

誤り・任意の弾き直し・弾き飛ばしを含む演奏音響信号への 高速な楽譜追跡

中村 友彦^{1,a)} 中村 栄太^{1,†1,b)} 嵯峨山 茂樹^{1,†1,c)}

概要: 本研究では、楽譜を参照しつつ演奏に伴奏を同期再生する自動伴奏の実現のため、誤りや弾き直し・弾き飛ばし（ジャンプ）を含む演奏に実時間で追従する音響入力楽譜追跡を計算量に着目し議論する。自動伴奏は、演奏者の生演奏を支援するだけでなく練習に対する効用も大きく、その中核技術である楽譜追跡は自動伴奏実現のため重要な課題である。練習中に、ジャンプは演奏者により任意に起こされるため、楽譜追跡を行うためには任意のジャンプを扱うことが望ましい。任意のジャンプを扱うと、楽譜追跡の計算量削減が必要となる。そこで我々は、演奏をジャンプ確率が直前の音符に関して独立であると仮定した隠れマルコフモデルを用いて表現し、計算量を総音符数の2乗オーダーから1乗オーダーまで削減できることを示した。また、これらの提案法が約10000音符までの実際的な楽譜に対し実時間で動作し、ジャンプ後3.8s程度で演奏に追従できることを実験により示す。

1. はじめに

楽譜を参照しつつ人間の演奏に合わせて伴奏を自動で再生する技術は、自動伴奏と呼ばれる。この技術は、一人でのオーケストラを伴う演奏の支援だけでなく、合奏曲などの練習への効用も期待できる。演奏音響信号から楽譜上の演奏箇所を逐次推定する技術（以下、楽譜追跡）は自動伴奏の中核技術である上、他にも自動譜めくりや自動タグ付けなど幅広い応用があり、それ自体も有用である。

人間の演奏には、テンポや音量の変動のような楽譜から得にくい不確定要素が含まれる。また、演奏時には演奏者は音高を誤ったり、同一の音符を誤って再度弾くことや、いくつかの音符を飛ばして演奏してしまうこともある。これらに加え、演奏の音響信号には環境による雑音や、楽器による音響的な変動が含まれる。そのため、楽譜追跡は容易ではない課題であり、Dannenberg [1] や Vercoe [2] を研究の発端として多くの研究が行われてきた [3–19]。

特に練習においては、誤りの修正や特定の箇所の練習を行うため、演奏者は何度も弾き直し・弾き飛ばし（以下、ジャンプ）を行う。実際的な場面で楽譜追跡をするために

は、このようなジャンプを扱うことが望ましい。従来、特定箇所から特定箇所へのジャンプは扱われていた [5–7,16]。しかし、ジャンプが起こりやすい箇所には演奏者によって異なる偏りがあるため、これらの特定箇所を得ることは一般的に困難であり、特に不特定多数の演奏者を対象とする場合や、musical instrument digital interface（以下、MIDI）形式のような楽譜情報を用いる場合は尚更である。

任意のジャンプを許すと探索空間が膨大となり、その結果計算量が増大し、楽譜追跡精度が低下する危険性があるという2つの問題が生じる。4節で示すように、[3,5,6]を単純に拡張して得られる手法では数千から10000程度のオーダーの音符数の実際的な楽譜^{*1}を実時間で処理することは困難である。そのため、計算量を削減する必要がある。

楽譜追跡に対するアプローチとして、演奏の音響的な変動や不確定な要素を扱うためには統計的な方法論が有力であると考えられており、多くの従来法でも用いられている [4]。これらの手法には、演奏箇所を扱うアプローチとテンポも同時に扱うアプローチがある。後者の方法は探索空間が大きくなりがちであるため、本稿ではより狭い探索空間をもつ前者のアプローチをとる。

以下では、単旋律の演奏に対し、明示的に誤り・任意のジャンプをモデル化した隠れマルコフモデル（Hidden Markov Model, 以下、HMM）を用いて、楽譜追跡アルゴリズムを議論する。本稿では [18] を詳細化し、提案法と実験データの詳細を記述する。

