

リズム入力による音楽検索方式「タタタタップ」

池谷 直紀^{†1} 服部 正典^{†2} 大須賀 昭彦^{†3}株式会社東芝 研究開発センター[‡]

1. はじめに

現行の音楽検索の方式としては、あらかじめジャンル別に分類したり曲名の五十音順などで整列したリストを用いたりして探す方式、曲名や実演者名、あるいは歌詞の一部などの曲に付随する情報の文字列をキーとして入力する検索方式、さらには曲の旋律を口ずさみ音声入力で検索するハミング検索[1]や、演奏情報からその曲を識別する方式などがある。

一方、筆者らは1つのボタンをリズムカルに打鍵する入力方式「タタタタップ」を提案しており[2]、本稿ではタタタタップによる音楽検索を提案し、実験による検索精度の結果を示した上でその実用可能性と課題について考察する。

2. タタタタップの概要

タタタタップは、モールス信号のように1つのボタンにより時系列信号を入力する方式である。モールス信号と比較すると、音楽の複雑なリズムの入力を受理することにより、利用者が入力パターンを学習しやすく、多彩な入力が可能である。そしてボタンを押している時間情報は用いず、打鍵のタイミングだけを用いていることから簡易な入力が可能である。

また、ハミング検索とも共通点はあるが、入力時に発声を必要とせずボタン一つがあればよい。そのため入力デバイスの適用範囲が広いことも特徴である。

タタタタップの性質は文字列の入力には向いていないが、あらかじめリズムと動作を関連付けておくことで、リズムをコマンドとして入力することにより機器を操作することもできる。そしてタタタタップの主な応用の一つが、入力リズムを含む音楽の検索である。

3. 構成と動作

タタタタップを音楽検索に用いる場合のシステム構成を図1に示す。あらかじめ、検索対象である音楽のリズムをリズムパターンDBに登録しておく。

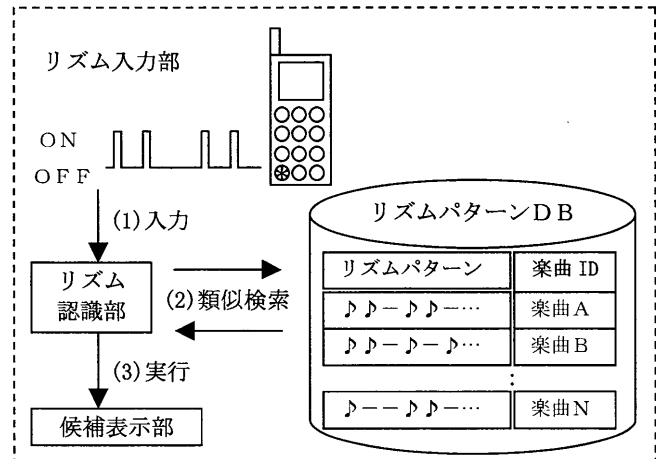


図1：タタタタップの構成と動作

録しておく。そして入力装置にはリズムの入力に使用する特定のボタンが1つあるものとする。そのボタンの ON/OFF の時系列信号がリズム認識部に入力される。リズム認識部では、入力された信号とリズムパターンDBに登録されたリズムパターンとの類似度判定を行い、類似度の高いものを検索結果として出力する。ここで、前述の通り ON（打鍵）のタイミングのみを扱うことで簡易な入力を実現する。また、入力のテンポが原曲と一致せずとも、またテンポが一定でなく揺らいでいても変動パターンが類似しているリズムを検索することで、より簡易な入力を実現する。

4. リズムの認識

タタタタップの大きな技術的課題は、入力された時系列信号と用意されたリズムパターンとの類似度判定である。テンポの指定なしに手入力されたリズムは、テンポの不明と入力の誤差により認識が容易ではない。演奏リズムの認識手法については楽器の演奏情報からその意図するリズムを特定する関連研究[3]が挙げられるが、タタタタップにおいては用意された複数のリズムパターンと入力リズムの類似判定であるため条件が異なる。本稿では以下の3つの判定アルゴリズムについて比較実験を行った。

"Tatata-Tap" - Music Search by Rhythmical Input

†¹ Naoki Iketani †² Masanori Hattori †³ Akihiko Ohsuga

‡ Corporate Research & Development Center, Toshiba

4. 1 リズムの類似判定アルゴリズム

(1) リズムベクトルに基づいた類似判定

文献[3]と同様に、連続する3つの音長の比をリズムベクトルとし、リズムパターンに含まれるリズムベクトルの列を比較して類似度を判定している。リズムベクトル r_{in} と r_{jn} の距離 $d(r_{in}, r_{jn})$ をユークリッド距離を用いて算出し、その総和 $D = \sum d(r_{in}, r_{jn})$ をリズムパターン間の距離として類似判定を行った。

