画像フィルタリングを利用した楽曲構造解析

三浦　亮† 寺島　裕貴† 喜田　拓也†

† 北海道大学大学院 情報科学研究科
〒 060-0814 北海道札幌市北区北 14 条西 9 丁目
E-mail: †{ryo_miura,tera,kida}@ist.hokudai.ac.jp

あらまし　近年，膨大な数の楽曲を個人がインターネットを介して利用できるようになった．特に，一般の利用者自身が制作した楽曲の爆発的な増加が顕著である．利用者が，そのような大量の楽曲の中から好みの楽曲を見つけるためには，各楽曲を短時間で効率よく試聴できることが求められる．そのため，自動的に楽曲の構造を解析し，楽曲中で最も盛り上がる部分を提示するシステムが提案されている．楽曲構造上意味のある区間を割り出す方法として，楽曲からその自己相似性行列 (Seif-Similarity Matrix) を計算し，2 次元構造上の類似ブロックを見つけるという手法が用いられる．本稿では，自己相似性行列を画像とみなし，既存の画像フィルタリングを利用して区間を割り出す方法について論じる．

キーワード　音楽情報処理，楽曲構造解析，自己相似性行列，画像フィルタリング，サビ抽出，楽曲理解，ハフ変換

Music Structure Analysis Using Image Filtering

Ryo MIURA†, Hiroki TERASHIMA†, and Takuya KIDA†

† Graduate School of Information Science and Technology, Hokkaido University
Kita-14-jo Nishi-9-chome, Kita-ku, Sapporo, Hokkaido, 060-0814 Japan
E-mail: †{ryo_miura,tera,kida}@ist.hokudai.ac.jp

1. はじめに

近年，iTunes Store [2] などのマルチメディア媒体購入サイトや，YouTube [3] やニコニコ動画 [4] といった動画サイトの普及に伴い，インターネットを通じたマルチメディア情報に接する機会が増加している．その中でも音楽データの配信は顕著であり，音楽 CD の販売実績が下降線を取っている現在，音楽をインターネット経由で入手する人が増加傾向にある．

iTunes Store では，販売されている楽曲について，それぞれ試聴が可能である．この試聴を行うことで，自分の好みにあった楽曲を選んで購入することができる．アルバムの形式になっているものも，欲しいと思った楽曲だけを自分で選んで購入することも可能である．この試聴の際，それぞれの楽曲について象徴的な部分から再生されることが多い．ポピュラーソングというジャンルにおいては**サビ**と呼ばれるものである．サビとされる区間は，その楽曲において雰囲気が最も盛り上がる部分であり，また楽曲中でよく繰り返し流れる部分でもある．『COUNTDOWN TV』 [22] や，『COUNTDOWN jp』 [23] のようなテレビ，ラジオで放送されている音楽情報番組の中でも，人気の楽曲をランキング形式で楽曲を紹介しているタイプの番組では，放送時間の都合でほとんどの楽曲は 1 コーラス分 (しばしば「1 番」と呼ばれる部分) 以上放送されることがなく，ランキングの下位の楽曲や、ランキング上位の楽曲の一部は、フラッシュバック形式で盛り上がり部分，すなわちサビと呼ばれる部分のみが放送されることが多い．楽曲の入手をする際に，サビ部分を聴くことで自分の好みに沿った楽曲か否かを判断する人は多い．サビ以外にも楽曲にはさまざまな楽曲的に意味を持った区間が存在し，その区間の頭出しをして再生する方が効率よく楽曲を聴くことができる．大量に存在する楽曲から好みの楽曲をできるだけ多く探すには，この判断を的確にかつ迅速に行うことが不可欠である．

また，CGM (Consumer Generated Media，消費者生成メディア) の成長も著しい．今までは配信されている楽曲を視聴し，購入するのみであったインターネットユーザが，楽曲を自分で制作しそれをオンラインなどで頒布する，いわゆる UGC (User Generated Contents，ユーザ生成コンテンツ) としての楽曲データもインターネット上では増えてきている．

UGC としての楽曲は，CGM サイトにアップロードされているものや，個人所有のアップローダによりアップロードされているものなどがあり，頒布の形式は多岐である．またこれら

の楽曲データは，iTunes Store などとは違い，楽曲のデータに対してジャンル表記やアーティスト表記，あるいは曲の代表部分などの付加データが付いていないことがある．iTunes Store などで販売されている楽曲データと異なり，サビから聴くといった楽曲の一部分の頭出しをしての試聴も容易ではない．楽曲はひとつひとつ違った構造を持つため，サビを選んで試聴しようとしても，結局はサビを探すために楽曲を最初から聴きサビの始まりの部分を探す必要が生じる．したがって，これらの UGC 楽曲について，自分好みの楽曲をできるだけ多く探すということは，iTunes Store などの楽曲配信サイトにおいて楽曲を探すことよりも煩わしい作業となる．

