

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5154886号
(P5154886)

(45) 発行日 平成25年2月27日 (2013. 2. 27)

(24) 登録日 平成24年12月14日 (2012. 12. 14)

(51) Int. Cl.	F I
G 1 O G 3/04 (2006. 01)	G 1 O G 3/04
G 1 O H 1/00 (2006. 01)	G 1 O H 1/00 Z
G O 6 K 9/00 (2006. 01)	G 1 O H 1/00 1 O 2 Z
	G O 6 K 9/00 N

請求項の数 5 (全 28 頁)

(21) 出願番号	特願2007-266448 (P2007-266448)	(73) 特許権者	000001410
(22) 出願日	平成19年10月12日 (2007. 10. 12)		株式会社河合楽器製作所
(65) 公開番号	特開2009-98161 (P2009-98161A)		静岡県浜松市中区寺島町2〇〇番地
(43) 公開日	平成21年5月7日 (2009. 5. 7)	(74) 代理人	100090273
審査請求日	平成22年9月24日 (2010. 9. 24)		弁理士 國分 孝悦
		(72) 発明者	中野 誠至
			静岡県浜松市中区寺島町2〇〇番地 株式
			会社河合楽器製作所内
		審査官	清水 正一

最終頁に続く

(54) 【発明の名称】 楽譜認識装置及びコンピュータプログラム

(57) 【特許請求の範囲】

【請求項 1】

紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第1の取得手段と、
前記第1の取得手段により取得された画像に含まれる楽譜を認識する認識手段と、
前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶する記憶手段と、

前記記憶手段に記憶されたパートテンプレートの中から、前記読み取り手段により読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第2の取得手段と、
前記第2の取得手段により取得されたパートテンプレートを用いて、前記読み取り手段により読み取られた画像に含まれる楽譜に対する設定を自動的に行う設定手段と、

前記第2の取得手段により取得されたパートテンプレートを示す情報を、前記楽譜のパート毎に表示する表示手段と、

前記表示手段により表示されたパートテンプレートを示す情報に対するユーザの操作に基づいて、前記第2の取得手段により取得されたパートテンプレートの少なくとも1つの変更を受け付ける受付手段と、

前記受付手段により受け付けられたパートテンプレートの前記記憶手段による更新を制限する制限手段と、を有し、

前記記憶手段は、前記受付手段により受け付けられたパートテンプレートであって、前記制限手段により変更が制限されないパートテンプレートを、ユーザの操作に基づいて更新し、

10

20

前記表示手段は、前記記憶手段によりパートテンプレートが更新された場合には、前記取得手段により取得されたパートテンプレートを示す情報に代えて、前記更新されたパートテンプレートを示す情報を表示し、

前記制限手段は、更新によってM I D I のプログラムナンバーが変わるパートテンプレートへの更新を制限することを特徴とする楽譜認識装置。

【請求項 2】

紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第 1 の取得手段と、
前記第 1 の取得手段により取得された画像に含まれる楽譜を認識する認識手段と、
前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶する記憶手段と、

10

前記記憶手段に記憶されたパートテンプレートの中から、前記読み取り手段により読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第 2 の取得手段と、
前記第 2 の取得手段により取得されたパートテンプレートを用いて、前記認識手段による楽譜の認識の結果を自動的に修正する設定手段とを有し、

前記設定手段は、前記認識手段により認識された楽譜の 1 つのパート内の情報を用いて、前記認識手段による楽譜の認識の結果を修正する処理と、前記認識手段により認識された楽譜の複数のパート内の情報を用いて、前記認識手段による楽譜の認識の結果を修正する処理とを別々に行うことを特徴とする楽譜認識装置。

【請求項 3】

前記第 2 の取得手段により取得されたパートテンプレートを示す情報を、前記楽譜のパート毎に表示する表示手段と、

20

前記表示手段により表示されたパートテンプレートを示す情報に対するユーザの操作に基づいて、前記第 2 の取得手段により取得されたパートテンプレートの少なくとも 1 つの変更を受け付ける受付手段と、

前記受付手段により受け付けられたパートテンプレートの前記記憶手段による更新を制限する制限手段と、を有し、

前記記憶手段は、前記受付手段により受け付けられたパートテンプレートであって、前記制限手段により変更が制限されないパートテンプレートを、ユーザの操作に基づいて更新し、

前記表示手段は、前記記憶手段によりパートテンプレートが更新された場合には、前記取得手段により取得されたパートテンプレートを示す情報に代えて、前記更新されたパートテンプレートを示す情報を表示し、

30

前記制限手段は、更新によってM I D I のプログラムナンバーが変わるパートテンプレートへの更新を制限することを特徴とする請求項 2 に記載の楽譜認識装置。

【請求項 4】

紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第 1 の取得ステップと、

前記第 1 の取得ステップにより取得された画像に含まれる楽譜を認識する認識ステップと、

前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶媒体に記憶する記憶ステップと、

40

前記記憶ステップにより記憶されたパートテンプレートの中から、前記読み取りステップにより読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第 2 の取得ステップと、

前記第 2 の取得ステップにより取得されたパートテンプレートを用いて、前記読み取りステップにより読み取られた画像に含まれる楽譜に対する設定を自動的に行う設定ステップと、

前記第 2 の取得ステップにより取得されたパートテンプレートを示す情報を、前記楽譜のパート毎に表示装置に表示する表示ステップと、

前記表示ステップにより表示されたパートテンプレートを示す情報に対するユーザの操

50

作に基づいて、前記第2の取得ステップにより取得されたパートテンプレートの少なくとも1つの変更を受け付ける受付ステップと、

前記受付ステップにより受け付けられたパートテンプレートの前記記憶ステップによる更新を制限する制限ステップと、をコンピュータに実行させ、

前記記憶ステップは、前記受付ステップにより受け付けられたパートテンプレートであって、前記制限ステップにより変更が制限されないパートテンプレートを、ユーザの操作に基づいて更新し、

前記表示ステップは、前記記憶ステップによりパートテンプレートが更新された場合には、前記取得ステップにより取得されたパートテンプレートを示す情報に代えて、前記更新されたパートテンプレートを示す情報を表示装置に表示し、

前記制限ステップは、更新によってMIDIのプログラムナンバーが変わるパートテンプレートへの更新を制限することを特徴とするコンピュータプログラム。

【請求項5】

紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第1の取得ステップと、

前記第1の取得ステップにより取得された画像に含まれる楽譜を認識する認識ステップと、

前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶媒体に記憶する記憶ステップと、

前記記憶ステップにより記憶されたパートテンプレートの中から、前記読み取りステップにより読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第2の取得ステップと、

前記第2の取得ステップにより取得されたパートテンプレートを用いて、前記認識ステップによる楽譜の認識の結果を自動的に修正する設定ステップとをコンピュータに実行させ、

前記設定ステップは、前記認識ステップにより認識された楽譜の1つのパート内の情報を用いて、前記認識ステップによる楽譜の認識の結果を修正する処理と、前記認識ステップにより認識された楽譜の複数のパート内の情報を用いて、前記認識ステップによる楽譜の認識の結果を修正する処理とを別々に行うことを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、楽譜認識装置及びコンピュータプログラムに関し、特に、紙面の楽譜を認識するために用いて好適なものである。

【背景技術】

【0002】

従来から、スキャナ等を用いて、紙面上の楽譜を読み取り、読み取った楽譜のデータについて、五線、音符、及びその他の各種記号を認識して、MIDIファイルデータ等の演奏データを生成することが行われている（特許文献1を参照）。

例えば、五線を消去することにより、画像上のラベルを分離することで、五線と重なった音楽記号を分離し、音楽記号の夫々を認識する。その際、臨時記号等の形状が変化しない定型記号については、加重方向指数等の特徴量と、辞書の音楽記号データとをマッチングすることにより認識する。また、音符等の形状が変化する音楽記号については、縦横の細線や、符頭を構成する楕円や、連符を構成する太線を分離して認識し、それらの形状や位置関係に基づいて認識する。

【0003】

【特許文献1】特開平9-97058号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

ところで、例えば移調楽器で演奏することを反映した楽譜では、そのことが楽譜に表記されていない。しかしながら、従来の技術では、このような楽譜に表記されていない情報については、楽譜を認識した後、ユーザが手動で設定する必要があった。この他、認識した楽譜がタブ譜であり、そのタブ譜にチューニングの設定（表記）があり、ユーザがそのチューニングの設定を変更したい場合にも、ユーザは、そのチューニングの設定を手動で変更する必要があった。

また、従来の技術では、楽譜のあるべき状態が分からなかったので、楽譜の認識結果をそのまま採用するしかなく、楽譜の認識率を向上させることが困難であった。

【0005】

本発明は、以上のような問題点に鑑みてなされたものであり、楽譜を認識した後にユーザが行う設定を従来よりも少なくすることを第1の目的とする。

また、楽譜の認識率を従来よりも向上させることを第2の目的とする。

【課題を解決するための手段】

【0006】

本発明の楽譜認識装置の一態様例では、紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第1の取得手段と、前記第1の取得手段により取得された画像に含まれる楽譜を認識する認識手段と、前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶する記憶手段と、前記記憶手段に記憶されたパートテンプレートの中から、前記読み取り手段により読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第2の取得手段と、前記第2の取得手段により取得されたパートテンプレートを用いて、前記読み取り手段により読み取られた画像に含まれる楽譜に対する設定を自動的に行う設定手段と、前記第2の取得手段により取得されたパートテンプレートを示す情報を、前記楽譜のパート毎に表示する表示手段と、前記表示手段により表示されたパートテンプレートを示す情報に対するユーザの操作に基づいて、前記第2の取得手段により取得されたパートテンプレートの少なくとも1つの変更を受け付ける受付手段と、前記受付手段により受け付けられたパートテンプレートの前記記憶手段による更新を制限する制限手段と、を有し、前記記憶手段は、前記受付手段により受け付けられたパートテンプレートであって、前記制限手段により変更が制限されないパートテンプレートを、ユーザの操作に基づいて更新し、前記表示手段は、前記記憶手段によりパートテンプレートが更新された場合には、前記取得手段により取得されたパートテンプレートを示す情報に代えて、前記更新されたパートテンプレートを示す情報を表示し、前記制限手段は、更新によってMIDIのプログラムナンバーが変わるパートテンプレートへの更新を制限することを特徴とする。

本発明の楽譜認識装置の他の態様例では、紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第1の取得手段と、前記第1の取得手段により取得された画像に含まれる楽譜を認識する認識手段と、前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶する記憶手段と、前記記憶手段に記憶されたパートテンプレートの中から、前記読み取り手段により読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第2の取得手段と、前記第2の取得手段により取得されたパートテンプレートを用いて、前記認識手段による楽譜の認識の結果を自動的に修正する設定手段とを有し、前記設定手段は、前記認識手段により認識された楽譜の1つのパート内の情報を用いて、前記認識手段による楽譜の認識の結果を修正する処理と、前記認識手段により認識された楽譜の複数のパート内の情報を用いて、前記認識手段による楽譜の認識の結果を修正する処理とを別々に行うことを特徴とする。

【0007】

本発明のコンピュータプログラムの一態様例では、紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第1の取得ステップと、前記第1の取得ステップにより取得された画像に含まれる楽譜を認識する認識ステップと、前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶媒体に記憶する記憶ステップと、前記記憶ステップにより記憶されたパートテンプレートの中から、前記読

