2005－MUS－59　(4)
2005／2／18

楽曲圧縮過程において算出される自己相関係数列を用いた楽曲の節長抽出と構造分析

中西基浩[†]　小早川倫広[†]
星守[†]　大森匡[†]

本稿では, TwinVQ 楽曲圧縮過程において算出される自己相関係数列から, 楽曲中の繰り返される節のパターンを検出し, 楽曲の階層構造を分析する手法について述べる. 本手法では, 自己相関系数列から節長を抽出し, これを用いて楽曲を節長単位に分割する. 分割された自己相関系数列の各部分系列を節として扱うことで楽曲の構造分析を行う. 市販の楽曲を用いて実験を行い, 楽曲中の繰り返される大きな区間のパターン, 及びそれを構成するいくつかの小さなパターンを階層的に検出できているかどうかを確認する. その結果から, 抽出された繰り返しパターンは, 階層的に楽曲の構造を捉えていることがわかった.

A Mehod for Extracting Bar Length and Analyzing Stracture of Music in the Compressed Domain of TwinVQ Audio Compression

Motohiro Nakanishi ,[†] Michihiro Kobayakawa ,[†]
Mamoru Hoshi [†] and Tadashi Ohmori [†]

We propose a method for extracting musical length such as bar , half-bar , double-bar, which are musical units to phrase music data, and a method for analyzing a hierachical structure of music in the compressed domain of TwinVQ audio compression (MPEG-4 audio). We extract a musical unit from autocorrelation sequense computed in the encoding step of TwinVQ audio compression, and then phrase the autocorrelation sequence into the subsequences by using the extracted unit. Then, we classify the subsequences into several classes, and assign a label to each class. Thus, we obtain a sequence of labels for a piece of music. Thus, we find a hierachical structure of a piece of music by analyzing the repeated subseqences extracted from the sequence of labels. To evaluate the performance of extraction, we experiment on 64 pieces of music. From the results, we say that the repeated subsequences show the hierachical structure of a piece of music.

1. はじめに

近年, WEB 上では圧縮楽曲データによる音楽配信システムが急激に普及し, 利用者への試聴サービスや, さまざまな音楽検索サービスが提供されている. 提供されている試聴サービスでは, 低ビットレートの試聴用サンプルとして, 楽曲の先頭区間 (約 30 秒程度) が機械的に切り出され, ストリーミング試聴またはダウンロードできるようになっている. このサービスに対する利用者の要求は, サビと呼ばれる楽曲の最も印象深い区間を試聴できることである. より印象深い区間を自動的に切り出し提供するためには, 楽曲中で何度も繰り返される区間や, より特徴的な区間を分析するといったような楽曲の構造分析技術が必須である.

また, 楽曲検索サービスとして, 楽曲の一部分のメロディーを問い合わせとして楽曲の検索が行えるメロディー検索に代表される部分曲検索システムも注目されている. これらの検索システムでは, 部分曲数増加に伴うデータベース側での検索精度や速度性能の劣化が問題となる. そのため, 楽曲中で何度も繰り返される区間を把握し, 楽曲の構造を捉えることで楽曲を効率良く部分曲へと分割する方法が必要とされている.

このように, 楽曲の構造を把握する構造分析技術は, さまざまなサービスを提供するための基本的技術であり, 重要なテーマである. 関連研究としては研究 [1) 2)] などがある.

一方で, 現在流通してる楽曲データは, MPEG に代表される圧縮技術により圧縮され蓄積・管理されることがほとんどであり, 楽曲圧縮技術と楽曲の構造分析

† 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of Electro-Communications

技術に関する技術は合わせて考えるとより効率的であると考える．楽曲圧縮データにおける配信用途の需要の高さにおいては, 楽曲検索技術とも兼ね合わせることが望ましい．楽曲圧縮技術と楽曲検索技術を合わせて考えられた楽曲検索の代表例としては, 塡崎らが提案した圧縮楽曲データから抽出可能な自己相関係数列を特徴量として用いた類似曲検索システムがある [3)4)]．また, この特徴量を利用し楽曲分析を試みた奥鳴らによる楽曲の構造分析がある [5)6)]．これらの他に研究 [7)8)9)10)] などがある．

本研究では, 楽曲圧縮, 楽曲構造分析, 楽曲検索を同時に取り扱えることが重要だと考え, 検索システムなどに有効に活用できる TwinVQ 圧縮楽曲データ [11)12)13)] からの楽曲の構造分析を試みる．

近年の楽曲の多くは, 1 番や 2 番と呼ばれるような楽曲中に最も大きな繰り返し区間をもち, これらの区間は, A メロ, B メロ, サビと呼ばれる区間によって構成されている．さらに, その各々の区間は, より短い繰り返し区間により構成されるというような階層的な構造を有する．このような楽曲の長短さまざまな繰り返し区間からなる階層的な構造の抽出が可能ならば, さらに楽曲の傾向や常識, 個人の主観を踏まえることで, その中のどの区間がサビ区間であるかを特定可能になると考える．

