

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4743612号
(P4743612)

(45) 発行日 平成23年8月10日(2011.8.10)

(24) 登録日 平成23年5月20日(2011.5.20)

(51) Int.Cl.	F I		
G 0 9 B 15/00 (2006.01)	G 0 9 B	15/00	A
G 1 0 H 1/00 (2006.01)	G 1 0 H	1/00	1 0 2 Z
G 1 0 G 1/00 (2006.01)	G 1 0 H	1/00	Z
	G 1 0 G	1/00	

請求項の数 3 (全 28 頁)

(21) 出願番号	特願2006-53406 (P2006-53406)	(73) 特許権者	393031586
(22) 出願日	平成18年2月28日(2006.2.28)		株式会社国際電気通信基礎技術研究所
(65) 公開番号	特開2007-58164 (P2007-58164A)		京都府相楽郡精華町光台二丁目2番地2
(43) 公開日	平成19年3月8日(2007.3.8)	(74) 代理人	100090181
審査請求日	平成20年3月31日(2008.3.31)		弁理士 山田 義人
(31) 優先権主張番号	特願2005-218894 (P2005-218894)	(72) 発明者	大島 千佳
(32) 優先日	平成17年7月28日(2005.7.28)		京都府相楽郡精華町光台二丁目2番地2
(33) 優先権主張国	日本国(JP)		株式会社国際電気通信基礎技術研究所内
	(出願人による申告)平成17年度独立行政法人情報通信研究機構、研究テーマ「超高速知能ネットワーク社会に向けた新しいインタラクション・メディアの研究開発」に関する委託研究、産業活力再生特別措置法第30条の適用を受ける特許出願	(72) 発明者	樋川 直人
			京都府相楽郡精華町光台二丁目2番地2
		(72) 発明者	西本 一志
			京都府相楽郡精華町光台二丁目2番地2
			株式会社国際電気通信基礎技術研究所内

最終頁に続く

(54) 【発明の名称】 演奏学習支援システム

(57) 【特許請求の範囲】

【請求項1】

学習者によって操作され、演奏に応じて演奏データを出力する入力インタフェース、前記演奏データに基づいて発音する発音手段、演奏位置と、演奏位置毎の前記学習者が演奏すべき音の音高データを含む楽譜データを格納する楽譜データベース、手本映像を格納する映像データベース、前記学習者の演奏が停止したことを検出する停止検出手段、前記停止検出手段が停止を検出したとき、その停止位置より前において前記学習者の演奏に間違いがあったかどうか判断する判断手段、および前記判断手段が間違いがあったと判断したとき停止位置より前の前記手本映像を前記映像データベースから読み出して表示し、間違いがなかったと判断したとき前記停止位置より後の前記手本映像を表示する表示手段を備える、演奏学習支援システム。

【請求項2】

前記楽譜データには推奨戻り位置が設定されていて、前記判断手段は前記停止位置の直前の前記推奨戻り位置までの間に間違いがあったかどうか判断する、請求項1記載の演奏学習支援システム。

【請求項3】

前記学習者の手元映像を撮影するカメラをさらに備え、前記表示手段は前記学習者の演奏が継続しているとき前記手元映像を表示する、請求項

1 または 2 の演奏学習支援システム。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は合奏支援システムおよび演奏学習支援システムに関し、特にたとえば、初心者同士でも簡単に合奏や連弾を楽しむことができる、新規な合奏支援システムおよび演奏学習支援システムに関する。

【背景技術】

【0002】

子供がピアノのレッスンを受けている場合など、子供と一緒に合奏を楽しみたいという親の声は多い。 10

【0003】

このため、最近では子供向けのみならず、大人のための音楽教室や音楽ソフトも流行っている。中でも、なるべく短い時間で各々の大人が弾きたい曲を演奏できるようにするレッスンやソフトが多い。これらは、子供のように月日を重ねて基礎から学ぶことよりも、いち早く裡にある音楽を表現することを優先していると思われる。しかし、子供といろんな曲で合奏できるようになるには、やはり楽譜上の音符通りに楽器を奏するという基礎から習得しなければならず、長い年月を要してしまう。

【0004】

また、ピアノのレッスンでは、合奏の1つである連弾を取り入れることが奨励されている。たとえば、ピアノを習い始めて1週目の子供が、「ド」を数個並べただけの楽曲を練習する際にも、先生が連弾のパートナーとして多様な和音を響かせながら付き添うレッスンが行われている。連弾を通じて生徒の練習意欲が高まったという報告や、子供たちがパートナーの演奏を聴くことで自ら「やりたい表現」を自然に演奏に表すようになったという報告もある。どんなに初心者の子供も、連弾の中で先生と「呼吸」を通わせるという、音楽性を育てる上で不可欠な営みを行っている。 20

【0005】

しかし、家庭においてレッスンに通い始めの子供と、楽器演奏経験のほとんどない親とのペアで、練習や楽しみとして連弾を行うことは非常に困難である。これは、市販されているほとんどのピアノ連弾曲集が、少なくともペアの一方が中級以上に想定されていることから伺える。最近ではピアノ学習用として、予めパートナーの演奏データが用意されたマイナス・ワンを多くみかけるようになった。しかし、マイナス・ワンを利用してこの方法では一人ひとりの演奏行為と実在的な関わりを持たないため、パートナーと「呼吸」を通わせることができない。 30

【0006】

一方、近年、人間の演奏者と合奏するシステムの研究が進んでいる。予め楽譜が与えられている音楽を対象とした自動伴奏システムが非特許文献1や非特許文献2に開示される。非特許文献1のものは、演奏者が出力した音高情報だけを用いたアルゴリズムを示し、非特許文献2のものは、音高と一定時間ごとの演奏時刻とを認識して伴奏を変化させる手法を提案している。 40

【0007】

さらに、非特許文献3では、より自然な伴奏システムの演奏を目指して、人間とコンピュータの合奏を分析し、コンピュータと人間との「ずれ」から次の「時間長変化」を予測するモデルを提案している。

【0008】

また、非特許文献4では、人間とコンピュータの相互作用を考慮したモデルの提案が行われている。

【非特許文献1】Dannenbergh, R.: An On-Line Algorithm for Real-Time Accompaniment, Proc. ICMC 1984, pp. 193-198, 1984

【非特許文献2】Vercoe, B.: The Synthetic Performer in the Context of LivePerf 50

ormance、Proc. ICMC 1984、pp. 200、1984

【非特許文献3】堀内靖雄、坂本圭司、市川憲：合奏における人間の発音時刻制御モデルの推定、情報処理学会論文誌、Vol. 43、No. 2、pp. 260-267、2002

【非特許文献4】井川孝之、直井邦彰、大照完、橋本周司：相互作用モデルによる実時間適応自動伴奏とその動作解析、電子情報通信学会春季全国大会講演論文集、pp. 1. 389-390、1992

【発明の開示】

【発明が解決しようとする課題】

【0009】

しかしながら、以上の研究は、すべて、人間とシステム(コンピュータ)とによる合奏を目的とするもので、この発明が向けられる、人間同士特に初級者同士が合奏するのを支援するシステムを対象とするものではない。

【0010】

また、初級者同士でも合奏や連弾を容易に行なうことができるシステムが望まれている。

【0011】

それゆえに、この発明の主たる目的は、新規な、合奏支援システムおよび演奏学習支援システムを提供することである。

【0012】

この発明の他の目的は、初級者同士でも合奏や連弾が容易に行える、合奏支援システムを提供することである。

【課題を解決するための手段】

【0018】

請求項1の発明は、学習者によって操作され、演奏に応じて演奏データを出力する入力インタフェース、演奏データに基づいて発音する発音手段、演奏位置と、演奏位置毎の学習者が演奏すべき音の音高データを含む楽譜データを格納する楽譜データベース、手本映像を格納する映像データベース、学習者の演奏が停止したことを検出する停止検出手段、停止検出手段が停止を検出したとき、その停止位置より前において学習者の演奏に間違いがあったかどうか判断する判断手段、および判断手段が間違いがあったと判断したとき停止位置より前の手本映像を映像データベースから読み出して表示し、間違いがなかったと判断したとき停止位置より後の手本映像を表示する表示手段を備える、演奏学習支援システムである。

【0019】

請求項1の発明では、入力インタフェース(実施例において相当する部分の参照符号12.以下同様。)が学習者によって操作され、演奏に応じて演奏データとしてMIDIデータを出力する。入力インタフェースからのMIDIデータはそのまま発音手段(16, 18)から発音されるが、このMIDIは判断手段にも与えられる。そして、判断手段(20, S1, S117, S605, S607, S1001)では、演奏停止位置より前に演奏誤りがあったかどうか判断する。判断手段が肯定的な判断したときすなわち誤りがあったと判断したとき表示手段は、その停止位置より前の手本映像を「戻り手本」として表示する。判断手段が否定的な判断したときすなわち誤りがなかったと判断したとき表示手段は、その停止位置より先の手本映像を「先読み手本」として表示する。

【0020】

請求項1の発明によれば、学習者が支援を必要としているときにのみ手本映像を表示するようにしたので、学習者の読譜の学習を妨げることはない。しかも、演奏誤りをしたかどうかで違う手本映像が表示されるので、つまり、停止した理由に応じた手本映像を表示するので、学習者は自分の演奏が間違っていたのかどうか容易に判断することができる。

【0021】

請求項2の発明は、楽譜データには推奨戻り位置が設定されていて、判断手段は停止位置の直前の推奨戻り位置までの間に間違いがあったかどうか判断する、請求項1記載の演

10

20

30

40

50

奏学習支援システムである。

【 0 0 2 2 】

請求項 3 の発明は、学習者の手元映像を撮影するカメラをさらに備え、表示手段は学習者の演奏が継続しているとき手元映像を表示する、請求項 1 または 2 記載の演奏学習支援システムである。

【発明の効果】

【 0 0 2 3 】

この発明によれば、第 1 奏者の演奏位置を追跡し、その位置において第 2 奏者の演奏データの音高だけを差し替えて合奏や連弾が簡単に行えるような合奏支援システムにおいて、演奏の中断後や終了後に曲の途中から練習する場合でも、システムが推奨する戻り位置（推奨する練習開始位置）を選択または指定することにより、曲の途中であっても、第 2 奏者は 1 音目から正しい音高を出力することができる。しかも、支援が必要なときだけ手本映像を表示するようにしたので、学習者の読譜に対する意欲を減じることがない。

10

【 0 0 2 4 】

この発明の上述の目的、その他の目的、特徴および利点は、図面を参照して行う以下の実施例の詳細な説明から一層明らかとなる。

【発明を実施するための最良の形態】

【 0 0 2 5 】

図 1 はこの発明の背景となる合奏支援システム 10 を示し、この図 1 に示す合奏支援システム 10 は、プリモ用入力インタフェース 12 およびセコンド用入力インタフェース 14 を含む。ただし、この図 1 では 2 つのインタフェース 12 および 14 を別々に図示しているが、連弾の場合には、1 つのインタフェース（鍵盤）を 2 つの領域に分けて使用することになる。

20

【 0 0 2 6 】

なお、「プリモ」とは、ピアノ連弾の場合に高音部を担当する演奏者のことであり、「セコンド」は低音部を演奏する演奏者のことを言う。合奏の場合には必ずしもこのような呼称を用いないが、ここでは、便宜上、合奏をリードする演奏者を「プリモ」と呼び、そうでない演奏者を「セコンド」と呼ぶ。ただし、前者を「第 1 奏者」、後者を「第 2 奏者」と呼ぶこともある。