^{*1} 例えば、Mozart のクラリネット 5 重奏の第 1 楽章では、そのパートの音符数は約 1900 である。

¹ 東京大学 大学院情報理工学研究所
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
^{†1} 現在、国立情報学研究所。
Presently with National Institute of Informatics
^{a)} nakamura@hil.t.u-tokyo.ac.jp
^{b)} enakamura@hil.t.u-tokyo.ac.jp
^{c)} sagayama@hil.t.u-tokyo.ac.jp

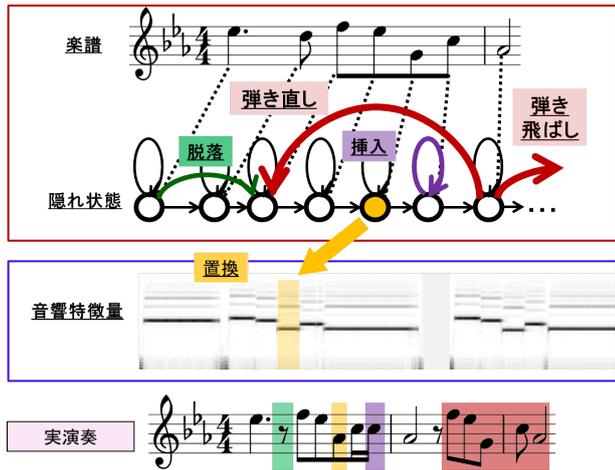


図 1 隠れマルコフモデルによる演奏モデルでの、挿入・脱落・置換誤り、弾き直し・弾き飛ばしの表現。挿入誤り（紫色）は自己遷移，脱落誤り（緑色）は一つ先の状態への遷移により表現される。置換誤り（橙色）は、異なる音高の音響特徴量の出力により表される。弾き直し・弾き飛ばし（赤色）は遠方への遷移に対応する。

2. 隠れマルコフモデルによる誤りを含む演奏モデル

楽譜追跡は、人間の演奏をモデル化し、音響信号から演奏箇所を推定する逆問題として捉えられる。誤り・ジャンプを含まない演奏は、順次1つ先の音符へ移りつつ音響信号を発する過程として捉えられ、音符を状態に対応させれば、音響信号を生成する状態を遷移する過程として記述できる。また、人間の演奏には、演奏表情や身体的な制約によるテンポや音長の変動が含まれ、その音響信号にも雑音や音響的な変動が含まれるため、演奏は確率的な過程として記述できる [3]。このとき、演奏される音符が直前に演奏された音符にのみ依存するならば、演奏モデルは HMM により表される。ここで、音響信号は同一音高でも音色が変化しうるため、音高に敏感でありながら音色変化に頑健な特徴量として、半音単位の中心周波数をもつ定 Q フィルタバンク (constant Q filterbank, 以下, CQF) の出力 (以下, CQF スペクトル) を用いた [12]。

誤りを含む演奏も HMM により記述できる [3]。脱落誤りは2つ先への状態への遷移，挿入誤りは自己遷移により表現される (図 2)。また、置換誤りは楽譜と異なる音高の出力により表現される (図 2)。そのため、誤りを含む演奏は left-to-right HMM で表現される。

3. 任意の弾き直し・弾き飛ばしのモデル化と計算量の削減

3.1 任意の弾き直し・弾き飛ばしのモデル化と計算量

2 節で述べたとおり、挿入・脱落・置換誤りを含む演奏は近傍への遷移と音響特徴量の出力によって表される。一方、[3, 5, 6] で提案されたモデルを一般化し、任意のジャンプ

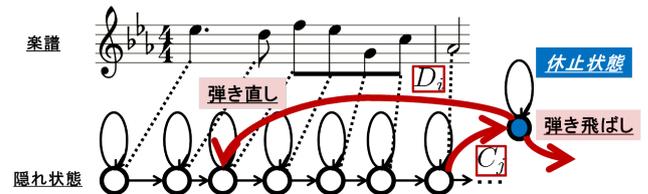


図 2 弾き直し・弾き飛ばし時の休止区間をモデル化した提案法での弾き直し・弾き飛ばしのモデル化。弾き直し・弾き飛ばしは、休止状態を経由する 2 段階の遷移によって表現できる。

プは各状態から全状態への遷移によってモデル化できる。したがって、誤りと任意のジャンプを含む演奏モデルは ergodic HMM により記述でき、その探索空間は膨大となる。