そこで, 本研究では, 楽曲中の長短さまざまな繰り返し区間を抽出し，階層的な楽曲構造の分析を試みる．

2. 楽曲構造分析の枠組

本節では，我々の楽曲構造分析の枠組を示す．提案する枠組の大きな特徴は，楽曲の構造分析を原音響信号から行うのではなく, TwinVQ 圧縮過程で算出される自己相関係数 $r_{k,n}$ (フレーム $n = 1, \ldots, N$, 次数 $k = 1, \ldots, 20$) を時系列データとみなし，自己相関係数列 $\boldsymbol{r}_k = r_{k,1}, \ldots, r_{k,N}$ を分析することにより, 楽曲の構造分析を行うというものである．

提案する楽曲構造分析は，以下の 3 つの手順からなる．

分割単位長の抽出　自己相関係数列 $\boldsymbol{r}_k$ $(k = 1, \ldots, 20)$ から音楽的に意味のある分割単位長を抽出する．

楽曲分割　自己相関係数列から, 分割単位長を用いて部分列へと分割するための分割点を算出し, これを用いて楽曲を部分列へと分割する．

構造分析　分割された部分列間の関係から，楽曲の構造を分析する．

1. 部分列の集合に対しクラスタ分析を実行し, ラベル付けを行う．
2. ラベル列から繰り返し成分を検出する．

上記の枠組上で, 奥鳴らは,

- 楽曲を分割する基本単位として拍長が抽出可能であることを示した [6)]．
- 楽曲が 4/4 拍子であることを既知とし, 抽出した拍長を用いて楽曲を節長単位に等分割し, これに対しクラスタ分析を行い, 得られたラベル列から楽曲構造の分析が可能であるとの見通しを得た [5)]．

しかし，拍子が既知であるとする奥鳴らの手法では，楽曲の拍子抽出が大きな課題として残る．そのような前提なしで楽曲から節長を抽出できれば，音楽的制約の少ない楽曲構造分析が可能となる．

本稿では，節長を分割単位長として考える．以下，節長抽出手法 (3 節)，楽曲分割手法 (4 節)，楽曲構造分析 (5 節) を示す．

3. 節 長 抽 出

楽曲は通常 $50bpm$ から $200bpm$ の範囲で演奏され, この範囲外の演奏速度をもつ楽曲はほとんど存在しない．そのため楽曲の拍と拍の間隔は 0.3 秒から 1.2 秒の間となっている．楽曲の 1 小節の長さ (節長) は，楽曲が 4/4 拍子ならば 1.2 秒から 4.8 秒であり, 3/4 拍子ならば 0.9 秒から 3.6 秒となる．つまり, 4/4 拍子と 3/4 拍子の楽曲を分析するのであれば, 節長の範囲は，0.9 秒から 4.8 秒である．本節では，この音楽的条件を用いて自己相関係数列から節長の抽出を行う．

3.1 節長抽出アルゴリズム

長さ N フレームからなる k 次の自己相関係数列 $(k = 1, \cdots, 20)$ において, j_k 次の自己相関係数 $a_k^{j_k}$ を求める．

$$a_k^{j_k} = \frac{\sum_{i=1}^{N-j_k}(r_{k,i} - \bar{r}_{k,1})(r_{k,i+j_k} - \bar{r}_{k,j_k})}{(N - j_k - 1)\sigma_{k,1}\sigma_{k,j_k}} \quad (1)$$

ただし,

$$\bar{r}_{k,j_k} = \frac{1}{N - j_k}\sum_{i=j_k}^{N-j_k} r_{k,i} \quad (2)$$

$$\sigma_{k,j_k} = \sqrt{\frac{1}{N - j_k}\sum_{i=j_k}^{N}(r_{k,i} - \bar{r}_{k,j_k})^2} \quad (3)$$

ここで, 音楽的条件より，節長として相応しい自己相関係数の次数は, $\lfloor 0.9_{sec} \times f_t \rfloor$ から $\lceil 4.8_{sec} \times f_t \rceil$ の範囲内である．ただし，f_t は, 自己相関係数 1 フレームの時間分解能である．

節長抽出のため, 以下の $\boldsymbol{StepB1} \sim \boldsymbol{StepB3}$ の操作を実行する．

$\boldsymbol{StepB1}$[自己相関係数の算出]: k 次の自己相関係数列

表 1 節長抽出実験の結果.

