音楽情報科学 21－8
（1997．7．21）

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 ―

後藤 真孝　　村岡 洋一

早稲田大学 理工学部

{goto, muraoka}@muraoka.info.waseda.ac.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リアルタイムに階層的なビート構造（四分音符～小節レベル）を認識するビートトラッキングシステムについて述べる．従来研究の多くはMIDIが対象であり，音響信号を対象とした我々の従来研究でも，二分音符レベルまでのビート構造しか認識できず，打楽器音の有無によって異なるシステムとなっていた．本研究では，打楽器音の有無に応じた音楽的知識を同一システム上で選択して適用するために，その有無の判定手法を提案する．さらに，トップダウン情報を用いた周波数解析で得られるコード変化を利用することで，小節線の検出も可能にする．並列計算機上に実装して実験した結果，市販のCDによる音響信号中のビート構造を認識できることを確認した．

A Beat Tracking System for Musical Audio Signals

— Bar-line Detection and Musical Knowledge Selection Based on the Presence of Drum-sounds —

Masataka Goto　　Yoichi Muraoka

School of Science and Engineering, Waseda University
3-4-1 Ohkubo Shinjuku-ku, Tokyo 169, JAPAN.

Abstract　This paper presents a real-time beat tracking system that recognizes a hierarchical beat structure in audio signals of popular music. Most previous systems dealt with MIDI signals. Although our two previous systems dealt with audio signals, they were not able to detect bar-lines and were separate: one for music with drums and the other for drumless music. To integrate these systems, we propose a method of judging the presence of drum-sounds, which enables selective application of musical knowledge. We also propose a method of detecting bar-lines by utilizing chord change possibilities. Our experimental results show that our system is robust enough to handle audio signals sampled from compact discs.

1　はじめに

これまで多くの音楽理解関連の研究がなされてきたが，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構築は依然として難しい課題である．計算機による音楽音響信号の理解へ向けた典型的なアプローチは，音響信号から楽譜やMIDIデータなどのシンボル表現を得る自動採譜システムや音源分離システムの研究である．このような詳細な採譜技術は重要であるが，これらのシステムがコンパクトディスク（CD）などによる我々が通常聞くのと同程度の複雑さを持った音響信号を扱うのは，現時点では大変難しい．音符やコード名を同定する能力は音楽的に訓練された人だけが持っていることからもわかるように，採譜のようなシンボル化の能力は，実は人間にとっても獲得するのが難しい高度な技能であると考えられる．

一方，音楽的に訓練されていない普通の人は，音

響信号を楽譜のような表現に変換できなくても，ある程度音楽を理解することができる．例えば，コード名を同定できない人でも，ハーモニーやコードの変化を感じることはできる．たとえすべての楽音を完全に音源分離して同定できない人でも，音楽に合わせて手拍子を打つことは比較的容易にできる．そのため我々は，まず最初に訓練されていない人のように音楽を理解するシステムを構築し，その後訓練された音楽家のように音楽を理解するシステムへと拡張するアプローチが重要であると考える．そこで，まずビートのような基本的なレベルでの音楽理解を実現した後に，より高次の音楽構造の理解を実現する方向へと研究を進めていく．

本研究では，多様な楽器音や歌声の含まれたポピュラー音楽の音響信号に対し，階層的なビート構造をリアルタイムに認識するビートトラッキングシステムを実現する．音楽的に訓練された人と訓練されていない人のいずれにとっても，ビートは西洋音楽を理解する上で基本的な概念であり，ビートトラッキングは計算機による音楽理解モデルを実現する上で重要である．さらに，音楽との同期を必要とする多様なアプリケーションにおいても有用である[1),2)]．本システムが認識するビート構造は，四分音符レベル，二分音符レベル，小節レベルの三つで構成される☆1．つまり，四分音符に相当するビートのパルス列 (四分音符レベル) を得るだけでなく，入力曲が 4/4 拍子であることを前提に，二分音符と小節の先頭の時刻も認識する．

従来のビートトラッキングに関する研究の多くは，MIDI などの音符がシンボル化された演奏情報を対象としていたため，シンボル化の困難な音響信号には適用できなかった[3)~12)]．音響信号を対象とした研究も報告されているが[13)~17)]，その多くは四分音符レベルしか認識できず，CD などによる音響信号をリアルタイムに処理できなかった．一方，我々はこれまで打楽器音☆2を含む音響信号を対象としたシステム[1),18),19)]と打楽器音を含まない音響信号を対象としたシステム[20),21)]を構築し，ポピュラー音楽に対して四分音符／二分音符レベルにおけるビート構造の認識を実現してきた．しかし両者は別々のシステムであり，小節レベルのビート構造の認識はできなかった．

