

自己相関特徴量を用いた圧縮楽曲データからの構造抽出

奥鳴 隆, 大西 建輔, 小早川 倫広, 星 守, 大森 匡

電気通信大学 大学院 情報システム学研究科

〒 182-8585 東京都調布市調布ヶ丘 1-5-1

E-mail: {okunaru, onishi, kobayaka}@hol.is.uec.ac.jp

概要: 本稿では圧縮楽曲データからの楽曲構造抽出手法を提案する。はじめに、MPEG4/audio(TwinVQ)方式で圧縮された楽曲データから算出可能な自己相関係数の系列に対して時間周波数分析を実行し、楽曲全体に現れる強い周期性を検出することにより楽曲の拍長が抽出可能であることを示す。そして抽出された拍長を楽曲分割の基本単位として楽曲を自己相関係数系列上で小節程度の単位に分割可能であることを示す。さらに分割された自己相関係数の各部分系列に対しクラスター分析を実行することにより、分割された各部分を複数のクラスに分類、ラベル付けする。得られたラベル付けが曲の構造と合致していることを示す。

A Structure Extraction of Music from MPEG4/audio Domain

Takashi Okunaru, Kensuke Onishi, Michihiko Kobayakawa, Mamoru Hoshi, Tadashi Ohmori

Graduate School of Information Systems, University of Electro-Communications

1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan

E-mail: {okunaru, onishi, kobayaka}@hol.is.uec.ac.jp

Abstract: In this paper, we propose a method for extracting the structure of music from compressed music data. First, we apply time-frequency analysis to the series of autocorrelation coefficients that are computed from music data compressed by MPEG4/audio(TwinVQ). We show that it is possible to extract the beat by examining the periodicity over the whole musical piece. Moreover, we also show that it is possible to divide the music represented as a series of autocorrelation coefficients into bar units based on the beat as a base unit. Furthermore, we divide the music into multiple clusters by applying cluster analysis to the subseries of the series of autocorrelation coefficients. The result of this clustering is coincide with the structure of music.

1 はじめに

現在、計算機の高速度化や記憶装置の大容量化によって、様々な音楽音響信号が計算機上で扱われるようになってきている。これに加え、MPEG4などの圧縮技術の開発により、扱われる楽曲データはもはや圧縮データであることが当然の状況となっており、急激に増えつつあるこれら圧縮楽曲データに対する効率のよい管理や検索などの要求も高まっている。このような状況を踏まえ、本稿では圧縮楽曲データから楽曲の構造を抽出することを考える。楽曲の構造が明らかになれば、楽曲中の繰り返し部分や類似区間などが分かることとなり、それらを考慮した

より効率的な検索、インデクス作成などが可能となる。このことから本稿は、圧縮された楽曲を対象に、その楽曲の構造を抽出する手法を提案する。

構造抽出としては、通常の音声音響信号から小節構造を認識する後藤ら [4] による研究がある。本稿で示す手法は、圧縮データから得られる特徴量を用いて楽曲の構造抽出を目指すものであり、上記研究とは大きく異なる。

本稿で提案する手法は、楽曲構造分析の最小単位に相当するものとしてその楽曲の拍を求め、それをもとに楽曲を分割する。そして分割された部分に対しクラスター分析を実行し、楽曲の各部分にラベル付けをし、楽曲の構造を分析する。

この手法を適用するにあたって、用いる楽曲は通常市販されるコンパクトディスクに収録されたポピュラー音楽とし、その圧縮方式についてはMPEG4/audio(TwinVQ)[1]とする。また、楽曲の構造抽出を実現するための特徴量は、TwinVQ方式による圧縮の際に得られる自己相関特徴量を用いる。

自己相関特徴量は、MPEG4/audioの入力MDCT(Modified Discrete Cosine Transform)係数のパワー包絡を求めするために用いられる21次元のベクトルである。この自己相関特徴量を用いた楽曲検索手法が墳崎[2]及び、大西ら[3]によって提案されている。

以下に提案手法の概要をまとめる。

1. 楽曲の自己相関特徴量の系列から楽曲分割での基本単位である拍を求める。
2. 自己相関特徴量の系列を拍の長さ(拍長と呼ぶ)の整数倍に等分割する。
3. クラスタ分析により、分割された部分を幾つかのクラスに分けラベル付けし、楽曲の構造を分析する。