10

20

30

40

50

み取りステップにより読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第2の取得ステップと、前記第2の取得ステップにより取得されたパートテンプレートを用いて、前記読み取りステップにより読み取られた画像に含まれる楽譜に対する設定を自動的に行う設定ステップと、前記第2の取得ステップにより取得されたパートテンプレートを示す情報を、前記楽譜のパート毎に表示装置に表示する表示ステップと、前記表示ステップにより表示されたパートテンプレートを示す情報に対するユーザの操作に基づいて、前記第2の取得ステップにより取得されたパートテンプレートの少なくとも1つの変更を受け付ける受付ステップと、前記受付ステップにより受け付けられたパートテンプレートの前記記憶ステップによる更新を制限する制限ステップと、をコンピュータに実行させ、前記記憶ステップは、前記受付ステップにより受け付けられたパートテンプレートであって、前記制限ステップにより変更が制限されないパートテンプレートを、ユーザの操作に基づいて更新し、前記表示ステップは、前記記憶ステップによりパートテンプレートが更新された場合には、前記取得ステップにより取得されたパートテンプレートを示す情報に代えて、前記更新されたパートテンプレートを示す情報を表示装置に表示し、前記制限ステップは、更新によってMIDIのプログラムナンバーが変わるパートテンプレートへの更新を制限することを特徴とする。

10

本発明のコンピュータプログラムの他の態様例では、紙面の楽譜の情報を含む画像を、画像読み取り手段から取得する第1の取得ステップと、前記第1の取得ステップにより取得された画像に含まれる楽譜を認識する認識ステップと、前記楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶媒体に記憶する記憶ステップと、前記記憶ステップにより記憶されたパートテンプレートの中から、前記読み取りステップにより読み取られた画像に含まれる楽譜に応じたパートテンプレートを取得する第2の取得ステップと、前記第2の取得ステップにより取得されたパートテンプレートを用いて、前記認識ステップによる楽譜の認識の結果を自動的に修正する設定ステップとをコンピュータに実行させ、前記設定ステップは、前記認識ステップにより認識された楽譜の1つのパート内の情報を用いて、前記認識ステップによる楽譜の認識の結果を修正する処理と、前記認識ステップにより認識された楽譜の複数のパート内の情報を用いて、前記認識ステップによる楽譜の認識の結果を修正する処理とを別々に行うことを特徴とする。

20

【発明の効果】

30

【0008】

本発明によれば、楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶しておく。そして、読み取られた楽譜に応じたパートテンプレートを予め記憶しておいたものの中から取得し、取得したパートテンプレートを用いて、当該楽譜に対する設定を自動的に行うようにした。したがって、楽譜を認識した後にユーザが行う設定を従来よりも少なくすることができる。

また、楽譜の認識後に当該楽譜に対して設定する必要がある設定情報を含むパートテンプレートを予め記憶しておく。そして、読み取られた楽譜に応じたパートテンプレートを予め記憶しておいたものの中から取得し、取得したパートテンプレートを用いて、楽譜の認識の結果を自動的に修正するよう¹にした。したがって、楽譜の認識率を従来よりも向上させることができる。

40

【発明を実施するための最良の形態】

【0009】

以下、図面を参照しながら、本発明の一実施形態について説明する。

図1は、楽譜認識装置の構成の一例を示すブロック図である。

図1において、楽譜認識装置100は、CPU1、ROM2、RAM3、HDD4、FDD5、MIDI（Musical Instrument Digital Interface）インタフェース回路6、スキャナインタフェース回路7、プリンタインタフェース回路8、キーボードインタフェース回路9、CRTインタフェース回路10、及びバス15を有している。そして、スキャナ11、プリンタ12、キーボード13、及びCRT14が楽譜認識装置100に接続さ

50

れている。

【0010】

図1に示すように、本実施形態の楽譜認識装置100は、パソコン等の一般的な計算機システムに、MIDIインタフェース回路6を付加したものである。

CPU1は、ROM2に格納されたプログラムをRAM3に展開して実行すること等により、楽譜認識装置100の全体の制御を行う中央処理装置である。また、楽譜認識装置100は、予め設定された所定の周期でCPU1に割り込みをかけるタイマ回路を内蔵している。RAM3は、プログラムエリアの他、画像データバッファ、ワークエリア等として使用される。HDD（ハードディスクドライブ）4及びFDD（フレキシブルディスクドライブ）5は、プログラム、画像データ、及び演奏データ等を格納する。

10

【0011】

CRT14は、CPU1の制御に基づき、CRTインタフェース回路10から出力される映像情報を表示する。キーボード13から入力された情報は、キーボードインタフェース回路9を経てCPU1に取り込まれる。プリンタ12は、CPU1の制御に基づき、プリンタインタフェース回路8から出力される印字情報を印字する。

【0012】

スキャナ11は、紙面に印刷された楽譜を光学的に走査して、2値、グレイスケール、あるいはカラーの画像データに変換するものである。スキャナ11としては、フラットベッド型、ハンディ型、フィーダー型等任意のタイプのものを使用できる。スキャナ11によって読み取られた画像情報は、スキャナインタフェース回路7を介して、RAM3又はHDD4に取り込まれる。MIDIインタフェース回路6は、音源モジュール等の外部のMIDI機器との間でMIDIデータの送受信を行う回路である。バス15は、楽譜認識装置100内の各回路を接続している。尚、この他に、マウス等のポインティングデバイスや、RS232C等のシリアルインターフェース回路等を楽譜認識装置100が備えていてもよい。

20

【0013】

図2は、CPU1の処理の一例を説明するメインフローチャートである。

まず、図2-1のステップS1において、CPU1は、スキャナ12によって読み取られた楽譜の画像データをRAM3に取り込む。ここでは、画像を2値の画像として取り込むとする。このように本実施形態では、ステップS1の処理を行うことにより第1の取得手段が実現される。

30

次に、ステップS2において、CPU1は、かすれやドットノイズ等を軽減するために、楽譜の画像データに対して、図形融合等の画質平滑化処理を行う。

次に、ステップS3において、CPU1は、楽譜の画像データに対して、画質チェック処理を行う。画質チェック処理においては、楽譜の画像データにおける"倍率の情報と濃度の情報"を得ると共に、後述する五線検出の基準データを得るために、五線の線幅と五線の各線間の間隔とを求める。ここでは、線幅と間隔とを求めるために、画像上の横(x)方向の数箇所において縦(y)方向に走査し、黒ラン（連続する黒画素）と白ランの長さを全て求め、求めた長さ毎に頻度分布（ヒストグラム）データを作成する。ここで、ランとは、黒画素ないし白画素がある方向に連続する長さのことをいう。

40

【0014】

楽譜上で最も多い音楽記号は五線であるので、作成された黒ランの長さのヒストグラムのピークと、白ランの長さのヒストグラムのピークとを夫々検出することで、五線の線幅と間隔とを推定できる。

また、楽譜の画像の倍率は、例えば、五線の間隔から推定できる。更に、楽譜の画像の濃度は、五線の線幅と間隔との比から推定できる。楽譜の認識処理においては、楽譜の画像の"倍率及び濃度"が所定の範囲から外れると、楽譜の画像の認識率が低下してしまう。そこで、ステップS4において、CPU1は、これらの値が、所定の範囲内にあるか否かを判定することにより、画質がOKか否かを判定する。この判定の結果、楽譜の画像の"倍率及び濃度"が所定の範囲から外れており、画質がOKでない場合には、ステップS1

50

に戻り、倍率や濃度を変えて、楽譜の画像を再度読み取る。

【0015】

一方、楽譜の画像データの"倍率及び濃度"が所定の範囲内にあり、画質がOKである場合には、ステップS5に進む。ステップS5に進むと、CPU1は、楽譜の画像データに対して、五線認識処理を行う。五線認識処理は、五線走査開始位置検出処理と、五線シフト量検出処理とに分かれる。五線走査開始位置検出処理の概略を述べると、横(x)軸方向のある位置で、黒画素の幅と白画素の幅とを順に求め、求めた幅に基づいて、五線状に並んでいる位置を、ある程度の誤差を考慮して検出する。そして、加線(五線からはみ出した音符を記載するために付加した横線)の影響を除くために、五線状の並びの両側に、五線の間隔より大きな白画素幅があるという条件を加える。この条件に合う白黒画素の並びがある横(x)方向の各黒ランの中点を五線走査開始位置とする。

10

【0016】

つぎに、五線シフト量検出処理の概略を述べる。まず、五線走査開始位置(5点の黒画素位置)から、1ドットずつ位置を左右に変えていき、5点の内、黒画素がある個数(例えば3あるいは4個)以下になった場合に、5点を上下にずらして黒画素数をチェックし、縦(y)方向のうち、黒画素の割合が高くなる方向へy座標をシフトする。この五線走査開始位置からのシフト量を五線のシフト量とする。そして、五線走査開始位置の左右方向(横方向)に、黒画素の個数が0になる位置まで走査することにより五線の検出を行う。

【0017】

20

次に、ステップS6において、CPU1は、楽譜の画像データに対して、段落認識処理を行う。この段落認識処理は、段落認識処理と、大かっこ認識処理とに分かれる。段落認識処理においては、画像全体で五線を検出し、五線同士で左端がほぼ同じ場所にある五線の組を探し、五線の端同士が、黒画素で結ばれているかどうかを検査し、段落を認識する。尚、予め、x軸及びy軸方向における黒画素のヒストグラムを求め、このヒストグラムに基づいて空白部分を検出することによって段落の存在を推定しても良い。五線同士が大かっこで結ばれていた場合には、五線同士にまたがる音符等が存在する場合があるので、大かっこで結ばれた五線は1つの単位で処理を行った方が良い。大かっこについては、段落線の左の所定の範囲で、後述する定型記号認識と同様の手法で認識を行う。

【0018】

30

次に、ステップS7において、CPU1は、パート名認識処理を行う。このパート名認識処理においては、文字列認識を行う。まず、アルファベットやその他の記号を認識し、その記号を囲む矩形が文字列状に並んでいるものを抽出する。そして、これと文字列辞書とのマッチングをとることで、文字列状の記号を認識する。

【0019】

次に、ステップS8において、CPU1は、パートテンプレート適用処理を行う。このパートテンプレート適用処理においては、後述するステップS13～S16における認識が終了した後に設定が必要な情報が記憶されたパートテンプレートを、HDD4等に予め記憶されている複数のパートテンプレートの中から選び出す。

図3は、パートテンプレートの一例を示す図である。図3において、パートテンプレート301は、テンプレート名302、パート名303、パート名短縮304、移調楽器設定305、音部記号306、タブ譜線数307、及びタブ譜チューニング情報308を有する。

40

【0020】

テンプレート名302には、パートテンプレート301を識別するための文字列として、例えば、日本語の文字列又はアルファベットの文字列が格納される。

パート名303には、パートテンプレート301が適用されるパートを識別するための文字列が格納される。

パート名短縮304には、パートテンプレート301が適用されるパートの短縮表記を識別するための文字列が格納されている。

50

移調楽器設定305には、パートテンプレート301が適用される楽器が移調楽器であるか否かと、その楽器が移調楽器である場合にはその移調楽器がどのようなものであるかとの識別するための数値が格納されている。本実施形態では、パートテンプレート301が適用される楽器が移調楽器でない場合、移調楽器設定305には、0が格納される。また、本実施形態では、パートテンプレート301が適用される楽器がトランペット、ウッドベース、ピッコロ、アルトサックスである場合、移調楽器設定305には、夫々、-2、-12、12、-9が格納される。

