(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4767691号
(P4767691)

(45)発行日　平成23年9月7日(2011.9.7)　(24)登録日　平成23年6月24日(2011.6.24)

(51)Int.Cl.		F I		
G10G 3/04	(2006.01)	G10G	3/04	
G10H 1/40	(2006.01)	G10H	1/40	
G10H 1/00	(2006.01)	G10H	1/00	B
G10H 1/38	(2006.01)	G10H	1/38	Z
G10L 11/00	(2006.01)	G10L	11/00	402K

請求項の数 8　(全 28 頁)

(21)出願番号　特願2006-1194(P2006-1194)
(22)出願日　平成18年1月6日(2006.1.6)
(65)公開番号　特開2007-52394(P2007-52394A)
(43)公開日　平成19年3月1日(2007.3.1)
審査請求日　平成20年10月20日(2008.10.20)
(31)優先権主張番号　特願2005-208062(P2005-208062)
(32)優先日　平成17年7月19日(2005.7.19)
(33)優先権主張国　日本国(JP)

(73)特許権者　000001410
株式会社河合楽器製作所
静岡県浜松市中区寺島町２００番地
(74)代理人　100086863
弁理士　佐藤　英世
(72)発明者　澄田　錬
静岡県浜松市寺島町２００番地　株式会社河合楽器製作所内

審査官　大野　弘

最終頁に続く

(54)【発明の名称】テンポ検出装置、コード名検出装置及びプログラム

(57)【特許請求の範囲】

【請求項１】

音響信号を入力する入力手段と、

入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出手段と、

この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、

このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と

を有すると共に、

上記小節検出手段により、拍子と小節線位置を求めるにあたり、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値の自己相関から拍子を求め、さらに、ビート毎の全体の音の変化度合いを示す値が最も大きな箇所を１拍目として、小節線位置に設定することを特徴とするテンポ検出装置。

【請求項２】
音響信号を入力する入力手段と、
入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出手段と、
この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、
このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と
を有しており、
上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔も、平均的な上記ビート間隔で周期を持つ関数との相互相関を計算して、求めると共に、
上記小節検出手段により、拍子と小節線位置を求めるにあたり、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値の自己相関から拍子を求め、さらに、ビート毎の全体の音の変化度合いを示す値が最も大きな箇所を１拍目として、小節線位置に設定することを特徴とするテンポ検出装置。
【請求項３】
音響信号を入力する入力手段と、
入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出手段と、
この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、
このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と
を有しており、
上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔を、平均的な上記ビート間隔に＋α又は－αの間隔を加算した関数との相互相関を計算して求めると共に、
上記小節検出手段により、拍子と小節線位置を求めるにあたり、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値の自己相関から拍子を求め、さらに、ビート毎の全体の音の変化度合いを示す値が最も大きな箇所を１拍目として、小節線位置に設定することを特徴とするテンポ検出装置。
【請求項４】
音響信号を入力する入力手段と、

入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出手段と、
この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、
このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と
を有しており、
上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔を、平均的な上記ビート間隔から次第に広くなる又は次第に狭くなる間隔にした関数との相互相関を計算して求めると共に、
上記小節検出手段により、拍子と小節線位置を求めるにあたり、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値の自己相関から拍子を求め、さらに、ビート毎の全体の音の変化度合いを示す値が最も大きな箇所を１拍目として、小節線位置に設定することを特徴とするテンポ検出装置。
【請求項５】
音響信号を入力する入力手段と、
入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出手段と、
この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、
このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と
を有しており、
上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔を、平均的な上記ビート間隔から次第に広くなる又は次第に狭くなる間隔にした関数との相互相関を、その途中のビート位置をずらして計算することにより求めると共に、
上記小節検出手段により、拍子と小節線位置を求めるにあたり、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値の自己相関から拍子を求め、さらに、ビート毎の全体の音の変化度合いを示す値が最も大きな箇所を１拍目として、小節線位置に設定することを特徴とするテンポ検出装置。
【請求項６】
音響信号を入力する入力手段と、

入力された音響信号から、所定の時間間隔で、ビート検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第１の音階音レベル検出手段と、

この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、

このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と、

上記入力された音響信号から、先のビート検出の時とは異なる別の所定の時間間隔で、コード検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第２の音階音レベル検出手段と、

検出した各音階音のレベルのうち、各小節内における低域側の音階音のレベルからベース音を検出するベース音検出手段と、

検出したベース音と各音階音のレベルから各小節のコード名を決定するコード名決定手段と

を有することを特徴とするコード名検出装置。

【請求項７】

上記ベース音検出手段において、ベース音が小節内で複数検出される場合は、そのベース音検出結果に応じて、上記コード名決定手段は、小節を幾つかのコード検出範囲に分断し、この各コード検出範囲におけるコード名を、ベース音と各コード検出範囲における各音階音のレベルから、決定する請求項６記載のコード名検出装置。

【請求項８】

コンピュータを、

音響信号を入力する入力手段と、

入力された音響信号から、所定の時間間隔で、ビート検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第１の音階音レベル検出手段と、

この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、

このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と、

上記入力された音響信号から、先のビート検出の時とは異なる別の所定の時間間隔で、コード検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第２の音階音レベル検出手段と、

検出した各音階音のレベルのうち、各小節内における低域側の音階音のレベルからベース音を検出するベース音検出手段と、

検出したベース音と各音階音のレベルから各小節のコード名を決定するコード名決定手段と

して機能させることを特徴とするコード名検出用プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、テンポ検出装置、コード名検出装置及びプログラムに関する。

【背景技術】
【０００２】
従来の自動伴奏装置では、あらかじめ演奏するテンポを使用者が設定し、このテンポに従って自動演奏が行われる。従って、この自動伴奏に合わせて演奏者が演奏する場合、この自動伴奏のテンポに合わせて演奏する必要があり、これは特に演奏初心者にとっては非常に困難なことであった。そのため、演奏者の演奏音から自動的にテンポを検出し、このテンポに合わせて自動伴奏を行うような自動伴奏装置が望まれていた。
【０００３】
また、演奏音が収録された音楽ＣＤ等の音源からコード名や音符情報を検出する採譜装置において、その演奏音からテンポを検出する機能はその前段階の処理として必須である。
【０００４】
このようなテンポ検出装置として、例えば、下記特許文献１のテンポ検出装置がある。
【０００５】
この特許文献１のテンポ検出装置は、外部から入力される演奏音の１音毎の音程、音量、及び発音のタイミングを表す演奏情報に基づいて、音量に起因するアクセントと、音量以外の音楽要素に起因するアクセントを検出し、これら双方のアクセントに基づいて演奏情報のテンポの変化を予測し、予測されたテンポに内部で生成するテンポを追従させるテンポ変更手段を備えたテンポ検出装置である。従って、テンポ検出するためには、音符情報が検出されていなければならず、ＭＩＤＩ等の音符情報を出力する機能を持った楽器で演奏された場合は、これを簡単に取得できるが、これを持たない一般の楽器で演奏された場合は、演奏音から音符情報を検出するという採譜技術が必要になる。
【０００６】
ＭＩＤＩ等の音符情報を出力する機能を持たない一般楽器の演奏音、すなわち音響信号を入力とするテンポ検出装置の例としては、下記特許文献２に示される構成がある。
【０００７】
該特許文献２に示される構成では、入力される音響信号を時分割でディジタルフィルタリング処理を行って音階を抽出し、検出した音階音のエンベロープ値に基づいてその音階音の発生周期を検出し、この音階音の発生周期とあらかじめ指定された入力音響信号の拍子に基づいてテンポを検出している。このテンポ検出装置は、音符情報を検出しないので、コード名や音符情報を検出する採譜装置の前処理としても使用できる。
【０００８】
同様なテンポ検出装置として、後述する非特許文献１がある。
【０００９】
他方、ポピュラー系の音楽においてコードは非常に重要な要素であり、このようなジャンルの音楽を小編成のバンドで演奏する場合にも、演奏する個々の音符が書かれた楽譜は使用しないで、コード譜またはリードシートと呼ばれるメロディとコード進行のみが書かれた楽譜を使用することが通常である。よって市販のＣＤ等の曲をバンドで演奏するためには曲のコード進行を採譜する必要があるが、この作業は特別な音楽的知識を有する専門家のみが可能であり、一般の人には不可能であった。そこで、市販のパーソナルコンピュータなどを使用して音楽音響信号からコード名を検出する自動採譜装置が求められていた。
【００１０】
このような音楽音響信号からコードを検出する装置として、下記特許文献３の構成がある。同文献の構成では、パワー・スペクトルの計算結果から基本周波数候補を抽出し、この基本周波数候補から倍音と思われるものを除去して音符情報を検出し、この音符情報から和音を検出している。
【００１１】
しかし、該特許文献３に示す構成では、上記倍音を除去する作業は、楽器の種類による倍音構造の違い、打鍵強さによる倍音の出方の違い、時間による倍音のパワー変化、同じ

周波数を倍音成分として持つ音同士の位相干渉の問題などから非常に困難であることが知られている。即ち、この音符情報を検出するという工程が、多くの楽器や歌唱などが混じった一般の音楽ＣＤなどの音源で必ずしも正しく機能するとは考えられない。

