

パターン認識を用いた音楽情報データベースの圧縮法

土屋 景一

九州芸術工科大学 芸術工学部 音響設計学科

〒815 福岡市南区塩原4-9-1

E-mail: tsuchiya@kyushu-id.ac.jp

本研究はパターン認識法を利用して楽曲中の反復パターンを検出し、音楽情報データベースを圧縮する試みを示したものである。サンプルにロックミュージック20曲を用い、実際に圧縮率を算出した。またパターンの単位長を変化させると、どのように圧縮率が変化するか、単位長と圧縮率の相関も調べた。またパターン認識に際し、類似しているパターンを認識する目的で、「距離」の概念の導入も試みた。

その結果、完全一致型60%、類似型50%という圧縮率が実現した。しかし圧縮率の分散性が大きく、どれだけデータ量を圧縮できるかは、実際に圧縮を終えるまで推測できない、などの問題点が残った。

Compression Method for DataBases of Musical Information by Pattern Recognition

Kei-ichi TSUCHIYA

Kyushu Institute of Design

4-9-1 Shiobaru, Minami-ku, Fukuoka, 815 Japan

E-mail: tsuchiya@kyushu-id.ac.jp

This paper shows a compression method for databases of musical information. Repeated patterns were picked up by using pattern matching method and the note information of each repeated patterns was replaced by a mark of repetition(pointer). Actually, standard MIDI data of 20-pieces were compressed and the ratio of compression were calculated. And in order to improve the ratio of compression, the idea of distance was introduced.

The results show that the compressor attained 60% ratio of compression with completely matching and 50% with approximately matching, if the one bar is taken as a pattern unit. However, the compression methods isn't effective for every kind of music. So we can't exactly know the ratio of compression, until we finish the compression process.

1 はじめに

音楽情報データベースの必要性は、コンピュータの普及率に伴って、年々大きく上昇している。実際にはカラオケやBGMに使用されているが、利用者が目的に応じた曲のデータを選択するためには、あらかじめ大多数のデータがシステムに保存されている必要がある。しかし音楽情報データベースのシステムを構築するには、大多数のデータベースが必要となり、限られたディスク容量では保存できるデータベースの数も限定される。また、データ伝送において、サイズの大きなデータを送受信するには多くの時間を要する。そこで、効率良い保存・伝送を実現するには、個々のデータサイズを小さくする必要がある。

本稿は、音楽情報データベースの圧縮法の試みを示したものである。他の圧縮ツールと併用できるように、ここでは音楽に特化した圧縮法を確立する事を目指した。圧縮に際し、音楽の反復の性質を利用し、パターン認識法で反復部分を検出し、反復部分に情報量の小さな符号を割り当てる過程を用いた。

2 反復パターンを利用した圧縮法

音楽情報データベースを圧縮するために、本研究では音楽の「反復」の性質に着目した。

多くの音楽は、幾つかのあるパターンが繰り返されることによって、一曲が成立している。例えば、ロックミュージックの低音ヤリズム部担当の楽器の進行において、ある小節単位で繰り返す、という性質が見られる。クラシック音楽ではロックの反復の性質に加えて、メロディ部分をはじめ、主題に繰り返しの性質があり、それが一曲中で数回出てくる。ロンド形式、ソナタ形式などが良い

例である。またカラオケ音楽では、歌の部分の2番、3番に相当する部分の演奏は、1番の繰り返し基本となっている。

そこで、反復パターンに相当する部分を、反復記号などの小さな情報量を持った符号に置き換える、という作業を施せば、大幅な圧縮が期待できる。またこの圧縮過程において、反復記号を時間軸上でのポイントとし、元の場所を示すようにすれば、ポイントが示す部分の情報を読みだすことにより、元の情報に戻せる。すると100%の可逆性を持つことができ、情報の削減は全く無いことになる。

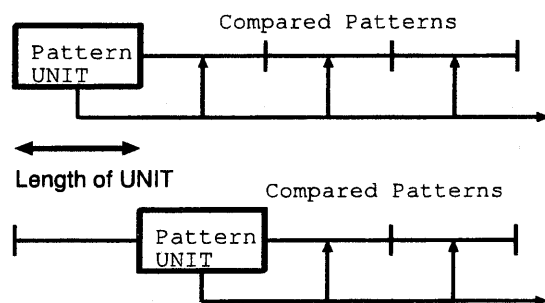


図 1: パターン認識法

反復パターンの検出にはパターン認識法(図1)を用いる。

まず一曲の音列データを、ある単位長毎に分割する。そして、一つめの単位を「基本パターン」、二つめの単位を「被比較パターン」とする。そしてこの二つのパターンを比較する。二つがある条件内にあると認識された場合は、被比較パターンを「反復パターン」と認識する。一致しなかった場合は、三つめの単位を被比較パターンとして基本パターンと比較する。

この作業を繰り返し、一つめの単位と残りの単位とをすべて比較する。それが終わると、二つめの単位を基本パターンとして、他の単位との比較を

行なう。この作業を繰り返すことによって反復パターンを検出していく。

3 完全一致型パターン認識における圧縮法

基本パターンと「音数、音価、音高が全く同じ」と認識された反復パターンを、完全一致型パターンとする。完全一致型パターンの部分の音列データを、「基本パターンと全く同じ」という意の符号（反復記号）に置き換える（図2）。

