

## 音列マッチングに基づく演奏現在位置解析アルゴリズム

尾崎 昭剛<sup>†</sup>, 原尾 政輝<sup>†</sup>

<sup>†</sup> 崇城大学情報学部ソフトウェアサイエンス学科

鍵盤楽器演奏を支援するシステムの開発には、ユーザの演奏の現在位置を正しく認識する現在位置解析アルゴリズムが必要不可欠である。これまでは、主に DP を用いた手法が用いられていたが、誤りを多く含む演奏に対しては認識率が低下し、特に演奏位置変更に対しては致命的な問題があった。本論文では音列同士のマッチングに基づいた新しいアルゴリズムを提案し、それが、音の追加、削除、音高変更、位置変更、といった演奏誤りに対しても頑強であり、DP を用いた手法に比べて同等の計算量で、高い認識率が達成できる事を示す。

## A Present Position Analysis Algorithm based on Plural Events Matching

<sup>†</sup> Shougo Ozaki, <sup>†</sup> Masateru Harao

<sup>†</sup> Sojo University Faculty of Computer and Information Sciences Department of Software Science

The present position analysis algorithm is necessary to develop a system which accompanies or supports human's instrumental performance. Until now, DP matching algorithm is mainly used, however the algorithm causes the miss-recognition when performance include mistakes. In particular the algorithm can't recognize positional changes. In this paper, we propose a new algorithm based on the match of the plural input events, and show the algorithm is robust for the performance mistakes such as, additions, deletions, pitch changes, and positional changes, and can achieve a high recognition rate compared with the DP approach.

### 1 はじめに

人間に合わせて鍵盤楽器の演奏を支援するシステムを開発するためには、ユーザの演奏のテンポ、リズム、演奏誤りといった特徴を解析する必要がある。

そのためには、ユーザの演奏の楽譜上の位置を正しく認識するための現在位置解析が必要不可欠である。特に、練習途上のユーザへの支援のためには、演奏に含まれる多様な誤りに対処できる手法が求められる。

これまでに提案されている現在位置解析アルゴリズムとしては、モチーフによるマッチング[1][2]、HMM によるもの[3]、DP を用いたもの[4]-[7]が知られている。特定の数の演奏イベントをモチーフとして、マッチングを行うアルゴリズム[1]は、時間的要素によりマッチングを行っているため、リズム、テンポが安定しない練習者の演奏に対しては、認識率が低下するという問題がある。また、HMM を用いたアルゴリズム[3]では、事前にユーザの演奏の統計量からモデルを構築する必要があるため、楽曲、

ユーザが変わると演奏の傾向が変化してしまい、認識率が低下する。そのため、不特定のユーザ、楽曲には対応できず、汎用性が低い。

汎用的に利用できる手法としては、DP によるアルゴリズムが知られており、自動伴奏システム[4][6]、演奏情報抽出[5]、連弾支援システム[7]といったシステムで用いられている。この手法は比較的小さい計算量で実現する事が可能であるが、演奏に誤りが含まれる場合には誤認識が生じやすい。特に楽譜上での演奏位置が過去方向に変更されると現在位置を見失ってしまう事が多い。練習者の各種演奏誤りへの対処を行ったとされる DP の改良アルゴリズム[7]も提案されているが、経験に基づく改良で本質的な問題は解決されておらず、どの程度誤りに対処できるかの検討は行われていない。

そこで、本論文では練習中のユーザの支援を行うシステム開発のために、音の削除、追加、音高変更、位置変更といった誤りを含む演奏に対しても、正しい演奏位置を認識できる音列マッチングに基づく現在位置解析アルゴリズムの提案を行う。

この手法は、複数の演奏イベント（音列）をマッチングに用いるため、誤って演奏された音が楽譜上の別の位置に出現する場合にも、誤認識を減らし、正しい認識を行う事が可能である。また、音列マッチングと単音マッチングを組み合わせる事で両者の長所を生かすアルゴリズムの構築が可能である。