【 0 0 2 7 】

プリモ用入力インタフェース 12 およびセコンド用入力インタフェース 14 はともに、演奏に応じて、MIDI (Musical Instrument Digital Interface) データを出力する。そして、プリモが演奏したことに応じて入力インタフェース 12 から出力される MIDI データは、そのまま、なんら処理を施されずに MIDI 音源 16 に入力され、スピーカ 18 から音として出力される。プリモ用入力インタフェース 12 からの演奏データ (MIDI) はまた、コンピュータ 20 にも同時に入力される。同じように、セコンドが演奏したことに応じて入力インタフェース 14 から出力される MIDI データもコンピュータ 20 に入力される。

30

【 0 0 2 8 】

簡単に言うと、コンピュータ 20 は、プリモの演奏データと楽譜データとを参照しながら、セコンドの演奏だけを支援する。プリモに対しては演奏支援を行わない。

40

【 0 0 2 9 】

コンピュータ 20 が参照すべき楽譜データは、楽譜データベース 22 に記憶されている。

【 0 0 3 0 】

図 2 がこのような楽譜データの基になる記述形式の一例を示す。この例で、「*」で始まる行には演奏データそのものが記述されているのではなく、その曲を初心者が演奏する場合に戻りやすい位置を示すマーカを記述するようにしている。

【 0 0 3 1 】

Bloch & Dannenberg (Bloch, J. and Dannenberg, R.B.: Real-Time Computer Accompa

50

niment of Keyboard Performances, Proc. ICMC 1985, pp.279-289, 1985) によれば、演奏の誤りは、以下の3つに分類される。

- (1) 挿入(extra):演奏された音が楽譜上の音と一致しない(余分な音数)。
- (2) 音高誤り(wrong):楽譜上の音数通りに音を出しているが、音高が不正確である。
- (3) 脱落(missing):演奏された音が楽譜上の音の数よりも少ない。

【0032】

しかし、特に初級者の場合はこれらに「弾き直し」が追加されると考えられる。弾き直しとは、「挿入」と同様に楽譜上の音数よりも多く音が演奏されたケースではあるが、楽譜上に記載されていない音や音高列ではなく、楽譜上に記載されている音や音高列が重複して演奏された場合のことを指す。中級者以上になると、練習中に意識して弾き直しする以外は、基本的に失敗しても演奏の流れを止めずに最後までいこうとする傾向がある。一方、初級者は途中で失敗した場合に戻ろうとする傾向が多い。

10

【0033】

ピアノレッスンに通いはじめて4年半の小学4年生である、被験者Aと、ピアノレッスンに通いはじめて1年の保育園生である、被験者Bとによって、いずれもプリモとして、様々な練習曲を初見で、演奏してもらった実験によれば、表1に示す結果が得られた。ただし、実験ではセコンドは、発明者の1人が担当した。表1は、弾き直した回数と弾き直すために戻った個所の特徴である。

【0034】

【表1】

20

戻った個所	A	B	合計	%
1拍目	315	141	456	81.7
第2の強拍	17	2	19	3.4
最終拍	0	6	6	1.1
フレーズ開始	5	0	5	0.9
手の替え	4	0	4	0.7
1拍目(1小節前)	26	25	51	9.1
最終拍(1小節前)	2	3	5	0.9
曲の最初	7	3	10	1.8
段の最初	1	1	2	0.4
合計	377	181	558	100.0

30

【0035】

「1拍目」とは、小節の1拍目に戻ったことを示す。「第2の強拍」とは、4拍子の3拍目や2拍子の2拍目を指す。「最終拍」とは3拍子の3拍目や4拍子の4拍目を指す。以上の3項目は、弾き直しする直前に弾いた音と同じ小節内の拍である。「フレーズ開始」とは、スラーのはじめや、弱起のメロディのはじまりを指す。「手の替え」とは、1つのメロディの中で、弾く手が右手から左手に替った個所に戻ったことを意味する。以上の2項目は「1拍目」「第2の強拍」「最終拍」のカテゴリには数えていない。「段」とは譜面上で横一列に印刷された数小節の塊を指す。なお「1小節前」とは、弾き直す直前の音の1小節前に戻ったことを意味している。

40

【0036】

弾き直しの数は被験者Aは377個所あり、被験者Bは181個所あった。また、戻り位置は小節の1拍目が8割を占めている。

【0037】

一方、「挿入」は、被験者Aに1音、被験者Bに6音見られた。「音高誤り」は被験者

50

Bに37音(延べ111音)見られた。「脱落」は被験者Aに3音見られた。

【0038】

その他、被験者Aは小節ごとに楽譜を見て弾く傾向があったため、小節の1拍目の直前に楽譜上には記述されていない空白の時間が起りやすかった。そのため、連弾曲ではセコンドが先走って1拍目の音を鳴らし、再度、被験者Aと一緒に鳴らすという場面が多くみられた。なお、課題にした独奏曲と連弾曲のレベルや曲数等が違うため、一概には言えないが、被験者Aは連弾曲のほうが弾き直しが少ない傾向にあった。

【0039】

この実施例では、上記のような実験の結果を参考にして、図2の「*」が付された行のように、曲を初心者が演奏する場合に戻りやすい位置をマーカとして記述するようにした。そして、マーカとしては、発明者等の実験では、次の4種類を設定できるようにしている。ただし、1箇所複数種のマーカを同時に設定することもできる。

【0040】

- B：小節の頭を示す
- P：フレーズの冒頭を示す
- R：印刷された楽譜の「段」の冒頭を示す
- S：その他、任意に設定可能なマーカ

なお、図1の実施例では、これらのマーカの種類の依存した動作の切り替えは行っており、マーカの有無だけをチェックするようにしている。つまり、この実施例では、複数種類のマーカを用意しているのは、人が楽譜データを作成するとき、あるいは読むときの便宜のためだけであるが、表1に示す弾き直し箇所の頻度を考慮して、これらの種類に応じて重みづけを調整することにより、より精度の高い弾き直し判定を実現することも可能である。

【0041】

さらに、図2の記述形式では、楽譜データは、次のように記述される。

- (1)「*」で始まるマーカ行以外は、すべて音符データを記述する。1行には、楽譜上で「同時に」発音されることになっている音を列挙する。
- (2)「音名」には、音符の「音価」（つまり4分音符、8分音符などの音の長さ）を無視して、音高のみを記述する。たとえば「C5」、「A3」など。
- (3)プリモあるいはセコンドの演奏者が1人で複数の音を同時に演奏する（和音を弾く）場合には、それらの音名を「,（カンマ）」で繋いで列挙する。
- (4)プリモとセコンドとの間は「:（コロン）」で繋ぐ。
- (5)ある瞬間、一方のみが発音し、他方は発音する音がない場合、発音する音がない方は音高名を指定しない。
- (6)「コメント」は記述してもしなくてもよい。

【0042】

具体例として図3に示す楽譜は、童謡「たき火」の冒頭4小節であり、この楽譜を図2の記述形式でデータ化したものが図4に示される。ただし、図3では、上段がプリモ、下段がセコンドの楽譜である。

【0043】

図4において、「*S」の箇所は、小節の冒頭でもフレーズの頭でもないが、特に戻り易い箇所として設定した例である。楽譜データは上記のように通常のテキストエディタで容易に編集可能な形式であるため、個々の演奏者に応じて、特に戻り易い箇所がある場合は、このように「*S」マーカなどを利用者が自由に埋め込んで、自分用に最適化することも可能である。

【0044】

この図4の記述データに従って、図5に示す楽譜データが、図1の楽譜データベース2に格納される。

【0045】

なお、図5の楽譜データで、ポジション（位置）とは楽譜の最初からのプリモとセコン

10

20

30

40

50

ド両方を通しての音符番号を示す。同時に複数の音が演奏される箇所は、それらの音すべてに同じポジション番号を付与する。たとえばポジション1や5などは、それぞれ3つの音に同じポジション番号を割り当てている。プリモまたはセコンドの一方に演奏すべき音があるが、もう一方には対応する箇所に演奏すべき音がない箇所は空欄となる。たとえばセコンドのポジション2、4など。リターンの欄には、楽譜データでいずれかの種類のマーカが設定されたポジションの箇所にのみ「*」記号がセットされ、他は空欄となる。現実には、C言語の構造体のリンクによってこのような楽譜データ表を構成している。

【0046】

このような図1の合奏支援システム10において、コンピュータ20は、図6のステップS1において、図7に詳細に示す演奏位置判定処理を実行する。この演奏位置判定処理では、簡単にいうと、楽譜データベース22から取得したプリモ用楽譜データと、逐次入力されてくるプリモの演奏データ(MIDI)とを照合し、現在楽譜上のどの位置(ポジション)が演奏されているかを判定し、この結果を、続くステップS3すなわち図8に詳細に示す演奏音高取得処理に通知する。

10

【0047】

ステップS3すなわち演奏音高取得処理では、簡単にいうと、楽譜データベースから取得したセコンド用楽譜データを参照し、現在のプリモの演奏位置に対応する箇所をセコンドの楽譜上で見だし、そこからセコンドが今演奏すべき音の音高データを取得し、これをステップS5すなわち図9に詳細に示す音高データ差し替え処理に渡す。

【0048】

20

音高データ差し替え処理では、簡単にいうと、セコンド用の入力インタフェースから入力された演奏音のMIDIデータのうち、音高データを指定する値(MIDIノートナンバー)のみを、演奏音高取得処理によって取得した音高データに差し替える。この際、その他のデータ(発音時刻や、音の強さに対応するベロシティ値など)はすべてセコンドが入力した値を保持する。

【0049】

こうして、音高データだけが差し替えられたセコンドの演奏データ(第2演奏データ)がプリモの演奏データ(第1演奏データ)とともにMIDI音源16(図1)に入力され、スピーカ18から音として出力される。つまり、MIDI音源16およびスピーカ18が発音手段として機能する。

30

【0050】

したがって、セコンドは鍵盤上のどの鍵を打鍵しても、各時点で演奏すべき正しい音高の音出力される。一方、音の強弱や発音タイミングなどの音楽表情に関わる要素はすべてセコンドが演奏したまま出力される。この結果、セコンドは楽譜どおりのメロディを容易に再現でき、かつ表情づけした通りに演奏に反映できる。

【0051】

また、プリモが演奏しているにもかかわらず、セコンドがしばらく休止した場合(打鍵しなかった場合)でも、コンピュータ20はプリモの演奏データから演奏位置を認識しているため、いつセコンドが再開しても、正確な音高でプリモと合わせることができる。逆に、プリモが演奏を休止したにも拘らずセコンドが打鍵した場合は、1回目の打鍵はプリモが次に演奏する予定の個所に相当した音高が鳴るが、それ以後はプリモが演奏し始めるまで、音が出なくなるように設定されている。

40

【0052】

ここで、図7に示す演奏位置判定処理について、詳細に説明する。ただし、この図7は、プリモの打鍵(入力インタフェース12の操作)というイベントで駆動されるもので、演奏時点 t_j においてプリモが P_j の音を演奏したものと表されていることを、予め留意されたい。

【0053】

演奏位置検出(いわゆるScore Following/Tracking)に関する研究は80年代より非常に多数なされている。過去のシステムは、「挿入」、「音高誤り」、「脱落」の3種類の

50

誤りに対して対処しているが、いずれも基本的に大きな誤りは起こらない演奏を前提としている。

【 0 0 5 4 】

これに対して、この発明の合奏支援システム 10 では、初心者による練習段階でのきわめて不完全な演奏を取り扱う。この場合、上記のように、弾き直しが非常に多く発生し、演奏位置が何度も繰り返して大きく遡ることが頻発する。したがって、従来のシステムではこのような弾き直しに対処できない。

【 0 0 5 5 】

そこで、この実施例では、先の非特許文献 1 で Dannenberg が提案した D P マッチングによる手法を拡張し、弾き直しに対処可能な演奏位置検出手法を改良した。