本稿では，UGC の楽曲データのようなタグ付けなどが行われていない楽曲に対しても容易にかつ高速に楽曲構造解析を行うことで，楽曲試聴を気軽に行えるようにするという動機で解析をする．楽曲構造解析では，楽曲データを読み込み，読み込んだデータをクロマベクトルに変換して，自己相似性行列 [1] を計算する．

自己相似性行列とは，データ系列における類似の配列を視覚的に表現するものである．自己相似性行列において類似性は，空間的な距離や，相関性などの比較によって説明することができる．自己相似性行列はその特性から，遺伝子配列のマッチングのようにデータ系列によって指定されたパターンを検索するとき [5] や，人間の動作の類似性を測定するとき [6]，またオーディオ信号中で似た音を探す際 [7] に，それらの類似構造を発見するために用いられる．類似構造を自己相似性行列から探索するときは，自己相似性行列の 2 次元構造上存在する類似したブロック構造を見つけるという手法が用いられる．

本稿では，自己相似性行列をひとつの画像とみなし，既存の画像フィルタリングを利用して区間を割り出す方法について論じる．

2. 楽曲構造の解析

2.1 楽曲の区間

本稿は音楽情報処理の中でも，楽曲構造解析に重点を置くものである．楽曲は，それぞれで全く異なるものであり，それらすべてに固有の構造を持つといえる．

楽曲において，ある程度の時間的長さを有するまとまりのことを**区間**(Segment) という．いくつかの音が集まってできたある程度のまとまりを区間とし，さらにその区間が集まってできたものを楽曲と見ることができる．

楽曲における区切りを表すものには小節というものがあるが，これは楽譜を読みやすくするように適当な長さに区切った区分を指すものである．小節と区間は異なるものである．小節がいくつか集まったものが区間となることが多い．

歌唱のある楽曲で言えば，歌い出しや盛り上がり部分など，節回しとしてまとまっていること，または音楽的な意味を有することが音楽的な「区間」となりえる主な条件となる．また，歌唱の無いインストゥルメンタル曲でも，始まりの部分，中間に盛り上がる部分など，それぞれの楽曲によって盛り上がりまでの展開の仕方や長さは変わったとしても区間分けは可能である．

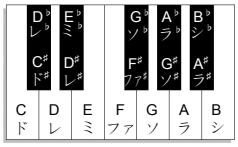

図 1　音名の表記方法．

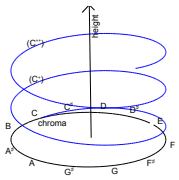

図 2　音楽的音高知覚の模式図．

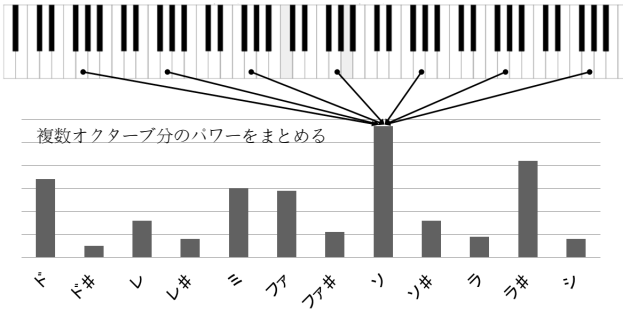

図 3　クロマベクトルへの変換．

日本のポピュラーソングにおいて，区間は A メロ，B メロ，サビなどと名前付けされることが多い．通常これらのサビなどの区間が楽曲中で数回繰り返し演奏される．繰り返しの構造を見つけることが，楽曲構造の解析となる，

2.2 クロマベクトル

クロマベクトル(chroma vector) は音響信号特徴量のひとつであり，極めて短い断片 (フレームという) において表現される特徴量である．クロマベクトルは**クロマ**(音名，chroma) を周波数軸としてパワー分布を表現する．

音名に関して，元になる音 (「ド」,「レ」……) から半音上下しているものに関して，たとえば「ド♯」と「レ♭」,「ファ♯」と「ソ♭」などは同一の音を表すものである (図 1)．本稿での実験においては，半音の異なりはすべて「♯ (シャープ)」を用いて表す．

音楽的音高の知覚は上昇する螺旋のような形状を持っているとされ，図 2 のようにこの螺旋構造を真上から見た円周上に存在するクロマと，横から見たときの縦方向におけるハイト (オクターブ位置，height) の 2 つの次元で表現することができる [10]．

クロマベクトルの表現では，周波数解析によって得られたパワースペクトルを螺旋の高さ，つまりオクターブの違いを考慮しないように，12 音階についてこれら各段階ごとのパワーを加算して構築する．図 3 は，クロマベクトル化を模式化したものである．時刻 t の入力音響信号に対する短時間フーリエ変換