music ID	bar length (sec)			R_{oh}	R_{os}	music ID	bar length (sec)			R_{oh}	R_{os}
	H	S	O				H	S	O		
01-12	1.738	1.740	1.742	1.002	1.001	93-21	1.684	1.792	1.695	1.007	0.946
02-11	2.500	2.500	2.508	1.003	1.003	93-24	1.608	1.580	1.579	0.982	0.999
03-09	2.055	2.052	2.043	0.994	0.996	94-04	2.470	2.448	2.461	0.996	1.005
13-03	2.336	2.332	2.322	0.994	0.996	94-14	1.964	1.952	3.878	1.974	1.987
13-13	2.400	2.376	4.760	1.983	2.003	94-16	2.222	2.224	2.229	1.003	1.002
20-14	4.796	4.000	1.184	0.248	0.296	94-22	1.892	1.920	1.858	0.982	0.968
20-05	2.180	2.164	2.160	0.991	0.998	94-26	1.792	1.776	1.788	0.998	1.007
20-07	1.908	1.920	3.808	1.996	1.983	95-01	2.132	2.332	4.249	1.993	1.822
20-10	1.756	1.740	1.742	0.992	1.001	95-02	2.032	2.032	4.064	2.000	2.000
20-13	2.824	2.792	2.832	1.003	1.014	95-03	1.972	1.968	1.974	1.001	1.003
23-03	2.108	2.088	4.180	1.983	2.002	95-04	1.320	1.320	1.324	1.003	1.003
23-10	1.968	1.968	1.974	1.003	1.003	95-07	1.892	1.920	0.952	0.503	0.495
24-02	2.444	2.424	2.461	1.007	1.015	95-08	1.328	1.320	3.297	2.482	2.498
24-05	2.408	2.144	4.714	1.958	2.199	95-09	1.848	1.848	1.858	1.005	1.005
24-08	2.172	2.164	2.160	0.994	0.998	95-10	2.312	2.308	4.621	1.999	2.002
25-04	2.428	2.424	1.207	0.497	0.498	95-11	1.380	1.372	1.370	0.993	0.999
25-06	2.696	2.696	2.694	0.999	0.999	96-01	2.072	2.068	2.090	1.009	1.011
31-04	1.948	1.968	3.901	2.003	1.982	96-02	1.596	1.480	3.204	2.008	2.165
31-12	2.476	2.476	2.461	0.994	0.994	96-03	2.065	2.068	2.067	1.001	0.999
32-03	1.952	1.968	3.901	1.998	1.982	96-04	2.552	2.552	2.554	1.001	1.001
32-04	2.264	2.264	2.276	1.005	1.005	96-05	2.136	2.144	2.136	1.000	0.996
32-06	3.015	3.000	2.996	0.995	0.999	97-02	2.464	2.448	2.438	0.989	0.996
32-11	2.120	2.052	4.110	1.939	2.003	97-03	2.523	2.528	2.531	1.003	1.001
65-03	1.852	1.820	3.646	1.968	2.003	97-04	2.564	2.580	1.300	0.507	0.504
65-04	1.995	2.016	1.997	1.001	0.991	97-06	3.243	3.244	3.251	1.002	1.002
65-11	2.320	2.284	4.528	1.952	1.982	97-07	2.582	2.580	2.577	0.998	0.999
93-04	3.080	3.076	1.486	0.483	0.483	97-08	2.608	2.608	2.601	0.997	0.997
93-13	1.281	1.304	1.277	0.997	0.979	97-09	2.529	2.528	2.531	1.001	1.001
93-14	1.809	1.804	1.811	1.001	1.004	97-10	3.629	3.380	3.646	1.005	1.079
93-16	2.394	2.376	2.392	0.999	1.007	97-11	3.817	3.808	3.808	0.998	1.000
93-17	1.740	1.820	1.742	1.001	0.957	20-12	1.902	1.894	1.904	1.002	1.004
93-20	2.434	2.492	2.438	1.002	0.978	98-09	1.459	1.417	1.439	0.986	1.015

において, j_k 次の自己相関係数 $a_k^{j_k}(j_k{=}s_{min}, \cdots, s_{max}$ 算出する.

StepB2[ピーク検出]: ***StepB1*** にて得られた自己相関係列 $(a_k^{s_{min}}, \cdots, a_k^{s_{max}})$ に対し, 2 階微分をかけ微分係数 $d_k^{j_k}(k = 1, \cdots, 20)$ を得る. そして得られた微分係数の中で最大の値 $\max_{k=1}^{20} d_k^{p_k}$ をとる微分係数 $d_k^{p_k}(k = 1, \cdots, 20)$ を選び, 相関次数 p_k を得る.

StepB3[節長の決定]: ***StepB2*** において得られた相関次数 p_k の中で, 最大の値 $\max_{k=1}^{20} d_k^{p_k}$ を選び, その次数 p_i から節長 $l = p_i * f_t$ を算出する.

3.2 実験と評価

節長抽出の性能を評価するために, 64 曲の日本のポップスとロックを用いて実験を行った. ただし, 節長は以下の 3 つの方法で求めた : (1) 音響信号波形から波形ビューアを用いて実際に節長 H を測定, (2) 楽譜に記述されているメトロノーム記号から節長 S を算出, (3) 市販の CD より音響信号へと変換し (44.1kHz/ch), さらに TwinVQ encoder(MPEG-4 audio Reference Software, Verification Model ver 1.2) を用いて 44kbps/ch のビットストリームへと変換を行い, その時算出される自己相関係数列を抽出し, この自己相関係数列に対して本節長アルゴリズムを適用し, 節長 O を抽出.

さらに, 抽出された節長 O と 節長 H (S) との比 $R_{oh} = \frac{O}{H}$ $(R_{os} = \frac{O}{S})$ を算出する (表 1). 比 R_{oh} (R_{os}) は, 節長抽出の性能を表し, 比 R_{oh} (R_{os}) が 1 に近ければ近いほど, 提案した節長抽出アルゴリズムは 1 小節の長さをよく抽出していることを表す. また, 比 R_{oh} (R_{os}) が 0.5 (2) に近ければ近いほど, 提案した節長抽出アルゴリズムは半 (倍) 小節の長さを抽出していることを表す.