この演奏モデルに基づき、現在までの観測特徴量系列 $y_{1:t} = \{y_\tau\}_{\tau=1}^t$ が与えられた際に時刻 t での最尤状態 \hat{s}_t を求めるには、

$$\hat{s}_t = \operatorname{argmax}_{s_t} p(s_t | y_{1:t}) = \operatorname{argmax}_{s_t} p(y_{1:t}, s_t) \quad (1)$$

を解けばよい。ここで、 s_t は状態確率変数である。(1) の右辺は Bayes の定理により導出される。(1) は前向きアルゴリズムにより効率的に解け、時刻 t ・状態 i での前向き変数を $\alpha_t(i) (= p(y_{1:t}, s_t = i))$ とすると、時刻 $t = 1$ のときは、

$$\alpha_1(i) = b_i(y_1) \pi_i \quad (2)$$

時刻 $t \geq 2$ のときは、

$$\alpha_t(i) = b_i(y_t) \sum_{j=1}^M \alpha_{t-1}(j) A_{j,i} \quad (t \geq 2) \quad (3)$$

と表される。ここで、 M は状態数 (楽譜の総音符数と同一)、 π は初期分布、 A は遷移確率行列、 $b_i(y_t)$ は時刻 t ・状態 i での出力確率である。(3) の右辺では、 M 回の計算を M 状態毎に行うため、各時刻での計算量は $O(M^2)$ である。4 節で示すように、数千から 10000 程度の音符数の実際的な楽譜に対しては、(3) による実時間での計算は困難である。したがって、このような楽譜を処理するためには、計算量の削減が必要である。

3.2 計算量削減のためのモデルとアルゴリズム

計算量削減のためには、遷移確率行列に制約を設ける必要がある。本節では、2通りの演奏モデルを提案し、線形オーダーまで計算量を削減した楽譜追跡アルゴリズムを提案する。

1つ目のモデルでは、ジャンプ確率が直前の音符に関して独立であると仮定する。このとき A は、

$$A_{j,i} = B_{j,i} + C_j D_i \quad (4)$$

と表される。 B は誤りを含む演奏を表す行列であり、近傍 (2つ先の状態まで) への遷移のみで記述される。また、

ジャンプ前の演奏休止・ジャンプ後の演奏再開の起こりやすさは、 C_j と D_i により表現できる。ここで、 B, C_j, D_i は、 $C_j = 1 - \sum_{i=1}^M B_{j,i}$, $\sum_{i=1}^M D_i = 1$ を満たすように正規化されている。(4) を (3) に代入すると、

$$\alpha_t(i) = b_i(y_t) \left[\sum_{j=i}^{i+2} \alpha_{t-1}(j) B_{j,i} + \left(\sum_{j=1}^M \alpha_{t-1}(j) C_j \right) D_i \right] \quad (5)$$

を得る。ここで、右辺第2項の括弧内が i に依存しないため、この項は各時刻で1度計算すれば十分である。この項の計算量は $O(M)$ であり、残りの部分の計算量も $O(M)$ であるため、前向き変数の更新時の計算量を $O(M^2)$ から $O(M)$ に削減できる。また、 $C_i D_j$ を i, j によらず一定とすれば以前我々が開発した高速推定法と一致する [17, 19]。

ジャンプ時の無音の演奏休止区間に着目し、同様のモデルが構築できる。この休止を表す状態（以下、休止状態）を演奏モデルに付与すると、ジャンプは図2に示すように休止状態を経由する2段階の遷移として表現できる。このとき、遷移確率行列 $\tilde{A}_{i,j}$ は休止状態を含めた状態数 $N (= M + 1)$ を用いて、 $i, j \in [1, M]$ に関し

$$\tilde{A}_{i,j} = B_{i,j}, \tilde{A}_{i,N} = C_i, \tilde{A}_{N,j} = (1 - \tilde{A}_{N,N}) D_j \quad (6)$$