本論文では、大島らの改良 DP アルゴリズム[7]との比較を行い、提案するアルゴリズムが改良 DP 法と同等の計算量である事を示す。また、実際に人間に単旋律の演奏を行ってもらい、本アルゴリズムがより高速な応答時間、より高い解析精度を実現している事を確認する。

本論文の構成は次の通りである。第2章でこれまでに提案されている現在位置解析アルゴリズムの概要と問題点を述べる。第3章で新しく提案するアルゴリズムの解説を行う。第4章では誤りを含む演奏への対処能力の評価実験を行い、改良 DP 法との比較を行う。第5章はまとめと今後の課題である。

## 2 演奏解析アルゴリズム

### 2.1 現在位置解析

現在位置解析とは、演奏情報から楽譜上の位置を特定するアルゴリズムである。本論文では、演奏誤りとして次の4つを想定する。

- (1) 音の削除：楽譜上の音が演奏されない
- (2) 音高変更：楽譜上の音の高さが変更される
- (3) 音の追加：楽譜上に無い音が演奏される
- (4) 位置変更：演奏がこれまでの楽譜上の現在位置より過去方向へ移動する

この中で、(1)~(3)の3つを局所的誤りと呼ぶ。また、(4)の位置変更については、未来方向への移動は、音の削除と同じであるため（移動前と移動後の間の音符が削除されたとみなす事ができる）、ここでは過去方向への移動のみを扱う。

### 2.2 これまでの手法

これまでに提案されている現在位置解析アルゴリズムとしては、モチーフによるマッチング[1][2]、HMM によるもの[3]、DP を用いたもの[4]-[7]が知られている。

特定の数の演奏イベントをモチーフとして、マッチングを行うアルゴリズム[1][2]は、時間的要素によりマッチングを行っているため、リズム、テンポが安定しない練習者の演奏に対しては、認識率が低下するという問題がある。

また、HMM を用いたアルゴリズム[3]では、事前にユーザの演奏の統計量からモデルを構築する必要

があるため、楽曲、ユーザが変わると演奏の傾向が変化してしまい、認識率が低下する。そのため、不特定のユーザ、楽曲には対応できず、汎用性が低い。

汎用的に利用できる手法としては、DP によるアルゴリズムが知られており、自動伴奏システム[4][7]、演奏情報抽出[5]、連弾支援システム[6]といったシステムで用いられている。これらは比較的小さい計算量で実現する事が可能であるが、演奏に誤りが含まれる場合には誤認識が生じやすい。この手法の問題点は、次節で述べる。

### 2.3 DP マッチングに基づくアルゴリズムと問題点

Dannenberg らによって提案された手法(以下 DP 法と呼ぶ)は、楽譜を行、演奏イベントを列とする行列により処理を行う。

DP 法は音の追加、音高変更に対しては、解析成功率が大幅に悪化する問題がある。これは、楽譜と演奏がマッチする位置が、直前の認識位置の前後にあった場合には、未来位置の方が優先されるためである。このため、音の追加、音高変更が発生し、誤って演奏された音が1つ前の認識位置よりも未来位置でマッチすると、誤認識が起こる。

また、DP 法では、演奏位置の変更は考慮されていないため、正しい認識は行えない。

楽譜、演奏が

楽譜：1234567

演奏：123451234

となったとする。この例では、演奏が5まで進んだ後に、1への位置変更が起きている。仮に、楽譜1~7の音がそれぞれ別の音高であれば、先述のような未来方向への位置変更は発生しない。しかし、DP によるベストマッチの計算では、2度目の1~4の演奏が入力されても、5の時点でのベストマッチを超える事ができないため、位置変更を検出する事が不可能である。この場合、6以降の音が入力されるまで、DP による認識位置は5のままになる。

このように、DP 法は過去方向への位置変更に対して非常に脆弱である。

### 2.4 DP マッチングの改良と問題点

楽譜上の検索範囲を限定するウィンドウを設定し、さらに位置変更[3]にペナルティを用いる事で誤認識を抑制するという改良もなされている[7]が、本質的に過去方向への移動への対処は行われていない。これは、ユーザが「より良い演奏」を行うために、常に楽譜上の順番通りに演奏を行うという前提のある、自動伴奏システムにおいては致命的な問題とはなら