ここで, 抽出された節長 O と節長 H の誤差が $\pm 3\%$ 以内であった場合, すなわち, 比 R_{oh} の値が $[0.485, 0.515]$, $[0.97, 1.03]$, $[1.94, 2.06]$ の範囲内であっ

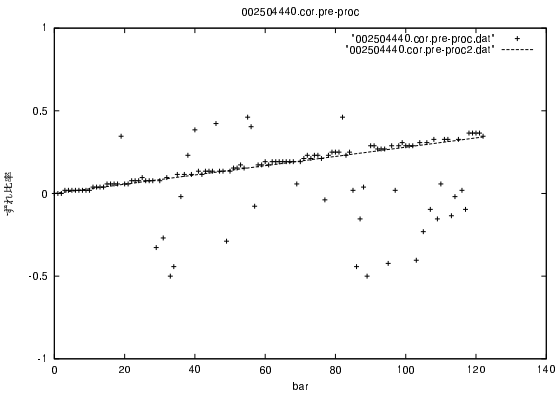

図 1 サンプル点の直線への近似

た場合, 節長抽出 (半節長， 倍節長を含めて) が成功であったとする．抽出に成功した楽曲の総数は，61 曲であった．95.3%($=\frac{61}{64}$) で音楽的に意味のある長さを抽出している．実際, 64 曲中で, 44 曲から 1 小節長，3 曲から半節長，14 曲から倍節長を抽出していた．

4. 楽 曲 分 割

本節では, 前節にて抽出した節長 O を用いて楽曲分割を行う処理について述べる．以下では, 抽出節長 O に相当するフレームの長さを $l_f(=\frac{O}{f_t})$ とする．N フレームからなる k 次 $(k=1,\cdots,20)$ の自己相関係数列を部分系列長 l_f の部分系列へと分割する．

しかし, 自己相関係数列の時間分解能 f_t は $\frac{1024}{44100}(=0.02321\cdots)_{sec}$ であり，とても粗いため, 抽出した節長にて等分割を行うと, 非常にズレが生じ易くそれが蓄積するため, 音楽周期が終止一定である楽曲構造分析を行う際に分析性能を落とす原因となる．

例えば, 100 小節からなる楽曲がある場合, ある程度性能良く節長を抽出しても大抵 3 小節目あたりで 1 フレーム分のズレが生まれてしまう．つまり, 先頭から 1 小節長で等分割した場合, 楽曲の終わりには 30 フレーム分ものズレが蓄積することになる．もし, 1 小節長が 90 フレーム長だとすると, 終りには $\frac{1}{3}$ 小節長ものズレが生じてしまう．

4.1 楽曲分割処理

上記の理由から, 楽曲を長さ l_f の部分系列へと分割することを考える場合, 楽曲を先頭から系列長 l_f にて等分割するのではなく, ズレの蓄積率 ϵ(抽出節長 l_f に対するズレの割合) を算出して楽曲分割を行う．

いま, N フレームからなり, 仮に第 $\dot{M}(=\lfloor\frac{N}{l_f}\rfloor)$ 小節で構成される楽曲があるとすると, 第 m 小節の分割点 $y_m(m=1,\cdots,\dot{M})$ を以下の式 (4) で定める．

$$y_m = \lfloor l_f(m-1) + \epsilon(m-1) + \frac{1}{2} \rfloor \qquad (4)$$

ズレの蓄積率 ϵ は, 1 次から $\frac{N}{2}$ 次までの自己相関係数 $a_i^{j_i}$ のピーク位置と部分系列長 l_f の整数倍とのズレを算出し, これを直線近似 (直線検出) することで得る．

蓄積するズレの補正を行うため, 以下の ***StepS*1** ～ ***StepS*4** の操作を実行する．

***StepS*1**[ピーク検出]: まず, ***StepB*1** にて算出される第 $\tau_i(<\frac{N}{2})$ 次までの微分係数 $d_i^{j_i}$ において, $[m\times l_f$ - $\frac{l_f}{2}$,$m\times l_f + \frac{l_f}{2}]$ の範囲から最も高い値を, 第 m 番目のピーク h_m として抽出する．

***StepS*2**[サンプル点のプロット]: 次に, 算出した h_ω と $l_f\times m$ との差からサンプル点 $s_m(=\frac{h_m - l_f\times m}{l_f})$ を求める．

***StepS*3**[直線近似]: ***StepS*2** で得られた各サンプル点に対し, ハフ変換を用いて直線近似 (直線検出) を行う．検出した直線からズレの蓄積率 ϵ を得る (図 1)．

***StepS*4**[適用判断]: 近似した直線上のプロット数が全体のプロット数の 1 割を下回る場合, ズレの補正を行う．

そして, 式 (4) より分割点 $y_m(m=1,\cdots,\dot{M})$ を始点とする長さ l_f フレームの部分系列を得る．

ただし, 楽曲最後の第 $\dot{M}$ 小節目が l_f フレーム長を持たない場合, これを切り捨て $M(=\dot{M}-1)$ 個 (M 小節) の部分系列を得る．

4.2 実験と評価

本手法の性能を評価するために, 3 節と同一の楽曲をもちいて実験を行った．ただし, 3 節の実験にて, R_{oh} が [0.97, 1.03] の範囲で小節長を抽出できた楽曲 (44 曲) の中で, StepS4 にてズレの補正を行なった楽曲のみ (41 曲) を使用した．評価に, 前節の表 1 に記載される実測小節長 H と抽出小節長 O の値を使用したところ, $(l_f+\epsilon)*f_t$ の値が O の値より, H の値に近い値となる楽曲は, 41 曲中 35 曲であった．蓄積するズレの補正を行うために本手法は有効であったことが言える．