10

【 0 0 5 6 】

この実施例の演奏位置検出方法では、Dannenberg の手法同様、各音の楽譜上での音価と演奏音の音長は無視して、音高のみのマッチングで演奏位置を判断する。これは、初心者の演奏では演奏時の音長の変動が極めて大きく、楽譜には無い長い停止も頻発するため、音価と音長はマッチングの対象として扱えないという判断に基づく。

【 0 0 5 7 】

プリモ用の楽譜に含まれる音の数を N とする。ただし、和音のように複数の音が同時に発音される場合、その個所の音数は、同時に発音する音の数に拘らず「1」とし、最高音のみをマッチングの対象とする。さて、プリモが演奏開始から j 番目の音 P_j を演奏したとき(この時刻を「演奏時点 t_j 」とする)、前回のマッチングで演奏時点 t_{j-1} での楽譜上の演奏位置が S_i ($1 \leq i \leq N$) の音であると判定されていたとする。このとき、 P_j が楽譜上のどの音にあたるかの判定は、以下のアルゴリズムによって行う。

20

【 0 0 5 8 】

(1) 楽譜上のすべての音 S_k ($1 \leq k \leq N$) の音高 $Pitch(S_k)$ と、 P_j の音高 $Pitch(P_j)$ とを比較し、すべての音についてプリモの演奏時点 t_j における重み $W(S_k, t_j)$ を以下の数 1 に示す方法で評価する。

【 0 0 5 9 】

[数 1]

(a) if $Pitch(S_k) = Pitch(P_j)$

then $W(S_k, t_j) := W(S_{k-1}, t_{j-1}) + 1$

30

(b) else $W(S_k, t_j) := W(S_{k-1}, t_{j-1}) - 1$

ただし、 $W(S_{k-1}, t_{j-1}) - 1 < 0$ ならば $W(S_k, t_j) := 0$ とする。

【 0 0 6 0 】

このことを図 7 で示すと、ステップ S 1 0 1 で音数 k を「1」に設定する。最初は $k < N$ (総音数) であるから、ステップ S 1 0 3 で“YES”となり、次のステップ S 1 0 5 で、コンピュータ 2 0 (図 1) は、音 S_k における音高 $Pitch(S_k)$ と、音 P_j における音高 $Pitch(P_j)$ とを比較する。このステップ S 1 0 5 で“YES”なら、コンピュータ 2 0 は、次のステップ S 1 0 7 で、時点 t_j での音符 S_k の重み $W(S_k, t_j)$ として、 $W(S_{k-1}, t_{j-1}) + 1$ を設定する。そして、ステップ S 1 0 9 で音数 k をインクリメントして先のステップ S 1 0 3 に戻る。ただし、この実施例では、ステップ S 1 0 7 で加える値を「1」としたが、この値は「1」でなければならないというものではなく、他の適宜の値を加算するようにしてもよい。

40

【 0 0 6 1 】

ステップ S 1 0 5 で“NO”なら、コンピュータ 2 0 は、ステップ S 1 1 1 で、時点 t_{j-1} での音符 S_{k-1} の重み $W(S_{k-1}, t_{j-1})$ が「1」より大きいかどうか、すなわち $W(S_{k-1}, t_{j-1}) - 1 > 0$ かどうか判断する。“YES”なら、次のステップ S 1 1 3 で時点 t_j での音符 S_k の重み $W(S_k, t_j)$ として $W(S_{k-1}, t_{j-1}) - 1$ を設定する。ただし、ステップ S 1 1 1 で“NO”なら、次のステップ S 1 1 5 で、時点 t_j での音符 S_k の重み $W(S_k, t_j)$ として「0」を設定する。

【 0 0 6 2 】

50

このようなステップS103 - S115をステップS103で“NO”が判断されるまで、つまり、すべての音について重みを評価するまで、繰り返し実行する。すべてのポジションで重みを評価した後は、原則的には、最大の重みを示す演奏位置を次の演奏位置として同定する。

【0063】

すなわち、すべてのポジションについて重みWを評価したなら、ステップS103で“NO”となり、コンピュータ20は、続くステップS117で、時点 t_j での演奏位置 S_{i+1} における重みが、時点 t_{j-1} での演奏位置 S_i における重みに「1」加えたものと等しいかどうか、つまり $W(S_{i+1}, t_j) = W(S_i, t_{j-1}) + 1$ かどうか判断する。つまり、演奏位置が正しく更新されているかどうか判断する。ただし、この実施例では、ステップS117で加える値を「1」としたが、この値は「1」でなければならないというものではなく、他の適宜の所定値を加算するようにしてもよい。

10

【0064】

ここで、このステップS117での条件判定について詳しく説明する。このステップS117では、演奏時点 t_j における演奏位置の判定を試みている状態である。そして、演奏時点 t_{j-1} での演奏位置が S_i である。したがって、 $W(S_i, t_{j-1})$ とは、前回の判定(t_{j-1})時点での判定で「ここが演奏位置だ」と判定した箇所の重みである。そして、 $W(S_{i+1}, t_j)$ とは、楽譜上で S_i の次の音 S_{i+1} の、今回(t_j 時点)での重み付け処理の結果得られた重みである。そして、 $W(S_{i+1}, t_j)$ の値が $W(S_i, t_{j-1})$ の値に「1」を加えた値になっていた場合($W(S_i, t_{j-1})$ が図7のS107の処理によって1加算されていた場合)に、 S_{i+1} を t_j 時点での演奏位置と判定するものである。

20

【0065】

そして、このステップS117で“YES”なら、演奏位置が正しいのであるから、次の演奏位置 S_{i+1} を t_j 時点の演奏箇所として、図7の演奏位置判定処理(図6のステップS1)を終了する。

【0066】

(2)もし、 $W(S_{i+1}, t_j) < W(S_i, t_{j-1}) + 1$ であれば、ステップS121以降を実行して、演奏誤り(弾き直しを含む)への対応を行う。つまり、ステップS117で“NO”が判断されるということは、プリモは演奏を間違ったことを意味し、以下、正しい演奏位置を同定する処理を実行することになる。

30

【0067】

まず、弾き直しについて、詳しく説明すると、「1」から「i」の範囲にあって、かつ楽譜上の小節、フレーズ、段およびページ頭など、前述の実験(表1)で得られた「弾き直しが発生し易い箇所」として予め楽譜データ上に指定されている箇所すべての重みを、 $W(S_{i+1}, t_j) - m$ (m は正の定数)とする。つまり、弾き直し位置マーカの設定されている演奏位置の重みをすべて最大より「 m 」小さい値に設定する。ただし、その箇所の元の重みがこの計算で得られた重みより大きい場合は、値を変更しない。なお、発明者等の実験において m の値は、経験的に「2」としている。 m の値を小さくすると、弾き直しへの追従性が向上するが、一方で類似したパターンが繰り返し現れる楽曲の場合、軽微な誤り(音脱落など)で演奏箇所の認識誤りを招く可能性が高くなる。つまり、 m 個の音数だけ余裕を見て「弾き直し箇所」と思われる箇所からの音が m 個連続して弾かれたとき初めて「弾き直し」として判定するようにしている。これによって、1音程度の音の抜けや挿入などの微少な誤りに過剰に反応しないようにしている。

40

【0068】

ついで、挿入、音高誤り、脱落への対応について説明する。位置 S_{i-r} から S_{i+r} までの範囲の音すべてについて、重みを $W(S_{i+1}, t_j)$ と同じにする。つまり、 t_{j-1} 時点での現在演奏位置の前後 r 音ずつを t_j 時点での演奏位置としての可能性を高く評価することにより、 r 個までの挿入や音高誤り、脱落に対処している。なお、音数 r は正の定数とし、実験においては r の値は、経験的に「2」としている。この値を大きくする

50

と、より多数の音にわたる誤りに対処できるが、大きくし過ぎると可能性の範囲が広がりすぎ、逆に誤った一致箇所を見いだしてしまう危険性が高くなる。

【0069】

(3)そして、重み $W(S_k, t_j)$ (ただし、 $1 \leq k \leq N$) 最大値をとる個所を現在の演奏位置とするのであるが、もし同じ重みの個所が複数ある場合は、以下の順に優先する。
(a) 重み $W(S_k, t_j)$ が最大値をとる箇所が唯一であるならば、そのときの重み $W(S_k, t_j)$ が最大値となる位置 S_k を現在の演奏位置とする。

(b) 重み $W(S_k, t_j)$ が最大値をとる箇所が複数個あるならば、現在の演奏位置が次のようにして決められる。

【0070】

(i) S_i に最も近い最大値を取る個所 S_k が唯一であるならばその個所 S_k を現在の演奏位置とする。

【0071】

(ii) S_i に最も近い最大値を取る個所が複数個ある場合、すなわち S_i から等距離の位置 S_{i-d} と S_{i+d} とに最大値が生じる場合は、 S_{i-d} を現在の演奏位置とする。

【0072】

このことを図7を参照して説明すると、図7のステップS117で“NO”となったときには、コンピュータ20は、次のステップS121において、すべての弾き直し位置 S_h (表1においてリターンのマークが付与されている箇所)の重みとして、それらの重みを、最大より m だけ小さい値にするために、 $W(S_{i+1}, t_j) - m$ を設定する。ついで、ステップS123で、コンピュータ20は、ステップS123で、位置 $S_g (= i - r \leq g \leq i + r)$ での重み $W(S_g, t_j)$ を $W(S_{i+1}, t_j)$ とする。つまり、そのときの演奏位置の前後の演奏位置の重みを等しくする。

【0073】

そして、ステップS125において、重み $W(S_k, t_j)$ ($k = 1 \sim N$) が最大値をとる個所が複数あるかどうか判断する。“NO”の場合には、ステップS127において、コンピュータ20は、重み $W(S_k, t_j)$ が最大値となる位置 S_k を現在 t_j の演奏位置とする。

【0074】

ステップS125において“YES”が判断されたときには、すなわち、重み $W(S_k, t_j)$ が最大値を取る個所が2以上存在するときには、続くステップS129において、コンピュータ20は、位置 S_i に最も近い最大値の演奏個所 S_k が複数存在するかどうか、判断する。ステップS129で“YES”のときには、コンピュータ20は、ステップS131において、 $k < i$ の演奏個所 S_k を現在の演奏位置とする。ステップS129で“NO”なら、ステップS133で、コンピュータ20は、位置 S_k を現在の演奏位置とする。

【0075】

以上のアルゴリズムによって、従来から扱われてきた3種類の演奏誤りに加えて、初心者の演奏で頻発する「弾き直し」に対しても追従できる頑健な演奏追跡処理が実現される。

【0076】

実験によれば、「弾き直しへの対応」を行った場合と行わなかった場合とでは、前者では、たとえば3音の誤認識の後正しい位置を検出した。これに対して、同じ曲でも後者の場合には、弾き直し後に正しい位置を検出するまでに13音の誤認識が生じた。このように、この実施例の演奏位置判断アルゴリズムを採用した場合には、弾き直しへも十分実用的なレベルでの追従が可能となった。

【0077】

次に、図8を参照して、演奏音高取得処理(図6のステップS3)を説明する。

【0078】

演奏音高取得処理も、図7と同様に、プリモの打鍵というイベントによって駆動される

10

20

30

40

50

。したがって、図8の最初のステップS301で、コンピュータ20は、プリモのそのときの演奏位置 S_i のデータを受ける。そして、ステップS303における一定時間 T_w ミリ秒の経過の後、ステップS305で、コンピュータ20は、位置 S_{i+1} にセコンドが演奏すべき音が楽譜上にあるかどうか判断する。“NO”なら、ステップS307において、コンピュータ20は、セコンドが演奏すべき音を「なし(0)」に設定する。一方、ステップS305で“YES”の判断をしたとき、つまり、位置 S_{i+1} においてセコンドが演奏すべき音があるとの判断をしたときには、次のステップS309において、コンピュータ20は、そのときセコンドが演奏可能な音をたとえば図5のセコンドの欄に指定された音に設定する。

