

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4368222号  
(P4368222)

(45) 発行日 平成21年11月18日(2009.11.18)

(24) 登録日 平成21年9月4日(2009.9.4)

(51) Int.Cl.		F I		
<b>G 1 0 H</b>	<b>1/00</b>	<b>(2006.01)</b>	<b>G 1 0 H</b>	<b>1/00 1 0 2 Z</b>
<b>G 0 9 B</b>	<b>15/04</b>	<b>(2006.01)</b>	<b>G 0 9 B</b>	<b>15/04</b>
<b>G 1 0 G</b>	<b>1/00</b>	<b>(2006.01)</b>	<b>G 1 0 G</b>	<b>1/00</b>

請求項の数 7 (全 17 頁)

<p>(21) 出願番号 特願2004-59540 (P2004-59540)</p> <p>(22) 出願日 平成16年3月3日(2004.3.3)</p> <p>(65) 公開番号 特開2005-250053 (P2005-250053A)</p> <p>(43) 公開日 平成17年9月15日(2005.9.15)</p> <p>審査請求日 平成18年9月7日(2006.9.7)</p> <p>(出願人による申告) 国等の委託研究の成果に係る特許出願(平成15年度通信・放送機構、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条の適用を受けるもの)</p> <p>特許権者において、実施許諾の用意がある。</p>	<p>(73) 特許権者 393031586 株式会社国際電気通信基礎技術研究所 京都府相楽郡精華町光台二丁目2番地2</p> <p>(74) 代理人 100090181 弁理士 山田 義人</p> <p>(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</p> <p>(72) 発明者 大島 千佳 石川県能美郡辰口町旭台1-1</p> <p>審査官 井出 和水</p>
---	---

最終頁に続く

(54) 【発明の名称】 合奏支援システム

(57) 【特許請求の範囲】

【請求項1】

第1奏者によって操作され、演奏に応じて第1演奏データを出力する第1入力インタフェース、

第2奏者によって操作され、演奏に応じて第2演奏データを出力する第2入力インタフェース、

演奏位置と、各演奏位置毎の前記第1奏者が演奏すべき音の音高データと、各演奏位置毎の前記第2奏者が演奏すべき音の音高データとを含む楽譜データを格納する楽譜データベース、

前記第1演奏データと前記楽譜データとに基づいて前記第1奏者の演奏位置を判定する演奏位置判定手段、

前記楽譜データを参照して前記演奏位置判定手段によって判定された前記第1奏者の演奏位置における前記第2奏者が演奏すべき音の音高データを取得する音高データ取得手段、

前記第2演奏データの内の音高データだけを前記音高データ取得手段が取得した音高データに差し替える音高データ差替え手段、および

前記第1演奏データと、前記音高データ差替え手段によって音高データだけが差し替えられた前記第2演奏データとに応じて発音する発音手段を備える、合奏支援システム。

【請求項2】

前記演奏位置判定手段は、

前記第1演奏データの音高と前記楽譜データの音高とを比較することによって前記楽譜データ上の前記第1奏者が演奏すべきすべての音の現時点での重みを評価する重み評価手段、

1つ前の時点の演奏位置のその時点での重みと当該1つ前の時点の演奏位置の前記楽譜データ上の次の演奏位置の現時点での重みとが所定の条件を充足するか否かを判断する充足判断手段、および

前記充足判断手段が前記所定の条件を充足すると判断した場合に、前記楽譜データ上の前記次の演奏位置を現時点の演奏位置と決定する第1決定手段を含む、請求項1記載の合奏支援システム。

【請求項3】

前記重み評価手段は、

前記第1演奏データの現時点の音高と前記楽譜データの前記第1奏者が演奏すべきすべての演奏位置の音の音高とを比較する比較手段、および

前記比較手段による比較の結果両者が一致する場合に、その音高が一致する前記楽譜データ上の演奏位置の現時点での重みを1つ前の演奏位置の1つ前の時点の重みに所定値を加えたものとし、前記比較手段による比較の結果両者が一致しない場合に、その音高が一致しない前記楽譜データ上の演奏位置の現時点での重みを1つ前の演奏位置の1つ前の時点の重みから所定値を引いたものとする重み算出手段を含み、

