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, October 2002.

音 楽 情 報 科 学 47－6
（２００２．１０．２５）

リアルタイム音楽情景記述システム: サビ区間検出手法

後藤 真孝

科学技術振興事業団さきがけ研究21「情報と知」領域 / 産業技術総合研究所
m.goto@aist.go.jp

あらまし　本稿では，ポピュラー音楽の音響信号に対して，サビの区間の一覧を求める手法を提案する．従来，楽曲の音響信号中に何度も出現するサビのどこか一箇所を，指定した長さだけ切り出して提示する研究はあったが，サビ区間の開始点と終了点はわからず，サビの転調も扱えなかった．本手法は，様々な繰り返し区間の相互関係を調べることで，楽曲中で繰り返されるすべてのサビ区間を網羅的に検出し，それらの開始点と終了点を推定できる．また，転調後でも繰り返しと判断できる類似度を導入することで，転調を伴うサビも検出できる．この検出結果は，リアルタイム音楽情景記述システムにおける大局的な記述に相当する．RWC研究用音楽データベース100曲を用いて本手法を評価したところ，80曲のサビが検出できた．

A Real-time Music Scene Description System: A Chorus-Section Detecting Method

Masataka Goto

"Information and Human Activity," PRESTO, Japan Science and Technology Corporation (JST) /
National Institute of Advanced Industrial Science and Technology (AIST)
Information Technology Research Institute, AIST, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

Abstract　This paper describes a method for obtaining a list of chorus sections in popular-music audio signals. Most previous methods detected a repeated section of a given length as a chorus and had difficulty in identifying both ends of a chorus section and in dealing with modulations (key changes). By analyzing relationships among various repeated sections, our method can detect all the repeated chorus sections in a song and estimate their both ends. It can also detect modulated chorus sections by introducing a similarity measure that enables correct judgement in finding modulated repetition. The detected results correspond to global music descriptions in our real-time music scene description system. Experimental results with the RWC Music Database showed that our method correctly dealt with 80 out of 100 songs.

1　はじめに

本研究では，実世界の音楽音響信号を人間のように理解できる計算機システムの実現を目指して，リアルタイム音楽情景記述システム[1]の構築に取り組んでいる.「音楽情景記述」とは，音楽演奏中の刻一刻と変化する情景を分析・理解した結果を記述する処理過程である．人間はこの記述として楽譜や分離信号を必ずしも得ていない，という立場から,「何ができれば音楽を理解したといえるのか」を問い直し，音楽的に訓練されていない「しろうと」が容易にわかる記述を得るアプローチを提案してきた[1]．そして，メロディーやベース[2]，階層的なビート構造[3]等の音楽的要素の記述を得る手法を既に報告した．これらの音楽的要素は，人間には容易にわかるけれど，従来は複数種類の楽器音や歌声を含む複雑な混合音からは推定困難だと考えられていた．

本研究では，より大局的な音楽的要素の記述として，**サビ (chorus, refrain)** の区間を検出する処理を実現する．サビは，楽曲全体の構造の中で，一番代表的な盛り上がる主題の部分である．楽曲中のどこがサビであるかを検出することは,「しろうと」の音楽理解を実現する上で重要である．通常サビは楽曲中で最も多く繰り返され，印象に残るため，訓練されていない人が音楽を聴いたときでも，どこがサビかを容易に判断できる．さらに，サビ検出の結果は，様々な応用において有用である．例えば，多数の楽曲をブラウジングするときや，楽曲検索システムにおいて検索結果を提示するときに，サビの冒頭を短く再生(プレビュー)できると便利である(画像のサムネールの音楽版とみなせる)．また，歌声等を検索キーとした楽曲検索では，検索対象をサビ区間に限定すると精度と効率が上がるが，サビ検出により，その区間を自動的にインデキシングすることも可能になる．

従来のサビ検出の研究では，上記のような応用の観点から，楽曲の音響信号の代表部分としてサビを一箇所切り出す手法が提案されていた☆1．Logan ら[6]は，短い断片(1秒間)にその特徴量に基づいてラベルを付与し，最頻出のラベルを持つ区間をサビとみなす手法を提案した．ラベルの付与には，特徴量間の類似度に基づくクラスタリングや隠れマルコフモデルを用いていた．Bartsch ら[7]は，ビートトラッキングの結果に基づいて拍ごとの断片に分割し，それらの特徴量間の類似度が，指定した一定の長さの区間に渡って最も高い箇所を，サビとして切り出す手法を提案した．また，Foote[8]は，非常に短い断片(フレーム)ごとの特徴量間の類似度に基づく境界検出の応用例として，サビが切り出せる可能性を指摘していた．しかし，いずれの研究も，音楽理解が目的でなかったために，楽曲中に何度も出現するサビのどこか一箇所だけを検出する問題設定をしていた．また，常に指定した一定の長さを切り出して提示するだけで，サビの区間がどこからどこまでかは推定していなかった．サビが繰り返されるときに転調することがあるが，いずれの研究も転調を考慮していないために断片間の類似度が低くなり，サビとして検出することができなかった．

本稿では，それらの問題点を克服し，楽曲中に出現するすべてのサビの区間を網羅的に検出する手法 RefraiD (Refrain Detecting Method) を提案する．本手法は，音楽 CD (compact disc) 等による実世界の複雑な混合音に対して，各サビの区間の開始点と終了点の一覧を求めることができるだけでなく，転調を伴うサビを検出することも可能である．さらに，楽曲全体の中での様々な断片の繰り返し構造に基づいてサビを検出するため，その中間結果として，繰り返し構造の一覧も同時に得ることができる．これらは，文献1)で提案したリアルタイム音楽情景記述システムの大局的な記述に相当する．以下，まず2 章で，サビ区間検出を実現する上で取り組まなければならない課題を述べ，3 章で，それらを解決する具体的な手法を説明する．そして，4 章では，提案手法に基づいて実装したシステムの有効性を評価実験により示す．最後に，5 章でまとめを述べる．