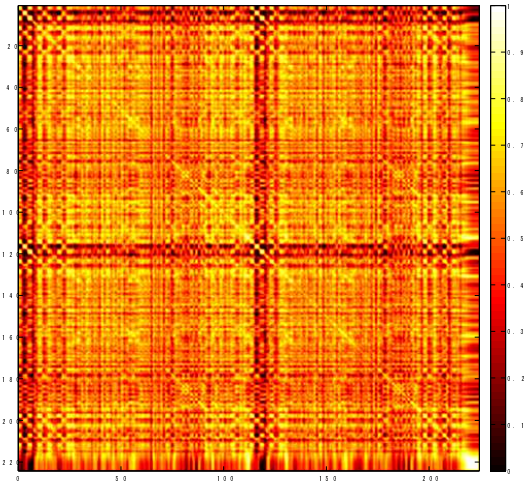

図 4　自己相似性行列 (Self-Similarity Matrix) の例 : 入力データには RWC 研究用音楽データベースのクラシック音楽データベース RWC-MDB-C-2001-C04，楽曲番号 03 :「シューベルト - ピアノ五重奏曲イ長調 op.32 D.667『ます』第 4 楽章」を用いた．

を計算した後，周波数軸を対数スケール周波数 f に変換し，パワースペクトルを求めるといった方法を取る [8], [12].

出力として 12 次元のベクトル $\boldsymbol{x} = (x_1, x_2, \ldots x_{12})^T$ を得る．

周波数解析によるパワースペクトルの扱い方に関してはクロマベクトルのほかにも手法は存在するが，音楽情報解析に対しては図 1 のようにいわゆる「ド」,「レ」,「ミ」，あるいは「ド♯」,「シ♭」などと言った音の高さの構成に一致した特徴量となっているため，コード進行・和音・和声にしたがって楽曲内で類似する区間が得られることが期待される構造となっている．

ある節回しが楽曲中で繰り返すされるたびにメロディーラインや伴奏が少なからず変化をしながら繰り返されることがある．しかし，全体の響き (同時になっているすべての音の構成) が類似しているならば，繰り返しの区間として検出することが可能になるということである．たとえば，サビ部分の歌詞が 1 番と 2 番で異なっていてもメロディーが同様ならば同じものとして扱うようなものに近い．そのような構造を高い精度で検出することが期待できる構造である．

2.3　自己相似性行列 (SSM)

あるひとつの楽曲において開始点と終止点を持つ一定の長さを有する対象となる部分が，楽曲内における他のどの部分と類似していて，かつその部分が当該の楽曲においてどれくらいの領域 (長さ・演奏時間) を占有しているかを判定する指標を**区間類似度**(Segment Similarity Measure) と呼ぶ [12]．さらに，あるひとつの楽曲における区間類似度の高低を表現した 2 次元の行列として表現したものを**自己相似性行列**(Self-Similarlity Matrix) と呼ぶ [9], [12].

自己相似性行列は，データ系列において似たような構造を持つような構造を視覚的に表すことを目的とした表現手法である．類似性は，空間的な距離や，相関性などの比較によって説明することができる．自己相似性行列を用いる例としては，遺伝子配列のマッチングのようにデータ系列によって指定されたパターンの検索 [5] や，人間の動作の類似性の計測 [6]，またオーディオ信号中における似た音の探索 [7] などがあり, [6] においては，対象となるデータに在る類似構造を発見するために用いられる．モーションキャプチャーを装着した被験者に複数の動作をしてもらい，キャプションデータを基にそれぞれの動作に対して自己相似性行列を計算し，現れてきた自己相似性行列から動作の類似性を求めるという手法が提案されている．

表 1　クロマベクトル化における音名の設定．

和名	西洋名	音名番号
ド	C	1
ド♯	C ♯	2
レ	D	3
レ♯	D ♯	4
ミ	E	5
ファ	F	6
ファ♯	F ♯	7
ソ	G	8
ソ♯	G ♯	9
ラ	A	10
ラ♯	A ♯	11
シ	B	12

本稿における自己相似性行列の計算ならびに区間類似度の計算を説明する．全 m フレームを持つ楽曲の i フレーム目と j フレーム目(注1) における区間類似度を $s_{x,y}$ とし，自己相似性行列を $\mathcal{M}$ とする．このとき自己相似性行列 $\mathcal{M}$ のサイズは m 次正方行列となる．

入力として，エネルギー正規化などを処理を行った $12 \times m$ 次元クロマベクトル $\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m)$ を与える．(注2) (注3)．

対応に関しては表 1 に掲載しているものとする．

i 番目のフレームのクロマベクトル $\mathcal{X}_i$ と j 番目のフレームのクロマベクトル $\mathcal{X}_j$ との間の区間類似度 $s_{i,j}$ を求めるには，

$$s_{i,j} = \frac{\langle \mathcal{X}_i, \mathcal{X}_j \rangle}{\| \mathcal{X}_i \| \cdot \| \mathcal{X}_j \|} \qquad (1)$$

を計算する．使用する演算はベクトルのコサイン類似度である．楽曲データはクロマベクトル化を行っているため，それぞれのフレームを表すベクトルに関してコサイン類似度を計算することで区間同士の類似度を計算できる [11], [14].

