5R-7

歌唱ロボットのためのビート情報とメロディ・ハーモニー情報の統合による音楽音響信号と楽譜の実時間同期手法の開発

大塚 琢馬 † 村田 和真 * 武田 龍 ‡ 中臺 一博 §* 高橋 徹 ‡ 尾形 哲也 ‡ 奥乃 博 ‡

† 京都大学 工学部情報学科 ‡ 京都大学大学院 情報学研究科 知能情報学専攻

* 東京工業大学大学院 情報理工学研究科 § (株) ホンダ・リサーチ・インスティチュート・ジャパン

1. はじめに

音楽が聞こえる環境で, ロボットが人と共生していく上で, 自らの耳で音楽を聞き, それに合わせて歌を唱い, 踊ることができる機能の実現, すなわち, 音楽ロボットの開発は重要である. これまでに, 吉井ら [1] は自分の耳で音楽を聞いて, ビートに合わせて足踏みをする音楽ロボットを, 村田ら [2] は, さらに歌唱する音楽ロボットを開発してきた. 一般に伴奏に合わせて歌を唱う場合, 伴奏の音楽音響信号と楽譜との時間的同期 (対応づけ) を実時間で行う必要がある.

村田らの歌唱ロボットは, 伴奏と楽譜との時間的同期を冒頭から行っているので, 一端同期がずれると, 楽譜のどの部分が演奏されているのか分からず迷う, という問題があった. このような経験は人でもよくするが, サビの部分を発見して, 楽譜の位置を再発見し, 唱い出すことができる. また, 途中から聞こえてきた音楽に合わせて唱う場合にも, 冒頭からの時間的同期法では対応できない.

2. 歌唱ロボットのシステムアーキテクチャ

我々が開発している歌唱ロボットのシステムアーキテクチャを図 1 に示す. ロボットは, 自分の耳で聞いた音から, 自分の発声した音響信号を抑制し, 伴奏音楽だけを抽出して, ビートトラッキングを行う. 得られたビート情報に合わせて, 楽譜位置を同期させ, さらに次のビートを予測して, 歌を唱う. 本システムアーキテクチャの新規性は, 自己発声分離機能にある.

音楽音響信号と楽譜との同期を行う音楽ロボットはほとんどなく, リズム安定性などを抽出するマリンバ演奏ロボット [3] などに留まっている. 本稿では, 村田らによる事前知識を用いないビートトラッキング手法 [2] を拡張し, 事前知識として楽譜を用いて, 伴奏音入力に対し同期する手法 (図 1 の破線部) について報告する. また, 今回は曲の冒頭からの同期のみについて報告する.

2.1 歌唱のための楽譜同期における問題

本稿における問題設定は次の通り.

> 入力 同期すべき音楽音響信号と MIDI 形式の楽譜
> 出力 楽譜上の現在演奏中の位置
> 仮定 今回の場合, 入力は楽譜の冒頭から

我々の目的は歌唱ロボットなので, 次の要求条件を満たす必要がある. (1) 実時間で同期する, (2) 人間の多彩な演奏に対応するため, 使用楽器やテンポが未知の多楽器・多声音楽に対応する. Cont は, 楽譜同期問題を HMM で定式化した [5] が, ピアノの和声に対して追従していることを報告するにとどまり, (2) を満たさない.

Realtime Syncronization Method between Audio Signal and Score Using Beats, Melodies, and Harmonies for Singer Robots: *Takuma Otsuka (Kyoto Univ.), Kazumasa Murata (Tokyo Inst. of Tech.), Ryu Takeda (Kyoto Univ.), Kazuhiro Nakadai (HRI-JP/TITech.), Toru Takahashi, Tetsuya Ogata, and Hiroshi G. Okuno (Kyoto Univ.)*

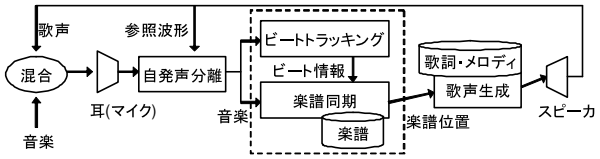

図 1: システム図: 破線で囲んだ範囲が本稿の対象

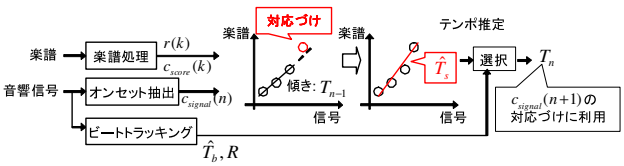

図 2: 楽譜同期の流れ

2.2 ビートのみに基づく同期の問題点

従来の歌唱ロボットには次の問題点がある.

