

MIDIにおけるメロディ情報を利用した繰り返し構造の検出

Detection of refrain from melody track in MIDI

高田友則 橋口博樹

Tomonori Takada and Hiroki Hashiguchi

This paper concerns with the detection of refrain in MIDI format music. This format separates some tracks and we analyze a melody track to find the music structure based on the repetition. In our analysis, we utilize the consecutive silent symbols that discriminate the music structure. Furthermore, we adopt some matching methods. Experiments were conducted by actually using 115 MIDI files in the RWC popular music database.

Keyword: Dynamic Programming, Matching, Recall rate, Precision rate.

1 はじめに

最近の音楽情報処理のニーズ拡大とともに、音楽家向けの技術だけでなく、音楽家でない一般ユーザーが恩恵を受けるような技術が注目されている⁴⁾。コンピュータによる楽曲構造の理解、大量の楽曲集合からの検索技術やリアルタイム音楽情景記述システム^{1, 2)}の構築が取り組まれている。著者らも、楽曲の音響信号をデータベース、鼻歌をクエリ(検索要求)とした鼻歌検索に着手し、検索手法の提案⁶⁾¹⁰⁾⁹⁾や、データベースの圧縮を目的とした楽曲の要約⁷⁾を行っている。

本研究では、音符相当表現である標準 MIDI ファイルを用い繰り返し構造を検出することを目的とする。音響信号については、繰り返し構造、サビ検出の方法 (RefrainD)²⁾が提案されている。RefrainD の考え方に従って繰り返し構造を検出するにあたり、まずは MIDI のメロディトラックの部分解析対象とし、その無音部分を楽曲の構造の切れ目として有効に利用する。

2 特徴量の抽出と問題の定式化

2.1 特徴量の抽出

MIDI ファイルからメロディトラック (一般的には 4 トラック目) を取り出し、1 フレームを 1/16 秒として、

*埼玉大学工学部情報システム工学科

Department of Information and Computer Sciences, Faculty of Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama, Saitama, 338-8570, Japan

音を抽出する。ただし、音階のド (C) を 0 とし、そこから半音上がるごとに 1 上がるようにする。こうして、得られた単音の系列を $(v(1), v(2), \dots, v(W))$ とする。ただし、 W は曲の総フレーム数を表す。

2.2 無音部分を利用したグループ分け

一般的な楽曲では、構造の区切れ (A メロ・B メロの間など) では歌が歌われていないことが多い。この特徴を利用し、メロディトラックの連続する無音部分の長さがある閾値以上のとき、構造が分かれていると判断する。つまり、閾値より短い無音部分は同じ構造の一部とみなす。A メロ・B メロといった楽曲の構造は、一般に 7~10 程度に分かれているという先見知識に基づき、連続する無音の長さを調整する。楽曲 M 、連続する無音の長さ β を区切りとするグループの数を $C(M, \beta)$ とすると、 β が大きいほど $C(M, \beta)$ は小さくなり、 β が小さいほど $C(M, \beta)$ は大きくなるので、 $C(M, \beta) \geq 7$ を満たす最大の β をその楽曲の無音長の閾値候補とする。しかし、この β があまり小さすぎると、構造の区切れでないような所で分けてしまう可能性があるので、 β の下限を Z (実装では $Z = 20, 1.25$ 秒) とし、

$$\alpha = \max(\max\{\beta \mid C(M, \beta) \geq 7\}, Z)$$

となる α を閾値とする。閾値 α による無音部分で分けた互いに交わらない有音のグループ構造を

$$\Pi(\alpha) = \{\pi = [s, e] \mid 1 \leq s < e \leq W,$$

$$\pi \text{ に含まれる無音区間の長さはいずれも } \alpha \text{ 以下} \}$$

とする。ここで上の互いに交わらないとは、 $\pi_i = [s_i, e_i]$, $\pi_j = [s_j, e_j]$ ($i \neq j$) に対して、 $\pi_i \cap \pi_j = \emptyset$ を意味する。したがって、一般性を失うことなく $\Pi(\alpha)$ の要素数を n として