2　サビ区間検出問題

本研究が取り組むサビ区間検出の問題設定を示し，これを解く際の主要な課題を述べる．

2.1　問題設定

本研究では，楽曲一曲分の音響信号に対し，そこに出現するすべてのサビの区間の開始点と終了点を求める問題を解く．サビは，コーラス (chorus) あるいはリフレイン (refrain) とも呼ばれ，楽曲構造上，主題 (theme) を提示している部分を指す．サビは，ときには伴奏の変化やメロディーの変形を伴いながら，通常楽曲中で最も多く繰り返される．例えば，典型的なポピュラー音楽の楽曲構造は，

{ イントロ, サビ }
((⇒ A メロ [⇒ B メロ]) × n_1 ⇒ サビ) × n_2
[⇒ 間奏] [⇒ A メロ] [⇒ B メロ] ⇒ サビ × n_3
[⇒ 間奏 ⇒ サビ × n_4] [⇒ エンディング]

のようになっており，他の部分よりもサビの繰り返しが多くなっている．ここで，{a, b} は a か b のいずれか一方，[a] は a が省略可能であることを表す記号とし，n_1, n_2, n_3, n_4 は繰り返し回数を表す整数である (多くの場合，$1 \leq n_1 \leq 2, 1 \leq n_2 \leq 4, n_3 \geq 0, n_4 \geq 0$)．イントロ (introduction) は前奏部分，A メロ，B メロ (verse A, verse B) は序奏部分を指す．

2.2　実現上の課題

楽曲中で最も多く繰り返されるサビの区間を検出するには，基本的には，ある区間の繰り返しを見つけ出し，最も出現頻度の大きい区間を出力すればよい．しかし，「繰り返し」とは言っても完全に一致する状態で繰り返されることはまれで，人間にとっては容易に繰り返しとわかる場合でも，計算機にとっては判断が難しい．その際の主要な課題は，以下のようにまとめられる．

課題 1: 特徴量と類似度の検討

完全一致しないために，ある区間とある区間が繰り返されているということを，各区間から求めた特徴量間の類似度に基づいて判断しなければならない．その際，繰り返す度に細部が多少異なっても (メロディーが変形したり，伴奏のベース，ドラム等が演奏されなくなったりしても)，特徴量間の類似度は高い必要がある．パワースペクトルを直接特徴量とすると，こうした要件を満たすのは困難となる．

課題 2: 繰り返しの判断基準

類似度がどれぐらい高ければ繰り返しとみなせるかという基準は，楽曲に依存して変わる．例えば，似た伴奏が多用される楽曲では，全体的に多くの部分の類似度が高くなるため，かなり高い類似度でなければ，サビに関連する繰り返しとは判断しない方がよい．逆に，サビが繰り返されるときに，伴奏が大きく変化するような楽曲では，類似度がやや低くても，繰り返しと判断する方がよい．こうした基準を，ある楽曲に特化して人間が手作業で設定するのは容易だが，幅広い楽曲に適用可能な手法とするためには，その基準を現在処理中の楽曲に基づいて自動的

☆1 標準 MIDI ファイル等の音符相当表現を対象とした研究事例[4),5)]もあるが，音源分離が困難な実世界の混合音には適用できなかった．

に変える必要がある．また，このことは，サビ区間検出手法の性能を評価するのに，数曲のサンプル曲が扱えたからと言って，必ずしもその手法に汎用性があるとは限らないことを意味する．

課題3: 繰り返し区間の端点(開始点と終了点)の推定

サビの区間の長さ(**区間長**)は楽曲ごとに異なるため，各区間長と共に，どこからどこまでがサビであるかを推定しなければならない．その際，前後の区間も一緒に繰り返すことがあるため，端点の推定は，楽曲中の様々な箇所の情報を統合しておこなう必要がある．例えば，(A B C B C C) のような構造の楽曲のときには(A, B, C は，それぞれ A メロ，B メロ，サビの区間とする)，単純に長い繰り返しを見つけると，(B C) が一つのまとまった区間として見つかる．この場合，最後の C の繰り返し情報に基づいて，(B C) の C の区間の端点を推定する，といった処理が求められる．

課題4: 転調を伴う繰り返しの検出

転調後の区間は，一般に特徴量が大きく変わるために転調前の区間との類似度が低くなり，繰り返しと判断するのが困難となる．特に，転調は曲の後半のサビの繰り返しで起きることが多く，そうした繰り返しを的確に判断することは，サビの検出において重要な課題である．

3 サビ区間検出手法 RefraiD

本手法では，以上の課題を解決しつつ，基本的に楽曲中で多く繰り返される区間をサビとして検出する．入力としては，ポピュラー音楽のモノラルの音響信号を対象とし，混合音中の楽器の数や種類には特に制限を設けない．ステレオ信号は，左右を混合してモノラル信号に変換するものとする．また，以下を仮定する．

仮定1: 演奏のテンポは一定でなく変化してもよいが，サビの区間は，毎回ほぼ類似したテンポで，一定の長さの区間として繰り返し演奏される．その区間は長い方が望ましいが，区間長には，許容される適切な範囲(現在の実装では，7.7〜40 sec)がある．

仮定2: 2.1 節で述べた楽曲構造の例の

$$(\ (\ \Rightarrow \text{A メロ}\ [\ \Rightarrow \text{B メロ}]\ )\ \times n_1 \Rightarrow \text{サビ}\ )\ \times n_2$$

に相当するような，長い区間の繰り返しがある場合，その末尾の部分がサビである可能性が高い．

仮定3: サビ区間内では，その区間の半分程度の短い区間が繰り返されることが多いため，ある繰り返し区間内にさらにそうした繰り返しがある場合には，それがサビである可能性が高い．