【0021】

音部記号306には、パートテンプレート301が適用される音部記号の種類に対応した数値が格納されている。本実施形態では、パートテンプレート301が適用される音部記号がト音記号である場合、音部記号306には0が格納される。また、本実施形態では、パートテンプレート301が適用される音部記号がヘ音記号である場合、音部記号306には1が格納される。

10

タブ譜線数307には、パートテンプレート301が適用される"楽譜の線数"を表す数値が格納されている。

タブ譜チューニング情報308には、楽譜の各線に対するチューニングを表す数値が格納されている。タブ譜チューニング情報308には、パートテンプレート301が適用される"楽譜の線数"分の情報が格納されている。

【0022】

尚、パートテンプレート301には、移調楽器設定305、音部記号306、タブ譜線数307、及びタブ譜チューニング情報308が少なくとも含まれていればよい。ただし、テンプレート名302、パート名303は含まれていた方が好ましい。また、テンプレート名302、パート名303、パート名短縮304、移調楽器設定305、音部記号306、タブ譜線数307、及びタブ譜チューニング情報308に格納される情報は、前述したものに限定されるものではない。

20

また、このステップS8におけるパートテンプレート適用処理の詳細については後述する。

【0023】

次に、ステップS9において、CPU1は、ステップS6の段落認識処理の結果と共に、ステップS7のパートテンプレート適用処理の結果をCRT14に表示させて、段落の認識結果とパートテンプレート301の適用結果とが正しいか否かをユーザにチェックさせることにより、認識がOKか否かを判定する。この判定の結果、認識がOKでない場合にはステップS10に進み、CPU1は、段落の認識結果や、パートテンプレート301の適用結果の修正を行う認識修正処理を行った後、ステップS9に戻る。スコア譜においては、各段落のパート構成が等しいものの他に、途中でパートの省略や追加があったり、同じパートで単独譜表と大譜表とが段落ごとに変化したりする場合もある。このようなパートの対応は、大かっこの対比等で行うが、パートの対応が一意に決められない場合もあるので、段落の認識結果の修正を予め行えるようにする。また、パートテンプレート301をユーザの好みのものにしたい場合があるので、ステップS7で自動的に選択したパートテンプレート301を変更することができるようにする。この認識修正処理（パートテンプレート301の適用結果の修正）の詳細については後述する。

30

40

以上のように本実施形態では、ステップS9の処理を行うことにより表示手段の一部が実現される。

【0024】

尚、五線認識が失敗した場合には、その後の処理が行えないので、倍率や濃度を変更して再度画像を取り込む必要がある。従って、ステップS9において、まず五線の認識結果をCRT14に表示し、表示した"五線の認識結果"に対するユーザの操作の結果に基づいて、五線の認識結果が正しいか否かを判定するようにしてもよい。このようにした場合、五線の認識結果が正しくないと判定されると、ステップS1に戻り、楽譜の画像データを再度読み取るようにする。一方、五線の認識結果が正しいと判定された場合には、段落の

50

認識結果を表示し、その段落の認識結果をユーザにチェックさせるようにするのが好ましい。

【0025】

一方、ステップS9において、認識がOKであると判定された場合には、CPU1は、ステップS1で得られた楽譜に含まれるパートの1つを選択し、図2-2のステップS11に進む。ステップS11に進むと、CPU1は、五線傾き補正処理を行う。五線傾き補正処理の概略を述べると、先に求めた五線のシフト量に基づいて、認識処理矩形の列毎に画素列を上下にシフトする。尚、五線毎にシフト量を計算し、認識処理矩形の画像内でシフト補正を行った方がより正確であるが、このシフト量を、ステップS1で読み取られた楽譜の画像全体で1つ計算し、楽譜の画像の全体をシフトしても良い。

10

この後、CPU1は、認識処理矩形の上下端に接したラベルを、上下のパートの構成要素として消去する。最後に、CPU1は、上下端の空白部分を検出して、認識処理矩形を縮小する。ここで、ラベルとは、他の画素領域とは独立した画素領域をいう。例えば、白画素と黒画素とからなる2値画像における黒画素のラベルについて説明する。この場合、黒画素の上下左右、斜め方向の画素の連結状態を検出し、検出した結果に基づき1つの塊として判定できる黒画素の集まりをラベルと呼ぶ。尚、上下左右に加えて斜め方向の画素の連結を判定する場合を8連結ラベルと称し、上下左右のみの画素の連結を判定する場合を4連結ラベルと称する。また、本実施形態では、例えば黒画素が連結している部分だけでなく、その黒画素が連結している部分を囲む多角形（矩形等）の領域もラベルとして扱うものとする。

20

次に、ステップS12において、CPU1は、五線消去処理を行う。この五線消去処理においては、ステップS5で認識され、ステップS11で傾きが補正された五線を消去する。

【0026】

ステップS13～S16においては、各種の音楽記号の認識処理が行われる。楽譜の音楽記号には、形、位置に関して大まかに以下の3つの種類がある。

(1) 定型で、上下位置がほぼ決まっているもの（音部記号、拍子記号等）。

(2) 定型で、上下位置は自由度があるもの（臨時記号、休符等）。

(3) 不定型且つ位置も不定のもの（音符、スラー、タイ等）。

本実施形態では、これらをそれぞれに適した方式で、音部記号・拍子認識、音符認識、定型記号認識、文字列認識、スラー・タイ認識の順に認識する。尚、音楽記号は、これらに限定されるものではなく、楽譜に記載されるものであれば、どのようなものであってもよい。

30

【0027】

音部記号・拍子認識を最初に行うのは、処理コストの低い認識を最初に行って、この記号を消去することによって、後の認識の処理コストを軽減するためである。更に、最初により確実なものを認識することで、後の認識での誤認識を減らすためでもある。また、音符認識の後に定型記号認識を行うのは、ラベルの接触に影響されにくい認識方式である音符認識を行って、この音符を消去することで、音符に接触した臨時記号等の認識を可能にするためである。スラー・タイ認識が最後になっているのは、処理コストの高いスラー認識の対象になるラベルをなるべく少なくするためである。また、先に検出された音符の周りのラベルだけをスラー・タイ認識の対象にすることによって、更にスラー・タイ認識の処理コストを下げ、且つ、スラー及びタイの誤認識も減らすことができる。

40

【0028】

まず、ステップS13において、CPU1は、五線に対して定位置にある音楽記号として、音部記号と拍子記号とを認識する音部記号・拍子認識処理を行う。ここでは、まず、求められた五線を含む矩形領域において、縦方向における黒画素のヒストグラムを求め、黒画素の量がしきい値以上の帯域を、音部記号又は拍子記号が存在する可能性のある場所としてマッチングの対象とする。マッチングは、例えば、五線間の数箇所について求められた"横方向のペリフェラル特徴"に基づいて行われる。ペリフェラル特徴とは、マッピン

50

グの対象となる記号のみを含む矩形領域の左右端から五線間の数箇所の白画素の領域に向けて内方向に走査し、黒画素の領域に達するまでの距離を1次（最初）あるいは数次（2回目以降）まで求めたものである。また、マッチングに失敗した場合には、隣接した帯域を併合して再度認識を行う。そして、認識された音楽記号は画像データから消去する。

【0029】

次に、ステップS14において、CPU1は、音符認識処理を行う。この音符認識処理においては、まず、画像データを縦細線、横細線、及び太線画像に分離する。そして、各画像から記号や記号の部品を検出し、それらを総合して音符等の記号を認識する。

次に、ステップS15において、CPU1は、定型記号認識処理を行う。この定型記号認識処理においては、まず、公知の輪郭線加重方向指数を取り、辞書の各記号データについてラベルのサイズと輪郭線加重方向指数のマッチング度を計算して、各マッチング度を正規化し、統合した結果が最も高い記号を出力する。尚、サイズと加重方向指数の他に、ペリフェラル等の他の特徴を使っても良い。また、五線消去によりラベルが切れたものの対策として、五線消去によって切れたラベルを辞書に登録し、音楽記号であると認識された場合には、その周りのラベルを結合して再認識する。

【0030】

次に、ステップS16において、CPU1は、その他の記号認識処理を行う。本実施形態では、その他の記号認識処理において、文字列認識処理と、スラー・タイ認識処理とを含む記号認識処理を行う。

文字列認識処理では、まず、速度記号等の文字列を認識するために、定型記号認識処理で認識されたアルファベットやその他の記号を使い、その記号を囲む矩形が文字列状に並んでいるものを抽出する。そして、これと文字列辞書とのマッチングをとることで、文字列状の記号を認識する。

【0031】

次に、CPU1は、スラー・タイ認識処理を行う。このスラー・タイ認識処理においては、まず、これまでの処理で残ったラベルのうち、検出された音符の周りのラベルを細線化し、多円弧近似する。そして、以前に消された記号により線が切れている場合があるので、求められた多円弧同士の間接を行う。最後に、求められた円弧の形や、元画像の図の太さや、音符との関係等からスラーやタイを認識する。

【0032】

本実施形態では、以上のステップS13～S16の処理において、CPU1は、音楽記号の認識に失敗した場合、その音楽記号の認識の際に候補に挙げた音楽記号を識別するためのIDを、マッチング度の高いものから順に、所定個数RAM3等に記憶するようにしている。

図4は、認識の際に候補に挙げた音楽記号を識別するためのIDの記憶方法の一例を概念的に示す図である。

図4に示すように、ステップS1で得られた楽譜の画像から抽出された音楽記号401に対して、辞書マッチング402が行われる。この辞書マッチング402においてCPU1は、音楽記号401に対するマッチング度（得点）が閾値以上である音楽記号を辞書から得ることができた場合には、音楽記号401の認識に成功したと判定する。更に、CPU1は、音楽記号401に対するマッチング度（得点）が閾値以上でない音楽記号のうち、マッチング度（得点）が高い音楽記号から順に、その音楽記号を識別するためのID403、404を、例えば、RAM3の所定の記憶領域に記憶する。

【0033】

一方、音楽記号401に対するマッチング度（得点）が閾値以上である音楽記号を辞書から得ることができなかった場合、CPU1は、マッチング度（得点）が高い音楽記号から順に、その音楽記号を識別するためのID405、406を、例えば、RAM3の所定の記憶領域に記憶する。本実施形態では、マッチング度（得点）が閾値以上でない音楽記号のうち、マッチング度（得点）が最も高い音楽記号を識別するためのID403、405を次点IDと称し、その次点ID403、405の次にマッチング度（得点）が高い音

10

20

30

40

50

楽記号を識別するための I D 4 0 4、4 0 6 を次次点 I D と称する。

以上のように本実施形態では、ステップ S 1 3 ～ S 1 6 の処理を行うことにより認識手段が実現される。

【0034】

次に、ステップ S 1 7 において、C P U 1 は、パート内誤認識修正処理を行う。パート内認識修正処理においては、ステップ S 1 で得られた楽譜に含まれるパートのうち、1つのパート内だけで誤認識であると判別することができる部分を、ステップ S 8 で選択されたパートテンプレート 3 0 1 を用いて修正する。このパート内誤認識修正処理の詳細については後述する。