本稿では、まず第2節において拍を抽出する手法を示し、そして第3節において楽曲を分割、構造抽出する手法を示す。最後に第4節で考察と問題点について述べる。なお、本稿で用いた音楽用語は[5]を参考にした。

2 拍抽出

2.1 拍抽出概要

楽曲を分割・構造分析するためには全体を妥当な長さで分割する必要がある。本稿では楽曲を分割するにあたって、基本単位を拍とするのがふさわしいと考えた。拍とは楽曲の時間的進行の基準である[5]。

よって、1拍に相当する時間(拍長)を求めることが出来れば、楽曲を様々な単位に分割出来る。本稿では、拍長を以下の手順に従って抽出する。

拍長の算出

step1 自己相関係数の系列から周期を抽出

step2 抽出した周期から拍長の候補を選出

step3 候補間の比により拍長を決定

step1では自己相関係数の系列から楽曲全体で強い周期性をもつ周期を抽出することを考える。本稿では楽曲全体に渡って強く現れる刻みのような周期があると考え。この周期の長さは楽曲の分割に有効な拍長として考えることが出来る。これを発見するため、楽曲における主要な周期を抽出する。

楽曲の音響信号の場合、サンプリング点数が非常に多い(通常の楽曲で1000万から2000万点程度)ので離散コサイン変換(Discrete Cosine Transform、以下DCT)のような時間周波数解析を楽曲全体に対して実行することは現実的ではない。自己相関係数の系列は通常1万から2万点と少ない点数であり、系列全体に対してDCTを実行することも可能である。よって、本稿では楽曲全体に対してDCTを実行することで、楽曲全体にわたって現れる強い周期性を検出する。

step2では、得られる周期から拍長の候補の絞り込みを行なう。得られる周期の中から拍長として使える値は、ある程度の範囲に絞り込むことが可能である。楽曲の演奏速度を1分間あたりの拍数として、"beat per minute"、bpmという単位で表現すると、多くの楽曲は60bpmから200bpm程度の範囲で演奏される。これは、1拍あたりの秒数にして $\frac{3}{10}$ 秒から1秒の範囲である。よって本稿では、抽出する拍長の上限 t_{max} と下限 t_{min} を設ける。step1の結果得られた周期からこの制約の範囲内の周期を拍長の候補とする。

step3では候補の中から拍長を決定する。自己相関係数の系列から得られる周期には同程度の周期性をもつ異なる周期が複数あることもある。このような場合にどれを選択するかが問題となる。この問題を解決するために楽曲中の主要な周期間の比の関係に着目した。

音符の音長間には明らかな比例関係がある。例えば、全音符、2分音符、4分音符、8分音符の音長の間には8:4:2:1といった2のべき乗の比がある。実際の楽曲データでも音の長さはこの比に強く従うと考えられる。このことから、楽曲中には拍長に相当する周期の他に、その半周期、倍周期、4倍周期などに強い周期性があると予想される。この周期に対応した長さを便宜上本稿では、半拍長、倍拍長、4

倍拍長と呼ぶことにし、拍長の候補の絞り込みのためにこれらについても拍長と同様に候補を選び出す。

ここでは、拍長の候補以外に拍長決定の補助として、半拍長と4倍拍長の候補を求める。この場合、求めた半拍長、拍長、4倍拍長候補の要素の組合せの中には、比が $\frac{1}{2} : 1 : 4$ に非常に近い組合せがあるはずである。そのような組合せを与える拍長候補をこの楽曲における各拍長として求める。