以上は，多くのポピュラー音楽に当てはまる妥当な仮定である．

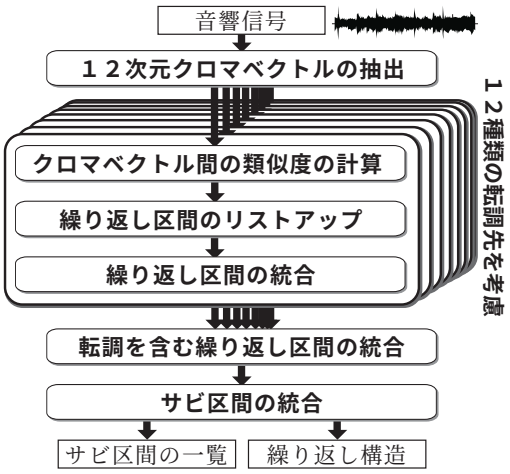

図 1: サビ区間検出手法 RefraiD の処理の流れ

サビ区間検出手法 RefraiD の処理の流れを図 1に示す．まず，入力音響信号の各フレームから，細部の変形の影響を受けにくい12次元の特徴量(12音名の各音名の周波数のパワーを複数のオクターブに渡って加算した12次元クロマベクトル)を抽出し，それと過去の全フレームの特徴量との間の類似度を計算する(課題1に対応)．次に，判別基準に基づく自動閾値選定法[9]によって，繰り返しの判断基準を楽曲ごとに自動的に変えながら，繰り返し区間のペアをリストアップする(課題2に対応)．そして，それらのペアを楽曲全体に渡って統合することで，繰り返し区間のグループを作り，それぞれの端点も適切に求める(課題3に対応)．ここで転調を考えると，クロマベクトルの各次元は音名に対応しているため，その転調幅に応じて次元間で値をシフトさせた転調後のクロマベクトルと，転調前のクロマベクトルとは値が近くなる．そこで，そのように12種類の転調先を考慮して，転調前後のクロマベクトルの類似度を計算する．それを出発点として，上記の繰り返し区間の検出処理も12種類分おこない，それらすべての繰り返し区間を統合する．(課題4に対応)．最終的に，得られた各区間のサビらしさを上記の仮定に基づいて評価し，最もサビらしい区間の一覧を出力する．同時に，中間結果として得られた繰り返し構造も出力する．

3.1 特徴量の抽出

非常に短い断片(フレーム)における音響信号の特徴量として，**クロマベクトル** (chroma vector) を求める．クロマベクトルは，文献10)のクロマを周波数軸としてパワーの分布を表現した特徴量である☆2．文献10)によれば，音楽的な音高の知覚は上に昇る螺旋状の構造を持ち，螺旋を真上から見た円周上のクロマ(音名, chroma)と，横から見たときの縦方向のハイト (オクターブ位置,

☆2 クロマベクトルは，文献11)の chroma spectrum のクロマの軸を12個の音名に離散化したものに近い．

height) の二つの次元で表現することができる☆3．クロマベクトルでは，パワースペクトルの周波数軸がこの螺旋状の構造に沿っているとみなし，ハイト軸方向をつぶして円にすることで，周波数スペクトルを円周上 (1周1オクターブ) のクロマの軸だけで表現する．つまり，異なるオクターブの同じ音名の位置のパワーを加算して，クロマ軸上のその音名の位置のパワーとする．

本研究では，このクロマベクトルを12次元で表し，ベクトルの各次元の値が平均律の異なる音名のパワーを表すものとする．時刻tの入力音響信号に対する短時間フーリエ変換（STFT）を計算した後に，周波数軸を対数スケールの周波数 f に変換して，パワースペクトル $\Psi_p(f,t)$ を求める．対数スケールの周波数はcentの単位で表し，Hzで表された周波数 $f_{\rm Hz}$ を，次のようにcentで表された周波数 $f_{\rm cent}$ に変換する．

$$f_{\rm cent} = 1200 \log_2 \frac{f_{\rm Hz}}{440 \times 2^{\frac{3}{12}-5}} \tag{1}$$

平均律の半音は100 centに，1オクターブは1200 centに相当するため，音名 c (c は $1 \leq c \leq 12$ の整数でクロマに対応)，オクターブ位置 h (ハイトに対応) の周波数 $F_{c,h}$ centは

$$F_{c,h} = 1200h + 100(c-1) \tag{2}$$

と表せる．この対数スケール軸のパワースペクトル $\Psi_p(f,t)$ から，音名 c の位置のパワーを $\mathrm{Oct_L}$ から $\mathrm{Oct_H}$ (現在の実装では，3から8) のオクターブ範囲で加算して，12次元クロマベクトル $\vec{v}(t)$ の各次元 $v_c(t)$

$$v_c(t) = \sum_{h=\mathrm{Oct_L}}^{\mathrm{Oct_H}} \int_{-\infty}^{\infty} BPF_{c,h}(f)\, \Psi_p(f,t)\, df \tag{3}$$

を求める．ここで，$BPF_{c,h}(f)$ は，音名 c，オクターブ位置 hの位置のパワーを通過させるバンドパスフィルタで，

$$BPF_{c,h}(f) = \frac{1}{2}\left(1 - \cos\frac{2\pi(f-(F_{c,h}-100))}{200}\right) \tag{4}$$

のようにハニング窓の形状で定義する．

こうして得られたクロマベクトルを特徴量とすることで，繰り返す度にメロディーや伴奏が多少変わっても，全体の響き (同時に鳴っている音名の構成) が類似していれば，繰り返し区間として検出できる．さらに，3.5 節で後述するように，類似度の工夫によって転調された繰り返しの検出も可能となる．なお，文献13) の手法により，クロマベクトルに基づいてコード名を同定できることも確かめられている．クロマベクトルを特徴量とすることで，コード進行の類似性を反映した繰り返し検出ができることが期待される．