次に、ステップ S 1 8 において、C P U 1 は、認識記号のラベル消去処理を行う。認識記号のラベル消去処理においては、C P U 1 は、ステップ S 1 3 ～ S 1 6 で認識された音楽記号や、ステップ S 1 7 で認識されたスラーやタイ等を画像から消去する。

【0035】

次に、ステップ S 1 9 において、C P U 1 は、パート内設定処理を行う。パート内設定処理においては、ステップ S 1 3 ～ S 1 7 で認識された音楽記号に基づいて、各小節に含まれる音楽記号の設定（小節属性の設定）を行ったり、音楽記号の親子関係の設定を行ったりする。音楽記号の親子関係の設定は、例えば、音符との位置関係が適切な臨時記号を、その音符の臨時記号として設定したりすることにより行われる。尚、音符との位置関係が適切でない臨時記号については、誤認識したとして、認識結果から削除したり、五線や小節との位置関係に応じて調号としたりすることができる。また、ステップ S 1 9 では、ステップ S 1 で得られた楽譜に含まれるパートのうち、1つのパート内の情報で設定できるものだけを対象として処理を行う。

【0036】

次に、ステップ S 2 0 において、C P U 1 は、ステップ S 1 で得られた楽譜に含まれるパートの全てに対して、ステップ S 1 1 ～ S 2 0 の処理を行ったか否かを判定する。この判定の結果、ステップ S 1 で得られた楽譜に含まれるパートの全てに対して処理を行っていない場合には、ステップ S 1 1 に戻り、ステップ S 1 で得られた楽譜に含まれるパートの全てに対して処理を行うまでステップ S 1 1 ～ S 2 0 を繰り返し行う。

【0037】

次に、ステップ S 2 1 において、C P U 1 は、パート間設定処理を行う。ステップ S 1 9 で行ったパート内設定処理では、ステップ S 1 で得られた楽譜に含まれるパートのうち、1つのパート内の情報で設定できるものだけを対象として処理を行うが、パート間設定処理では、ステップ S 1 で得られた楽譜に含まれる複数のパートの情報を用いなければ設定できないものだけを対象として処理を行う。具体的には、小節属性の設定や、音楽記号の親子関係の設定等を行う。

【0038】

次に、ステップ S 2 2 において、C P U 1 は、パート間認識修正処理を行う。ステップ S 1 7 で行ったパート内認識修正処理では、ステップ S 1 で得られた楽譜に含まれるパートのうち、1つのパート内だけで誤認識であると判別することができる部分を修正するが、パート間認識修正処理では、ステップ S 1 で得られた楽譜に含まれる複数のパートを考慮しなければ誤認識であると判別することができない部分を、ステップ S 8 で選択されたパートテンプレート 3 0 1 を用いて修正する。このパート間誤認識修正処理の詳細については後述する。

以上のように本実施形態では、パート内誤認識修正処理（ステップ S 1 7）と、パート間誤認識修正処理（ステップ S 2 2）とを別々に行うようにして処理の効率化を図るようにしている。

【0039】

次に、ステップ S 2 3 において、C P U 1 は、例えば、ステップ S 1 ～ S 2 2 の処理結果に基づき、ステップ S 1 で読み取られた楽譜の画像データ（以下、必要に応じて楽譜画像データと称する）を生成して、C R T 1 4 に表示させる。そして、その楽譜画像データ

が正しいか否かをユーザにチェックさせることにより、その楽譜画像データがOKか否かを判定する。この判定の結果、楽譜画像データがOKでない場合には、ステップS24に進む。そして、CPU1は、ユーザによるマウスやキーボード13等の操作に基づいて、音楽記号の認識結果を修正する。そして、ステップS23に戻る。

一方、ステップS23において、楽譜画像データがOKであると判定された場合には、ステップS24を省略してステップS25に進む。

【0040】

ステップS25に進むと、CPU1は、演奏データ作成処理を行う。この演奏データ作成処理においては、認識及び修正した各種の音楽記号（音符情報、音部記号情報、拍子記号情報、定型記号情報、文字列情報、スラー・タイ情報）に基づいて、例えば公知の演奏データ形式であるMIDIファイルデータを、演奏データとして生成する。そして、図2のフローチャートによる処理を終了する。

10

【0041】

次に、図5のフローチャートを参照しながら、図2のステップS8におけるパートテンプレート適用処理の一例を説明する。

まず、ステップS101において、CPU1は、図2のステップS7で認識されたパート名に対応する楽器IDを、HDD4等に予め登録されているデータベースから取得する。楽器IDとは、演奏するための手段を識別するためのIDであり、楽器を識別するためのIDの他に、例えば合唱（の1パート）であることを識別するためのID等も楽器IDに含まれる。このステップS101では、例えば、MIDIのプログラムナンバーに対応した楽器IDを取得することができる。また、ステップS1で得られた楽譜がタブ譜であることがステップS5の五線認識処理で分かった場合には、そのタブ譜に応じた楽器IDを取得することができる。

20

【0042】

次に、ステップS102において、CPU1は、ステップS101で取得した楽器IDに基づいて、ステップS1で得られた楽譜のジャンルを推定することができるか否かを判定する。例えば、"Contralto"に対応する楽器IDが取得された場合、CPU1は、その楽器IDを参照するだけで、ステップS1で得られた楽譜のジャンルは合唱であると推定することができる。この判定の結果、ステップS1で得られた楽譜のジャンルを推定することができない場合には、ステップS103に進む。ステップS103に進むと、CPU1は、ステップS5～S7における処理の結果等に基づいて、ステップS1で得られた楽譜のジャンルを推定する。そして、ステップS104に進む。

30

【0043】

一方、ステップS1で得られた楽譜のジャンルを推定することができる場合には、ステップS103を省略してステップS104に進む。

以上のようにしてステップS104に進むと、CPU1は、ステップS101で取得した楽器IDと、前述したようにして推定したジャンルとに対応するパートテンプレート301が、認識パートテンプレート適用表にあるか否かを判定する。

図6は、認識パートテンプレート適用表の一例を示す図である。図6に示す認識パートテンプレート適用表601は、例えばHDD4に予め記憶されている。

40

図6において、認識パートテンプレート適用表601には、楽器IDとジャンルとに対応付けられて、パートテンプレート301を識別するための情報（テンプレート名302に格納されている情報）が記憶されている。尚、図6において、「設定なし」は、その部分に該当するパートテンプレート301がないことを示す。また、認識パートテンプレート適用表601をユーザが自由に編集する場合には、ユーザが認識し易い情報を、パートテンプレート301を識別するための情報として記憶するのが好ましい。このようにすれば、認識パートテンプレート適用表601からパートテンプレート301をユーザが容易に検索できるからである。

【0044】

ステップS104の判定の結果、該当するパートテンプレート301が認識パートテン

50

プレート適用表601にある場合には、後述するステップS107に進む。一方、該当するパートテンプレート301が認識パートテンプレート適用表601にない場合には、ステップS105に進む。

ステップS105に進むと、CPU1は、ステップS101で取得した楽器IDと、前述したようにして推定したジャンルとは異なるジャンルとに対応するパートテンプレート301が、認識パートテンプレート適用表601にあるか否かを判定する。この判定の結果、該当するパートテンプレート301が認識パートテンプレート適用表601にある場合には、後述するステップS107に進む。

【0045】

一方、該当するパートテンプレート301が認識パートテンプレート適用表601にない場合、認識パートテンプレート適用表601には、ステップS101で取得した楽器IDに対応するパートテンプレート301が登録されていない（ステップS101で取得した楽器IDに対応する欄が全て「設定なし」である）ので、ステップS106に進む。そして、ステップS106において、CPU1は、パートテンプレートの適用ができない旨をCRT14に表示させる。そして、図2のステップS9に進む。このようにしてステップS9に進んだ場合、ステップS7のパートテンプレート適用処理の結果は表示されないことになる。

【0046】

一方、該当するパートテンプレート301が認識パートテンプレート適用表601にある場合には、ステップS107に進む。ステップS107に進むと、CPU1は、該当するパートテンプレート301（の情報）を取得する。そして、図2のステップS9に進む。このようにしてステップS9に進んだ場合、ステップS6の段落認識処理の結果と共に、ステップS7のパートテンプレート適用処理の結果が表示されることになる。

図7は、パートテンプレート適用処理の結果の表示内容の一例を示す図である。図7に示すように、本実施形態では、パートナンバ（パートNo）と、パート名と、パートテンプレート名とが互いに関連付けられたテーブル701を、パートテンプレート適用処理の結果として、段落認識処理の結果と共に表示する。本実施形態では、テーブル701におけるパートテンプレート名の内容は、ユーザによる操作に基づいて変更することが可能になっている。

以上のように本実施形態では、ステップS101～S105、S107の処理を行うことにより第2の取得手段が実現される。

尚、ステップS105において、ステップS101で取得した楽器IDと、推定したジャンルとは異なるジャンルとに対応するパートテンプレート301が認識パートテンプレート適用表601に複数ある場合、例えば、他のジャンルを、優先順位をつけて順に選出し、そのジャンル内でパートテンプレート301を検索することにより、CPU1は、推定したジャンルに最も近いジャンルに対応するパートテンプレート301を取得することができる。

【0047】

次に、図8のフローチャートを参照しながら、図2のステップS10の認識修正処理の一例を説明する。

まず、ステップS201において、CPU1は、ユーザによる操作に基づいて、図2のステップS8のパートテンプレート適用処理で取得されたパートテンプレート301を変更するための操作がユーザによってなされたか否かを判定する。この判定の結果、パートテンプレート301を変更するための操作がユーザによってなされていない場合には、ステップS205に進み、CPU1は、その他の処理を行う。その他の処理としては、例えば、段落の認識結果の修正等をユーザの操作に基づいて行う処理が挙げられる。

一方、パートテンプレート301を変更するための操作がユーザによってなされた場合には、ステップS202に進む。

以上のように本実施形態では、ステップS201の処理を行うことにより、受付手段が実現される。

10

20

30

40

50

【0048】

ステップS202に進むと、CPU1は、その操作に従ってパートテンプレート301を変更しても、パートテンプレート301が適用されるMIDIのプログラムナンバーが変更しないかどうかを判定する。この判定の結果、MIDIのプログラムナンバーが変更する場合には、後述するステップS206に進む。一方、MIDIのプログラムナンバーが変更しない場合には、ステップS203に進む。

【0049】

そして、ステップS203に進むと、CPU1は、ステップS201で判定された操作の内容に従って、認識パートテンプレート適用表601の内容（パートテンプレート301を識別するための情報）を変更（更新）する。次回以降は、ここで変更された内容に従って、パートテンプレート301が取得されることになる（図5のステップS107を参照）。このように本実施形態では、ステップS203の処理を行うことにより記憶手段の一部が実現される。

10

次に、ステップS204において、CPU1は、認識パートテンプレート適用表601の変更結果に基づいて、パートテンプレート適用処理の結果の表示内容（図7に示したテーブル701のパートテンプレート名の内容）を変更させる。このように本実施形態では、ステップS204の処理を行うことにより表示手段の一部が実現される。