$$1 \leq s_1 < e_1 < s_2 < e_2 < \dots < s_n < e_n \leq W$$

としてよい。

2.3 繰り返しの検出

整数の数列 $\mathbf{a} = (a_1, a_2, \dots, a_m)$ と $\mathbf{b} = (b_1, b_2, \dots, b_n)$ に対して、一致度 $d(\mathbf{a}, \mathbf{b})$ を

$$d(\mathbf{a}, \mathbf{b}) = \begin{cases} \max_{0 \leq l \leq n-m} \left\{ \sum_{i=1}^m \frac{\delta(a_i, b_{i+l})}{m} \right\} & \text{if } m \leq n \\ \max_{0 \leq l \leq m-n} \left\{ \sum_{i=1}^n \frac{\delta(a_{i+l}, b_i)}{n} \right\} & \text{if } m > n \end{cases}$$

と定義する。ただし $\delta(s, t)$ は、 s と t に関するクロネッカーのデルタを表す。明らかに、 $0 \leq d(\mathbf{a}, \mathbf{b}) \leq 1$ である。

任意の $\pi = [s, e] \in \Pi(\alpha)$ に対して、 $\mathbf{v} = (v(1), v(2), \dots, v(W))$ の部分列 $\mathbf{v}_1(\pi)$ とその変換 $\mathbf{v}_2(\pi)$, $\mathbf{v}_3(\pi)$ を次のように定義する。

$$\mathbf{v}_1(\pi) = (v(s), v(s+1), \dots, v(e))$$

$$\mathbf{v}_2(\pi) = (0, v(s+1) - v(s), v(s+2) - v(s), \dots, v(e) - v(s))$$

$$\mathbf{v}_3(\pi) = (v(s+1) - v(s), v(s+2) - v(s+1), \dots, v(e) - v(e-1))$$

すなわち、 $\mathbf{v}_1(\pi)$ は単純な \mathbf{v} の部分列であり、 $\mathbf{v}_2(\pi)$ は始点からの相対音高列、 $\mathbf{v}_3(\pi)$ は階差音高列である。さらに、

$$y_1(\pi_i, \pi_j) = d(\mathbf{v}_1(\pi_i), \mathbf{v}_1(\pi_j))$$

$$y_2(\pi_i, \pi_j) = d(\mathbf{v}_2(\pi_i), \mathbf{v}_2(\pi_j))$$

$$y_3(\pi_i, \pi_j) = d(\mathbf{v}_3(\pi_i), \mathbf{v}_3(\pi_j))$$

によって有音区間に対して、音高に基づく類似度を定義する。また、DP マッチングによる類似度⁸⁾ を 0 から 1 に正規化して定義し直した類似度を $y_4(\pi_i, \pi_j)$ とする。

これらの準備の下、楽曲の切れ目構造 $\Pi(\alpha) = \{\pi_1, \dots, \pi_n\}$ から、各 π_i ($i = 1, \dots, n$) に類似する要素を集めた部分集合を作る。

$$\phi_i = \{\pi_j \in \Pi(\alpha) \mid y(\pi_i, \pi_j) \geq \epsilon, j = 1, \dots, n\}$$

ただし、 y は y_1, y_2, y_3, y_4 のいずれかを利用するが、 y_1, y_2, y_3 は完全マッチングなので $\epsilon = 0.7$ とし、 y_4 は伸縮が考慮された DP マッチングなので、0.7 よりも多少小さい 0.5 を採用した。