(2) 差分二乗和による類似判定

これは時系列信号を線形に伸縮して重ね合わせ、打鍵時刻の差分二乗和を距離とする方式である。リズム系列を、開始から各打鍵までの経過時間の列 $r = \{t_n\}$ ($n=0, 1, \dots, \tau, t_0=0$)として表現する。そして二つのリズムパターン r_i と r_j の距離を、パラメータ $\phi = t_{j\tau} / t_{i\tau}$ を用いて、距離 $D = \sum (t_{jn} - \phi t_{in})^2$ として算出した。

(3) DPマッチングによる類似判定

時系列信号の比較方式として一般的なDPマッチングを用い、信号列を非線形に伸縮して重ね合わせて比較する方式。詳細な方式は紙面の都合上省略するが、リズムを音長の変化率 v の列 $r = \{v_n | n=1, 2, \dots, \tau-1, v_i = t_{i+1} / t_i\}$ として表現し、以下の漸化式を用いて距離を算出している。

$$g(i, j) = \text{Min} \left(\begin{array}{l} g(i-1, j) + d(i, j), \\ g(i-1, j-1) + 2d(i, j), \\ g(i, j-1) + d(i, j) \end{array} \right)$$

4. 2 リズムのマッチング方式

決定すべき方式は、単に類似度の判定方式だけではない。ある曲の主旋律のリズムを入力する場合にも、利用者によって曲のどの旋律のどの部分をどのように打鍵するかは異なる。そこで本稿では入力リズムが登録リズムのいずれの部分と一致するかの判定を、次の2つの方式で行った。

(A) 特定箇所との一致

あらかじめリズムDBにデータ登録を行う際に、打鍵される冒頭になる可能性の高い打鍵開始箇所を一つから複数設定しておく。これを $t_{j0}, t_{j1}, \dots, t_{jn}$ とする。実際には主に曲の冒頭やサビと呼ばれる主要部分の開始時刻を設定する。そして類似判定は全ての t_{jk} について入力リズムの打鍵時刻 $t_{i0} = t_{jk}$ として算出し、最も類似度の高かった t_{jk} を採用する。

(B) 全箇所との一致

これはさらに、全ての打鍵可能性のある箇所について重ね合わせを行うものである。すなわちリズムパターンの全ての打鍵時刻を $t_{k0}, t_{k1}, \dots, t_{kn}$ として、入力リズムの打鍵開始時刻 $t_{i0} = t_{jk}$ として類似度を算出し、その中で最も類似度が高く算出されたものをリズムパターンと入力リズムの類似度として用いる。

5. 実験

予備実験により、リズムの入力装置としてはマウスのボタンやタッチパッドよりも、キーボードのキーの方が入力誤差が少ないことがわかり、これを使用した。実験データには童謡60曲と歌謡曲など40曲の100曲を使用し、あらかじめ譜面およびMIDIデータから主旋律部分のリズムデータを作成した。被験者1人がこの100曲について一通り聴いて知っている状況にした上で、各曲の主旋律のリズムを想起しキーボードで打鍵する。ここでは各リズムの入力において21回の打鍵を行った。

前章に示した各アルゴリズム別に、十分打鍵した場合の最も高い正解率は以下であった。

表2：実験結果（一位正解率）

アルゴリズム	(1)	(2)	(3)
(A)	86%	80%	40%
(B)	65%	55%	28%

アルゴリズムとしては(1)の結果が比較的高い。(3)のDPマッチングの結果が劣った理由を分析したところ、全体的な伸縮により曖昧な検索を可能にするDPマッチングは、局所的な特徴量に乏しいタタタタップでは異なるものを類似していると判定してしまうためであった。

またマッチングの方式としては(A)の方式が(B)より優れている。これは単純に(A)の方式では1曲について一〜四箇所の部分との類似度を比較しているが、(B)の方式では300打鍵からなる曲においては300箇所近くの部分との類似度比較を行うため、意図していない曲の一部にも入力リズムと類似したパターンが出現する率が高まってしまったためである。さらに、数十〜数百倍もの部分とマッチングを行うため、特別な高速化を導入していない現時点では、(A)の方式では計算には0.1秒もかからないのに対し、(B)の方式では10秒を超える検索時間が必要であった。

また、このとき最も正解率が高かった(1)-(A)の方式において、打鍵数を6から21までに変化させた場合の正解率を、第一候補が正解である一位正解率および第一候補か第二候補が正解である二位正解率の変化をプロットしたものを図3に示す。このグラフから、今回実験で用いた100曲については、13回打った状況がピークであることがわかる。打鍵数を増やすことで徐々に正解率が下がってしまう原因は、打鍵の誤りが増えてしまうためである。