現在の実装では，音響信号を標本化周波数 16 kHz，量子化ビット数 16 bitで A/D 変換し，窓関数 $h(t)$ として窓幅4096点のハニング窓を用いたSTFTを，高速フーリエ変換 (FFT) によって計算する．FFTのフレームは1280点ずつシフトし，すべての処理の時間単位（1フレームシフト）を 80 ms とする．

☆3 文献12) によれば，クロマの同定には聴神経の発火の周期性が，ハイトの同定には基底膜振動のピーク位置が主要な役割を果たしていることが示唆されている．

3.2 類似度の計算

時刻tのクロマベクトル $\vec{v}(t)$ とそれよりラグ (lag) l ($0 \leq l < t$) だけ過去の $\vec{v}(t-l)$ との類似度 $r(t,l)$ を，

$$r(t,l) = 1 - \frac{\left|\frac{\vec{v}(t)}{\max_c v_c(t)} - \frac{\vec{v}(t-l)}{\max_c v_c(t-l)}\right|}{\sqrt{12}} \tag{5}$$

と定義する．分母の $\sqrt{12}$ は，1辺の長さが1の12次元超立方体の対角線の長さであり，$\frac{\vec{v}(t)}{\max_c v_c(t)}$ は，常にその超立方体の原点を含まない面上に位置するため，$0 \leq r(t,l) \leq 1$ となる．

3.3 繰り返し区間のリストアップ

類似度 $r(t,l)$ に基づいて，どの区間が繰り返されているかを調べる．類似度 $r(t,l)$ を，横軸が時間軸 t，縦軸がラグ軸 l の $t-l$ 平面に描画すると，繰り返されている区間に対応して，時間軸に平行な線分 (類似度が連続して高い領域) が現れる (図 2)．そこで，時刻 $T1$ から $T2$ の区間 (以下，$[T1,T2]$ と表記する) に渡ってラグ軸 $L1$ の位置に高い類似度を持つ線分を**類似線分**と呼び，$[t=[T1,T2], l=L1]$ で表す．これは，$[T1,T2]$ と $[T1-L1, T2-L1]$ が繰り返し区間であることを意味する．よって，$r(t,l)$ 中の類似線分をすべて検出すれば，繰り返し区間の一覧が得られる．

この線分検出には，画像処理においてロバストな直線検出手法として多用されるハフ (Hough) 変換を用いる．ハフ変換では，$t-l$ 平面における求めたい直線をパラメータ a,b を用いて $l=at+b$ で表すとき，画素 (T,L) ごとにパラメータ空間に $b=L-aT$ の軌跡を描く (画素の輝度を累積する)．そして，多くの軌跡が交わる点 (累積値の大きい点) のパラメータを持つ直線が，画像中に存在するものとみなす．類似線分の検出の場合には，時間軸に平行な線分だけを求めればよいので上記の直線の傾きは常に0となり，パラメータ空間は1次元と単純化される．具体的には，時刻 t におけるパラメータ空間 $R_{all}(t,l)$ は，

$$R_{all}(t,l) = \int_l^t \frac{r(\tau,l)}{t-l}\, d\tau \tag{6}$$

となり，$R_{all}(t,l)$ が大きい値を持つ l の位置に類似線分が存在する可能性が高いと考える (図 2)．

なお，広帯域ノイズ等に起因する各成分がほぼ等しいクロマベクトルからは，他のクロマベクトルへの距離が比較的近くなってしまう傾向があり，$r(t,l)$ 中に類似度の高い直線 (以下，ノイズ直線と呼ぶ) として現れることがある．このノイズ直線は，$t-l$ 平面において，

時間軸に垂直 (上下) 方向，あるいは，斜め右上・左下方向に現れる．そこで前処理として，式 (6) の計算前にノイズ直線の抑制をおこなう．まず，各 $r(t,l)$ において，右，左，上，下，右上，左下の 6 方向の近傍区間の平均値を計算し，その最大と最小を求める．そして，右か左の方向が最大のときは，類似線分の一部とみなして，強調するために $r(t,l)$ から最小を引く．その他の方向が最大のときは，ノイズ直線の一部とみなして，抑制するために $r(t,l)$ から最大を引く．

上記のように $R_{all}(t,l)$ を求めた後の類似線分の検出は，以下の手順でおこなう．

1. 線分候補ピークの検出

   $R_{all}(t,l)$ 中の十分に高いピークを，**線分候補ピーク**として検出する．まず，$R_{all}(t,l)$ 中の lag 軸方向のピークを，2 次多項式適合による平滑化微分を用いたピーク検出[14] により求める．具体的には，$R_{all}(t,l)$ の平滑化微分

   $$\sum_{w=-\mathrm{KSize}}^{\mathrm{KSize}} w\, R_{all}(t,l+w) \tag{7}$$

   が正から負に変わる箇所をピークとする (KSize = 0.32 sec)．ただし，このピーク検出の前に，$R_{all}(t,l)$ の lag 軸方向に，2 階のカーディナル B-スプライン関数を重み関数とする移動平均によってスムージングをかけたものを引いて，$r(t,l)$ のノイズ成分等の蓄積による大局的な変動を取り除いておく ($R_{all}(t,l)$ にハイパスフィルタをかけることに相当する)．

   次に，こうして得られたピークの集合から，ある閾値より大きいピークのみを，線分候補ピークとして選ぶ．2.2 節の課題 2 で述べたように，この閾値は楽曲ごとに適切な値が異なるため，楽曲に基づいて自動的に変える必要がある．そこで，$R_{all}(t,l)$ のピーク値を閾値によって二つのクラスに分けるときに，クラス分離度を最大とする判別基準に基づく自動閾値選定法[9] を用いる．ここでは，クラス分離度としてクラス間分散

   $$\sigma_B^2 = \omega_1\omega_2(\mu_1-\mu_2)^2 \tag{8}$$

   を最大とする閾値を求める．ただし，ω_1，ω_2 は，閾値によって分けられた各クラスの生起確率 (各クラスのピーク個数 / 全体のピーク個数)，μ_1，μ_2 は，各クラスのピーク値の平均である．