5. 構 造 分 析

本節では, 前節で得られた M 小節に対応する部分系列に対し階層的クラスタ分析を実行することで, 各小節を階層的に分類する．クラスタ分析に使用する非類似度にはユークリッド距離を使用する．任意の第 a 小節目の部分系列と第 b 小節目の部分系列の非類似度 $D(a,b)$ $(1\le b\le a\le M)$ は以下のようになる．

$$D(a,b) = \sqrt{\frac{1}{l_f * 20}\sum_{g=0}^{l_f-1}\sum_{k=1}^{20}(r_{k,y_a+g} - r_{k,y_b+g})^2}$$

得られた非類似度行列を用いて Ward 法によるクラスタ分析を行う．クラスタ分析の結果，類似する部分系列同士がデンドログラム上で近くにまとまって配置される．デンドログラムを頂点から分岐が起こる分岐点にて，分割することにより各部分系列をいくつかのクラスに分類することができる．

5.1 ラベル系列への変換

クラスタ分析にて同一のクラスに分類された部分系列に同じラベルを割り当てると，ラベル系列を得る．そしてラベル系列から楽曲がどのような構造を持つのかを分析する．

楽曲 03-09(曲名は付録を参照) に対応するラベル系列を 4 クラスに分類し，各クラスにラベル"A","B","C","D" を割り当てた（図 2）．図 2 から，22 小節から 58 小節までのラベル系列と，72 小節から 118 小節までのラベル系列が一致しており，この区間に楽曲の 1 番，2 番に相当するような大きな繰り返し構造があることが確認できる．

図 3 は，楽曲 03-09 を 8 クラスにて分類したときのラベル系列を示す．図 3 から，38 小節から 58 小節までのラベル系列と，87 小節から 118 小節までのラベル系列が一致しており，4 クラスにて分類した場合と比べて，より小さな繰り返し構造が確認できる．

また，図 4 は同じ楽曲 03-09 を 16 クラスに分類した場合のラベル系列を示しており，8 クラスで確認した繰り返し構造よりも，さらに小さな区間からなる繰り返し構造が確認できる。以上のことから，楽曲の階層的な繰り返し区間を検出するには，各部分系列をさまざまな分類クラス数にて分析する必要があることが分かる．

5.2 繰り返し区間の検出処理

本節では，デンドログラムをさまざまな距離で分割し，得られたラベル系列において最長一致区間を繰り返し区間として検出する．

ここで，デンドログラムを距離 D_e で分割したときのクラス数を e とすると，第 a 小節目の部分系列と第 b 小節目 $(1 \leq b < a \leq M)$ の部分系列間の距離 $D(a,b)$ が距離 D_e の値よりも小さいとき，第 a 小節目の部分系列と第 b 小節目の部分系列は同じラベルがつけられる．このとき，$f(a,b) = 1 \quad (D(a,b) < D_e)$ と表わし，$G_e = \{(a,b)|\ f(a,b) = 1\}$ とすると，$D_{\hat{e}} > D_e$ ならば，$G_{\hat{e}} \supseteq G_e$ となる．