以上の手順で拍の長さを得る。

2.2 拍抽出アルゴリズム

拍長を t_1^* とし、これを求めるアルゴリズムを説明する。

今、楽曲が N フレーム、(1フレーム α 秒)からなるととき、各フレーム毎に1次から20次の自己相関係数を求める。 i 次の系列

$$c_i = \{c_{i,1}, c_{i,2}, \dots, c_{i,N}\}$$

は時系列データとみなすことが出来る。

ここで、 i 次の自己相関係数系列 c_i に対してDCTを実行して得られるDCT係数列 d_i を

$$d_i = \{d_{i,m}\}, m = 1, \dots, N$$

$$d_{i,m} = \sum_{n=1}^N c_{i,n} \cos \frac{m(n+1/2)\pi}{N}$$

とする。 $d_{i,m}$ は c_i と周期 $\frac{N}{m}$ フレームの余弦波との相関の大きさを示す。この相関が大きいとき、 c_i には周期 $\frac{N}{m}$ フレームの強い周期性があるといえる。フレームを秒に変換すれば周期 $\frac{N}{m}$ フレームは $\frac{N\alpha}{m}$ 秒となる。

第2.1節で述べた制約から、拍長の候補は $t_{min} \leq \frac{N\alpha}{m} \leq t_{max}$ に存在し、対応するDCT係数の次数は

$$\frac{N\alpha}{t_{max}} \leq m \leq \frac{N\alpha}{t_{min}}$$

の範囲内の次数である。その範囲の次数の集合を

$$f_{i,1} = \left\{ d_{i,m} \left| \frac{N\alpha}{t_{max}} \leq m \leq \frac{N\alpha}{t_{min}} \right. \right\}$$

と記す。その中から

$d_{i,m}$ を絶対値の大きい順に K 個を取り出し、そのその際の次数をそれぞれ m_1, m_2, \dots, m_K とすると、拍長の候補として $\left\{ \frac{N\alpha}{m_1}, \dots, \frac{N\alpha}{m_K} \right\}$ を得る。これ

らを c_i における t_1^* の K 個の候補とし、 $t_{i,1}$ と記す:

$$t_{i,1} = \{t_{i,1,1}, t_{i,1,2}, \dots, t_{i,1,K}\}$$

c_i から得られる拍長の候補 $t_{i,1}$ の中から最終的に一つを決定するために、半拍長、4倍拍長の候補も求める。ここで言う半拍長 $t_{\frac{1}{2}}^*$ 、4倍拍長 t_4^* とはそれぞれ、拍長の半分、拍長の4倍であり、

$$t_{\frac{1}{2}}^* : t_1^* : t_4^* = \frac{1}{2} : 1 : 4 \quad (1)$$

という拍長の関係が成立する。楽曲中ではこれらの長さには強い周期性があると考える。拍長 $t_{\frac{1}{2}}^*, t_4^*$ の取り得る範囲は

$$\frac{1}{2}t_{min} \leq t_{\frac{1}{2}}^* \leq \frac{1}{2}t_{max}, 4t_{min} \leq t_4^* \leq 4t_{max}$$

である。このときに対応するDCT係数の集合は、それぞれ

$$f_{i,\frac{1}{2}} = \left\{ d_{i,m} \left| \frac{2N\alpha}{t_{max}} \leq m \leq \frac{2N\alpha}{t_{min}} \right. \right\},$$

$$f_{i,4} = \left\{ d_{i,m} \left| \frac{N\alpha}{4t_{max}} \leq m \leq \frac{N\alpha}{4t_{min}} \right. \right\}$$

である。よって、 $f_{i,1}$ と同様に、 $f_{i,\frac{1}{2}}, f_{i,4}$ の中から $d_{i,m}$ を絶対値が高いものから順に K 個取り出し、それぞれの次数に対応した周期の集合を $t_{i,\frac{1}{2}} = \{t_{i,\frac{1}{2},k}\}, t_{i,4} = \{t_{i,4,k}\} (k = 1, \dots, K)$ として求める。

今、各次数について $t_{i,\frac{1}{2}}, t_{i,1}, t_{i,4}$ を得たので、これらからそれぞれ一つずつ要素 $t_{i,\frac{1}{2},x}, t_{i,1,y}, t_{i,4,z}$ を取り出し、その比が最も、 $\frac{1}{2} : 1 : 4$ に近づく組を求める。そして、その組を i 次における拍の組 $(t_{i,\frac{1}{2}}^*, t_{i,1}^*, t_{i,4}^*)$ とする。