2. 類似線分の探索

   各線分候補ピークの lag 軸上の位置 l において，類似度 $r(t,l)$ の時間軸方向を一次元関数とみなして，それが連続して十分高い区間を探索し，類似線分とする．まず，$r(t,l)$ の時間軸方向に，2 階のカーディナル B-スプライン関数を重み関数とする移動平均によってスムージングをかけた $r_{smooth}(t,l)$ を求める．次に，$r_{smooth}(t,l)$ 中で，ある閾値を連続して越えているすべての区間のうち，一定の長さ (6.4 sec) 以上のものを類似線分として求める．この閾値も，上記の判別基準に基づく自動閾値選定法により定める．ただし，今度はピーク値を扱うのではなく，ピーク値が高い上位 5 個の線分候補ピークを選び，それらの l の位置の $r_{smooth}(\tau,l)$ $(l \le \tau \le t)$ がとる値を二つのクラスに分ける．

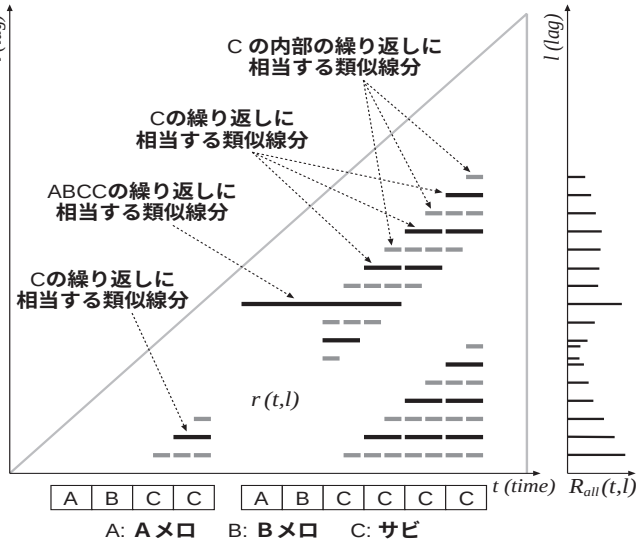

図 2: ある楽曲に対する類似線分，類似度 $r(t,l)$，パラメータ空間 $R_{all}(t,l)$ の概念図: $r(t,l)$ は，右下半分の三角形内で定義される．実際に得られる $r(t,l)$ はノイズを多く含み，サビに関連しない類似線分も存在して曖昧なことが多い．

3.4 繰り返し区間の統合

各類似線分は，ある区間が二回繰り返されていることだけを表すため，例えば A と A' のペア，A' と A" のペアが，それぞれ繰り返し区間として検出されたときには，それらを一つの繰り返し区間のグループとして統合する必要がある．ここで，ある区間が n 回 $(n \ge 3)$ 繰り返されている場合には，もれなく検出されるとすると，$\frac{n(n-1)}{2}$ 本の類似線分が検出される．そこで，同じ区間の繰り返しを表す類似線分をグルーピングし，繰り返し区間を統合する．さらに，もれていた類似線分の検出や，得られた類似線分が適切であるかの検証もおこなう．

この統合処理は，以下の手順で実現する．

1. 類似線分のグルーピング

   ほぼ同じ区間の類似線分を，一つのグループにまとめる．各グループ $\phi_i = [[Ts_i, Te_i], \Upsilon_i]$ は，区間 $[Ts_i, Te_i]$ と，類似線分 (区間が決まれば，線分候補ピークと対応する) の lag 値 υ_{ij} の集合 $\Upsilon_i = \{\upsilon_{ij} \mid j = 1,2,\ldots,M_i\}$ (M_i はピークの個数) で表される．そして，この類似線分のグループ ϕ_i の集合を，$\Phi = \{\phi_i \mid i = 1,2,\ldots,N\}$ (N はグループの個数) とする．

2. 線分候補ピークの再検出

グループ ϕ_i ごとに，区間 $[Ts_i, Te_i]$ 内の類似度 $r(t,l)$ に基づいて，類似線分を改めて求め直す．これにより，もれていた類似線分の検出ができ，例えば，図 2で，ABCCの繰り返しに相当する長い類似線分上で，Cの繰り返しに相当する類似線分2箇所が得られていなくても，この処理で検出されることが期待できる．まず，$[Ts_i, Te_i]$ 内に限定して，ハフ変換のパラメータ空間 $R_{[Ts_i,Te_i]}(l)$ $(0 \leq l < Ts_i)$

$$R_{[Ts_i,Te_i]}(l) = \int_{Ts_i}^{Te_i} \frac{r(\tau,l)}{Te_i - Ts_i}\, d\tau \qquad (9)$$

を作成する．次に，3.3 節の線分候補ピークの検出と同様に，平滑化微分を用いたピーク検出をおこない (KSize = 2.8 sec)，自動閾値選定法で定めた閾値を越えた線分候補ピークの lag 値 v_{ij} の集合を，改めて Υ_i とする．自動閾値選定法では，Φ の全グループの区間における $R_{[Ts_i,Te_i]}(l)$ のピーク値を，二つのクラスに分けるようにする．

3. 類似線分の適切さの検証1

サビと無関係な類似線分から成るグループ ϕ_i，あるいは，Υ_i の中で無関係な線分と考えられるピークを削除する．似た伴奏の繰り返しが多用される楽曲等の場合，サビと関係のない線分候補ピークが，$R_{[Ts_i,Te_i]}(l)$ に等間隔に多く現れる傾向がある．そこで，$R_{[Ts_i,Te_i]}(l)$ に対して平滑化微分を用いたピーク検出をおこない，一定間隔 (間隔は任意) で連続して並ぶ高いピークの個数が 10 個より多いとき，サビと無関係な類似線分から成るグループだと判断し，そのグループを Φ から削除する．また，一定間隔で連続して並ぶ低いピークの個数が 5 個より多いとき，サビと無関係な線分候補ピークだと判断し，その一連のピークを Υ_i から削除する．