$$f(a+\tau, b+\tau) = 1 \quad (0 \leq \tau \leq \kappa) \qquad (5)$$

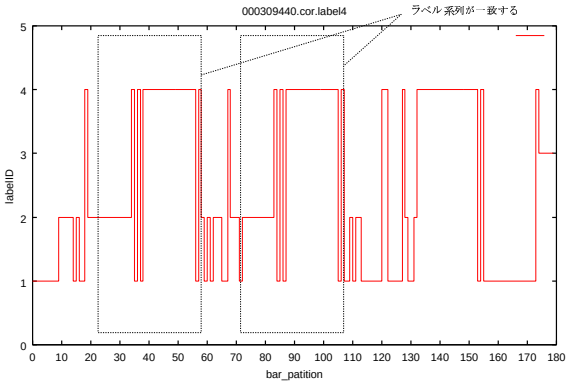

図 2 楽曲 03-09 の 4 クラスでのラベル系列

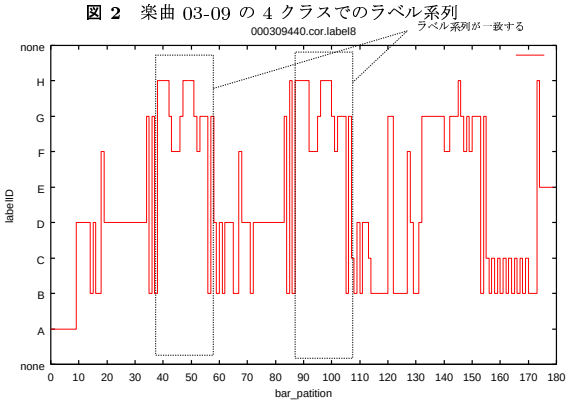

図 3 楽曲 03-09 の 8 クラスでのラベル系列

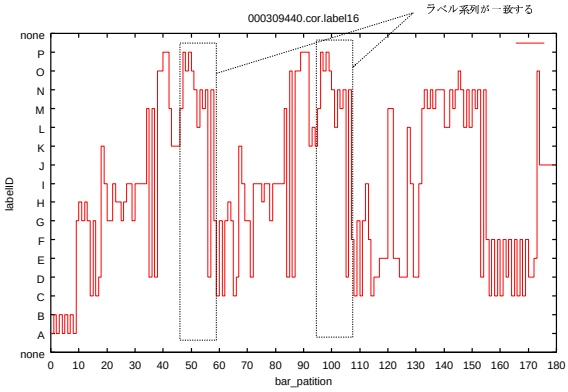

図 4 楽曲 03-09 の 16 クラスでのラベル系列

式 (5) を満たす点の集合 $\{(a+\tau, b+\tau)|\ 0 \leq \tau \leq \kappa\}$ を長さ κ の線分とよび，$((a,b),(a+\kappa,b+\kappa))$ と記す．

デンドログラムを距離 $D_e(D_{e_{max}} \leq D_e \leq D_{e_{min}})$ で分割したとき，得られるラベル系列から最長一致区間を検出する処理は以下のように行う．

***StepR*1** 式 (5) を満たす最大の κ を len とし，線分 $((a,b),(a+len,b+len))$ を検出する．ただし，$b+len \leq a$ 及び $len \geq len_{min} = 4$ とし，検出区間の長さ len に下限 len_{min} を設ける．

このとき, 長さ len の部分系列区間 $[a, a+len]$ と $[b, b+len]$ は同じラベル系列であり, 繰り返し区間となる.

StepR2 ***StepR1*** で, 線分 $((a,b),(a+len,b+len))$ を検出した場合、点集合 $G_{\hat{e}}(D_{\hat{e}} > D_e)$ において点 $(a+p, b+q)(0 \leq p \neq q \leq len)$ を削除する.

5.3 繰り返し区間のグループへの併合と削除

このようにして, 楽曲中で繰り返されている区間を検出した後, 同一区間同士を 1 つのグループとして扱う.

すなわち, 長さ $len(\geq len_{min})$ の繰り返し区間の集合 $S_1 = \{ [b_1, b_1+len], [a_1, a_1+len], \cdots \}$ と $S_2 = \{ [b_2, b_2+len], [a_2, a_2+len], \cdots \}$ が検出された場合, $S_1 \cup S_2 \neq \emptyset$ ならば $S_1 \cup S_2$ と併合する.

また, ある 2 本の重なり合う線分が検出された場合, 重なり合わない部分の長さが len_{min} 以下のときは, より長い方の線分を除去する.

5.4 実　　験

ここでは, $e_{max} = 24, e_{min} = 2$ として実験を行い、検出される繰り返し区間が楽曲の構造に合致することを確かめる。まず, 楽譜, 及び試聴から楽曲を

- 間奏部分 : 楽曲の前奏, 間奏部分.
- 特徴的部分 : 楽曲で最も印象にのこるサビと呼ばれる部分.
- 特殊部分 : 楽曲中に一度しか現れない特殊な段落からなる部分.
- 通常部分 : 上記 3 つ以外の部分 (楽曲の A メロ, B メロに相当).
- 段落部分: 楽曲の 1 番目, 2 番目に相当する大きな段落部分.

に分類し, 5 つの楽曲部分と, 検出される繰り返し区間との対応から楽曲の構造が抽出されたかの確認と議論を行う.

以下では, 上記の 5 つ部分と, 検出される繰り返し区間を同一のグラフに以下のようにして描く.

グラフ上の中央で上下に分ける線を引き, 上側に本手法で検出される繰り返し区間を, 各グループに属する繰り返し区間の長い順に縦軸に対応する $patternID$ を割り振り, 線分として描く. 下側には, 上記の 5 つの部分を, 段落部分を縦軸の $patternID$"-1", 間奏部分を"-2", 通常部分を"-3", 特徴的部分を"-4", 特殊部分を"-5" として同じように線分として描く. 横軸は小節番号となり時間進行を表す. これより先は, 割り振った縦軸の $patternID$ を, 略して ID と呼ぶことにする.

図 5 に楽曲 03-09 に対する実験結果をグラフとして

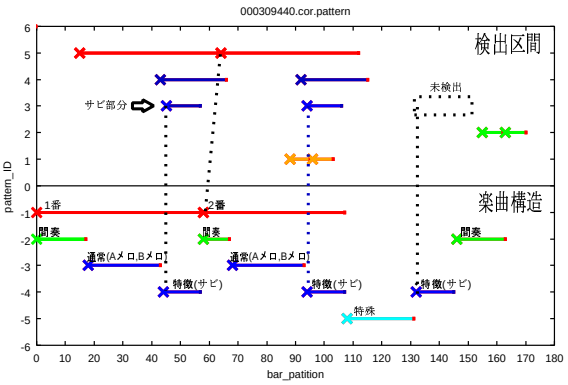

図 5　楽曲 03-09 の検出した繰り返し区間と楽曲構造の対応

示した. 検出された最も大きな繰り返し区間 ID"5" は, 繰り返し区間の先頭位置が大きくずれてはいるが, ID"-1" の段落部分とほぼ重なり合う区間を示している.