あるひとつの楽曲に関する区間類似度を求める場合は，楽曲の最初のフレームから最後のフレームまで，すべてのフレームの間について式 (1) を用いて計算する．

区間類似度を行列状に配置したものが自己相似性行列となる．縦方向，横方向，ともに時間 [秒] を単位とする．自己相似性行列の i 行 j 列目の要素は i フレーム目と j フレーム目の類似度が入るため，$\mathcal{M}_{i,j} = s_{i,j}$ となる．区間類似度は 0 以上 1 以下の実数値を取る．

楽曲の中でのそれぞれの区間の間の類似度は図 4 のように視覚的に表現することができる．数値の高低を画像の明暗で塗り

(注1) : フレームの大きさは，実験時に設定したサンプリング周波数に依存する．
(注2) : サンプリング間隔を 0.5 秒とすれば，i フレーム目は楽曲開始から $2i$ 秒目というようになる．
(注3) : $\boldsymbol{x}_i$ は各フレームでそれぞれ 12 次元を有する．

分けることにより，その楽曲中で類似した部分が行列構造に帯状となって現わる [9], [13]．したがって，楽曲の中で繰り返して使われているフレーズがどのあたりに存在し，そのフレーズがどれくらいの時間演奏されているか，ということが視覚的にも理解できるようになる．

入力として与えるべき特徴量が適度な長さと周期を持った変化を持っていると、可視化した際にブロック構造が目視しやすくなる．本稿で取り扱う対象である楽曲構造は適度な長さと周期の変化を持っているため，区間の間の類似性を調べるために自己相似性行列を用いる理由となっている．またこのような構造が作られるため，自己相似性行列は楽曲における類似した構造の存在しているかという点と，その構造をとっている場所の判定を行えるという点に対して便利であるといえる．

3. 境界抽出のためのデジタル画像

3.1 Scharr フィルタ

デジタル画像処理におけるフィルタリング技術のひとつに，領域境界抽出と呼ばれるがある．これは対象となる画像について，画素の明度が急激に変化する点を検出するものであり，すなわち画像における輪郭を検出するものである．画素の明度変化は微分演算（1 次微分，もしくは 2 次微分）を使うが，デジタル画像処理においては微分演算の代わりに差分演算を用いる．

入力した画像のそれぞれの画素位置に対してこれらの近傍領域を用いて出力を行う．このときに使用される近傍領域はフィルタリングに用いる行列の大きさに依存する．たとえば，フィルタリングに 3×3 の行列を用いた場合，計算に用いられる近傍領域はある注目画素を中心として上下左右と斜めの位置にある合計 9 つの画素を用いる．この 9 要素に対して以下のような係数をそれぞれ乗算し，その結果を合成することで計算する．

輪郭抽出を行うフィルタのひとつに **Scharr フィルタ** [15] がある．Scharr フィルタは

$$\begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \tag{2}$$

という行列を用いて処理を行う．したがって，フィルタリング行列を式 (2) の行列として，さらに入力として与えるデジタル画像を I，出力されるデジタル画像を R, 注目画素を (x, y) とすると，

$$\begin{aligned} R(x,y) = &-3(I(x-1,y-1) + I(x-1,y+1)) \\ &- 10(I(x-1,y)) + 10(I(x+1,y)) \\ &+ 3(I(x+1,y-1) + I(x+1,y+1)) \end{aligned}$$

となる．式 (2) のような行列を用いた場合は，画像に対して縦方向に存在する輪郭を強調させる．この行列構造を転置した場合は，横方向に存在する輪郭を強調させることができる．

3.2 Hough 変換

Hough 変換 [16] はデジタル画像処理における特徴抽出の手法である．とくに画像内に潜在する直線構造を検出し抽出するものである．パラメータ空間への投票と多数決に基づく特徴抽出手法といわれる．

ある 1 点を選んだとき，この点を通る直線は無数に存在する．この直線はあらゆる方向から集まりあらゆる方向に発散していくが，この直線の中でももっとも多く特徴点を通過するものを，対象画像の特徴的な直線であるとみなして検出する．画像における特徴点とは，画像に含まれる構造の輪郭を指す．

1 つの直線を 2 つのパラメータで表現する．Hough 変換においては，原点から対象となる直線に引いた法線の長さ r と．その角度 θ を用いて，$r = x\cos\theta + y\sin\theta$ と表す．