と表される。(6) を (3) に代入すると、

$$\alpha_t(i) = b_i(y_t) \left[\sum_{j=i}^{i+2} \alpha_{t-1}(j) B_{j,i} + \alpha_{t-1}(N) (1 - \tilde{A}_{N,N}) D_i \right] \quad (7)$$

$$\alpha_t(N) = b_N(y_t) \sum_{j=1}^N \tilde{A}_{j,N} \quad (8)$$

を得る。音符に対応する状態 ($i \in [1, M]$) の前向き変数の更新において、任意のジャンプは休止状態からの遷移に対応する。これは、(7) の第2項で記述できるため、状態 $i = 1, \dots, M$ までの更新時の計算量は計 $O(M)$ である。また、休止状態 ($i = N$) では、任意のジャンプが音符に対応する状態から休止状態への遷移によって表されるため、(8) の計算量は $O(M)$ である。したがって、更新時の全体の計算量は $O(N) \simeq O(M)$ ($M \gg 1$) に削減できる。

上記の計算量に関する議論は前向きアルゴリズムに基づくものであるが、同様の議論が Viterbi アルゴリズムでも、Mealy 型の出力分布をもつ HMM についても成り立つ。

3.3 提案するモデルの比較

前節で議論したモデルには、(4) と (6) の通り類似した構造が存在する。どちらのモデルでも、ジャンプ時の演奏休止・再開の起こりやすさを各状態ごとに表現でき、計算量削減の要は、ジャンプを表す遷移確率が演奏休止時の音符に関して独立と仮定した点である。一方で、両モデルの差異は、後者のモデルのようにジャンプ時の休止区間を

明示的にモデル化したか否かである。実際の演奏では、この休止区間は存在することが多く、ある程度の時間続くこともある。したがって、後者のモデルのほうが実際の演奏に対する楽譜追跡には適すると予期される。この差異の理論的な定量評価が困難であるため、次節で実験により評価する。

4. 計算量と楽譜追跡性能の実験による評価

4.1 実験条件

計算量と楽譜追跡性能の評価のため以下の3つの実験を行った。最初の実験では、提案法が実際の楽譜について実時間で動作するかを定量的に調査した。計算量は音高や音長ではなく主に音符数に左右されるため、人工的に用意した様々な音符数の楽譜を用いた。計算は、Intel Core 2 Duo P9400 2.40 GHz の CPU と 2 GB の RAM を持つ計算機上で行い、評価尺度としてシフト長に対する処理時間の割合を表すリアルタイムファクタ (Real Time Factor, 以下 RTF) を採用した。

第2の実験では、実際の練習での楽譜追跡の性能評価を行うため、1人のアマチュアクラリネット奏者による実際の練習での実演奏 (14 演奏, 計 1687 s) を用いた。演奏曲は、ポピュラー音楽・童謡・クラシック音楽のジャンルから異なる7曲を用いた (一部 RWC 音楽データベース [20] の曲を含む)。演奏には、45回の挿入・脱落・置換誤り、43回のジャンプが含まれており、これらは練習中自然に引き起こされたものであった。ジャンプ時の演奏休止・再開箇所は、楽譜上の時間で 0.1 s から 85 s (0 小節から 43 小節) 離れたものまでであった。また、実際的な状況での評価を行うために、楽譜追跡がジャンプ後に演奏に追従するまで演奏者は待たずに演奏した。評価尺度として、ジャンプの検出率とジャンプから追従するまでの時間 (秒・音符単位, 以下、追従時間) を用いて追従性能を評価した。追従時間は、ジャンプが起こされた時刻から、演奏のオンセット時刻の周り Δ ms 以内の演奏箇所が推定された時刻までの時間として定義した。

第2の実験では、3節の提案法と Cano らによる従来法 [3] に相当するジャンプをモデル化しないアルゴリズムを比較した。Cano らは音高やエネルギーを表す異なる音響特徴量を用いたが、本実験では CQF スペクトルを用いた。この差異は、楽譜追跡性能を低下させるものではなく、むしろ向上させるものであり [12]、この音響特徴量の選択は適切であると考えている。

最後の実験では、任意のジャンプのモデル化による楽譜追跡性能の低下があるかを検証した。評価するために十分な数の実演奏が用意できなかったため、ジャンプを含まない RWC ポピュラー・著作権切れデータベース [20] 中 112 曲の旋律パートの MIDI を音響信号に変換し代用した。評価尺度として、各曲毎の音符検出率の平均 (Piecewise Pre-