1. ビート抽出誤りによるずれを修正困難
2. テンポが仮定の範囲外の曲に対応が困難

ビートによる楽曲位置の決定は, 冒頭から抽出したビートを数えることで行う. 従って, ビート検出誤りが起こると, それ以降はずれ続けてしまう. また, ビートトラッキング [2] が出力するテンポは 60–120M.M.*であり, 例えば, テンポ 150M.M. の曲は 75M.M. 間隔でビートを抽出するため, ビートの数え方に調整が必要であった. 本手法は, メロディ・ハーモニーに対応するクロマベクトルを入力音から抽出し, 事前知識である楽譜と対応づける.

3. 楽譜同期手法

楽譜同期の流れを図 2 に示す. まず, 楽譜から事前に情報を抽出し, 順次入力される音で, オンセット抽出, 楽譜との対応づけ, テンポ推定を行う. 特徴量には, 多楽器・多声音楽に対応するため, オンセット判定で 2 値化したクロマベクトルを用いる. また, 稀な音名のオンセットは信号から抽出しやすいことから, 3.1 節で定義する音名の珍しさを用いる. 以下, 楽譜を時分割したフレームを f, オンセットの番号を k, k 個目のオンセットのフレームを f_k とする. 信号についても同様に, 時間フレームを t, オンセット番号を n, n 個目のオンセットのフレームを t_n とする.

3.1 楽譜情報の抽出

初めに, 楽譜からオンセット情報を抽出する. 四分音符を 12 分割して楽譜の 1 フレームとした. フレーム f_k について, 音名 i ($C=1, C\sharp=2, \ldots, B=12$) の音符が存在す

*Mälzel's Metronome:一分あたりの四分音符の数

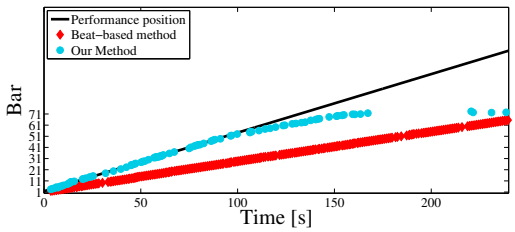

図 3: 楽譜同期結果 Pops No.63 (横軸:時間 縦軸:小節)

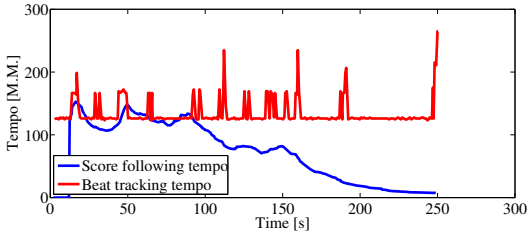

図 4: テンポ推定 Pops No.63 (横軸:時間 縦軸:テンポ)

れば $c_{score}(k,i)=1$ とする. 次に, 式 (1) の区間 L における各音名の珍しさ $r(k,i)$ を次式で定義する.

$$L = \{l | f_k \le f_l \le f_k + I\}, \quad (1)$$

$$r(k,i) = -\log_2 \sum_{l\in L} c_{score}(l,i) / \sum_{l\in L}\sum_{j=1}^{12} c_{score}(l,j) \quad (2)$$

ここで, L は区間 f_k から f_k+I の楽譜オンセット番号で, I は楽譜の 2 小節分とする.

3.2 音楽情報の抽出

入力信号を短時間周波数解析したスペクトログラムに帯域通過処理を行い, 12 次元クロマベクトル $c(t,i)$ を得る. また, Robet らの手法 [6] を用いてオンセット時刻 t_n を求める. 続いて, 以下の式に従い $c(t_n,i)$ を 2 値化した信号のオンセットクロマベクトル $c_{signal}(n,i)$ を得る.

$$c_{signal}(n,i) = \begin{cases} 1, & c(t_n,i) > \sum_{s=t_{n-1}}^{t_n-1} c(s,i)/(t_n - t_{n-1}) \\ 0, & otherwise \end{cases} \quad (3)$$

3.3 楽譜同期

類似度計算・対応づけ $c_{signal}(n)$ と, $c_{score}(k)$ の類似度 $D(n,k)$ を次式で定義する (i は省略).

$$D(n,k) = \sum_{i=1}^{12} r(k,i)(c_{signal}(n,i)c_{score}(k,i)) \quad (4)$$

$c_{signal}(n)$ が $c_{score}(\hat{k}_m)$ と対応しているとする. 新たなオンセット $c_{signal}(n+1)$ は, テンポ T_n を用いて次式により対応づける. このとき, 時系列性を考慮して, 式 (7) のように, オンセットを N 個遡った類似度の和を用いる.