画像における特徴点を検出した上で，検出された特徴点それぞれについて，それを通過するすべての直線のパラメータ (r, θ) に投票を行う．さらに検出された特徴点についての投票を行い，多数の投票を得たパラメータを，その画像を特徴付けする直線と見なすものである．

Hough 変換は 1 組のパラメータで表現できる形状はすべて変換方法を応用した上で，それらの図形を抽出することができる．中心と半径を表す，計 3 つのパラメータで表現することにより，円形を抽出する Hough 変換を作ることも可能である．

Hough 変換は検出したい図形に対してノイズが乗っている場合にその図形を抽出する手法ではあるが，あまりにもノイズが乗っている画像に対しては有効性は薄まる．そのためには，前処理などを行い画像のノイズの除去することで，変換の有効度を高める必要がある．

4. 手法と実験

実験を行うにあたって，入力に用いた楽曲データには，RWC 研究用音楽データベースのポピュラー音楽データベース [17] を，楽曲をクロマベクトルに変換をする際に MATLAB®(注4) と，MATLAB で使用できる関数ファイル（M-ファイル）の集合である「Chroma Toolbox」[18] を用いた．自己相似性行列の計算や画像変換処理も同様に MATLAB を使用している．

また，この章における解析結果の図は，とくに注釈が無い限り RWC-MDB-P-2001 No.09 (注5) を用いて行った実験結果として得られた出力データである．

4.1 クロマベクトル化と自己相似性行列の計算

実験の手順は以下のとおりである．

まず解析を行う楽曲データを入力する．Chroma Toolbox を用いるため，入力に用いるデータは RIFF waveform Audio Format（WAV）デジタル音声データとする．

最初に楽曲構造の可視化をするため，楽曲データをクロマベクトルに変換し，さらに自己相似性行列を求める．Chroma Toolbox は入力された WAV 形式の音声データについて，音の高さを検出することを目的としたものである．音声の周波数解析を行い (図 5)，データを 12 次元のクロマベクトルに変換し

(注4)：http://www.mathworks.co.jp/

(注5)：曲名：「慟哭」，演奏時間：4 分 37 秒 (559 フレーム)，テンポ (BPM)：70

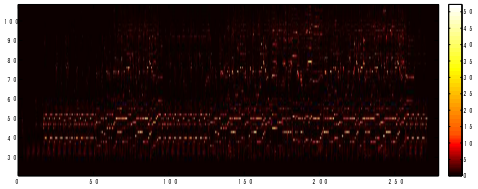

図 5　周波数解析された楽曲データ.

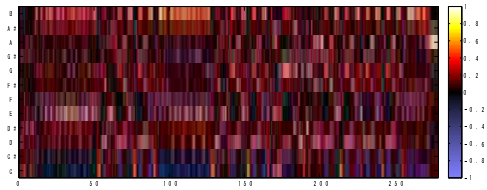

図 6　クロマベクトル化された楽曲データ.

(図 6)，さらに解析に適した形式にするため音の強さのムラをなくすために正規化や対数スケール化を行い，一度離散化させたデータを周波数領域に変換する離散コサイン変換処理を行う．以上の処理は Chroma Toolbox を用いて行う．

自己相似性行列の各要素は，楽曲の各フレーム毎についての類似度である．各フレームは 0.5 秒間隔であり，そのフレーム内でどのような高さの音がどれくらいの強さで鳴っているのかが 12 次元のベクトルで表されている．類似度は，出力された 12 次元ベクトル同士 (片方は転置する) の内積をとることで計算する．各要素は 0 から 1 の実数値をとり，それぞれの値をカラープロットすることで図 4 や図 7 のように可視化できる．図 4 や図 7 について，数値が大きいほど，すなわちフレーム間の類似度が高いほど白く表示される．

4.2　画 像 処 理

楽曲のクロマベクトルから作った自己相似性行列を可視化すると，四角形のブロック状の構造が見えることが多い．このブロック構造で表されている領域が，それぞれの楽曲に固有の「区間」を表現している．左上が楽曲の開始時刻，右下が終了時刻での類似度を指す．自己相似性行列内のブロック構造に焦点を当てると，図 7 において，130 フレーム目から 159 フレーム目 (秒数に換算して 65 秒目から 79.5 秒目) にかけて，細かいチェスボードのような模様の繰り返しが見られる．これと同じ模様を有するブロック構造がその直後の 159 フレーム目から 182 フレーム目 (79.5 秒目から 93 秒目) にも見えるが，これらの区間では同じような高さの音が鳴っている，言い換えれば，同じようなメロディー構造が存在しているといえる．全体を見ると，130 フレーム目から 159 フレーム目と同じような模様の繰り返しが 8 回見られるが，これは 130 フレーム目から 159 フレーム目と同じような楽曲構造がこの曲の中で 8 回繰り返されることを意味する．

