(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2011-237602
(P2011-237602A)
(43)公開日　平成23年11月24日(2011.11.24)

(51)Int.Cl.		F I		テーマコード（参考）
G10G 3/04	(2006.01)	G10G 3/04		5D082
G10L 19/00	(2006.01)	G10L 19/00	220Z	
G10L 21/04	(2006.01)	G10L 21/04	120C	
G10L 13/02	(2006.01)	G10L 13/02	122B	

審査請求 未請求　請求項の数 22　ＯＬ　（全 44 頁）

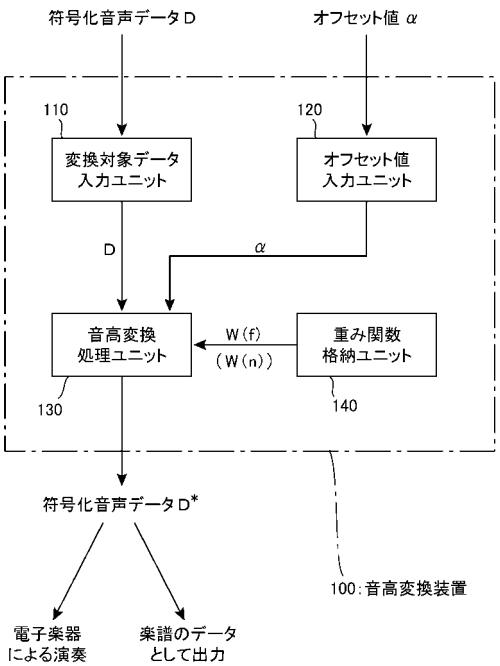
(21)出願番号　特願2010-109006 (P2010-109006)
(22)出願日　平成22年5月11日 (2010.5.11)

(71)出願人　000002897
大日本印刷株式会社
東京都新宿区市谷加賀町一丁目１番１号
(74)代理人　100091476
弁理士　志村　浩
(72)発明者　茂出木　敏雄
東京都新宿区市谷加賀町一丁目１番１号
大日本印刷株式会社内

Ｆターム(参考)　5D082 BB01　BB13　BB14　BB15

(54)【発明の名称】符号化音声データの音高変換装置

(57)【要約】
【課題】人間の声を表現した符号化音声データＤに対して、簡便な方法により音高シフトを施し、違和感のない自然な音声再生を可能にする。
【解決手段】ユニット１１０により変換対象となる符号化音声データＤを入力し、ユニット１２０により音高のシフト量を示すオフセット値αを入力する。ユニット１４０内には、周波数ｆの増加に従って単調減少する重み関数Ｗ（ｆ）を格納しておく。ユニット１３０は、データＤ内の各符号について、当該符号が示す周波数ｆを、α・Ｗ（ｆ）に応じた値だけ増減する処理を行い、処理後の符号を符号化音声データＤ＊として出力する。各符号に対する音高の実際のシフト量は、重み関数Ｗ（ｆ）を乗じて決定されるため、より高音の符号ほど音高シフト量が減少する。このため、フォルマント成分についての音高シフト量が小さくなり、簡便ながら、自然な音声再生が可能な符号化音声データＤ＊が得られる。
【選択図】図１２



【特許請求の範囲】
【請求項１】
特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成され人間の声を表現する符号化音声データを、異なる音高をもった別な音声データに変換する符号化音声データの音高変換装置であって、
変換対象となる符号化音声データＤを入力する変換対象データ入力ユニットと、
音高に関するオフセット値αを入力するオフセット値入力ユニットと、
周波数ｆについて定義された所定の重み関数Ｗ（ｆ）を格納した重み関数格納ユニットと、
前記変換対象となる符号化音声データＤに対して、前記オフセット値αに基づく音高の変更処理を行い、変更後の符号化音声データＤ＊を出力する音高変換処理ユニットと、
を備え、
前記重み関数格納ユニットが、前記重み関数Ｗ（ｆ）として、周波数ｆ軸上の所定区間においてＷ（ｆ）が周波数ｆの増加に従って単調減少する関数を格納しており、
前記音高変換処理ユニットが、前記重み関数Ｗ（ｆ）を用いて、前記変換対象となる符号化音声データＤに含まれている個々の符号について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行うことを特徴とする符号化音声データの音高変換装置。
【請求項２】
請求項１に記載の音高変換装置において、
重み関数格納ユニットが、重み関数Ｗ（ｆ）として、周波数ｆ軸上の第１設定値ｆａおよび第２設定値ｆｂ（ｆｂ＞ｆａ）について、ｆ≦ｆａの区間は、Ｗ（ｆ）＝１、ｆａ＜ｆ＜ｆｂの区間は、１＞Ｗ（ｆ）＞０（但し、Ｗ（ｆ）はｆの増加に従って単調減少）、ｆ≧ｆｂの区間は、Ｗ（ｆ）＝０となる関数を格納していることを特徴とする符号化音声データの音高変換装置。
【請求項３】
請求項２に記載の音高変換装置において、
重み関数格納ユニットが、１００Ｈｚ～２００Ｈｚの範囲内の第１設定値ｆａと３ｋＨｚ～６ｋＨｚの範囲内の第２設定値ｆｂとを用いた男性用重み関数Ｗ-male（ｆ）と、２００Ｈｚ～４００Ｈｚの範囲内の第１設定値ｆａと４ｋＨｚ～８ｋＨｚの範囲内の第２設定値ｆｂとを用いた女性用重み関数Ｗ-female（ｆ）と、を格納しており、
音高変換処理ユニットが、変換対象となる符号化音声データＤによって表現される声が男性の声か女性の声かを示す指示に基づいて、男性の声の場合には前記男性用重み関数Ｗ-male（ｆ）を用いた変更処理を行い、女性の声の場合には前記女性用重み関数Ｗ-female（ｆ）を用いた変更処理を行うことを特徴とする符号化音声データの音高変換装置。
【請求項４】
請求項２または３に記載の音高変換装置において、
オフセット値入力ユニットが、音高を高める場合は正、低める場合は負のオフセット値αを入力し、
音高変換処理ユニットが、所定の係数ｋ（ｋ＞１）を用いた式ｆ′＝ｆ・ｋ$^{\alpha \cdot W(f)}$により新たな周波数ｆ′を求めることを特徴とする符号化音声データの音高変換装置。
【請求項５】
請求項４に記載の音高変換装置において、
重み関数格納ユニットが、ｆａ＜ｆ＜ｆｂの区間は、Ｗ（ｆ）の値が周波数ｆの対数値に対して反比例する値となる重み関数Ｗ（ｆ）を格納していることを特徴とする符号化音声データの音高変換装置。
【請求項６】
請求項２または３に記載の音高変換装置において、
変換対象データ入力ユニットが、周波数ｆをノートナンバーｎによって示す符号を含む

符号化音声データＤを入力し、
　重み関数格納ユニットが、ノートナンバーｎについて定義された重み関数Ｗ（ｎ）を格納し、
　オフセット値入力ユニットが、音高を高める場合は正、低める場合は負の値をとるノートナンバーの差をオフセット値αとして入力し、
　音高変換処理ユニットが、変換対象となる符号化音声データＤに含まれる個々の符号について、当該符号が示すノートナンバーｎを用いた「ｎ′＝ｎ＋α・Ｗ（ｎ）」なる演算式により新たなノートナンバーｎ′を求め、当該符号を、それが示すノートナンバーｎをｎ′に変更した新たな符号に置き換える処理を行うことを特徴とする符号化音声データの音高変換装置。
【請求項７】
　請求項６に記載の音高変換装置において、
　重み関数格納ユニットが、周波数ｆ軸上の第１設定値ｆａに対応するノートナンバーｎａおよび第２設定値ｆｂに対応するノートナンバーｎｂについて、ｎａ＜ｎ＜ｎｂの区間は、Ｗ（ｎ）の値がノートナンバーｎに反比例する値となる重み関数Ｗ（ｎ）を格納していることを特徴とする符号化音声データの音高変換装置。
【請求項８】
　請求項１～７のいずれかに記載の音高変換装置において、
　変換対象データ入力ユニットが、音の持続時間が時間軸上で同一期間を占め、互いに異なる周波数を示す複数の符号を含む符号化音声データＤを入力し、
　音高変換処理ユニットが、前記複数の符号のそれぞれについて新たな符号への置換処理を行うことを特徴とする符号化音声データの音高変換装置。
【請求項９】
　請求項８に記載の音高変換装置において、
　変換対象データ入力ユニットが、周波数をノートナンバーによって示す符号を含む符号化音声データＤを入力し、
　音高変換処理ユニットが、音の持続時間が時間軸上で同一期間を占める複数の符号についてそれぞれ新たな符号への置換処理を行う際に、同一のノートナンバーを示す新たな符号が複数ｍ個生じた場合には、当該複数ｍ個の符号のうち１つのみを残し、その余の（ｍ－１）個を削除する重複回避処理を行うことを特徴とする符号化音声データの音高変換装置。
【請求項１０】
　請求項９に記載の音高変換装置において、
　変換対象データ入力ユニットが、音の強度の情報をもった符号を含む符号化音声データＤを入力し、
　音高変換処理ユニットが、重複回避処理を行う際に、１つのみ残された符号についての強度を、削除された符号についての強度に応じて修正することを特徴とする符号化音声データの音高変換装置。
【請求項１１】
　請求項１～１０のいずれかに記載の音高変換装置において、
　変換対象データ入力ユニットが、符号化音声データＤとしてＭＩＤＩ規格のデータを入力し、
　音高変換処理ユニットが、変更後の符号化音声データＤ＊としてＭＩＤＩ規格のデータを出力することを特徴とする符号化音声データの音高変換装置。
【請求項１２】
　請求項１１に記載の音高変換装置において、
　音高変換処理ユニットが、変更後の符号化音声データＤ＊を、五線譜上に音符を配置した楽譜のデータとして出力することを特徴とする符号化音声データの音高変換装置。
【請求項１３】
　請求項１～１２のいずれかに記載の音高変換装置を含む音声変換装置であって、

人間の声を含む音声信号Ｓをアナログ信号もしくはデジタル信号として入力する音声信号入力ユニットと、
前記音声信号Ｓを、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成された符号化音声データＤに変換する符号化ユニットと、
を有する音声符号化装置を更に備え、
前記符号化ユニットによって変換された符号化音声データＤの音高を、前記音高変換装置によって変更し、変更後の符号化音声データＤ＊を出力することを特徴とする音声変換装置。
【請求項１４】
請求項１～１２のいずれかに記載の音高変換装置を含む音声合成装置であって、
所定の言語による単語を文字列として表現したテキストデータを入力するテキストデータ入力ユニットと、
前記所定の言語による単語を構成する個々の音節にそれぞれ対応する符号群（特定周波数の音が特定時間だけ持続することを示す符号の集合体）を格納した符号データベースユニットと、
前記符号データベースユニットを参照して、前記テキストデータの読みを構成する個々の音節にそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって、前記テキストデータの読みに対応する人間の声を表現した符号化音声データＤを合成し、これを前記音高変換装置に与える符号合成ユニットと、
を有するテキスト符号化装置を更に備え、
前記符号合成ユニットによって合成された符号化音声データＤの音高を、前記音高変換装置によって変更し、変更後の符号化音声データＤ＊を出力することを特徴とする音声合成装置。
【請求項１５】
請求項１４に記載の音声合成装置であって、
符号データベースユニットが、子音を構成する子音音素と母音を構成する母音音素とについて、それぞれ対応する符号群を格納しており、
符号合成ユニットが、テキストデータの読みを構成する個々の音節を子音音素と母音音素とに分解し、個々の音素ごとにそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって符号化音声データＤを合成することを特徴とする音声合成装置。
【請求項１６】
請求項１４または１５に記載の音声合成装置であって、
テキストデータ入力ユニットが、音節ごとのオフセット値αを含むテキストデータを入力し、
符号合成ユニットが、合成した符号化音声データＤとともに、個々の音節ごとの前記オフセット値αを音高変換装置に与え、
前記音高変換装置が、前記符号合成ユニットから与えられたオフセット値αを用いて、個々の音節ごとに音高の変更処理を行うことを特徴とする音声合成装置。
【請求項１７】
請求項１４または１５に記載の音声合成装置であって、
テキスト符号化装置が、
個々の単語について、当該単語を構成する各音節に与えるオフセット値αを格納した音高辞書ユニットを更に備え、
符号合成ユニットが、合成した符号化音声データＤとともに、前記音高辞書ユニットを参照することにより得られる個々の音節ごとの前記オフセット値αを音高変換装置に与え、
前記音高変換装置が、前記符号合成ユニットから与えられたオフセット値αを用いて、個々の音節ごとに音高の変更処理を行うことを特徴とする音声合成装置。
【請求項１８】
請求項１３～１７のいずれかに記載の音声変換装置もしくは音声合成装置であって、

所定の楽器による様々な周波数の演奏音響波形をデジタルデータとして格納した音源ユニットと、
符号化音声データＤ＊を構成する個々の符号を、前記音源ユニットに格納されている対応する演奏音響波形に置き換えることにより音声信号の復号化を行う復号化ユニットと、
復号化された音声信号に基づいて音波を生成する発音ユニットと、
を有する音声発生装置を更に備えることを特徴とする音声変換装置もしくは音声合成装置。
【請求項１９】
請求項１～１８のいずれかに記載の音高変換装置、音声変換装置もしくは音声合成装置としてコンピュータを機能させるためのプログラム。
【請求項２０】
特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成され人間の声を表現する符号化音声データについて、その抑揚を変換する符号化音声データの抑揚変換方法であって、
コンピュータが、変換対象となる符号化音声データＤを入力する変換対象データ入力段階と、
コンピュータが、音高に関するオフセット値αを入力するオフセット値入力段階と、
コンピュータが、前記変換対象となる符号化音声データＤに対して、前記オフセット値αに基づく音高の変更処理を行い、変更後の符号化音声データＤ＊を出力する音高変換処理段階と、
を有し、
前記音高変換処理段階において、周波数ｆ軸上の所定区間においてＷ（ｆ）が周波数ｆの増加に従って単調減少する所定の重み関数Ｗ（ｆ）を利用して、前記変換対象となる符号化音声データＤに含まれている個々の符号について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行うことを特徴とする符号化音声データの抑揚変換方法。
【請求項２１】
人間の声を表現したＭＩＤＩデータを、抑揚の異なる別なＭＩＤＩデータに変換するＭＩＤＩデータの抑揚変換方法であって、
コンピュータが、変換対象となるＭＩＤＩデータＤを入力する変換対象データ入力段階と、
コンピュータが、音高に関するオフセット値αを入力するオフセット値入力段階と、
コンピュータが、前記変換対象となるＭＩＤＩデータＤに対して、前記オフセット値αに基づく音高の変更処理を行い、変更後のＭＩＤＩデータＤ＊を出力する音高変換処理段階と、
を有し、
前記変換対象データ入力段階では、互いに異なるノートナンバーをもち、時間軸上の同一位置を占める複数のＭＩＤＩ符号を含むＭＩＤＩデータＤの入力を行い、
前記音高変換処理段階では、ノートナンバー軸上の第１設定値ｎａおよび第２設定値ｎｂ（ｎｂ＞ｎａ）について、ｎ≦ｎａの区間は、Ｗ（ｎ）＝１、ｎａ＜ｎ＜ｎｂの区間は、１＞Ｗ（ｎ）＞０（但し、Ｗ（ｎ）はｎの増加に従って単調減少）、ｎ≧ｎｂの区間は、Ｗ（ｎ）＝０となる所定の重み関数Ｗ（ｎ）を利用して、前記変換対象となるＭＩＤＩデータＤに含まれている個々のＭＩＤＩ符号について、当該ＭＩＤＩ符号が示すノートナンバーｎに対してα・Ｗ（ｎ）に応じた値だけ加減算を行うことにより新たなノートナンバーｎ′を求め、当該ＭＩＤＩ符号を、それが示すノートナンバーｎをｎ′に変更した新たなＭＩＤＩ符号に置き換える処理を行うことにより、変更後のＭＩＤＩデータＤ＊を生成することを特徴とするＭＩＤＩデータの抑揚変換方法。
【請求項２２】
請求項２１に記載のＭＩＤＩデータの抑揚変換方法において、

音高変換処理段階で、新たなＭＩＤＩ符号に置き換える処理を行う際に、時間軸上の同一位置を占め、同一のノートナンバーをもつ新たなＭＩＤＩ符号が複数ｍ個生じた場合には、当該複数ｍ個のＭＩＤＩ符号のうち１つのみを残し、その余の（ｍ－１）個を削除する重複回避処理を行うことを特徴とするＭＩＤＩデータの抑揚変換方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、符号化音声データの音高変換装置およびその応用技術に関し、特に、人間の声を表現したＭＩＤＩ規格等の符号化音声データを、異なる音高をもった別な音声データに変換する技術に関する。

【背景技術】

【０００２】

音響信号を符号化する技術としては、ＰＣＭ（Pulse Code Modulation ）の手法が古くから知られており、現在、コンピュータをはじめとする多くのデジタル機器で利用されているデジタル音響データは、このＰＣＭの手法を用いてデジタル化したデータである。

【０００３】

一方、最近では、パーソナルコンピュータの普及とともに、ＭＩＤＩ規格（Musical Instrument Digital Interface）による符号化も一般化してきている。もともとＭＩＤＩ規格は、電子楽器による楽器音を符号化しようという発想から生まれた規格であり、ＰＣＭとは全く異なる固有の特徴を有している。すなわち、ＭＩＤＩ規格による符号データ（以下、ＭＩＤＩデータという）は、基本的には、楽器のどの鍵盤キーを、どの程度の強さで弾いたか、という楽器演奏の操作を記述したデータであり、このＭＩＤＩデータ自身には、実際の音の波形は含まれていない。そのため、実際の音を再生する場合には、楽器音の波形を記憶したＭＩＤＩ音源が別途必要になる。しかしながら、上述したＰＣＭの手法で音を記録する場合に比べて、情報量が極めて少なくてすむという特徴を有し、その符号化効率の高さが注目を集めている。

【０００４】

最近では、楽器音に限らず、人間の声などを含めた様々な音響信号をＭＩＤＩデータによって符号化しようとする試みもなされている。たとえば、下記の特許文献１および２には、ＭＩＤＩデータを利用することが可能な新規な符号化方法が提案されている。これらの方法では、音響信号の時間軸に沿って複数の単位区間を設定し、各単位区間ごとにフーリエ変換を行ってスペクトルを求め、このスペクトルに応じたＭＩＤＩデータを作成するという手順が実行される。また、下記の特許文献３には、人間の声や歌声を含む、いわゆるヴォーカル音響信号について、ＭＩＤＩデータを作成する効率的な手法が提案されている。更に、下記の特許文献４には、日本語のカナ文字を構成する各音節を符号コードで表現し、人間の声を基にしてＭＩＤＩデータを作成する技術が提案されている。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開平１１－９５７５３号公報

【特許文献２】特開２０００－９９００９号公報

【特許文献３】特開２０００－９９０９３号公報

【特許文献４】特願２００９－２４４６９８号明細書

【発明の概要】

【発明が解決しようとする課題】

【０００６】

人間の話し声には抑揚があり、同じ音節からなる言葉でも、抑揚を変化させてしゃべると、それぞれ違った印象をもった言葉に聞こえてくる。また、人間の歌声、いわゆるヴォーカル音にも、当然ながら、音の高低が存在する。したがって、人間の話し声や歌声を表現した符号化音声データについて、その音高（ピッチ）を自由に改変することができれば

、その応用範囲は多岐に広がることになる。たとえば、人間の話し声に対して、音節単位で音高を上下させることができれば、言葉の抑揚を変化させたり、所望の曲に合わせた歌声を作成したりすることができる。

【０００７】

ＭＩＤＩデータなどの符号化データに対する音高シフトは、音楽でいう、いわゆる移調演奏に相当するものであり、通常、すべての音符のノートナンバーに対して、一律に所定のオフセット値を加減算することによって行われる。たとえば、一般的な音楽ＭＩＤＩデータについて、１オクターブだけ上げる音高シフトを行うのであれば、すべての音符のノートナンバーｎに対して、１オクターブの音程に相当するオフセット値「１２」を加える加算処理を行えばよい。

【０００８】

しかしながら、本願発明者が行った実験によると、人間の声を表現したＭＩＤＩデータについて、同様の方法で音高シフトを行うと、音声の明瞭性は維持されるものの、声質が破壊され、違和感のある奇怪な声になってしまうことが判明した。たとえば、人間の声を表現したＭＩＤＩデータについて、１オクターブだけ上げる音高シフトを行うために、すべての音符のノートナンバーｎに対して、一律にオフセット値「１２」を加える加算処理を実行すると、個々の単語の判別は可能であるものの、いわゆるボイスチェンジャーを通したような違和感のある不自然な声になってしまう。

【０００９】

本願発明者は、当初、人間の声を表現したＭＩＤＩ規格等の符号化音声データに対して、所望の音高シフト処理を施し、違和感のない自然な音声再生が可能な符号化音声データを生成するためには、人間の声に含まれる音声成分に対する高度な信号解析技術を導入する必要があると考えていた。しかしながら、そのような高度な音声信号解析技術を導入した処理を実行するためには、高性能なハードウエアを用いた複雑な処理が必要となり、そのような処理機能をもった音高変換装置は極めてコストの高い装置にならざるを得ない。