2.4 評価実験

「RWC 研究用音楽データベース：ポピュラー音楽と著作権切れ音楽」⁵⁾ の計 115 曲の MIDI を対象に、前節の方法で検出された繰り返し区間からサビ区間を特定する。サビ区間の特定には、RefraiD²⁾ の方法のうち、「曲中で多く繰り返される区間がサビである」という考え方と、「一般的な楽曲では、長い区間の繰り返しがあった場合、その末尾の部分がサビである可能性が高い」という特徴を利用している。なお、115 曲の正解のサビ区間は、一人の人間が聴いて作成した。また、評価方法も RefraiD の評価方法²⁾ に準じ、再現率 (R : recall rate)、適合率 (P : precision rate)、および両者を統合した F 値 (F : F-measure)¹²⁾ で評価する。

$$F = \frac{2R}{P+R}$$

$$R = \frac{\text{正しく検出したサビ区間の長さの合計}}{\text{正解のサビ区間の長さの合計}}$$

$$P = \frac{\text{正しく検出したサビ区間の長さの合計}}{\text{検出した区間の長さの合計}}$$

Table 1: Experimental results

	$\#\{F \geq 0.6\}$	$\#\{F \geq 0.75\}$	Ave. of Calc. Times
y_1	76	47	7.3 ms
y_2	74	45	9.2 ms
y_3	70	26	12.2 ms
y_4	74	36	98.6 ms

Table 1 に y_1, \dots, y_4 を用いたときの F 値が 0.6, 0.75 を超えた 115 曲中の曲数を示す。音響信号を扱った RefraiD²⁾ は $F \geq 0.75$ の (正答) 曲が 80 曲であり、本実験結果は MIDI を扱っているにもかかわらず、それに勝っていない。これは、メロディトラックのみを扱ったことが原因と考えられ、他のトラックを含めた方法に改良する必要性を意味する。これについては今後の課題としたい。

謝辞

本研究の一部は、平成 17 年度埼玉大学 21 世紀総合研究プロジェクト経費による支援を受けて行われた。

参考文献

- (1) 後藤真孝, リアルタイム音楽情景記述システム: 全体構想と音高推定手法の拡張, 情処研報音楽情報科学 2000-MUS-37-2,9, 16, 2000.
- (2) 後藤真孝, リアルタイム音楽情景記述システム: サビ区間検出法, 情報処理学会 音楽情報科学研究会 研究報告, 2002-MUS-47-6, Vol. 2002, No. 100, pp. 27-34, 2002.
- (3) 後藤真孝, SmartMusicKIOSK: サビ出し機能付き音楽視聴機, 情報処理学会 インタラクシオン 2003 論文集, pp.9-16, 2003.
- (4) 後藤真孝, 平田圭二, 音楽情報処理の最近の研究, 日本音響学会誌, Vol.60, No.11, pp.675-681, 2004.
- (5) 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: ポピュラー音楽データベースと著作権切れ音楽データベース, 情処研報音楽情報科学, 2001-MUS-42-6, pp. 35-42, 2001.
- (6) 橋口博樹, 西村拓一, 張 建新, 滝田順子, 岡隆一, モデル依存傾斜制限型の連続 DP を用いた鼻歌入力による楽曲信号のスポッティング検索, 電子情報通信学会論文誌, Vol.J84-D-II, No.12, pp.2479-2488, 2001.
- (7) 橋口博樹, 工藤雅志, クロマ類似度を用いた音楽データベースの圧縮, 埼玉大学紀要, 第 38 号, pp. 22-26, 2005.
- (8) Ning Hu, Roger B. Dannenberg and George Tzanetakis, Polyphonic Audio Matching and Alignment for Music Retrieval, 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New York: IEEE, pp. 185-188, 2003.
- (9) Nishimura, T., Hashiguchi, H., Takita, J., Zhang, J. X, Goto, M. and Oka, R., Music Signal Spotting Retrieval by a Humming Query Using Start Frame Feature Dependent Continuous Dynamic Programming, Proc. ISMIR 2001, pp.211-218, 2001.
- (10) 西村拓一, 橋口博樹, 関本信博, 張建新, 後藤真孝, 岡隆一, 始端特徴依存連続 DP を用いた鼻歌入力による楽曲信号のスポッティング検索の高速化, 情報処理学会音楽情報科学, Vol.42-2, pp.7-14, 2001.
- (11) 大津展之, 判別および最小 2 乗法基準に基づく自動しきい値選定法, 信学論 (D), J63-D, 4, pp. 349-356, 1980.
- (12) van Rijsbergen, C. J., Information Retrieval, Butterworths, 2nd edition, 1979.