また, ID"-4" に示される特徴部分 (サビ部分) は, ID"3" の検出区間と第 44 小節目からと第 94 小節目からの 14 小節長程度で重なりあっているが, 第 132 小節目からの楽曲最後の該当特徴部分は検出できていない. 一般的に, 楽曲最後のサビ部分ではより印象づけるために, 主音を大きく変化させる傾向があり, 自己相関係数列にも, この変化が反映されていると考えられる. 実際に, 図 3 でのラベル系列でも該当サビ区間が他のサビ区間と大きく異なっているため検出が困難である. この傾向は, この楽曲 03-09 だけでなく, いくつかの楽曲でみられた. これを改善するには, ラベル系列の完全一致から繰り返し区間を検出する方法を見直す必要があり, 大きな課題の一つである.

実験に使用した楽曲の検出結果はおおまかに以下の 3 つに分けられた.

- 本研究の目標である階層的な楽曲構造を, 検出した繰り返し区間がよく表わしている楽曲 (図 6).
- 階層的な構造を表してはいないが, おおよそ楽曲の構造と合致している楽曲 (図 7).
- 検出した繰り返し区間が楽曲の構造と合致しない楽曲 (図 8, 図 9).

以降, それぞれの場合について実験結果を示し, 考察をする.

図 6 の楽曲 96-02 は, 楽曲の階層構造を繰り返し区間を検出することにより, よく表わすことができている. 楽曲の 1 番, 2 番のような大きな繰り返し区間と, それを構成する A メロ, B メロ, サビからなる区間, さらにこれらの区間を構成する区間までもが検出できており, 楽曲の構造と合致している.

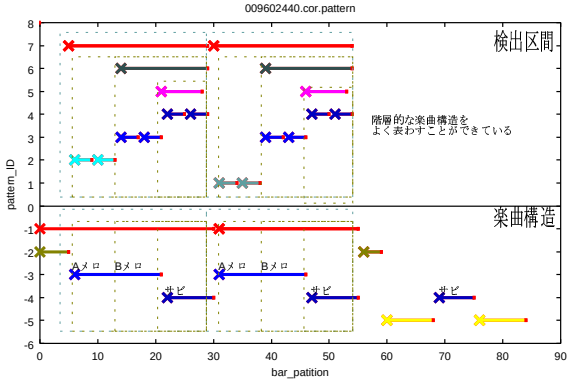

図 6　楽曲 96-02 の検出した繰り返し区間と楽曲構造の対応

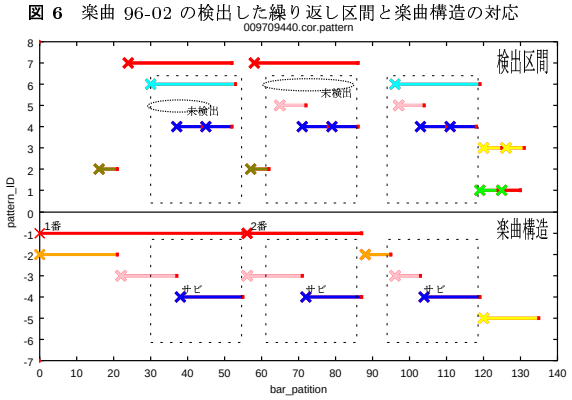

図 7　楽曲 97-09 の検出した繰り返し区間と楽曲構造の対応

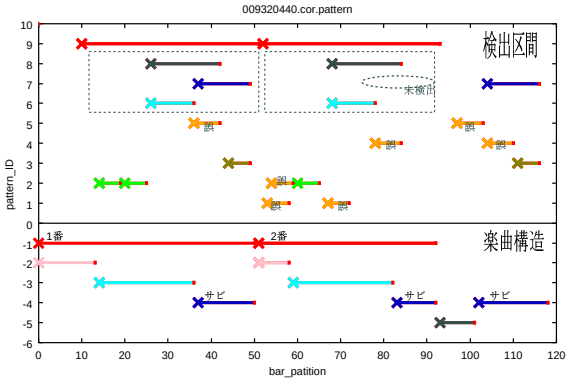

図 8　楽曲 93-20 の検出した繰り返し区間と楽曲構造の対応

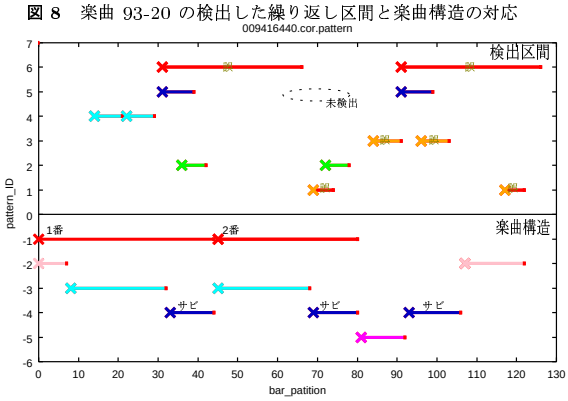

図 9　楽曲 94-16 の検出した繰り返し区間と楽曲構造の対応

図 7 の楽曲 97-09 は, 楽曲の構造と, 検出した繰り返し区間がおおよそ合致している. しかし, 検出されると予想した区間の検出ができていない部分が目立った. この問題に対しては, 繰り返し区間を検出し, グループへと併合した後, 各繰り返しグループ間の相互関係から, 欠落している区間を調べあげる方法が有効ではないかと考える.