4. 類似線分の適切さの検証2

Υ_i の中には，区間 $[Ts_i, Te_i]$ の一部分だけ類似度が高いピークが含まれることがあるため，そうした類似度の変動の大きいピークを削除する．そこで，当該区間の $r_{smooth}(\tau,l)$ の標準偏差を求め，ある閾値より大きいものは Υ_i から削除する．この閾値は，ϕ_i の中で，3.3 節 で求めた類似線分に対応する線分候補ピークは信頼できると考え，それらのピークでの上記標準偏差の最大値を定数倍 (1.4 倍) して定める．

5. 類似線分の間隔の考慮

繰り返し区間が重ならないようにするために，lag 軸上で隣接する類似線分 (線分候補ピーク) の間隔を，線分の長さ $Te_i - Ts_i$ 以上とする必要がある．そこで，線分の長さより狭い間隔を持つ二つのピークのいずれかを，全体として高いピーク集合が残るように削除し，すべての間隔が類似線分の長さ以上になるようにする．

6. 共通区間を持つグループを統合

Υ_i の各ピークについて，その lag 値 v_{ij} だけ過去の区間 $[Ts_i - v_{ij}, Te_i - v_{ij}]$ のグループがあるかを探索し，発見したら統合する．統合処理では，発見したグループのすべてのピークを，対応する lag 値の場所に持つように，Υ_i に線分候補ピークを追加する．発見したグループ自体は削除する．

さらに，区間 $[Ts_i - v_{ij}, Te_i - v_{ij}]$ に一致する線分候補ピークを持つグループ Υ_k (グループの区間自体は異なる) があるかも探索し，発見したら統合するか判断する．この場合，Υ_k の過半数のピークが Υ_i に含まれていれば，上記同様の統合処理をおこなう．含まれていなければ，Υ_i と Υ_k で同じ区間を指しているピークを比較し，低い方を削除する．

上記で実際に統合がなされたら，後処理として，5. の処理を再びおこなう．

3.5 転調を伴う繰り返しの検出

以上述べてきた処理は転調を考慮していなかったが，これは以下のように，転調を扱える処理へと容易に拡張できる．ここで，転調は平均律の半音 tr 個分上の調へ変わることで表すこととし，tr は $0, 1, \ldots, 11$ の 12 種類の値を取るものとする．$tr = 0$ は転調しないことを意味し，$tr = 10$ は半音 10 個分上か，全音分下へ転調することを意味する．

12 次元クロマベクトル $\vec{v}(t)$ は，各次元 $v_c(t)$ の値を次元間で tr 個分だけシフトさせることで，転調を表現できる特長を持つ．具体的には，ある演奏のクロマベクトルを $\vec{v}(t)$ とし，それを tr 個上へ転調した演奏のクロマベクトルを $\vec{v}(t)'$ とすると，

$$\vec{v}(t) \doteqdot S^{tr}\vec{v}(t)' \qquad (10)$$

となる．ただし，Sはシフト行列で，以下のように 12 次正方行列を一つ右にシフトした行列として定義される．

$$S = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix} \qquad (11)$$

転調を伴う繰り返しの検出の処理手順を以下に述べる．まず，クロマベクトルのこの特長を利用し，tr ごとの 12 種類の類似度 $r_{tr}(t,l)$ を

$$r_{tr}(t,l) = 1 - \frac{\left| \frac{S^{tr}\vec{v}(t)}{\max_c v_c(t)} - \frac{\vec{v}(t-l)}{\max_c v_c(t-l)} \right|}{\sqrt{12}} \qquad (12)$$

と定義し直す．次に，それぞれの類似度 $r_{tr}(t,l)$ に対して，3.3 節の繰り返し区間のリストアップをする．ただし，自動閾値選定法は $tr = 0$ のときだけ適用し，他の tr では，$tr = 0$ で定めた閾値を用いる．これによ

り，転調のない曲で，$tr = 0$ 以外のときに類似線分が誤検出されにくくなる．そして，こうして得られた各 tr ごとの類似度と類似線分に対して，3.4 節の統合処理をおこなう．その結果，tr ごとに別々の類似線分のグループ $\phi_{tr,i}$ の集合 Φ_{tr} が得られる．そこで，3.4 節の 6. の処理を，tr 間にまたがっておこなう (異なる tr に対して共通区間を持つグループを探索する) ことで，転調を含む繰り返し区間を一つのグループとして統合する．ただし，後半の処理で「Υ_k の過半数のピークが Υ_i に含まれていれば，上記同様の統合処理をおこなう」とあるが，ここでは常に統合処理をおこなう．

以下，異なる tr から得られたグループも合わせて，$\Phi = \{\phi_i\}$ で表す．転調区間が後からわかるように，どの tr から統合されたかという情報は保存しておく．

3.6 サビ区間の検出

類似線分のグループの集合 Φ の中から，ある一つのグループをサビ区間として選ぶ．そのために，各グループ ϕ_i の**サビらしさ** ν_i を，類似線分の平均類似度や 3 章の冒頭で述べた仮定に基づいて評価し，最も ν_i の高いグループをサビ区間であると判定する．その準備として，グループごとに，類似線分 (線分候補ピーク υ_{ij}) をそれが指す二つの区間へ展開し，すべての繰り返し区間 $[Ps_{ij}, Pe_{ij}]$ とその信頼度 λ_{ij} のペアの集合

$$\Lambda_i = \{[[Ps_{ij}, Pe_{ij}], \lambda_{ij}] \mid j = 1, 2, \ldots, M_i + 1\} \quad (13)$$