【0079】

したがって、プリモが誤りなく演奏している場合、位置(ポジション) S_i は1、2、3、4、...と順に演奏位置がこのルーチンに投入される。たとえば $S_i = 9$ とプリモの演奏ポジションが通知された場合、 T_w ミリ秒の間をとった後、ポジション $S_{i+1} = 10$ の位置にセコンドが演奏する音が指定されているかどうかを調べる。図5の例では、セコンドのポジション10の位置には音が指定されていないので、ステップS305での判定は“NO”となる。したがって、セコンドが演奏可能な音は「なし」と設定される。この状態では、セコンドが打鍵しても何も発音しない。

【0080】

引き続きプリモが誤りなく演奏を続けた場合、 $S_i = 10$ でも同様の処理が行われ、セコンドは何も発音できない状態が続く。さらに、プリモの演奏位置が $S_i = 11$ となったとき、 T_w ミリ秒の間を置いた後、ポジション $S_{i+1} (= 12)$ のセコンドの演奏音を確認する。表1の例では、セコンドのポジション12には「C5」の音が指定されているので、ステップS305の条件判定の結果は“YES”となり、ここでセコンドが演奏可能な音として、「C5」が設定され、発音可能状態になる。以上の手順で、プリモに誤りがない場合のセコンドの演奏可能音は切り替えられていく。

【0081】

次に、セコンドが誤って発音すべき音を発音し忘れた場合について説明する。セコンドの演奏可能音は、セコンドによる打鍵がない限りは実際には発音されない。セコンドによる発音がされないままの状態、プリモが次のポジションを演奏し、新しい演奏位置情報 S_i が図8のアルゴリズムに入力されると、セコンドが先の音を実際に発音したかどうかにかかわらず、このアルゴリズムが実行され、セコンドが演奏すべき音も更新される。たとえばポジション3の位置でセコンドが演奏し、「G4」の音を発音したとする。その後、プリモがポジション4、5と演奏を進めると、セコンドの演奏可能音は「B4」に更新される。しかし、この状態でセコンドが誤って演奏しないままプリモがさらにポジション6、7と演奏を進めると、「B4」の音は発音されないまま、セコンドの演奏可能音は「G4」に切り替わる。したがって、この状態でセコンドが演奏すると、演奏し忘れた「B4」ではなく「G4」の音が発音する。このようにして、セコンドが誤っていくつかの演奏音を飛ばしてしまっても、セコンドの演奏可能音は常にプリモの演奏位置に対応したものとなるため、セコンドが途中で演奏を再開しても、飛ばした音ではなく、今弾くべき音が正しく発音される。

【0082】

一方、セコンドが誤って本来演奏すべきでない箇所でも余分に演奏した場合について説明する。たとえばポジション9でセコンドが演奏し「A4」の音を発音したとする。この直後、プリモがポジション10の音を演奏しないうちにセコンドが再び打鍵した場合を考える。もしこの再打鍵がプリモのポジション9の音(C5)の打鍵より T_w ミリ秒以内であれば、再度「A4」の音が発生する。一方、プリモのポジション9の音(C5)の打鍵より T_w ミリ秒以上経過した場合は、次のポジション10の音が発音される。ただし、現在の例ではセコンドのポジション10には演奏可能音が指定されていないため、この場合は何も発音しないことになる。引き続き、プリモがポジション10に演奏を進めた場合、セコンドの演奏可能音はポジション11に指定されていない。ゆえに、ここでセコンドが打

10

20

30

40

50

鍵しても、依然として何も発音されない。さらにプリモがポジション 11 の音を発音し、Twミリ秒以上経過すると、セコンドの演奏可能音はようやくポジション 12 の「C5」に設定される。以上のように、セコンドが何度余分な打鍵をしても、セコンドの演奏位置だけが一方的に先走ることは起こらない。余分な音の挿入が生じる可能性はあるが、それによってプリモとセコンドの演奏位置がずれることはない。

【0083】

以上はプリモが演奏を誤らない場合の処理の流れの説明であるが、次にプリモが演奏を誤って、いくつかの音を飛ばした場合の処理を説明する。この場合も、基本的に処理の概要は変わらない。プリモが演奏を誤った場合は、演奏位置判定ルーチンから通知されるポジションデータ S_i が前回と不連続になるだけであり、そのことはこのアルゴリズムの処理に関係しない。たとえば、プリモがポジション 12 の音を演奏した後、ポジション 9 に戻った場合を考える。この場合、演奏位置判定モジュールから通知される演奏位置情報は、演奏位置判定が理想的に動作すれば、 $S_i = 12$ の次に $S_i = 9$ が図 7 に示す音高取得のアルゴリズムに通知される。 $S_i = 12$ が通知されて Twミリ秒経つと、セコンドの演奏可能音はポジション 13 の「B4」となっている。この状態でプリモがポジション 9 の音を弾き、ほぼ同時にセコンドが打鍵すると、セコンドの演奏音はそのままポジション 13 の「B4」となってしまう。これは回避できない誤りである。しかし、演奏ポジションとして $S_i = 9$ が通知されているので、次のセコンドの演奏可能音はポジション 10 の音となる。

【0084】

この例ではセコンドのポジション 10 の位置は空白なので、この場合は演奏可能音「なし」となる。以下、プリモが誤り無く演奏を続ければ、先に述べたのと同様に処理が進む。

【0085】

ここで、ステップ S303 での Twミリ秒の遅延の意味について説明する。楽譜を単純に用いるならば、たとえばプリモが $S_i = 13$ の音（すなわち「D5」）を弾いた瞬間に、セコンドの演奏可能音も同時にポジション 13 の「B4」にすればよいことになる。しかし、現実にはプリモとセコンドの演奏タイミングが全くずれなしで一致することはありえない。プリモが若干先行する場合もあれば、セコンドが若干先行する場合もある。プリモが先行し、セコンドが遅れる場合は、上記のような同時切り替えを行っても問題なく、ポジション 13 の位置でプリモが「D5」の音を弾いた直後にセコンドは「B4」の音を弾くことができる。しかし、セコンドが先行し、プリモが遅れた場合は、セコンドは切り替えられていない「前の音」を演奏してしまうことになる。つまり、ポジション 13 の箇所セコンドが先行してしまうと、セコンドは本来弾くべき「B4」の音ではなく、前の音である「C5」の音を弾いてしまうことになり、不都合である。

【0086】

この問題を回避するために、Twミリ秒の間を設けて、その後にセコンドの演奏可能音を切り替えている。Twミリ秒の時間は、プリモとセコンドが同時発音すべき時に、セコンドがプリモよりも遅れる可能性のある時間幅の最大値で、現在のシステムでは 200ミリ秒としている。つまり、このアルゴリズムでは、たとえばプリモが $S_i = 11$ の「E5」の音を弾いた Twミリ秒後に、セコンドが弾ける音はポジション 12 の「C5」の音に設定される。この状態で、プリモが $S_i = 12$ の「E5」を弾くと、その瞬間から Twミリ秒間は、セコンドは現在設定されている「C5」を弾ける状態が維持される。つまり、セコンドはプリモよりも遅延すること Twミリ秒未満であれば、ここで弾くべき正しい音である「C5」を演奏できる。しかし、Twミリ秒以上セコンドによる発音が無い場合は、セコンドが今弾くべき音を「弾き忘れた」とシステムは見なし、次のポジション 13 の「B4」の音を弾くべき音に設定し、次のプリモの $S_i = 13$ の発音に備える。このため、プリモが $S_i = 13$ の「D5」の音を演奏していない場合でも、セコンドの演奏音はすでにポジション 13 の音になっているため、セコンドがプリモに若干先行して発音してしまっても、演奏すべきポジション 13 の「B4」が正しく演奏される。以上のメカニズム

によって、プリモとセコンドの演奏タイミングのずれによる演奏音の不一致の問題を吸収している。

【0087】

続いて、図9に示す音高データ差替え処理（図6のステップS5）について詳しく説明する。この図9の処理は、セコンドによる打鍵というイベントで駆動され、最初のステップS501で、コンピュータ20は、楽譜データを参照して、位置 S_{i+1} にプリモの演奏すべき音があるかどうか判断する。“YES”ならそのままこのルーチンを抜けるが、“NO”なら、次のステップS503において、コンピュータ20は、セコンドが演奏可能な音を、楽譜データの位置 S_{i+1} のセコンドの欄に指定されている音に変更する。

【0088】

具体的に説明すると、図9の音高データ差替えのアルゴリズムは、プリモに演奏すべき音がなく、セコンドにのみ演奏すべき音がある場合の処理を行っている。ここで扱っている例で言えば、ポジション14～16の場所がこの場合にあたる。プリモがポジション13の「D5」を演奏してTWミリ秒経過すると、セコンドの演奏可能音はポジション14の「A4」になっている。この状態でセコンドが打鍵するとこの「A4」の音が発音されると同時に、この図9のアルゴリズムが動作する。

【0089】

プリモ側の演奏位置 S_i は13のままなので、ポジション S_{i+1} （=14）のプリモの楽譜データを調べ、プリモがポジション14で演奏する音があるかどうかを確認する。この例では、プリモにはポジション14で演奏するデータがないので、条件判定は“NO”となる。この場合、現在の演奏位置を $S_i = S_{i+1}$ （=14）とし、セコンドの演奏可能音を S_{i+1} （=15）の「G4」に設定する。この状態でセコンドが再度打鍵すると、「G4」の音が発音されると同時に、再度第2のアルゴリズムが動作し、以上と同様の処理を実施する。このようにして、プリモに演奏データが無い場合は、プリモではなくセコンドが演奏進行のイニシャティブを取って、プリモの演奏とは無関係に独自に演奏を進行させられるようになる。

【0090】

なお、セコンドがまだポジション16まで演奏し終わっていない状態でプリモがポジション17の音「E5」を演奏した場合、演奏位置 S_i は一気にポジション17に移動し、セコンドの未演奏音は無視されることになる。これは、演奏位置判定ルーチンが、セコンドの演奏位置を考慮しておらず、プリモの演奏データと楽譜データのみを参照しているためである。しかし、これはアルゴリズムの不備ではない。なぜならば、この実施例のシステムはあくまでプリモ主導のシステムであり、セコンドが主導となるのは限られた箇所であつても、プリモが演奏を再開した場合は、即座にプリモに主導権を渡す必要がある。この結果、プリモが間違えて早く次の演奏位置に移動してしまっても、セコンドは正しくプリモの演奏位置に相当する演奏音を発音できる。

【0091】

ところで、セコンドの打鍵があれば無条件にこの図9に示すアルゴリズムが起動されるので、実際にはプリモに演奏可能音が無い場合だけでなく、プリモに演奏可能音がある場合にもこの図9のアルゴリズムは起動する。しかし、その場合に、演奏位置 S_{i+1} にプリモが演奏すべき音があるかの条件判定が“YES”となるため、演奏ポジション S_i もセコンドの演奏可能音も変更されない。つまりこの場合は結局何もしないでこのアルゴリズムは終わる。

【0092】

以上のような、図8および図9に示す2つの簡単なアルゴリズムの非同期動作によって、プリモとセコンドの演奏位置が常に一致するように動作する。

【0093】

このように、この発明の背景となる図1の合奏支援システム10で発明者等が何組かのペアに頼って行った実験によれば、いずれのペアも、この支援システムを利用することに

10

20

30

40

50

よって、すぐに合奏（連弾）が可能になった。

【 0 0 9 4 】

図 1 の合奏支援システム 1 0 では、初級者が誤った後に弾き直しを開始しやすい箇所に予めマーカをつけたため、合奏途中にプリモが勝手に弾き直しても、システムはすぐに追従することができ、セコンドは適切な音を出力することができる。しかしながら、人間同士の合奏でもそうだが、いきなりプリモが前に戻って弾き直しをした場合に即座にその弾き直し箇所を見つけて合わせることはかなり難しい。実施例の合奏支援システム 1 0 でも、プリモが弾き直ししてから少なくとも 3 音程度の間はセコンドも正確な音を出すことはできない。