図 8 の楽曲 93-20 は, 他のグループの繰り返し区間との相互関係から欠落していると認識できる区間が目立つ. また, 楽曲の構造に合っていない区間同士が同じグループとして検出されている. これは, 少ないクラス数での分析にて検出された誤った区間だと考えられる.

図 9 の楽曲 94-16 は, 最もな大きな繰り返し区間である ID"6" のグループが段落部分の ID"-1" と大きく異なる位置で検出されてしまっている. また, 他のグループの区間も楽曲の構造と合致しない. この楽曲のラベル系列を観察したところ, ラベル系列が段落的な形になっておらず, 楽曲構造が確認できなかった. この場合, 楽曲を分割する基本単位長に問題があると考えられる. このようなラベル系列が段落的な形にならない楽曲は, 楽曲構造を壊さない程度の, より長い単位長 (節長の整数倍) を用いることで分析性能を上げる必要がある. しかし, 分割単位長を楽曲ごとに決定付けることは非常に難しい問題である.

6. ま と め

本研究では, 楽曲圧縮と楽曲構造分析を同時に考えることを重要とし, 楽曲圧縮過程で算出される自己相関係数列を用いて楽曲の階層的な構造を分析することを試みた. そして, デンドログラムを利用し, 楽曲中の繰り返される区間をさまざまな分類クラス数にクラス分けしたときのラベル系列上で検出することにより, 階層的な楽曲構造を捉えられることを確認した. しかし, これらが楽曲の構造を適切に捉えているかどうかの判定規準は難しいため, 性能評価にまで至らず, 確認実験に留まってしまった.

また, 数十曲での確認実験の結果, 多くの課題が判明した. 繰り返し区間の検出方法や, グループへの併合及び削減方法など, 多くの改良可能な点がある. これらを改良することにより, 楽曲の部分曲へのよりよい分割やサビ区間の特定も可能となる事と考えている.

参 考 文 献

1) Bartsch, M.A. and Wakefield, G.H. : To Catch A Chorus : Using Chroma-based Representations for Audio Thumb-nailing, in *Proceeding of WASPAA'01*, pp.15-18 (2001).

2) 後藤真孝 :リアルタイム音楽情景記述システム: サビ区間検出手法, 情処研報 音楽情報科学 2002-MUS-47-6, Vol.2002, No.100, pp.27-34, (2002).

3) 墳崎 英明, 大西 健輔, 星 守, 大森 匡 :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, 情報処理学会 第 59 回全国大会, pp.53-54(3), (1999).

4) Kensuke Onishi, Michihiro Kobayakawa, Mamoru Hoshi, and Tadashi Ohmori : A Feature Independent of Bit Rate For TwinVQ Audio Retrieval, in *Proceedings of IEEE International Conference On Multimedia and Expo (ICME2001)*, pp. 409–416, (2001).

5) 奥鳴 隆, 大西 健輔, 小早川 倫広, 星 守, 大森 匡 : 自己相関特徴量を用いた圧縮データからの構造抽出, 音楽情報科学 2001-MUS-043, Vol.2001, No.125, pp43-1, (2001).

6) Michihiro Kobayakawa, Takashi Okunaru, Kensuke Onishi, and Mamoru Hoshi : A New Method for Extracting a Period of Beat of Music in Compressed domain of TwinVQ Audio Compression, in *Proceedings of the Fourth IEEE Pacific-Rim comference on Multimedia(PCM2003)*, 3A1.2, (2003).

7) Chih-Chin Liu and Po-Hun Tsai : A Singer Identification Technique For Content-Based Classification of MP3 Music Objects, in *Proceedings of the tenth International Conference on Information and Knowledge Management (CIKM2001)*, pp.438–445, (2001).

8) Chih-Chin Liu and Po-Hun Tsai : Content-Based Retrieval of MP3 Music Objects, in *Proceedings of the eleventh International Conference on Information and Knowledge Management (CIKM2002)*, pp.506–511, (2002).

9) Ye Wang and Miika Vilermo : A Compressed Domain Beat Detector using MP3 Audio Bitstreams, in *Proceedings of the ninth ACM International Conference on Multimedia (MM2001)*, pp.194–202 (2001).

10) David Pye : Content-based Methods for Management of Digital Music, in *Proceedings of IEEE International Conference Acoustic, Speech, and Signal Processing (ICASSP2000)*, pp.2437–2440, (2000).

11) ISO/IEC JTC 1/SC 29/WG11 N2203 : Working Draft of ISO/IEC CD 144963, May (1998).

12) N Iwakami, T Moriya, and S Miki : High-quality audio-coding at less than 64/kbit/s using transform domain weighted interleave vector quantization, in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP1995)*, pp. 3095–3098, (1995).

13) T Moriya, N Iwakami, K Ikeda, and S Miki : Extension and Complexity of TwinVQ Audio Coder, in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP1996)*, pp. 1029–1032, (1996).

付　　録

表 2　5 節の確認実験に使用した楽曲.

musicID	楽曲名	作者
03-09	日なたの窓に憧れて	スピッツ
93-20	さよならベイビー	サザンオールスターズ
94-16	BLUE HEAVEN	サザンオールスターズ
96-02	LOVE LOVE SHOW	The Yellow Monkey
97-09	Chang for good	MISIA