ない。しかし、本論文が目標とする、ユーザの練習中の演奏を解析する場合には、位置移動を含む演奏誤りが頻発する事が予想されるため、用いる事ができない。

初心者のための支援を目的としたシステムには、これらの問題に対処するため、DP 法を改良したもの(以下、改良 DP 法と呼ぶ)も提案されている[7]。大島らの手法では、位置変更の発生時に演奏が再開される位置を調査し、DP の行列の重みを経験的に変更する事で、対処できるとしている。

しかし、大島らのアルゴリズムでも、演奏イベント全体と楽譜イベント全体のベストマッチを計算するという基本的な性質は変わっておらず、具体的にどの程度対処できるかの検証は行われていない。

### 3 音列マッチングによる現在位置認識

#### 識アルゴリズム

##### 3.1 音列マッチング

本論文では、本アルゴリズムは DP 法の問題点を考慮し、次のような考えに基づいた手法を構築する事で、認識率の向上を図る。

- ・テンポ、リズムの乱れに対処するため、ユーザの演奏は入力された順番のみ用いる
- ・複数の演奏音(音列)をマッチングに用いる事で、誤り入力時の予期せぬ位置移動を制限する
- ・直前の現在位置の前後に検索範囲を持ち、過去方向への位置移動も容易にする

複数イベント同士のマッチングを計算するために、まず、単独イベント同士の対応行列 MatchTable(p,s) を定義する。この行列は、演奏 p と楽譜 s の間で共通する音符の数を保持するためのものであり、MatchTable(p,s) の値は次のように決定される。

- ・p 番目の演奏と s 番目の楽譜上の音符が一致する場合：一致音符数
- ・p 番目の演奏と s 番目の楽譜上の音符が一致しない場合：0

本アルゴリズムでは、一定時間内に演奏された音は和音として扱うため、一致音符数は常に 1 とは限らない。

演奏イベントが入力されると、その列の単イベントマッチングが行われ、図 2 左のような行列となる。

次に、音列同士のマッチングを行い、共通する音符数(音列マッチ数)を調べる。本アルゴリズムでは、マッチングに用いる音列を連続する複数イベントと定義し、テンプレートと呼ぶ。テンプレート幅 T のイベントをマッチングに用いるとすると、演奏

イベント p, 楽譜 s の位置の音列マッチ数 M は

$$M=0$$

for I=0 to T-1

$$M=M+MatchTable(p-I,s-I)$$

next I

という処理で求めることができる。図 1 は、テンプレート幅 3 の場合の計算例である。そして、音列マッチング数の最大となる位置を検索する。音列マッチング数が最大となる箇所でも、常に最新の演奏イベントがマッチするとは限らないため、その中で、楽譜とマッチしている最も新しいイベントを現在位置とする。

これらの情報を利用する現在位置解析アルゴリズムとしては、次のようになる。

- (1) 単イベントマッチング行列を更新(図 1 左)
- (2) 新しい演奏が 1 つ前の認識位置の次の音であれば、その音を新しい現在位置とする
- (3) 単イベントマッチング行列より、テンプレート幅内の音符のマッチング数を計算(図 1 右)
- (4) 音列マッチング数の最も多い場所を検索。最大数が複数箇所ある場合には、以下の位置を優先
  - ・直前の認識位置に近い位置
  - ・楽譜上の未来方向の位置
- (5) 4 で決定された最多マッチング位置において、テンプレート内で楽譜とマッチングする最も新しい演奏の位置を現在位置とする

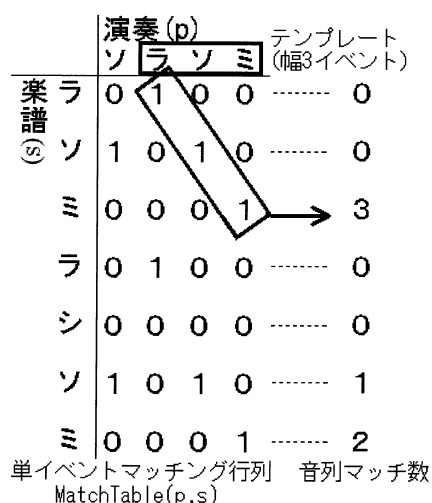


図 1 音列を用いたマッチ音符数の計算  
Fig.1 An calculation of matching notes

複数イベントをテンプレートとしてマッチングに用いることで、誤った音が入ってきても、DP 法のような位置変更は起こり難くなる。本手法では演奏誤りが入力された場合には、正しい演奏がある程度入ってくるまで認識位置が動かないという現象が起きる。どの程度で認識位置が移動するかは、テンプレート幅に依存する。