本稿では以下，我々の二つの従来システム (打楽器音あり／なし用) を一つに統合したシステムの実現方法，および小節レベルのビート構造の認識手法 (小節線の検出手法) について述べる．まず，2 においてビートトラッキングの問題を明確にし，それがなぜ難しいのかを考える．次に，3 において我々の解決法をビートトラッキングのモデルとして述べる．その中で，打楽器音の有無に応じた音楽的知識の選択によるシステム統合と，コード変化を利用した小節レベルのビート構造の認識手法も説明する．4 では実装した本システムによる実験結果を示し，実際の音響信号に有効であることを確認する．最後に 5 で結論と今後の課題を述べる．

☆1 本システムは楽譜表現には依存していないが，便宜上楽譜の用語を文献3),4) のように用いる．例えば，四分音符レベルは人間が音楽中に感じる基本的な時間単位を示すが，これは通常楽譜の四分音符に対応している．

☆2 本稿ではポピュラー音楽で多用されるバスドラムとスネアドラムを想定する．

2 ビートトラッキング問題

本研究におけるビートトラッキングの問題設定を示し，ビートトラッキングが，音楽中に非明示的にしか表現されていないビート構造を推定する逆問題であることを述べる．そして，これを解くビートトラッキングシステムを実現する際の主要な課題を述べる．

2.1 問題設定

我々はビートトラッキングを，音楽から四分音符レベル，二分音符レベル，小節レベルで構成される**階層的なビート構造**を得る過程であると定義する (図 1)．これを得るには，まず四分音符に相当するほぼ等間隔なビートの存在する時刻 (ビート時刻) を認識する必要がある．このビート時刻の系列を**四分音符レベル**と呼ぶ．次にさらに上位のビート構造として，二分音符と小節の先頭の時刻を認識する必要がある．二分音符の時刻の系列は**二分音符レベル**と呼ばれ，各ビートが強拍か弱拍か☆3(**二分音符レベルのタイプ**) を判定することで得られる．小節の時刻の系列は**小節レベル**と呼ばれ，各強拍が小節の先頭か途中か (**小節レベルのタイプ**) を判定することで得られる．二分音符レベルと小節レベルの両者のタイプをまとめて**ビートタイプ**と呼ぶ．

本研究では，入力は市販の CD などから得た複数

☆3 4/4 拍子であることを前提に，各小節において 1 拍目と 3 拍目を強拍，2 拍目と 4 拍目を弱拍と定義する．

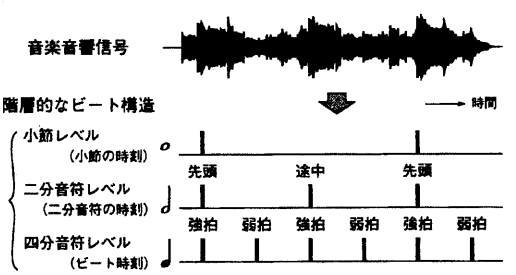

図 1: ビートトラッキング問題

種類の楽器音を含む音楽音響信号とし，そのテンポは打楽器音を含まないとき 61～120 M.M. (Mälzel's Metronome: 四分音符/分)，打楽器音を含むとき 61～185 M.M. の間で曲中を通じてほぼ一定とする．また，入力曲は 4/4 拍子であると仮定する．これらは多くのポピュラー音楽に当てはまる妥当な制約である．

2.2 逆問題としてのビートトラッキング

2.1 のビートトラッキング問題を見通し良く解くために，本研究ではビートトラッキングを演奏 (特にビート構造を演奏音中に示す行為) の逆問題であるととらえる．西洋音楽では，演奏者達の頭の中の階層的なビート構造に従って，時間軸方向に調和した演奏がなされている．これを音楽が生成される順問題☆4と考えると，演奏された音楽から元の頭の中のビート構造を推定するビートトラッキングはその逆問題となる．順問題に相当する演奏では，ビートの構造を特定の音で明示的に (元の構造が一意に決まるように) 表現しているわけではなく，様々な音楽的要素の関係の中に非明示的に表している．しかも，それらの音楽的要素は一定でなく，音楽ジャンルや楽曲によって異なることが多い．

ビートトラッキングが本質的に難しいのは，音楽中に非明示的にしか表現されていない階層的なビート構造を音楽から推定する逆問題であるからである．したがってビートトラッキングの難易度は，どれくらい明示的にビート構造が表現されているかで決まり，単純に楽器数等では決まらない．例えば，楽器数が多くてもそれらが同様に四分音符でビートを明示的に表す演奏をしていれば，容易な逆問題となりうる．また，ジャンルや楽器種により難易度に傾向があるのは，この明示性に傾向があるからだと考えられる．

さらに，上記の音楽は実際には音響信号であるため，演奏の後にさらに楽器発音や音響伝達を経て音楽音響信号が生成される全過程を順問題と考える必要がある．そのため，逆問題であるビートトラッキングには，音響信号から手がかりとなる音楽的要素を抽出する問題も含まれる．この場合には，音響信号中の楽器数が多く音源分離が困難になるほど，一般に抽出が難しくなる．