【 0 0 9 5 】

一方で、このような 1 曲を通した演奏途中での弾き直しではなく、演奏を途中で止めたり、最後まで演奏した後に、誤りのあった箇所だけを部分的に練習しようとするところがある。そこで、図 1 0 以下に説明するこの発明の一実施例の合奏支援システム 1 0 では、プリモが誤りをした箇所を 2 人で練習するとき、1 音目からセコンドも正確な音を出せるようにすることを意図している。

【 0 0 9 6 】

また、合奏支援システム 1 0 の弾き直しに対応する手法は、演奏と映像とを同期的に再生する場面にも応用することができる。各奏者が演奏すべきパートの教師による模範演奏の映像とともに独習する場合には、奏者が弾き直しても、映像が自動的に奏者の位置を追従して適切な映像を再生することができる。たとえば、合奏支援システムを使うまったくの初心者であるセコンド（第 2 奏者）が打鍵するタイミングを知るために、教師による模範映像を前にしながら、プリモと合奏練習することができる。この場合は、プリモが弾き直しすると、セコンドの模範映像もプリモの弾き直し位置に戻ることができる、という使い方である。

【 0 0 9 7 】

図 1 0 の実施例の合奏支援システム 1 0 は、図 1 実施例のシステムに比べて、楽譜データベース 2 2 が楽譜および映像データベース 2 2 a に変更され、さらに、コンピュータ 2 0 にタッチパネル表示器 2 4 が付設された点が異なるだけで、他の部分は共通である。共通部分は説明は省略する。

【 0 0 9 8 】

なお、タッチパネル表示器 2 4 とは、たとえば C R T、液晶、プラズマなどからなる表示器の表示画面上にタッチパネルが付設されていて、ユーザがペンや手あるいは指でのタッチパネルの任意の位置にタッチするとそのタッチ位置の座標が検出され、したがって、ユーザのタッチ位置がどこか判別できる、そのような表示器のことである。

【 0 0 9 9 】

楽譜および映像データベース 2 2 a には、先に説明した図 5 に示すような楽譜データが格納されているとともに、図 1 1 に示す連動テーブルのように、各楽譜ポジション毎に、そのポジションで再生する映像データのタイムコードが、開始時間と終了時間として設定されている。つまり、この図 1 0 のシステム 1 0 では、プリモの演奏の進行に合わせて、予め準備している映像を同期的に再生するものであるが、上述のようにプリモの「弾き直し」に対応できるようにするために、楽譜ポジション毎に、そのポジションでの映像の再生開始のタイムコードが開始時間として、再生終了のタイムコードが終了時間として、それぞれ設定される。たとえばポジション「 1 2 」まで進んだときタイムコード「 0 0 0 0 0 0 0 0 1 1 0 1 」から映像が再生されるが、間違ったのでポジション「 9 」まで戻って弾き直ししたとすると、映像は、タイムコード「 0 0 0 0 0 0 0 0 0 8 0 1 」で示す位置に飛んで、そのポジション「 9 」に固有に割り当てられている映像が正しく再生されるのである。

【 0 1 0 0 】

なお、このように演奏と同期して再生される映像は、基本的にはどのような種類のものであってもよいが、たとえばピアノを弾く手の映像（模範映像）などが考えられる。そし

10

20

30

40

50

て、そのような映像のデータは、図5に示す楽譜データや図11に示す連動テーブルとともに、楽譜および映像データベース22aに格納されている。

【0101】

図10の実施例においても、図6のフローチャートで示す手順に従ってプリモとセコンドの演奏が進行されるのであるが、コンピュータ20は、図6の演奏位置判定処理S1において、さらに、図12に示すフローチャートを実行して、プリモの演奏に間違いがあったかどうか判定する。

【0102】

図12のステップS601では、まず、コンピュータ20は、このコンピュータ20内のメモリに設定されたタイマTに所定数値たとえば「4」を設定する。このタイマTは、演奏が停止されたかどうかを判断するためのタイマである。実施例でいえば、プリモが打鍵してからたとえば4秒間打鍵しなかった場合に演奏が停止されたと判断できる。このタイマTは、図示しない別のタイマ割り込みによって、1秒毎にディクリメントされ、T=0になったとき、プリモの演奏が停止したと判断する。

【0103】

次のステップS603では、コンピュータ20は、レジスタShおよびSpを初期化する。レジスタShは間違いのあった位置(ポジション)を記憶しておくためのレジスタであり、レジスタSpは、コンピュータ20が推奨する、弾き直し位置(推奨戻り位置あるいは練習開始位置)を記憶しておくためのレジスタである。この初期設定が終わった後、コンピュータ20は、ステップS101-S117を実行して、プリモの演奏位置に誤りがあったかどうか判断する。

【0104】

ただし、これらのステップS101-S117は先の図6に示すステップS101-S117と同一であるので、ここでは、重複する説明は省略するが、ステップS117では、楽譜上での次の位置での重み $W(S_{i+1}, t_j)$ の値と今回の位置での重み $W(S_i, t_{j-1})$ の値に「1」を加えた値とが同じであるとき、演奏位置に間違いはなかったと判断する。

【0105】

したがって、ステップS117で“YES”が判断されたときには、演奏位置は正しいものとして、次のステップS605で演奏間違いはないとの判定を行い、たとえば図7のステップS119を実行する。

【0106】

逆に、ステップS117で“NO”が判断されたときには、次のステップS607で、コンピュータ20は、次の演奏位置 S_{i+1} を間違い位置としてレジスタShに設定される。したがって、レジスタShには、プリモが演奏を停止するまでに間違った演奏位置のすべてが保存されることになる。

【0107】

図12のステップS601で設定したタイマTが実施例では4秒をカウントしたとき、すなわち、プリモが演奏を停止したときに図13に示す処理が開始される。

【0108】

最初のステップS701では、コンピュータ20は、レジスタShを参照して、その時刻 T_j における演奏位置(S_k とする)までに、誤り位置があったかどうか、すなわち $S_h - 1$ か、かつ $S_h < S_k$ となる S_h があるかどうか判断する。このステップS701で“NO”が判断されるということは、プリモの演奏は停止したものの、間違いはなかったことを意味し、この場合には、そのまま終了する。

【0109】

ステップS701で“YES”が判断されたときには、少なくとも1つ、間違いがあったことが登録されているので、次のステップS703で、コンピュータ20は、それぞれの誤り位置 S_h より所定 m (実施例では経験的に「4」としている。)小節前までに、つまり停止位置から4小節以内に、推奨戻り位置がレジスタSpに登録されているかどうか

10

20

30

40

50

判断する。なお、この推奨戻り位置（弾き直し位置または練習開始位置）とは、先の図5の表で「*」で示したリターン位置のことを意味する。つまり、この実施例のシステムでは、プリモが弾き直しのために戻り易い位置をリターン位置マーカとして予め設定するようにしているので、そのリターン位置マーカをSpとして検出するようにしている。したがって、このステップS703では、図4のような楽譜データを参照して、戻り位置Spの有無を検出する。

【0110】

そして、ステップS703で“YES”のときには、ステップS705で、すべてのマーカ箇所（*印）Spに相当する電子楽譜上の五線譜の上にマーク（実施例では二重丸）をつけ、ステップS709において、その五線譜の電子楽譜をタッチパネル表示器24（図10）に表示して終了する。ステップS703で“NO”のときには、ステップS707でその曲の最初にマーク（実施例では二重丸）をつけて終了する。

10

【0111】

一例が図14に示される。図14の例の電子楽譜では、プリモのパートが上段に五線譜として表示され、セコンドのパートはその下にリズム譜として表示される。その電子楽譜の五線譜の中に「Sh」の記号が示されているが、その位置は間違えた位置である。図14の例では、ポジション「11」すなわち第3小節目の第3番目の音符の位置が間違い位置である。そして、その位置より前に3つの二重丸マークが表示されている。ただし、括弧書きで示した先頭の二重丸マークはステップS703で“NO”の判断がなされたとき、つまり、間違い位置Shより4小節前までにマーカ（*）が設定されていないと判断されたときに付されるものである。つまり、図14の例で、仮に、第2小節にも第3小節にも二重丸マークがないと判断したときには、曲の先頭に二重丸マークが付される。図14の例では、第2小節および第3小節の2箇所にも二重丸マークが表示されているので、この例の場合には、間違い位置Shより前に3つの推奨戻り位置Spが設定されていることを意味している。

20

【0112】

図14のようにタッチパネル表示器24に表示された電子楽譜を見て、推奨戻り位置の中で自分が戻りたい位置をタッチする。コンピュータ20がプリモのタッチ位置を検出し、そのタッチ位置が推奨戻り位置マーカを表示している位置であると、それに応じて、図15の処理が実行される。

30

【0113】

図15の最初のステップS801では、コンピュータ20は、演奏位置kとして、レジスタ（図示せず）に、プリモが停止したtj時の演奏位置として設定する。続くステップS803で、コンピュータ20は、レジスタmaxに「-1」を設定する。このレジスタmaxは、上記停止位置から前の各ポジションのうち最大の重みWを有するポジションを探すために利用される。

【0114】

詳しく述べると、ステップS803でレジスタmaxに-1を設定した後、次のステップS805において、コンピュータ20は、kが「0」より大きいかどうか、すなわち演奏位置が曲の先頭（ポジション0）より後かどうか判断する。通常、“YES”であるから、そのままステップS807に進み、そこでは、コンピュータ20は、tj時間のときの演奏位置Skの重みW(Sk, tj)がレジスタmaxの設定値より大きいかどうか判断する。レジスタmaxには上述のように「-1」を初期値として設定しているので、最初は、このステップS807では“YES”が判断される。したがって、次のステップS809に進み、レジスタmaxの値をそのときの演奏位置の重みW(Sk, tj)に更新する。そして、次のステップS811で演奏位置kを更新して、ステップS805に戻る。このようにして、ステップS805-S811を繰り返し実行することによって、プリモの停止位置より前で最大の重みWを持つポジションを検出する。

40

【0115】

最大重みを検出するために変更している演奏位置が先頭までくると、ステップS805

50

で“NO”が判断されることになり、次のステップS813で、コンピュータ20は、そのときレジスタmaxに設定されている重みWに1を加えた値を、電子楽譜上で表示されている二重丸マーク(図14)のうち、プリモまたはセコンドがタッチしたマークの直前の位置Sp-1の重みとして設定する。そうすることによって、次に弾き始める位置、図14の例では第3小節の最初の音符(第9ポジション)が図7の処理によって設定または決定されることになる。つまり、次に弾き始める1つ手前のポジションに重みの最大値を強制的に書き込む。それによって、プリモが指定した推奨戻り位置Spから確実に追跡することができる。

【0116】

そして、ステップS815でコンピュータ20は、その位置Spにセコンドの演奏すべき音(休符ではない)があるかどうか、楽譜データを参照して判断する。あれば、ステップS817で、図8のステップS309と同様に、セコンドの演奏音を位置Spで指定されている音に設定する。なければ、図8のステップS307と同様にセコンドの音はないものとして処理する。したがって、演奏の中断後や終了後に曲の途中から練習する場合には、システムが推奨する戻り位置(推奨する練習開始位置)の中でプリモ等が位置を選択または指定することにより、曲の途中であっても、セコンドは1音目から正しい音高を出力することができる。