前記充足判断手段は、1つ前の時点の演奏位置のその時点での重みに所定値を加えたものが、当該1つ前の時点の演奏位置の前記楽譜データ上の次の演奏位置の現時点の重みに等しいという条件を充足するか否かを判断する、請求項2記載の合奏支援システム。

【請求項4】

前記楽譜データは、弾き直しが発生し易い演奏位置を指定演奏位置として設定しておき、

前記演奏位置判定手段は、

前記充足判断手段が前記条件を充足しないと判断したとき、最初から現在の演奏位置までのすべての前記指定演奏位置の重みの値を、1つ前の時点の演奏位置の前記楽譜データ上の次の演奏位置の現時点の重みの値より所定値だけ小さく設定する第1重み設定手段

1つ前の時点の演奏位置を基点に前記楽譜データ上で所定値だけ前の演奏位置から所定値だけ後までの演奏位置の現時点での重みを、1つ前の時点の演奏位置の前記楽譜データ上の次の演奏位置の現時点の重みと同じ値に設定する第2重み設定手段、および

最大の重みを有する演奏位置を現時点の演奏位置と決定する第2決定手段を含む、請求項2または3記載の合奏支援システム。

【請求項5】

前記演奏位置判定手段は、最大の重みを有する演奏位置が複数存在するか否かを判断する第1存在判断手段を含み、

前記第2決定手段は、前記第1存在判断手段が、最大の重みを有する演奏位置が複数存在しないと判断したとき、前記最大の重みを有する演奏位置を現時点の演奏位置として決定する、請求項4記載の合奏支援システム。

【請求項6】

前記演奏位置判定手段は、前記第1存在判断手段が、最大の重みを有する演奏位置が複数存在すると判断したとき、最大の重みを有し前記楽譜データ上で1つ前の時点の演奏位置に最も近い演奏位置が複数存在するか否かを判断する第2存在判断手段を含み、

前記第2決定手段は、前記第2存在判断手段が、最大の重みを有し前記楽譜データ上で1つ前の演奏位置に最も近い演奏位置が複数存在すると判断したとき、最大の重みを有し前記楽譜データ上で1つ前の時点の演奏位置に最も近い演奏位置のうち、1つ前の時点の演奏位置より前である演奏位置を現時点の演奏位置として決定する、請求項5記載の合奏支援システム。

【請求項7】

10

20

30

40

50

前記演奏位置判定手段は、前記第 1 存在判断手段が、最大の重みを有する演奏位置が複数存在すると判断したとき、最大の重みを有し前記楽譜データ上で 1 つ前の時点の演奏位置に最も近い演奏位置が複数存在するか否かを判断する第 2 存在判断手段を含み、

前記第 2 決定手段は、前記第 2 存在判断手段が、最大の重みを有し前記楽譜データ上で 1 つ前の時点の演奏位置に最も近い演奏位置が複数存在しないと判断したとき、最大の重みを有し前記楽譜データ上で 1 つ前の時点の演奏位置に最も近い演奏位置を現時点の演奏位置として決定する、請求項 5 記載の合奏支援システム。

【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。

【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。

【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・ワンを利用しても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。

【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。

【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】

10

20

30

40

50

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 Live Performance, 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

10

【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】

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

20

【0011】

この発明の他の目的は、たとえば楽器演奏の初心者の親と初級者の子供とのようなペアでも、すぐに合奏できるようにする、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】

【0012】

請求項1の発明は、第1奏者によって操作され、演奏に応じて第1演奏データを出力する第1入力インタフェース、第2奏者によって操作され、演奏に応じて第2演奏データを出力する第2入力インタフェース、演奏位置と、各演奏位置毎の第1奏者が演奏すべき音の音高データと、各演奏位置毎の第2奏者が演奏すべき音の音高データとを含む楽譜データを格納する楽譜データベース、第1演奏データと楽譜データとに基づいて第1奏者の演奏位置を判定する演奏位置判定手段、楽譜データを参照して演奏位置判定手段によって判定された第1奏者の演奏位置における第2奏者が演奏すべき音の音高データを取得する音高データ取得手段、第2演奏データの内の音高データだけを音高データ取得手段が取得した音高データに差し替える音高データ差替え手段、および第1演奏データと、音高データ差替え手段によって音高データだけが差し替えられた第2演奏データとに応じて発音する発音手段を備える、合奏支援システムである。