テンプレート内に含まれる演奏は、過去の演奏イベントと最新の演奏イベントに分けられる (図 2)。

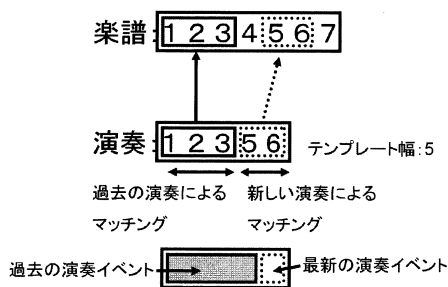


図 2 テンプレート内のイベントのマッチング  
Fig 2 An Event matching of template

演奏誤りが起きた場合には、新しい演奏によるマッチングが過去の演奏によるマッチングより大きくなった時点で、認識位置が移動することになる。図 2 は、音の削除が起きた場合の例である。この状態では、新しい演奏は 2 つで、過去の演奏によるマッチングの 3 つより少ないため、認識位置は 3 のままとなる。この後、楽譜 7 の音が入力されると、新しい演奏によるマッチングがテンプレート内で過半数となるため、認識位置が 7 へと移動する。

このように、音列マッチングに基づくアルゴリズムでは、誤って入力された音が楽譜の他の位置とマッチングしても、連続する複数イベントが楽譜と一致しない限りは位置移動が起きない事が分かる。

しかし、この性質は逆に、演奏誤りが発生してから正しい認識位置に到達するまでのタイムラグが起る原因となる。そこで、音列マッチングアルゴリズムが認識位置を移動するまでの間、以下の通り補助的に単イベントマッチングを用いることで、演奏誤り発生時のレスポンスの向上を図る。

- (1) テンプレート幅  $T$  で音列マッチングアルゴリズムを実行
- (2) テンプレート幅  $T$  での認識が 1 つ前の演奏の認識位置から移動しているか判定
  - ・移動している場合：テンプレート幅  $T$  の認識位置を新しい現在位置として使用
  - ・移動していない場合：(3) へ進む

- (3) テンプレート幅 1 で音列マッチングアルゴリズムを実行し、新しい現在位置とする

このようなアルゴリズムにする事で、基本的には音列マッチングが優先されるが、演奏誤り発生時に認識位置が移動しない場合のみ、暫定的に認識範囲内にあるマッチする音を認識位置とする。

本論文では、この手法を複合テンプレートによる音列マッチング (複合音列法) と呼ぶ。

### 3. 2 検索範囲

練習中の演奏者が演奏位置を変更する場合には、直前に演奏中の小節の最初、もしくは、一つ前の小節の最初の部分から演奏再開される可能性が高い事が知られている[7]。

しかし、楽譜上での決まった音符数しか検索しないと、小節に含まれる音符数が多い場合には、検索範囲が前後の小節をどの程度含むかの保証ができなくなる。そこで、本システムでは検索範囲を、現在演奏中の小節の前後 2 小節とし、楽譜に応じて動的に検索イベント数 (音符数) を変化させる。これにより、演奏位置変更が起こった際に、最も可能性が高い変更先を常に検索範囲に収める事が可能となる。

## 4 現在位置解析能力の検証

### 4. 1 計算機演奏による解析能力検証実験

単旋律の演奏に対して、音列法のテンプレート幅を 2~5 に変化させ、現在位置解析能力にどのように影響するか実験を行った。実験曲は「バッハ ピアノ小品集 (全音楽譜出版社)」「ブルグミュラー 25 の練習曲 (全音楽譜出版社)」から、それぞれ最初の 6 曲ずつ、計 12 曲の右手部分を用い、演奏の入力は仮想的に計算機内部でデータを発生させる。