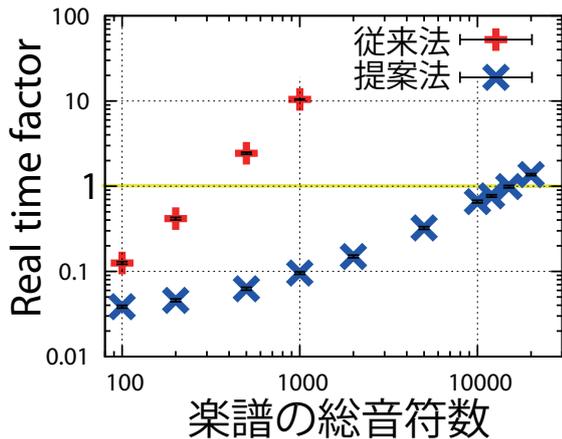


図 3 楽譜に含まれる総音符数に対する Real time factor (RTF). 赤点は, [3, 5, 6] のアルゴリズムを任意のジャンプに一般化したアルゴリズムの RTF を表し, 青点は休止状態を含むアルゴリズムの RTF を示している.

表 1 休止状態を含むアルゴリズム (w/ pause), 休止状態を含まないアルゴリズム (no pause), 従来のアルゴリズム (no repeat/skip) での弾き直し・弾き飛ばしの検出率.

評価尺度	w/ pause	no pause	no repeat/skip
ジャンプの検出率	32/43	29/43	8/43

cision Rate, 以下, PPR) と全曲中の音符検出率 (Overall Precision Rate, 以下, OPR) を用いた [21].

以上全ての実験で, サンプリング周波数 16 kHz, 16 bit 量子化の音響信号を, 窓長 128 ms・シフト長 20 ms の hanning 窓を用いて特徴量に変換した. 出力分布は, 事前に RWC 楽器音データベース [20] のクラリネット演奏で学習した. パラメータは, $\pi=[1, 0, 0, \dots, 0]^T$, $C_i=e^{-1000}$, $D_i=1/M$ ($i \in [1, M]$), $\hat{A}_{N,N}=0.98$ とした. また, 挿入・脱落誤り確率は $\exp(-500)$ とし, 半音・全音・完全 12 度の置換誤りをそれぞれ 0.001, 0.001, 0.0001 とした.

4.2 結果と考察

楽譜追跡の計算時間から導出した RTF を, 図 3 に示す. ここで, 休止状態のある提案法とない提案法では, 同様の結果が得られたため休止状態のある場合のみを示している. この図を見ると, 休止状態のある提案法は音符数に漸近的に比例して (図 3 での傾き 1 の直線に相当), 従来法は音符数の 2 乗に比例 (図 3 での傾き 2 の直線に対応) しており, 3 節の理論的な結果と一致する. また, 従来法では約数百個程度の総音符数の楽譜までであったが, 提案法では約 10000 個の音符数まで実時間処理できた. もちろん, 具体的な音符数は計算機の性能によっても変わりうるが, 提案アルゴリズムによる計算量削減の有効性は十分に確認できる.

$\Delta = 500$ ms での第 2 の実験の結果を, 表 1 と図 4 に示す. ジャンプをモデル化しない従来法よりも, モデル化する提案法が検出率が高く, 平均追従時間も 7 s (8 音符) 以

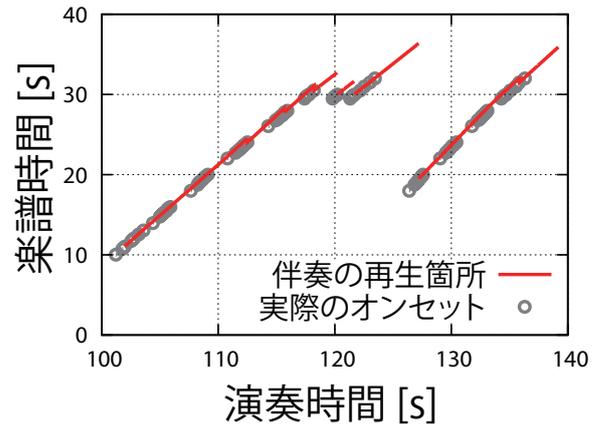


図 5 弾き直しを含む人間の演奏への自動伴奏の結果. 灰色の円は実際のオンセットを表し, 赤線は再生された伴奏を表す.

上速かった. これは, ジャンプのモデル化によりジャンプに対する追従性能が有意に向上することを示している. また, 3.3 節の考察の通り, 休止状態のある提案法が休止状態のない提案法より検出率で 0.06 程度高く, 平均追従時間も 1 s (2 音符) 程度速かった.