を求める．ここで，$[Ps_{ij}, Pe_{ij}] = [Ts_i - \upsilon_{ij}, Te_i - \upsilon_{ij}]$ とし，信頼度 λ_{ij} は，対応する類似線分における類似度 $r_{tr}(t, l)$ の平均とする．ただし，$j = M_i + 1$ のときは，$[Ps_{ij}, Pe_{ij}] = [Ts_i, Te_i]$，$\lambda_{ij} = \max_{k=1}^{M_i} \lambda_{ik}$ とする．

サビらしさ ν_i は，以下の手順で評価する．

1. 仮定 2 を満たす区間の信頼度を増加

   仮定 2 で述べた A メロ〜サビに相当するような十分に長い区間 (50 sec 以上) を持つグループ ϕ_h に関して，その各区間の終了点 Pe_{hk} とほぼ等しい終了点 Pe_{ij} を持つ区間が他のグループにあるか探索する．発見されれば，発見された区間がサビである可能性が高いと考え，その信頼度 λ_{ij} を 2 倍する．

2. 仮定 3 を満たす区間の信頼度を増加

   サビとして適切な区間長の範囲 (仮定 1) の区間 $[Ps_{ij}, Pe_{ij}]$ に関して，その区間の半分程度の短い区間が前半と後半に一つずつ存在するか調べる．存在する場合には，それら二つの区間の信頼度の平均の半分を，元の区間の信頼度 λ_{ij} に加える．

3. サビらしさを算出

   上記で得られた信頼度に基づき，サビらしさを

$$\nu_i = \left(\sum_{j=1}^{M_i+1} \lambda_{ij} \right) \log \frac{Te_i - Ts_i}{\mathrm{D_{len}}} \quad (14)$$

   と定義する．$\sum$ の項は，グループ ϕ_i の区間の数が多いほど，また，それらの信頼度が高いほど，サビらしさが高いことを意味する．log の項は，そのグループの区間が長いほど，サビらしさが高いことを意味する．定数 $\mathrm{D_{len}}$ は，予備実験の結果から 1.4 sec とした．

最終的に，サビとして適切な区間長の範囲 (仮定 1) を持つグループの中で，

$$m = \operatorname*{argmax}_i \nu_i \quad (15)$$

によって決まる集合 Λ_m 中の区間 $[Ps_{mj}, Pe_{mj}]$ を，サビ区間とする．ここで後処理として，隣接する Ps_{mj} の最小間隔を求め，区間長が最小間隔となるように Pe_{mj} を移動して各区間を広げ，隙間を埋める．これは，本来はサビ区間が連続して隙間がないにも関わらず，得られた繰り返し区間では隙間が空いてしまうことがあるからである．ただし，埋める隙間が大きすぎるとき (12 sec 以上で区間長の半分より広いとき) は埋めない．

4 システムの実装と実験結果

音楽音響信号を入力し，検出したサビ区間の一覧をリアルタイムに出力するシステムを，提案手法 RefraiD に基づいて構築した．本システムは，刻一刻と，過去の音響信号中でサビだと考えられる区間の一覧を求め，中間結果として得られた繰り返し構造 (繰り返し区間の一覧 Λ_i) と共に出力し続ける．この出力を視覚化した例を図 3 に示す．本システムは，リアルタイム音楽情景記述システム [1] の他の処理と同様，分散環境で RACP (Remote Audio Control Protocol) を用いて実装され，音響信号の入力，RefraiD の計算，中間結果や出力の視覚化といったシステムを構成する各機能は異なるプロセスとして実行される．

評価実験として，「RWC 研究用音楽データベース: ポピュラー音楽」[15] の 100 曲 (RWC-MDB-P-2001 No. 1 〜 100) を対象に，本システムのサビ検出性能を調べた．一曲すべてを入力し終わった時点で，サビ区間として検出されたものを対象に評価する．この正誤を判定するためには，基準となる正解のサビ区間を人間が手作業で指定する必要がある．そこで，楽曲を分割して各部にサビ，A メロ，B メロ，間奏等をラベリングできる，楽曲構造ラベリング用エディタを開発した．ラベリングでは，相対的な調の移動幅 (曲の先頭の調に対して半音何個分上か) も正解に付与する．

こうして作成した正解に基づき，各曲に対する出力結果の区間と正解のサビ区間がどれぐらい重なっているかを，再現率 (recall rate)，適合率 (precision rate)，および両者を統合した F 値 (F-measure)[16] の観点から評価した．以下に定義を示す．

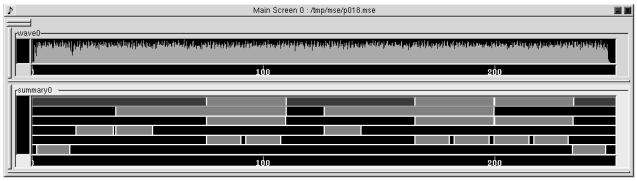

図 3: RWC-MDB-P-2001 No. 18 の楽曲終了時点での正しいサビ検出結果: 横軸は時間軸 (sec) で楽曲全体を表示しており，上半分がパワー変化，下半分の最上段がサビ区間の一覧 (最後のサビは転調を伴う)，下5段が繰り返し構造を表す.

$$\text{再現率}\,(R) = \frac{\text{正しく検出したサビ区間の長さの合計}}{\text{正解のサビ区間の長さの合計}}$$

$$\text{適合率}\,(P) = \frac{\text{正しく検出したサビ区間の長さの合計}}{\text{検出した区間の長さの合計}}$$

$$\text{F 値} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad (\beta = 1 \text{を使用})$$

ただし，転調を伴う場合には，相対的な調の移動幅が正解と一致したときだけ，正しく検出したと判断した．そして，F 値が 0.75 以上のとき，その曲のサビ区間を正しく得られた (正答した) と判定した．