そして、ステップS205に進み、CPU1はその他の処理を行った後、図2のステップS9に戻る。

【0050】

20

ステップS202において、MIDIプログラムナンバーが変更すると判定された場合には、ステップS206に進み、CPU1は、認識パートテンプレート適用表601の更新ができない旨をCRT14に表示させる。そして、ステップS205に進み、CPU1はその他の処理を行った後、図2のステップS9に戻る。

以上のように本実施形態では、MIDIプログラムナンバーが変更するような認識パートテンプレート適用表601の変更を受け付けないようにしている。すなわち、パートテンプレート301の種類をユーザの好みに合わせて単に変更したい場合に認識パートテンプレート適用表601を変更できるようにし、パート名の認識に誤認識があった可能性がある場合には、認識パートテンプレート適用表601の変更を制限（変更できないように）している。

30

【0051】

例えば、認識パートテンプレート適用表601において、「トランペット」に適用されるパートテンプレート301を、「ソロトランペット」に適用されるパートテンプレート301に変更することができるようにしている。これに対し、「トロンボーン」に適用されるパートテンプレート301を「チューバ」に適用されるパートテンプレート301に変更することはできないようにしている。「Tub」を「Tb」に誤認識した可能性があるからである。

【0052】

もし、「トロンボーン」に適用されるパートテンプレート301を「チューバ」に適用されるパートテンプレート301に変更することを認めてしまうと、次のような問題点がある。すなわち、次回以降の楽譜の認識において、「トロンボーン」と認識した場合には、「チューバ」に適用されるパートテンプレート301が、認識パートテンプレート適用表601から取得されてしまう可能性が生じてしまうからである。

40

以上のように本実施形態では、ステップS202、S206の処理を行うことにより、制限手段が実現される。

尚、本実施形態では、MIDIプログラムナンバーが変更するような認識パートテンプレート適用表601の変更を受け付けないようにして、前述したような問題点を解消するようにしているが、必ずしもこのようにする必要はない。例えば、MIDIプログラムナンバーが変更するような認識パートテンプレート適用表601の変更を受け付けるようにしてもよい。また、前述したのと異なる方法によって、場合によってはMIDIプログラ

50

ムナンバーが変更するような認識パートテンプレート適用表601の変更を受け付けないようにしてもよい。更に、認識パートテンプレート適用表601の内容を、ユーザの操作に基づいて初期状態に戻せるようにしてもよい。

【0053】

次に、図9のフローチャートを参照しながら、図2のステップS17のパート内誤認識修正処理の一例を説明する。

まず、ステップS301において、CPU1は、図5のフローチャートにおいて、パートテンプレート301（の情報）を取得できたか否かを判定する。この判定の結果、パートテンプレート301を取得できなかった場合には、図2のステップS19に戻る。

一方、パートテンプレート301を取得できた場合には、ステップS302に進み、CPU1は、取得したパートテンプレート301（の情報）を用いて、オクターブ付き音部記号修正処理を行う。図10に、オクターブ付き音部記号1001の一例を示す。オクターブ付き音部記号修正処理においては、パートテンプレート301（の情報）を用いて、図2のステップS13で認識された音部記号を修正する。尚、オクターブ付き音部記号修正処理の詳細については後述する。

【0054】

次に、ステップS303において、CPU1は、オクターブ付き音部記号修正処理以外のパート内誤認識修正処理を行う。前述したようにパート内誤認識修正処理は、1つのパート内だけで誤認識であると判別することができる部分を、パートテンプレート301を用いて修正する処理である。そして、図2のステップS18に戻る。

以上のように本実施形態では、図9のフローチャートを実行することにより、設定手段の一部が実現される。

【0055】

次に、図11のフローチャートを参照しながら、図9のステップS302のオクターブ付き音部記号修正処理の一例を説明する。

まず、ステップS401において、CPU1は、図5のステップS107で取得されたパートテンプレート301の音部記号306に、オクターブ付きの音部記号に対応する数値が格納されているか否かを判定する。この判定の結果、図5のステップS107で取得されたパートテンプレート301の音部記号306に、オクターブ付きの音部記号に対応する数値が格納されていない場合には、図9のステップS303に戻る。このように本実施形態では、パートテンプレート301の音部記号306に、オクターブ付きの音部記号に対応する数値が格納されていない場合には、図3のフローチャートに戻るようにすることにより、オクターブ付き音部記号修正処理を高速化するようにしている。

【0056】

一方、図5のステップS107で取得されたパートテンプレート301の音部記号306に、オクターブ付きの音部記号に対応する数値が格納されている場合には、ステップS402に進む。

ステップS402に進むと、CPU1は、図2のステップS13における音部記号の認識の結果、他の音部記号と分離された「8」が認識された部分があるか否かを判定する。この判定の結果、他の音部記号と分離された「8」が認識されていない場合には、後述するステップS407に進む。

【0057】

一方、他の音部記号と分離された「8」が認識された部分がある場合には、ステップS403に進む。ステップS403に進むと、CPU1は、ステップS1で得られた楽譜の全段落のN（Nは正の値）[%]以上で、他の音部記号と分離された「8」が認識されたか否かを判定する。この判定の結果、ステップS1で得られた楽譜の全段落のN[%]以上で、他の音部記号と分離された「8」が認識されていない場合には、図9のステップS303に戻る。

【0058】

一方、ステップS1で得られた楽譜の全段落のN[%]以上で、他の音部記号と分離さ

10

20

30

40

50

れた「8」が認識されている場合には、ステップS404に進む。ステップS404に進むと、CPU1は、図2のステップS13における音部記号の認識の結果に基づいて、「8」の認識に失敗した部分があるか否かを判定する。この判定の結果、「8」の認識に失敗した部分がない場合には、図9のステップS303に戻る。

【0059】

一方、「8」の認識に失敗した部分がある場合には、ステップS405に進む。ステップS405に進むと、CPU1は、「8」の認識に失敗した部分の次点ID405が、オクターブ付きの音部記号を識別するIDであるか否かを判定する。この判定の結果、「8」の認識に失敗した部分の次点ID405が、オクターブ付きの音部記号を識別するIDでない場合には、図9のステップS303に戻る。

10

【0060】

一方、「8」の認識に失敗した部分の次点ID405が、オクターブ付きの音部記号を識別するIDである場合には、ステップS406に進む。ステップS406に進むと、CPU1は、「8」の認識に失敗した部分に「8」を追加して、オクターブなしの音部記号をオクターブ付きの音部記号に変更する。そして、図9のステップS303に戻る。

【0061】

ステップS402において、他の音部記号と分離された「8」が認識されていないと判定された場合には、オクターブ付きでないものとして音部記号の認識に成功していると判定され、ステップS407に進む。ステップS407に進むと、CPU1は、認識に成功した音部記号の次点ID403が、オクターブ付きの音部記号を識別するIDであるか否かを判定する。この判定の結果、認識に成功した音部記号の次点ID403が、オクターブ付きの音部記号を識別するIDでない場合には、図9のステップS303に戻る。

20

【0062】

一方、認識に成功した音部記号の次点ID403が、オクターブ付きの音部記号を識別するIDである場合には、ステップS408に進む。ステップS408に進むと、CPU1は、認識に成功した音部記号に「8」を追加して、オクターブなしの音部記号をオクターブ付きの音部記号に変更する。そして、図9のステップS303に戻る。この場合、図2のステップS13では、オクターブなしの音部記号であるとして認識に成功しているが、図5のステップS107で取得されたパートテンプレート301の音部記号306に基づく、その音部記号は実はオクターブ付きの音部記号であったということになる。

30

【0063】

尚、図11では、ステップS405において、「8」の認識に失敗した部分の次点ID405が、オクターブ付きの音部記号を識別するIDでないと判定された場合には、図11の処理を終了させるようにした。しかしながら、必ずしもこのようにする必要はない。例えば、「8」の認識に失敗した部分の次点ID405が、オクターブ付きの音部記号を識別するIDでないと判定された場合には、「8」の認識に失敗した部分の次次点ID406が、オクターブ付きの音部記号を識別するIDであるか否かを判定する。そして、この判定の結果、次次点ID406が、オクターブ付きの音部記号を識別するIDである場合にはステップS406に進み、そうでない場合には図11の処理を終了させるようにしてもよい。

40

【0064】

次に、図12のフローチャートを参照しながら、図2のステップS22のパート間誤認識修正処理の一例を説明する。

まず、ステップS501において、CPU1は、図5のフローチャートにおいて、パートテンプレート301（の情報）を取得できたか否かを判定する。この判定の結果、パートテンプレート301を取得できなかった場合には、図2のステップS23に戻る。

一方、パートテンプレート301を取得できた場合には、ステップS502に進み、CPU1は、取得したパートテンプレート301（の情報）を用いて、調号修正処理を行う。調号修正処理においては、パートテンプレート301（の情報）を用いて、図2のステップS15で認識された調号を修正する。尚、調号修正処理の詳細については後述する。

50

【0065】

次に、ステップS503において、CPU1は、取得したパートテンプレート301（の情報）を用いて、音部記号修正処理を行う。音部記号修正処理においては、パートテンプレート301（の情報）を用いて、図2のステップS13で認識された音部記号を修正する。尚、前述した図9のステップS302（図11）のオクターブ付き音部記号修正処理では、1つのパートの情報のみから、音部記号を修正するが、このステップS503の音部記号修正処理は、複数のパートの情報から、音部記号を修正する。このステップS503における音部記号修正処理の詳細については後述する。

【0066】

そして、ステップS504に進み、CPU1は、調号修正処理及び音部記号修正処理以外のパート間誤認識修正処理を行う。前述したようにパート間誤認識修正処理は、複数のパートの情報をを用いないと誤認識であると判別することができない部分を、パートテンプレート301を用いて修正する処理である。そして、図2のステップS23に戻る。

以上のように本実施形態では、図12のフローチャートを実行することにより、設定手段の一部が実現される。

【0067】

次に、図13のフローチャートを参照しながら、図12のステップS502の調号修正処理について説明する。尚、この図13のフローチャートを開始する前に、CPU1は、図2のステップS1で得られた楽譜の各パートを認識し、認識したパートが、ドラム譜やリズム譜等の調号の存在しないパートである場合には、そのパートを調号修正処理の対象外とする。すなわち、ここで調号修正処理の対象外としたパートについては、以下に説明する図13のフローチャートによる処理を実行しない。

まず、ステップS601において、CPU1は、図2のステップS1で得られた楽譜の各パートの各小節について、移調量nKeyを算出する。このnKeyの値は、調号のIDから、楽典的に決められる数値であり、正負の値が存在するが、本実施形態では、12半音のうちどの位置になるかが分かればよいとして、正のみの値とする。このnKeyは、計算式で算出してもよいが、本実施形態では、図14に示すようなテーブルを予めHDD4に記憶しておき、各調号のIDから算出される"シャープ、フラットの数（nKeyNumの絶対値）"をキーにして、図14に示すテーブルから取得される。

【0068】

次に、ステップS602において、CPU1は、図2のステップS1で得られた楽譜の各パートの各小節について、移調楽器を考慮した移調量nAddKeyを、以下の（1）式を用いて算出する。

$$nAddKey = (nInstKey + nKey + 96) \bmod 12 \quad \cdots (1)$$

（1）式において、modは、剰余を算出することを表している。nAddKeyについても、本実施形態では、12半音のうちどの位置になるかが分かればよいとし、（1）式に示すように、得られた値が正になるのに十分な量（（1）式の例では96）を加えてから剰余をとるようにしている。（1）式の計算の結果、nAddKeyが負の値となった場合には、nAddKeyが正の値になるまで、（nInstKey+nKey+96）に12を加算して剰余をとる。また、（1）式において、nInstKeyは、図5のステップS107で取得されたパートテンプレート301に含まれる移調楽器設定305の内容である。