休止状態がある提案法では, 11 個のジャンプが検出できなかった. これらは, ポピュラー音楽でのサビの繰り返しなどの類似した区間やフレーズが存在することによる影響や, ジャンプ後に数音符しか演奏されなかったことが原因である. この場合には, 人間の伴奏者でも計算機でも一般的に楽譜追跡は困難である. 人間もジャンプへの追従に数秒・数音符はかかると考えられるため, 休止状態のある提案法の平均追従時間 3.8 ± 0.8 s (8 ± 1 音符) は実用に耐えうる程度であると考えられる.

最後の実験で, PPR は全てのアルゴリズムで 0.839 ± 0.009 であり, OPR は休止状態のない提案法で 30070/36051, その他のアルゴリズムは 30073/36051 であった. ほぼ同等の PPR, OPR を提案法でも従来法でも得たため, 任意のジャンプのモデル化による楽譜追跡性能の有意な低下は認められなかった.

4.3 自動伴奏の実装

我々は, 提案した楽譜追跡アルゴリズムを自動伴奏にも実装した. 伴奏再生モジュールは, 我々の先行研究である [17] で提案されたテンポ推定法と伴奏再生速度変換法を修正し用いた. 図 5 に, ジャンプを含む演奏に対する伴奏の追従結果の一部を示す. また, 自動伴奏時の映像を複数 http://hil.t.u-tokyo.ac.jp/~nakamura/demo/automatic_accompaniment.html で公開している.

5. 結論

本論文では, 線形オーダーまで計算量を削減した, 誤り・任意のジャンプを含む演奏への楽譜追跡アルゴリズムを 2 つ提案した. ジャンプの確率が直前の演奏した音符に関し

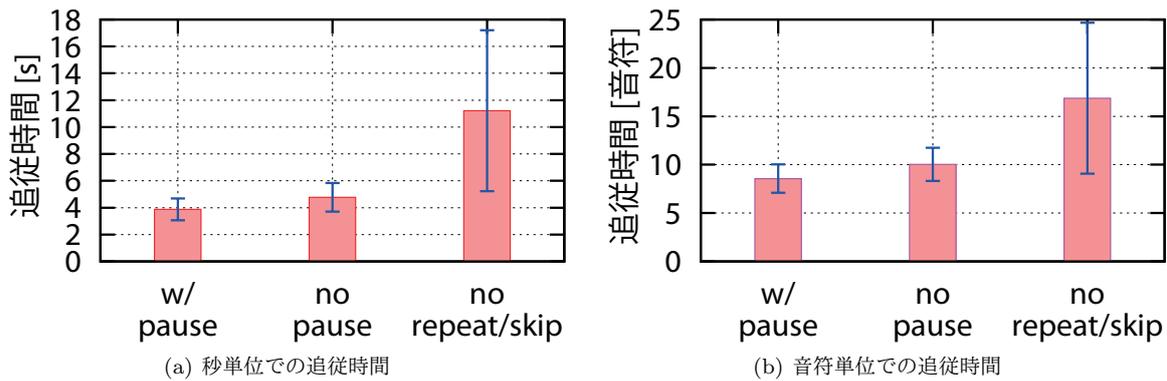


図 4 休止状態のある提案法 (w/ pause), 休止状態のない提案法 (no pause), 従来法 (no repeat/skip) での追従時間 ((a) 秒単位, (b) 音符単位).

て独立とするモデルにより, 計算量を $O(M^2)$ から $O(M)$ へと削減できた. また, ジャンプ中の休止区間の存在に着目し, この休止を明示的にモデル化することにより計算量が同様に削減できた. 実験により実時間処理と楽譜追跡性能に対する定量評価を行い, 提案法であれば 10000 音符程度の実際的な楽譜まで実時間で処理でき, ジャンプ後 3.8 s (8 音符) 程度で追従できることを示した. また, ジャンプ時の休止区間のモデル化によりジャンプに対する追従性能が向上することを確認した. また, ジャンプのモデル化による追従性能の有意な低下は見られなかった.

今後は, [9,11] で議論されているような, 多重音楽器による演奏への拡張が課題である. また, テンポ情報や拍情報による追跡性能の向上が多く報告されているため [13–15], これらの情報のモデルへの導入も課題である.