【００１２】

同様に音楽音響信号からコードを検出する装置として、後述する特許文献４の構成がある。該特許文献４の構成では、入力される音響信号に対して、異なる特性のディジタルフィルタリング処理を時分割で行い、各音階音のレベルを検出し、検出したレベルのうちオクターブ内で同じ音階関係にあるレベル同士を積算して、その積算レベルのうち値が大きい所定数を使って和音検出をしている。この方法では音響信号に含まれる個々の音符情報を検出しないので、特許文献３にあげたような問題は発生しない。

【特許文献１】特許第３２３１４８２号

【特許文献２】特許第３１２７４０６号

【非特許文献１】後藤真孝著「リアルタイムビートトラッキングシステム」（共立出版 コンピュータサイエンス誌 bit Vol.28 No.3 1996年）

【特許文献３】特許第２８７６８６１号

【特許文献４】特許第３１５６２９９号

【発明の開示】

【発明が解決しようとする課題】

【００１３】

ところが、上記特許文献２のテンポ装置では、音階音のエンベロープからその音階音の発生周期を検出する部分が、エンベロープ値の最大値を検出し、その最大値に対して所定割合以上の部分を検出することにより行う構成となっている。しかし、このように所定割合を一意に定めてしまうと、音量の大小によって発音タイミングを検出できたりできなかったりしてしまい、それが最終的なテンポの決定に大きな影響を与えてしまうという問題を抱えている。

【００１４】

上記非特許文献１に示されるビートトラッキングシステムも、音響信号をＦＦＴして得られた周波数スペクトルから音の立ち上がり成分を抽出するので、先の特許文献２のテンポ検出装置と同様、この立ち上がりを検出できるかどうかが最終的なテンポの決定に大きな影響を与えてしまう。

【００１５】

また、これら２つのテンポ検出装置に共通して言えることは、この音の立ち上がりを検出するのをどの音階音、あるいは、周波数で行うのかということである。たまたま検出する音階音（周波数）で細かいリズムを刻んでいる曲があった場合、間違えて速いテンポを検出してしまう問題があった。

【００１６】

他方、音楽音響信号からコードを検出する上記特許文献４に示す構成では、各音階音のレベルをオクターブ内で同じ音階関係にあるもの同士、つまり１２の音名毎に積算してしまっているので、同じ構成音からなる複数の和音、例えばラ、ド、ミ、ソからなるＡｍ７とド、ミ、ソ、ラからなるＣ６という２つのコードを判別することができない。

【００１７】

また、この特許文献４の和音検出装置には、テンポや小節の検出機能はなく、和音検出は所定のタイミング毎に行うとなっている。つまり、あらかじめ曲のテンポを設定してそのテンポで発音するメトロノームに合わせて演奏するようなケースを想定しており、音楽ＣＤ等のような演奏後の音響信号に適用した場合、一定時間間隔毎のコード名は検出できるが、テンポや小節を検出していないので、コード譜またはリードシートと呼ばれているような各小節のコード名が書かれた楽譜のような形式に出力することはできない。

【００１８】

仮に曲のテンポを与えたとしても、一般的に音楽ＣＤに収録されている演奏のテンポは一定ではなく多少揺らぐため、正しく小節毎のコードを検出することはできない。

【００１９】
また、一定のテンポで発音されるメトロノームなどに合わせて正確なテンポで演奏することは初心者の演奏者にとっては非常に困難であり、一般的には演奏のテンポは揺らいでしまうのが通常である。
【００２０】
さらに、特許文献４の構成では、入力される音響信号に対して、異なる特性のディジタルフィルタリング処理を時分割で行う構成が採用されているが、この構成の採用理由として、ＦＦＴ演算では低域で周波数分解が悪いことをあげている。しかし、入力音響信号をダウンサンプリングしてＦＦＴを行うことで低域でもある程度の周波数分解能を得ることは可能であるし、ディジタルフィルタリング処理では、フィルタ出力信号のレベルを求めるためにエンベロープ抽出部が必要になってしまうのに対し、ＦＦＴでは、ＦＦＴ後のパワーそのものが各周波数でのレベルを表しているためそのようなものは必要なく、ＦＦＴポイント数とシフト量のパラメータを適宜選ぶことで周波数分解能や時間分解能を自由に設定できるメリットもある。
【００２１】
本発明は、以上のような問題に鑑み創案されたもので、人間が演奏したテンポの揺らぐ演奏の音響信号から、曲全体の平均的なテンポと正確なビート（拍）の位置、さらに曲の拍子と１拍目の位置を検出することが可能なテンポ検出装置を提供せんとするものである。
【００２２】
またもう１つの本発明の構成は、特別な音楽的知識を有する専門家でなくても、音楽ＣＤ等の複数の楽器音の混ざった音楽音響信号（オーディオ信号）から、コード名（和音名）を検出することができるコード名検出装置を提供することを目的とする。
【００２３】
さらに詳しくは、入力された音響信号に対し、個々の音符情報を検出することなしに、全体の響きから、コードを決定することができるコード名検出装置を提供することを目的とする。
【００２４】
加えて、構成音が同じ和音でも判別可能で、演奏のテンポが揺らいでしまった場合や、逆にわざとテンポを揺らして演奏しているような音源に関しても、小節毎の和音が検出可能なコード名検出装置を提供することを目的とする。
【００２５】
以上のように、本発明構成では、簡単な構成のみでビート検出という時間分解能が必要な処理（上記テンポ検出装置の構成と同じ）と、和音検出という周波数分解能が必要な処理（上記テンポ検出装置の構成を基にさらに和音を検出できる構成）を同時に行うことができるコード名検出装置を提供することを目的とする。
【００２６】
併せて、これらの装置をコンピュータ上に実現できるテンポ検出用及びコード名検出用のコンピュータ・プログラムについても、提供する。
【課題を解決するための手段】
【００２７】
そのため本発明に係るテンポ検出装置は、
音響信号を入力する入力手段と、
入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出手段と、
この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、
このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レ

ベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と
を有することを前提構成としている。

【００２８】

上記構成によれば、入力手段に入力された音響信号から所定の時間毎の各音階音のレベルを音階音レベル検出手段によって求め、上記ビート検出手段によって、この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、同じくビート検出手段により、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート（拍）間隔（つまりテンポ）と各ビートの位置を検出し、次に上記小節検出手段により、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す上記値求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置（1拍目の位置）を検出することになる。

【００２９】

すなわち、入力された音響信号から所定の時間毎の各音階音のレベルを求め、この所定の時間毎の各音階音のレベルの変化から平均的なビート（拍）間隔（つまりテンポ）と各ビートの位置を検出し、次にこのビート毎の各音階音のレベルの変化から拍子と小節線位置（１拍目の位置）を検出することになる。

このような構成を前提構成として有しており、請求項１のテンポ検出装置の構成は、上記小節検出手段により、拍子と小節線位置を求めるにあたり、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値の自己相関から拍子を求め、さらに、ビート毎の全体の音の変化度合いを示す値が最も大きな箇所を１拍目として、小節線位置に設定することを特徴としている。

請求項２の構成は、上記請求項１の構成に、さらに、上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔も、平均的な上記ビート間隔で周期を持つ関数との相互相関を計算して、求めるというものである。

請求項３の構成は、上記請求項１の構成に、さらに、上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔を、平均的な上記ビート間隔に＋α又は－αの間隔を加算した関数との相互相関を計算して求めるというものである。

請求項４の構成は、上記請求項１の構成に、さらに、上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔を、平均的な上記ビート間隔から次第に広くなる又は次第に狭くなる間隔にした関数との相互相関を計算して求めるというものである。

請求項５の構成は、上記請求項１の構成に、さらに、上記ビート検出手段による平均的なビート間隔と各ビートの位置を検出するにあたり、各音階音のレベル増分値合計の自己相関から平均的なビート間隔を求め、次にこの各音階音のレベル増分値の合計と、上記平均的なビート間隔で周期を持つ関数との相互相関を計算して、最初のビート位置を求め、さらに、２番目以降のビート間隔を、平均的な上記ビート間隔から次第に広くなる又は次

第に狭くなる間隔にした関数との相互相関を、その途中のビート位置をずらして計算することにより求めるというものである。

【００３０】

またコード名検出装置の構成は、

音響信号を入力する入力手段と、

入力された音響信号から、所定の時間間隔で、ビート検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第１の音階音レベル検出手段と、

この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、

このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と、

上記入力された音響信号から、先のビート検出の時とは異なる別の所定の時間間隔で、コード検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第２の音階音レベル検出手段と、

検出した各音階音のレベルのうち、各小節内における低域側の音階音のレベルからベース音を検出するベース音検出手段と、

検出したベース音と各音階音のレベルから各小節のコード名を決定するコード名決定手段と

を有することを特徴としている。

【００３１】

また上記ベース音検出手段において、ベース音が小節内で複数検出される場合は、そのベース音検出結果に応じて、上記コード名決定手段は、小節を幾つかのコード検出範囲に分断し、この各コード検出範囲におけるコード名を、ベース音と各コード検出範囲における各音階音のレベルから、決定するものとする。