図15、図16は、以上のようにして算出されたnKeyとnAddKeyとの具体例を示す図である。

【0069】

次に、ステップS603において、CPU1は、図2のステップS1で得られた楽譜が、移調楽器を反映した楽譜であるか否かを判定する。ここで、図2のステップS1で得られた楽譜が、移調楽器を反映した楽器であるか否かを判定する方法の一例を、図15、図16を参照しながら説明する。

まず、CPU1は、nKeyの値を小節毎参照し、同じ値を有するnKeyを除くnKeyの数nEls eNumを算出する。

10

20

30

40

50

【0070】

図15に示す例の小節1において、nKeyの値は0、10、7であるので全て異なる。よって、同じ値を有するnKeyを除くnKeyの数nElseNumは2である。また、図15に示す例の小節2において、nKeyの値は2、0、9であるので全て異なる。よって、同じ値を有するnKeyを除くnKeyの数nElseNumは2である。

一方、図16に示す例の小節1において、nKeyの値は0、0、7であるので、同じ値(=0)を有するnKeyを除くnKeyの数nElseNumは1である。また、図16に示す例の小節2において、nKeyの値は2、0、9であるので全て異なる。よって、同じ値を有するnKeyを除くnKeyの数nElseNumは2である。

【0071】

次に、CPU1は、nAddKeyの値を小節毎参照し、同じ値を有するnAddKeyを除くnKeyの数nElseNumを算出する。

図15に示す例の小節1において、nAddKeyの値は10、10、10であるので全て同じである。よって、同じ値を有するnAddKeyを除くnAddKeyの数nElseNumは0である。また、図15に示す例の小節2において、nAddKeyの値は0、0、0であるので全て同じである。よって、同じ値を有するnAddKeyを除くnAddKeyの数nElseNumは0である。

一方、図16に示す例の小節1において、nAddKeyの値は10、0、10であるので、同じ値(=10)を有するnKeyを除くnKeyの数nElseNumは1である。また、図16に示す例の小節2において、nAddKeyの値は0、0、0であるので全て同じである。よって、同じ値を有するnAddKeyを除くnAddKeyの数nElseNumは0である。

【0072】

次に、CPU1は、各小節のnKeyについて算出したnElseNumを加算すると共に、各小節のnAddKeyについて算出したnElseNumを加算する。そして、例えば、CPU1は、nKeyについて算出したnElseNumの加算値の方が、nAddKeyについて算出したnElseNumの加算値よりも大きければ、図2のステップS1で得られた楽譜は、移調楽器を反映した楽譜であると判定し、そうでなければ、図2のステップS1で得られた楽譜は、移調楽器を反映した楽譜でないと判定する。

【0073】

図15に示す例では、nKeyについて算出したnElseNumの加算値は4(=2+2)であり、nAddKeyについて算出したnElseNumの加算値は0であるので、図2のステップS1で得られた楽譜は、移調楽器を反映した楽譜であると判定される。同様に、図16に示す例では、nKeyについて算出したnElseNumの加算値は3(=1+2)であり、nAddKeyについて算出したnElseNumの加算値は1(=1+0)であるので、図2のステップS1で得られた楽譜は、移調楽器を反映した楽譜であると判定される。

【0074】

以上のように本実施形態では、移調楽器を反映した楽譜である場合に値が大きくなる、nKeyから算出されるnElseNum(の加算値)と、移調楽器を反映した楽譜である場合に値が小さくなる、nAddKeyから算出されるnElseNum(の加算値)とを比較することにより、図2のステップS1で得られた楽譜が移調楽器を反映した楽譜であるか否かを判定するようにしている。

【0075】

以上のようなステップS603の判定の結果、図2のステップS1で得られた楽譜が移調楽器を反映した楽譜でない場合には、後述するステップS611に進む。一方、図2のステップS1で得られた楽譜が移調楽器を反映した楽譜である場合には、ステップS604に進む。

ステップS604に進むと、CPU1は、調号の認識に失敗しているか否かを判定する。ここで、調号の認識に失敗しているか否かを判定する方法の一例を、図15、図16を参照しながら説明する。

まず、CPU1は、nKeyについて算出したnElseNumの加算値と、nAddKeyについて算出したnElseNumの加算値とのうち、小さい方の加算値を得るために使用したnKey又はnAddKe

10

20

30

40

50

yを各小節について参照し、それらnKey又はnAddKeyの値が全て同じである場合には調号の認識に成功したと判定し、そうでない場合には調号の認識に失敗したと判定する。

【0076】

図15に示す例では、nKeyについて算出したnElseNumの加算値(=4)よりも、nAddKeyについて算出したnElseNumの加算値(=0)の方が小さい。そして、nAddKeyについて算出したnElseNumの加算値(=0)を得るために使用したnAddKeyの値は、小節1においても小節2においても全て同じ(=10、0)である。したがって、調号の認識に成功したと判定する。

一方、図16に示す例では、nKeyについて算出したnElseNumの加算値(=3)よりも、nAddKeyについて算出したnElseNumの加算値(=1)の方が小さい。そして、nAddKeyについて算出したnElseNumの加算値(=1)を得るために使用したnAddKeyの値は、小節2においては全て同じ(=0)であるが、小節1においては1つだけ0となり、他の値(=10)と異なる。したがって、トロンボーンの小節1において調号の認識に失敗したと判定する。

【0077】

以上のようなステップS604の判定の結果、調号の認識に失敗していない(成功した)と判定された場合には、図12のステップS503に戻る。一方、調号の認識に失敗したと判定された場合には、ステップS605に進む。

ステップS605に進むと、CPU1は、ステップS604で認識に失敗したと判定された調号を修正する。ここで、調号を修正する方法の一例を、図16を参照しながら説明する。

まず、CPU1は、各小節において、調号の認識に失敗したパートのnAddKeyの値が他の値と同じになるような新たなnKeyの値を、以下の(3)式を用いて算出する。

$$nKey = (S - nInstKey) \bmod 12 \cdots (3)$$

(3)式においても、modは、剰余を算出することを表している。また、Sは、調号の認識に失敗したパートと異なるパートのnAddKeyの値である。

【0078】

図16に示した例では、小節1において、調号の認識に失敗したパート(トロンボーン)と異なるパート(トランペット、アルトサックス)のnAddKeyの値Sは10である。よって、新たなnKeyの値は10となる。そして、この数値(=10)に基づいて、変調すべき調号の種類を決定する。図14に示すテーブルにおけるnKeyの値がこの数値と同じ値である調号が変更すべき調号となる。例えば、新たなnKeyの値が10である場合には、図14に示すテーブルから、変更すべき調号はフラット2個と決定する。尚、図14のテーブルにおいて、シャープ個数に対応するnKeyの値とフラット個数に対応するnKeyの値とが同じ場合がある。このような場合には、その他の情報に基づいて、変更すべき調号として、シャープ及びフラットの何れかを選択することになる。

そして、CPU1は、調号の認識に失敗したパートと異なるパートの内容(調号等)と、調号の認識に失敗したパートの他の小節の内容(調号等)と、調号の認識に成功したパートの他の小節の内容(調号等)と、認識に失敗した調号の配置箇所における次点IDの内容と、認識に失敗した調号よりも前にあるナチュラル(認識に失敗した調号が属する小節にあるナチュラルと、認識に失敗した調号が属する小節よりも前の小節にあるナチュラルとの少なくとも何れか一方)の状態と、認識に失敗した調号が属する小節よりも後の小節にあるナチュラルの状態との少なくとも何れか1つに基づいて修正する。

図16に示した例のように古いnKeyが0であり、新たなnKeyが0以外である場合には調号を追加することになる。また、古いnKeyが0以外であり、新たなnKeyが0以外である場合には調号を変更することになる。更に、古いnKeyが0以外であり、新たなnKeyが0である場合には、認識された調号を削除することになる。

【0079】

次に、ステップS606において、CPU1は、ステップS605において調号を削除したか否かを判定する。この判定の結果、調号を削除していない場合には、図12のステ

10

20

30

40

50

ップS503に戻る。一方、調号を削除した場合には、認識に失敗した調号は臨時記号である可能性が高いと判定しステップS607に進む。

ステップS607に進むと、CPU1は、調号として誤認識した音楽記号（削除した調号）を個々の音楽記号に分解する。

次に、ステップS608において、CPU1は、ステップS607で得られた音楽記号の親となる音楽記号（音符等）を検索する。このとき、その音楽記号が、親となる音楽記号（音符等）の臨時記号であると判定するための閾値（距離）を小さくして、その音楽記号の親となる音楽記号（音符等）を検索する。

【0080】

次に、ステップS609において、CPU1は、ステップS607で得られた音楽記号の親となる音楽記号（音符等）を検索することができたか否かを判定する。この判定の結果、音楽記号の親となる音楽記号（音符等）を検索することができなかった場合には、図12のステップS503に戻る。一方、音楽記号の親となる音楽記号（音符等）を検索することができた場合には、ステップS610に進む。

10

ステップS610に進むと、CPU1は、調号として誤認識した記号（削除した調号）を臨時記号として設定する。そして、図12のステップS503に戻る。

【0081】

ステップS603において、図2のステップS1で得られた楽譜が移調楽器を反映した楽譜でないと判定された場合には、ステップS611に進む。ステップS611に進むと、各小節において全パートの調号が同じであるか否かを判定する。この判定の結果、全パートの調号が同じである場合には、図12のステップS503に戻る。一方、全パートの調号が同じでない場合には、ステップS612に進む。

20

【0082】

ステップS612に進むと、CPU1は、全パートの調号が同じとなるように、調号を修正する。具体的にCPU1は、各小節において、nKeyの値が他と異なるパートにおける調号を、nkeyの値が同じである複数のパートにおける調号に修正する。そして、前述したステップS606に進む。

【0083】

次に、図17のフローチャートを参照しながら、図12のステップS503の音部記号修正処理について説明する。

30

まず、ステップS701において、CPU1は、ステップS1で得られた楽譜の一部のパートにおいて、音部記号の認識に失敗しているか否かを判定する（すなわち、一部のパートにおいて、他のパートと異なる音部記号が存在していたり、音部記号が存在していなかったりしているか否かを判定する）。この判定の結果、音部記号の認識に失敗していない場合には、図12のステップS504に戻る。一方、音部記号の認識に失敗している場合には、ステップS702に進む。

【0084】

ステップS702に進むと、CPU1は、ステップS701で認識に失敗している音部記号に対する次点ID405が記憶されているか否かを判定する。この判定の結果、認識に失敗している音部記号に対する次点ID405が記憶されていない場合には、図12のステップS504に戻る。一方、認識に失敗している音部記号に対する次点ID405が記憶されている場合には、ステップS703に進む。

40

【0085】

ステップS703に進むと、CPU1は、ステップS702で記憶されていると判定された次点ID405に対応する音部記号と、図5のステップS107で取得されたパートテンプレート301の音部記号306とが一致するか否かを判定する。この判定の結果、音部記号が一致しない場合には、図12のステップS504に戻る。一方、音部記号が一致する場合には、ステップS704に進む。