今回の実験では楽曲のもつそれぞれの区間について，始点の抽出を行う．つまり，自己相似性行列に存在するブロック構造の端部がある座標を求める．

そのためには，ブロック構造の始点座標を抽出しやすくする

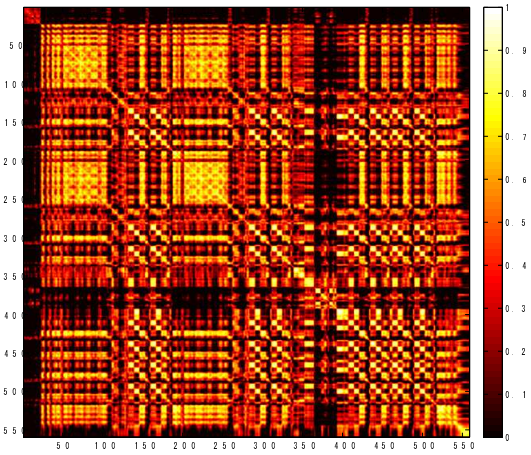

図 7　自己相似性行列に変換された楽曲データ.

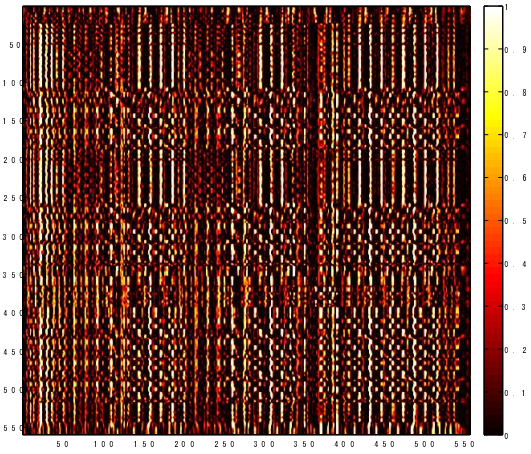

図 8　Scharr フィルタ処理された自己相似性行列.

必要がある．そこで，自己相似性行列を 1 つの画像とみなしてデジタル画像処理を行う．今回は縦方向にのみ模様が現われるような Scharr フィルタと呼ばれるフィルタをかけ，その後に全体に Hough 変換を実行する．この処理において，Scharr フィルタによって自己相似性行列に存在するブロック構造の始点座標のみを抽出し，可視化した際に直線状に明度を際立たせる．また Hough 変換で，Scharr フィルタで目立たせた直線状の構造をその座標とともに抽出する．

まず画像化した自己相似性行列に Scharr フィルタをかけると図 8 のようになる，ここでは Scharr フィルタとして式 (2) を用いて処理を行った．この処理により，ブロック構造の端があった位置とその周辺にのみ明るい部分が現われ，それ以外の部分の明度が下がる．このように直線構造を縦に集約することで，Hough 変換を行ったときの特徴抽出の有効性が上昇する．

Hough 変換を終えると図 9 のようになる．Hough 変換後の画像に付けられた，変換により抽出された特徴的な直線構造の座標を取る．本稿の実験では，Hough 変換については MATLAB の hough 関数，houghpeaks 関数，houghlines 関数など，Hough

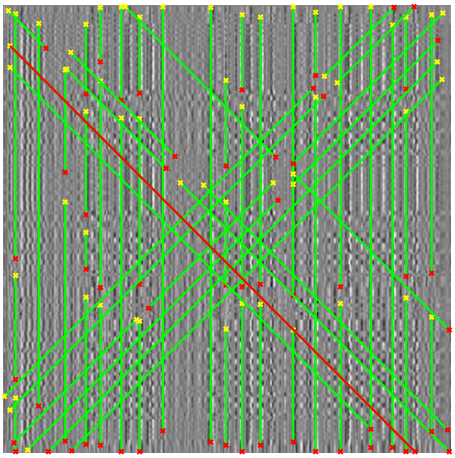

図 9　Hough 変換により直線状構造を抽出された自己相似性行列：図中の緑色線が Hough 変換の結果抽出された特徴構造，赤色線は最も強い特徴とされた構造．

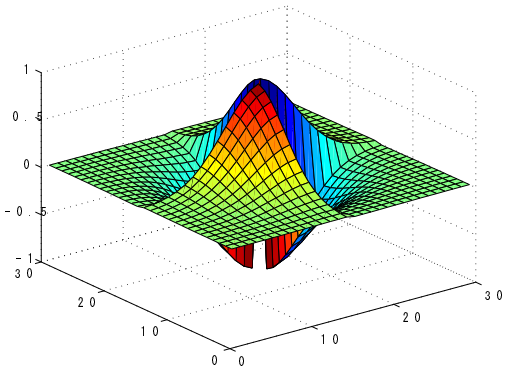

図 10　文献 [20] において用いられるフィルタ．

変換の専用関数を用いた．この関数を実行した後，画像に対して正しく垂直になっている線分のみを選び出し，その座標を記録する．図 9 の解析結果には，画像に対して斜めに走る特徴線があるが，これは区間分けの結果の中には含まないようにする．座標の値は楽曲の再生時間となり，これをまとめた結果がこの解析により楽曲構造の区切れ目と扱われた部分となる．