2.3 実現上の課題

このような逆問題を解くための主要な課題は以下の三つである．

課題 1 手がかりの決定と抽出

ビートトラッキングの手がかりとしてどのような音楽的要素を用いるか決定し，それらの抽出方法を実現する必要がある．

課題 2 手がかりの解釈

音楽的要素の関係から，階層的なビート構造の各レベルを推定する方法を実現する必要がある．

課題 3 解釈の曖昧性の取り扱い

ビート構造は非明示的にしか表現されていないので，手がかりの解釈は一意に決まらず曖昧性がある．そのために，様々な解釈の可能性を調べ，それらの解釈の中から適切なものを決めるために，各解釈がどれくらい適切かを評価する必要がある．

3 ビートトラッキングのモデル

音楽音響信号から階層的なビート構造を推定するビートトラッキングのモデルは，演奏 (ビート提示行為) の逆モデルと，その前段階である音楽的要素を音響信号から抽出する聴覚の順モデル (周波数解析) の両者を含む必要がある (図 2)☆5．つまり，様々な音楽的要素の関係からビート構造を推定する前者のモデルだけでなく，それらの音楽的要素を音響信号から抽出する後者のモデルも必要である．そこで，まず演奏者がビート構造をどう音楽中に示すかという演奏の順モデルを考え，次にその逆モデルを用いることで音楽からビート構造を推定する．その後，実際に逆モデルを用いるために不可欠な音楽的要素を抽出する周波数解析を検討する．

3.1 演奏 (ビート提示行為) の順モデル

ポピュラー音楽における順モデルは，ヒューリスティックな性質として，以下の三つの音楽的要素に関

☆4 一般に，順問題とは原因 (入力) から結果 (出力) を予測する問題であり，逆問題とは結果から原因を推定する問題である．本稿ではこれらの問題を解くためのモデルを，それぞれ順モデル，逆モデルと呼ぶ．

☆5 このモデルは，「階層的なビート構造」を他の理解結果に置き換えることで，より一般的な音楽理解のモデルに拡張できる．

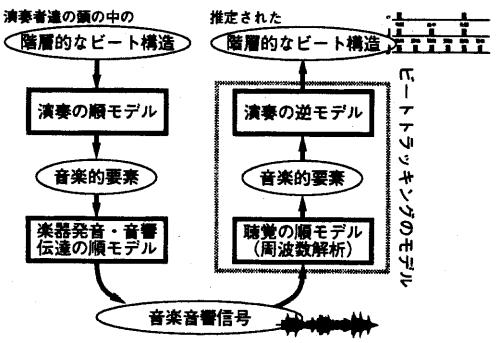

図 2: ビートトラッキングのモデル

する傾向 (時間軸上の配置の傾向) を持っていると本研究では考える．演奏者は，意識的あるいは無意識にこうした順モデルを利用して演奏していると考えられる．

傾向 (a) 発音時刻に関する傾向

「ビート構造上で音を鳴らす傾向がある.」

これは四分音符レベルの構造を示す際に影響し，二分音符／小節レベルでの影響は今回は考えない．つまり，ビート時刻上に発音時刻が多く存在するが，弱拍より強拍に発音時刻が多く存在したり，小節の途中より先頭に発音時刻が多く存在するとは限らない．

傾向 (b) コード変化に関する傾向

「ビート構造上でコードを変える傾向がある.」

これは主に打楽器音がない場合に重要となり，四分音符／二分音符／小節レベルの構造を示す際に影響する．つまり，ビート時刻でコードが変わりやすいだけでなく，特に小節の先頭や強拍の時刻でコードが変わりやすい．

傾向 (c) ドラムパターンに関する傾向

「打楽器音を演奏する場合に，ビート構造を示す典型的なドラムパターンを多用する.」

これは四分音符／二分音符レベルの構造を示す際に影響する．例えば，1 拍目と 3 拍目にバスドラム (BD)，2 拍目と 4 拍目にスネアドラム (SD) が鳴る典型的なドラムパターンは，各 BD と SD の位置がビート時刻で，BD の位置が強拍，SD の位置が弱拍であるという構造を示している．

このような順モデルは音楽ジャンルによって (場合によっては曲ごとに) 異なり，それが音楽ジャンルごとの特徴になっていると我々は考えている．例えば，上記の順モデルに当てはまらない無伴奏独唱のビートトラッキングを実現するには，その音楽でビート構造が演奏音中へどう提示されているかを検討する必要がある．

3.2 逆モデルによるビート構造の推定

3.1 から，ビートトラッキングの手がかりとして，三つの音楽的要素 (発音時刻，コード変化，ドラムパターン) を利用すればよいことがわかる．そこで，これらの手がかりからビート構造を推定するために，**3.1** から得られる妥当な仮定として，以下の三種類の音楽的知識を逆モデルとして用いる．