【００１０】

そこで本発明は、人間の声を表現した符号化音声データに対して、できるだけ簡便な方法により所望の音高シフト処理を施し、違和感のない自然な音声再生が可能な符号化音声データを得ることができる符号化音声データの音高変換装置および符号化音声データの抑揚変換方法を提供することを目的とする。

【００１１】

また、本発明は、上記音高変換装置を利用して、入力した人間の音声の音高を変更し、違和感のない自然な音声再生が可能な符号化音声データを得ることができる音声変換装置を提供することを目的とし、更に、所定の言語による文章を文字列として表現したテキストデータに基づいて、自然な抑揚をもった音声再生が可能な符号化音声データを得ることができる音声合成装置を提供することを目的とする。

【課題を解決するための手段】

【００１２】

(1)　本発明の第１の態様は、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成され人間の声を表現する符号化音声データを、異なる音高をもった別な音声データに変換する符号化音声データの音高変換装置において、

変換対象となる符号化音声データＤを入力する変換対象データ入力ユニットと、

音高に関するオフセット値αを入力するオフセット値入力ユニットと、

周波数ｆについて定義された所定の重み関数Ｗ（ｆ）を格納した重み関数格納ユニットと、

変換対象となる符号化音声データＤに対して、オフセット値αに基づく音高の変更処理を行い、変更後の符号化音声データＤ＊を出力する音高変換処理ユニットと、

を設け、

重み関数格納ユニットが、重み関数Ｗ（ｆ）として、周波数ｆ軸上の所定区間においてＷ（ｆ）が周波数ｆの増加に従って単調減少する関数を格納しており、

音高変換処理ユニットが、重み関数Ｗ（ｆ）を用いて、変換対象となる符号化音声データＤに含まれている個々の符号について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行うようにしたものである。

【００１３】

(2)　本発明の第２の態様は、上述の第１の態様に係る符号化音声データの音高変換装置において、

重み関数格納ユニットが、重み関数Ｗ（ｆ）として、周波数ｆ軸上の第１設定値ｆａおよび第２設定値ｆｂ（ｆｂ＞ｆａ）について、ｆ≦ｆａの区間は、Ｗ（ｆ）＝１、ｆａ＜ｆ＜ｆｂの区間は、１＞Ｗ（ｆ）＞０（但し、Ｗ（ｆ）はｆの増加に従って単調減少）、ｆ≧ｆｂの区間は、Ｗ（ｆ）＝０となる関数を格納しているようにしたものである。

【００１４】

(3)　本発明の第３の態様は、上述の第２の態様に係る符号化音声データの音高変換装置において、

重み関数格納ユニットが、１００Ｈｚ～２００Ｈｚの範囲内の第１設定値ｆａと３ｋＨｚ～６ｋＨｚの範囲内の第２設定値ｆｂとを用いた男性用重み関数W-male（ｆ）と、２００Ｈｚ～４００Ｈｚの範囲内の第１設定値ｆａと４ｋＨｚ～８ｋＨｚの範囲内の第２設定値ｆｂとを用いた女性用重み関数W-female（ｆ）と、を格納しており、

音高変換処理ユニットが、変換対象となる符号化音声データＤによって表現される声が男性の声か女性の声かを示す指示に基づいて、男性の声の場合には男性用重み関数W-male（ｆ）を用いた変更処理を行い、女性の声の場合には女性用重み関数W-female（ｆ）を用いた変更処理を行うようにしたものである。

【００１５】

(4)　本発明の第４の態様は、上述の第２または第３の態様に係る符号化音声データの音高変換装置において、

オフセット値入力ユニットが、音高を高める場合は正、低める場合は負のオフセット値αを入力し、

音高変換処理ユニットが、所定の係数ｋ（ｋ＞１）を用いた式$f' = f \cdot k^{\alpha \cdot W(f)}$により新たな周波数ｆ′を求めるようにしたものである。

【００１６】

(5)　本発明の第５の態様は、上述の第４の態様に係る符号化音声データの音高変換装置において、

重み関数格納ユニットが、ｆａ＜ｆ＜ｆｂの区間は、Ｗ（ｆ）の値が周波数ｆの対数値に対して反比例する値となる重み関数Ｗ（ｆ）を格納しているようにしたものである。

【００１７】

(6)　本発明の第６の態様は、上述の第２または第３の態様に係る符号化音声データの音高変換装置において、

変換対象データ入力ユニットが、周波数ｆをノートナンバーｎによって示す符号を含む符号化音声データＤを入力し、

重み関数格納ユニットが、ノートナンバーｎについて定義された重み関数Ｗ（ｎ）を格納し、

オフセット値入力ユニットが、音高を高める場合は正、低める場合は負の値をとるノートナンバーの差をオフセット値αとして入力し、

音高変換処理ユニットが、変換対象となる符号化音声データＤに含まれる個々の符号について、当該符号が示すノートナンバーｎを用いた「ｎ′＝ｎ＋α・Ｗ（ｎ）」なる演算式により新たなノートナンバーｎ′を求め、当該符号を、それが示すノートナンバーｎをｎ′に変更した新たな符号に置き換える処理を行うようにしたものである。

【００１８】

(7)　本発明の第７の態様は、上述の第６の態様に係る符号化音声データの音高変換装置において、

重み関数格納ユニットが、周波数ｆ軸上の第１設定値ｆａに対応するノートナンバーｎａおよび第２設定値ｆｂに対応するノートナンバーｎｂについて、ｎａ＜ｎ＜ｎｂの区間は、Ｗ（ｎ）の値がノートナンバーｎに反比例する値となる重み関数Ｗ（ｎ）を格納しているようにしたものである。

【００１９】

(8)　本発明の第８の態様は、上述の第１～第７の態様に係る符号化音声データの音高変換装置において、

変換対象データ入力ユニットが、音の持続時間が時間軸上で同一期間を占め、互いに異なる周波数を示す複数の符号を含む符号化音声データＤを入力し、

音高変換処理ユニットが、複数の符号のそれぞれについて新たな符号への置換処理を行うようにしたものである。

【００２０】

(9)　本発明の第９の態様は、上述の第８の態様に係る符号化音声データの音高変換装置において、

変換対象データ入力ユニットが、周波数をノートナンバーによって示す符号を含む符号化音声データＤを入力し、

音高変換処理ユニットが、音の持続時間が時間軸上で同一期間を占める複数の符号についてそれぞれ新たな符号への置換処理を行う際に、同一のノートナンバーを示す新たな符号が複数ｍ個生じた場合には、当該複数ｍ個の符号のうち１つのみを残し、その余の（ｍ－１）個を削除する重複回避処理を行うようにしたものである。

【００２１】

(10)　本発明の第１０の態様は、上述の第９の態様に係る符号化音声データの音高変換装置において、

変換対象データ入力ユニットが、音の強度の情報をもった符号を含む符号化音声データＤを入力し、

音高変換処理ユニットが、重複回避処理を行う際に、１つのみ残された符号についての強度を、削除された符号についての強度に応じて修正するようにしたものである。

【００２２】

(11)　本発明の第１１の態様は、上述の第１～第１０の態様に係る符号化音声データの音高変換装置において、

変換対象データ入力ユニットが、符号化音声データＤとしてＭＩＤＩ規格のデータを入力し、

音高変換処理ユニットが、変更後の符号化音声データＤ＊としてＭＩＤＩ規格のデータを出力するようにしたものである。

【００２３】

(12)　本発明の第１２の態様は、上述の第１１の態様に係る符号化音声データの音高変換装置において、

音高変換処理ユニットが、変更後の符号化音声データＤ＊を、五線譜上に音符を配置した楽譜のデータとして出力するようにしたものである。

【００２４】

(13)　本発明の第１３の態様は、上述の第１～第１２の態様に係る符号化音声データの音高変換装置に、

人間の声を含む音声信号Ｓをアナログ信号もしくはデジタル信号として入力する音声信号入力ユニットと、

音声信号Ｓを、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成された符号化音声データＤに変換する符号化ユニットと、

を有する音声符号化装置を更に付加し、

符号化ユニットによって変換された符号化音声データＤの音高を、音高変換装置によって変更し、変更後の符号化音声データＤ＊を出力する機能をもった音声変換装置を構成するようにしたものである。

【００２５】
(14) 本発明の第１４の態様は、上述の第１～第１２の態様に係る符号化音声データの音高変換装置に、
所定の言語による単語を文字列として表現したテキストデータを入力するテキストデータ入力ユニットと、
所定の言語による単語を構成する個々の音節にそれぞれ対応する符号群（特定周波数の音が特定時間だけ持続することを示す符号の集合体）を格納した符号データベースユニットと、
符号データベースユニットを参照して、テキストデータの読みを構成する個々の音節にそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって、テキストデータの読みに対応する人間の声を表現した符号化音声データＤを合成し、これを音高変換装置に与える符号合成ユニットと、
を有するテキスト符号化装置を更に付加し、
符号合成ユニットによって合成された符号化音声データＤの音高を、音高変換装置によって変更し、変更後の符号化音声データＤ＊を出力する機能をもった音声合成装置を構成するようにしたものである。
【００２６】
(15) 本発明の第１５の態様は、上述の第１４の態様に係る音声合成装置において、
符号データベースユニットが、子音を構成する子音音素と母音を構成する母音音素とについて、それぞれ対応する符号群を格納しており、
符号合成ユニットが、テキストデータの読みを構成する個々の音節を子音音素と母音音素とに分解し、個々の音素ごとにそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって符号化音声データＤを合成するようにしたものである。
【００２７】
(16) 本発明の第１６の態様は、上述の第１４または第１５の態様に係る音声合成装置において、
テキストデータ入力ユニットが、音節ごとのオフセット値αを含むテキストデータを入力し、
符号合成ユニットが、合成した符号化音声データＤとともに、個々の音節ごとのオフセット値αを音高変換装置に与え、
音高変換装置が、符号合成ユニットから与えられたオフセット値αを用いて、個々の音節ごとに音高の変更処理を行うようにしたものである。
【００２８】
(17) 本発明の第１７の態様は、上述の第１４または第１５の態様に係る音声合成装置において、
テキスト符号化装置が、
個々の単語について、当該単語を構成する各音節に与えるオフセット値αを格納した音高辞書ユニットを更に備え、
符号合成ユニットが、合成した符号化音声データＤとともに、音高辞書ユニットを参照することにより得られる個々の音節ごとのオフセット値αを音高変換装置に与え、
音高変換装置が、符号合成ユニットから与えられたオフセット値αを用いて、個々の音節ごとに音高の変更処理を行うようにしたものである。
【００２９】
(18) 本発明の第１８の態様は、上述の第１３～第１７の態様に係る音声変換装置もしくは音声合成装置において、
所定の楽器による様々な周波数の演奏音響波形をデジタルデータとして格納した音源ユニットと、
符号化音声データＤ＊を構成する個々の符号を、音源ユニットに格納されている対応する演奏音響波形に置き換えることにより音声信号の復号化を行う復号化ユニットと、
復号化された音声信号に基づいて音波を生成する発音ユニットと、

を有する音声発生装置を更に設けるようにしたものである。

【００３０】

(19) 本発明の第１９の態様は、上述の第１～第１８の態様に係る音高変換装置、音声変換装置もしくは音声合成装置を、コンピュータにプログラムを組み込むことにより構成したものである。

【００３１】

(20) 本発明の第２０の態様は、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成され人間の声を表現する符号化音声データについて、その抑揚を変換する符号化音声データの抑揚変換方法において、

コンピュータが、変換対象となる符号化音声データＤを入力する変換対象データ入力段階と、

コンピュータが、音高に関するオフセット値αを入力するオフセット値入力段階と、

コンピュータが、変換対象となる符号化音声データＤに対して、オフセット値αに基づく音高の変更処理を行い、変更後の符号化音声データＤ＊を出力する音高変換処理段階と、

を行い、

音高変換処理段階において、周波数ｆ軸上の所定区間においてＷ（ｆ）が周波数ｆの増加に従って単調減少する所定の重み関数Ｗ（ｆ）を利用して、変換対象となる符号化音声データＤに含まれている個々の符号について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行うようにしたものである。

【００３２】

(21) 本発明の第２１の態様は、人間の声を表現したＭＩＤＩデータを、抑揚の異なる別なＭＩＤＩデータに変換するＭＩＤＩデータの抑揚変換方法において、

コンピュータが、変換対象となるＭＩＤＩデータＤを入力する変換対象データ入力段階と、

コンピュータが、音高に関するオフセット値αを入力するオフセット値入力段階と、

コンピュータが、変換対象となるＭＩＤＩデータＤに対して、オフセット値αに基づく音高の変更処理を行い、変更後のＭＩＤＩデータＤ＊を出力する音高変換処理段階と、

を行い、

変換対象データ入力段階では、互いに異なるノートナンバーをもち、時間軸上の同一位置を占める複数のＭＩＤＩ符号を含むＭＩＤＩデータＤの入力を行い、

音高変換処理段階では、ノートナンバー軸上の第１設定値ｎａおよび第２設定値ｎｂ（ｎｂ＞ｎａ）について、ｎ≦ｎａの区間は、Ｗ（ｎ）＝１、ｎａ＜ｎ＜ｎｂの区間は、１＞Ｗ（ｎ）＞０（但し、Ｗ（ｎ）はｎの増加に従って単調減少）、ｎ≧ｎｂの区間は、Ｗ（ｎ）＝０となる所定の重み関数Ｗ（ｎ）を利用して、変換対象となるＭＩＤＩデータＤに含まれている個々のＭＩＤＩ符号について、当該ＭＩＤＩ符号が示すノートナンバーｎに対してα・Ｗ（ｎ）に応じた値だけ加減算を行うことにより新たなノートナンバーｎ′を求め、当該ＭＩＤＩ符号を、それが示すノートナンバーｎをｎ′に変更した新たなＭＩＤＩ符号に置き換える処理を行うことにより、変更後のＭＩＤＩデータＤ＊を生成するようにしたものである。

【００３３】

(22) 本発明の第２２の態様は、上述の第２１の態様に係るＭＩＤＩデータの抑揚変換方法において、

音高変換処理段階で、新たなＭＩＤＩ符号に置き換える処理を行う際に、時間軸上の同一位置を占め、同一のノートナンバーをもつ新たなＭＩＤＩ符号が複数ｍ個生じた場合には、当該複数ｍ個のＭＩＤＩ符号のうち１つのみを残し、その余の（ｍ－１）個を削除する重複回避処理を行うようにしたものである。

【発明の効果】

【００３４】

本発明に係る符号化音声データの音高変換装置および符号化音声データの抑揚変換方法によれば、変換対象となる符号化音声データＤに対して、オフセット値αに基づく音高の変更処理を行って符号化音声データＤ＊を得る際に、周波数ｆの増加に従って単調減少する重み関数Ｗ（ｆ）を用いて、符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減するようにしたため、より高音の符号ほど音高シフト量が減少することになる。このため、人間の発生する音声に含まれる基本周波数成分の音高シフト量に比べて、当該基本周波数成分の整数倍の周波数をもつフォルマント成分についての音高シフト量が小さくなり、「フォルマント成分の絶対周波数値はほぼ一定」という人間の声音に固有の特徴を維持したまま、音高を変更することができる。かくして、比較的簡便な方法で、人間の声を表現したＭＩＤＩ規格等の符号化音声データを、異なる音高、抑揚をもった別な音声データに変換することができるようになり、しかも違和感のない自然な音声再生が可能な符号化音声データが得られる。

【００３５】

なお、本発明で取り扱う符号化音声データは、たとえばＭＩＤＩデータのように、特定周波数の音が特定時間だけ持続することを示すいくつかの符号を時間軸上に並べることによって構成されるデータであり、いわば人間の声を音符で表現したデータというべきものである。したがって、そのような符号化音声データを再生したとしても、当然ながら、元の人間の声に忠実な音が生成されるわけではなく、人間の声に対して、かなりかけ離れた音が得られることになる。したがって、人間の声に忠実なリアルな音か否かという観点では、当然ながら、本発明は「違和感のない自然な音声再生」を可能にする技術ではない。本発明の効果にいう「違和感のない自然な音声再生」とは、「人間の声がもつフォルマント成分に起因して生じる固有の特徴が損なわれることのない再生」を意味するものであり、いわゆるボイスチェンジャーを通した場合に得られるような違和感や不自然さが排除されることを意味するものである。

【００３６】

本発明では、特に、平均的な音声についての基本周波数成分（Ｆ０：０次フォルマント成分）が含まれると予想される周波数領域の上限に第１設定値ｆａを定義し、聴取可能な最高次フォルマントF^{max}成分が含まれると予想される周波数領域の上限に第２設定値ｆｂを定義し、ｆ≦ｆａの区間は、Ｗ（ｆ）＝１、ｆａ＜ｆ＜ｆｂの区間は、１＞Ｗ（ｆ）＞０（但し、Ｗ（ｆ）はｆの増加に従って単調減少）、ｆ≧ｆｂの区間は、Ｗ（ｆ）＝０となる関数を重み関数Ｗ（ｆ）として設定すれば、非常に単純な演算によって音高の変更処理を行うことができる。特に、ＭＩＤＩデータのように、周波数ｆをその対数値に対応するノートナンバーｎで表す符号データを用いる場合、重み関数としてｎの関数Ｗ（ｎ）を用いるようにすれば、音高の変更処理は、ノートナンバーｎに対して、α・Ｗ（ｎ）を加算もしくは減算するだけの単純な処理になる。上記周波数ｆａ＜ｆ＜ｆｂの区間について、ノートナンバーｎに反比例する値をとる関数値Ｗ（ｎ）を用いるようにすれば、演算はより単純になる。

【００３７】

なお、ＭＩＤＩ規格等の符号化音声データによって人間の声を表すと、音の持続時間が時間軸上で同一期間を占め、互いに異なる周波数を示す複数の符号（いわゆる、和音を構成する複数の符号）を含むデータが生じることになるが、このような和音を構成する複数の符号に対して音高の変更処理を行うと、複数の符号が変更後に同じ音高を占める場合がありえる。このように同じ音高の符号が重複した場合には、１つのみを残して他を削除するようにすれば、重複を許さないＭＩＤＩ規格のような符号化データに対しても問題なく適用可能になる。また、削除された符号についての強度に応じて強度修正を行えば、音高の変更処理後も適切な強度バランスをもった符号化データが得られる。

【００３８】

また、上記音高変換装置を利用すれば、入力した人間の音声の音高を自由に変更して、違和感のない自然な音声再生が可能な符号化音声データを生成する音声変換装置を実現することができる。更に、所定の言語による文章を文字列として表現したテキストデータに

基づいて、自然な抑揚をもった音声再生が可能な符号化音声データを得ることができる音声合成装置を実現することもできる。

【図面の簡単な説明】

【００３９】

【図１】フーリエ変換を利用した音響信号の符号化方法の基本原理を示す図である。

【図２】図１(c) に示す強度グラフに基いて作成された符号コードを示す図である。

【図３】時間軸上に部分的に重複するように単位区間設定を行うことにより作成された符号コードを示す図である。

【図４】和音に対して音高を１オクターブだけ上げる一般的な処理を示す楽譜である。

【図５】人間の声に含まれるフォルマント成分を例示する強度グラフ（人間の音声スペクトルのピーク位置を示すグラフ）である。

【図６】図(a) は、標準的な音高で発声を行った人間の音声スペクトルの強度グラフ、図(b) は、同じ人が音高を高めて発声を行った場合の強度グラフ、図(c) は、同じ人が音高を低めて発声を行った場合の強度グラフである。

【図７】図(a) は、本発明で用いる、周波数ｆについての重み関数Ｗ（ｆ）の一例を示すグラフ（横軸は、周波数ｆについての線形スケール）、図(b) は、本発明で用いる、ノートナンバーｎについての重み関数Ｗ（ｎ）の一例を示すグラフ（横軸は、周波数ｆについての対数スケール）である。

【図８】図(a) は、変換前の符号化データに含まれる和音の一例を示す楽譜、図(b) は、従来の方法で音高を１オクターブ上げる処理を行った結果を示す楽譜、図(c) は、本発明の方法で音高を１オクターブ上げる処理を行った結果を示す楽譜である。

【図９】図(a) は、変換前の符号化データに含まれる和音の一例を示す楽譜、図(b) は、従来の方法で音高を１オクターブ下げる処理を行った結果を示す楽譜、図(c) は、本発明の方法で音高を１オクターブ下げる処理を行った結果を示す楽譜である。

【図１０】図(a) は、変換前の符号化データに含まれる和音の一例を示す楽譜、図(b) は、従来の方法で音高を１オクターブ上げる処理を行った結果を示す楽譜、図(c) は、本発明の方法で音高を１オクターブ上げる処理を行った結果（３つの音符が重なっている状態）を示す楽譜である。