謝辞 クラリネット演奏により評価に協力して下さった伊東直哉氏に感謝する. 本研究の一部は, 文部科学省/学術振興会科学研究補助費 課題番号 (23240021) から補助を受けて行われた.

参考文献

- [1] Dannenberg, R.: An On-line Algorithm for Real-time Accompaniment, *Proc. of ICMC*, pp. 193–198 (1984).
- [2] Vercoe, B.: The Synthetic Performer in the Context of Live Performance, *Proc. of ICMC*, pp. 199–200 (1984).
- [3] Cano, P., Liscos, A. and Bonada, J.: Score-Performance Matching using HMMs, *Proc. of ICMC*, pp. 441–444 (1999).
- [4] Orio, N., Lemouton, S., Schwarz, D. and Schnell, N.: Score Following: State of the Art and New Developments, *Proc. of NIME*, pp. 36–41 (2003).
- [5] Tekin, M., Anagnostopoulou, C. and Tomita, Y.: Towards an intelligent score following system: Handling of mistakes and jumps encountered during piano practicing, *Proc. of CMMR*, pp. 211–219 (2004).
- [6] Pardo, B. and Birmingham, W.: Modeling form for online following of musical performances, *Proc. of AAAI*, Vol. 2, pp. 1018–1023 (2005).
- [7] 大島 千佳, 西本 一志, 鈴木雅美: 家庭における子供の練習意欲を高めるピアノ連弾支援システムの提案, *情処学論*, Vol. 46, No. 1, pp. 157–170 (2005).
- [8] 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, *情処学音情研報*, pp. 109–116 (2006).
- [9] Cont, A.: Realtime audio to score alignment for polyphonic music instruments, using sparse non-negative constraints and hierarchical HMMs, *Proc. of ICASSP*, pp. V-245–V-248 (2006).
- [10] Raphael, C.: Aligning music audio with symbolic scores using a hybrid graphical model, *Mach. learn.*, Vol. 65, No. 2, pp. 389–409 (2006).
- [11] Cont, A.: ANTESCOFO: Anticipatory synchronization and control of interactive parameters in computer music, *Proc. of ICMC* (2008).
- [12] Joder, C., Essid, S. and Richard, G.: A comparative study of tonal acoustic features for a symbolic level music-to-score alignment, *Proc. of ICASSP*, pp. 409–412 (2010).
- [13] Duan, Z. and Pardo, B.: A state space model for online polyphonic audio-score alignment, *Proc. of ICASSP*, pp. 197–200 (2011).
- [14] Otsuka, T., Nakadai, K., Takahashi, T., Ogata, T. and Okuno, H.: Real-time audio-to-score alignment using particle filter for coplayer music robots, *EURASIP JASP*, Vol. 2011, p. 2 (2011).
- [15] Joder, C., Essid, S. and Richard, G.: A conditional random field framework for robust and scalable audio-to-score matching, *IEEE TASLP*, Vol. 19, No. 8, pp. 2385–2397 (2011).
- [16] Montecchio, N. and Cont, A.: A unified approach to real time audio-to-score and audio-to-audio alignment using sequential monte-carlo inference techniques, *Proc. of ICASSP*, pp. 193–196 (2011).
- [17] 中村 友彦, 水野 優, 鈴木 孝輔, 中村 栄太, 樋口 祐介, 深山 寛, 嵯峨山茂樹: 音楽演奏の誤りや反復に頑健な音響入力自動伴奏, *音講論 (秋)*, pp. 931–934 (2012).
- [18] 中村 友彦, 中村 栄太, 嵯峨山茂樹: 弾き直し・弾き飛ばしを含む音楽演奏への高速な音響入力楽譜追跡, *情処学大* (2013).
- [19] 中村 栄太, 武田 晴登, 山本 龍一, 齋藤 康之, 酒向 慎司, 嵯峨山茂樹: 任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏, *情処学論*, Vol. 54, No. 4, pp. 1338–1349 (2013).
- [20] Goto, M.: Development of the RWC Music Database, *Proc. of ICA*, pp. 1–553–556 (2004).
- [21] Cont, A., Schwarz, D., Schnell, N. and Raphael, C.: Evaluation of real-time audio-to-score alignment, *Proc. of ISMIR* (2007).