【００３２】

上記構成によれば、入力手段から入力された入力音響信号に対し、第１の音階音レベル検出手段により、所定の時間間隔で、まずビート検出に適したパラメータでＦＦＴ演算を行い、これにより所定の時間毎の各音階音のレベルを求め、ビート検出手段により、この所定の時間毎の各音階音のレベルの変化から平均的なビート間隔と各ビートの位置を検出する。次に、小節検出手段により、このビート毎の各音階音のレベルの変化から拍子と小節線位置を検出する。さらに、本発明のコード名検出装置は、第２の音階音レベル検出手段により、入力音響信号に対し先のビート検出の時とは異なる別の所定の時間間隔で、今度はコード検出に適したパラメータでＦＦＴ演算を行い、これにより所定の時間毎の各音階音のレベルを求める。そしてベース音検出手段により、この各音階音のレベルの内、低域側の音階音のレベルから各小節のベース音を検出し、コード名決定手段により、検出したベース音と各音階音のレベルから各小節のコード名を決定することになる。

【００３３】

また上記のように、ベース音検出手段でこのベース音が小節内で複数検出される場合は、そのベース音検出結果に応じて、上記コード名決定手段は、小節を幾つかのコード検出範囲に分断し、この各コード検出範囲におけるコード名をベース音と各コード検出範囲における各音階音のレベルから決定することになる。

【００３７】

さらに、請求項８の構成は、請求項６記載の構成を、コンピュータに実行させるために、該コンピュータで実行可能なプログラム自身を規定している。すなわち、コンピュータに上記各手段を実現させるためのプログラムが該コンピュータに読み出されると、請求項

６に規定された各機能実現手段と同様な機能実現手段が達成されることになる。
この場合、コンピュータとは中央演算処理装置の構成を含んだ汎用的なコンピュータの構成の他、特定の処理に向けられた専用機などを含むものであっても良く、中央演算処理装置の構成を伴うものであれば特に限定はない。

【００３８】
請求項８のより具体的構成は、
コンピュータを、
音響信号を入力する入力手段と、
入力された音響信号から、所定の時間間隔で、ビート検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第１の音階音レベル検出手段と、
この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出手段と、
このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出手段と、
上記入力された音響信号から、先のビート検出の時とは異なる別の所定の時間間隔で、コード検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める第２の音階音レベル検出手段と、
検出した各音階音のレベルのうち、各小節内における低域側の音階音のレベルからベース音を検出するベース音検出手段と、
検出したベース音と各音階音のレベルから各小節のコード名を決定するコード名決定手段と
して機能させることを特徴とするコード名検出用プログラムである。

【００３９】
以上のようなプログラムの構成であれば、既存のハードウェア資源を用いてこのプログラムを使用することにより、既存のハードウェアで新たなアプリケーションとしての本発明の夫々の装置が容易に実現できるようになる。

【００４０】
このプログラムという態様では、通信などを利用して、これを容易に使用、配布、販売することができるようになる。また、既存のハードウェア資源を用いてこのプログラムを使用することにより、既存のハードウェアで新たなアプリケーションとしての本発明の装置が容易に実行できるようになる。

【００４１】
尚、請求項８記載の各機能実現手段のうち一部の機能は、コンピュータに組み込まれた機能（コンピュータにハードウェア的に組み込まれている機能でも良く、該コンピュータに組み込まれているオペレーティングシステムや他のアプリケーションプログラムなどによって実現される機能でも良い）によって実現され、前記プログラムには、該コンピュータによって達成される機能を呼び出すあるいはリンクさせる命令が含まれていても良い。

【００４２】
これは、請求項６に規定された各機能実現手段の一部が、例えばオペレーティングシステムなどによって達成される機能の一部で代行され、その機能を実現するためのプログラムないしモジュールなどは直接存在するわけではないが、それらの機能を達成するオペレーティングシステムの機能の一部を、呼び出したりリンクさせるようにしてあれば、実質的に同じ構成となるからである。

【発明の効果】
【００４３】

本発明の請求項１～請求項５記載のテンポ検出装置によれば、人間が演奏したテンポの揺らぐ演奏の音響信号から、曲全体の平均的なテンポと正確なビート（拍）の位置、さらに曲の拍子と１拍目の位置を検出することができるようになるという優れた効果を奏し得る。

【００４４】

また請求項６及び請求項７記載のコード名検出装置、請求項８記載のプログラムによれば、特別な音楽的知識を有する専門家でなくても、音楽ＣＤ等の複数の楽器音の混ざった入力された音楽音響信号（オーディオ信号）に対し、個々の音符情報を検出することなしに全体の響きから、コード名（和音名）を検出することが可能となる。

【００４５】

さらに、該構成によれば、構成音が同じ和音でも判別可能で、演奏のテンポが揺らいでしまった場合や、逆にわざとテンポを揺らして演奏しているような音源に関しても、小節毎の和音が検出可能となる。

【００４６】

特に請求項６及び請求項７記載のコード名検出装置の構成及び請求項８記載のプログラムでは、簡単な構成のみでビート検出という時間分解能が必要な処理（上記テンポ検出装置の構成と同じ）と、和音検出という周波数分解能が必要な処理（上記テンポ検出装置の構成を基にさらに和音を検出できる構成）を同時に行うことができるようになる。

【発明を実施するための最良の形態】

【００４７】

以下、本発明の実施の形態を図示例と共に説明する。

【実施例１】

【００４８】

図１は、本発明に係るテンポ検出装置の全体ブロック図である。同図によれば、本テンポ検出装置の構成は、音響信号を入力する入力部１と、入力された音響信号から、所定の時間間隔で、ＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求める音階音レベル検出部２と、この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出部３と、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出部４とを有している。

【００４９】

音楽音響信号を入力する上記入力部１は、テンポ検出をする対象の音楽音響信号を入力する部分である。マイク等の機器から入力されたアナログ信号をＡ／Ｄ変換器（図示無し）によりディジタル信号に変換しても良いし、音楽ＣＤなどのディジタル化された音楽データの場合は、そのままファイルとして取り込み（リッピング）、これを指定して開くようにしても良い。このようにして入力したディジタル信号がステレオの場合、後の処理を簡略化するためにモノラルに変換する。

【００５０】

このディジタル信号は、音階音レベル検出部２に入力される。この音階音レベル検出部は図２の各部から構成される。

【００５１】

そのうち波形前処理部２０は、音楽音響信号の上記入力部１からの音響信号を今後の処理に適したサンプリング周波数にダウンサンプリングする構成である。

【００５２】

ダウンサンプリングレートは、ビート検出に使う楽器の音域によって決定する。すなわち、シンバル、ハイハット等の高音域のリズム楽器の演奏音をビート検出に反映させるには、ダウンサンプリング後のサンプリング周波数を高い周波数にする必要があるが、ベー

ス音とバスドラム、スネアドラム等の楽器音と中音域の楽器音から主にビート検出させる場合には、ダウンサンプリング後のサンプリング周波数はそれほど高くする必要はない。

【００５３】

例えば検出する最高音をＡ６（Ｃ４が中央のド）とする場合、Ａ６の基本周波数は約１７６０Ｈｚ（Ａ４＝４４０Ｈｚとした場合）となるので、ダウンサンプリング後のサンプリング周波数は、ナイキスト周波数が１７６０Ｈｚ以上となる、３５２０Ｈｚ以上にすれば良い。これから、ダウンサンプリングレートは、元のサンプリング周波数が４４．１ｋＨｚ（音楽ＣＤ）の場合、１／１２程度にすれば良いことになる。この時、ダウンサンプリング後のサンプリング周波数は、３６７５Ｈｚとなる。

【００５４】

ダウンサンプリングの処理は、通常、ダウンサンプリング後のサンプリング周波数の半分の周波数であるナイキスト周波数（今の例では１８３７．５Ｈｚ）以上の成分をカットするローパスフィルタを通した後に、データを読み飛ばす（今の例では波形サンプルの１２個に１１個を破棄する）ことによって行われる。

【００５５】

このようにダウンサンプリングの処理を行うのは、この後のＦＦＴ演算において、同じ周波数分解能を得るために必要なＦＦＴポイント数を下げることで、ＦＦＴの演算時間を減らすのが目的である。

【００５６】

なお、音楽ＣＤのように、音源が固定のサンプリング周波数で既にサンプリングされている場合は、このようなダウンサンプリングが必要になるが、音楽音響信号の入力部１が、マイク等の機器から入力されたアナログ信号をＡ／Ｄ変換器によりディジタル信号に変換するような場合には、当然Ａ／Ｄ変換器のサンプリング周波数を、ダウンサンプリング後のサンプリング周波数に設定することで、この波形前処理部を省くことが可能である。

【００５７】

このようにして波形前処理部２０によるダウンサンプリングが終了したら、所定の時間間隔で、波形前処理部の出力信号を、ＦＦＴ演算部２１によりＦＦＴ（高速フーリエ変換）する。

【００５８】

ＦＦＴのパラメータ（ＦＦＴポイント数とＦＦＴ窓のシフト量）は、ビート検出に適した値とする。つまり、周波数分解能を上げるためにＦＦＴポイント数を大きくすると、ＦＦＴ窓のサイズが大きくなってしまい、より長い時間から１回のＦＦＴを行うことになり、時間分解能が低下する、というＦＦＴの特性を考慮しなくてはならない（つまりビート検出時は周波数分解能を犠牲にして時間分解能をあげるのが良い）。窓のサイズと同じだけの長さの波形を使わないで、窓の一部だけに波形データをセットし残りは０で埋めることによって、ＦＦＴポイント数を大きくしても時間分解能が悪くならない方法もあるが、低音側のパワーも正しく検出するためには、ある程度の波形サンプル数は必要である。