【図１１】音高変更後に複数の符号が同じ音高を占める場合に、１つのみを残して他を削除する処理を示す楽譜である。

【図１２】本発明の基本的実施形態に係る符号化音声データの音高変換装置の構成を示すブロック図である。

【図１３】図１２に示す音高変換装置を利用した音声変換装置の構成を示すブロック図である。

【図１４】図１２に示す音高変換装置を利用した音声合成装置の構成を示すブロック図である。

【図１５】図１４に示す音声合成装置の符号合成ユニット３３０内に用意されている変換テーブルを示す図である。

【図１６】図１５に示す変換テーブルに基づいて、テキストデータを構成する個々の音節を子音音素と母音音素とに分解した例を示す図である。

【図１７】図１４に示す音声合成装置の符号データベースユニット３１０内に用意されている男性用のＭＩＤＩ符号データベースの一例を示す表である。

【図１８】図１４に示す音声合成装置の符号データベースユニット３１０内に用意されている女性用のＭＩＤＩ符号データベースの一例を示す表である。

【図１９】図１７に示す男性用のＭＩＤＩ符号データベースを用いて、各母音音素と各子音音素とに対応する符号を、音符として表示した例を示す楽譜である。

【図２０】図１８に示す女性用のＭＩＤＩ符号データベースを用いて、各母音音素と各子音音素とに対応する符号を、音符として表示した例を示す楽譜である。

【図２１】図１４に示す音声合成装置の第１の変形例を示すブロック図である。

【図２２】図２１に示す音声合成装置の符号合成ユニット３４０の処理機能を説明する図

である。
【図２３】図１４に示す音声合成装置の第２の変形例を示すブロック図である。
【図２４】図２３に示す音声合成装置の音高辞書ユニット３６０の内容を例示する表である。
【図２５】本発明に係る音高変換装置から出力された符号化音声データＤ＊に基づいて、音声を発声させる音声発声装置の基本構成を示すブロック図である。
【発明を実施するための形態】
【００４０】
以下、本発明を図示する実施形態に基づいて説明する。
【００４１】
＜＜＜　§１．音響信号の符号化方法の基本原理　＞＞＞
はじめに、前掲の特許文献１～３に開示されているフーリエ変換を利用した音響信号の符号化方法の基本原理を簡単に説明しておく。この符号化方法を利用すれば、たとえば、人間の音声をアナログ音響信号として取り込み、ＭＩＤＩデータなどのデジタル符号データに変換することができる。
【００４２】
いま、図１(a) に示すように、時系列の強度信号としてアナログ音響信号が与えられたものとしよう。図示の例では、横軸に時間ｔ、縦軸に振幅（強度）をとってこの音響信号を示している。ここでは、まずこのアナログ音響信号を、デジタルの音響データとして取り込む処理を行う。これは、従来の一般的なＰＣＭの手法を用い、所定のサンプリング周期でこのアナログ音響信号をサンプリングし、振幅を所定の量子化ビット数を用いてデジタルデータに変換する処理を行えばよい。ここでは、説明の便宜上、ＰＣＭの手法でデジタル化した音響データの波形も、図１(a) のアナログ音響信号と同一の波形で示すことにする。
【００４３】
続いて、この符号化対象となる音響信号の時間軸上に、複数の単位区間を設定する。図１(a) に示す例では、時間軸ｔ上に等間隔に６つの時刻ｔ１～ｔ６が定義され、これら各時刻を始点および終点とする５つの単位区間ｄ１～ｄ５が設定されている。実際には、後述するように、各単位区間ｄ１～ｄ５は部分的に重複するように設定するのが好ましい。
【００４４】
こうして単位区間が設定されたら、単位区間ごとの音響信号（ここでは、区間信号と呼ぶことにする）に対してそれぞれフーリエ変換を行い、スペクトルを作成する。このとき、ハニング窓（Hanning Window)などの重み関数で、切り出した区間信号にフィルタをかけてフーリエ変換を施す。一般にフーリエ変換は、切り出した区間前後に同様な信号が無限に存在することが想定されているため、重み関数を用いない場合、作成したスペクトルに高周波ノイズがのることが多い。ハニング窓関数など区間の両端の重みが０になるような重み関数を用いると、このような弊害をある程度抑制できる。ハニング窓関数Ｈ（ｋ）は、単位区間長をＬとすると、ｋ＝１…Ｌに対して、

$$H(k) = 0.5 - 0.5 \cdot \cos(2\pi k / L)$$

で与えられる関数である。
【００４５】
図１(b) には、単位区間ｄ１について作成されたスペクトルの一例が示されている。このスペクトルでは、横軸上に定義された周波数ｆによって、単位区間ｄ１についての区間信号に含まれる周波数成分（０～Ｆ：ここでＦはサンプリング周波数の１／２）が示されており、縦軸上に定義された複素強度Ａによって、周波数成分ごとの複素強度が示されている。
【００４６】
次に、このスペクトルの周波数軸ｆに対応させて、離散的に複数Ｘ個の符号コードを定義する。この例では、符号コードとしてＭＩＤＩデータで利用されるノートナンバーｎを用いており、ｎ＝０～１２７までの１２８個の符号コードを定義している。ノートナンバ

ーｎは、音符の音階を示すパラメータであり、たとえば、ノートナンバーｎ＝６９は、ピアノの鍵盤中央の「ラ音（Ａ３音）」を示しており、４４０Ｈｚの音に相当する。このように、１２８個のノートナンバーには、いずれも所定の周波数が対応づけられるので、スペクトルの周波数軸ｆ上の所定位置に、それぞれ１２８個のノートナンバーｎが離散的に定義されることになる。

【００４７】

ここで、ノートナンバーｎは、１オクターブ上がると、周波数が２倍になる対数尺度の音階を示すため、周波数軸ｆに対して線形には対応しない。そこで、ここでは周波数軸ｆを対数尺度で表し、この対数尺度軸上にノートナンバーｎを定義した強度グラフを作成してみる。図１(c) は、このようにして作成された単位区間ｄ１についての強度グラフを示す。この強度グラフの横軸は、図１(b) に示すスペクトルの横軸を対数尺度に変換したものであり、ノートナンバーｎ＝０～１２７が等間隔にプロットされている。一方、この強度グラフの縦軸は、図１(b) に示すスペクトルの複素強度Ａを実効強度Ｅに変換したものであり、各ノートナンバーｎの位置における強度を示している。一般に、フーリエ変換によって得られる複素強度Ａは、実数部Ｒと虚数部Ｉとによって表されるが、実効強度Ｅ（エネルギー）は、$E=\sqrt{(R^2+I^2)}$ なる演算によって求めることができる。

【００４８】

こうして求められた単位区間ｄ１の強度グラフは、単位区間ｄ１についての区間信号に含まれる振動成分について、ノートナンバーｎ＝０～１２７に相当する各振動成分の割合を実効強度として示すグラフということができる。そこで、この強度グラフに示されている各実効強度に基いて、全Ｘ個（この例ではＸ＝１２８）のノートナンバーの中からＰ個のノートナンバーを選択し、このＰ個のノートナンバーｎを、単位区間ｄ１を代表する代表符号コードとして抽出する。ここでは、説明の便宜上、Ｐ＝３として、全１２８個の候補の中から３個のノートナンバーを代表符号コードとして抽出する場合を示すことにする。たとえば、「候補の中から強度の大きい順にＰ個の符号コードを抽出する」という基準に基いて抽出を行えば、図１(c) に示す例では、第１番目の代表符号コードとしてノートナンバーｎ（ｄ１，１）が、第２番目の代表符号コードとしてノートナンバーｎ（ｄ１，２）が、第３番目の代表符号コードとしてノートナンバーｎ（ｄ１，３）が、それぞれ抽出されることになる。

【００４９】

このようにして、Ｐ個の代表符号コードが抽出されたら、これらの代表符号コードとその実効強度によって、単位区間ｄ１についての区間信号を表現することができる。たとえば、上述の例の場合、図１(c) に示す強度グラフにおいて、ノートナンバーｎ（ｄ１，１）、ｎ（ｄ１，２）、ｎ（ｄ１，３）の実効強度がそれぞれｅ（ｄ１，１）、ｅ（ｄ１，２）、ｅ（ｄ１，３）であったとすれば、以下に示す３組のデータ対によって、単位区間ｄ１の音響信号を表現することができる。

ｎ（ｄ１，１），ｅ（ｄ１，１）
ｎ（ｄ１，２），ｅ（ｄ１，２）
ｎ（ｄ１，３），ｅ（ｄ１，３）

【００５０】

以上、単位区間ｄ１についての処理について説明したが、単位区間ｄ２～ｄ５についても、それぞれ別個に同様の処理が行われ、代表符号コードおよびその強度を示すデータが得られることになる。たとえば、単位区間ｄ２については、

ｎ（ｄ２，１），ｅ（ｄ２，１）
ｎ（ｄ２，２），ｅ（ｄ２，２）
ｎ（ｄ２，３），ｅ（ｄ２，３）

なる３組のデータ対が得られる。このようにして単位区間ごとに得られたデータによって、原音響信号を符号化することができる。

【００５１】

図２は、上述の方法による符号化の概念図である。図２(a) には、図１(a) と同様に、

原音響信号について５つの単位区間ｄ１～ｄ５を設定した状態が示されており、図２(b)には、単位区間ごとに得られた符号データが音符の形式で示されている。この例では、個々の単位区間ごとに３個の代表符号コードを抽出しており（Ｐ＝３）、これら代表符号コードに関するデータを３つのトラックＴ１～Ｔ３に分けて収容するようにしている。たとえば、単位区間ｄ１について抽出された代表符号コードｎ（ｄ１，１），ｎ（ｄ１，２），ｎ（ｄ１，３）は、それぞれトラックＴ１，Ｔ２，Ｔ３に収容されている。もっとも、図２(b)は、上述の方法によって得られる符号データを音符の形式で示した概念図であり、実際には、各音符にはそれぞれ強度に関するデータが付加されている。たとえば、トラックＴ１には、ノートナンバーｎ（ｄ１，１），ｎ（ｄ２，１），ｎ（ｄ３，１）…なる音階を示すデータとともに、ｅ（ｄ１，１），ｅ（ｄ２，１），ｅ（ｄ３，１）…なる強度を示すデータが収容されることになる。

【００５２】

なお、ここで採用する符号化の形式としては、必ずしもＭＩＤＩ形式を採用する必要はないが、この種の符号化形式としてはＭＩＤＩ形式が最も普及しているため、実用上はＭＩＤＩ形式の符号データを用いるのが最も好ましい。ＭＩＤＩ形式では、「ノートオン」データもしくは「ノートオフ」データが、「デルタタイム」データを介在させながら存在する。「ノートオン」データは、特定のノートナンバーＮとベロシティーＶとを指定して特定の音の演奏開始を指示するデータであり、「ノートオフ」データは、特定のノートナンバーＮとベロシティーＶとを指定して特定の音の演奏終了を指示するデータである。また、「デルタタイム」データは、所定の時間間隔を示すデータである。ベロシティーＶは、たとえば、ピアノの鍵盤などを押し下げる速度（ノートオン時のベロシティー）および鍵盤から指を離す速度（ノートオフ時のベロシティー）を示すパラメータであり、特定の音の演奏開始操作もしくは演奏終了操作の強さを示すことになる。

【００５３】

前述の方法では、第ｉ番目の単位区間ｄｉについて、代表符号コードとしてＰ個のノートナンバーｎ（ｄｉ，１），ｎ（ｄｉ，２），…，ｎ（ｄｉ，Ｐ）が得られ、このそれぞれについて実効強度ｅ（ｄｉ，１），ｅ（ｄｉ，２），…，ｅ（ｄｉ，Ｐ）が得られる。そこで、次のような手法により、ＭＩＤＩ形式の符号データを作成することができる。まず、「ノートオン」データもしくは「ノートオフ」データの中で記述するノートナンバーＮとしては、得られたノートナンバーｎ（ｄｉ，１），ｎ（ｄｉ，２），…，ｎ（ｄｉ，Ｐ）をそのまま用いていればよい。一方、「ノートオン」データもしくは　「ノートオフ」データの中で記述するベロシティーＶとしては、得られた実効強度ｅ　（ｄｉ，１），ｅ（ｄｉ，２），…，ｅ（ｄｉ，Ｐ）を、値が０～１の範囲となるように規格化し、この規格化後の実効強度Ｅの平方根に、たとえば１２７を乗じた値を用いるようにする。すなわち、実効強度Ｅについての最大値をＥmax とした場合、

$$V = \sqrt{(E / E_{max})} \cdot 127$$

なる演算で求まる値Ｖをベロシティーとして用いる。あるいは対数をとって、

$$V = \log(E / E_{max}) \cdot 127 + 127$$

（ただし、Ｖ＜０の場合はＶ＝０とする）

なる演算で求まる値Ｖをベロシティーとして用いてもよい。また、「デルタタイム」データは、各単位区間の長さに応じて設定すればよい。

【００５４】

結局、上述した実施形態では、３トラックからなるＭＩＤＩ符号データが得られることになる。このＭＩＤＩ符号データを３台のＭＩＤＩ音源を用いて再生すれば、６チャンネルのステレオ再生音として音響信号が再生される。

【００５５】

上述した図１および図２を用いた説明では、非常に単純な区間設定例を示したが、実際には、このような区間設定に基いて符号化を行った場合、再生時に、境界となる時刻において音の不連続が発生しやすい。したがって、実用上は、隣接する単位区間が時間軸上で部分的に重複するような区間設定を行うのが好ましい。

【００５６】
図３(a) は、このように部分的に重複する区間設定を行った例である。図示されている単位区間ｄ１～ｄ４は、いずれも部分的に重なっており、このような区間設定に基いて前述の処理を行うと、図３(b) の概念図に示されているような符号化が行われることになる。この例では、それぞれの単位区間の中心を基準位置として、各音符をそれぞれの基準位置に配置しているが、単位区間に対する相対的な基準位置は、必ずしも中心に設定する必要はない。図３(b) に示す概念図を図２(b) に示す概念図と比較すると、音符の密度が高まっていることがわかる。このように重複した区間設定を行うと、作成される符号データの数は増加することになるが、再生時に音の不連続が生じない自然な符号化が可能になる。

【００５７】
＜＜＜　§２．人間の声に対する音高シフト　＞＞＞
さて、§１で述べた技術を利用すれば、任意の音響信号をＭＩＤＩデータなどのデジタル符号データに変換することができるので、楽器の演奏音に限らず、人間の話し声や歌声を符号化することが可能であり、人間の声を、五線譜上の音符として表現することも可能である。もちろん、ＭＩＤＩ規格は、もともと楽器演奏の操作を記述するための符号化規格であるため、個々の符号は、基本的に、特定周波数の音（特定の鍵盤の音）が特定時間だけ持続する（特定時間だけ鳴る）ことを示しているにすぎない。したがって、符号化したＭＩＤＩデータを再生、すなわち、所定の音源を用いて演奏しても、元の人間の声がそのまま再生されるわけではない。ただ、楽器を使って人間のしゃべる声に似せた演奏を行うことができるので、エンターテインメントとして様々な利用形態が広がることになる。

【００５８】
そのような利用形態を考えた場合、ＭＩＤＩデータとして表現された人間の声について、音高を変化させたいという要求が生まれるのは当然である。たとえば、ＭＩＤＩデータとして表現された声についてその抑揚を変化させたいという場合や、ＭＩＤＩデータとして表現された声に節をつけて歌声を作りたいという場合、当該ＭＩＤＩデータに対して音高をシフトする処理が必要になる。このような音高シフトは、音楽でいう移調演奏に相当するものであり、基本的には、一律に周波数を上げ下げする処理によって行うことができる。

【００５９】
たとえば、図４(a) に示すような和音を考えてみよう。この和音は、３つの音符ｎ１，ｎ２，ｎ３から構成されている。本願明細書では、便宜上、個々の音符ｎ１，ｎ２，ｎ３についてのノートナンバーも、同じ記号ｎ１，ｎ２，ｎ３で表すことにする。ここで、ノートナンバーｎは、ＭＩＤＩ規格上で定義された０～１２７までの数値であり、半音の差をもった鍵盤（ピアノの白鍵および黒鍵）の番号を示す数字である。任意のノートナンバーｎをもつ音符に対して、ノートナンバー（ｎ＋１）をもつ音符は半音だけ高い音符を示し、ノートナンバー（ｎ－１）をもつ音符は半音だけ低い音符を示している。また、本願明細書では、ノートナンバーｎ１，ｎ２，ｎ３に対応する周波数をそれぞれｆ１，ｆ２，ｆ３と表すことにする。図４(a) に括弧書きで示すｆ１，ｆ２，ｆ３は、それぞれ各音符ｎ１，ｎ２，ｎ３に対応する周波数を示している。

【００６０】
ノートナンバーｎは、周波数ｆの対数値（２を底とする対数）に比例しており、周波数が２倍になるたびに、ノートナンバーは１２だけ増加し、周波数が１／２倍になるたびに、ノートナンバーは１２だけ減少する。西洋音楽でいう「１オクターブの音程」は、周波数が２倍となる関係にある音の差を意味しており、ノートナンバーでは１２の差に対応する。いわゆる「ドレミファソラシド」の音階において、「ミ」と「ファ」の間、および「シ」と「ド」の間のみが半音、その他は全音となっているため、１２-半音の差が１オクターブに相当する。

【００６１】
図４(b) は、図４(a) に示す３つの音符ｎ１，ｎ２，ｎ３を、それぞれ１オクターブず

つ上げることにより得られる新たな音符ｎ１＊，ｎ２＊，ｎ３＊を示している。すなわち、図４(a) に示す３つの音符ｎ１，ｎ２，ｎ３（いずれも「ラ」の音に対応する音符）が、図４(b) では、１オクターブ高い（１２-半音だけ高い）３つの音符ｎ１＊，ｎ２＊，ｎ３＊（これらも「ラ」の音に対応する音符）に置き換えられている。図に括弧書きで示すとおり、各音符ｎ１＊，ｎ２＊，ｎ３＊に対応する周波数はｆ１＊，ｆ２＊，ｆ３＊である。ここで、ノートナンバーの関係は、ｎ１＊＝ｎ１＋１２、ｎ２＊＝ｎ２＋１２、ｎ３＊＝ｎ３＋１２であるが、周波数の関係は、ｆ１＊＝ｆ１×２、ｆ２＊＝ｆ２×２、ｆ３＊＝ｆ３×２となる。

【００６２】

結局、図４(b) に示す和音は、図４(a) に示す和音を構成する音符を一律「１オクターブ」上げる（すなわち、周波数ｆとしては２倍にする、ノートナンバーｎとしては１２を加算する）ことによって得られる。

【００６３】

このように、一般的には、任意の和音について、音高を上げ下げする処理とは、当該和音を構成する個々の音符のノートナンバーに対して、一律に所定のオフセット値を加減算することによって行われる。上例のように、１オクターブだけ上げる音高シフトを行うのであれば、すべての音符のノートナンバーｎに対して、１オクターブの音程に相当するオフセット値「１２」を加える加算処理を行えばよい。

【００６４】

したがって、何らかの楽曲を表すＭＩＤＩデータに含まれているすべての符号について、そのノートナンバー値を１２だけ増加させる変換処理を行えば、変換後のＭＩＤＩデータは、元のＭＩＤＩデータに対して１オクターブ分の音高シフト処理を施した移調演奏用のデータということになる。そして、一般的には、そのような移調演奏用のＭＩＤＩデータを再生（所定の音源を用いて演奏）したとしても、不自然と思われるほどの違和感は生じない。これは、ＭＩＤＩ規格やピアノ鍵盤では平均律音階を採用しているためである。西洋音楽の音階「ドレミファソラシド」は、紀元前のピタゴラス音階が原点で、当初はこれを改良した純正律音階（構成音の周波数比が全て整数になる）が使用されていた。しかし、移調演奏を行うと、協和音が不協和音になるなど、メロディと和声の関係が崩れてしまうという問題があり、１７世紀にＪＳバッハらにより平均律音階が発明された。これは１オクターブ内の構成音を１２音に等比級数で分割する方法で、構成音の周波数比が全て$2^{-1/12}$を基本とした端数をもつ実数比になるため、協和音でも若干の唸り音が生じるという欠点はあるが、いかなる移調を施しても（半音単位にノートナンバーを上下させても）、メロディと和声の関係が常に維持されるという利点があり、以降の音楽は殆ど平均律音階に基づいて作曲されている。

【００６５】

ところが、既に述べたとおり、本願発明者が行った実験によると、人間の声を表現したＭＩＤＩデータについて、同様の方法で音高シフトを行うと、音声の明瞭性は維持されるものの、声質が破壊され、違和感のある奇怪な声になってしまうことが判明した。具体的には、プロのアナウンサーによって伝えられた天気予報のニュース番組を録音し、§１で述べた方法で符号化してＭＩＤＩデータを作成し、当該ＭＩＤＩデータに含まれているすべての符号について、そのノートナンバー値を１２だけ増加させる変換処理を行い、変換後のＭＩＤＩデータを再生したところ、個々の単語の判別は可能であるものの、いわゆるボイスチェンジャーを通したような違和感のある不自然な声になってしまった。

【００６６】

本願発明者は、そのような結果が得られる理由を探求するために種々の実験を行った結果、人の声に含まれているフォルマント成分に対して行った音高シフト処理が、違和感や不自然さの原因になっていることを見出した。以下、この原因について、具体例を挙げて説明する。

【００６７】

一般に、楽器音や人間が発声する音声には、基本周波数成分と、その整数倍の周波数を

もつ倍音成分が離散的に含まれており、特に音声の場合は、倍音成分の中で強度が前後の倍音に比べ大きなピークをもつ成分がみられ、これらの特別な倍音はフォルマント成分と命名されている。（これに対し、楽器音の場合は周波数が高い倍音ほど強度が単調に小さくなる傾向を示すのが一般的である。）フォルマント成分は基本周波数をＦ０と命名し、周波数が低い方から順にＦ１、Ｆ２、Ｆ３・・・と命名される。図５は、人間の声に含まれるフォルマント成分を例示する強度グラフである。このグラフは、横軸に周波数ｆをとり、縦軸に実効強度Ｅをとった、人間の音声スペクトルのピーク位置を示すグラフである。人間の声の周波数スペクトルは、かなり広い周波数域にわたって分布するが、ところどころの周波数位置にピークが存在する。図５は、このようなピークのみを抜き出して示したグラフであり、個々の周波数位置に示された垂直のバーは、当該周波数位置に当該バーの長さに相当する実効強度をもつピークが存在することを示している。ここで、太線のバーは、特に強度の大きなピークを示している。すなわち、図示の例の場合、周波数Ｆ０の位置に最大ピークが見られ、以下、周波数Ｆ１，Ｆ２，Ｆ３の位置のピークが続く。