知識 (a) 発音時刻に基づく推定のための知識

四分音符レベルを推定するために次の二つの知識を用いる.「発音時刻の位置がビート時刻である可能性が高い.」「発音時刻の間隔にビートの間隔が現れやすい.」

知識 (b) コード変化に基づく推定のための知識

各レベルを推定するために次の三つの知識を用いる.「コードの変化点がビートからずれた位置でなくビート時刻である可能性が高い.」「コードの変化点が弱拍でなく強拍である可能性が高い.」「コードの変化点が小節の途中でなく先頭である可能性が高い.」

知識 (c) ドラムパターンに基づく推定のための知識

典型的なドラムパターンが強拍から始まる二拍以上の長さのパターンであることを前提として，音楽から抽出されたパターンが典型的なパターンに良く一致するとき，四分音符／二分音符レベルを推定するために次の二つの知識を用いる.「抽出されたパターンの各拍の長さは適切なビートの間隔である.」「抽出されたパターンの先頭が強拍を示す.」

これらはすべて我々がシステムに事前知識として与えているが，将来的にはシステムが逆モデルとして獲得できるのが望ましい．人間は音楽ジャンルごとの逆モデルを無意識に (ときには意識的に) 獲得していると考えられる☆6.

上記の音楽的知識を用いることで，**2.3** で述べた三つの課題が解決できる．以下，その概要を述べる．

1. 手がかりの決定と抽出の方法

前述したように，発音時刻，コード変化，ドラムパターンを手がかりとする．これらの具体的な抽出方法は **3.3** で述べる．

2. 手がかりの解釈の方法

三種類の音楽的知識を用いて手がかりを解釈する．まず知識 (a) に基づき，発音時刻 (実際には全帯域の発音時刻を同時に考慮するために発音時刻ベクトル[20]を用いる) の自己相関関数からビートの間隔を決定し，発音時刻とビート時刻の系列の相互相関関数から次のビート時刻を予測する．その際に予測場という概念も導入するが，紙面の都合上これらの詳細については文献 22) に委ねて省略する．一方，知識 (b) と (c) は，打楽器音の有無に応じて選択して用いる必要がある．その方法は **3.4** で述べる．

3. 解釈の曖昧性の取り扱いの方法

マルチエージェントアーキテクチャを導入し，異なる戦略で解釈をおこなう複数のエージェントが，様々な解釈の可能性を並列に追跡する[21]．そして各エージェントは，演奏の逆モデルをどれくらい適切に適用できたかによって，解釈の適切さを自己評

☆6 一般化すると,「音楽の原因 (音楽の素: 意図，感情など) が与えられたときに音楽が生成される順モデルの逆モデルを人間は獲得し，それを用いて音楽を理解している」という仮説になる．

価する．この評価値を解釈の確信度と呼び，基本的に全体の中で最も確信度の高い解釈に基づいて出力が決定される．

3.3 音源分離が困難な音響信号からの情報抽出

音響信号から三つの音楽的要素を抽出する処理は，図 2 の聴覚の順モデルに相当する．その中で発音時刻に関しては，まず分割された周波数帯域ごとの発音時刻を検出し，次にそれらを発音時刻ベクトルとしてまとめるだけでよい．具体的な抽出法は文献 22) に記述されている．

一方，複数種類の楽器音を含む複雑な音響信号から，コード変化とドラムパターンをこのようにボトムアップに抽出するのは難しい．そこで我々は，仮に得られたビート時刻をトップダウン情報として用いて周波数解析することで，両者の抽出を可能にする手法を提案する．この仮のビート時刻は，ボトムアップに得られる発音時刻に基づいて，知識 (a) を用いることで推定できる．全体の流れ図を図 3 に示す．

3.3.1 コード変化度

仮のビート時刻をトップダウン情報として活用することで，周波数スペクトルから音符やコード名などにシンボル化せずに，直接コード変化の度合い (コード変化度) を求める．これは，たとえコード名がわからない人でもコードの変化はわかる，という現象に着目した手法である．

コードの構成音の倍音も含めたすべての周波数成分を考えると☆7，コードが変わらない場合はこれらの周波数成分は比較的変化が少ないが，コードが変わる場合には大きく変化することが多い．複数楽器による音響信号中から，すべての周波数成分を正確に求めるのは一般に難しいが，ある一定の期間内で優勢な周波数成分はヒストグラムを計算することで推定できる．

そこで本手法では，FFT(高速フーリエ変換) で得た周波数スペクトル (ここでは 10 Hz から 1 kHz までの帯域だけを考慮する) の時間軸を仮のビート時刻で短冊状に切断し，各短冊内において時間軸方向にパワーを合計してヒストグラムを作成する．ヒストグラムのピークは短冊内で支配的な音高であり，コードやメロディーの周波数成分に相当することが多い．そこで，隣接する短冊間でピークを比較することでコード変化度を求める．前の短冊に比べてより多くのピークやより大きいピークが生じるほど，その間でコードが変化した可能性が高い．