評価結果として，100 曲中の正答曲数を表 1 に示す．通常の RefraiD の性能は一番左の 80 曲 (80 曲の平均 F 値は 0.938) である．誤検出は，サビの繰り返しが他の箇所の繰り返しより多くなかったり，曲中ほとんどが類似伴奏の繰り返しだったりしたのが主な原因だった．100 曲中には，サビに転調のある曲が 10 曲含まれているが，そのうち 9 曲は検出できていた．3.5 節の転調を伴う繰り返しの検出をやめた場合，左から二番目のように性能が落ちた．一方，3.6 節の仮定 2, 3 に基づく信頼度の増加をやめた場合は，右二つのように性能が落ちた．サビの繰り返しで，伴奏やメロディーに大幅な変化を伴う曲は 22 曲あったが，そのうち 21 曲は検出できており，その中で，変化を伴うサビ自体は 16 曲で検出できていた．

表 1: 様々な条件における 100 曲中のサビ正答曲数

	条件 (使用: ○, 未使用: ×)			
転調区間の検出	○	×	○	×
仮定 2,3 の使用	○	○	×	×
正答曲数	80 曲	74 曲	72 曲	68 曲

5 おわりに

本稿では，リアルタイム音楽情景記述システムにおける大局的な記述として，楽曲の音響信号中のサビ区間を検出する手法 RefraiD について述べた．本手法は，基本的に楽曲中で最も多く繰り返される区間をサビとして検出する．その際，様々な区間の繰り返しを楽曲全体の情報を統合しながら調べることで，従来実現されていなかった，すべてのサビ区間の開始点・終了点の一覧を得ることを可能にした．また，転調後でも繰り返しと判断できるような，クロマベクトル間の類似度を導入したことで，サビの転調も検出できるようになった．RWC 研究用音楽データベース (RWC-MDB-P-2001) 100 曲を用いて評価した結果，80 曲で正答でき，実世界の音響信号中のサビ区間が検出できることが確認された．

なお，本研究は音楽要約 [17] とも関連しており，RefraiD を，楽曲の要約結果としてサビ区間を提示する音楽要約手法と捉えることもできる．さらに，サビよりも長い区間の要約が必要なときは，中間結果として得られた繰り返し構造を用いることで，楽曲全体の冗長性を減らした要約の提示も可能となる．例えば，中間結果として (A メロ ⇒ B メロ ⇒ サビ) の繰り返しが捉えられているときは，それを提示できる．

本稿の実験ではポピュラー音楽を用いて評価したが，RefraiD は他の音楽ジャンルにも適用できる可能性を持つ．実際に，数曲のクラシック音楽に適用したところ，その楽曲で最も代表的な主題が提示される部分を求めることができた．そこで今後は，他ジャンルの楽曲に対する評価実験等もおこなっていく予定である．

謝 辞

本研究に対し有益な議論をして頂いた，麻生 英樹 氏 (産業技術総合研究所) に感謝する．

参 考 文 献

[1] 後藤真孝: リアルタイム音楽情景記述システム: 全体構想と音高推定手法の拡張, 情処研報 音楽情報科学 2000-MUS-37-2, 9–16 (2000).

[2] 後藤真孝: 音楽音響信号を対象としたメロディーとベースの音高推定, 信学論 (D-II), **J84-D-II**, 1, 12–22 (2001).

[3] Goto, M.: An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds, *J. of New Music Research*, **30**, 2, 159–171 (2001).

[4] Meek, C. and Birmingham, W. P.: Thematic Extractor, *Proc. of ISMIR 2001*, 119–128 (2001).

[5] 村松純: 歌謡曲における「さび」の楽譜情報に基づく特徴抽出 — 小室哲哉の場合—, 情処研報 音楽情報科学 2000-MUS-35-1, 1–6 (2000).

[6] Logan, B. and Chu, S.: Music Summarization Using Key Phrases, *Proc. of ICASSP 2000*, II–749–752 (2000).

[7] Bartsch, M. A. and Wakefield, G. H.: To Catch A Chorus: Using Chroma-based Representations for Audio Thumbnailing, *Proc. of WASPAA'01*, 15–18 (2001).

[8] Foote, J.: Automatic Audio Segmentation Using A Measure of Audio Novelty, *Proc. of ICME 2000*, I–452–455 (2000).

[9] 大津展之: 判別および最小2乗規準に基づく自動しきい値選定法, 信学論 (D), **J63-D**, 4, 349–356 (1980).

[10] Shepard, R. N.: Circularity in Judgments of Relative Pitch, *J. Acoust. Soc. Am.*, **36**, 12, 2346–2353 (1964).

[11] Wakefield, G. H.: Mathematical Representation of Joint Time-Chroma Distributions, *SPIE'99*, 637–645 (1999).

[12] 藤崎和香, 柏野牧夫: 絶対音感保持者の音高知覚特性, 日本音響学会誌, **57**, 12, 759–767 (2001).

[13] 山田洋子, 後藤真孝, 猿渡洋, 鹿野清宏: 音楽音響信号を対象とした和音名同定手法, 音講論集 秋季 1-1-3, 641–642 (2002).

[14] Savitzky, A. and Golay, M. J.: Smoothing and Differentiation of Data by Simplified Least Squares Procedures, *Analytical Chemistry*, **36**, 8, 1627–1639 (1964).

[15] 後藤真孝, 橋口博樹, 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, 情処研報音楽情報科学 2001-MUS-42-6, 35–42 (2001).

[16] van Rijsbergen, C. J.: *Information Retrieval*, Butterworths, second edition (1979).

[17] 平田圭二, 松田周: パピプーン: GTTM に基づく音楽要約システム, 情処研報音楽情報科学 2002-MUS-46-5, 29–36 (2002).