【００６８】

ここで、基本周波数と一致する最小周波数に位置するフォルマントの周波数はＦ０と呼ばれており、この基本周波数Ｆ０をもった成分は、人間の声の最も主要な成分ということになる。これに対して、その他の太線バーで示されている周波数Ｆ１，Ｆ２，Ｆ３をもつ成分は、低い方から順に、それぞれ第１フォルマント周波数Ｆ１，第２フォルマント周波数Ｆ２，第３フォルマント周波数Ｆ３と呼ばれる。これら各フォルマント周波数Ｆ１，Ｆ２，Ｆ３は、基本周波数Ｆ０の整数倍の位置に現れることが知られている。基本周波数Ｆ０は、人間の声帯の本来の振動周波数に対応するものであるが、この基本周波数Ｆ０の整数倍をとる倍音系列のうち、声道共鳴によっていくつかの倍音が顕著なエネルギーピークをもつ。そして、このようにエネルギーピークをもつに至った倍音成分がフォルマントを形成するとされている。

【００６９】

図５に示す例の場合、たとえば、基本周波数Ｆ０＝１００Ｈｚ（声帯の振動周波数）とすると、他の細線のバーや太線のバーの位置は、いずれもその整数倍である２００Ｈｚ，３００Ｈｚ，...となっており、これらの倍音成分のうち、声道共鳴によって顕著なピークをもつに至った周波数Ｆ１（＝６００Ｈｚ），Ｆ２（＝１２００Ｈｚ），Ｆ３（＝１６００Ｈｚ）の成分が、フォルマントを構成することになる。

【００７０】

もちろん、このようなフォルマントの特性には個人差があり、どの倍音成分がフォルマントになるか、第３フォルマント周波数Ｆ３以上の高次フォルマントが生じるか否か、といった条件は、発声主である人によって千差万別である。もちろん、しゃべる言語によっても、その特性に変化が生じる。しかし、Ｆ０以外の高次のフォルマント周波数は言語ごとに母音ごとにほぼ一定の範囲に収まっているため、音声によるコミュニケーションが成立する。言語習得の過程において自ら発声する母国語の各母音のフォルマント周波数を所定の範囲に収めるように声道の筋肉が親や教師より訓練されている。（鳥も親鳥より同様な訓練がなされていることも知られている。）現在のところ、母音を発音したときのフォルマント特性についてはある程度の解析がなされているものの、子音を発音したときのフォルマント特性については未解明な部分が多い。ただ、いずれにせよ、このフォルマント成分が、人間の声の質に大きな影響を与える要素であることは事実である。

【００７１】

本願発明者は、このようなフォルマントによって特徴づけられる人間の声において、音高を変えるような発声を試みた場合、基本周波数Ｆ０は上下するものの、フォルマント周波数Ｆ１，Ｆ２，Ｆ３，...には、大きな変化が生じない傾向があることを実験により確認した。具体的には、同一人物に、同一の母音を、標準的な音高、高めの音高、低めの音高の３通りで発声してもらい、そのスペクトルをとり、図５のグラフに例示するようなピーク位置を求める実験を行った。図６(a)，(b)，(c)に示すグラフは、このような実験の一例を示す強度グラフである。すなわち、図６(a)は、標準的な音高で発声を行った音

声スペクトルの強度グラフ、図６(b) は、音高を高めて発声を行った場合の強度グラフ、図６(c) は、音高を低めて発声を行った場合の強度グラフである。

【００７２】

図６(a) ，(b) ，(c) において、破線は、基本周波数Ｆ０の倍音位置を示している。たとえば、図６(a) における基本周波数Ｆ０＝１００Ｈｚだとすると、図６(a) に示す破線は１００Ｈｚの間隔で引かれていることになり、この例の場合、第１フォルマント周波数Ｆ１＝６００Ｈｚ、第２フォルマント周波数Ｆ２＝１２００Ｈｚ、第３フォルマント周波数Ｆ３＝１６００Ｈｚである。一方、図６(b) における基本周波数Ｆ０＝２００Ｈｚだとすると、図６(b) に示す破線は２００Ｈｚの間隔で引かれていることになり、図６(c) における基本周波数Ｆ０＝５０Ｈｚだとすると、図６(c) に示す破線は５０Ｈｚの間隔で引かれていることになる。

【００７３】

このように、いずれの場合も、ピーク強度が得られる周波数（細線のバーや太線のバーで示す周波数）が、基本周波数Ｆ０の倍音位置（破線の位置）になる点に変わりはない。ただ、これらのピーク強度が得られる周波数（倍音系列の周波数）のうち、顕著な強度値をもつフォルマント周波数Ｆ１，Ｆ２，Ｆ３（太線のバー）に着目すると、基本周波数Ｆ０の変化にもかかわらず、全く変化していないことがわかる。

【００７４】

たとえば、図６(b) の場合、基本周波数Ｆ０は、１００Ｈｚから２００Ｈｚにシフトしている（発声者が高い声を出そうとしたため、声帯の振動周波数が高くなっている）。ところが、フォルマント周波数Ｆ１，Ｆ２，Ｆ３の位置は、６００Ｈｚ，１２００Ｈｚ，１６００Ｈｚであり、図６(a) の場合と全く変わりはない。同様に、図６(c) の場合、基本周波数Ｆ０は、１００Ｈｚから５０Ｈｚにシフトしている（発声者が低い声を出そうとしたため、声帯の振動周波数が低くなっている）。ところが、フォルマント周波数Ｆ１，Ｆ２，Ｆ３の位置は、６００Ｈｚ，１２００Ｈｚ，１６００Ｈｚであり、図６(a) の場合と全く変わりはない。

【００７５】

この図６に示す例は、説明の便宜上、最も単純な典型例を示すものであり、実際の実験では、必ずしもこのような結果が得られるわけではない。上述したとおり、フォルマント特性には個人差があり、基本周波数Ｆ０が変わることによって、新たなフォルマント成分が追加されたり、逆に一部のフォルマント成分が消失したりするケースもある。

【００７６】

また、図６に示す例では、図６(a) に示す基本周波数Ｆ０に対して、図６(b) に示す基本周波数Ｆ０はたまたま「正確に２倍」となっており、図６(c) に示す基本周波数Ｆ０はたまたま「正確に１／２倍」となっているため、破線で示す倍音位置に関して、図６(a) ，(b) ，(c) 間で整合性がとれ、フォルマント周波数Ｆ１，Ｆ２，Ｆ３の位置が、３つの図で完全に一致している。しかしながら、実際の実験では、基本周波数Ｆ０の値は、発声者の「標準的な声」，「高めの声」，「低めの声」といった感覚で決められる値であるから、当然ながら、「正確に２倍」や「正確に１／２倍」といった値にはならない。したがって、図に破線で示す倍音位置も、グラフ間で完全に整合性がとれるわけではない。

【００７７】

たとえば、図６(b) における基本周波数Ｆ０が、１９０Ｈｚであったとすると、図６(b) の破線の間隔（倍音位置）は１９０Ｈｚおきになるので、フォルマント周波数Ｆ１，Ｆ２，Ｆ３の位置は、理論的には、５７０Ｈｚ，１１４０Ｈｚ，１５２０Ｈｚとなり、図６(a) におけるフォルマント周波数Ｆ１，Ｆ２，Ｆ３の位置からは若干ずれることになる。

【００７８】

このように、実際に行った実験では、図６に示すような典型的な結果が得られるわけではない。しかしながら、同じ発声者が同じ母音を発音しながら、音高を上昇あるいは下降させると、基本周波数Ｆ０は、それに連動して上下し、新たな基本周波数Ｆ０の倍音系列を含む音が再構成されることは確かである。しかも、基本周波数Ｆ０をシフトさせても、

各フォルマント周波数Ｆ１，Ｆ２，Ｆ３の位置はそれほど大きくシフトせず、「高めの声」を出した場合でも、あるいは「低めの声」を出した場合でも、「標準的な声」を出した場合の各フォルマント周波数Ｆ１，Ｆ２，Ｆ３の近傍位置にそのままとどまることが確認できた。これは、基本周波数Ｆ０が声帯の振動によって定まる周波数であるのに対して、各フォルマント周波数Ｆ１，Ｆ２，Ｆ３は、声道での共鳴に依存して定まる周波数であるためと考えられる。

【００７９】

このような事実を踏まえれば、図６(a) に示すような基本周波数Ｆ０およびフォルマント周波数Ｆ１，Ｆ２，Ｆ３をもった人間の声を符号化したＭＩＤＩデータに対して、音高を１オクターブだけ上げる音高シフト処理を行うのであれば、基本周波数Ｆ０の音については２倍の周波数に変更するが、フォルマント周波数Ｆ１，Ｆ２，Ｆ３の音については周波数変更を一切行わない、という処理を行うのが好ましいことがわかる。同様に、音高を１オクターブだけ下げる音高シフト処理を行うのであれば、基本周波数Ｆ０の音については１／２倍の周波数に変更するが、フォルマント周波数Ｆ１，Ｆ２，Ｆ３の音については周波数変更を一切行わない、という処理を行うのが好ましいことがわかる。

【００８０】

結局、人間の声を表現したＭＩＤＩデータについて、一般の楽曲を表現したＭＩＤＩデータについての音高シフト方法をそのまま適用すると、ボイスチェンジャーを通したような違和感のある不自然な声になる理由は、すべての周波数の音に対して、一律に音高シフト処理を施すと、本来は音高シフトすべきではないフォルマント成分に対しても音高シフト処理が行われてしまうためであることがわかる。したがって、人間の声に対する音高シフトを行う際には、「フォルマント成分の絶対周波数値はほぼ一定」という人間の声音に固有の特徴を維持したまま、音高を変更する処理を行う必要がある。

【００８１】

＜＜＜　§３．本発明の基本概念　＞＞＞

§２で述べた人間の声の特性を考慮すれば、人間の声を表現したＭＩＤＩデータについての音高シフトを行う際には、基本周波数Ｆ０をもつ音には音高のシフト処理（周波数／ノートナンバーの増減処理）を行い、フォルマント周波数Ｆ１，Ｆ２，Ｆ３をもつ音には音高のシフト処理を行わない、という分別処理を行うとよいことがわかる。しかしながら、実際にそのような分別処理を行うことは非常に困難である。たとえば、図６(a) に示す例の場合、基本周波数Ｆ０は、最も大きな強度ピークをとる周波数であり、かつ、複数の強度ピークに対応する周波数の中で最も小さな周波数となっている。このような典型的な例の場合、基本周波数Ｆ０とフォルマント周波数Ｆ１，Ｆ２，Ｆ３とを弁別することは容易であり、基本周波数Ｆ０をもつ音に対してのみシフト処理を行うことができる。

【００８２】

しかしながら、図６(a) に示す例は、説明の便宜のために示した理想的な典型例であり、実際の人間の声に対して、§１で述べた符号化方法を適用することによって得られるＭＩＤＩデータの場合、必ずしもこの典型例のような結果が得られるわけではない。したがって、実用上は、基本周波数Ｆ０とフォルマント周波数Ｆ１，Ｆ２，Ｆ３とを簡単に弁別できるものではなく、実際に分別を行うとなると、高度な音声信号解析技術が必要になる。上述したとおり、フォルマント特性には個人差があり、しかも現在の技術では、子音についてのフォルマントに関する解明は十分になされていないため、一般的な人間の話し言葉や歌声について、最も大きなピーク、あるいは、最も周波数の小さなピーク、といった単純な方法で基本周波数Ｆ０を認識することは困難である。

【００８３】

そこで、本願発明者は、人間の声を表現した符号化音声データに対して、できるだけ簡便な方法により所望の音高シフト処理を施し、違和感のない自然な音声再生が可能な符号化音声データを得る方法を探るうちに、次のような着想を得ることができた。

【００８４】

いま、図７(a) に示すような重み関数Ｗ（ｆ）を定義してみる。この図７(a) のグラフ

は、横軸に周波数ｆ（線形スケール）、縦軸に関数値Ｗ（ｆ）をとり、周波数ｆ軸上の所定区間ｆａ～ｆｂにおいて関数値Ｗ（ｆ）が周波数ｆの増加に従って単調減少する関数を示している。より具体的に言えば、この重み関数Ｗ（ｆ）は、周波数ｆ軸上の第１設定値ｆａおよび第２設定値ｆｂ（ｆｂ＞ｆａ）について、ｆ≦ｆａの区間は、Ｗ（ｆ）＝１、ｆａ＜ｆ＜ｆｂの区間は、１＞Ｗ（ｆ）＞０（但し、Ｗ（ｆ）はｆの増加に従って単調減少）、ｆ≧ｆｂの区間は、Ｗ（ｆ）＝０となる関数である。

【００８５】

さて、ここで、ＭＩＤＩデータに対して、オフセット値αに応じた音高シフト処理を行う際に、このＭＩＤＩデータに含まれている個々の符号（音符）について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行うことにする。

【００８６】

たとえば、音高を高める場合は正、低める場合は負のオフセット値αを設定し、もとの周波数ｆに対して、所定の係数ｋ（ｋ＞１）を用いた式$f' = f \cdot k^{\alpha \cdot W(f)}$により新たな周波数ｆ′を求めればよい。

【００８７】

具体的には、α＝１に設定した場合、$f' = f \cdot k^{W(f)}$なる演算により新たな周波数ｆ′が求まる。この場合、図７のグラフにおいて、第１設定値ｆａ以下の周波数ｆに対しては、Ｗ（ｆ）＝１なる重み関数値が与えられるので、ｆ′＝ｋ・ｆとなり、周波数はｋ倍にシフトする（たとえば、ｋ＝２に設定しておけば、音程は１オクターブ上がることになる）。これに対して、第２設定値ｆｂ以上の周波数ｆに対しては、Ｗ（ｆ）＝０なる重み関数値が与えられるので、ｆ′＝ｆとなり、周波数は変化しない。第１設定値ｆａを超え、第２設定値ｆｂ未満の周波数ｆに対しては、その中間的な量（ｋ倍～１倍）の周波数シフトが行われることになる。

【００８８】

また、α＝－１に設定した場合も、図７のグラフにおいて、第１設定値ｆａ以下の周波数ｆに対しては、Ｗ（ｆ）＝１なる重み関数値が与えられるので、ｆ′＝ｆ／ｋとなり、周波数は１／ｋ倍にシフトする（たとえば、ｋ＝２に設定しておけば、音程は１オクターブ下がることになる）。これに対して、第２設定値ｆｂ以上の周波数ｆに対しては、Ｗ（ｆ）＝０なる重み関数値が与えられるので、ｆ′＝ｆとなり、周波数は変化しない。第１設定値ｆａを超え、第２設定値ｆｂ未満の周波数ｆに対しては、その中間的な量（１／ｋ倍～１倍）の周波数シフトが行われることになる。

【００８９】

このように、周波数ｆの増加に従って単調減少する重み関数Ｗ（ｆ）を用いて、符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減するようにすると、より高音の符号ほど音高シフト量が減少することになる。このため、人間の発生する音声についての音高シフトを行う場合、基本周波数成分の音高シフト量に比べて、当該基本周波数成分の整数倍の周波数をもつフォルマント成分についての音高シフト量が小さくなり、「フォルマント成分の絶対周波数値はほぼ一定」という人間の声の特徴をできるだけ維持したまま、音高を変更することができる。

【００９０】

ここで、第１設定値ｆａは、平均的な音声についての基本周波数成分（Ｆ０：０次フォルマント成分）が含まれると予想される周波数領域の上限に設定し、第２設定値ｆｂは、聴取可能な最高次フォルマントF^{max}成分が含まれると予想される周波数領域の上限に設定するのが好ましい。

【００９１】

具体的には、第１設定値ｆａは、音高シフト処理の対象となる音声が男性の声の場合、１００Ｈｚ～２００Ｈｚの範囲内に設定し、音高シフト処理の対象となる音声が女性の声の場合、２００Ｈｚ～４００Ｈｚの範囲内に設定するのが好ましい。これは、一般的な男

性の声の場合、その基本周波数成分が含まれると予想される周波数領域の上限は、概ね１００Ｈｚ～２００Ｈｚあたりとなり、一般的な女性の声の場合、その基本周波数成分が含まれると予想される周波数領域の上限は、概ね２００Ｈｚ～４００Ｈｚあたりとなるためである。

【００９２】

一方、第２設定値ｆｂは、音高シフト処理の対象となる音声が男性の声の場合、３ｋＨｚ～６ｋＨｚの範囲内に設定し、音高シフト処理の対象となる音声が女性の声の場合、４ｋＨｚ～８ｋＨｚの範囲内に設定するのが好ましい。これは、一般的な男性の声の場合、人間が聴取可能な最高次フォルマントF^{max}成分が含まれると予想される周波数領域の上限は、概ね３ｋＨｚ～６ｋＨｚあたりとなり、一般的な女性の声の場合、人間が聴取可能な最高次フォルマントF^{max}成分が含まれると予想される周波数領域の上限は、概ね４ｋＨｚ～８ｋＨｚあたりとなるためである。

【００９３】

既に述べたとおり、基本周波数Ｆ０や各フォルマント周波数Ｆ１，Ｆ２，Ｆ３が、周波数軸上のどこに位置するかという事項は、個人差や言語差によって大きく左右される事項であり、一概に決定することはできない。しかしながら、図７(a) に示すような重み関数Ｗ（ｆ）を設定し、周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減するようにすれば、「より高音の符号ほど音高シフト量が減少する」という一般的な傾向をもった音高シフト処理を行うことができるので、たとえ個人差や言語差によるバリエーションがあったとしても、かなり広いバリエーションをカバーした概括的な適用が可能になり、概ね良好な処理結果を得ることができる。

【００９４】

たとえば、ある人物については、その声の基本周波数Ｆ０が第１設定値ｆａを超えている場合もあろう。そのような場合でも、基本周波数Ｆ０が第２設定値ｆａに達しない限り（ｆａ＝３ｋＨｚ程度に設定しておけば、通常の話し言葉では、基本周波数Ｆ０が第２設定値ｆａに達することは考えにくい）、関数値Ｗ（ｆ）は０にはならないので、オフセット値αに応じたシフト量で、基本周波数Ｆ０に対する音高シフト処理が実行される。

【００９５】

また、多くの場合、いくつかのフォルマント周波数は、第１設定値ｆａ～第２設定値ｆｂの範囲内に位置することになるであろう。それでも、これらフォルマント周波数に対する音高シフト量は、基本周波数Ｆ０に対する音高シフト量よりも必ず小さくなるので、音高シフト処理後の音声をできるだけ違和感のない自然な音声に近づけるという効果は少なからず得られることになる。

【００９６】

実際、本願発明者が試した多くの事例では、上述したように、第１設定値ｆａを、男性の声の場合には、１００Ｈｚ～２００Ｈｚの範囲内に設定し、女性の声の場合には、２００Ｈｚ～４００Ｈｚの範囲内に設定し、第２設定値ｆｂを、男性の声の場合には、３ｋＨｚ～６ｋＨｚの範囲内に設定し、女性の声の場合には、４ｋＨｚ～８ｋＨｚの範囲内に設定すれば、ほぼすべてのケースについて、違和感のない自然な音声再生が可能な符号化音声データを得るという本発明に特有の作用効果を得ることができた。

【００９７】

＜＜＜　§４．ノートナンバーの増減による音高シフト　＞＞＞

§３では、本発明の基本概念を、「周波数をシフトする」という物理的な事象の観点から述べたが、ＭＩＤＩ規格などの符号化音声データの場合、音高は周波数ｆではなく、ノートナンバーｎによって表現される。したがって、実用上は、「周波数をシフトする」処理の代わりに、「ノートナンバーをシフトする」処理を行う必要がある。そこで、以下、このような実用上の観点から本発明の基本概念を説明する。

【００９８】

既に述べたとおり、ノートナンバーｎは、周波数ｆの対数値に比例しており、周波数が２倍になるたびに、ノートナンバーは１２だけ増加し、周波数が１／２倍になるたびに、

ノートナンバーは１２だけ減少する関係にある。したがって、周波数ｆの増減によって音高シフトを行う代わりに、ノートナンバーの増減によって音高シフトを行う場合、重み関数も、周波数ｆの関数Ｗ（ｆ）の代わりに、ノートナンバーｎの関数Ｗ（ｎ）を用いる必要がある。

【００９９】

図７(b) は、このようなノートナンバーｎについての重み関数Ｗ（ｎ）の一例を示すグラフである。この図７(b) のグラフは、横軸がノートナンバーｎ、すなわち、周波数ｆについての対数スケールとなっているため、図７(a) のグラフと形状は異なっているが、実際には、関数Ｗ（ｆ）と関数Ｗ（ｎ）とは等価な関数である。関数Ｗ（ｎ）は、ノートナンバーｎの座標軸上に設定された第１設定値ｎａ，第２設定値ｎｂについて、ｎ≦ｎａの区間は、Ｗ（ｎ）＝１、ｎａ＜ｎ＜ｎｂの区間は、１＞Ｗ（ｎ）＞０（但し、Ｗ（ｎ）はｎの増加に従って単調減少）、ｎ≧ｎｂの区間は、Ｗ（ｎ）＝０となる関数である。特に、この図７(b) に示す例の場合、ｎａ＜ｎ＜ｎｂの区間において、Ｗ（ｎ）の値は、ノートナンバーｎに反比例する値となっている。