我々のシステムは，四分音符レベルでのコード変化度と八分音符レベルでのコード変化度の二種類を算出する．前者は各四分音符の位置 (仮のビート時刻) でコードがどれくらい変わった可能性があるかを表し，二分音符／小節レベルのビート構造を推定する際に用いられる．後者は八分音符の位置での可能性を表し，四分音符レベルのビート構造を推定する際に用いられる．これらを算出する具体的な式は，文献 22) に詳しく記述されている．

3.3.2 バスドラムとスネアドラムのパターン

まず打楽器音 (BD と SD) の候補をボトムアップな処理により求め，次に仮のビート時刻を用いてパターンを形成する (図 4)．BD と SD の音色は曲ごとに異なるため，事前にテンプレートを与えることはできない．そこで，BD の音は低域に固定した周波数の特徴的な成分を持ち，SD の音はノイズ成分が周波数軸方向に広く分布する特徴を持つことを利用して候補を得る．

BD を検出するために，まず低域の立ち上がり成分

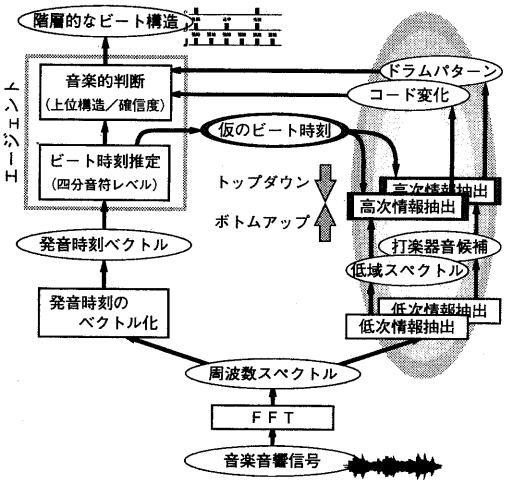

図 3: 仮のビート時刻をトップダウン情報として用いた周波数解析

☆7 実際の曲中では，メロディー等他の音の周波数成分も含めて考える．これらはコードと調和する音高であることが多い．

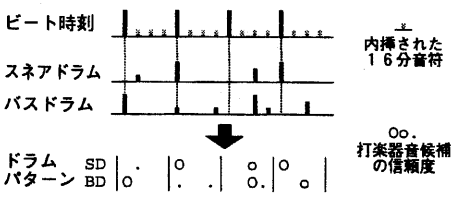

図 4: 仮のビート時刻を用いたドラムパターンの形成

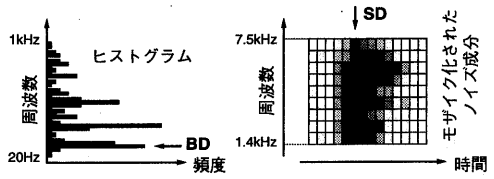

図 5: 打楽器音候補の検出

の周波数軸方向のピークを求め，そのヒストグラムを作成する (図 5)．ヒストグラム中で最も周波数の低いピークを BD の特徴周波数とみなし，立ち上がり成分のピークにその周波数の成分が現れた時刻を BD の発音候補とする．それがどれくらい信頼できるかを表す信頼度は，ピーク値に基づいて決める．

SD を検出するために，まず式 (1) で定義されるノイズ成分 $n(t,f)$ の周波数軸を粗く分割して帯域ごとの合計値を求め (モザイク化)，大きい値が周波数軸方向に連続している時刻を求める (図 5)．

$$n(t,f) = \begin{cases} p(t,f) & (\text{if } p(t,f) < 2\min(hp,lp)) \\ 0 & (\text{otherwise}) \end{cases} \quad (1)$$

$$hp = (p(t,f+2) + \sum_{i=-1}^{1} p(t+i,f+1))/4 \quad (2)$$

$$lp = (p(t,f-2) + \sum_{i=-1}^{1} p(t+i,f-1))/4 \quad (3)$$

ただし，時刻 t，周波数 f における周波数スペクトルのパワーを $p(t,f)$ とする．

1. ノイズ成分のモザイク化

   ノイズ成分 $n(t,f)$ を NoiseBAND 個の帯域に分割し (現在の実装では NoiseBAND=16 で各帯域の幅は 689.06 Hz とした)，各帯域内の合計値 $N(t,F)$ を求める．ただし，F ($0 \leq F <$ NoiseBAND) は周波数帯域の番号とする．

   $$N(t,F) = \sum_{f \text{ in } F\text{番の帯域}} \sum_{i=-1}^{1} n(t+i,f) \quad (4)$$

2. 周波数軸方向に連続して分布している成分の強調

   F が SDlow(=2) から SDhigh(=10) の範囲の $N(t,F)$ を掛け合わせ，周波数成分の連続度 $C(t)$ を求める．

   $$C(t) = \prod_{F=\text{SDlow}}^{\text{SDhigh}} \text{limiter}(t, N(t,F)) \quad (5)$$