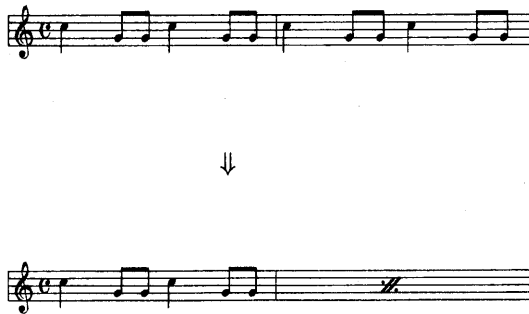


図 2: 反復記号の例

上図において6個の音符の情報が、一つの反復記号で表現でき、この部分で情報量が減少する。

パターン認識を行なうに当たって、まず基本パターン音列データの総情報量を I_A とする。そしてパターン認識を行なった後、曲中の反復パターンの認識数を $n-1$ 個とすると、基本パターン音列データを m 種に変えた場合、次のような数式で一曲の音列データの総情報量を表現できる。

$$I_{A_1}n_1 + I_{A_2}n_2 + I_{A_3}n_3 + \dots + I_{A_m}n_m = \sum_{l=1}^m I_{A_l}n_l \quad (1)$$

反復記号の情報量がほとんど0であるとすれば、各 n は1とできるので、圧縮後の情報量 $I_{A_{ft}}$ 及

び圧縮量 I_{Com} は

$$I_{A_{ft}} = \sum_{l=1}^m I_{A_l} \quad (2)$$

$$I_{Com} = \sum_{l=1}^m I_{A_l}(n_l - 1) \quad (3)$$

となる。これより、基本パターンの情報量が大きければ大きいほど、完全一致型パターンの出現個数が多ければ多いほど、圧縮量及び圧縮率は高くなる。

これに基づき、実際に圧縮を行なった。圧縮に用いたデータはロックミュージック20曲で、データ形式は標準 MIDI データである。

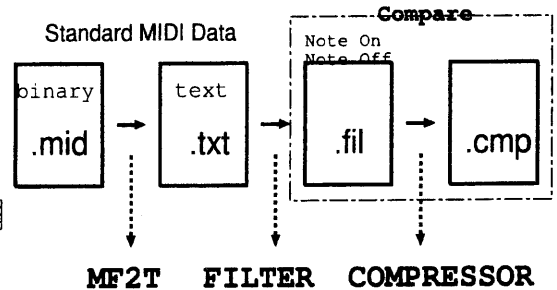


図 3: データ変換過程

標準 MIDI データはプログラムを簡便にするため、2段階のデータ加工が施行され（図3）、その後には圧縮が行われる。まず最初に、標準 MIDI データを Piet Van Oostrum 製作の MF2T というソフトを用いて、バイナリ形式からテキスト形式に変換する。次に自作のフィルタを用いて、ノートオン、ノートオフ情報のみをピックアップする。それらの情報より、音符の長さに相当する音長情報を計算する。そしてカウント情報、チャンネル情報、音高情報、音長情報を4つ横に並べて、.fil ファイルに記述する。そして、この4種の情報が記述されたデータを圧縮する。圧縮されたデータは.cmp ファイルに記述される。

実際の圧縮率を示すために、圧縮前のデータ (.fil) の情報量と圧縮後のデータ (.cmp) の情報量を比較した。圧縮率 I は

$$I = \frac{\text{圧縮後の情報量(Bytes)}}{\text{圧縮前の情報量(Bytes)}} \times 100 \quad (4)$$

として計算した。

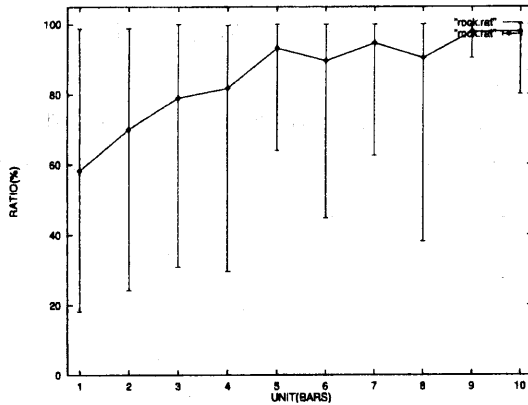


図 4: 完全一致型の圧縮結果

結果 (図 4) を示す。グラフの縦軸は圧縮率 I を示している。横軸は単位長を示し、パターンの単位長の、曲中での小節数を示す。また折れ線グラフは全 20 曲の圧縮率の平均値を示し、棒グラフは 20 曲中の最大値と最小値を採ったときの、圧縮率の範囲を示す。

その結果、単位長 1 小節で平均 60%、という圧縮率が実現できた。また単位長を小さくした方が、平均的に高い圧縮率を得られている。しかし単位長を 4 小節、あるいは 8 小節に採ったときには、圧縮率が少し低くなっている。これは、4、8 小節毎のまとまりを持った動きが顕著である、ロックミュージックをサンプルデータに用いたことにより、生じたものであると考えられる。これより、圧縮率は音楽の特性に強く依存する、ということになる。また、今回用いたデータ 20 曲に