4.3　解析の評価

入力に使う音楽データには，RWC 研究用音楽データベースのポピュラー音楽データベースから全 100 曲 (RWC-MDB-P-2001 No.01-100) を使用した [17]．また，楽曲構造判定の正解データとして，RWC 音楽データベースのための AIST Annotation データから，ポピュラー音楽データベースの Chorus Section アノテーションデータを利用した [19]．このデータは音楽データベースに収録されている楽曲のコーラス区間，いわゆるサビ区間や A メロの区間が楽曲の何秒目から何秒目までなのかということを記録した楽曲構造解析のための補足資料である．解析の比較とする既存手法には, [20] で楽曲構造の区切れ目を見つける手法の一部を採用した．この手法は，自己相似性行列に対して，2 次元ガウス分布に基づく関数を用いたフィルタ (図 10)

表 2　解析の結果．

手法	本稿の手法	ガウシアンフィルタ分割
調整ランド指数平均	0.56363	0.47811

で画像処理を行い，これにより得られる 1 次元配列について値が極小となる点を楽曲の区切れ目の時刻と見るものである．このフィルタは 2 次元ガウス分布に対し，フィルタの原点を中心として第 2 象限と第 4 象限にあたる領域の正負を反転させたものである．便宜上このフィルタを「ガウシアンフィルタ」と呼び，ガウシアンフィルタを用いた区間分割を「ガウシアンフィルタ分割」と呼ぶことにする．

本稿で論じた手法による解析の結果得られた区切れ目として出てきた秒数のデータと AIST Annotation データに含まれている正解となる区切れ目時刻の秒数のデータとで両者がどれくらい一致しているかを調べる．そのためにまず，区間の区切れ目に対して楽曲全体のクロマベクトル列を前から順番にクラスタに分割する．たとえば，楽曲のクロマベクトルが 8 本あり，これを解析した結果として 3 番目, と 5 番目に区切れ目がある場合を考える．このとき，各クロマベクトルに対して $\begin{pmatrix}1 & 1 & 2 & 2 & 3 & 3 & 3 & 3\end{pmatrix}$ とラベリングして 3 つのクラスタを生成するものとする．解析結果の評価する際の評価指標としては，調整ランド指数 (adjusted rand index) [21] を用いる．調整ランド指数は，クラスタリングの結果を比較するために用いられる指標のことである．指数の値は-1 から 1 の間をとり，クラスタのラベリングが完全に一致していれば 1 となる．

以上の方法で，各楽曲について提案手法とガウシアンフィルタ分割それぞれについて正解データとの調整ランド指数を計算し，100 曲分の平均値を算出した．

結果を表 2 に示す．比較の結果，本稿の手法が調整ランド指数にして 0.085 程度良い結果が得られている．

ガウシアンフィルタ分割は，解析の結果として現れてきた区切れ目時刻に関して，正解データに対して区切れ目として提示されてくるデータ数が多かった．解析結果のクラスタ数が正解データのクラスタ数と大きな差がある場合，調整ランド指数は高い値にならない．本稿の手法では正解データの区切れ目のデータ数とほぼ同じかあるいは少ないことの方が多かった．そのため調整ランド指数が比較的高い値を示したと考えられる．

今回，Hough 変換を行う際に特別にパラメータ調整を行っていない．自己相似性行列を計算する際に，適切なパラメータを同時に計算することができれば各々の楽曲の自己相似性行列に最適なデジタル画像処理が行え，解析能力の向上につながると考えられる．

5.　まとめと今後の課題

本稿では楽曲構造の解析を，音楽情報処理の手法とデジタル画像処理の手法とを組み合わせた手法を考えた．楽曲のデータを，音の高さに関するデータを特徴量として持つクロマベクトルに変換し，得られたクロマベクトルから類似構造を可視化する構造である自己相似性行列を計算する．この自己相似性行列をひとつのデジタル画像と見て，自己相似性行列内のブロック

状の構造の輪郭を Scharr フィルタで抽出し，さらに Hough 変換を用いたデジタル画像特徴抽出を行うことで，楽曲構造の解析を行うという手法である．

解析の結果を調整ランド指数というクラスタリング結果比較を行う指標を用いて比較すると，既存手法より調整ランド指数にして 0.085 程度良い結果が得られた．本稿における実験では，Hough 変換を行う関数などをすべて MATLAB 既存の関数とし，パラメータ設定も既定値のものを使用しているため，サイズの大きなノイズを特徴的な構造と判断しその結果正解データと全く合致しないところに特徴的構造を発見してしまうなど，この実験に最適化された変換機構で無かったといえる．