   ここで，$\text{limiter}(t,N(t,F))$ は，時刻 t において SDn(=4) 番目に大きい $N(t,F)$ が上限となるように，それより大きいものを同じ値に制限する処理である．こうして局所的に大きな値を抑えることで，周波数成分の連続性が適切に $C(t)$ に反映される．もし周波数成分が連続していないときは，0 に近い $N(t,F)$ によって $C(t)$ が小さくなる．

以上の手順で求めた連続度 $C(t)$ のピーク時刻を，SD の発音候補とする．その信頼度は，$C(t)$ のピーク値に基づいて決める．

3.4 打楽器音の有無に応じた音楽的知識の選択

音楽的知識 (b) と (c) は，打楽器音の有無に応じて選択して用いる必要がある．そこで打楽器音の有無の判定手法を **3.4.1** で提案し，その判定結果を用いてビート構造を推定する方法を **3.4.2** で述べる．

3.4.1 打楽器音の有無の判定手法

打楽器音の検出結果は誤検出が多く含まれるため，単純に検出結果があるかどうかで打楽器音の有無を判定することはできない．そこで，ポピュラー音楽においてスネアドラム (SD) は弱拍 (2 拍目と 4 拍目) で演奏されることが多いことに着目する．検出された SD の候補に対し，窓つき (現在の実装は過去 10 秒間) 自己相関関数を計算し，その相関値が高いときは打楽器音があり，低いときは打楽器音がないと判定する．そのための閾値は予備実験により経験的に定めた．ただし検出した SD の時刻が揺らぐことを考慮し，ダウンサンプリングにより時間軸の分解能を粗くして計算する (現在は分解能 46.4 msec)．

音響信号に打楽器音が含まれていなく，SD 以外の音 (歌声の子音が誤検出されやすい) を誤検出している場合には，この相関値は低くなる．一方，打楽器音が含まれている場合でも，SD が一時的に演奏されなかったときには相関値が低くなってしまう．これは，一旦高い相関値が得られた後は，しばらく打楽器音がある状態が続くとみなすことで対処する．

3.4.2 音楽的知識の選択

ビート構造の各レベルを推定するために，知識 (b) と (c) を表 1 のように適用する．打楽器音がないときは知識 (b) (コード変化) だけを用いるが，打楽器音があるときは知識 (c) (ドラムパターン) も用いる．

[四分音符レベル]

仮のビート時刻は知識 (a) により既に求まっているので，それが適切かどうかだけを確信度として評価する．打楽器音なしのときは，八分音符レベルでのコード変化度が，八分音符ずれた位置よりもビートの位置で大きいほど確信度を上げる．打楽器音ありのときは，抽出したドラムパターンが，システムに事

表 1: 打楽器音の有無に応じた音楽的知識の選択

ビート構造	打楽器音なし	打楽器音あり
小節レベル	四分音符レベルでのコード変化度	四分音符レベルでのコード変化度
二分音符レベル	四分音符レベルでのコード変化度	ドラムパターン
四分音符レベル	八分音符レベルでのコード変化度	ドラムパターン

前に登録された典型的なドラムパターンと一致するほど確信度を上げる.

[二分音符レベル]

二分音符レベルのタイプ (四分音符レベルの各ビート時刻が強拍か弱拍か) を判定する. 打楽器音なしのときは, 四分音符レベルでのコード変化度が他よりも十分大きい位置を強拍とみなし, 打楽器音ありのときは, 典型的なドラムパターンと良く一致したときの先頭時刻を強拍とみなす. あとは強拍と弱拍が交互に現れる性質を利用して決定する.

[小節レベル]

小節レベルのタイプ (各強拍が小節の先頭か途中か) を判定する. 打楽器音の有無を問わず, 四分音符レベルでのコード変化度が, 弱拍の位置よりも強拍の位置で十分大きければ小節の先頭とみなす.

3.5 処理全体の流れ

処理全体の流れを図 6 に示す. まず, 音響信号の入力で A/D 変換された音響信号に対して周波数解析をおこなう. 発音時刻の検出器が各周波数帯域ごとの発音時刻を求め, ベクトル化器がそれらを発音時刻ベクトルにまとめる. 次に, ビート予測の各エージェント (現在の実装では全 12 個) が, 知識 (a) に基づいて発音時刻ベクトルからビートの間隔を求め, 次のビート時刻を予測する. エージェントと一対一に対応する高次情報抽出器は, 前述したようにこのビート時刻をトップダウン情報として周波数解析し, コード変化度とドラムパターンをエージェントへ送り返す. これらを受け取ったエージェントは, 知識 (b) と (c) に基づいてビートタイプを判定し確信度を評価する. そして解釈の統合処理では, 全エージェントの解釈から最も確信度の高い解釈に基づいて, ビート情報 (ビート時刻, ビートタイプ, 現在のテンポから成る) を生成する. 最後にビート情報の出力が, ネットワークを通じて他のアプリケーションプログラムへとビート情報を送信する.