【0117】

なお、図14に示すように、推奨戻り位置Spが付記された電子楽譜をタッチパネル表示器24に表示しても、プリモが、誤り箇所の少し前(システムが推奨する戻り位置)から練習しないという選択をするかもしれない。この実施例はその場合にも対応できるようにして、一例として、プリモはソステヌートペダル(sostenute pedal:ピアノで3つあるペダルのうちの真ん中のペダル。このペダルは普通の演奏時には、たとえば和音を弾くために、その時弾いている音だけを伸ばしたいときに使用する。)を踏むことによって、弾き直し練習はしない、という意思表示が行える。

【0118】

ソステヌートペダルをプリモが踏んだときの処理が図16に示される。プリモが弾き直しをしないためにソステヌートペダル(図示せず)を踏んだとき、コンピュータ20は、最初のステップS901で間違い位置レジスタShを初期化するとともに、次のステップS903において、タッチパネル表示器24に推奨戻り位置Spが表示されているかどうか判断する。推奨戻り位置が表示されていないときには、コンピュータ20は何もしない(ステップS905)が、表示されている場合には、ステップS907で、レジスタの推奨戻り位置をクリア(初期化)するとともに、次のステップS909でそのマーク(二重丸)を消す。このように、ソステヌートペダルを踏むことによってプリモが推奨戻り位置からの戻り演奏を拒否したときには、電子楽譜の推奨戻り位置のマークが消され、システムは演奏追従を引き続き行う。

【0119】

また、図12のステップS603でも、レジスタShおよびSpの初期化を行わず、いきなり演奏を続けても、弾き直し練習をしないことを選択することはできる。

【0120】

なお、上述の実施例では、推奨戻り位置とともに電子楽譜を表示するためにタッチパネル表示器24を用い、推奨戻り位置をプリモが選択または指定するために、そのタッチパネルを利用するようにした。つまり、戻り位置の指定のための手段(戻り位置指定手段)としてタッチパネルを使用した。

【0121】

しかしながら、このような戻り位置指定手段としては、タッチパネル以外に、たとえばコンピュータ20に関連して設けられたキーボードやマウスのような入力装置が利用可能である。

【0122】

さらに、上述の実施例では、戻り練習をしないことを入力するためにソステヌートペダ

10

20

30

40

50

ルを用いることにした。しかしながら、この場合も、ソステヌートペダルに代えて、上述のような入力装置が利用されてもよい。

【 0 1 2 3 】

また、上述の説明は、プリモとセコンドとが合奏または連弾する場合について説明した。しかしながら、図 1 0 実施例の合奏支援システムでは、プリモが 1 人で演奏する場合にも、上述のように誤り位置を判定し、推奨戻り位置を表示し、それを奏者（ユーザ）に指定させることは可能である。その意味では、図 1 0 実施例の合奏支援システム 1 0 は演奏学習支援システムということもできる。

【 0 1 2 4 】

図 1 7 の実施例の合奏支援システム 1 0 は、図 1 0 実施例のシステムに比べて、さらに、動画カメラ（以下、単に「カメラ」という。）2 6 が付設された点が異なるだけで、他の部分は共通である。このカメラ 2 6 はプリモ用インタフェース 1 2 を操作するプリモ（または学習者）の手元映像を撮影するためのもので、インタフェース 1 2 を操作するプリモ（図示せず）の手元の前方斜め 4 5 度の位置からその手元の映像を撮影するように配置される。このようなカメラ 2 6 の配置位置は、特に限定されるものではないが、この実施例では、プリモまたは学習者の手や指の位置と、プリモの指が打鍵したかどうかを同時に撮影するために、上述の配置位置を採用した。たとえば、プリモの手元の真上から撮影すると、その撮影映像から手指の位置は明瞭に判別できるが、打鍵したかどうかは明瞭には判別できない。他方、プリモの手元の真正面から撮影すると、撮影映像からインタフェース 1 2 の鍵盤を打鍵したかどうかは明確に判別できるが、鍵盤に対する手指の位置は必ずしも明確には判別できない。そこで、真上と真正面との中間位置にカメラ 2 6 を配置することとした。

【 0 1 2 5 】

なお、この実施例においては、楽譜および映像データベース 2 2 a には、ピアノを弾く手の実写映像（模範映像ないし手本映像）の映像データが、図 5 に示す楽譜データや図 1 1 に示す連動テーブルとともに、格納されている。そして、図 1 1 に示す連動テーブルを参照することによって、その模範映像または手本映像の再生および停止タイミングを制御することができる。たとえば、ポジション「2」に対応する手本映像を再生するためには、タイムコード「0 0 0 0 0 0 0 0 0 1 0 1」から開始し、タイムコード「0 0 0 0 0 0 0 0 0 2 0 0」で停止すればよい。同様に、たとえば、ポジション「1 0」に対応する手本映像を再生するためには、タイムコード「0 0 0 0 0 0 0 0 0 9 0 1」から開始し、タイムコード「0 0 0 0 0 0 0 0 0 1 0 0 0」で停止すればよい。

【 0 1 2 6 】

ただし、この手本映像は鍵盤およびそれを打鍵する教師の手指の実写映像であるが、この手本映像についても、手指の位置と打鍵のタイミングとの両方を学習者（プリモ）に明瞭に視認させることができるように、カメラ 2 6 からの手元映像と同様に、斜め 4 5 度前方から撮影した映像であることが望ましい。ただし、手本映像は真上から撮影したものをいい、その手本映像中の打鍵位置または打鍵タイミングは何らかの目印を付与することによって学習者に感知させるようにしてもよい。

【 0 1 2 7 】

さらに、この実施例のタッチパネル表示器 2 4 には、上下に 2 つの表示領域が設定されていて、上側の楽譜表示領域 2 4 a には楽譜および映像データベース 2 2 a から読み出した図 5 に示すような楽譜の全部または一部を表示し、下側の映像表示領域 2 4 b には、楽譜および映像データベース 2 2 a から読み出した手本映像や、上記カメラ 2 6 で撮影した手元映像を表示する。ただし、この図 1 7 実施例では、タッチパネル機能のない表示器が用いられてもよい。

【 0 1 2 8 】

上述のように、この実施例では楽譜の映像を表示器 2 4 の上側領域 2 4 a には表示し、下側領域 2 4 b に手本映像や手元映像を表示するようにした。しかしながら、これらの領域 2 4 a および 2 4 b は表示器 2 4 の横方向に並置されてもよい。ただし、発明者等の実

10

20

30

40

50

験によれば、横方向並置の場合には被験者から視線移動が困難だったとの意見があったので、2つの領域24 aおよび24 bは上下に並置することによって、余分な視線移動による学習者への負担を回避することにした。

【0129】

このように、表示器24の映像表示領域24 bには手本映像または手元映像を表示するのであるが、学習者（プリモ）の視覚的な混乱を避けることができるように、実施例では、その手元映像と手本映像とを同じ縮尺でかつ同じ位置関係で表示するようにした。そのためには、楽譜および映像データベース22 aに収録している手本映像を工夫したり、手元映像を撮影するカメラ26のズームングを工夫したりすればよい。

【0130】

図17の実施例は、基本的には、図10実施例と同様に、学習者がピアノを練習するときを使用できるシステムであるので、プリモが1人だけでも使用することができる。その意味では、図17実施例の合奏支援システム10も図10実施例と同様に、演奏学習支援システムということもできる。

【0131】

そして、この実施例は、図10実施例の戻り位置推奨機能とともに、プリモ（学習者）が楽曲を演奏しているとき、その演奏を停止すると、上記手本映像を表示器24の映像表示領域24 bに表示する機能を有する。ただし、停止直前に音高誤りがあった場合には、その停止位置より手前のフレーズの手本映像を映し出し、音高誤りがなかった場合には、その停止位置より先のフレーズの手本映像を映し出す。

【0132】

したがって、この実施例においても、コンピュータ20は図12に示すフローチャートに従って推奨戻り位置検出動作を実行する。そして、この実施例においては、たとえばコンピュータ20の内部メモリの適宜の領域に割り付けられている演奏停止判定用タイマ（図12のタイマT）が一定時間、たとえば2秒～5秒（デフォルトでは4秒に設定した）学習者による打鍵がなかった場合、図18のフローチャートで示すイベントを処理する。

【0133】

ただし、図17においてインタフェース12からMIDIデータがタイマTでカウントする一定時間の間入力されなかったとき、演奏停止として検出するようにしてもよい。

【0134】

ただし、一定時間打鍵がなかったということは、カメラ26からの手元映像をサンプリングし、そのサンプリングデータをコンピュータ20で処理することによって、検出または判定することができる。たとえば、鍵盤の1つのキーを打鍵したとき、その打鍵したキーは押し下げられる。この押し下げによって、それに隣り合うキーとの間に押し下げに伴う陰影が形成されるので、コンピュータ20は一定時間どのキーに関連してもそのような陰影を検出しなかったとき、学習者が演奏を停止したと判断することができる。

【0135】

コンピュータ20は、学習者の演奏の停止を検出したとき、図18の最初のステップS1001において、時刻 t_j における演奏停止位置 S_k より手前側でありかつその停止位置 S_k に一番近いマーカ S_p （推奨戻り位置）を S_{pa} としたとき、その S_{pa} から、停止位置 S_k までの間に誤り位置 S_h があったかどうか判断する。この誤り位置 S_h は、もし誤っていれば、たとえば図12のステップS607で保存されているので、その誤り位置レジスタ S_h を参照することによって、ステップS1001の判断を実行することができる。

【0136】

ステップS1001で“YES”の判断をしたとき、コンピュータ20は、続くステップS1003において、楽譜および映像データベース22 aから読み出して表示器24の楽譜表示領域24 aに表示している楽譜上の五線譜のマーカ S_{pa} （図19）の位置にマーク（演奏誤りがあったので、たとえば赤色マーク）を表示する。この赤色マークの画像データはコンピュータ20内に、または楽譜および映像データベース22 a内に予め設定

10

20

30

40

50

しておけばよい。

【0137】

そして、次のステップS1005において、コンピュータ20は、マーカSpaの位置から停止位置Skまでの手本映像を表示器24の映像表示領域24bに表示する。このとき、手本映像は楽譜および映像データベース22aから読み出すが、どの映像を再生するかは、図11のような連動テーブルのポジション毎の映像開始時間および終了時間を参照することによって選択することができる。

【0138】

たとえば、マーカSpaから停止位置Skまでの手本映像を表示するとして、Spaがポジション1で、Skがポジション5の場合、図11のテーブルによると、開始時間「0000000000」から終了時間「00000000500」までの手本映像を再生することになる。

10

【0139】

なお、マーカSpaは図19に示すように停止位置Skが含まれるフレーズ（小節）の先頭に付与されるので、結局、ステップS1005で再生される手本映像は、演奏を停止した位置を含むフレーズの最初から停止した位置までの手本映像である。つまり、学習者が演奏を停止したとき直前に音高誤りがあった場合には、そのフレーズの最初に戻ってそのフレーズの手本映像を再生する。したがって、この場合の手本映像を「戻り手本」と呼ぶことができる。

【0140】

20

また、ステップS1001で“NO”の判断をしたとき、コンピュータ20は、続くステップS1007において、図19に示すように、停止位置Skより先でかつその停止位置Skに一番近いマーカSp（推奨戻り位置）をSpbとする。

【0141】

そして、次のステップS1009において、コンピュータ20は、楽譜および映像データベース22aから読み出して表示器24の楽譜表示領域24aに表示している楽譜上の五線譜の、停止位置の次の位置Sk+1にマークを付加的に表示する。ただし、この場合は、演奏誤りがあって停止したのではないので、このマークは、たとえば青色マークとする。誤り停止の場合には上述のように赤色マークを付与するので、それと異なる青色マークを表示することによって、学習者に、停止したときに誤りがあったのかどうかを容易に知らせる（感知させる）ことができる。