【００５９】

以上のようなことを考慮し、本実施例では、ＦＦＴポイント数５１２、窓のシフトは３２サンプルで、０埋めなしという設定にした。このような設定でＦＦＴ演算を行うと、時間分解能約８．７ｍｓ、周波数分解能約７．２Ｈｚとなる。時間分解能約８．７ｍｓという値は、四分音符＝３００のテンポの曲で、３２分音符の長さが、２５ｍｓであることを考えると、十分な値であることがわかる。

【００６０】

このようにして、所定の時間間隔毎にＦＦＴ演算が行われ、その実数部と虚数部のそれぞれを二乗したものの和の平方根からパワーが計算され、その結果がレベル検出部２２に送られる。

【００６１】

レベル検出部２２では、ＦＦＴ演算部２１で計算されたパワー・スペクトルから、各音階音のレベルを計算する。ＦＦＴは、サンプリング周波数をＦＦＴポイント数で割った値

の整数倍の周波数のパワーが計算されるだけであるので、このパワー・スペクトルから各音階音のレベルを検出するために、以下のような処理を行う。つまり、音階音を計算するすべての音（Ｃ１からＡ６）について、その各音の基本周波数の上下５０セントの範囲（１００セントが半音）の周波数に相当するパワー・スペクトルの内、最大のパワーを持つスペクトルのパワーをこの音階音のレベルとする。

【００６２】

すべての音階音についてレベルが検出されたら、これをバッファに保存し、波形の読み出し位置を所定の時間間隔（先の例では３２サンプル）進めて、ＦＦＴ演算部２１とレベル検出部２２を波形の終わりまで繰り返す。

【００６３】

以上により、音楽音響信号の入力部１に入力された音響信号の、所定時間毎の各音階音のレベルが、バッファ２３に保存される。

【００６４】

次に、図１のビート検出部３の構成について説明する。該ビート検出部３は、図３のような処理の流れで実行される。

【００６５】

ビート検出部３は、音階音レベル検出部が出力した所定時間（以下、この１所定時間を１フレームと呼ぶ）毎の各音階音のレベルの変化を元に平均的なビート（拍）間隔（つまりテンポ）とビートの位置を検出する。そのために、まずビート検出部３は、各音階音のレベル増分値の合計（前のフレームとのレベルの増分値をすべての音階音で合計したもの。前のフレームからレベルが減少している場合は０として加算する）を計算する（ステップＳ１００）。

【００６６】

つまり、フレーム時間ｔにおけるｉ番目の音階音のレベルを$L_i(t)$とするとき、ｉ番目の音階音のレベル増分値$L_{addi}(t)$は、下式数１に示すようになり、この$L_{addi}(t)$を使って、フレーム時間ｔにおける各音階音のレベル増分値の合計Ｌ（ｔ）は、下式数２で計算できる。ここで、Ｔは音階音の総数である。

【００６７】

【数１】

$$L_{addi}(t)=\begin{cases} L_i(t)-L_i(t-1) & (L_i(t-1) \leqq L_i(t)\text{のとき}) \\ 0 & (L_i(t-1) > L_i(t)\text{のとき}) \end{cases}$$

【００６８】

【数２】

$$L(t)=\sum_{i=0}^{T-1} L_{addi}(t)$$

【００６９】

この合計Ｌ（ｔ）値は、フレーム毎の全体での音の変化度合いを表している。この値は、音の鳴り始めで急激に大きくなり、同時に鳴り始める音が多いほど大きな値となる。音楽はビートの位置で音が鳴り始めることが多いので、この値が大きなところはビートの位置である可能性が高いことになる。

【００７０】

例として、図４に、ある曲の一部分の波形と各音階音のレベル、各音階音のレベル増分

値の合計の図を示す。上段が波形、中央がフレーム毎の各音階音のレベルを濃淡で表したもの（下が低い音、上が高い音。この図では、Ｃ１からＡ６の範囲）、下段がフレーム毎の各音階音のレベル増分値の合計を示している。この図の各音階音のレベルは、音階音レベル検出部から出力されたものであるので、周波数分解能が約７．２Ｈｚであり、Ｇ＃２以下の一部の音階音でレベルが計算できずに歯抜け状態になっているが、この場合はビートを検出するのが目的であるので、低音の一部の音階音のレベルが測定できないのは、問題ない。

【００７１】

この図の下段に見られるように、各音階音のレベル増分値の合計は、定期的にピークをもつ形となっている。この定期的なピークの位置が、ビートの位置である。

【００７２】

ビートの位置を求めるために、ビート検出部３では、まずこの定期的なピークの間隔、つまり平均的なビート間隔を求める。平均的なビート間隔はこの各音階音のレベル増分値の合計の自己相関から計算できる（図３；ステップＳ１０２）。

【００７３】

あるフレーム時間ｔにおける各音階音のレベル増分値の合計をＬ（ｔ）とすると、この自己相関φ（τ）は、以下の式数３で計算される。

【００７４】

【数３】

$$\phi(\tau) = \frac{\sum_{t=0}^{N-\tau-1} L(t) \cdot L(t+\tau)}{N-\tau}$$

ここで、Ｎは総フレーム数、τは時間遅れである。

【００７５】

自己相関計算の概念図を、図５に示す。この図のように、時間遅れτがＬ（ｔ）のピークの周期の整数倍の時に、φ（τ）は大きな値となる。よって、ある範囲のτについてφ（τ）の最大値を求めれば、曲のテンポを求めることができる。

【００７６】

自己相関を求めるτの範囲は、想定する曲のテンポ範囲によって変えれば良い。例えば、メトロノーム記号で四分音符＝３０から３００の範囲を計算するならば、自己相関を計算する範囲は、０．２秒から２秒となる。時間（秒）からフレームへの変換式は、以下の数４式に示す通りとなる。

【００７７】

【数４】

$$\text{フレーム数} = \frac{\text{時間(秒)} \cdot \text{サンプリング周波数}}{\text{１フレームあたりのサンプル数}}$$

【００７８】

この範囲の自己相関φ（τ）が最大となるτをビート間隔としても良いが、必ずしもすべての曲で自己相関が最大となる時のτがビート間隔とはならないので、自己相関が極大値となる時のτからビート間隔の候補を求め（図３；ステップＳ１０４）、これら複数の候補からユーザにビート間隔を決定させるのが良い（図３；ステップＳ１０６）。

【００７９】

このようにしてビート間隔が決定したら（決定したビート間隔をτ_{max}とする）、まず最初に先頭のビート位置を決定する。

【００８０】

先頭のビート位置の決定方法を、図６を用いて説明する。図６の上段はフレーム時間ｔ

における各音階音のレベル増分値の合計Ｌ（ｔ）で、下段Ｍ（ｔ）は決定したビート間隔τ_{max}の周期で値を持つ関数である。式で表すと、下式数５に示すようになる。

【００８１】

【数５】

$$M(t)=\begin{cases}1 & (t\text{が}\tau_{max}\text{の整数倍のとき})\\ 0 & (\text{上記以外のとき})\end{cases}$$

【００８２】

この関数Ｍ（ｔ）を、０から$\tau_{max}-1$の範囲でずらしながら、Ｌ（ｔ）とＭ（ｔ）の相互相関を計算する。

【００８３】

相互相関ｒ（ｓ）は、上記Ｍ（ｔ）の特性から、下式数６で計算できる。

【００８４】

【数６】

$$r(s)=\sum_{j=0}^{n-1}L(\tau_{max}\cdot j+s)\qquad(0\leqq s<\tau_{max})$$

【００８５】

この場合のｎは、最初の無音部分の長さに応じて適当に決めれば良い（図６の例では、ｎ＝１０）。

【００８６】

ｒ（ｓ）をｓが０から$\tau_{max}-1$の範囲で求め、ｒ（ｓ）が最大となるｓを求めれば、このｓのフレームが最初のビート位置である。

【００８７】

最初のビート位置が決まったら、それ以降のビートの位置を１つずつ決定していく（図３；ステップＳ１０８）。

【００８８】

その方法を、図７を用いて説明する。図７の三角印の位置に先頭のビートが見つかったとする。２番目のビート位置は、この先頭のビート位置からビート間隔τ_{max}だけ離れた位置を仮のビート位置とし、その近辺でＬ（ｔ）とＭ（ｔ）が最も相関が取れる位置から決定する。つまり、先頭のビート位置をb_0とするとき、以下の式のｒ（ｓ）が最大となるようなｓの値を求める。この式のｓは仮のビート位置からのずれで、以下の式数７の範囲の整数とする。Ｆは揺らぎのパラメータで０．１程度の値が適当であるが、テンポの揺らぎの大きい曲では、もっと大きな値にしてもよい。ｎは５程度でよい。

【００８９】

ｋは、ｓの値に応じて変える係数で、例えば図８のような正規分布とする。

【００９０】

【数７】

$$r(s)=\sum_{j=1}^{n}k\cdot L(b_0+\tau_{max}\cdot j+s)\qquad(-\tau_{max}\cdot F\leqq s\leqq\tau_{max}\cdot F)$$

