

## 生音源を使った現在演奏位置解析システム

## Current Score Position Analysis based on Live Sound Source

大日野 舞† 森 雅智‡ 金澤 貴俊† 徳田 英幸‡

## 1. 背景

楽器演奏をより広め、間口を広くすることは我々の生活をより豊かにし、充実したものにしていく助けとなる。しかし、楽器演奏には多くの練習が必要であり、習熟度を上げるためには快適な練習環境が必要である。具体的な支援の内容としては、楽譜の自動譜めくり、演奏間違い箇所の訂正やアドバイスといったものが考えられる。こうした楽器演奏を支援する情報システムを構築する上で重要となるのが、演奏者が演奏する曲の現在演奏位置推定機構である。演奏者が今曲のどの位置を演奏しているのかをリアルタイムに表示することで、演奏中の演奏者に対して有用なサービスを提供することが実現可能になる。

本稿では、楽器演奏をサポートするためのリアルタイム演奏位置取得システム STRAW (Score Tracking RAW) を提案する。我々の提案するシステムでは、電子楽器のような特別な楽器を用いず、生音源からの入力に対応する。これにより、演奏者側に専用の楽器を使用してもらい負担無しにサービスを提供することができる。

## 2. 目的

STRAW は、楽器演奏において、ある程度練習を重ねた演奏者が曲を通して演奏する際に現在演奏位置を解析し、演奏の支援をすることを目的とする。

特徴として、入力方法に生音源を用いるということが挙げられる。従来のシステムでは正確な音程の入力を解析上の前提条件としていたため、電子的な構造を持たない楽器での演奏においては利用することができない。しかし、これでは演奏者は使い慣れた楽器を使うことができず、また電子楽器が存在しないか高価な場合、システムを利用することが困難となる。

STRAW では、生音源を解析して現在演奏位置を特定することで、よりユーザに受け入れやすいシステムを目指す。STRAW を利用することで、自動譜めくり装置や自動伴奏システムの支援を行うことができる。

## 3. 既存研究

現在演奏位置推定のためのシステムは、既にいくつかの手法が提案されている。尾崎らが提案する現在演奏位置解析シ

テム[3]は、セル空間マッチングを用いることで演奏位置を取得する。この手法により音抜け、音追加などのミスを指摘することができるが、一方で、MIDI 楽器を使用し、音程を正確に入力しなければならないという制限があり、実際の楽器演奏には向かない。また、三浦らの提案する現在演奏位置解析システム[4]の音数と波長をマッチングする手法は本研究とアプローチの方向性が似ている。しかし、波形比較に用いる演奏楽器数の多い部分/少ない部分の境目が不明瞭であり、解析に最適な楽器数を特定するのが困難であると考えられる。

## 4. システム概要

本章では我々の提案する現在演奏位置解析システム STRAW の概要を述べる。図1に、STRAW のシステム構成を示す。

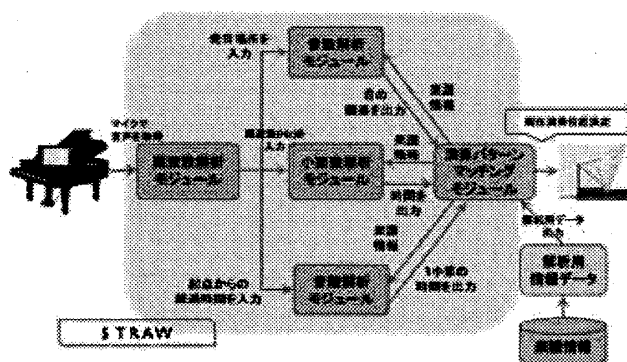


図1 システム構成

## 4.1. モジュール構成

本節では STRAW の各モジュールについて述べる。

## 周波数解析モジュール

マイクで取得した生音源を解析し、周波数、経過時間、音のレベルを出力する。音数解析モジュール、小節数解析モジュール、音階解析モジュールはこれらの情報を用いて処理を行う。

## 音数解析モジュール

解析用情報データから読み込んだデータをもとに、演奏している音が小節内のどの音かを把握する。音数とは楽譜情報において演奏開始位置からの音の順番を示す。解析用情報データからは音の並び順を読み込み、周波数解析モジュールからは発音回数を取得する。発音回数は音のレベルの変化によって得る。