【０１００】

ここで、図７(b) に示すノートナンバーについての第１設定値ｎａ，第２設定値ｎｂは、それぞれ図７(a) に示す周波数についての第１設定値ｆａ，第２設定値ｆｂに対応するものである。別言すれば、周波数ｆａはノートナンバーｎａに対応し、周波数ｆｂはノートナンバーｎｂに対応する。結局、図７(b) に示す重み関数Ｗ（ｎ）のグラフは、図７(a) に示す重み関数Ｗ（ｆ）のグラフの周波数軸を対数スケールに修正したものに相当する。

【０１０１】

なお、図７(b) に示す例において、ｎａ＜ｎ＜ｎｂの区間についてのＷ（ｎ）の値が、ノートナンバーｎに反比例する値となるようにしているのは、演算負担を軽減するための配慮である。この図７(b) に示すような重み関数Ｗ（ｎ）を用いるようにすれば、ｎ≦ｎａの区間は、Ｗ（ｎ）＝１、ｎ≧ｎｂの区間は、Ｗ（ｎ）＝０、そしてｎａ＜ｎ＜ｎｂの区間は、Ｗ（ｎ）＝ｋ／ｎ（ｋは比例定数）となり、関数値Ｗ（ｎ）を求める演算負担は極めて軽くなる。もちろん、Ｗ（ｎ）の値は、ｎａ＜ｎ＜ｎｂの区間について、ｎの増加に従って単調減少すればよいので、必ずしも反比例するような値に設定する必要はない。

【０１０２】

ＭＩＤＩデータに対する音高シフト処理は、§２で述べたとおり、ＭＩＤＩ符号に含まれるノートナンバーｎに対して、オフセット値αを加算もしくは減算することによって行われる。すなわち、１オクターブだけ上げる音高シフトを行うのであれば、ノートナンバーｎに対して、１オクターブの音程に相当するオフセット値「１２」を加える加算処理を行い、１オクターブだけ下げる音高シフトを行うのであれば、ノートナンバーｎに対して、１オクターブの音程に相当するオフセット値「１２」を減じる減算処理を行えばよい。

【０１０３】

そこで、ここでは、具体例を参照しながら、従来の方法による音高シフト処理と、本発明による音高シフト処理との相違を説明しよう。いま、図８(a) に示すように、変換前のＭＩＤＩデータに、時間軸上で同一期間を占める３つの符号が含まれていたものとしよう。図８(a) では、この３つのＭＩＤＩ符号を音符により表している。実際のＭＩＤＩ符号には、前述したとおり、「ノートナンバー」、「ベロシティー」、「ノートオン」、「ノートオフ」、「デルタタイム」などのデータが含まれるが、音高シフト処理で変更されるデータは、周波数を示す「ノートナンバー」のみであるから、以下、便宜上、ＭＩＤＩ符号を音符で表現した説明を行うことにする。

【０１０４】

図８(a) に示す例の場合、音高シフト処理の対象となる音は、３つの音符ｎ１，ｎ２，ｎ３の和音として構成される音である。これらの各音符ｎ１，ｎ２，ｎ３は、それぞれ同じ記号で示すノートナンバーｎ１，ｎ２，ｎ３をもった音を示し、これらのノートナンバーは、それぞれ括弧書きで示す周波数ｆ１，ｆ２，ｆ３に対応するものとする。

【０１０５】
§１で述べた方法で、人間の声をＭＩＤＩデータによって表すと、図３(b) に示す例のように、複数のトラックにそれぞれ符号が得られることになり、これら複数トラックに分布している符号を五線譜上に音符として表現すると、図８(a) に例示するような和音が形成されることになる。要するに、人間の声を表したＭＩＤＩデータは、音の持続時間が時間軸上で同一期間を占め、互いに異なる周波数を示す複数の符号によって構成され、五線譜上では和音の形式で表されることになる。
【０１０６】
さて、このような和音に対して、従来の方法による音高シフト処理を行う場合、§２で述べたとおり、和音を構成する個々の音符に対して、一律、同じシフト量だけノートナンバーを増減する処理が施される。たとえば、１オクターブだけ上げる処理を行う場合、すべての音符ｎ１，ｎ２，ｎ３のノートナンバーに、それぞれ一律して、１オクターブの音程に相当するオフセット値「１２」を加える加算処理が行われる。図８(b) は、このような音高シフト処理を施した後に得られる新たな音符ｎ１＊，ｎ２＊，ｎ３＊を示している。ここで、各音符ｎ１＊，ｎ２＊，ｎ３＊のノートナンバーを同じ記号ｎ１＊，ｎ２＊，ｎ３＊で示せば、ｎ１＊＝ｎ１＋１２，ｎ２＊＝ｎ２＋１２，ｎ２＊＝ｎ２＋１２が成り立つ。また、シフト処理後の各ノートナンバーに対応する周波数をｆ１＊，ｆ２＊，ｆ３＊とすれば、ｆ１＊＝ｆ１×２，ｆ２＊＝ｆ２×２，ｆ３＊＝ｆ３×２が成り立ち、いずれの音符についても一律に周波数を２倍にする処理が行われていることになる。
【０１０７】
このように、すべての音符のノートナンバーについて一律に同じシフト量を増減する音高シフトは、楽器音を想定した一般的な楽曲に対して行う場合は問題ないが、人間の声に対して行うと、基本周波数成分のみならず、フォルマント成分に対しても同じシフト量の音高シフトが行われてしまうため、問題が生じることは既に述べたとおりである。
【０１０８】
そこで、本発明の方法による音高シフト処理では、オフセット値αに対して、図７(b) に示す重み関数Ｗ（ｎ）を乗じることにより、ノートナンバーの大きな音符ほど、シフト量が小さくなるような調整を行うことになる。具体的には、「ｎ′＝ｎ＋α・Ｗ（ｎ）」なる演算式により新たなノートナンバーｎ′を求めることになる。
【０１０９】
図８(c) は、本発明に係る音高シフト処理を施した後に得られる新たな音符ｎ１＊，ｎ２＊，ｎ３＊を示している。ここで、各音符ｎ１＊，ｎ２＊，ｎ３＊のノートナンバーを同じ記号ｎ１＊，ｎ２＊，ｎ３＊で示せば、ｎ１＊＝ｎ１＋１２，ｎ２＊＝ｎ２＋６，ｎ３＊＝ｎ３＋３となっており、シフト量は音符ごとに異なる。これは、オフセット値α＝１２を与えたとしても、重み関数値が、Ｗ（ｎ１）＝１，Ｗ（ｎ２）＝０．５，Ｗ（ｎ３）＝０．２５と、ノートナンバーの増加にともなって徐々に減少してゆくため、「ｎ′＝ｎ＋α・Ｗ（ｎ）」なる演算式により新たなノートナンバーｎ′を求めると、シフト量「α・Ｗ（ｎ）」は、１２，６，３と徐々に減少してゆくためである。シフト処理後の各ノートナンバーに対応する周波数をｆ１＊，ｆ２＊，ｆ３＊とすれば、ｆ１＊＝ｆ１×２，ｆ２＊　ｆ２×１．４１，ｆ３＊　ｆ３×１．１９になる。
【０１１０】
このように、図８(a) のような和音として表される人間の声のＭＩＤＩデータに対して、１オクターブ上げる音高シフト処理を行う場合、本発明による方法を適用すれば、図８(c) のような和音に対応するＭＩＤＩデータが得られることになる。この場合、基本周波数成分に対応する可能性の高い音符ｎ１については、１オクターブの音程に相当するオフセット値「１２」のシフト量がそのまま加算され、音符ｎ１＊に変換されることになるが、フォルマント成分に対応する可能性の高い音符ｎ２，ｎ３についてのシフト量はより小さくなる。このため、人間の声のＭＩＤＩデータに対して、できるだけ違和感のない自然な音高シフト処理が可能になる。
【０１１１】

図９は、１オクターブ下げる音高シフト処理を示す例である。図９(a) に示すような３つの音符ｎ１，ｎ２，ｎ３の和音として構成される音に対して、従来の方法による音高シフト処理を行い、１オクターブだけ下げる処理を行うと、図９(b) に示す音符ｎ１＊，ｎ２＊，ｎ３＊が得られる。ここで、新たに得られた音符のノートナンバーｎ１＊，ｎ２＊，ｎ３＊は、ｎ１＊＝ｎ１－１２，ｎ２＊＝ｎ２－１２，ｎ３＊＝ｎ３－１２で表される。また、これらの各ノートナンバーに対応する周波数をｆ１＊，ｆ２＊，ｆ３＊とすれば、ｆ１＊＝ｆ１／２，ｆ２＊＝ｆ２／２，ｆ３＊＝ｆ３／２が成り立ち、いずれの音符についても一律に周波数を１／２倍にする処理が行われていることになる。

【０１１２】

これに対して、図９(c) は、本発明に係る音高シフト処理を施した後に得られる新たな音符ｎ１＊，ｎ２＊，ｎ３＊を示している。ここで、各音符のノートナンバーｎ１＊，ｎ２＊，ｎ３＊は、ｎ１＊＝ｎ１－１２，ｎ２＊＝ｎ２－６，ｎ３＊＝ｎ３－３となっており、やはりシフト量は音符ごとに異なる。シフト処理後の各ノートナンバーに対応する周波数をｆ１＊，ｆ２＊，ｆ３＊とすれば、ｆ１＊＝ｆ１／２，ｆ２＊　ｆ２／１．４１，ｆ３＊　ｆ３／１．１９になる。

【０１１３】

このように、図９(a) のような和音として表される人間の声のＭＩＤＩデータに対して、１オクターブ下げる音高シフト処理を行う場合、本発明による方法を適用すれば、図９(c) のような和音に対応するＭＩＤＩデータが得られることになる。この場合、基本周波数成分に対応する可能性の高い音符ｎ１については、１オクターブの音程に相当するオフセット値「１２」のシフト量がそのまま減算され、音符ｎ１＊に変換されることになるが、フォルマント成分に対応する可能性の高い音符ｎ２，ｎ３についてのシフト量はより小さくなる。このため、人間の声のＭＩＤＩデータに対して、違和感のない自然な音高シフト処理が可能になる。

【０１１４】

ところで、和音を構成する複数の音符に対して、それぞれ異なるシフト量をもった音高シフトを行うと、複数の音符がシフト後に同じ音高を占める場合がありうる。ＭＩＤＩ規格では、このように同じ音高を占める符号が同一チャンネルにおいて、同一時刻に重複して存在することは許されない。ただし、最大１６種類定義可能なチャンネル番号を変えれば重複が認められる。例えば、ピアノ音をチャンネル０、バイオリン音をチャンネル１に設定すれば、これらの楽器音を同一音高で同一時刻に演奏させることができる。しかし、本願では音声という単一の音色を扱うため、以下は単一のチャンネル番号を使用するという前提で説明する。このように、単一のチャンネル番号を使用する場合、ＭＩＤＩデータに対して、本発明に係る音高シフト処理を行う際には、処理後に同じ音高の符号が重複した場合に、１つのみを残して他を削除する処理を行うようにすればよい。

【０１１５】

図１０は、本発明に係る音高シフト処理を実行すると、複数の音符がシフト後に同じ音高を占めるケースを示す図である。図１０(a) は、変換前の符号化データに含まれる和音を示し、図１０(b) は、従来の方法で音高を１オクターブ上げる処理を行った結果を示し、図１０(c) は、本発明の方法で音高を１オクターブ上げる処理を行った結果を示す。従来の方法では、図１０(b) に示すように、全符号のノートナンバーに対して一律に同じオフセット値「１２」が加算される。このため、３つの音符の相互位置関係はそのまま維持され、全体が一群となって平行移動する。ところが、本発明の方法では、３つの音符はそれぞれシフト量が異なるため、図１０(c) に示すように、シフト後の３つの音符の位置は重なってしまう。

【０１１６】

すなわち、図１０(a) に示す音符ｎ１，ｎ２，ｎ３に対するシフト量は、それぞれ１２，６，３になり、各音符を当該シフト量に応じてシフトすると、本発明に係る音高シフト処理を施した後に得られる新たな音符ｎ１＊，ｎ２＊，ｎ３＊の位置は、図１０(c) に示すとおり重なってしまう。別言すれば、ノートナンバーｎ１，ｎ２，ｎ３に、それぞれシ

フト量１２，６，３を加えると、偶然、同じ値になってしまう。

【０１１７】

ＭＩＤＩ規格では、このように同じ音高をもつ複数の符号が重複することは許されない。そこで、このような場合、図１０(c) に重複して示されている新たな音符ｎ１＊，ｎ２＊，ｎ３＊のうちの１つのみを残し、その他の２個を削除する重複回避処理を行うようにすれば、重複を許さないＭＩＤＩ規格のような符号化データに対しても本発明を問題なく適用できる。すなわち、図１０(c) の場合、データとしては、３つの音符ｎ１＊，ｎ２＊，ｎ３＊が存在しているので、たとえば、音符ｎ１＊のデータのみを残し、音符ｎ２＊，ｎ３＊のデータを削除する処理を行えばよい。

【０１１８】

図１１は、このような重複回避処理を伴う音高シフトの別な例を示す図である。ここでは、図１１(a) に示すとおり、５つの音符ｎ１～ｎ５からなる和音について、本発明に係る方法で１オクターブ上げる音高シフトを行った例が示されている。この例では、オフセット値α＝１２であるが、各音符のノートナンバーｎに応じて定まる重み関数値Ｗ（ｎ）は、ノートナンバーｎの増加に応じて徐々に減少してゆくため、各音符のシフト量は図示のとおり、それぞれ１２，９，７，４，２となっている。

【０１１９】

図１１(b) は、これらのシフト量に従って、各音符ｎ１～ｎ５をシフトすることによって得られる新たな音符ｎ１＊～ｎ５＊を示している（破線の音符は、図１１(a) に示す音符ｎ１～ｎ５である）。なお、図示の便宜上、シフト後の５つの音符を左右２列に分けて描いているが、これら５つの音符ｎ１＊～ｎ５＊は、本来は、時間軸上の同一位置に配置されるべき音符である。図示のとおり、これら５つの音符のうちの一部は、音高変更後に同じ音高を占めている。すなわち、音符ｎ１＊と音符ｎ２＊とは同じ音高を占め、音符ｎ３＊と音符ｎ４＊とは同じ音高を占めている。

【０１２０】

そこで、重複した複数の音符については、１つのみを残して他を削除する重複回避処理を行う。図１１(c) は、このような重複回避処理後の結果を示している。この例では、括弧書きで示した音符ｎ２＊と音符ｎ４＊とが削除されており、結局、音高変更後の音は、３つの音符ｎ１＊，ｎ３＊，ｎ５＊の和音として表されている。

【０１２１】

なお、このような重複回避処理を行う場合、一部の符号の削除により、和音全体の強度は低下してしまう。すなわち、音高シフトを行う前は、図１１(a) に示すように、５つの音符ｎ１～ｎ５の和音によって構成されていた音が、重複回避処理を伴う音高シフトを行った後は、図１１(c) に示すように、３つの音符ｎ１＊，ｎ３＊，ｎ５＊の和音によって構成される音になってしまう。そこで、音の強度を維持する上では、重複回避処理を行う際に、１つのみ残された符号についての強度を、削除された符号についての強度に応じて修正するのが好ましい。

【０１２２】

ＭＩＤＩ符号の場合、音の強度の情報は、「ベロシティー」というデータとして符号に含まれているので、上例の場合、残された音符ｎ１＊についてのベロシティーに、削除された音符ｎ２＊についてのベロシティーを加える修正処理を施し、同様に、残された音符ｎ３＊についてのベロシティーに、削除された音符ｎ４＊についてのベロシティーを加える修正処理を施せばよい。このように、削除された符号についての強度に応じて強度修正を行えば、音高の変更処理後も適切な強度バランスをもった符号化データが得られる。

【０１２３】

＜＜＜　§５．本発明に係る符号化音声データの音高変換装置　＞＞＞

図１２は、本発明の基本的実施形態に係る符号化音声データの音高変換装置１００の構成を示すブロック図である。この装置は、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成され人間の声を表現する符号化音声データＤ（たとえば、ＭＩＤＩデータ）を、異なる音高をもった別な音声データＤ＊に変換する

符号化音声データの音高変換装置であり、その基本原理は、既に§３，§４で述べたとおりである。

【０１２４】

図示のとおり、この音高変換装置１００は、変換対象データ入力ユニット１１０、オフセット値入力ユニット１２０、音高変換処理ユニット１３０、重み関数格納ユニット１４０によって構成される。変換対象データ入力ユニット１１０は、変換対象となる符号化音声データＤを入力する構成要素であり、オフセット値入力ユニット１２０は音高に関するオフセット値αを入力する構成要素である。また、音高変換処理ユニット１３０は、変換対象となる符号化音声データＤに対して、オフセット値αに基づく音高の変更処理を行い、変更後の符号化音声データＤ＊を出力する構成要素であり、重み関数格納ユニット１４０は、周波数ｆについて定義された所定の重み関数Ｗ（ｆ）を格納する構成要素である。

【０１２５】

音高変換処理ユニット１３０は、§３で述べた基本原理に基づいて、音高変換処理（音高シフト処理）を行う。具体的には、重み関数格納ユニット１４０に格納されている重み関数Ｗ（ｆ）を用いて、変換対象となる符号化音声データＤに含まれている個々の符号について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行う。

【０１２６】

ここで、重み関数格納ユニット１４０に格納されている重み関数Ｗ（ｆ）は、図７(a)に例示するように、周波数ｆ軸上の所定区間においてＷ（ｆ）が周波数ｆの増加に従って単調減少する関数である。より具体的には、§３で述べたとおり、周波数ｆ軸上の第１設定値ｆａおよび第２設定値ｆｂ（ｆｂ＞ｆａ）について、ｆ≦ｆａの区間は、Ｗ（ｆ）＝１、ｆａ＜ｆ＜ｆｂの区間は、１＞Ｗ（ｆ）＞０（但し、Ｗ（ｆ）はｆの増加に従って単調減少、たとえば、周波数ｆの対数値に対して反比例）、ｆ≧ｆｂの区間は、Ｗ（ｆ）＝０となる重み関数Ｗ（ｆ）が用いられている。

【０１２７】

第１設定値ｆａは、平均的な音声についての基本周波数成分が含まれると予想される周波数領域の上限に設定し、第２設定値ｆｂは、第１設定値ｆａに対して十分に高い任意の値に設定するのが好ましい。実際には、基本周波数成分が含まれると予想される周波数領域は男女でかなりの差があるので、実用上は、重み関数格納ユニット１４０には、男性用重み関数Ｗ-male（ｆ）と女性用重み関数Ｗ-female（ｆ）とを用意しておき、変換対象となる符号化音声データＤの声主の性別により、いずれか一方の重み関数を選択して利用するのが好ましい。

【０１２８】

すなわち、音高変換処理ユニット１３０は、変換対象となる符号化音声データＤによって表現される声が男性の声か女性の声かを示すオペレータからの指示に基づいて、男性の声の場合には男性用重み関数Ｗ-male（ｆ）を用いた変更処理を行い、女性の声の場合には女性用重み関数Ｗ-female（ｆ）を用いた変更処理を行うことになる。

【０１２９】

本願発明者が行った実験によれば、男性用重み関数Ｗ-male（ｆ）については、第１設定値ｆａを１００Ｈｚ～２００Ｈｚの範囲内に設定し、第２設定値ｆｂを３ｋＨｚ～６ｋＨｚの範囲内に設定すれば、良好な結果が得られた。また、女性用重み関数Ｗ-female（ｆ）については、第１設定値ｆａを２００Ｈｚ～４００Ｈｚの範囲内に設定し、第２設定値ｆｂを４ｋＨｚ～８ｋＨｚの範囲内に設定すれば、良好な結果が得られた。

【０１３０】

もちろん、必要に応じて、子供用重み関数を用意したり、年齢別の重み関数を用意したり、言語ごとに異なる重み関数を用意してもよい。

【０１３１】

オフセット値入力ユニット１２０が入力するオフセット値αは、符号化音声データＤに

対する音高の変更指示を示す値であり、音高をどれだけ上げるのか、もしくは下げるのか、を示すことができる値であれば、どのように定義された値であってもかまわないが、一般的には、符号をもった数とし、音高を高める場合は正、低める場合は負の値をとり、絶対値が大きいほど、音高の変更量が大きいことを示す値として定義するのが好ましい。

【０１３２】

重み関数Ｗ（ｆ）の値域を「１≧Ｗ（ｆ）≧０」とし、オフセット値αを、上例のように定義した場合、音高変換処理ユニット１３０は、所定の係数ｋ（ｋ＞１）を用いた式ｆ′＝ｆ・ｋ$^{\alpha \cdot W(f)}$により新たな周波数ｆ′を求めることができる。αが正負いずれの場合も、Ｗ（ｆ）＝０の場合は、ｆ′＝ｆとなり、周波数の変更は生じないので、第２設定値ｆｂ以上の周波数をもつ符号に対しては、音高のシフト処理は行われないことになる。また、Ｗ（ｆ）＝１の場合は、ｆ′＝ｆ・ｋ$^{\alpha}$となり、オフセット値αに対応するシフト量をもった周波数変更が行われる（α＝＋１であれば、周波数はｋ倍になり、α＝－１であれば、周波数は１／ｋ倍になる）。また、１＞Ｗ（ｆ）＞０の場合には、周波数ｆが大きいほど、周波数変更のシフト量は小さくなる。