【００９１】

ｒ（ｓ）が最大となるようなｓの値が求まれば、２番目のビート位置b_1は、下式数８で計算される。

【００９２】

【数８】

$$b_1=b_0+\tau_{max}+s$$

【００９３】
以降、同じようにして３番目以降のビート位置も求めることができる。
【００９４】
テンポがほとんど変わらない曲ではこの方法でビート位置を曲の終わりまで求めることができるが、実際の演奏は多少テンポが揺らいだり、部分的にだんだん遅くなったりすることがよくある。
【００９５】
そこで、これらのテンポの揺らぎにも対応できるように以下のような方法を考えた。
【００９６】
つまり、図７のＭ（ｔ）の関数を、図９のように変化させるものである。
１）は、従来の方法で、図のように各パルスの間隔をτ１、τ２、τ３、τ４としたとき、
$\tau 1 = \tau 2 = \tau 3 = \tau 4 = \tau_{max}$
である。
２）は、τ１からτ４を均等に大きくしたり小さくしたりするものである。
$\tau 1 = \tau 2 = \tau 3 = \tau 4 = \tau_{max} + s$ $(-\tau_{max} \cdot F \leqq s \leqq \tau_{max} \cdot F)$ これにより、急にテンポが変わった場合に対応できる。
３）は、ｒｉｔ．（リタルダンド、だんだん遅く）又は、ａｃｃｅｌ．（アッチェレランド、だんだん速く）に対応したもので、各パルス間隔は、
$\tau 1 = \tau_{max}$
$\tau 2 = \tau_{max} + 1 \cdot s$
$\tau 3 = \tau_{max} + 2 \cdot s$ $(-\tau_{max} \cdot F \leqq s \leqq \tau_{max} \cdot F)$
$\tau 4 = \tau_{max} + 4 \cdot s$
で計算される。
１、２、４の係数は、あくまで例であり、テンポ変化の大きさによって変えてもよい。
４）は、３）のようなｒｉｔ．やａｃｃｅｌ．の場合の、５個のパルスの位置のどこが現在ビートを求めようとしている場所かを変えるものである。
【００９７】
これらをすべて組み合わせて、Ｌ（ｔ）とＭ（ｔ）の相関を計算し、それらの最大からビート位置を決めれば、テンポが揺らぐ曲に対してもビート位置の決定が可能である。なお、２）と３）の場合には、相関を計算するときの係数ｋの値を、やはりｓの値に応じて変えるようにする。
【００９８】
さらに、５個のパルスの大きさは現在すべて同じにしてあるが、ビートを求める位置（図９の仮のビート位置）のパルスのみ大きくしたり、ビートを求める位置から離れるほど値を小さくして、ビートを求める位置の各音階音のレベル増分値の合計を強調するようにしてもよい［図９の５）］。
【００９９】
以上のようにして、各ビートの位置が決定したら、この結果をバッファ３０に保存すると共に、検出した結果を表示し、ユーザに確認してもらい、間違っている箇所を修正してもらうようにしてもよい。
【０１００】
ビート検出結果の確認画面の例を、図１０に示す。同図の三角印の位置が検出したビート位置である。
【０１０１】
「再生」のボタンを押すと、現在の音楽音響信号が、Ｄ／Ａ変換され、スピーカ等から再生される。現在の再生位置は、図のように縦線等の再生位置ポインタで表示されるので、演奏を聞きながら、ビート検出位置の誤りを確認できる。さらに、検出の元波形の再生と同時に、ビート位置のタイミングで例えばメトロノームのような音を再生させるようにすれば、目で確認するだけでなく音でも確認でき、より容易に誤検出を判断できる。この

メトロノームの音を再生させる方法としては、例えばＭＩＤＩ機器等が考えられる。

【０１０２】

ビート検出位置の修正は、「ビート位置の修正」ボタンを押して行う。このボタンを押すと、画面に十字のカーソルが現れるので、最初のビート検出が間違っている箇所で正しいビート位置をクリックする。クリックされた場所の少し前（例えばτ_{max}の半分の位置）から後のビート位置をすべてクリアし、クリックされた場所を、仮のビート位置として、以降のビート位置を再検出する。

【０１０３】

次に、拍子および小節の検出について説明する。

【０１０４】

これまでの処理で、ビートの位置が確定しているので、今度は、ビート毎の音の変化度合いを求める。ビート毎の音の変化度合いは、音階音レベル検出部が出力した、フレーム毎の各音階音のレベルから計算する。

【０１０５】

ｊ番目のビートのフレーム数をb_jとし、その前後のビートのフレームをb_{j-1}、b_{j+1}とする時、ｊ番目のビートのビート毎の音の変化度合いは、フレームb_{j-1}からb_j-1までのフレームの各音階音のレベルの平均とフレームb_jから$b_{j+1}-1$までのフレームの各音階音のレベルの平均を計算し、その増分値から各音階音のビート毎の音の変化度合いを求め、それらをすべての音階音で合計して計算することができる。

【０１０６】

つまり、フレーム時間ｔにおけるｉ番目の音階音のレベルをL_i（ｔ）とするとき、ｊ番目のビートのｉ番目の音階音のレベルの平均L_{avgi}（ｊ）は、下式数９であるから、ｊ番目のビートのｉ番目の音階音のビート毎の音の変化度合いB_{addi}（ｊ）は、下式数１０に示すようになる。

【０１０７】

【数９】

$$L_{avgi}(j)=\frac{\sum_{t=b_j}^{b_{j+1}-1} L_i(t)}{b_{j+1}-b_j}$$

【０１０８】

【数１０】

$$B_{addi}(j)=\begin{cases} L_{avgi}(j)-L_{avgi}(j-1) & (L_{avgi}(j-1) \leqq L_{avgi}(j)\text{のとき}) \\ 0 & (L_{avgi}(j-1) > L_{avgi}(j)\text{のとき}) \end{cases}$$

【０１０９】

よって、ｊ番目のビートのビート毎の音の変化度合いＢ（ｊ）は、下式数１１に示すようになる。ここで、Ｔは音階音の総数である。

【０１１０】

【数１１】

$$B(j)=\sum_{i=0}^{T-1} B_{addi}(j)$$

【０１１１】

図１１の最下段は、このビート毎の音の変化度合いである。このビート毎の音の変化度

合いから拍子と１拍目の位置を求める。

【０１１２】

拍子は、ビート毎の音の変化度合いの自己相関から求める。一般的に音楽は１拍目で音が変わることが多いと考えられるので、このビート毎の音の変化度合いの自己相関から拍子を求めることができる。例えば、下式数１２に示す自己相関φ（τ）を求める式から、ビート毎の音の変化度合いＢ（ｊ）の自己相関φ（τ）を遅れτが、２から４の範囲で求め、自己相関φ（τ）が最大となる遅れτを拍子の数とする。

【０１１３】

【数１２】

$$\phi(\tau) = \frac{\sum_{j=0}^{N-\tau-1} B(j) \cdot B(j+\tau)}{N-\tau}$$

【０１１４】

Ｎは、総ビート数、τ＝２～４の範囲でφ（τ）を計算し、φ（τ）が最大となるτを拍子の数とする。

【０１１５】

次に１拍目を求めるが、これは、ビート毎の音の変化度合いＢ（ｊ）がもっとも大きい箇所を１拍目とする。つまり、φ（τ）が最大となるτをτ_{max}、下式数１３のＸ（ｋ）が最大となるｋをk_{max}とするとき、k_{max}番目のビートが最初の１拍目の位置となり、以降、τ_{max}を足したビート位置が１拍目となる。

【０１１６】

【数１３】

$$X(k) = \frac{\sum_{n=0}^{n_{max}} B(\tau_{max} \cdot n + k)}{n_{max} + 1} \qquad (0 \leqq k < \tau_{max})$$

n_{max}は、$\tau_{max} \cdot n + k < N$の条件で最大となるｎ

【０１１７】

以上のようにして、拍子及び１拍目の位置（小節線の位置）が決定したら、この結果をバッファ４０に保存すると共に、検出した結果を画面表示して、ユーザに変更させるようにすることが望ましい。特に変拍子の曲は、この方法では対応できないので、変拍子の箇所をユーザに指定してもらう必要がある。

【０１１８】

以上の実施例構成により、人間が演奏したテンポの揺らぐ演奏の音響信号から、曲全体の平均的なテンポと正確なビート（拍）の位置、さらに曲の拍子と１拍目の位置を検出することが可能となる。

【実施例２】

【０１１９】

図１２は、本発明のコード名検出装置の全体ブロック図である。同図において、ビート検出及び小節検出の構成は、実施例１と基本的に同じであり、同一構成において、テンポ検出用とコード検出用の構成について、上記実施例１の場合と異なるものもあるので、数式等を除き、同じ説明が重なるが、以下に示す。

【０１２０】

同図によれば、本コード検出装置の構成は、音響信号を入力する入力部１と、入力された音響信号から、所定の時間間隔で、ビート検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求めるビート検出用音階音レベル検出部２と、この所定の時間毎の各音階音のレベルの増分値をすべての音階音について合計して、所

定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計を求め、この所定の時間毎の全体の音の変化度合いを示すレベルの増分値の合計から、平均的なビート間隔と各ビートの位置を検出するビート検出部３と、このビート毎の各音階音のレベルの平均値を計算し、このビート毎の各音階音の平均レベルの増分値をすべての音階音について合計して、ビート毎の全体の音の変化度合いを示す値を求め、このビート毎の全体の音の変化度合いを示す値から、拍子と小節線位置を検出する小節検出部４と、上記入力された音響信号から、先のビート検出の時とは異なる別の所定の時間間隔で、コード検出に適したパラメータを使ってＦＦＴ演算を行い、所定の時間毎の各音階音のレベルを求めるコード検出用音階音レベル検出部５と、検出した各音階音のレベルのうち、各小節内における低域側の音階音のレベルからベース音を検出するベース音検出部６と、検出したベース音と各音階音のレベルから各小節のコード名を決定するコード名決定部７とを有している。