以上を各次数 $i (= 1, 2, \dots, 20)$ について行なう。そして、 $(t_{i,\frac{1}{2}}^*, t_{i,1}^*, t_{i,4}^*)$ の中で最も比が $\frac{1}{2} : 1 : 4$ に近づく組を $t_{\frac{1}{2}}^*, t_1^*, t_4^*$ とし、この楽曲における半拍長、拍長、4倍拍長とする。以上がアルゴリズムである。図2.2にこれを手続き形式としたものを示す。

2.3 拍抽出実験

拍抽出アルゴリズムを実装し、表1に示す楽曲全10曲に対し拍抽出を実行した結果を示す。なお、表中の拍長は、楽譜に記載されたテンポ(例えば $\text{♩} = 120$)から逆算した、拍長(秒)である。

Procedure detect-beat(C, t, K, t^*)

入力 自己相関係数系列 $C = (c_1, \dots, c_{20})$, $c_i = \{c_{i,1}, \dots, c_{i,N}\}$, $N = |c_i|$

拍長の制約区間 $t: t_{min} \leq t \leq t_{max}$

各次数で列挙する拍候補数: K

出力 拍の長さ: t^*

for $i \leftarrow 1$ to 20 do

i 次の自己相関係数系列 c_i に DCT を実行し、DCT 係数

$d_i = \{d_{i,m} \mid d_{i,m} = \sum_{n=1}^N c_{i,n} \cos \frac{m(n+1/2)}{N} \pi, m = 1, \dots, N\}$ を計算;

for each $l \in (\frac{1}{2}, 1, 4)$

次数 i の半拍、拍、4 倍拍の長さに対応する、DCT 係数集合

$f_{i,l} = \{d_{i,m} \mid \frac{1}{l} \frac{N\alpha}{t_{max}} \leq m \leq \frac{1}{l} \frac{N\alpha}{t_{min}}\}$ を計算、ただし $\alpha = \frac{1 \text{ 秒}}{1 \text{ フレーム}}$;

$f_{i,l}$ 内で $d_{i,m}$ を絶対値の大きい順に K 個を取り出し、その次数に対応した K 個の周期の組 $t_{i,l} = \{t_{i,l,1}, \dots, t_{i,l,K}\}$ を計算;

end each

for $p \leftarrow 1$ to K do

for $q \leftarrow 1$ to K do

for $r \leftarrow 1$ to K do

評価値 $e_{i,p,q,r} = \left| 1 - \frac{8t_{i,\frac{1}{2},q}}{t_{i,4,p}} \right| + \left| 1 - \frac{4t_{i,1,r}}{t_{i,4,p}} \right|$ を計算;

end for

end for

end for

end for

評価値 $e_{i,p,q,r}$ の最小値を与える i, p, q, r を i^*, p^*, q^*, r^* とする;

$(t_{\frac{1}{2}}^*, t_1^*, t_4^*) = (t_{i^*, \frac{1}{2}, p^*}^*, t_{i^*, 1, q^*}^*, t_{i^*, 4, r^*}^*);$

$t^* = t_1^*$

end procedure

図 1: 拍抽出手続き

これら楽曲は全て $\frac{4}{4}$ 拍子の邦楽歌謡曲であり、市販されている CD に収録されている楽曲である。これを TwinVQ 方式で圧縮し、その際に各楽曲ごとに自己相関係数の系列を抽出した。TwinVQ 圧縮に使用するプログラムは、"MPEG4 audio Reference Software, Verification Model ver1.2" を用いた。拍長のとり得る範囲は $t_{min} = \frac{3}{10}$ (秒) から $t_{max} = 1$ (秒) とし、各次数毎で列挙する拍の候補の数は $K = 10$ 個とした。

以上の条件でアルゴリズムを適用した。図 2 は曲 1 の 20 次の自己相関係数に DCT を実行した結果のパワースペクトルである。曲 1 の 20 次における $f_{20, \frac{1}{2}}, f_{20, 1}, f_{20, 4}$ はそれぞれ、 $f_{20, \frac{1}{2}} = \{d_{20, m} \mid 65 \leq m \leq 259\}$ 、 $f_{20, 1} = \{d_{20, m} \mid 259 \leq m \leq 864\}$ 、 $f_{20, 4} = \{d_{20, m} \mid 518 \leq m \leq 1727\}$ となった。この範囲からそれぞれ半拍長、拍長、4 倍拍長の候補を選んだ。この図 2 で分かるように DCT の次数が 210, 478, 1434 あたりに鋭いピークがたつことが分かる。このこと

