 1 フレーム内での推定基本周波数の変化 (C#4)

オクターブ違いの同一ピッチが重みをもつためだと考えられる。

### 5.4 音高推定システムの性能評価

一般化調和解析とハーモニッククラスタリングによる音高推定システムの実行結果 (14 フレーム) は、表 1 の合計の部分である。これは、一般化調和解析からハーモニッククラスタリングへのデータの受け渡しを GPU 上で行った場合の数値となる。但し、今回の計測では、一般化調和解析の結果をテキストファイルとして保存し、それをもとにハーモニッククラスタリングを行っている。

次に、表 1 で用いたノクターン 第 2 番全体 (10400 フレーム、演奏時間 3 分 59 秒) を用いて性能評価した結果を図 3 に示す。ここでは、音高推定システム全体で計算時間を割いている一般化調和解析部分について GPU 数を増やして高速化を実現している。GPU を 4 基用いた場合は、FFT の前処理ありの一般化調和解析とハーモニッククラスタリングの並列時間が、GPU1 基と比べて 3.7 倍の速度向上が得られており、その実行時間は 237.70 秒となる。演奏時間は 239 秒なのでほぼリアルタイムに解析できたことになる。

同様に、楽曲を『きらきら星』の主題による変奏曲 K.



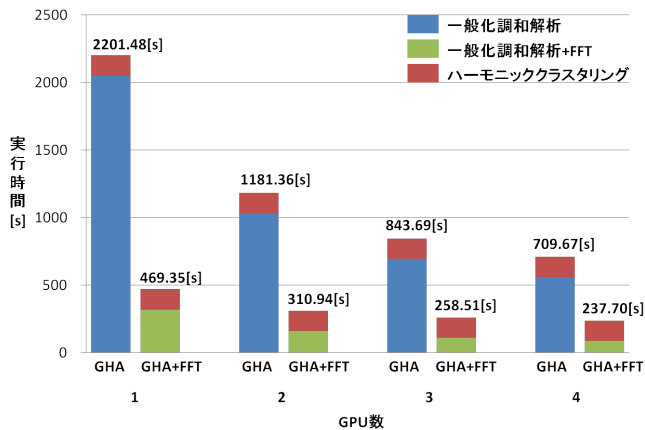


図 3 複数 GPU による並列処理 (ショパンノクターン第 2 番)

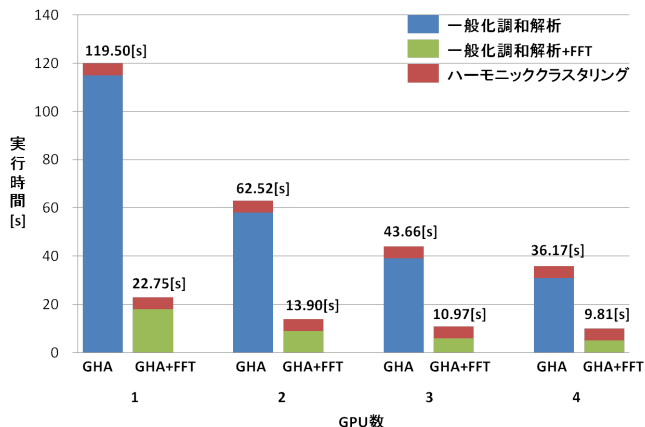


図 4 複数 GPU による並列処理 (モーツァルトきらきら星)

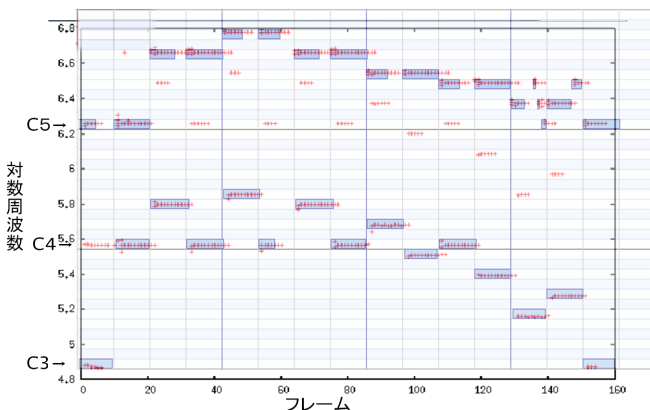


図 5 出力されたピアノロールと元信号の比較 (モーツァルトきらきら星)