【０１２１】

音楽音響信号を入力する上記入力部１は、コード検出をする対象の音楽音響信号を入力する部分であるが、基本的構成は上記実施例１の入力部１と同じであるので、その詳細な説明は省略する。ただし、通常センタに定位されるボーカルが後のコード検出でじゃまになる場合は、右チャンネルの波形と左チャンネルの波形を引き算することでボーカルキャンセルするようにしても良い。

【０１２２】

このディジタル信号は、ビート検出用音階音レベル検出部２とコード検出用音階音レベル検出部５とに入力される。これらの音階音レベル検出部は、どちらも上記図２の各部から構成され、構成はまったく同じなので、同じものをパラメータだけを変えて再利用できる。

【０１２３】

そしてその構成として使用される波形前処理部２０は、上記と同様な構成であり、音楽音響信号の上記入力部１からの音響信号を今後の処理に適したサンプリング周波数にダウンサンプリングする。ただし、ダウンサンプリング後のサンプリング周波数、つまり、ダウンサンプリングレートは、ビート検出用とコード検出用で変えるようにしても良いし、ダウンサンプリングする時間を節約するために同じにしても良い。

【０１２４】

ビート検出用の場合は、ビート検出に使う音域によってダウンサンプリングレートを決定する。シンバル、ハイハット等の高音域のリズム楽器の演奏音をビート検出に反映させるには、ダウンサンプリング後のサンプリング周波数を高い周波数にする必要があるが、ベース音とバスドラム、スネアドラム等の楽器音と中音域の楽器音から主にビート検出させる場合には、以下のコード検出時と同じダウンサンプリングレートで構わない。

【０１２５】

コード検出用の波形前処理部のダウンサンプリングレートは、コード検出音域によって変える。コード検出音域とは、コード名決定部でコード検出するときに使う音域のことである。例えばコード検出音域をＣ３からＡ６（Ｃ４が中央のド）とする場合、Ａ６の基本周波数は約１７６０Ｈｚ（Ａ４＝４４０Ｈｚとした場合）となるので、ダウンサンプリング後のサンプリング周波数はナイキスト周波数が１７６０Ｈｚ以上となる、３５２０Ｈｚ以上にすれば良い。これから、ダウンサンプリングレートは、元のサンプリング周波数が４４．１ｋＨｚ（音楽ＣＤ）の場合、１／１２程度にすれば良いことになる。この時、ダウンサンプリング後のサンプリング周波数は、３６７５Ｈｚとなる。

【０１２６】

ダウンサンプリングの処理は、通常、ダウンサンプリング後のサンプリング周波数の半分の周波数であるナイキスト周波数（今の例では１８３７．５Ｈｚ）以上の成分をカットするローパスフィルタを通した後に、データを読み飛ばす（今の例では波形サンプルの１２個に１１個を破棄する）ことによって行われる。これについては、実施例１に説明したことと同じ理由による。

【０１２７】

このようにして波形前処理部２０によるダウンサンプリングが終了したら、所定の時間間隔で、波形前処理部の出力信号をＦＦＴ演算部２１により、ＦＦＴ（高速フーリエ変換）する。

【０１２８】

ＦＦＴのパラメータ（ＦＦＴポイント数とＦＦＴ窓のシフト量）は、ビート検出時とコード検出時で異なる値とする。これは、周波数分解能を上げるためにＦＦＴポイント数を大きくすると、ＦＦＴ窓のサイズが大きくなってしまい、より長い時間から１回のＦＦＴを行うことになり、時間分解能が低下する、というＦＦＴの特性によるものである（つまりビート検出時は周波数分解能を犠牲にして時間分解能をあげるのが良い）。窓のサイズと同じだけの長さの波形を使わないで、窓の一部だけに波形データをセットし、残りは０で埋めることによってＦＦＴポイント数を大きくしても時間分解能が悪くならない方法もあるが、本実施例のケースでは、低音側のパワーも正しく検出するためにある程度の波形サンプル数は必要である。

【０１２９】

以上のようなことを考慮し、本実施例では、ビート検出時はＦＦＴポイント数５１２、窓のシフトは３２サンプルで、０埋めなし、コード検出時はＦＦＴポイント数８１９２、窓のシフトは１２８サンプルで、波形サンプルは一度のＦＦＴで１０２４サンプル使うようにした。このような設定でＦＦＴ演算を行うと、ビート検出時は、時間分解能約８．７ｍｓ、周波数分解能約７．２Ｈｚ、コード検出時は、時間分解能約３５ｍｓ、周波数分解能約０．４Ｈｚとなる。今レベルを求めようとしている音階音は、Ｃ１からＡ６の範囲であるので、コード検出時の周波数分解能約０．４Ｈｚは、最も周波数差の小さいＣ１とＣ＃１の基本周波数の差、約１．９Ｈｚにも対応できる。また、四分音符＝３００のテンポの曲で３２分音符の長さが２５ｍｓであることを考えると、ビート検出時の時間分解能約８．７ｍｓは、十分な値であることがわかる。

【０１３０】

このようにして、所定の時間間隔毎にＦＦＴ演算が行われ、その実数部と虚数部のそれぞれを二乗したものの和の平方根からパワーが計算され、その結果がレベル検出部２２に送られる。

【０１３１】

レベル検出部２２では、ＦＦＴ演算部２１で計算されたパワー・スペクトルから、各音階音のレベルを計算する。ＦＦＴは、サンプリング周波数をＦＦＴポイント数で割った値の整数倍の周波数のパワーが計算されるだけであるので、このパワー・スペクトルから各音階音のレベルを検出するために、実施例１と同様な処理を行う。すなわち、音階音を計算するすべての音（Ｃ１からＡ６）について、その各音の基本周波数の上下５０セントの範囲（１００セントが半音）の周波数に相当するパワー・スペクトルの内、最大のパワーを持つスペクトルのパワーをこの音階音のレベルとする。

【０１３２】

すべての音階音についてレベルが検出されたら、これをバッファに保存し、波形の読み出し位置を所定の時間間隔（先の例ではビート検出時は３２サンプル、コード検出時は１２８サンプル）進めて、ＦＦＴ演算部２１とレベル検出部２２を波形の終わりまで繰り返す。

【０１３３】

以上により、音楽音響信号の入力部１に入力された音響信号の、所定時間毎の各音階音のレベルが、ビート検出用とコード検出用の２種類のバッファ２３及び５０に保存される。

【０１３４】

次に、図１２のビート検出部３及び小節検出部４の構成については、実施例１のビート検出部３及び小節検出部４と同じ構成なので、その詳細な説明は、ここでは、省略する。

【０１３５】

実施例１と同様な構成と手順で、小節線の位置（各小節のフレーム番号）が確定したの

で、今度は各小節のベース音を検出する。

【０１３６】

ベース音は、コード検出用音階音レベル検出部５が出力した各フレームの音階音のレベルから検出する。

【０１３７】

図１３に実施例１の図４と同じ曲の同じ部分のコード検出用音階音レベル検出部５が出力した各フレームの音階音のレベルを示す。この図のように、コード検出用音階音レベル検出部５での周波数分解能は、約０．４Ｈｚであるので、Ｃ１からＡ６のすべての音階音のレベルが抽出されている。

【０１３８】

ベース音は、小節の前半と後半で異なる可能性があるので、ベース音検出部６により、各小節の前半と後半でそれぞれ検出する。前半と後半のベース音が同じ音のときは、小節のベース音としてこれを確定し、コードも小節全体で検出する。前半と後半で別の音のベース音が検出されたときは、コードも前半と後半に分けて検出する。場合によっては、分割する範囲を更に半分にまで（小節の４分の１まで）狭めてもよい。

【０１３９】

ベース音は、ベース検出期間におけるベース検出音域の音階音のレベルの平均的な強さから求める。

【０１４０】

フレーム時間ｔにおけるｉ番目の音階音のレベルを$L_i(t)$とすると、フレームf_sからf_eのｉ番目の音階音の平均的なレベル$L_{avgi}(f_s, f_e)$は、下式数１４で計算できる。

【０１４１】

【数１４】

$$L_{avgi}(f_s, f_e) = \frac{\sum_{t=f_s}^{f_e} L_i(t)}{f_e - f_s + 1} \qquad (f_s \leqq f_e)$$

【０１４２】

この平均的なレベルをベース検出音域、例えばＣ２からＢ３の範囲で計算し、平均的なレベルが最も大きな音階音をベース音として、ベース音検出部６は、決定する。ベース検出音域に音が含まれない曲や無音部分で間違ってベース音を検出しないために、適当な閾値を設定し、検出したベース音の平均的なレベルが、この閾値以下の場合は、ベース音を検出しないようにしてもよい。また、後のコード検出でベース音を重要視する場合には、検出したベース音がベース検出期間中継続してあるレベル以上を保っているかどうかをチェックするようにして、より確実なものだけをベース音として検出するようにしてもよい。さらに、ベース検出音域中、平均的なレベルが最も大きい音階音をベース音として決定するのではなく、この各音名の平均的なレベルを１２の音名毎に平均し、この音名毎のレベルが最も大きな音名をベース音名として決定し、その音名を持つベース検出音域の中の音階音で、平均的なレベルが最も大きい音階音をベース音として決定するようにしてもよい。