については分散性が大きく、すべてにおいて良い圧縮率が得られるわけではない、という結果になった。サンプルの中には単位長によらず、全く圧縮できない曲もあった。よって、実際に圧縮し終えるまで、どれだけデータ量が圧縮できるかが正確には推測できない、などの問題点が残る。

4 類似型パターン認識における圧縮法

ここでは、完全一致型パターンとして認識されなかったパターンにおいて、条件を緩めて認識パターン数を向上させるため、「距離」の概念を導入する。i 番めの基本パターン x_i と j 番めの被比較パターン y_j の間の距離は、次のように表現する。

$$d_{ij} = d(x_i, y_j) \quad (5)$$

パターン認識の際、

- $d_{ij} = 0$ のとき完全一致型パターン
- $d_{ij} > D$ のとき不一致型パターン
- $d_{ij} \leq D, d_{ij} \neq 0$ のとき類似型パターン

(D は定数)

とする。

類似型パターンであると認識された反復パターンについては、それを反復記号と付加情報 (基本パターンと異なっている音符のみの情報) (図 5) で表現する。



↓



図 5: 反復記号と付加情報

今回は、距離については一つの数値で示さず、次のような段階を置くことによって採った。

1. 基本パターンの音符数に対し、被比較パターン内の音符数の割合が $\frac{3}{4} \sim \frac{4}{3}$ のものを採り入れる。それ以外は「不一致型パターン」とする。
2. 両パターンの音符を初めの音符から一つずつ比較していき、2つの音符が違ふとき、異音数を1カウントする。
3. 異音数が基本パターン内の音符数の半数以内なら「類似型パターン」、半数より大きければ「不一致型パターン」とする。

距離を考慮して実際に圧縮を試みた結果を図6に示す。他の条件は、完全一致型のとくと全く同じである。

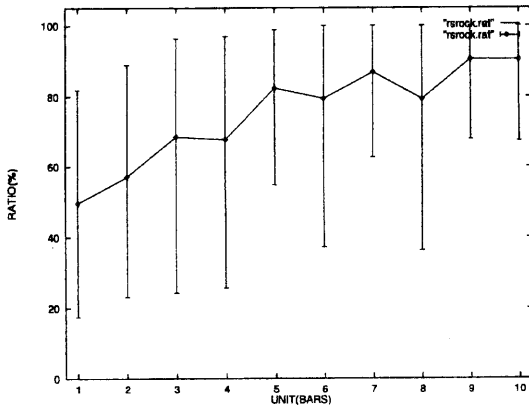


図 6: 類似型の圧縮結果

全単位長にわたって、先の完全一致型パターン認識の結果より、約10%低い圧縮率が得られた。よって類似型認識を行なうことによってより良い圧縮率が得られる、ということが言える。特に単位長が4、6、8小節のとき、完全一致型との差が顕著である。これはロックミュージックをサンプル曲に用いたので、このジャンルの曲によく出現する、フィル・インの奏法の部分を検出したためと思われる。また完全一致型と同様に、単位長が短い時のほうが、良く圧縮できている。これは、単位長が短い程、基本パターンがより多くなり、それにあてはまる反復パターンも多くなるからである。同様に単位長が長い程、「反復パターン」とみなされるものが、少なくなる傾向があるからである。またここでも分散性が大きく、全ての曲で良い値が得られる、とは言えない。

5 まとめ

音楽情報データベースの圧縮のために、パターン認識法を利用し、反復パターンを情報量の小さな符号で置き換えた。実際に圧縮を行なったところ、単位長1小節で60%という圧縮率が実現した。また、距離概念を導入した圧縮法では、単位長1小節で50%にまで圧縮率を低くすることができた。ただ、分散性については完全一致型類似型共に大きく、汎用性はあまり持ち得なかった。今後は距離と圧縮率の相関を調べたり、あらゆるジャンルの多くの曲のデータを用いて、圧縮率の音楽性での違いを調査する必要がある。

6 今後の課題

今後の課題として、次に示すことが挙げられる。

- 距離を変化させて、圧縮に有効な距離の値を得る。
- サンプルデータに、より多くの、あらゆるジャンルの音楽データを使用する。
- 階層構造を考慮する。
- パターンの様々な変形を、認識要素に入れる。

参考文献

- [1] Jeff Rona 著 藤井 美保 訳 “MIDI ブック”
リットーミュージック
- [2] 坂内 正夫 大沢 裕, “画像データベース” 昭晃堂
- [3] 志村 正道, “パターン認識と学習機械” 昭晃堂
- [4] Center for Computer Assisted Research in Humanities, “DataBases of Musical Information” Center for Computer Assisted Research in Humanities
- [5] 奥村 晴彦, “ファイル圧縮技術 (上)” bit 共立出版 Vol.26, No.12 pp.4-13(1994)
- [6] 奥村 晴彦, “ファイル圧縮技術 (下)” bit 共立出版 Vol.27, No.1 pp.53-62(1995)