実験の条件は以下の通りである。

- ・演奏誤り：全演奏イベントに対して設定した割合で演奏誤りを発生。音の削除、音高変更、追加、位置変更の局所的誤りが、等確率で出現する。
  - ・試行回数：一つの条件につき 1000 回ずつ試行。
- また、この実験では演奏誤りとして、2.1 節で紹介した 4 種類を発生させた。この実験では、解析能力の指標として、次に示す正解率を用いる。

S : 正解率

A : アルゴリズムによる正解イベント数

E : 入力イベント数

$$S = \frac{A}{E} \times 100 \quad (1)$$

比較対象として、大島らの提案する改良 DP 法による解析結果を用いる。実験結果を図 3、図 4 に示す。

#### 4. 2 人間の演奏者に対する単パート解析実験

前節での実験は、計算機に演奏を行わせて、3種の局所誤りと位置変更に対する対処能力の検証を行ったが、実際のシステムに利用する際には、人間の演奏に対して、どの程度の精度で解析が行えるかが重要である。そこで、鍵盤楽器の経験者 4 名、未経験者 4 名の計 8 名の演奏に対して、大島らの改良 DP マッチング法と音列マッチングによる解析の比較を行う。

実験の条件は以下の通りである。

- ・人間が実験曲を両手で演奏したデータを用いる
- ・演奏者はピアノ経験 5 年以上、ブランク 6 年以上の 3 名
- ・正解データは、音楽経験 13 年の協力者が各演奏を解析したもの

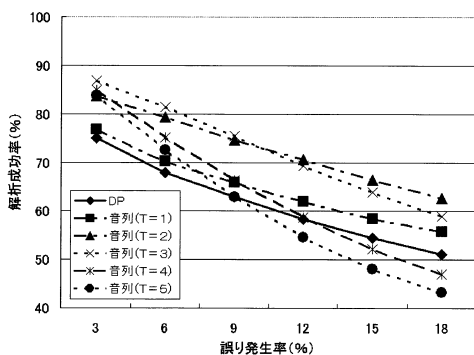


図 3 単一テンプレートによる解析成功率

Fig 3 The success rate of an template

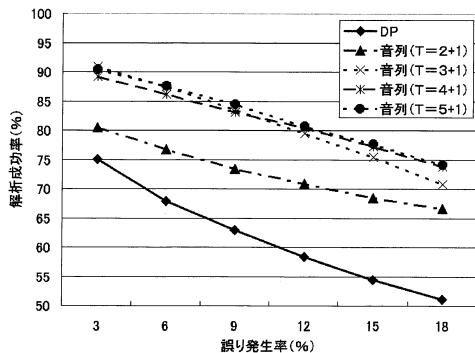


図 4 複合テンプレートによる解析成功率

Fig 4 The success rate of multiple template

本実験では人間が演奏を行い、意図せず誤りが発生するため、演奏の位置や誤りの発生を正確に知る事ができない。1 つの演奏イベントに対し、複数の正解が考えられる場合もあるが、今回は、人間による解析結果を正解データとして用いる。

本実験による実験結果を表 1 に示す。

表 1 人間の演奏に対する解析成功率

Table 1 The result of position analysis

改良 DP 法	複合音列法
83.9%	96.1%

#### 4. 3 応答時間

1 音入力された際に、アルゴリズムが現在位置を応答するまでの時間の測定を行う。実験環境は以下の通りである。

CPU:AMD Athlon 64X2 4200+

Memory:DDR2 SDRAM 2GB

OS : Windows XP Professional

表 2 に大島らの改良 DP 法と複合テンプレートマッチングの 1 イベント辺りの応答時間を示す。

表 2 1 イベント辺りの応答時間

Fig 2 The response time of one event

改良 DP 法	音列法
3.0 $\mu$ s	2.5 $\mu$ s

### 5 考察

単一テンプレート幅による認識結果では、テンプレート幅内の新旧イベントのマッチ音符数の比の問題で位置変更が遅れ、結果として解析成功率の低下の原因となっていた。しかし、テンプレート幅 1 による補正を加えた複合テンプレートによるアルゴリズムでは、すべてのテンプレート幅において、DP より良い結果が得られた (図 3、図 4)。演奏誤り発生率 15% (6~7 音に 1 度誤りが発生) という場合でも、複合音列法はテンプレート幅 3~5 の場合で正解率 75% を超えているのに対し、改良 DP 法は 55% 程度となっている。この事より、改良 DP 法でも演奏誤りに脆弱である一方、複合音列法は+20%前後と大幅に改善されている事が分かる。