【０１３３】

なお、§４で述べたとおり、ＭＩＤＩデータのように、周波数ｆをその対数値に対応するノートナンバーｎで表す符号データを用いる場合、すなわち、変換対象データ入力ユニット１１０が入力した符号化音声データＤが、周波数ｆをノートナンバーｎによって示す符号を含むデータである場合、音高のシフト処理は、実際には、ノートナンバーの増減によって行うことになる。この場合、重み関数格納ユニット１４０には、図７(b) に例示するように、ノートナンバーｎについて定義された重み関数Ｗ（ｎ）を格納しておけばよい。図７(b) に示す例は、周波数ｆ軸上の第１設定値ｆａに対応するノートナンバーｎａおよび第２設定値ｆｂに対応するノートナンバーｎｂについて、ｎａ＜ｎ＜ｎｂの区間は、Ｗ（ｎ）の値がノートナンバーｎに反比例する値となる重み関数Ｗ（ｎ）の例である。このような重み関数Ｗ（ｎ）を用いると、前述したとおり、演算負担は大幅に軽減される。

【０１３４】

一方、オフセット値入力ユニット１２０によって入力されるオフセット値αとしては、音高を高める場合は正、低める場合は負の値をとるノートナンバーの差を用いるようにすればよい。たとえば、音高を１オクターブ上げる場合は、α＝＋１２、音高を１オクターブ下げる場合は、α＝－１２というオフセット値を与えればよい。

【０１３５】

また、音高変換処理ユニット１３０は、変換対象となる符号化音声データＤに含まれる個々の符号について、当該符号が示すノートナンバーｎを用いた「ｎ′＝ｎ＋α・Ｗ（ｎ）」なる演算式により新たなノートナンバーｎ′を求め、当該符号を、それが示すノートナンバーｎをｎ′に変更した新たな符号に置き換える処理を行うことになる。たとえば、α＝＋１２であった場合（１オクターブ上げる指示）、新たなノートナンバーｎ′は、「ｎ′＝ｎ＋１２・Ｗ（ｎ）」で与えられ、α＝－１２であった場合（１オクターブ下げる指示）、「ｎ′＝ｎ－１２・Ｗ（ｎ）」で与えられる。

【０１３６】

§１で述べた方法により人間の音声を符号化したＭＩＤＩデータは、音の持続時間が時間軸上で同一期間を占め、互いに異なる周波数を示す複数の符号を含む符号化音声データであり、いわば複数の音符からなる和音として音声を表現したデータである。したがって、変換対象データ入力ユニット１１０が、符号化音声データＤとしてＭＩＤＩ規格のデータを入力した場合、音高変換処理ユニット１３０が行う処理は、和音を構成する複数の符号のそれぞれについて、新たな符号への置換を行う処理ということになる。

【０１３７】

既に述べたとおり、ＭＩＤＩ規格では、同じ音高をもつ複数の符号が重複して存在することは許されないので、音高変換処理ユニット１３０は、音の持続時間が時間軸上で同一期間を占める複数の符号についてそれぞれ新たな符号への置換処理を行う際に、同一のノートナンバーを示す新たな符号が複数ｍ個生じた場合には、当該複数ｍ個の符号のうち１

つのみを残し、その余の（ｍ－１）個を削除する重複回避処理を行う機能を有している。また、ＭＩＤＩデータには、音の強度の情報（ベロシティー）が含まれているので、音高変換処理ユニット１３０は、重複回避処理を行う際に、１つのみ残された符号についての強度を、削除された符号についての強度に応じて修正する処理を行うようにするのが好ましい。

【０１３８】

なお、変換対象データ入力ユニット１１０によって入力された符号化音声データＤがＭＩＤＩデータであった場合、音高変換処理ユニット１３０から出力される符号化音声データＤ＊もＭＩＤＩ規格のデータとすれば、この音高変換装置１００は、符号化音声データの規格は維持したまま、音高のみを変換する処理を行うことができる。すなわち、ユーザは、所望のＭＩＤＩデータＤをこの音高変換装置１００に与え、所望のオフセット値αを指定する操作を行えば、音高変換処理が施されたＭＩＤＩデータＤ＊を得ることができる。

【０１３９】

こうして得られたＭＩＤＩデータＤ＊は、もちろん、電子楽器による演奏の対象とすることができ、音高変換後の人間の音声を再生することができる。また、音高変換処理ユニット１３０に、符号化音声データＤ＊を、五線譜上に音符を配置した楽譜のデータとして出力する機能をもたせておけば、音高変換後の人間の音声を表現した楽譜を得ることができ、実際の楽器を用いて、この楽譜上の音声情報を演奏して再生することもできる。

【０１４０】

＜＜＜　§６．本発明に係る音声変換装置　＞＞＞

ここでは、§５で述べた本発明に係る音高変換装置１００を利用して、音声変換装置を構成した例を、図１３のブロック図を参照しながら説明する。図１３に示す音声変換装置は、図１２に示す音高変換装置１００に、音声符号化装置２００を更に付加することによって構成される装置である。

【０１４１】

図示のとおり、音声符号化装置２００は、音声信号入力ユニット２１０と符号化ユニット２２０とによって構成される。ここで、音声信号入力ユニット２１０は、人間の声を含む音声信号Ｓをアナログ信号もしくはデジタル信号として入力する構成要素である。図には、オーディオＣＤ再生器や一般のパソコンなどのオーディオ機器１０から、デジタル音声信号Ｓを取り込む例と、マイク２０で集音してアンプ装置３０で増幅したアナログ音声信号Ｓを取り込む例とが示されている。

【０１４２】

一方、符号化ユニット２２０は、こうして取り込んだ音声信号Ｓを、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成された符号化音声データＤに変換する処理を行う構成要素である。具体的には、§１で述べた方法によって、人間の音声をＭＩＤＩ符号化する装置（たとえば、前掲の特許文献１～３に開示されている装置）によって、符号化ユニット２２０を構成することができる。

【０１４３】

こうして、符号化ユニット２２０から出力された符号化音声データＤ（たとえば、ＭＩＤＩデータ）は、音高変換装置１００に与えられる。この音高変換装置１００は、図１２に示す構成要素であり、その機能は既に§５で述べたとおりである。すなわち、所望のオフセット値αを与えることにより、入力された符号化音声データＤに対して音高変更処理を施し、符号化音声データＤ＊を出力する機能を有している。

【０１４４】

結局、この音声変換装置を利用すると、オーディオ機器１０によって再生した歌手の歌声を、デジタル音声信号Ｓとして与えることにより、当該歌声を表現したＭＩＤＩ形式等の符号化音声データＤ＊を得ることができ、しかもその音高を、与えるオフセット値αによって任意に変化させることができる。あるいは、マイク２０でリアルタイムに集音した人間の音声を、その場でＭＩＤＩ形式等の符号化音声データＤ＊に変換することができ、

しかもその音高を、与えるオフセット値αによって任意に変化させることができる。要するに、入力した人間の音声の音高を自由に変更した上で、符号化音声データＤ＊として出力することができる。

【０１４５】

もちろん、こうして出力された符号化音声データＤ＊は、ＭＩＤＩ音源等を利用して即座に再生することが可能であり、楽譜として出力すれば、楽器を用いて演奏することが可能である。

【０１４６】

＜＜＜　§７．本発明に係る音声合成装置　＞＞＞

ここでは、§５で述べた本発明に係る音高変換装置１００を利用して、音声合成装置を構成した例を、図１４のブロック図を参照しながら説明する。図１４に示す音声合成装置は、図１２に示す音高変換装置１００に、テキスト符号化装置３００を更に付加することによって構成される装置である。このテキスト符号化装置３００の詳細な構成や動作は、前掲の特許文献４に記載されているが、ここではその原理だけを簡単に述べておく。

【０１４７】

図示のとおり、テキスト符号化装置３００は、符号データベースユニット３１０と、テキストデータ入力ユニット３２０と、符号合成ユニット３３０と、によって構成される。ここで、符号データベースユニット３１０は、所定の言語による単語を構成する個々の音節にそれぞれ対応する符号群（特定周波数の音が特定時間だけ持続することを示す符号の集合体）を格納した構成要素であり、テキストデータ入力ユニット３２０は、所定の言語による単語を文字列として表現したテキストデータＴを入力する構成要素であり、符号合成ユニット３３０は、符号データベースユニット３１０を参照して、入力されたテキストデータＴの読みを構成する個々の音節にそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって、テキストデータＴの読みに対応する人間の声を表現した符号化音声データＤを合成する構成要素である。

【０１４８】

こうして、符号合成ユニット３３０で合成された符号化音声データＤ（たとえば、ＭＩＤＩデータ）は、音高変換装置１００に与えられる。この音高変換装置１００は、図１２に示す構成要素であり、その機能は既に§５で述べたとおりである。すなわち、所望のオフセット値αを与えることにより、入力された符号化音声データＤに対して音高変更処理を施し、符号化音声データＤ＊を出力する機能を有している。

【０１４９】

図１３に示す音声変換装置が、音声信号Ｓに基づいて符号化音声データＤ＊を生成する機能を有する装置であるのに対して、この図１４に示す音声合成装置は、テキストデータＴに基づいて符号化音声データＤ＊を生成する機能を有する装置ということができる。そのため、符号データベースユニット３１０には、所定の言語による単語を構成する個々の音節にそれぞれ対応する符号群が格納されている。特に、ここに示す実施形態の場合、日本語による単語を構成する個々の音節にそれぞれ対応する符号群として、子音を構成する子音音素に対応する符号群（子音音素用ＤＢ）と、母音を構成する母音音素に対応する符号群（母音音素用ＤＢ）と、が格納されている。

【０１５０】

符号合成ユニット３３０は、テキストデータＴの読みを構成する個々の音節を子音音素と母音音素とに分解し、符号データベースユニット３１０を参照して、個々の音素ごとにそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって符号化音声データＤを合成する処理を行う。

【０１５１】

図１５は、符号合成ユニット３３０内に用意されている変換テーブルを示す図である。この変換テーブルは、日本語の音節を構成する７１個のカタカナを、それぞれ子音音素と母音音素とに分解するためのテーブルであり、各カタカナは、テーブルの１行目に記載された子音音素（Ｋ，Ｓ，Ｔ，... 等の合計１４種類）と、テーブルの１列目に記載された

母音音素（Ａ，Ｉ，Ｕ，Ｅ，Ｏ，ｎの合計６種類）とに分解される。なお、カタカナの「ア，イ，ウ，エ，オ，ン」は、母音音素のみから構成される。また、「キャ」，「シュ」などの拗音を含むカナや、「トッ」のような促音を含むカナは、これらの組み合わせとして取り扱われる。

【０１５２】

たとえば、テキストデータ入力ユニット３２０によって、「コンニチワ」なる５文字のカタカナから構成されるテキストデータＴが入力された場合、符号合成ユニット３３０は、図１５に示す変換テーブルを利用して、この「コンニチワ」の５音節を、それぞれ子音音素と母音音素とに分解して、図１６に示すように、「ＫＯｎＮＩＴＩＷＡ」なる９音素を生成する。

【０１５３】

一方、符号データベースユニット３１０内には、図１７(a) ～(d) に示すようなＭＩＤＩ符号データベース（男性）と、図１８(a) ～(d) に示すようなＭＩＤＩ符号データベース（女性）とが用意されている。これらのデータベースは、６種類の母音音素「Ａ，Ｉ，Ｕ，Ｅ，Ｏ，ｎ」と１４種類の子音音素「Ｋ，Ｓ，Ｔ，Ｎ，Ｈ，Ｍ，Ｒ，Ｇ，Ｚ，Ｄ，Ｂ，Ｐ，Ｙ，Ｗ」について、それぞれ所定の符号群を対応づけたものである。すなわち、各音素は、それぞれ８個の音符からなる符号群に対応づけられている。たとえば、図１７(a) の［Ａ］の欄には、「Ｆ５，Ｅ５，Ｄ＃５，Ａ４，Ｇ＃４，Ｃ２，Ｂ１，Ａ＃１」なる８個の記号が収容されているが、これら８個の記号は、五線譜上の音階を示す記号である。したがって、この例の場合、１つの音素は、五線譜上において、８個の音符の和音として表現されることになる。なお、この図１７や図１８に示すようなデータベースは、アナウンサーや声優などに依頼して録音したサンプルについて、§１で述べた符号化方法を適用することにより準備することができる。

【０１５４】

図１９および図２０は、それぞれ図１７に示す男性用のＭＩＤＩ符号データベースおよび図１８に示す女性用のＭＩＤＩ符号データベースを用いて、各母音音素と各子音音素とに対応する符号を、音符として表示した例を示す楽譜である。実際には、符号合成ユニット３３０は、ＭＩＤＩデータから構成される符号化音声データＤを生成する処理を実行する。なお、図１７および図１８に示すデータベースには、強度や時間の情報は含まれていないので、ＭＩＤＩデータを生成する際に必要なベロシティーやデルタタイムのデータは、たとえば、予め設定した固定値を用いるようにすればよい。

【０１５５】

結局、符号合成ユニット３３０は、「コンニチワ」の５音節を「ＫＯｎＮＩＴＩＷＡ」なる９音素に分解した後、符号データベースユニット３１０を参照して、個々の音素ごとにそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって符号化音声データＤを合成する処理を行うことができる。ここに示す例では、符号データベースユニット３１０には、男性用のデータベース（図１７）と女性用のデータベース（図１８）とが用意されているので、オペレータにいずれか一方を選択させ、選択されたデータベースを利用して、符号群の抽出を行うようにする。もちろん、同じ男性用のデータベースであっても、低音声の男性用，高音声の男性用のように別々のデータベースを用意することもできる。

【０１５６】

こうして、上例の場合、テキスト符号化装置３００によって、「コンニチワ」なるテキストデータＴに基づいて、この５音節を男性もしくは女性が発音した場合に対応する符号化音声データＤ（ＭＩＤＩデータ）が出力される。この符号化音声データＤは、音高変換装置１００に与えられるので、所望のオフセット値αを与えることにより、音高変更処理を施した符号化音声データＤ＊を得ることができる。

【０１５７】

この音声合成装置を利用すると、カナ文字からなる任意のテキストデータＴに基づいて、当該カナ文字の読みを表現したＭＩＤＩ形式等の符号化音声データＤ＊を得ることがで

き、しかもその音高を、与えるオフセット値αによって任意に変化させることができる。もちろん、こうして出力された符号化音声データＤ＊は、ＭＩＤＩ音源等を利用して再生することが可能であり、楽譜として出力すれば、楽器を用いて演奏することが可能である。また、符号データベースとして外国語の音節に対応するデータベースを用意しておけば、外国語のテキストに基づいて符号化音声データを得ることも可能である。

【０１５８】

＜＜＜　§８．いくつかの変形例　＞＞＞

ここでは、これまで述べてきた様々な実施形態についての変形例を述べる。

【０１５９】

＜８－１．音声合成装置の第１の変形例＞

図２１は、図１４に示す音声合成装置の第１の変形例を示すブロック図である。図１４に示す音声合成装置では、音高を変更するためには、音高変換装置１００に対してオフセット値αを指定する必要があるが、この第１の変形例に係る音声合成装置では、予めテキストデータ内にオフセット値αを埋め込んでおくことにより、テキスト符号化装置３００Ａから音高変換装置１００に対して、自動的に、かつ、音素ごとにオフセット値αを指示することができる。

【０１６０】

図２１に示すとおり、テキスト符号化装置３００Ａは、符号データベースユニット３１０と、テキストデータ入力ユニット３２０と、符号合成ユニット３４０と、によって構成されている。ここで、符号データベースユニット３１０とテキストデータ入力ユニット３２０は、図１４に示すテキスト符号化装置３００に用いられているものと同じものでかまわない。ただ、図１４に示すテキスト符号化装置３００の場合、入力対象となるテキストデータＴとして、図２２(a) に例示する「コンニチワ」のように、カナ文字からなる文字列を用いていたが、この図２１に示すテキスト符号化装置３００Ａの場合、所定のオフセット値αが埋め込まれたテキストデータＴ（α）を用いることになる。

【０１６１】

図２２(b) は、オフセット値αが埋め込まれたテキストデータＴ（α）の一例を示す図である。例示したテキストデータＴ（α）は、「-3コ+3ン+6ニ+3チ-3ワ」のように、「半角の＋もしくは－」，「半角数字」，「全角カナ文字」によって構成される。ここで、「全角カナ文字」は、図２２(a) に示す「コンニチワ」なる文字列であり、本来のテキストデータを構成する文字列である。一方、「半角の＋もしくは－」および「半角数字」は、個々の「全角カナ文字」の前に挿入された記号であり、後続するカナ文字に対するオフセット値αを示すものである。たとえば、図２２(b) に示す例において、「-3」は、後続するカナ文字「コ」についてのオフセット値α＝－３を示しており、「+3」は、後続するカナ文字「ン」についてのオフセット値α＝＋３を示している。

【０１６２】

結局、図２２(b) に示すテキストデータＴ（α）は、個々の「全角カナ文字」の前に、当該カナ文字のオフセット値αを示すための記号として、「半角の＋もしくは－」および「半角数字」を挿入する、という書式をもったテキストデータということになる。テキストデータ入力ユニット３２０は、このような音節ごとのオフセット値αを含むテキストデータＴ（α）を取り込み、そのまま符号合成ユニット３４０へと与える。

【０１６３】

一方、符号合成ユニット３４０は、上記書式に基づいて、与えられたテキストデータＴ（α）を、オフセット値αとカナ文字とに分離し、カナ文字については、個々の音節を子音音素と母音音素とに分解し、符号データベースユニット３１０を参照して、個々の音素ごとにそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって符号化音声データＤを合成する（この処理は、図１４に示す音声合成装置３００内の符号合成ユニット３３０が行う処理と全く同様である）。そして、こうして合成した符号化音声データＤとともに、個々の音節ごとのオフセット値αを音高変換装置１００に対して出力する。

【０１６４】
たとえば、図２２(b)に示すような「-3コ+3ン+6ニ+3チ-3ワ」というテキストデータＴ（α）が与えられた場合、符号合成ユニット３４０は、これを「コ，ン，ニ，チ，ワ」なる５つの音節と、各音節に対応するオフセット値「－３，＋３，＋６，＋３，－３」とに分離し、「コ，ン，ニ，チ，ワ」なる音節を更に、「Ｋ，Ｏ，ｎ，Ｎ，Ｉ，Ｔ，Ｉ，Ｗ，Ａ」なる９つの音素に分解し、個々の音素に対して、図２２(c)に示すように、元の音節に対応するオフセット値αを割り当てる。そして、「Ｋ，Ｏ，ｎ，Ｎ，Ｉ，Ｔ，Ｉ，Ｗ，Ａ」なる９つの音素について、符号データベースユニット３１０を参照して、対応する符号群を抽出し、符号化音声データ（ＭＩＤＩデータ）を生成する。ここでは、音素ｘについて生成した符号化音声データをＤ（ｘ）と表すことにする。最後に、これら各符号化音声データをＤ（ｘ）に、割り当てられたオフセット値αを示す情報を付加し、音高変換装置１００へと出力する。

【０１６５】
図２２(d)は、上例の場合に、符号合成ユニット３４０から出力されるデータを示す図である。たとえば、１行目の符号化音声データＤ（Ｋ）は、子音音素「Ｋ」に対応するデータであり（たとえば、図１７(b)の［Ｋ］欄に記載された８個の音符に対応するＭＩＤＩデータ）、それに後続する「α＝－３」は、当該データＤ（Ｋ）に割り当てられたオフセット値αが－３であることを示すデータである。図示のとおり、「コンニチワ」なる単語を構成する９つの音素のすべてについて、それぞれ符号化音声データとオフセット値αとが与えられている。

【０１６６】
この図２２(d)に示すような「個々の音節（音素）ごとのオフセット値αを伴う符号化音声データＤ」が与えられた音高変換装置１００は、個々の音節（音素）ごとに、割り当てられているオフセット値αを用いた音高シフト処理を実行する。たとえば、図２２(d)の１行目に示す符号化音声データＤ（Ｋ）については、オフセット値α＝－３に応じた音高シフト処理が実行される。データＤ（Ｋ）を構成する符号が、たとえば、図１７(b)の［Ｋ］欄に記載された８個の音符に対応するＭＩＤＩデータであった場合、この８個の音符のノートナンバーに対して、オフセット値αに応じた増減処理（元のノートナンバーをｎとしたときに、「ｎ′＝ｎ＋α・Ｗ（ｎ）」で与えられる新たなノートナンバーｎ′に変更する処理）が行われる。

【０１６７】
このように、音高変換装置１００は、符号合成ユニット３４０から与えられたオフセット値αを用いて、個々の音節ごとに音高の変更処理を行うことができるので、同じ「コンニチワ」という単語であっても、音節ごとのオフセット値αを変化させることによって、自然な抑揚をもった標準語のアクセントによる「コンニチワ」や、関西弁のアクセントによる「コンニチワ」などのバリエーションをもった符号化音声データＤ＊を生成することができる。もちろん、通常は耳にしないような奇妙なアクセントをもった「コンニチワ」を生成することも可能なので、エンターテインメント分野での利用に適している。