【０１４３】

ベース音が決定したら、この結果をバッファ６０に保存すると共に、ベース検出結果を画面表示して、間違っている場合にはユーザに修正させるようにしてもよい。また、曲によってベース音域が変わることも考えられるので、ユーザがベース検出音域を変更できるようにしてもよい。

【０１４４】

図１４に、ベース音検出部６によるベース検出結果の表示例を示す。

【０１４５】

次にコード名決定部７によるコード検出処理であるが、該コード検出処理も、同じよう

にコード検出期間における各音階音の平均的なレベルを計算することによって決定する。

【０１４６】

本実施例では、コード検出期間とベース検出期間は同一としている。コード検出音域、例えばＣ３からＡ６の各音階音のコード検出期間における平均的なレベルを計算し、これが大きな値を持つ音階音から順に数個の音名を検出し、これとベース音の音名からコード名候補を抽出する。

【０１４７】

この際、必ずしもレベルが大きな音がコード構成音であるとは限らないので、複数の音名の音を例えば５つ検出し、その中の２つ以上をすべての組み合わせで抜き出して、これとベース音の音名とからコード名候補の抽出を行う。

【０１４８】

コードに関しても、平均的なレベルが閾値以下のものは検出しないようにしてもよい。また、コード検出音域もユーザが変更できるようにしてもよい。さらに、コード検出音域中、平均的なレベルが最も大きい音階音から順にコード構成音候補を抽出するのではなく、このコード検出音域内の各音名の平均的なレベルを１２の音名毎に平均し、この音名毎のレベルの最も大きな音名から順にコード構成音候補を抽出してもよい。

【０１４９】

コード名候補の抽出は、コードのタイプ（ｍ、Ｍ７等）とコード構成音のルート音からの音程を保存したコード名データベースを、コード名決定部７により検索することによって抽出する。つまり、検出した５つの音名の中からすべての２つ以上の組み合わせを抜き出し、これらの音名間の音程が、このコード名データベースのコード構成音の音程の関係にあるかどうかをしらみつぶしに調べ、同じ音程関係にあれば、コード構成音のいずれかの音名からルート音を算出し、そのルート音の音名にコードタイプを付けて、コード名を決定する。このとき、コードのルート音（根音）や５度の音は、コードを演奏する楽器では省略されることがあるので、これらを含まなくてもコード名候補として抽出するようにする。ベース音を検出した場合には、このコード名候補のコード名にベース音の音名を加える。すなわち、コードのルート音とベース音が同じ音名であればそのままでよいし、異なる音名の場合は分数コードとする。

【０１５０】

上記方法では、抽出されるコード名候補が多すぎるという場合には、ベース音による限定を行ってもよい。つまり、ベース音が検出された場合には、コード名候補の中でそのルート音がベース音と同じ音名でないものは削除する。

【０１５１】

コード名候補が複数抽出された場合には、これらの中でどれか１つを決定するために、コード名決定部７により、尤度（もっともらしさ）の計算をする。

【０１５２】

尤度は、コード検出音域におけるすべてのコード構成音のレベルの強さの平均とベース検出音域におけるコードのルート音のレベルの強さから計算する。すなわち、抽出されたあるコード名候補のすべての構成音のコード検出期間における平均レベルの平均値をL_{avgc}、コードのルート音のベース検出期間における平均レベルをL_{avgr}とすると、下式数１５のように、この２つの平均により尤度を計算する。

【０１５３】

【数１５】

$$尤度 = \frac{L_{avgc} + L_{avgr}}{2}$$

【０１５４】

この際、コード検出音域やベース検出音域に同一音名の音が複数含まれる場合には、そ

れらのうち、平均レベルの強い方を使うようにする。あるいは、コード検出音域とベース検出音域のそれぞれで、各音階音の平均レベルを１２の音名毎に平均し、その音名毎の平均値を使うようにしてもよい。

【０１５５】

さらに、この尤度の計算に音楽的な知識を導入してもよい。例えば、各音階音のレベルを全フレームで平均し、それを１２の音名毎に平均して各音名の強さを計算し、その強さの分布から曲の調を検出する。そして、調のダイアトニックコードには尤度が大きくなるようにある定数を掛ける、あるいは、調のダイアトニックスケール上の音から外れた音を構成音に含むコードはその外れた音の数に応じて尤度が小さくなるようにする等が、考えられる。さらにコード進行のよくあるパターンをデータベースとして記憶しておき、それと比較することで、コード候補の中からよく使われる進行になるようなものは尤度が大きくなるようにある定数を掛けるようにしてもよい。

【０１５６】

最も尤度が大きいものをコード名として決定するが、コード名の候補を尤度とともに表示し、ユーザに選択させるようにしてもよい。

【０１５７】

いずれにしても、コード名決定部７により、コード名が決定したら、この結果をバッファ７０に保存すると共に、コード名が、画面出力されることになる。

【０１５８】

図１５に、コード名決定部７によるコード検出結果の表示例を示す。このように検出されたコード名を画面表示するだけでなく、ＭＩＤＩ機器等を使って、検出されたコードとベース音を再生するようにすることが望ましい。一般的には、コード名を見ただけで正しいかどうかは判断できないからである。

【０１５９】

以上説明した本実施例構成によれば、特別な音楽的知識を有する専門家でなくても、音楽ＣＤ等の複数の楽器音の混ざった入力された音楽音響信号に対し、個々の音符情報を検出することなしに全体の響きから、コード名を検出することができるようになる。

【０１６０】

さらに、該構成によれば、構成音が同じ和音でも判別可能で、演奏のテンポが揺らいでしまった場合や、逆にわざとテンポを揺らして演奏しているような音源に関しても、小節毎のコード名が検出可能となる。

【０１６１】

特に本実施例構成では、簡単な構成のみでビート検出という時間分解能が必要な処理（上記テンポ検出装置の構成と同じ）と、コード検出という周波数分解能が必要な処理（上記テンポ検出装置の構成を基にさらにコード名を検出できる構成）を同時に行うことができるようになる。

【０１６２】

尚、本発明のテンポ検出装置、コード名検出装置及びそれらを実現できるプログラムは、上述の図示例にのみ限定されるものではなく、本発明の要旨を逸脱しない範囲内において種々変更を加え得ることは勿論である。

【産業上の利用可能性】

【０１６３】

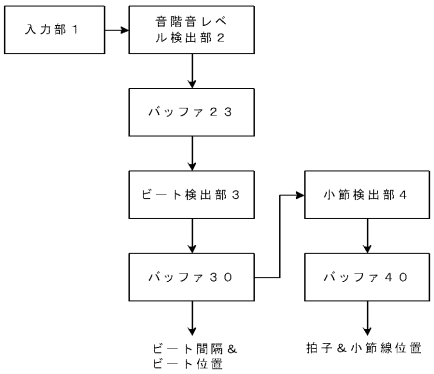
本発明のテンポ検出装置、コード名検出装置及びそれらを実現できるプログラムは、ミュージックプロモーションビデオの作成の際などに音楽トラック中のビートの時刻に対して映像トラック中のイベントを同期させるビデオ編集処理や、ビートトラッキングによりビートの位置を見つけ音楽の音響信号の波形を切り貼りするオーディオ編集処理、人間の演奏に同期して照明の色・明るさ・方向・特殊効果などといった要素を制御したり、観客の手拍子や歓声などを自動制御するライブステージのイベント制御、音楽に同期したコンピュータグラフィックスなど、種々の分野で利用可能である。

【図面の簡単な説明】

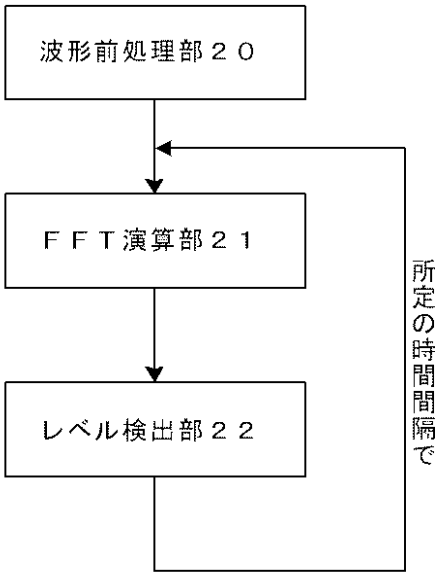
【０１６４】
【図１】本発明に係るテンポ検出装置の全体ブロック図である。
【図２】音階音レベル検出部２の構成のブロック図である。
【図３】ビート検出部３の処理の流れを示すフローチャートである。
【図４】ある曲の一部分の波形と各音階音のレベル、各音階音のレベル増分値の合計の図を示すグラフである。
【図５】自己相関計算の概念を示す説明図である。
【図６】先頭のビート位置の決定方法を説明する説明図である。
【図７】最初のビート位置決定後のそれ以降のビートの位置を決定していく方法を示す説明図である。
【図８】ｓの値に応じて変えられる係数ｋの分布状態を示すグラフである。
【図９】２番目以降のビート位置の決定方法を示す説明図である。
【図１０】ビート検出結果の確認画面の例を示す画面表示図である。
【図１１】小節検出結果の確認画面の例を示す画面表示図である。
【図１２】実施例２に係る本発明のコード検出装置の全体ブロック図である。
【図１３】曲の同じ部分のコード検出用音階音レベル検出部５が出力した各フレームの音階音のレベルを示すグラフである。
【図１４】ベース音検出部６によるベース検出結果の表示例を示すグラフである。
【図１５】コード検出結果の確認画面の例を示す画面表示図である。
【符号の説明】
【０１６５】