表 1: 実験に用いた楽曲:表中の拍長は楽譜に記載された1分間あたりの拍数から逆算した1拍あたりの秒数

曲番号	曲名	拍長 (秒)
1	ロビンソン	0.541
2	青い車	0.492
3	渚	0.492
4	チェリー	0.619
5	冷たい頬	0.492
6	運命の人	0.566
7	楓	0.750
8	愛の言霊	0.583
9	名も無き詩	0.480
10	マシンガンをぶっ放せ	0.435

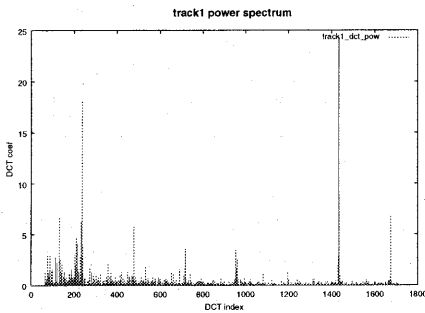


図 2: 曲 1 の 20 次での自己相関係数時系列から得られる DCT パワースペクトル

より曲 1 における拍抽出の結果、 $t_1 = 0.542$ を得た。抽出した拍について評価するにあたって、楽譜に記載された拍は実際の演奏と異なることも考えられる。本稿では 10 曲の楽曲に対し、手作業による拍の抽出を行った。その方法は楽曲中の小節数を、楽譜を参考にしながら数え上げ、全体の時間から拍を逆算するというものである。この際、曲の開始直後から曲の終了(音量が 0 になる)までほとんど無音の部分も演奏が続いていると想定し、小節を数えた。曲の終了部分において小節の途中で曲が終っていた場合、小節の半分以上を演奏し終えているならこれを 1 小節として切り上げて数えた。

この結果から計算された実測による拍と、抽出された拍との比を計算した結果を表 2 に示す。

この結果 10 曲中 5 曲に関しては拍長がほぼ正確に抽出出来たと言える。残りの楽曲に関しては、実際の拍長に対して半分の拍長、倍の拍長となったものがそれぞれ 2 曲、3 曲であった。拍長が一致しなかつ

表 2: 拍抽出結果: 各楽曲における、実測より得た拍長、及び抽出された拍の拍長、およびその比(表中ではそれぞれ、“実測(秒)”、“抽出(秒)”、“抽出実測(比率)”と表記)

曲番号	実測(秒)	抽出(秒)	抽出実測(比率)
1	0.543	0.542	0.998
2	0.492	0.492	1.000
3	0.487	0.977	2.006
4	0.619	0.308	0.498
5	0.488	0.963	1.973
6	0.578	0.563	0.974
7	0.749	0.376	0.502
8	0.584	0.582	0.997
9	0.477	0.482	1.010
10	0.437	0.877	2.007

た 5 曲に関してもほぼ $\frac{1}{2}$, 2 倍であり、半拍、倍拍として検出されたと言える。この結果から、本稿が提案する手法は、楽曲分割の単位としての拍を発見するという目的には非常に有効であると思われる。

3 楽曲構造抽出

前節のアルゴリズムで抽出された拍を用いて、楽曲構造を抽出する。

我々は求めた拍長と拍子から小節の長さを計算し、楽曲を n 個の小節に相当する長さで楽曲の自己相関係数の系列を等分割することを考える。そして分割された部分同士の距離を計算しクラスタ分析を行うことで、全体をいくつかのクラスにわけ、類似区間や繰り返し構造の発見を目指す。

1 小節の長さは、拍の長さから得ることが出来る。拍子は基本とする音符が 1 小節中何個あるかということを示す(楽譜の拍子記号は基本とする音符の種類を分母、小節中の個数を分子となるような分数の形である)。本稿では、拍子の情報は既知であるとして構造抽出を行なう。

はじめに第 3.1 節において楽曲分割について解説し、第 3.2 節で分割された部分の距離計算、およびクラスタ分析による構造分析について説明する。第 3.3 節において楽曲の構造分析実験結果について述べる。