【０１６８】
＜８－２．音声合成装置の第２の変形例＞
図２３は、図１４に示す音声合成装置の第２の変形例を示すブロック図であり、テキスト符号化装置３００の代わりに、テキスト符号化装置３００Ｂが用いられている。テキスト符号化装置３００Ｂは、図１４に示すテキスト符号化装置３００における符号合成ユニット３３０を、付加機能をもった符号合成ユニット３５０に置き換え、新たに、音高辞書ユニット３６０を追加したものである。符号データベースユニット３１０とテキストデータ入力ユニット３２０は、図１４に示すテキスト符号化装置３００に用いられているものと同じものでかまわない。

【０１６９】
図１４に示す音声合成装置では、音高変換装置１００に対して、外部からオフセット値αを指示する必要があった。一方、図２１に示す第１の変形例に係る音声合成装置では、

オフセット値αが埋め込まれたテキストデータＴ（α）を与える必要があった。これに対して、図２３に示す第２の変形例に係る音声合成装置では、オフセット値αが自動的に付与されるので、外部からオフセット値αを指定する必要がないというメリットが得られる。

【０１７０】

オフセット値αの自動付与に貢献するのは、音高辞書ユニット３６０である。この音高辞書ユニット３６０は、多数の単語について、それぞれ当該単語を構成する各音節に与えるオフセット値αを格納した辞書である。図２４は、この音高辞書ユニット３６０の内容を例示する表である。図示の例では、左欄の各単語について、当該単語を構成する各音節に適したオフセット値αが右欄に示されている。たとえば、「オハヨウ」という単語について、右欄には「－１，＋６，＋４，－３」というオフセット値αが記載されているが、これは、「オハヨウ」という単語を構成する４音節のそれぞれのオフセット値を示すものであり、音節「オ」についてはα＝－１，音節「ハ」についてはα＝＋６，音節「ヨ」についてはα＝＋４，音節「ウ」についてはα＝－３であることを示している。

【０１７１】

この図２４には、わずか３単語についてのオフセット値しか示されていないが、実際の音高辞書ユニットには、通常の辞書と同様に、多数の単語について、それぞれ当該単語を構成する各音節に適したオフセット値αが格納されている。このような音高辞書ユニット３６０を用意するには、アナウンサーなどに標準語で各単語を発音してもらい、当該単語に含まれる各音節の音高（たとえば、スペクトルでのピーク周波数）を測定し、規準音高（予め設定した特定の周波数、たとえば、多数の単語の各音節についての平均音高）に対する音程差を、各音節のオフセット値αとすればよい。もちろん、標準語辞書の他に、関西弁辞書、東北弁辞書などを用意することも可能である。

【０１７２】

さて、この変形例２に係る装置の場合、テキストデータＴとしては、カナ文字からなる通常のテキストを与えればよく、オフセット値αを含むテキストデータＴ（α）を与える必要はない。たとえば、「コンニチワ」なる文字列からなるテキストデータＴをテキストデータ入力ユニット３２０に与えたとすると、当該テキストデータＴは、そのまま符号合成ユニット３５０へ伝えられる。符号合成ユニット３５０は、個々の音節を子音音素と母音音素とに分解し、符号データベースユニット３１０を参照して、個々の音素ごとにそれぞれ対応する符号群を抽出し、これらを時間軸上に並べることによって符号化音声データＤを合成する（この処理は、図１４に示す音声合成装置３００内の符号合成ユニット３３０が行う処理と全く同様である）。

【０１７３】

ただ、この符号合成ユニット３５０には、オフセット値αの自動付与機能が備わっている。すなわち、符号合成ユニット３５０は、音高辞書ユニット３６０を参照することにより、単語「コンニチワ」についての個々の音節ごとのオフセット値αを認識し、符号化音声データＤとともに、このオフセット値αを音高変換装置１００に与える処理を実行する。具体的には、単語「コンニチワ」については、図２４に示す辞書を引くことにより、「－３，＋３，＋６，＋３，－３」なるオフセット値αが得られるので、符号化音声データＤに、これらのオフセット値αを付加したデータが、音高変換装置１００に対して出力される。

【０１７４】

こうして符号合成ユニット３５０から出力されるデータは、本質的には、図２２(d) に示すデータと全く同じになる。符号合成ユニット３５０は、辞書を引くことにより、図２２(c) に示すように、個々の音素ごとに割り当てるべきオフセット値αを認識することができるので、図２２(d) に示すように、個々の音素ごとの符号化音声データＤ（ｘ）と、これに割り当てられたオフセット値αを、音高変換装置１００に対して出力することができる。なお、テキストデータＴが文章からなる場合に、当該文章を構成するカナ文字をどのように区切って個々の単語として認識するか、という方法に関しては、既に形態素解析

などの言語解析手法として様々な手法が知られているので、ここでは詳しい説明は省略する。

【０１７５】

結局、変形例１に係る装置では、図２２(b) に例示するように、オフセット値αを含むテキストデータＴ（α）を与えることにより、個々の音節についてのオフセット値αを指示していたのに対して、変形例２に係る装置では、予め音高辞書ユニット３６０を用意しておくことにより、オフセット値αの指示を自動化したことになる。音高変換装置１００は、変形例１に係る装置と同様に、符号合成ユニット３５０から与えられたオフセット値αを用いて、個々の音節ごとに音高の変更処理を行うことができる。

【０１７６】

＜８－３．音声発声装置の付加＞

図２５は、本発明に係る音高変換装置１００から出力された符号化音声データＤ＊に基づいて、音声を発声させる音声発声装置４００の基本構成を示すブロック図である。ＭＩＤＩデータのような符号化音声データには、実際の音の波形に関する情報は含まれていないので、この符号化音声データを復号化して音を再生するためには、音の波形に関する情報をもった音源を用いる必要がある。

【０１７７】

図２５に示す音声発声装置４００は、図示のとおり、復号化ユニット４１０、音源ユニット４２０、発音ユニット４３０によって構成されている。ここで、音源ユニット４２０は、所定の楽器による様々な周波数の演奏音響波形をデジタルデータとして格納した構成要素である。復号化ユニット４１０は、与えられた符号化音声データＤ＊を構成する個々の符号を、音源ユニット４２０に格納されている対応する演奏音響波形に置き換えることにより音声信号Ｓの復号化を行う。また、発音ユニット４３０は、こうして復号化された音声信号Ｓに基づいて音波を生成する機能を果たし、実際には、アンプ装置やスピーカなどによって構成される。

【０１７８】

これまで述べてきた音声変換装置（図１３）や音声合成装置（図１４，図２１，図２３）に、図２５に示すような音声発声装置４００を付加すれば、音高変更が行われた後の符号化音声データＤ＊に基づいて、その場で実際の音を発声させることができるようになる。

【０１７９】

＜８－４．ＭＩＤＩ以外の規格による符号化＞

これまで述べた実施形態では、主として、本発明をＭＩＤＩ規格の符号化音声データを用いて実施した例を述べてきた。しかしながら、本発明を実施する場合、必ずしもＭＩＤＩ規格の符号化音声データを用いる必要はない。本発明は、特定周波数の音が特定時間だけ持続することを示す符号を時間軸上に並べることによって構成され人間の声を表現する符号化音声データであれば、どのような規格の符号化音声データを用いた場合でも、その作用効果が得られるものである。

【０１８０】

＜８－５．コンピュータプログラムによる実現＞

これまでの説明では、本発明に係る音高変換装置、音声変換装置、音声合成装置を、ブロック図として示し、個々のブロックで示す構成要素の集合体として示したが、実用上、これらの装置はコンピュータに専用のプログラムを組み込むことによって構成することができる。

【０１８１】

＜＜＜　§９．方法発明としての把握　＞＞＞

これまで、本発明を装置として把握した説明を行ったが、本発明に係る基本思想は、方法発明としても捉えることができる。

【０１８２】

具体的には、本発明は、特定周波数の音が特定時間だけ持続することを示す符号を時間

軸上に並べることによって構成され人間の声を表現する符号化音声データについて、その抑揚を変換する符号化音声データの抑揚変換方法として捉えることができる。この方法は、コンピュータが実行する方法であり、コンピュータが、変換対象となる符号化音声データＤを入力する変換対象データ入力段階と、コンピュータが、音高に関するオフセット値αを入力するオフセット値入力段階と、コンピュータが、変換対象となる符号化音声データＤに対して、オフセット値αに基づく音高の変更処理を行い、変更後の符号化音声データＤ＊を出力する音高変換処理段階と、を実行する。

【０１８３】

ここで、音高変換処理段階において、周波数ｆ軸上の所定区間においてＷ（ｆ）が周波数ｆの増加に従って単調減少する所定の重み関数Ｗ（ｆ）を利用して、変換対象となる符号化音声データＤに含まれている個々の符号について、当該符号が示す周波数ｆをα・Ｗ（ｆ）に応じた値だけ増減することにより新たな周波数ｆ′を求め、当該符号を、それが示す周波数ｆをｆ′に変更した新たな符号に置き換える処理を行うことが特徴となる。

【０１８４】

また、本発明をＭＩＤＩデータに適用した場合は、人間の声を表現したＭＩＤＩデータを、抑揚の異なる別なＭＩＤＩデータに変換するＭＩＤＩデータの抑揚変換方法として捉えることができる。この方法は、コンピュータが実行する方法であり、コンピュータが、変換対象となるＭＩＤＩデータＤを入力する変換対象データ入力段階と、コンピュータが、音高に関するオフセット値αを入力するオフセット値入力段階と、コンピュータが、変換対象となるＭＩＤＩデータＤに対して、オフセット値αに基づく音高の変更処理を行い、変更後のＭＩＤＩデータＤ＊を出力する音高変換処理段階と、を実行する。

【０１８５】

ここで、変換対象データ入力段階では、互いに異なるノートナンバーをもち、時間軸上の同一位置を占める複数のＭＩＤＩ符号を含むＭＩＤＩデータＤの入力を行い、音高変換処理段階では、ノートナンバー軸上の第１設定値ｎａおよび第２設定値ｎｂ（ｎｂ＞ｎａ）について、ｎ≦ｎａの区間は、Ｗ（ｎ）＝１、ｎａ＜ｎ＜ｎｂの区間は、１＞Ｗ（ｎ）＞０（但し、Ｗ（ｎ）はｎの増加に従って単調減少）、ｎ≧ｎｂの区間は、Ｗ（ｎ）＝０となる所定の重み関数Ｗ（ｎ）を利用して、変換対象となるＭＩＤＩデータＤに含まれている個々のＭＩＤＩ符号について、当該ＭＩＤＩ符号が示すノートナンバーｎに対してα・Ｗ（ｎ）に応じた値だけ加減算を行うことにより新たなノートナンバーｎ′を求め、当該ＭＩＤＩ符号を、それが示すノートナンバーｎをｎ′に変更した新たなＭＩＤＩ符号に置き換える処理を行うことにより、変更後のＭＩＤＩデータＤ＊を生成することが特徴となる。

【０１８６】

このＭＩＤＩデータの抑揚変換方法では、音高変換処理段階で、新たなＭＩＤＩ符号に置き換える処理を行う際に、時間軸上の同一位置を占め、同一のノートナンバーをもつ新たなＭＩＤＩ符号が複数ｍ個生じた場合には、当該複数ｍ個のＭＩＤＩ符号のうち１つのみを残し、その余の（ｍ－１）個を削除する重複回避処理を行うのが好ましい。

【産業上の利用可能性】

【０１８７】

本発明に係る技術は、人間の声をＭＩＤＩ規格などに基づいて符号化して取り扱うことを前提とする技術である。しかも、ＰＣＭのような符号化とは根本的に異なり、本発明で用いられる符号化音声データは、特定周波数の音が特定時間だけ持続することを示すいくつかの符号を時間軸上に並べることによって構成されるデータであるため、再生した場合、当然ながら、元の人間の声に対して、かなりかけ離れた音声が得られることになる。したがって、本発明は、人間の声を正確に記録し、再生するという用途に利用されるものではない。

【０１８８】

しかしながら、「人間の声を音符等の符号によって表現し、楽器によって演奏できるようにする」という着想は極めてユニークであり、エンターテインメントの分野では、様々

な利用形態が期待できる。しかも本発明では、オフセット値αを指定することにより、音高を自由に調整することができるので、その用途は更に広がることになる。

【０１８９】

具体的には、本発明は、イベントの余興目的に行われる人間の音声再生を模倣した音楽作品の制作や作曲の支援に利用することができる。また、エンターテインメントの分野において、電子楽器を主体とした玩具（ロボットやぬいぐるみなども含む）、玩具型のアコースティック楽器（室内装飾用のミチチュアピアノなど）、オルゴールなどにも利用することができ、更に、携帯電話の着信メロディなどの音階再生媒体に対して音声合成機能を付加する用途にも適している。

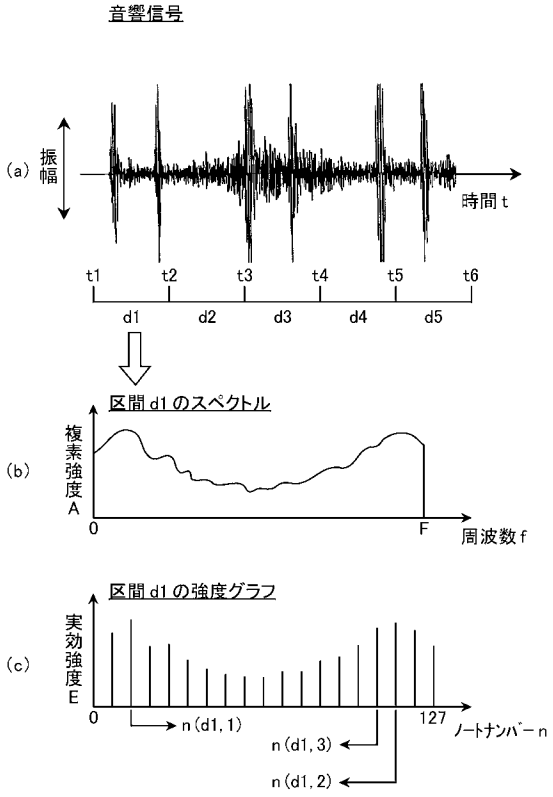
【符号の説明】

【０１９０】

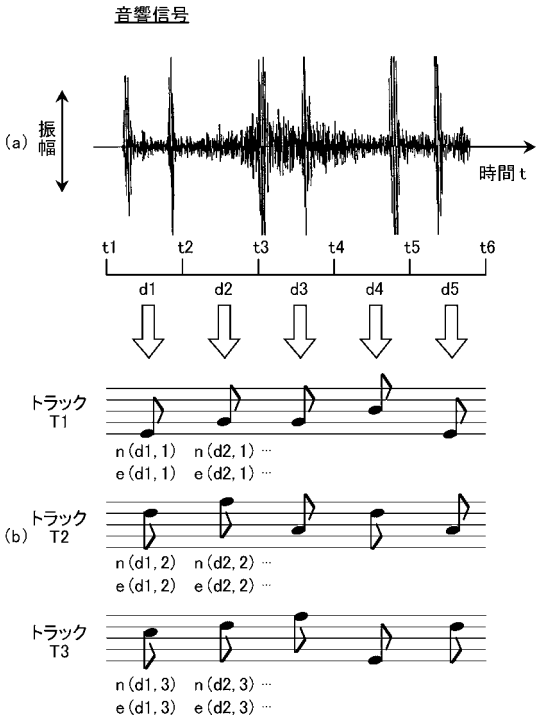
１０：オーディオ機器
２０：マイク
３０：アンプ装置
１００：音高変換装置
１１０：変換対象データ入力ユニット
１２０：オフセット値入力ユニット
１３０：音高変換処理ユニット
１４０：重み関数格納ユニット
２００：音声符号化装置
２１０：音声信号入力ユニット
２２０：符号化ユニット
３００，３００Ａ，３００Ｂ：テキスト符号化装置
３１０：符号データベースユニット
３２０：テキストデータ入力ユニット
３３０：符号合成ユニット
３４０：符号合成ユニット
３５０：符号合成ユニット
３６０：音高辞書ユニット
４００：音声発生装置
４１０：復号化ユニット
４２０：音源ユニット
４３０：発音ユニット（アンプ装置・スピーカ）
Ａ：複素強度
Ｄ：符号化音声データ
Ｄ＊：音高変更後の符号化音声データ
Ｄ（ｘ）：音素ｘの符号化音声データ
ｄ１～ｄ５：単位区間
Ｅ：実効強度（エネルギー）
ｅ（ｉ，ｊ）：符号コードｎ（ｉ，ｊ）の実効強度
Ｆ：サンプリング周波数
Ｆ０：基本周波数
Ｆ１：第１フォルマント周波数
Ｆ２：第２フォルマント周波数
Ｆ３：第３フォルマント周波数
ｆ：周波数
ｆ′：音高変更後の周波数
ｆ１～ｆ３：個々の音符が示す周波数
ｆ１＊～ｆ３＊：音高変更後の個々の音符が示す周波数
ｆａ：周波数の第１設定値

ｆｂ：周波数の第２設定値
ｎ：ノートナンバー
ｎ１～ｎ５：音符／個々の音符が示すノートナンバー
ｎ１＊～ｎ５＊：音高変更後の音符／個々の音符が示すノートナンバー
ｎａ：周波数ｆａに対応するノートナンバー
ｎｂ：周波数ｆｂに対応するノートナンバー
ｎ（ｉ，ｊ）：単位区間ｄｉについて抽出された第ｊ番目の符号コード
Ｓ：音声信号
Ｔ：テキストデータ
Ｔ（α）：オフセット値αを含むテキストデータ
Ｔ１～Ｔ３：トラック
ｔ：時間
ｔ１～ｔ６：時刻
Ｗ（ｆ），Ｗ（ｎ）：重み関数
α：オフセット値

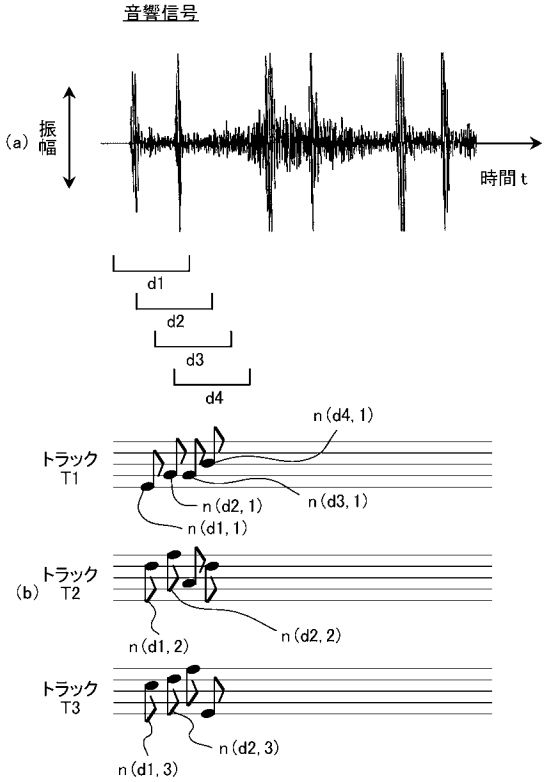
【図１】



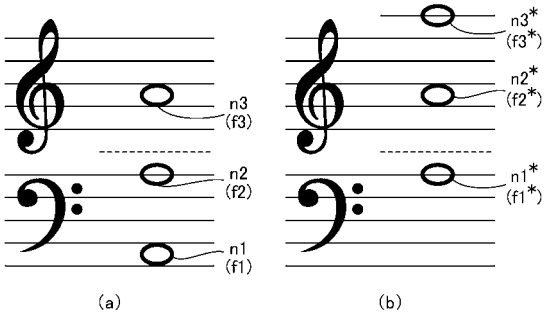
【図２】



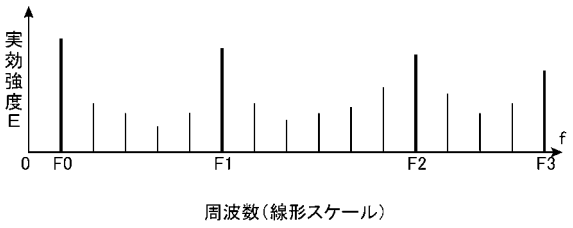
【図３】



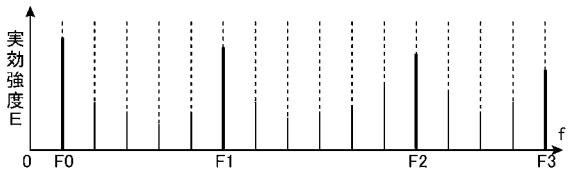
【図４】



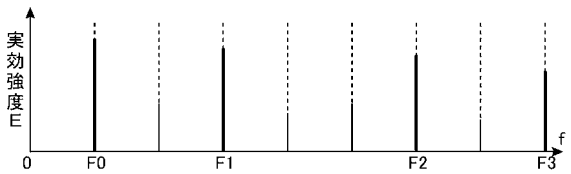
【図５】



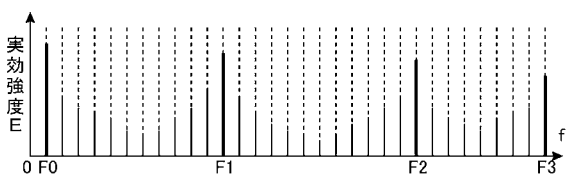
【図６】



(a)周波数(線形スケール)