$$A = T_n(t_{n+1} - t_n) \quad (5)$$

$$K = \{k | f_{\hat{k}_m} + A - J < f_k < f_{\hat{k}_m} + A + J\} \quad (6)$$

$$\hat{k}_{m+1} = \arg\max_{k\in K} \sum_{p=0}^{N-1} D(n-p, k-p) \quad (7)$$

ただし, J は楽譜の探索範囲 K を決める定数で, 楽譜の 2 拍分とする. ビートトラッキングが時刻 t_{n+1} をビート時刻と推定した場合, K を楽譜の拍の位置に制限する.

テンポ推定 テンポは対応づけ $(t_n, f_{\hat{k}_m})$ と, ビートトラッキングの出力から求める. 過去 L 個の対応づけ $(t_n, f_{\hat{k}_m}), \ldots, (t_{n-L+1}, f_{\hat{k}_{m-L+1}})$ を最小二乗法により線形近似した傾きをテンポ $\hat{T}_s$ とする. ビートトラッキングの出力は, ビート時刻 t, テンポ $\hat{T}_b$, 信頼度 R である. R はビートが安定しているとき高い値を取る. R が閾値以上のとき, T_s の値を元に, ビートトラッキング結果の T_b を本来のテンポに修正する. R が閾値以下のとき, $T_n = \hat{T}_s$ とする.

4. 評価実験

4.1 実験目的・条件

ビートトラッキングでは同期が困難な曲が本手法で同期可能であることを示す. 評価実験には, RWC 音楽データベース [7] の Jazz 曲 No.1–50, Pops 曲 No.1–50 曲の用いた. 入力音はマイクを介さず, 楽曲の波形を入力し, サンプリング周波数は 44.1 [kHz], FFT 窓長 4096 pts., シフト幅 512 pts. とした. 類似度計算とテンポ推定のパラメータは, $N = 2,4,8,16$, $L = 2,4,8,16,32,64$ の各曲 24 通りの組合せを用いた.

4.2 実験結果・考察

本手法のメリットと問題点をよく表した Pops No.63 の結果を図 3 に示す. 黒線が正解, 青円が本手法, 赤菱がビートトラッキングに基づく楽譜同期結果である. この曲のテンポは 126M.M. のため, ビートトラッキングのテンポを 2 倍に補正する必要がある. ビートトラッキングのみでは補正ができず, 時間がたつにつれて赤菱は正解から乖離した. 一方, 本手法は 100 秒までは同期したが, その後, 失敗した. 図 4 は青線: 本手法が保持したテンポ T_n, 赤線: ビートトラッキングによる補正後テンポ $2\hat{T}_b$ である. 楽譜同期が成功した 100 秒までは, テンポ推定も安定した. しかし, 100 秒以降は, テンポ T_n の推定が安定性を失い, 結果として楽譜同期を失敗した. テンポ推定が失敗する傾向は, 他の曲においても見られた. 以上より, 本手法はテンポ推定の改良が必要であることが分かる.

5. おわりに

本稿では, 歌唱ロボットのためのビート情報と楽譜情報を併用した楽譜同期手法を開発した. 今後の展開・課題は, (1) 本手法を実際のロボットへ適用, (2) 複数の楽譜を持った状態での楽曲同定・途中からの演奏への同期などがある. 本研究は科研費 (S), GCOE の 支援を受けた.

参考文献

[1] Yoshii *et al*. A Biped Robot that Keeps Steps in Time with Musical Beats while Listening to Music with Its Own Ears, *IEEE/RSJ IROS-2007*, 1743–1750.

[2] Murata, *et al*. A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing, *IEEE/RSJ IROS-2008*, 2459–2464.

[3] Weinberg, *et al*. Toward Robotic Musicianship, *Computer Music Journal*, 28–45, 2006.

[4] 柏野 他. ヒストグラム特徴を用いた音響信号の高速探索法, 信学論, **J81-D-II**:9 (1999) 1365–1373.

[5] Cont. Realtime Audio to Score Alignment For Polyphonic Music Instruments Using Sparse Non-negative Constraints and Hierarchical HMMs, *ICASSP-2006*, V-245–248.

[6] Rodet, *et al*. Detection and Modeling of Fast Attack Transients, *Computer Music Conf. 2001*, 30–33.

[7] 後藤, 他. RWC 研究用音楽データベース, 情処学論, **45**:3 (2004) 728–738.