ステップS704に進むと、CPU1は、認識に失敗している音部記号をパートテンプレート301の音部記号306に変更する。そして、図12のステップS504に戻る。

50

【0086】

尚、図17では、ステップS702において、ステップS701で認識に失敗している音部記号に対する次点ID405が記憶されていないと判定された場合には、図17のフローチャートの処理を終了するようにしたが、必ずしもこのようにする必要はない。例えば、ステップS701で認識に失敗している音部記号に対する次点ID405が記憶されていないと判定された場合には、その音部記号に対する次次点ID406が記憶されているか否かを判定する。この判定の結果、次次点ID406が記憶されていない場合には、図17のフローチャートの処理を終了させる。一方、次次点ID406が記憶されている場合には、その次次点ID406に対応する音部記号と、図5のステップS107で取得されたパートテンプレート301の音部記号306とが一致するか否かを判定する。そして、これらが一致する場合にはステップS704に進み、そうでない場合には図17のフローチャートの処理を終了させるようにしてもよい。

10

【0087】

また、図17では、ステップS701で認識に失敗している音部記号に対する次点ID405が記憶されている場合に音部記号を修正するようにしたが、必ずしもこのようにする必要はなく、次点ID405に関わらず、全段落で強制的に音部記号の変更を行うようにしてもよい。このようにする場合には、図17のステップS702の処理が少なくとも不要となる。

ただし、より妥当な調号の修正を行うために、次点IDに加え、認識に成功している音部記号を用いて、調号の修正を行うか否かを判定するようにしてもよい。例えば、図17において、ステップS702の後に、認識に成功している他の音部記号を参照し、参照した音部記号が、認識に失敗している音部記号に対する次点ID405に対応するものであるか否かを判定し、その判定の結果、参照した音部記号が、認識に失敗している音部記号に対する次点ID405に対応するものである場合にステップS703を行い、そうでない場合に図17のフローチャートの処理を終了させるようにしてもよい。

20

【0088】

以上のように本実施形態では、スキャナ11で読み取られた楽譜の画像が認識された後に設定が必要となる情報を含むパートテンプレート301を予め記憶しておき、スキャナ11で読み取られた楽譜に応じたパートテンプレート301を複数のパートテンプレートから選び出す。そして、選び出したパートテンプレートを用いて、スキャナ11で読み取られた楽譜に対し、自動的に設定（修正）を行うようにした。したがって、楽譜を認識した後にユーザが行う設定を従来よりも少なくすることができ、且つ楽譜の認識率を従来よりも向上させることができる。

30

【0089】

尚、本実施形態では、パートテンプレート301の情報に基づいて、調号や音部記号を楽譜の認識結果に対して設定（修正）する場合を例に挙げて説明したが、前述したように、楽譜の認識結果に対して設定（修正）するのは、調号や音部記号に限定されない。例えば、パートテンプレート301に音色に関する情報を含めた場合には、図2のステップS1で得られた楽譜に対して、図5のステップS107で取得されたパートテンプレート301に含まれている音色を、状況に応じて（例えばユーザによる操作に基づいて）設定するようにしてもよい。

40

【0090】

また、図2のステップS5で認識された線数よりも少ない線数がパートテンプレート301に設定された場合には、ステップS5で認識された線の有効性を、ドット数の割合等に基づいて再度判定したり、パートテンプレート301に設定された線数に線数を変更したりすることができる。一方、ステップS5で認識された線数よりも多い線数がパートテンプレート301に設定された場合には、ステップS5で認識された線の上下の想定される位置に線が存在するか否かを再度判定したり、パートテンプレート301に設定された線数に線数を変更したりすることができる。

【0091】

50

また、図2のステップS17（図9）のパート内認識修正処理の少なくとも一部と、図2のステップS22（図12）のパート間認識修正処理の少なくとも一部とを、状況に応じて（例えばユーザによる操作に基づいて）行わないようにしてもよい。

【0092】

また、図12のステップS502（図13）の調号修正処理のステップS605において、図15、図16に示したような情報が得られた段階で、ある小節での調号が、実は臨時記号であると判定できる場合がある。例えば、nKey及びnAddKeyが略全ての小節で同じであり、一部の小節のみ、nElseNumが大きい場合である。このような場合には、ステップS605で説明した処理を行うことなく、その調号を臨時記号として設定し直すようにする。

10

【0093】

以上説明した本発明の各実施形態は、コンピュータがプログラムを実行することによって実現することができる。また、プログラムをコンピュータに供給するための手段、例えばかかるプログラムを記録したCD-ROM等のコンピュータ読み取り可能な記録媒体、又はかかるプログラムを伝送する伝送媒体も本発明の実施の形態として適用することができる。また、上記プログラムを記録したコンピュータ読み取り可能な記録媒体などのプログラムプロダクトも本発明の実施の形態として適用することができる。上記のプログラム、コンピュータ読み取り可能な記録媒体、伝送媒体及びプログラムプロダクトは、本発明の範疇に含まれる。

また、前述した各実施形態は、何れも本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその技術思想、またはその主要な特徴から逸脱することなく、様々な形で実施することができる。

20

【図面の簡単な説明】

【0094】

【図1】本発明の実施形態を示し、楽譜認識装置の構成の一例を示すブロック図である。

【図2-1】本発明の実施形態を示し、CPUの処理の一例を説明するメインフローチャートである。

【図2-2】本発明の実施形態を示し、図2-1に続くフローチャートである。

【図3】本発明の実施形態を示し、パートテンプレートの一例を示す図である。

30

【図4】本発明の実施形態を示し、認識の際に候補に挙げた音楽記号を識別するためのIDの記憶方法の一例を概念的に示す図である。

【図5】本発明の実施形態を示し、図2のステップS8におけるパートテンプレート適用処理の一例を説明するフローチャートである。

【図6】本発明の実施形態を示し、認識パートテンプレート適用表の一例を示す図である。

【図7】本発明の実施形態を示し、パートテンプレート適用処理の結果の表示内容の一例を示す図である。

【図8】本発明の実施形態を示し、図2のステップS10の認識修正処理の一例を説明するフローチャートである。

40

【図9】本発明の実施形態を示し、図2のステップS17のパート内誤認識修正処理の一例を説明するフローチャートである。

【図10】本発明の実施形態を示し、オクターブ付き音部記号1の一例を示す図である。

【図11】本発明の実施形態を示し、図9のステップS302のオクターブ付き音部記号修正処理の一例を説明するフローチャートである。

【図12】本発明の実施形態を示し、図2のステップS22のパート間誤認識修正処理の一例を説明するフローチャートである。

【図13】本発明の実施形態を示し、図12のステップS502の調号修正処理について説明するフローチャートである。

【図14】本発明の実施形態を示し、nKeyとnKeyNumと調号との関係を記憶したテーブル

50

の一例を示す図である。

【図 1 5】本発明の実施形態を示し、nKeyとnAddKeyの第 1 の具体例を示す図である。

【図 1 6】本発明の実施形態を示し、nKeyとnAddKeyの第 2 の具体例を示す図である。

【図 1 7】本発明の実施形態を示し、図 1 2 のステップ S 5 0 3 の音部記号修正処理について説明するフローチャートである。

【符号の説明】

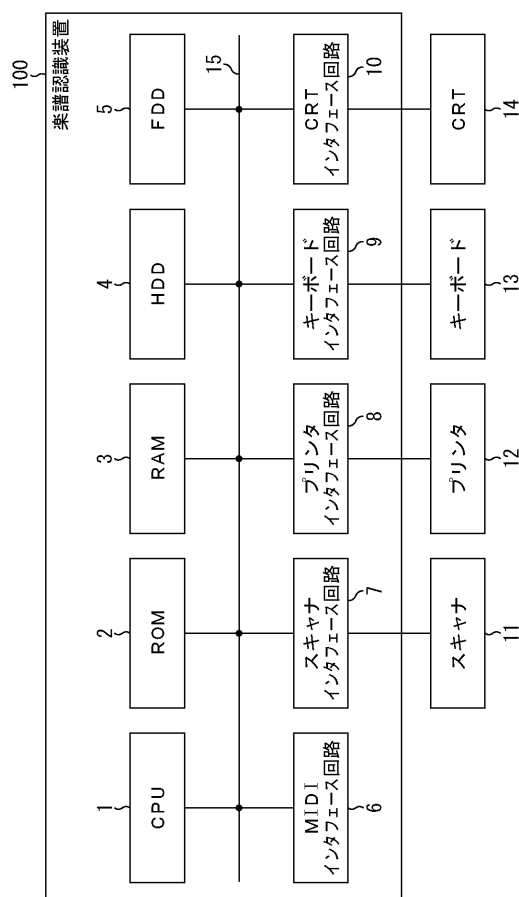
【0 0 9 5】

- 1 CPU
- 2 ROM
- 3 RAM
- 4 HDD
- 5 FDD
- 6 M I D I
- 7 スキャナインタフェース回路
- 8 プリンタインタフェース回路
- 9 キーボードインタフェース回路
- 10 C R T インタフェース回路
- 11 スキャナ
- 12 プリンタ
- 13 キーボード
- 14 C R T
- 100 楽譜認識装置
- 301 パートテンプレート