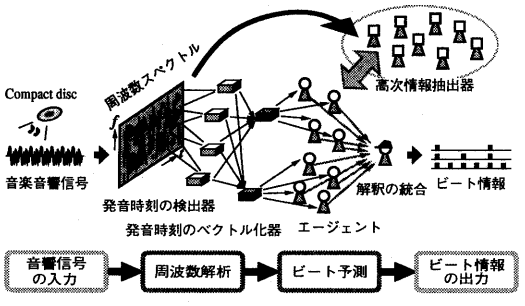

図 6: 処理全体の流れ

4 実験結果

3 のモデルをリアルタイムに実行するシステムを, 富士通の分散メモリ型並列計算機 AP1000 上に実装し, 本システムの有効性を確認する実験をおこなった. また, 実験で対象とした曲の難易度の評価も試みた.

4.1 システムの認識精度

実験には, 市販の CD からサンプリングしたモノラルの音響信号を用いた. 4/4 拍子でテンポがほぼ一定なポピュラー音楽の, 最初の 1〜2 分間を入力して実験した. 対象曲は, 統合前のシステムの実験に用いていた, 打楽器音を含まない 40 曲 (テンポ範囲: 62〜116 M.M., アーティスト数: 28) と打楽器音を含む 45 曲 (テンポ範囲: 67〜185 M.M., アーティスト数: 32) の全 85 曲とした.

ビート構造の各レベルにおける認識精度の実験結果を表 2 に示す. ビートトラッキングの正誤の判定は, 誤差の平均と標準偏差がビートの間隔の 10%より小さく, 誤差の最大がビートの間隔の 17.5%より小さく, ビートトラッキングを開始した時刻が曲の先頭から 45 秒未満であるときに正解とした. ただし, 誤差を求める際に基準となる正しい階層的なビート構造は, 我々が開発したビート情報エディタ[23]を用いて人間が手作業で作成した. 誤差の評価尺度については, 文献 23) に詳しく記述されている.

本システムが誤認識した曲の中には, ほぼ正しいビート構造を推定していたが, 誤差が判定基準よりも多少大きかったり, 正しくトラッキングし始めた時刻が遅過ぎたりした曲が多かった. 四分音符レベルでの誤りは, シンコペーションで 8 分音符ずれたり, 音数が非常に少なくてずれてしまったりしたのが原因だった. 二分音符/小節レベルでの誤りは, コード変化の仕方が知識 (b) と合致しなかったか, コードに相当する成分が小さ過ぎて変化を適切に検出できなかったのが原因だった.

4.2 対象曲の難易度

2.2 で述べたように, ビートトラッキングの難易度はビート構造がどれだけ明示的に表現されているか

表 2: ビート構造の各レベルにおける認識精度の実験結果

ビート構造	打楽器音なし	打楽器音あり
小節レベル	31 曲/34 曲 (91.2%)	34 曲/39 曲 (87.2%)
二分音符レベル	34 曲/35 曲 (97.1%)	39 曲/39 曲 (100%)
四分音符レベル	35 曲/40 曲 (87.5%)	39 曲/45 曲 (86.7%)

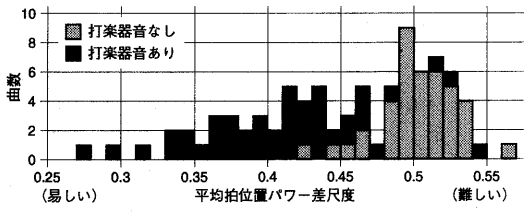

図 7: 対象曲の難易度の評価例

で評価すべきであるが，現実の曲には様々な要因があるためにそれは難しい．そこで四分音符レベルにおける難易度の目安として，平均拍位置パワー差尺度を考案した．以下その概要を述べる．詳細は文献23)に委ねて省略する．

本尺度では，ビートの位置でのパワーに比べてビート以外の位置でのパワーが大きいほど，難易度が高いと考える．そこで，各ビートに対して，ビートの間隔の前半25%(ビート位置) と後半75%(それ以外) のそれぞれの区間で，入力信号のパワーの極大値 L^b，L^o を求める．次に，$0.5\,(L^o - L^b)/max(L^o, L^b) + 0.5$ を全ビートに対して計算する．この平均値 D を平均拍位置パワー差尺度と呼ぶ．メトロノームを入力すると D は0になり，ビート位置とそれ以外とのパワーが等しければ0.5になる．実験に用いた85曲に対して本尺度を計算した結果を図7に示す．

5 おわりに

本稿では，打楽器音の有無を問わず音楽の音響信号をビートトラッキングできるシステムについて，実現上の課題とそれを解決するモデルを中心に述べた．本モデルは演奏 (ビート提示行為) の逆モデルを音楽的知識として利用することで，階層的なビート構造を推定する特徴を持つ．トップダウン情報を用いた周波数解析によって，複雑な音響信号からの情報抽出が可能になり，打楽器音の有無を判定することで適切な音楽的知識を選択的に適用できた．実験の結果，市販のCDによる複数の楽器で演奏されたポピュラー音楽の音響信号に対し，小節レベルまでの階層的なビート構造をリアルタイムに推定できることを確認した．

今後は，他の音楽ジャンルやテンポ変化に対応するための音楽的知識を検討すると共に，より高次の音楽構造を理解できるモデルへと発展させていく予定である．

謝 辞

本稿に関し有益な御意見を頂いた早稲田大学の奥椙 正克 氏に深く感謝する．また，AP1000の実行環境を提供して頂いた富士通研究所 並列処理研究センターに感謝する．

参考文献

[1] Goto, M. and Muraoka, Y.: A Beat Tracking System for Acoustic Signals of Music, *Proc. of the Second ACM Intl. Conf. on Multimedia*, pp. 365–372 (1994).