3.1 楽曲分割

拍長と拍子から小節の長さを求め、楽曲の自己相関係数の系列を λ 小節毎に分割する。

分割を考えるにあたって、どの次数の自己相関係数系列について分割するかを考える必要がある。本稿ではもっとも式 (1) に近い比を持つ各拍長が得られた次数の系列を用いて分割を行なう。その次数を i^* とする。

我々は i^* 次における自己相関係数の系列を c^* とおき、また、

$$c^* = \{c_{i^*,1}, \dots, c_{i^*,N}\}$$

として、この c^* を等分割する。次に、 c^* を λ 小節に分割するために、1 小節の長さ U_t を求める。拍子の分子は小節が何拍から構成されているかを示す値であり、今この拍子の分子が u で与えられているとすると、小節の長さ U_t は

$$U_t = u \cdot t_1^*$$

として得られる。

この U_t の単位は時間であり、これをフレーム単位に変換した U_c は、

$$U_c = \left\lfloor u \cdot \frac{t_1^*}{\alpha} \right\rfloor$$

として表され、1 小節の長さは U_c フレームとなる。これより、 λ 小節の長さは $\lambda \cdot U_c$ フレーム、またこのとき c^* は $\left\lfloor \frac{N}{\lambda \cdot U_c} \right\rfloor$ 等分に分割される。

c^* を $\left\lfloor \frac{N}{\lambda \cdot U_c} \right\rfloor$ 等分割したときの第 p 部分系列を P_p と記す:

$$P_p = \{c_j^* | (p-1) \cdot \lambda \cdot U_c \leq n \leq p \cdot \lambda \cdot U_c, j \text{ は自然数}\} \\ \text{ただし } p = 1, 2, \dots, \left\lfloor \frac{N}{\lambda \cdot U_c} \right\rfloor$$

を得る。

3.2 ラベル割り当てによる構造分析

部分系列 P_p の間の距離を計算し、その結果をクラスタ分析にかけ、デンドログラムを求めることで構造を分析する。

P_x と P_y との距離 $\delta_{x,y}$ は以下の式で表わされるユークリッド距離とする。

$$\delta_{x,y} = \left(\sum_{q=0}^{\lambda \cdot U_c - 1} (c_{x,\lambda \cdot U_c + q}^* - c_{y,\lambda \cdot U_c + q}^*)^2 \right)^{\frac{1}{2}} \\ 1 \leq x < y \leq \left\lfloor \frac{N}{\lambda \cdot U_c} \right\rfloor$$

表 3: 実際の小節数と抽出した拍長より得た小節数の比較表 (表中ではそれぞれ順に“実測”、“抽出”として表記する)

曲番号	実測	抽出	抽出実測
1	120	121	1.008
2	143	144	1.007
3	148	74	0.500
4	106	213	2.009
5	125	64	0.512
6	135	139	1.030
7	109	218	2.000
8	146	147	1.007
9	172	171	0.994
10	152	76	0.500

そして、この $\delta_{x,y}$ をもとにクラスタ分析を行う。

クラスタ分析の手法としては最遠距離法を用いた。この最遠距離法は、平均法や最短距離法によるクラスタ分析と比べて、鎖効果と呼ばれる現象 (一つの大きいクラスタと、それ以外の部分とが一つづつ併合されていき、各クラスタが分類出来なくなる) が出にくく、全体的に大きな複数のクラスタができて、全体を分類しやすい。

クラスタ分析の結果、同一のクラスに分類された部分系列 P_p に同じラベルを割り当て、楽曲を少数のラベルで表されるパターンの系列に変換する。これにより楽曲の構造を分析する。以下に、分析の結果をまとめる。

3.3 分析結果

第 2.3 節において求めた拍、及び、楽曲が $\frac{4}{4}$ 拍子であるという前提のもとに、表 1 の楽曲について自己相関係数の系列上で分割をおこない、クラスタ分析によるラベル付けをおこなった。その結果を以下に示す。

はじめに確認のために、求めた拍の長さから求めた小節数と、第 2.3 節にて行った実測から得た小節数との比較を行なった。この結果を表 3 に示す。結果は第 2.3 節での拍長抽出結果をそのまま反映する形となる。