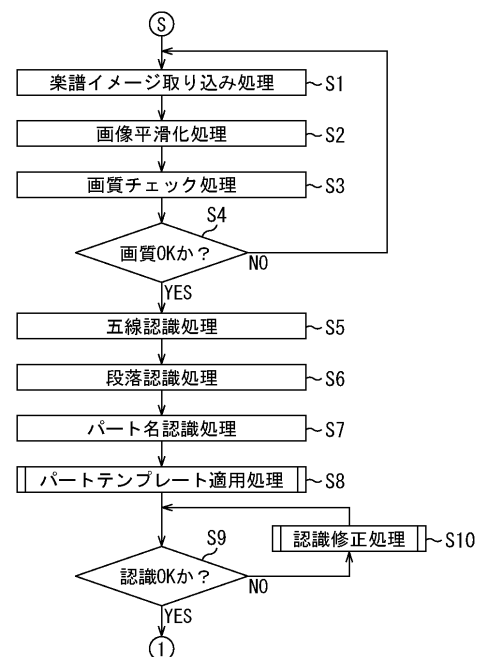
10

20

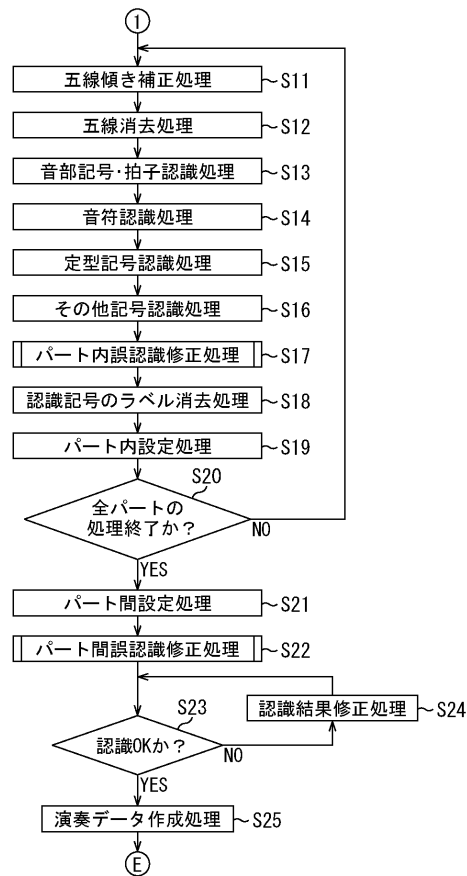
【図 1】



【図 2 - 1】



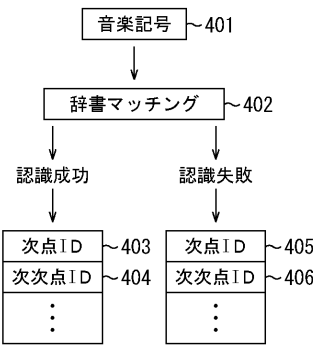
【図 2 - 2】



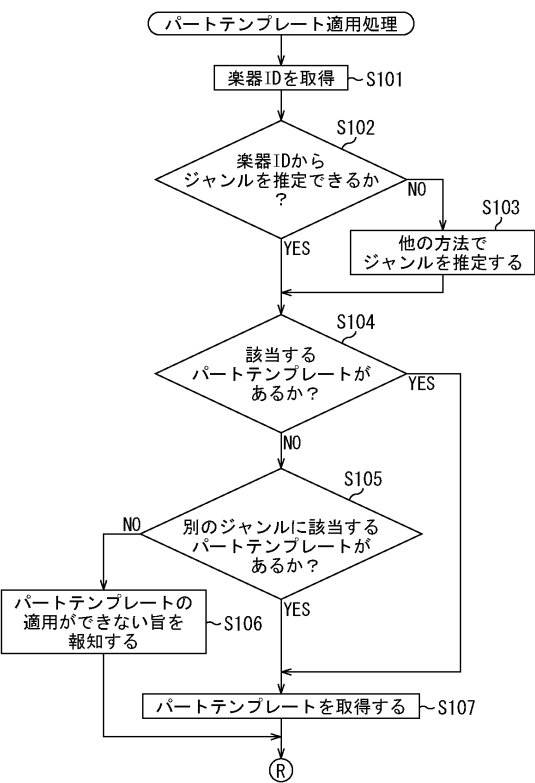
【図 3】

301	
テンプレート名	~302
パート名	~303
パート名短縮	~304
移調楽器設定	~305
音部記号	~306
タブ譜線数	~307
タブ譜チューニング情報	~308

【図 4】



【図 5】



【図 6】

601

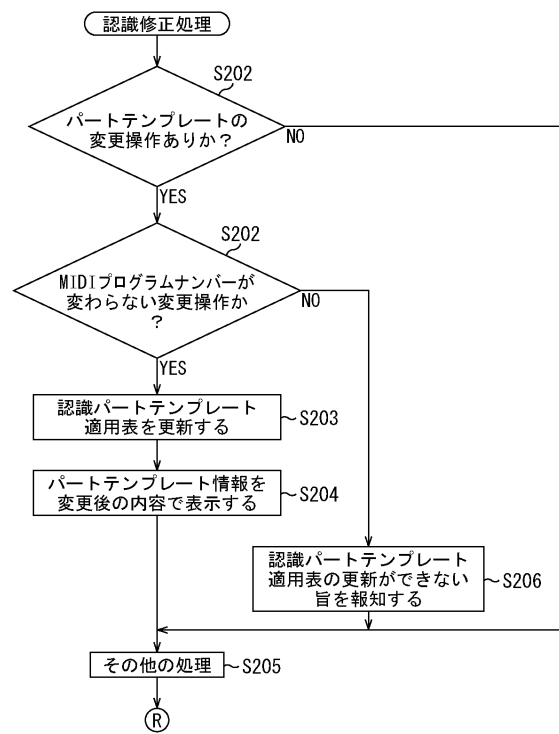
楽器ID					
ジャンル	ピアノ	トランペット	合唱(テノール)	タブ譜6弦	...
クラシック	TGr andPiano	TOrclTrumpet	設定なし	設定なし	...
合唱	TGr andPiano	設定なし	TChorusTenor	設定なし	...
ポップス	TPopPiano	TLeadTrumpet	設定なし	TGuitarTub	...
ジャズ	TJazzPiano	TJazzTrumpet	設定なし	TJazzGuitarTub	...
：	：	：	：	：	：
標準	TGr andPiano	TTrumpet	設定なし	設定なし	...

【図 7】

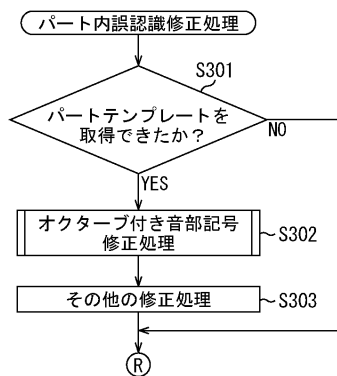
701

パートNo.	パート名	パートテンプレート名
1	Soprano	コーラスソプラノA
2	Alto	コーラスアルトA
3	Tenor	コーラステノールA
4	Bass	コーラスバスA

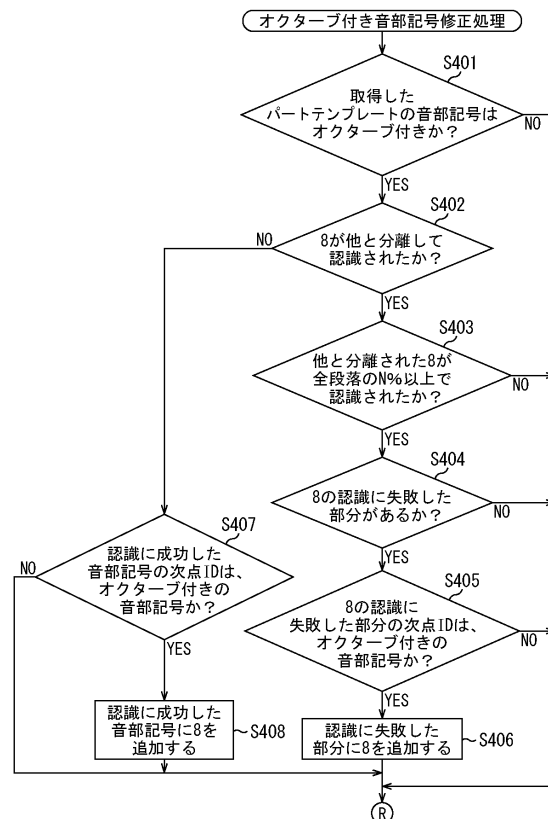
【図 8】



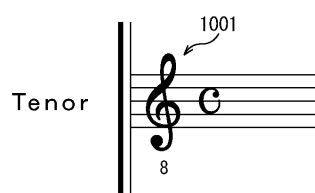
【図 9】



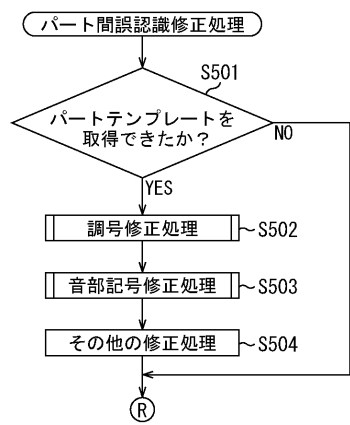
【図 11】



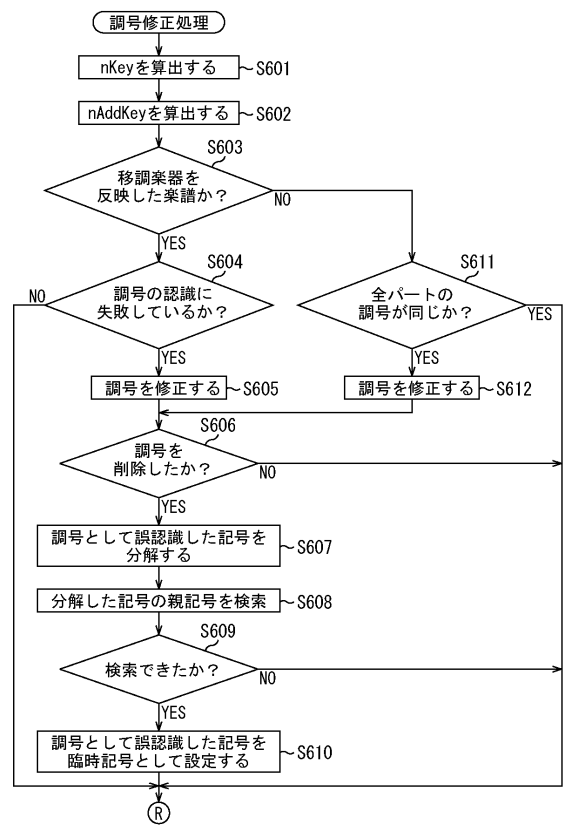
【図 10】



【図 1 2】



【図 1 3】



【図 1 4】

フラット個数	nKeyNum	nKey	シャープ個数	nKeyNum	nKey
0	0	0	0	0	0
1	-1	5	1	1	7
2	-2	10	2	2	2
3	-3	3	3	3	9
4	-4	8	4	4	4
5	-5	1	5	5	11
6	-6	6	6	6	6
7	-7	11	7	7	1

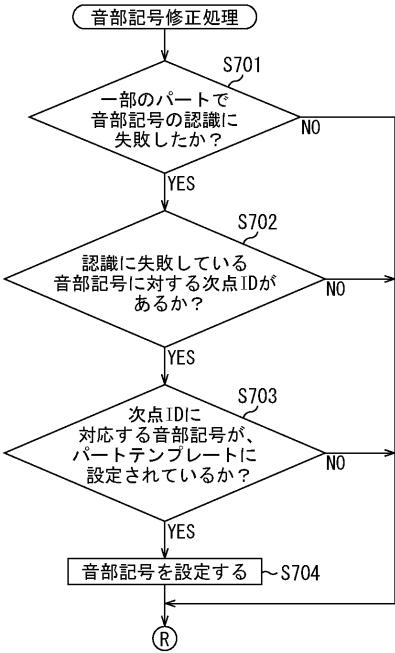
【図 1 5】

楽器	nInstKey	小節1の記号	nKey	nAddKey	小節2の記号	nKey	nAddKey
トランペット	-2	なし	0	10	シャープ2個	2	0
トロンボーン	0	フラット2個	10	10	なし	0	0
アルトサックス	-9	シャープ1個	7	10	シャープ3個	9	0
nElseNum	—	—	2	0	—	2	0

【図 1 6】

楽器	nInstKey	小節1の記号	nKey	nAddKey	小節2の記号	nKey	nAddKey
トランペット	-2	なし	0	10	シャープ2個	2	0
トロンボーン	0	なし	0	0	なし	0	0
アルトサックス	-9	シャープ1個	7	10	シャープ3個	9	0
nElseNum	—	—	1	1	—	2	0

【図 1 7】



フロントページの続き

- (56)参考文献 特開平09-097061 (JP, A)
特開2001-100740 (JP, A)
特開2004-341552 (JP, A)
特開2002-006842 (JP, A)
特開2007-264509 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G10G 1/00-7/02
G10H 1/00-7/00