[2] 後藤真孝: 計算機は音楽に合わせて手拍子を打てるか? — リアルタイムビートトラッキングシステム —, *bit*, Vol. 28, No. 3, pp. 4–11 (1996).

[3] Rosenthal, D.: Emulation of Human Rhythm Perception, *Computer Music Journal*, Vol. 16, No. 1, pp. 64–76 (1992).

[4] Rosenthal, D.: *Machine Rhythm: Computer Emulation of Human Rhythm Perception*, PhD Thesis, Massachusetts Institute of Technology (1992).

[5] Dannenberg, R. B. and Mont-Reynaud, B.: Following an Improvisation in Real Time, *Proc. of the 1987 ICMC*, pp. 241–248 (1987).

[6] Desain, P. and Honing, H.: The Quantization of Musical Time: A Connectionist Approach, *Computer Music Journal*, Vol. 13, No. 3, pp. 56–66 (1989).

[7] Desain, P. and Honing, H.: Advanced issues in beat induction modeling: syncopation, tempo and timing, *Proc. of the 1994 ICMC*, pp. 92–94 (1994).

[8] Allen, P. E. and Dannenberg, R. B.: Tracking Musical Beats in Real Time, *Proc. of the 1990 ICMC*, pp. 140–143 (1990).

[9] Driesse, A.: Real-Time Tempo Tracking Using Rules to Analyze Rhythmic Qualities, *Proc. of the 1991 ICMC*, pp. 578–581 (1991).

[10] Rowe, R.: *Interactive Music Systems*, The MIT Press (1993).

[11] Large, E. W.: Beat Tracking with a Nonlinear Oscillator, *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music*, pp. 24–31 (1995).

[12] Smith, L. M.: Modelling Rhythm Perception by Continuous Time-Frequency Analysis, *Proc. of the 1996 ICMC*, pp. 392–395 (1996).

[13] Schloss, W. A.: *On The Automatic Transcription of Percussive Music — From Acoustic Signal to High-Level Analysis*, PhD Thesis, CCRMA, Stanford Univ. (1985).

[14] Katayose, H., Kato, H., Imai, M. and Inokuchi, S.: An Approach to an Artificial Music Expert, *Proc. of the 1989 ICMC*, pp. 139–146 (1989).

[15] Vercoe, B.: Perceptually-based music pattern recognition and response, *Proc. of the Third Intl. Conf. for the Perception and Cognition of Music*, pp. 59–60 (1994).

[16] Todd, N. P. M.: The Auditory "Primal Sketch": A Multiscale Model of Rhythmic Grouping, *Journal of New Music Research*, Vol. 23, No. 1, pp. 25–70 (1994).

[17] Scheirer, E. D.: Using bandpass and comb filters to beat-track digital audio (1996). (unpublished)

[18] Goto, M. and Muraoka, Y.: A Real-time Beat Tracking System for Audio Signals, *Proc. of the 1995 ICMC*, pp. 171–174 (1995).

[19] 後藤真孝, 村岡洋一: ビートトラッキングシステムの並列計算機への実装 — AP1000によるリアルタイム音楽情報処理 —, 情処学論, Vol. 37, No. 7, pp. 1460–1468 (1996).

[20] 後藤真孝, 村岡洋一: 音響信号に対するリアルタイムビートトラッキングシステム — 打楽器音を含まない音楽に対するビートトラッキング —, 情処研報 音楽情報科学 96-MUS-16-3, Vol. 96, No. 75, pp. 13–20 (1996).

[21] Goto, M. and Muraoka, Y.: Beat Tracking based on Multiple-agent Architecture — A Real-time Beat Tracking System for Audio Signals —, *Proc. of the Second Intl. Conf. on Multiagent Systems*, pp. 103–110 (1996).

[22] Goto, M. and Muraoka, Y.: Real-time Rhythm Tracking for Drumless Audio Signals — Chord Change Detection for Musical Decisions —, *Working Notes of the IJCAI-97 Workshop on Computational Auditory Scene Analysis* (1997). (in press)

[23] Goto, M. and Muraoka, Y.: Issues in Evaluating Beat Tracking Systems, *Working Notes of the IJCAI-97 Workshop on Issues in AI and Music* (1997). (in press)