以上の結果より、テンプレート幅が広がった際の”誤った演奏位置に動きにくい”という性質と、テンプレート幅 1 の”近辺のマッチする音を即座に現在位置とする”という両方の長所を組み合わせる事に成功していると考えられる。



各種演奏誤りが複合的に発生する人間の演奏に対する実験では、表 1 より平均 12.3%改良 DP 法より良い正解率を得ている。個々の演奏事例でも、複合音列法の正解率は改良 DP 法に対して最低+7%、最大+21%と全ての場合で上回っており、本論文で提案する手法が、実際のユーザに対しても有効である事が分かる。

計算量としては、DP 法は楽譜の長さに比例するため、楽譜上の音符数  $n$  とすると、計算量  $O(n)$  となり、改良 DP 法でも同様である。それに対し、パート認識付き音列マッチングアルゴリズムでは、テンプレート幅 (T)、検索範囲内音符数 (V) に比例するため、 $O(TV)$  となる。しかし、テンプレート幅は定数であるため、計算量  $O(V)$  となる。

DP 法も検索範囲を制限して実装される事があり、その場合には検索範囲内音符数に比例するため、実質的に、両アルゴリズムの計算量は同等であると考えられる。

また、音域限定音列法の場合、楽譜データ読み込み時に出現音高リストを作成するために、楽譜上の出現音をすべて 1 度走査する処理を行う。この処理の計算量は  $O(n)$  となるが、演奏入力前の操作であるため、演奏解析時の応答時間への影響は無い。

実際の応答時間では表 2 より、改良 DP 法、複合テンプレートマッチング、ともに非常に高速であり、実用上問題ない応答時間である事が分かる。

## 6 まとめと今後の課題

本論文では、誤りを含む演奏に対しても正しい現在位置の認識を行うことができるアルゴリズムについての提案を行った。大島らの改良 DP 法との認識能力の比較を行い、本アルゴリズムが、誤りの多い演奏に対して有効である事を確認した。

今後の課題としては、鍵盤以外の楽器や、多人数の演奏者への対応といった拡張が挙げられる。

### 参考文献

- [1] Bridet Baird, Donald Blevins, Noel Zahler, "The Artificially Intelligent Computer Performer on The Macintosh II and a Pattern Matching Algorithm for Real-Time Interactive Performance," ICMC 1989 pp.13-16, 1989.
- [2] Bridget Baird, Donald Blevins, Noel Zahler, "Artificial Intelligence and Music: Implementing an Interactive Computer Performer," Computer Music Journal, Vol.17, No.2, pp.73-79, 1993.
- [3] 武田晴登, 西本卓也, 嵯峨山茂樹, "HMMによ

- る MIDI 演奏の楽譜追跡と自動伴奏," 情処学音楽情報科学研報 MUS-66, pp.109-116, 2006.
- [4] R.B.Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," Proc of ICMC, pp. 193-198, 1984.
- [5] 高見啓史, 片寄晴弘, 井口征士, "DP マッチングを用いた演奏情報の抽出," 日本音響学会講演論文集 pp. 457-458, 1988.
- [6] Bloch, J.J. and R.B.Dannenberg, "Real-Time Computer Accompaniment of Keyboard Performances," Proc of ICMC, pp.279-289, 1985.
- [7] 大島千佳, 西本一志, 鈴木雅実: 家庭における子どもの練習意欲を高めるピアノ連弾支援システムの提案, 情報処理学会論文誌, 「知の共有から知の協創へ」特集号, Vol.46, No.1, pp.157-171, 2005