265/300e (モーツァルト, W.A.) として, 音高推定の解析時間を GPU 数を変化させて測定した結果を図 4 に示す. 図 4 は 300 フレーム (6.9 秒) の解析時間を計測しており, 4 基の GPU を使うことにより, 1 基の GPU と比べて 3.7 倍の速度向上が得られた.

この場合の解析結果をピアノロールとして出力し, 精度の確認を行った. 正解のピアノロールと重ねたものが図 5 である. 図 5 のように元の信号と比較したところ, オクターブ位置の間違が見られたものの, 同一ピッチのオクターブ違いを検出することができた. 手作業で正誤判断を行い, (正答数 - 誤答数) / 元信号の発音総数で正答率を求め

た結果, 70.2 % の正答率を得た.

誤りのほとんどが, オクターブ位置の異なる同一ピッチを多く検出してしまっていることにある. また, 音の減衰に伴う発音終了時刻のブレがみられるが, 発音開始フレームが合っていれば正答とみなした.

## 6. おわりに

本稿では, GPU 上での一般化調和解析を用いたハーモニッククラスタリングによる基本周波数解析システムの並列処理手法を提案した.

Tesla S2050 上での性能評価から, 本手法では一般化調和解析の並列化によりフレーム内並列処理で 192 倍, フレーム間並列処理で 10.8 倍の速度向上が得られた. ハーモニッククラスタリングではフレーム内並列処理で 276 倍, フレーム間並列処理では使用ブロック数とほぼ同等の 13.6 倍の速度向上を得た. それらを合わせた基本周波数解析システム全体では 1CUDA コア比で 2933 倍の速度向上を得た. さらに, 一般化調和解析実行時に FFT による前処理を採用することで, 1CUDA コア比で 14282 倍の速度向上を確認することができた. また, ピアノロールを出力し元の信号と比較したところ, 70.2 % の正答率を得ている.

今後の課題としては, GPU のメモリ階層やストリームの利用, ハーモニッククラスタリングの複数 GPU での並列処理によるさらなる速度向上が挙げられる.

## 参考文献

- [1] Curtis Roads, "コンピュータ音楽 - 歴史・テクノロジー・アート," 東京電機大学出版局, 2001.
- [2] N. Wiener, "Generalized harmonic analysis," Acta Mathematica, vol.55, pp.117-285, 1930.
- [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," 情報処理学会論文誌, No.48, pp.27-32, 2003.
- [4] 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と情報量基準による音楽の音高/音源数の推定," 情報処理学会研究報告, MUS-62, pp.27-32, 2005.
- [5] 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.
- [6] 牛山聡, 東山三樹夫, 飯塚昌弘, 平田能睦, "一般調和解析による波形分析," 信学技報, EA93-103, pp.39-44, 1994.
- [7] 大内康裕, 及川靖広, 山崎芳男, "一般化調和解析を用いた基本周波数の推定および雑音と信号の分離," JASjournal, Vol.2, pp.30-36, 2000.
- [8] H.Noda, A.Nishihara, "Fast and Accurate Generalized Harmonic Analysis Using Newton's Method," APSIPA ASC 2009, pp.866-875, 2009.
- [9] 赤池弘次, 甘利俊一, 北川源四郎, 樺島祥介, 下平英寿, 室田一雄, 土屋隆, "赤池情報量基準 AIC-モデリング・予測・知識発見-" 共立出版, 2007.
- [10] 笠原博徳, "並列処理技術" コロナ社, 1991.
- [11] 後藤真孝, 橋口博樹, 西村拓一, 岡隆一, "RWC 研究用音楽データベース:音楽ジャンルデータベースと楽器音データベース," 2002-MUS-45-4, pp.19-26, 2002.