## 小節数解析モジュール

$n$  小節目の最初の音から  $(n+1)$  小節目の最初の音までの時

† 慶應義塾大学 環境情報学部

‡ 慶應義塾大学大学院 政策・メディア研究科

Email:jpn@ht.sfc.keio.ac.jp

間を算出する。それら二つの音は、音数解析モジュールから得る。

#### 音階解析モジュール

1つ前の音との周波数(Hz)の差によって解析を行う。Hzの差は音階(音の乗降)とする。

#### 演奏パターンマッチングモジュール:

音数解析モジュール、小節数解析モジュール、音階解析モジュールから出力された情報をまとめ、現在演奏位置を出力するモジュールである。

各モジュールを組み合わせた現在演奏位置解析アルゴリズムを次節に述べる。

### 4.2. 現在演奏位置解析アルゴリズム

STRAWは、時間経過によって現在演奏位置を解析する。入力部分は以下の手順で行う。

1. 周波数解析モジュールにて発音を取得。
2. 情報を音数解析モジュールに入力し、音数  $m$  を増加。

また、解析手法はキャリブレーションフェーズと継続解析フェーズから成る。演奏開始 2 小節はキャリブレーションフェーズにおいて解析を行い、その後は継続解析フェーズで解析を続行する。キャリブレーションフェーズで算出した 1 小節の時間を継続解析フェーズで修正することにより、現在演奏位置を解析する。図 2 は 1 小節の時間を算出する手法を図に表したものである。



音数(m)	1	2	3	4	5	6	7	8	9
小節	1	1	1	2	2	2	2	2	3

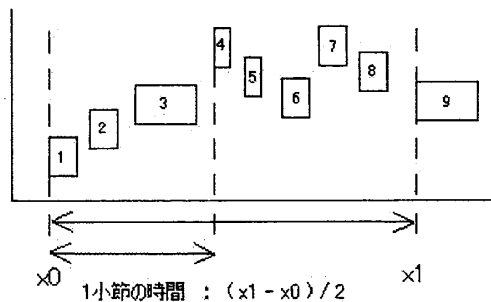


図 2 1小節にかかる時間の割り出し

#### キャリブレーションフェーズ

最初はキャリブレーションのため、2小節分の時間を算出し、平均を取って 1 小節の時間を割り出す。その手法は以下の手順を取る。

- (1) 楽譜情報から 2 小節分の音数  $m$  を取得する
- (2) 最初の発音時間を  $x_0$  に、(演奏開始小節+2)小節の 1 小節目の発音を時間  $x_1$  とする
- (3)  $(x_0 - x_1)/2$  で平均を出し、1 小節の時間を算出する。

#### 継続解析フェーズ

キャリブレーションフェーズで算出した 1 小節にかかった時間を修正していくことで現在演奏位置を解析する。修正を

行うことで、音抜けによる拍のズレに対応する。修正方法は以下の手順を取る。

- (1) 1つ前の音との周波数の差より音階を取得し、楽譜情報とマッチングさせ正しく演奏されていた場合は 1 小節にかかった時間を記録する。正しい演奏でなかった場合は破棄する。時間の取得方法はキャリブレーションと同様に行う。
- (2) (1) で取得した値を 4 小節分として計算する。平均を取って 1 小節分の時間を算出し、演奏位置の修正を行う。

## 5. 実装と評価

実装環境は ThinkPad T43p を用いて Windows XP 上で実装した。開発言語は C++ 言語を使い、Win32 API ライブラリと FFT ライブラリを用いた。

STRAW と既存研究を比較し、定性的評価を行った。

表 1 既存研究との比較

	STRAW	既存研究[1]	既存研究[5]
生音源取得	◎	△	×
音程ミス	×	◎	◎
位置解析	◎	○	◎
実現性	◎	○	△

様々な現在演奏位置解析システムがある中で、STRAW はより実現性が高い。楽器演奏を機械化することなく生音源で取得できる STRAW はユーザに受け入れやすいシステムと考えられ、通し演奏を行うための支援システムに向いている。

## 6. まとめ

本稿では、生音源による現在演奏位置解析システムの提案を行った。STRAW は、音数、小節数、音階を用いて演奏開始時刻からの経過時間を算出し、楽譜情報とマッチングさせることで現在演奏位置を解析する。STRAW を利用することで、手軽に現在演奏位置解析を実現出来る。さらに、自動譜めくり装置の支援や自動伴奏システムなどの支援を行い、ユーザの楽器演奏がより充実したものとなるだろう。

## 7. 関連研究

- [1] Roger B. Dannenberg. An On-Line Algorithm for Real-Time Accompaniment. In *Proceedings of the ICMC*, pages 193--198, 1984.
- [2] Miller Puckette. EXPLODE: A User Interface for Sequencing and Score Following. In *Proceedings of the ICMC*, pages 259--261, 1990.
- [3] 尾崎昭剛, 原尾政輝, 平田耕一: 演奏習得支援システムのための効率的な演奏現在位置解析アルゴリズム, エンタテインメントコンピューティング 2003 論文集, IPSJ Symposium Series, Vol.2003 No.1, pp.41-46(2003)
- [4] T. Miura, A. Akabane, M. Sato, T. Tsuda and S. Inoue: "Score Following of Orchestral Music Using Acoustic Pressure Peak-Tracking and Linear Stretch Matching", *Proceedings of the 2006 International Computer Music Conference (ICMC 2006)*, pp.624--627