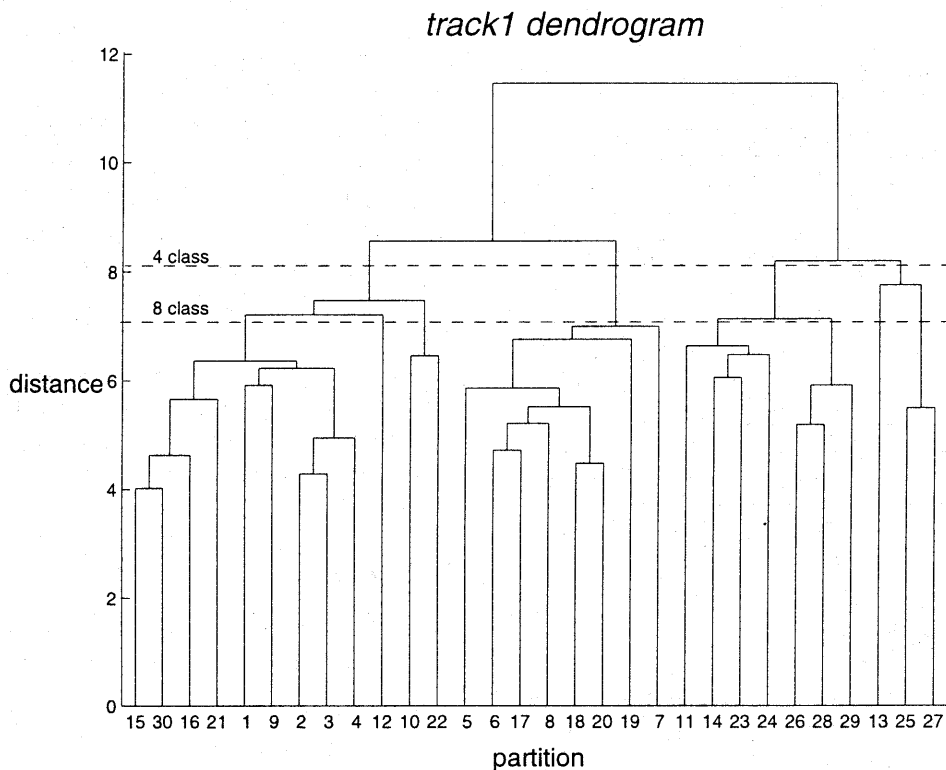


図3: 曲1を4小節単位($\lambda = 4$)で分割した部分系列に対するデンドログラム

続いて我々は、楽曲10曲について4小節単位で分割したデータを作成した。そして、作成したデータに対し数値計算ソフト”Matlab”を使用して最遠距離法によるクラスタ分析を実行した。

曲1に対するデンドログラムを図3に示す。なお、図3の横軸は4小節単位で楽曲を等分割したときの分割部分の番号を示す。例として、部分1は楽曲の第1小節から第4小節に相当し、部分5は第17小節から第20小節にあたる。

このデンドログラムから、曲1を4クラス $a_i (i = 1, 2, 3, 4)$ に分け、各クラスに属する部分系列 p に a_1 から a_4 のラベル付けを行なった。また、同様に曲1を8クラス、 $b_i (i = 1, 2, \dots, 8)$ に分け、ラベル付けを行なった。ラベル付けした結果を表4に示す。

曲1の楽譜を調べて、このラベル付けが実際の楽曲に対して意味のあるものになっているかを確認した。楽譜による情報から対応する部分の構造を調べたところ、部分1から部分4までが導入(イントロ)

部分、部分5から部分8が同じメロディの繰り返しが続く序盤、部分9から部分10が中盤、そして部分11から部分14がいわゆるサビと呼ばれるような終盤部分となっていた。4クラスに分けた際のラベル付けは、構造的に別の部分には別のラベルが割り当てられているということが確認出来る。このことから、曲1に対するクラスタ分析によるラベル付けは曲1の構造に非常によく対応したものとなっていると言える。8クラスに分けた際のラベル付けは、これをより詳細にしたものと言える。