１	入力部
２	ビート検出用音階音レベル検出部
３	ビート検出部
４	小節検出部
５	コード検出用音階音レベル検出部
６	ベース音検出部
７	コード名決定部
２０	波形前処理部
２１	ＦＦＴ演算部
２２	レベル検出部
２３、３０、４０、５０、６０、７０	バッファ

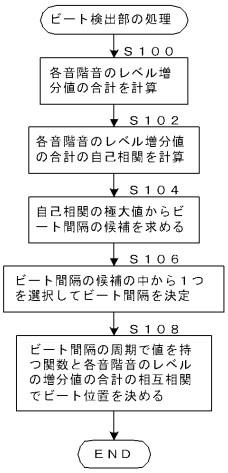
【図１】



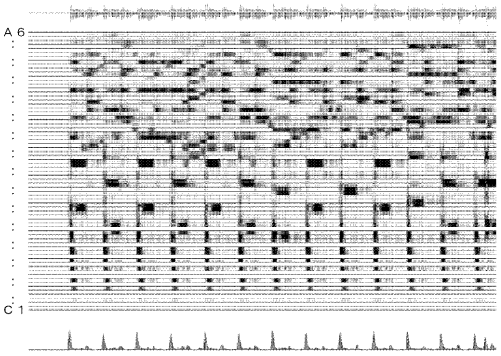
【図２】



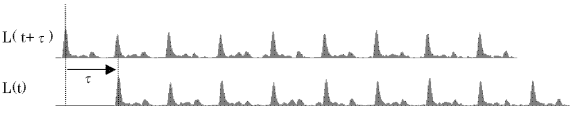
【図３】



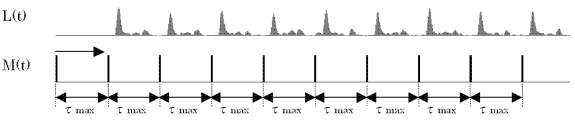
【図４】



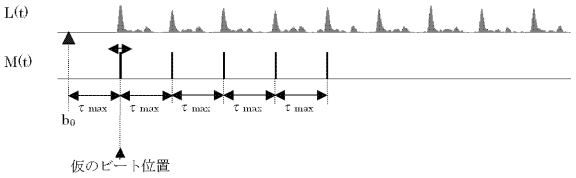
【図５】



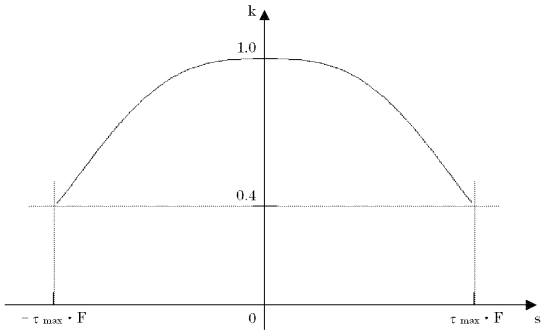
【図６】



【図７】



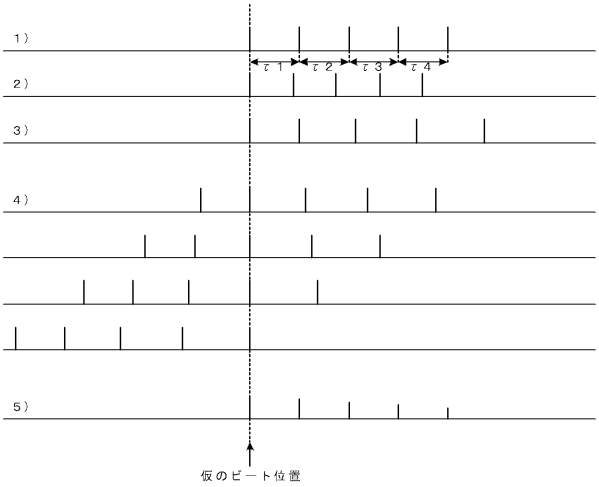
【図８】



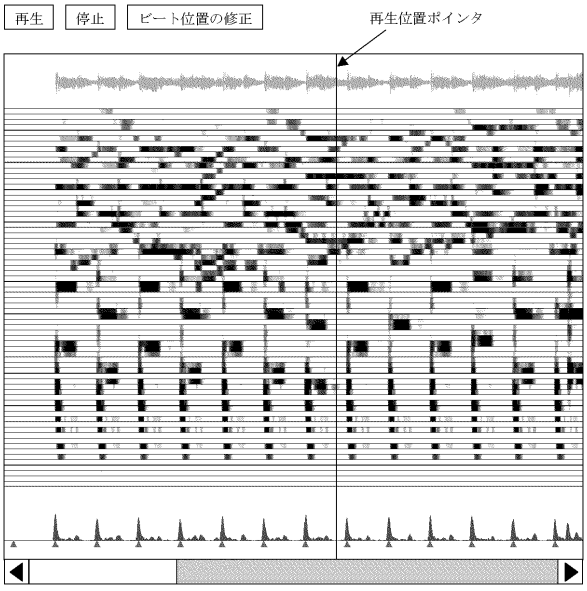
$$\sigma = \sqrt{\frac{-1}{2 \cdot \log_e(0.4)}}$$ とするとき、

$$k = e^{-\frac{\left|\frac{s}{\tau_{max} \cdot F}\right|^3}{2 \cdot \sigma^2}}$$

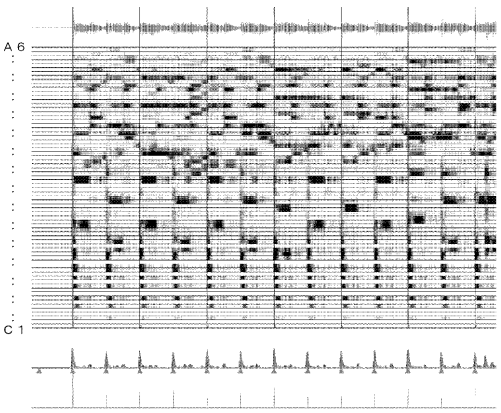
【図９】



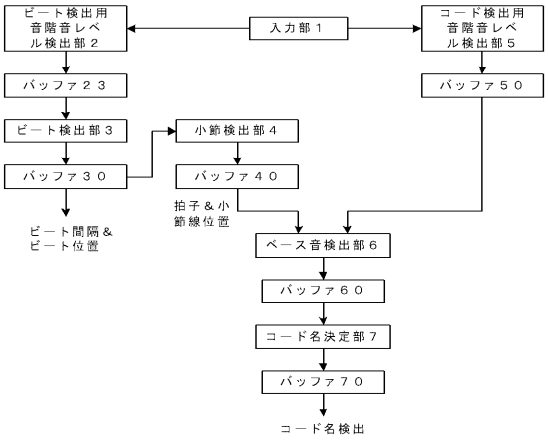
【図１０】



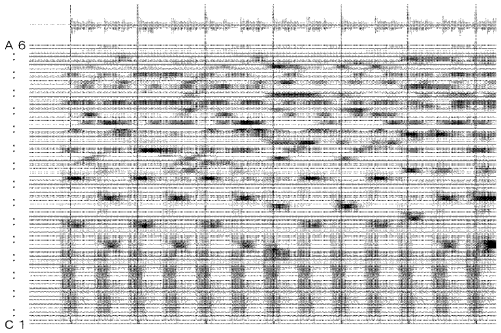
【図１１】



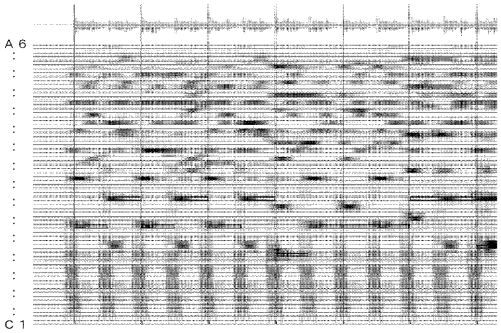
【図１２】



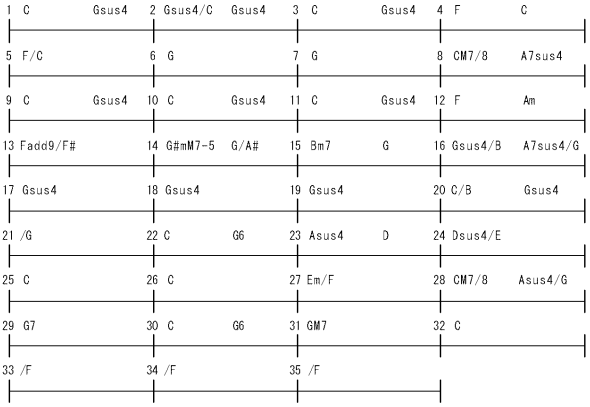
【図１３】



【図１４】



【図１５】



フロントページの続き

(56)参考文献 特開２００２－２１５１９５（ＪＰ，Ａ）
特開平０７－２９５５６０（ＪＰ，Ａ）
特開平０５－１００６６１（ＪＰ，Ａ）
特開２００２－１１６７５４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ１０Ｇ　　３／０４
Ｇ１０Ｈ　　１／００
Ｇ１０Ｈ　　１／３８
Ｇ１０Ｈ　　１／４０