30

【0142】

その後、ステップS1011では、コンピュータ20は、停止位置の次の位置Sk+1からマーカSpb-1の位置までの手本映像を表示器24の映像表示領域24bに表示する。このとき、手本映像は楽譜および映像データベース22aから読み出すが、どの映像を再生するかは、上記と同様にして、図11のような連動テーブルのポジション毎の映像開始時間および終了時間を参照することによって選択することができる。

【0143】

なお、マーカSpbは図19に示すように停止位置Skが含まれるフレーズ（小節）の次のフレーズの先頭に付与されるので、結局、ステップS1011で再生される手本映像は、演奏を停止した位置からそのフレーズの最後までの手本映像である。つまり、学習者が演奏を停止したとき直前に音高誤りがなかった場合には、その停止位置から先のフレーズの手本映像を再生する。したがって、この場合の手本映像を「先読み手本」と呼ぶことができる。

40

【0144】

この実施例のように、学習者が演奏を停止したときだけ、つまり支援が必要なときだけ手本映像を表示するようにしたので、学習者の読譜に対する意欲を減じることがない。たとえば、学習者に次の演奏位置を、鍵盤の各キーに設けた発光ダイオードを点灯することによって、逐一教えるような過保護な支援を行うと、学習者が楽譜上の音符を読む必要がないので、読譜の意欲が失われるが、演奏停止したときだけ支援するこの実施例によれば

50

、学習者はできるだけ読譜によって次に演奏する位置を理解しようとするのが期待できる。

【0145】

また、複数の音符単位（時系列に連続する音符列）で読譜の訓練をすれば、個々の音符（次の演奏位置）ごとの読譜に比べて効果的な学習成果が得られる。後者の場合には、前後の音符との高低差と鍵盤の位置を捉える学習がされにくい。

【0146】

さらに、音楽的な表現力を習得するためにはフレーズ（文のようにまとまりのある音符列）をフレーズらしく演奏する必要があるため、初期段階からフレーズを意識させることは重要である。そこで、上述の実施例では手本映像をフレーズ単位で表示するようにして、学習者にフレーズを意識させるようにした。また、フレーズ単位で手本映像を表示することにより、1音単位での手本映像の表示と比較して、学習者がリズムを分かりやすいという利点もある。

【0147】

その後、学習者（ユーザ）は上述の手本映像を参考にして演奏学習を続行する訳であるが、学習者が演奏を再開することによって開始されるイベントが図20に示される。

【0148】

プリモ（学習者）が演奏を再開したとき、コンピュータ20は、最初のステップS1101で間違い位置レジスタShを初期化するとともに、次のステップS1103において、表示器24に手本映像が表示されているかどうか判断する。手本映像が表示されていないときには、コンピュータ20は何もしない（ステップS1005）が、表示されている場合には、ステップS1107で、先のマーカSpaまたはSpbの位置を記憶しておくためのレジスタ（マーカレジスタ：図示せず）をクリア（初期化）するとともに、次のステップS1109で手本映像に代えて、カメラ26が撮影した手元映像を映像表示領域24bに表示させる。したがって、プリモが演奏を継続する限り、装置のプリモの手元映像が表示される。

【0149】

さらに、上述の実施例では手本映像はたとえば学習者の前に設けた表示器24の映像表示領域24bに表示した。したがって、学習者は楽譜表示領域24aの楽譜と、映像表示領域24bの手本映像と、自分の手元（インタフェース12の鍵盤）とを見なければならない。上述のような2つの領域の縦配列によって学習者の視線移動を最小にする工夫はしているものの、それでも視線移動は学習者の負担である。そこで、手本映像を学習者の手元、すなわちインタフェース12の鍵盤上に投影表示するようにすれば、学習者が見なければならないのは手元の鍵盤と楽譜との2箇所済む。手本映像を鍵盤上に投影表示するためには、インタフェース12の上方にプロジェクタ（フロントプロジェクタ）を設置し、そこからインタフェース12の鍵盤上に投影するようにすればよい。ただし、その場合には、手本映像の手指の大きさや位置をインタフェース12の鍵盤の大きさや位置に確実に適合させる必要があるため、プロジェクタの拡大率や位置を最適に設定するように調整するものとする。

【図面の簡単な説明】

【0150】

【図1】図1はこの発明の背景となる合奏支援システムを示す概略ブロック図である。

【図2】図2は楽譜の記述形式の一例を示す図解図である。

【図3】図3は楽譜の一例を示す図解図である。

【図4】図4は図3の楽譜を記述した記述形式を示す図解図である。

【図5】図5は図3の楽譜に従った楽譜データの一例を示す図解図である。

【図6】図6は図1実施例における演奏動作を全体的に示すフロー図である。

【図7】図7は、図6実施例における演奏位置判定処理の動作を示すフロー図である。

【図8】図8は、図6実施例における音高データ取得処理の動作を示すフロー図である。

【図9】図9は、図1実施例における音高データ差替え処理の動作を示すフロー図である

10

20

30

40

50

。【図 1 0】図 1 0 はこの発明の一実施例の合奏支援システムを示す概略ブロック図である。

。【図 1 1】図 1 1 は図 1 0 実施例における楽譜および映像データベースに格納される演奏ポジションと映像タイムコードとの関係を図解する図解図である。

【図 1 2】図 1 1 は図 1 0 実施例において演奏間違い位置を検出するための動作を示すフロー図である。

【図 1 3】図 1 3 は図 1 0 実施例においてマーク付電子楽譜を表示するための動作を示すフロー図である。

【図 1 4】図 1 4 は図 1 3 によってタッチパネル表示器に表示される電子楽譜の一例を示す図解図である。 10

【図 1 5】図 1 5 は図 1 0 実施例において弾き直し演奏位置を追跡するための動作を示すフロー図である。

【図 1 6】図 1 6 は図 1 0 実施例においてプリモが弾き直しをしないときの動作を示すフロー図である。

【図 1 7】図 1 7 はこの発明の他の実施例の合奏支援システムを示す概略ブロック図である。

【図 1 8】図 1 8 は図 1 7 実施例において演奏間違い位置を検出したとき処理されるイベントを示すフロー図である。

【図 1 9】図 1 9 は図 1 7 実施例における停止位置およびマーカを図解する図解図である 20

。【図 2 0】図 2 0 は図 1 7 実施例において演奏を再開したときに処理されるイベントを示すフロー図である。

【符号の説明】

【 0 1 5 1 】

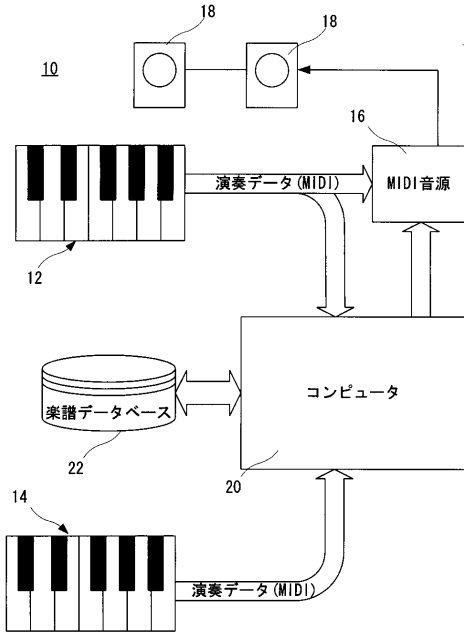
- 1 0 ...合奏支援システム
- 1 2 ...プリモ用入力インタフェース
- 1 4 ...セコンド用入力インタフェース
- 1 6 ...M I D I 音源
- 1 8 ...スピーカ
- 2 0 ...コンピュータ
- 2 2 ...楽譜データベース
- 2 2 a ...楽譜および映像データベース
- 2 4 ...タッチパネル表示器(表示器)
- 2 4 a ...楽譜表示領域
- 2 4 b ...映像表示領域
- 2 6 ...カメラ

10

20

30

【図1】



【図2】

* [BPRS]
 プリモ 音名 : セCOND音名 : コメント
 プリモ音名 : セCOND音名 : コメント

 * [BPRS]
 プリモ音名 : セCOND音名 : コメント

【図5】

ポジション	プリモ	セCOND	リターン
1	G5, E5	C5	*
2	A5		
3	G5	G4	
4	E5		
5	G5, D5	B4	*
6	A5		
7	G5	G4	*
8	E5		
9	C5	A4	*
10	D5		
11	E5		
12	E5	C5	
13	D5	B4	*
14		A4	
15		G4	
16		F4	
17	E5	C4	*

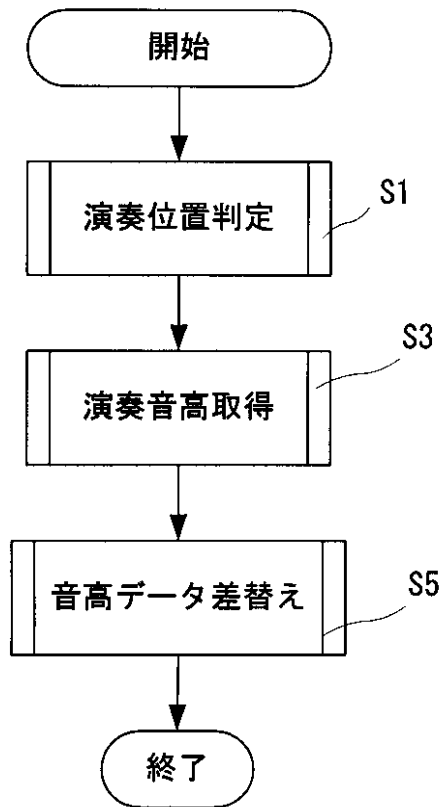
【図3】



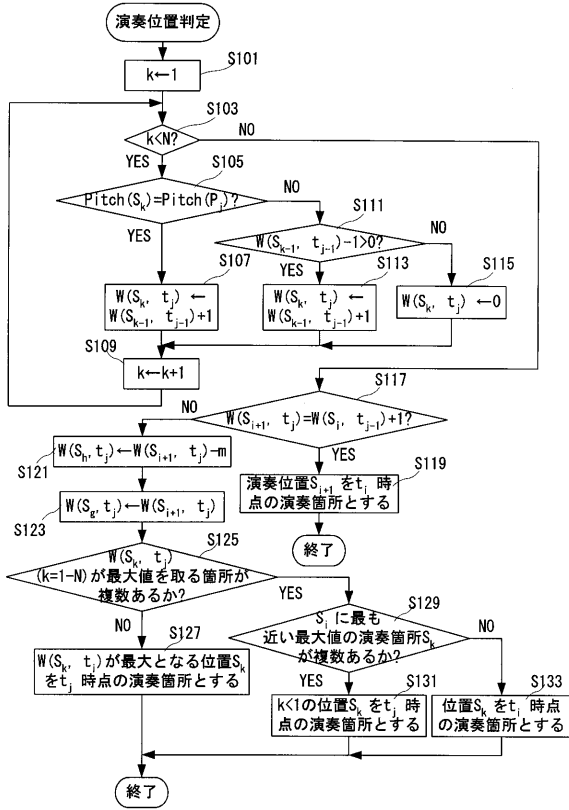
【図4】

* BPR
 G5, E5: C5: 最初の音. プリモは2つの音を弾いている.
 A5: : セCONDは新たに演奏すべき音がない
 G5: G4:
 E5: :
 * B
 G5, D5: B4: 2小節目. プリモは2つの音を弾いている.
 A5: :
 * S
 G5: G4: このプリモの演奏者はここに戻り易い癖があるので、
 直前に * Sを埋め込んでいる。
 E5: :
 * BP
 C5: A4: 3小節目。
 D5: :
 E5: :
 E5: C5:
 * B
 D5: B4:
 : A4: セCONDのみ演奏音がある
 : G4:
 : F4:

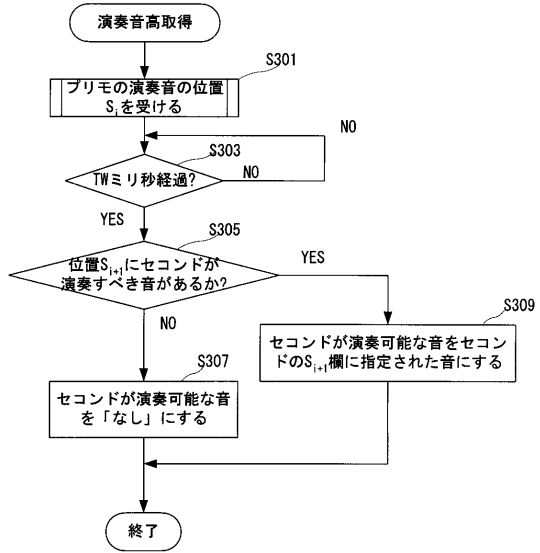
【図6】



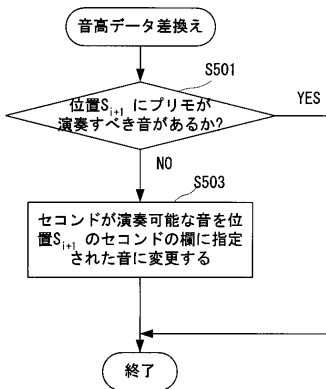
【図7】



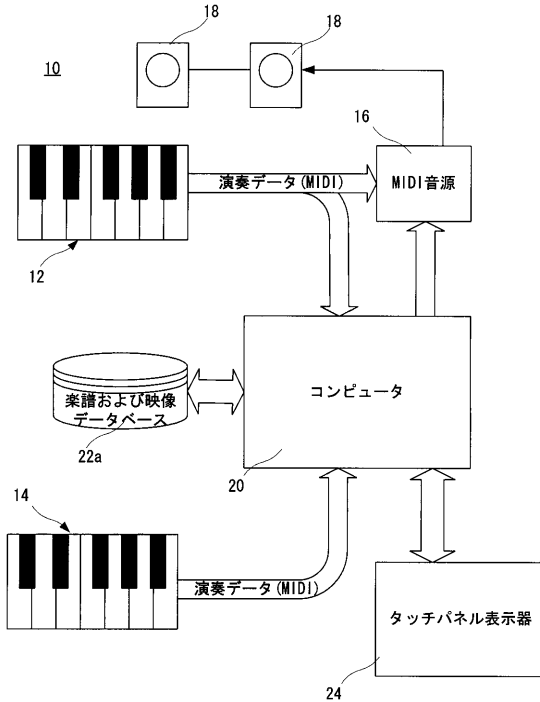
【図8】



【図9】



【図10】

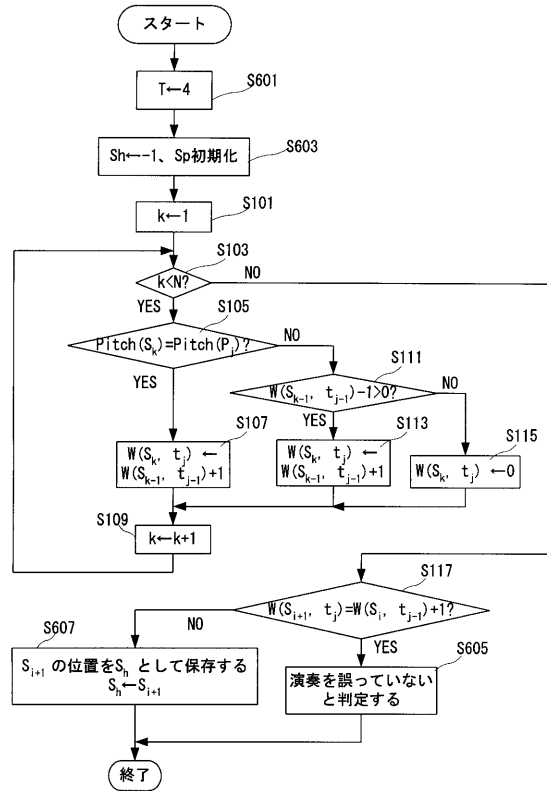


【図11】

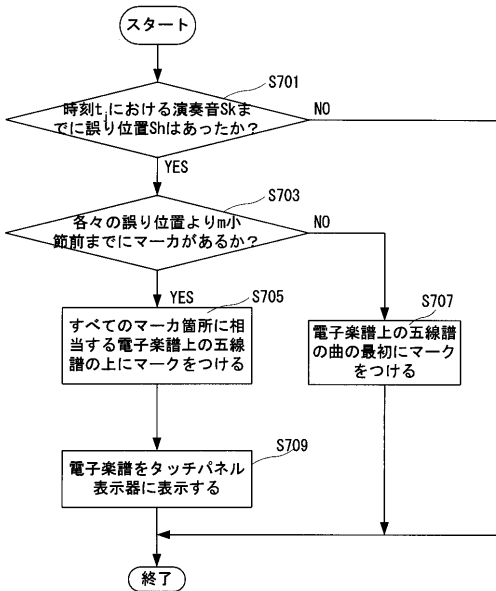
楽譜および映像データベース

ポジション	開始時間	終了時間
1	000000000000	000000000100
2	000000000101	000000000200
3	000000000201	000000000300
4	000000000301	000000000400
5	000000000401	000000000500
6	000000000501	000000000600
7	000000000601	000000000700
8	000000000701	000000000800
9	000000000801	000000000900
10	000000000901	000000001000
11	000000001001	000000001100
12	000000001101	000000001200
13	000000001201	000000001400
14
15
16
17

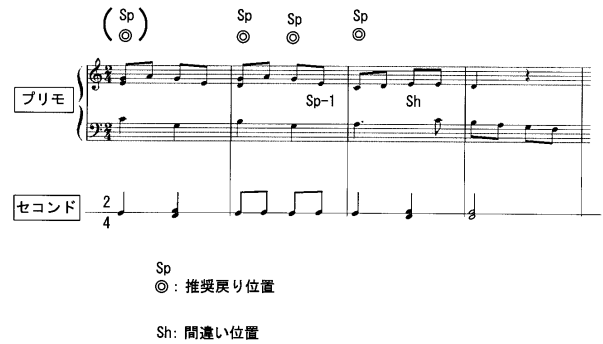
【図12】



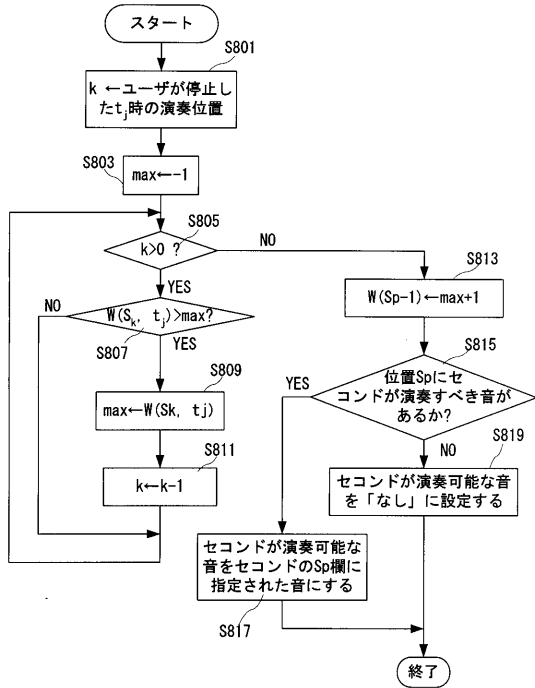
【図13】



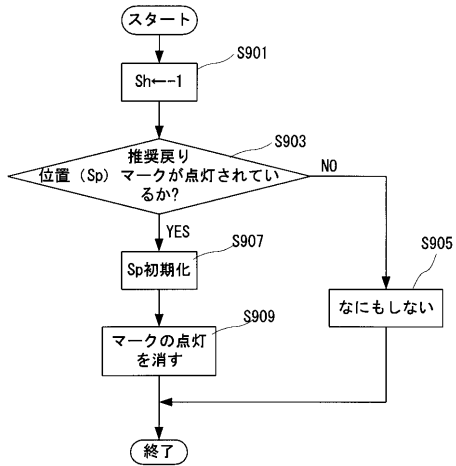
【図14】



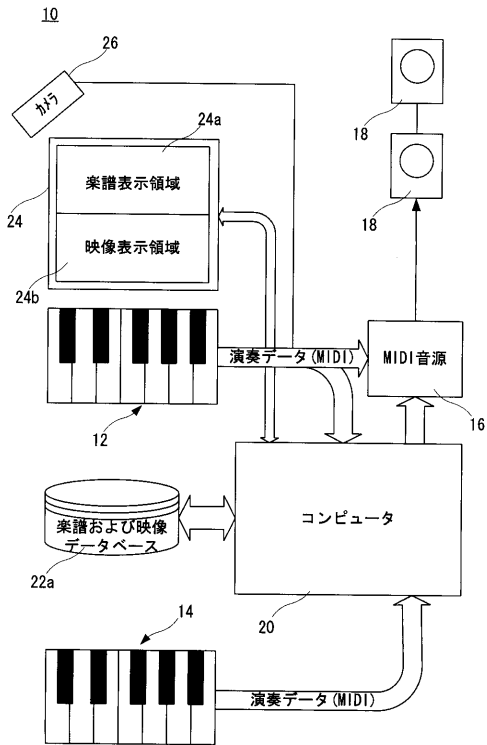
【図15】



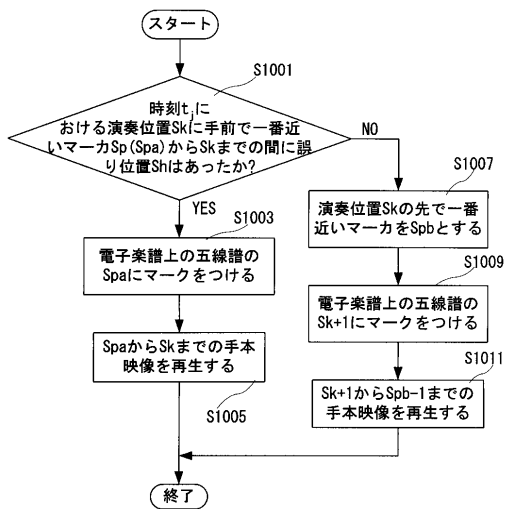
【図16】



【図17】



【図18】



【図19】

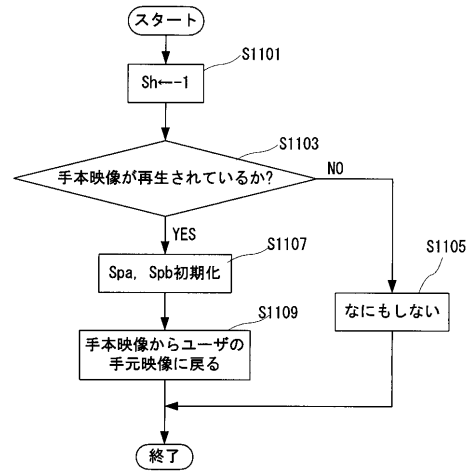
(A) 演奏誤りがあった場合



(B) 演奏誤りがなかった場合



【図20】



フロントページの続き

(72)発明者 苗村 昌秀

京都府相楽郡精華町光台二丁目2番地2 株式会社国際電気通信基礎技術研究所内

審査官 植野 孝郎

(56)参考文献 特開2005-250053(JP,A)

特開2002-323849(JP,A)

特開2002-91291(JP,A)

(58)調査した分野(Int.Cl., DB名)

G09B15/00-15/08

G10H 1/00

G10G 1/00-1/04