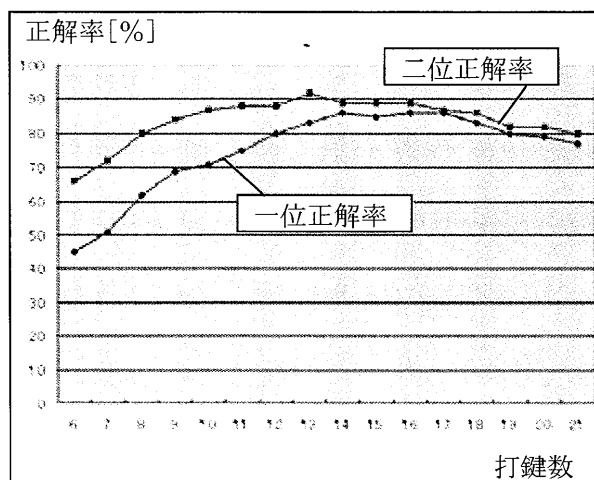


図3：入力打鍵数と正解率

二位正解率が高い理由は、同一リズムや酷似するリズムがリズムパターンDBに存在する場合に、ほぼ正しく打鍵できていても、意図した音楽が二位となるケースが多いためである。

6. 考察

入力にかかる時間は、10打鍵で平均約4秒、20打鍵で約8秒であった。この入力時間は曲により大きなばらつきがあった。

識別を失敗するケースの大半は、MIDIの譜面と入力者のリズムの認識が異なることに起因する。例えば「ジングルベル」というフレーズに対して、利用者は「ジン」、「グル」、「ベル」に合わせて「♪ー♪ー♪ー」と打鍵する場合もあれば、「ジ」、「ン」、「グ」、「ル」、「ベ」、「ル」に合わせて「♪♪♪♪♪ー♪」と打鍵する場合などもあり、同一曲の同一部分についてもリズムの解釈にはかなり幅がある。またどの音を主旋律と捉えるかも人によっては異なるであろうし、修飾音を打鍵すべきかどうかの場合によるであろう。この揺らぎを吸収することが今後の課題であるが、これを吸収しすぎると今回の(3)のアル

ゴリズムであるDPマッチングのように却って精度が落ちてしまうため簡単ではない。DPマッチングの伸縮の制約を厳しくする方式や、それを他の(1)や(2)のアルゴリズムと融合させる方式などが考えられる。

次に多い認識失敗の原因は、打鍵する部分がリズムパターンとしてあらかじめ指定されていないことである。現在は冒頭以外に数箇所のサビの部分、打鍵開始位置としてリズムパターンDBに登録しているが、それが入力者の打鍵と一致しなくては正しい認識が行われない。全箇所で行う方式(B)は妥当ではないと判断したが、マッチングの対象部分を設定する方針や方式が今後の課題となる。

7. まとめ

本稿では、タタタップによる音楽検索の可能性を実験に基づいて示した。音楽の検索方式として実用するためにはより高い精度が求められる。先に挙げたさまざまな問題を解決し、意図した音楽を検索する精度を高める必要があるが、原理的に異なる局が同一リズムで構成されていればタタタップでの識別は不可能である。そこで、大量の音楽データを用いてリズムに基づく判別可能性がどの程度あるのかを理論的に算出することも重要な課題である。

そして音楽検索システムとしての実用を考える場合に何より重要かつ困難となるのが、リズムパターンDBの作成方法である。今回はMIDIから主旋律を手動で抜き出した上で、サビの数箇所にマーキングをする方式が有効であった。しかしより多数の曲に適用する場合には自動化が望ましく、サビ部分を検出するアルゴリズムなどの活用があるかもしれない。

その他にも、より多くの曲による実験、曲からリズムを抽出する周辺技術の確立、携帯電話での着メロ検索を目指した携帯電話での実証などが今後の課題として挙げられる。

参考文献

- [1] 蔭山哲也, 高島洋典: “ハミング検索を手掛りとするメロディ検索”, 電子情報通信学会論文誌 Vol. J77-D-II, No. 8, 1999
- [2] 池谷直紀, 服部正典, 大須賀昭彦: “リズム入力インタフェース「タタタップ」”, 情報処理学会第66回全国大会 4A-4
- [3] 武田晴登, 篠田浩一, 嵯峨山茂樹等: “リズムベクトルを用いたリズム認識”, 情報処理学会研究報告「音楽情報科学」No. 46, 2002