他の9曲に関して同様の実験を行なった結果、クラスタ分析によるラベル付けは楽曲の構造にあったラベル付けが出来ているものが多く、よい実験結果を得た。

表 4: 曲 1 を 4 クラス、及び 8 クラスに分けラベル付けしたときの対応 (表中では、4 クラス時のラベル対応を"4"、8 クラス時の対応を"8"と表記する)

部分番号	4	8	部分番号	4	8
1	a_1	b_1	16	a_1	b_1
2	a_1	b_1	17	a_2	b_5
3	a_1	b_1	18	a_2	b_5
4	a_1	b_1	19	a_2	b_5
5	a_2	b_5	20	a_2	b_5
6	a_2	b_5	21	a_1	b_1
7	a_2	b_5	22	a_1	b_2
8	a_2	b_5	23	a_3	b_3
9	a_1	b_1	24	a_3	b_3
10	a_1	b_2	25	a_4	b_7
11	a_3	b_3	26	a_3	b_8
12	a_1	b_4	27	a_4	b_7
13	a_4	b_6	28	a_3	b_8
14	a_3	b_3	29	a_3	b_8
15	a_1	b_1	30	a_1	b_1

4 考察・まとめ

以上、圧縮楽曲データからの拍長抽出法と、抽出された拍長から楽曲を分割し構造分析する手法を提案し、小規模な実験を行なった。これにより我々は、圧縮された楽曲データからの自己相関係数の系列による拍長抽出は十分可能であること、それをもとに自己相関係数の系列上で楽曲の構造分析が可能であるという見通しを得た。

以下、研究の問題点と今後の課題について述べる。

まず最初に挙げられる大きな問題点として、今回の実験データが非常に小規模であることが挙げられる。これに関しては早急に多様かつ大規模なデータセットを用意し、本発表による手法がどの程度まで有効であるのかを検証する必要がある。また、構造分析に関しては、本稿では詳細な分析及び評価が出来なかった。これに関してもラベル付け以降の適切な分析及び評価法を確立し、実験の結果を正しく評価する必要がある。

これら以外に、拍抽出、構造分析それぞれについていくつかの問題点を残した。拍子の自動抽出は今後の大きな問題である。

また、細かい点ではあるが、拍抽出に関しては、各拍長の候補をいくつ列挙するのか、自己相関特量量の全次数を候補とすべきであるか、比をとるべき拍長はどのようなものが良いのかといった問題がある。

構造分析に関しては、分割の単位に関する問題点

と、クラスタ分析の手法のうちどの分類法が適切であるかどうかという問題がある。本発表では分割の単位を 4 小節単位とした。しかし、この場合 2 小節や 6 小節の繰り返しをもつ楽曲の場合、好ましくない分割となる。このような場合、より細かい単位で分割したのちに、繰り返し構造を調べることで最適な分割単位を発見するという方法が考えられる。

分割部分の分類方法としてクラスタ分析で適切であるかどうかは、今後実験を重ねて確認していく必要があるだろう。

最後に、ラベル付けされた系列が得られたあとの分析法は様々なものが考えられる。本稿では、この最後の過程を十分に検討するに至らなかった。

以上を今後の課題として取り組んで行く。

参考文献

- [1] ISO/IEC JTC 1/SC 20/WG11 N2203" Working Draft of ISO/IEC CD 144963", May 1998.
- [2] 墳崎 英明, "ビットレートに依存しない検索のための MPEG-4 audio TwinVQ データからの特徴量抽出", 電気通信大学大学院情報システム学研究科修士論文, 1999.
- [3] Kensuke Onishi, Michihiro Kobayakawa, Mamoru Hoshi, Tadashi Ohmori: A Feature independent of bit rate for TwinVQ Audio Retrieval, 2001 IEEE International Conference on Multimedia and Expo (ICME 2001) pp.409-412, Tokyo, Japan, 2001.
- [4] 後藤 真考, 村岡 洋一, "音楽音響信号を対象としたビートトラッキングシステム -小節線の検出と打楽器音の有無に応じた音楽的知識の選択", 情報処理学会音楽情報科学研究会報告 97-MUS-21-8, Vol.97, No.67, July 1997.
- [5] 西岡 龍彦, "楽典 -まとめと問題-", アカデミア・ミュージック株式会社, 1986.