また，自己相似性行列のブロック構造を直線状構造に変換するフィルタリング行列に関しても，再考の余地がある．現在のフィルタリング行列による処理では，ブロック構造の端の座標だけに直線構造を持たせることが必ずできているわけではない．完全な直線構造を作ることができていないため，Hough 変換をしても正確に直線の座標を解析できない．

今後の課題としては，自己相似性行列の解析に適した画像フィルタリング手法の開発と，Hough 変換機構の最適化が挙げられる．

文　　献

[1] Matthew Cooper and Jonasan Foote : Summarizing Popular Music via Structural Similarity Analysis, 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.

[2] Apple.com,「iTunes Store」, <http://www.apple.com/jp/itunes>, 2012/12/28.

[3] YouTube LLC,「YouTube」, <www.youtube.com/>, 2012/12/28.

[4] niwango,「ニコニコ動画」, <www.niconico.jp/>, 2012/12/28.

[5] Petri Törönen, Mikko Kolehmainen, Garry Wong, and Eero Castrén : Analysis of gene expression data using self-organizing maps, FEBS Letters Volume 451, Issue 2, 21 May 1999, Pages 142-146.

[6] Imran N. Junejo, Emilie Dexter, Ivan Laptev, and Patrick Pérez : Cross-View Action Recognition from Temporal Self-similarities, Computer Vision - ECCV 2008 Lecture Notes in Computer Science Volume 5303, 2008, pp 293-306.

[7] Jonathan Foote : Visualizing Music and Audio using Self-Similarity, MULTIMEDIA '99 Proceedings of the seventh ACM international conference on Multimedia (Part 1).

[8] 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).

[9] Meinard Müller, Peter Grosche, and Nanzhu Jiang : A Segment-Based Fitness Measure for Capturing Repetitive Structures of Music Recordings, 12th International Society for Music Information Retrieval Conference (ISMIR 2011), OS7-1 (2011).

[10] Roger N.' Shepard : Circularity in Judgments of Relative Pitch, The Journal of The Acoustical Society of America Vol.36 No.12 (1964).

[11] Roger B. Dannenberg and Masataka Goto : Music structure analysis from acoustic signals. In David Havelock, Sonoko Kuwano, and Michael Vorländer, editors, Handbook of Signal Processing in Acoustics, Vol 1, pp.305-331. Springer, New York(2008).

[12] Meinard Müller and Frank Kurth : Towards Structural Analysis of Audio Recordings in the Presence of Musical Variations, EURASIP Journal on Advances in Signal Processing Volume 2007, Article ID 89686, 18 pages (2007).

[13] Meinard Müller, Michael Clausen : Transposition-Invariant Self-Similarity Matrices, In Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR), pages 47-50, Vienna, Austria, September 2007.

[14] Mark A. Bartsch and Gregory H. Wakefield : Audio thumbnailing of popular music using chroma-based representations, IEEE Transactions on Multimedia, 7(1):96-104 (2005).

[15] B. Jahne, H. Scharr, and S. Korkel : Principles of filter design, Handbook of Computer Vision and Applications, Academic Press, 1999.

[16] Dana H. Ballard : Generalizin the Hough transform to Detect arbitrary Shapes, Pattern Recognition Volume 13, Issue 2, 1981, Pages 111-122.

[17] 後藤 真孝, 橘口 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", 情報処理学会 音楽情報科学研究会 研究報告 2001-MUS-42-6, Vol.2001, No.103, pp.35-42, October 2001.

[18] Meinard Müller and Sebastian Ewert : Chroma Toolbox : MATLAB Implementations for Extracting Variants of Chroma-Based Audio Features, 12th International Society for Music Information Retrieval Conference (ISMIR 2011).

[19] Masataka Goto: AIST Annotation for the RWC Music Database, Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006), pp.359-360, October 2006.

[20] Jouni Paulus and Anssi Klapuri: Music Structure Analysis Using a Probabilistic Fitness Measure and a Greedy Search Algorithm, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 17, NO. 6, AUGUST 2009.

[21] Nguyen Xuan Vinh, Julien Epps and James Bailey : Information Theoretic Measures for Clustering Comparison: Is a Correction for Chance Necessary?, ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning. ACM. pp. 1073-1080 (2009).

[22] 東京放送，TBS「CDTV」 オフィシャルサイト, <http://www.tbs.co.jp/cdtv/>, 2013/01/07.

[23] エフエム東京,「COUNTDOWN jp -カウントダウン ジェーピー-」オフィシャルサイト，<http://www.tfm.co.jp/cdj/>, 2013/01/07.