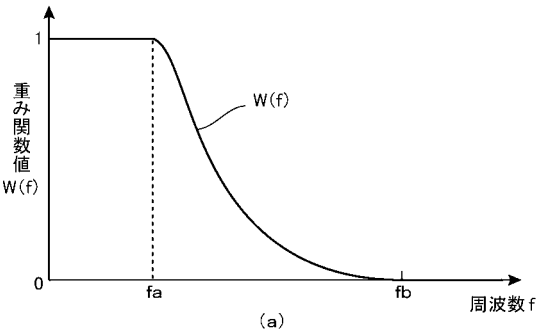


(b)周波数(線形スケール)

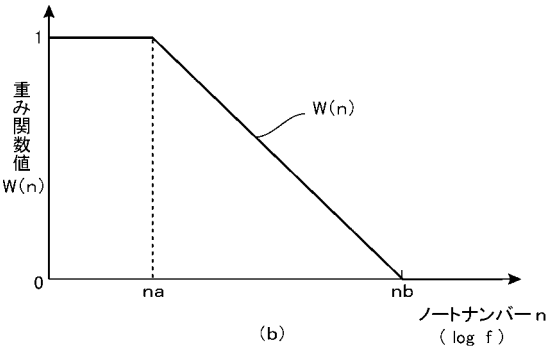


(c)周波数(線形スケール)

【図７】

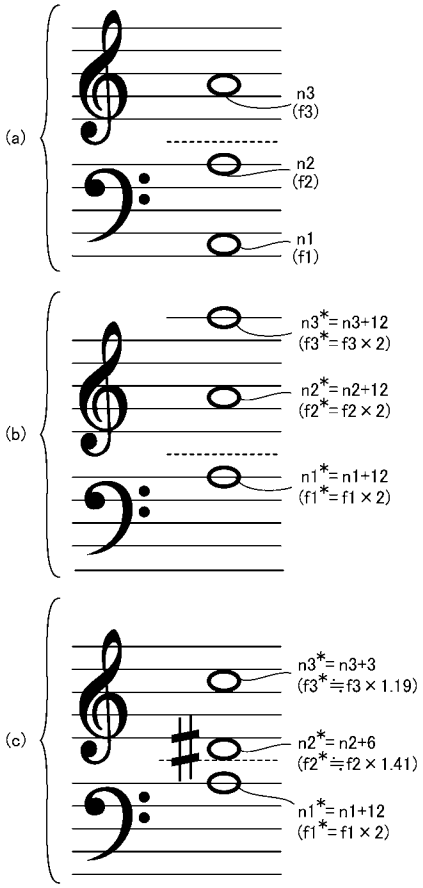


(a)

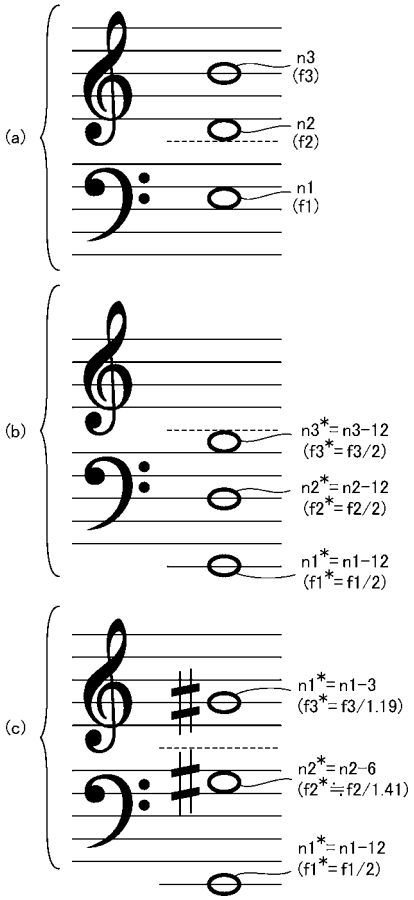


(b)

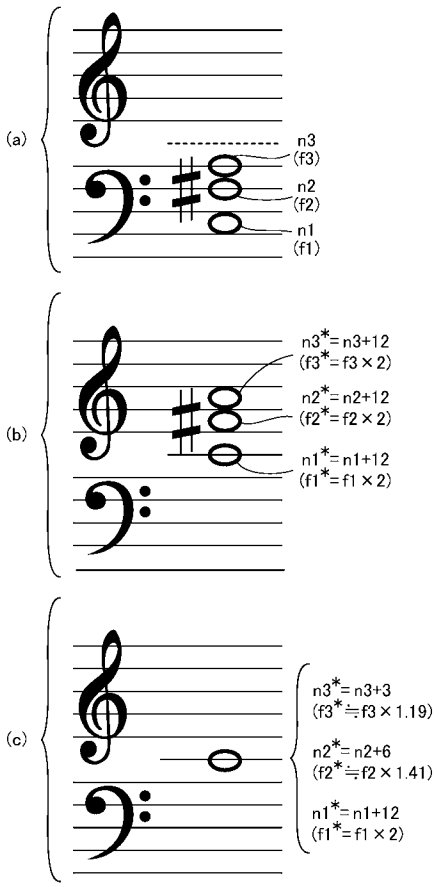
【図８】


(a)
n3
(f3)
n2
(f2)
n1
(f1)
(b)
n3*= n3+12
(f3*= f3 × 2)
n2*= n2+12
(f2*= f2 × 2)
n1*= n1+12
(f1*= f1 × 2)
(c)
n3*= n3+3
(f3*≒f3 × 1.19)
n2*= n2+6
(f2*≒f2 × 1.41)
n1*= n1+12
(f1*= f1 × 2)

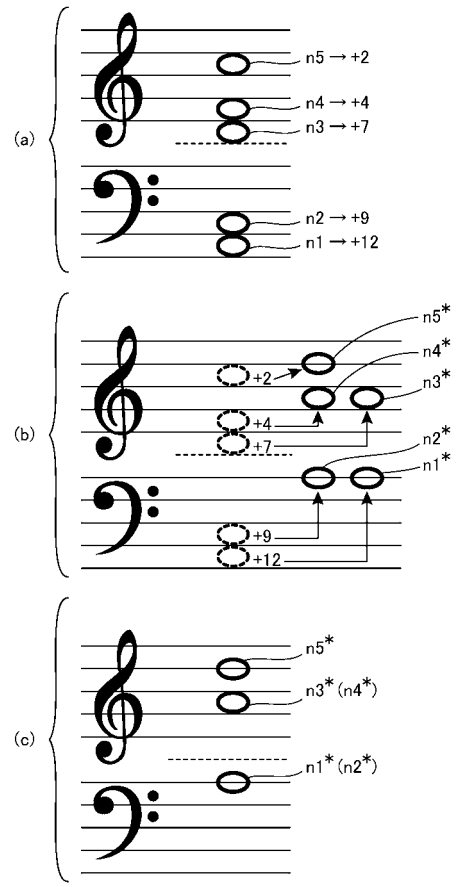
【図９】


(a)
n3
(f3)
n2
(f2)
n1
(f1)
(b)
n3*= n3−12
(f3*= f3/2)
n2*= n2−12
(f2*= f2/2)
n1*= n1−12
(f1*= f1/2)
(c)
n1*= n1−3
(f3*= f3/1.19)
n2*= n2−6
(f2*≒f2/1.41)
n1*= n1−12
(f1*= f1/2)

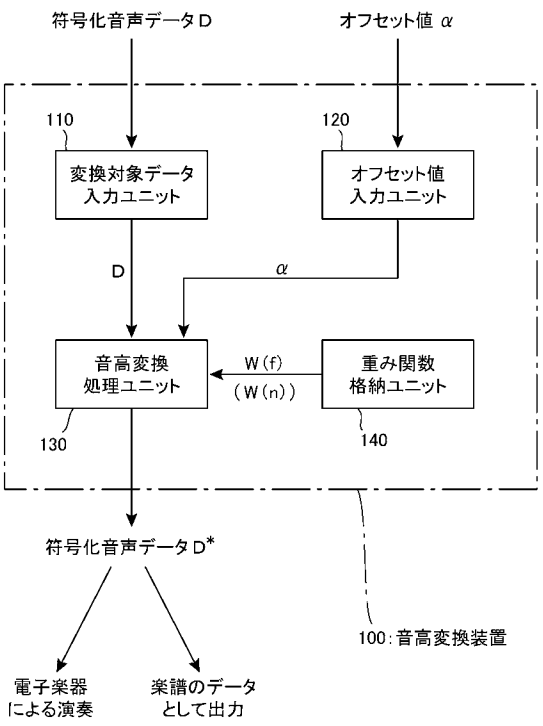
【図１０】


(a)
n3
(f3)
n2
(f2)
n1
(f1)
(b)
n3*= n3+12
(f3*= f3 × 2)
n2*= n2+12
(f2*= f2 × 2)
n1*= n1+12
(f1*= f1 × 2)
(c)
n3*= n3+3
(f3*≒f3 × 1.19)
n2*= n2+6
(f2*≒f2 × 1.41)
n1*= n1+12
(f1*= f1 × 2)

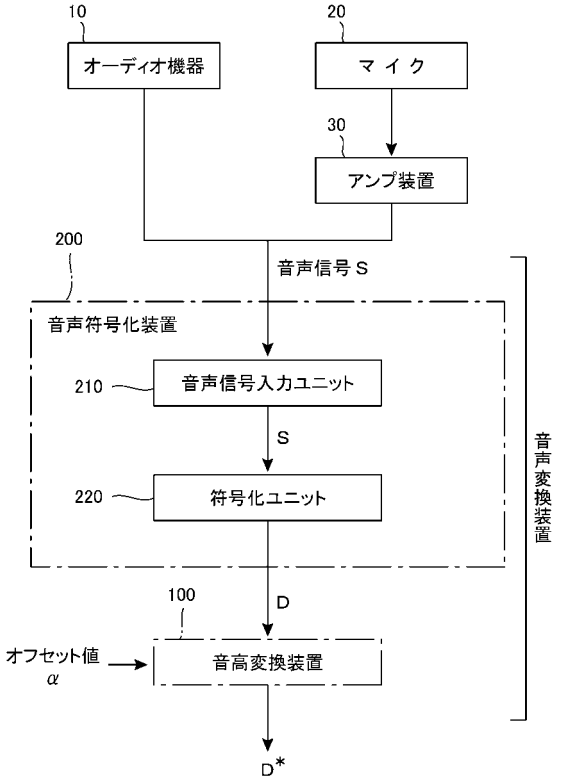
【図１１】


(a)
n5 → +2
n4 → +4
n3 → +7
n2 → +9
n1 → +12
(b)
n5*
n4*
n3*
n2*
n1*
+2
+4
+7
+9
+12
(c)
n5*
n3* (n4*)
n1* (n2*)

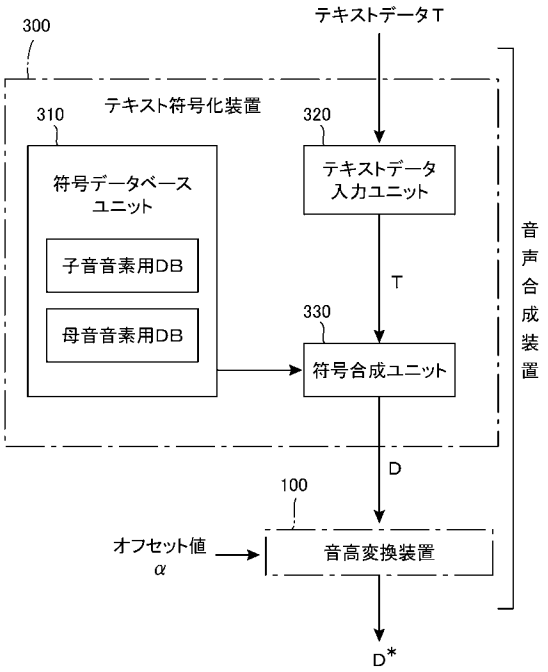
【図１２】



【図１３】



【図１４】



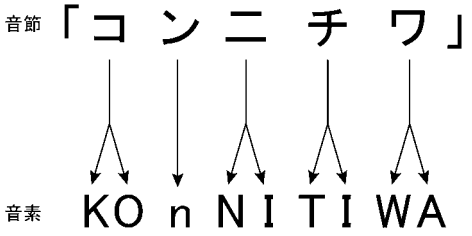
【図１５】

変換テーブル

		K	S	T	N	H	M	R	G	Z	D	B	P	Y	W
A	ア	カ	サ	タ	ナ	ハ	マ	ラ	ガ	ザ	ダ	バ	パ	ヤ	ワ
I	イ	キ	シ	チ	ニ	ヒ	ミ	リ	ギ	ジ	ヂ	ビ	ピ		
U	ウ	ク	ス	ツ	ヌ	フ	ム	ル	グ	ズ	ヅ	ブ	プ	ユ	
E	エ	ケ	セ	テ	ネ	ヘ	メ	レ	ゲ	ゼ	デ	ベ	ペ		
O	オ	コ	ソ	ト	ノ	ホ	モ	ロ	ゴ	ゾ	ド	ボ	ポ	ヨ	ヲ
n					ン										

音節：カタカナ（71個）
音素：アルファベット（20個）

【図１６】



【図１７】

MIDI符号データベース（男性）

[A]	[I]	[U]	[E]	[O]
F5	A#6	D3	A#3	F#4
E5	C#3	C#3	A3	F4
D#5	A#2	A#2	G#3	E4
A4	D2	A2	E3	G#3
G#4	C#2	D2	G#2	G3
C2	C2	C#2	D2	F#3
B1	B1	C2	C#2	C2
A#1	A#1	B1	A1	B1

(a)

[K]	[S]	[T]	[N]	[H]
F4	B6	D#4	C4	D4
A#3	A6	F3	D#3	A3
C3	C4	D#3	G#2	F3
B2	B3	B2	G2	E3
A#2	F3	G#2	F#2	D#3
G2	F2	F#2	F2	F#2
C#2	E2	E2	E2	E2
A1	D#2	A1	A1	A1

(b)

[M]	[R]	[G]	[Z]	[D]
B3	B3	C#4	C#4	D4
D#3	D#3	B3	C4	F3
C3	C3	F3	C3	C3
G2	B2	C3	B2	G#2
F#2	G#2	B2	G2	G2
F2	G2	G2	F#2	F2
E2	F2	F#2	E2	E2
D#2	A1	E2	D#2	A1

(c)

[B]	[P]	[Y]	[W]	[n]
C4	C4	C#4	E4	D3
B3	A#3	B3	D4	B2
F3	A3	A3	C#4	A2
D#3	F3	F3	B3	G#2
C3	G2	C3	G#3	D2
B2	E2	G#2	G3	C#2
G2	D#2	D#2	C2	C2
E2	A1	A1	A#1	B1

(d)

【図１８】

MIDI符号データベース（女性）

[A]	[I]	[U]	[E]	[O]
B5	A6	E5	G6	B4
A#5	G#6	D#5	D#6	A#4
F#5	G6	D#4	B4	A4
D#5	F#6	C#4	F4	G#4
C#5	G3	D#3	E4	D4
C5	F3	D3	D4	C#4
B4	E3	C#3	C#4	A#3
G#4	D#3	A#2	C4	A3

(a)

[K]	[S]	[T]	[N]	[H]
D#5	G#6	G#6	C3	F4
B4	E4	D6	B2	E4
D#4	D#4	G#5	A#2	D#4
C4	B3	F4	A2	B3
A3	E3	E4	G#2	A#3
D#3	D3	D#4	F2	E3
D3	C#3	E3	C2	D#3
C#3	B2	G#2	A#1	B2

(b)

[M]	[R]	[G]	[Z]	[D]
F#4	G#6	G6	A6	G#6
C3	D#4	F#6	G#6	D#4
B2	C4	E4	G#5	C4
A#2	B3	D4	D3	B3
G#2	C3	C4	C3	D3
F#2	B2	B3	A2	C3
F2	A2	D#3	G#2	G2
C#2	G#2	B2	C2	C2

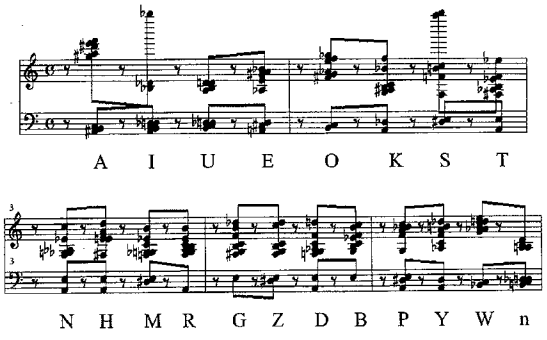
(c)

[B]	[P]	[Y]	[W]	[n]
G4	E5	D#6	D#5	B5
D#4	E4	F#4	D5	A#5
B3	D#4	E4	B4	C4
B2	D4	C4	A#4	A#3
A2	B3	B3	E4	D#3
G#2	A#3	E3	D#4	D3
G2	E3	A2	D4	C3
B1	D#3	G#2	C4	B2

(d)

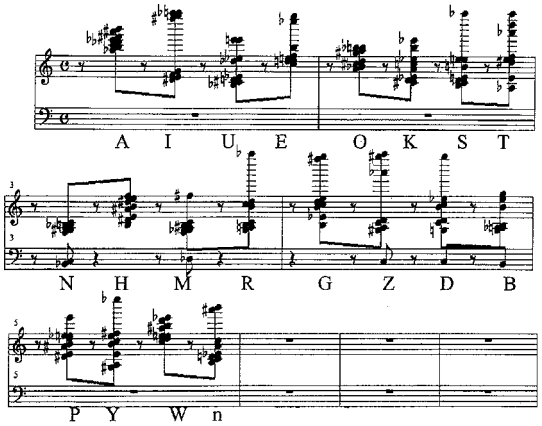
【図１９】

各音素の五線譜化の例（男性）

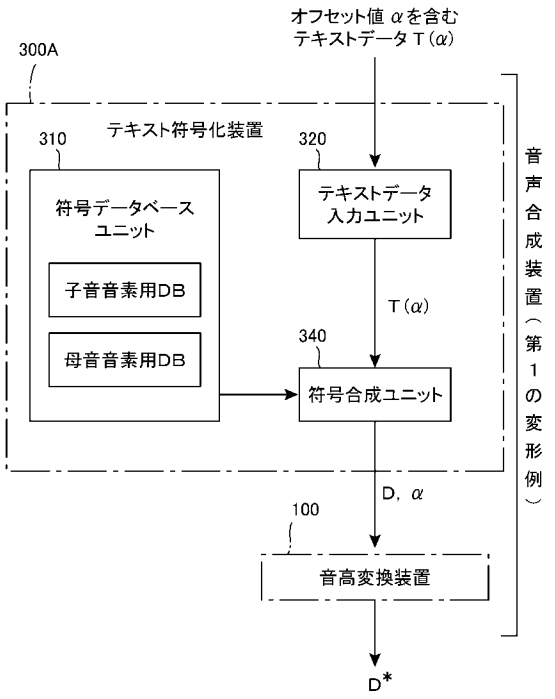


【図２０】

各音素の五線譜化の例（女性）



【図２１】



【図２２】

(a)「コンニチワ」

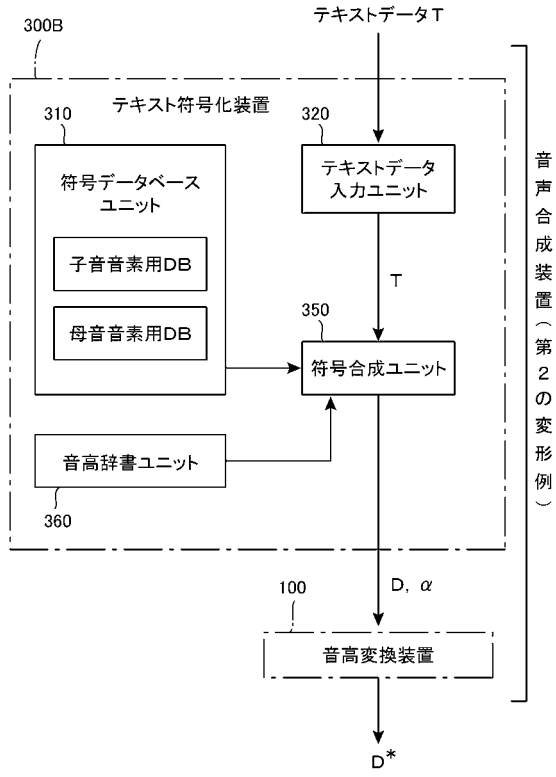
(b)「−3コ+3ン+6ニ+3チ−3ワ」

(c)

KO	n	NI	TI	WA
−3	+3	+6	+3	−3

(d)
符号化音声データ　D(K), α＝−3
符号化音声データ　D(O), α＝−3
符号化音声データ　D(n), α＝＋3
符号化音声データ　D(N), α＝＋6
符号化音声データ　D(I), α＝＋6
符号化音声データ　D(T), α＝＋3
符号化音声データ　D(I), α＝＋3
符号化音声データ　D(W), α＝−3
符号化音声データ　D(A), α＝−3

【図２３】



【図２４】

360：音高辞書ユニット

単語	オフセット値α
オハヨウ	−１，＋６，＋４，−３
コンニチワ	−３，＋３，＋６，＋３，−３
コンバンワ	−２，＋２，＋５，＋２，−４
⋮	⋮

【図２５】