30

【0013】

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。)が第1奏者(プリモ)によって操作され、演奏に応じて第1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(14)が第2奏者(セコンド)によって操作され、演奏に応じて第2演奏データとしてMIDIデータを出力する。第1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は演奏位置判定手段にも与えられる。

40

【0014】

演奏位置判定手段(20, S1, S101-S133)では、第1入力インタフェースからの第1演奏データ(MIDIデータ)と、楽譜データベース(22)の楽譜データとに基づいて、第1奏者の楽譜上での演奏位置を判定する。

【0015】

音高データ取得手段(20, S3, S301-S311)は、このようにして判定された第1奏者の演奏位置に基づいて楽譜データから、第2奏者が演奏すべき音の音高データ

50

を取得する。そして、音高データ差替え手段(20, S5, S501 - S503)は、第2入力インタフェースから入力される第2演奏データの内の音高データだけを音高データ取得手段が取得した音高データに差し替えて、発音手段に与える。したがって、発音手段は、第1奏者が演奏した第1演奏データと、第2奏者が演奏したものであるが音高データのみが差し替えられている第2演奏データとに従って、音楽を出力する。

【発明の効果】

【0020】

この発明によれば、第1奏者の演奏位置における第2奏者の演奏データの音高だけを差し替えるようにしているため、第2奏者は容易に正しい演奏をすることができる。したがって、たとえば、第1奏者が初級者であってかつ第2奏者が初心者であっても、合奏や連弾が簡単に行える。

10

【0021】

また、弾き直しへの対応を可能にすれば、初級者に多く見られる演奏途中での弾き直しが生じた場合でも、迅速に正しい演奏位置を判定できる。

【0022】

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ろう。

【発明を実施するための最良の形態】

【0023】

図1に示すこの実施例の合奏支援システム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、同一インタフェース(鍵盤)を2つの領域に分けて使用することになる。また、この実施例の入力インタフェースとしては鍵盤(キーボード)が用いられるが、他の楽器のインタフェースが用いられてもよい。

20

【0024】

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。

30

【0025】

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。

40

【0026】

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。

【0027】

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。

【0028】

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。

50

## 【0029】

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

## 【0030】

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。

## 【0031】

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。

## 【0032】

【表1】

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

## 【0033】

「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。

## 【0034】

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。

## 【 0 0 3 5 】

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 B に 37 音(延べ 1 1 1 音)見られた。「脱落」は被験者 A に 3 音見られた。

## 【 0 0 3 6 】

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異なるため、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。

## 【 0 0 3 7 】

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。

## 【 0 0 3 8 】

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らず、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。

## 【 0 0 3 9 】

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「C 5」、「A 3」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。

## 【 0 0 4 0 】

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。

## 【 0 0 4 1 】

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。

## 【 0 0 4 2 】

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。

10

20

30

40

50

## 【 0 0 4 3 】

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、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。リターン欄には、楽譜データでいずれかの種類のマーカー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。

## 【 0 0 4 4 】

このような図 1 実施例において、コンピュータ 20 は、図 6 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、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、この結果を図 7 に示す演奏音高取得処理に通知する。

## 【 0 0 4 5 】

演奏音高取得処理では、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、これを図 8 に示す音高データ差し替え処理に渡す。

## 【 0 0 4 6 】

音高データ差し替え処理では、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）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。

## 【 0 0 4 7 】

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。

## 【 0 0 4 8 】

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が出力される。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。

## 【 0 0 4 9 】

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。

## 【 0 0 5 0 】

ここで、図 6 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 6 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。

## 【 0 0 5 1 】

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

10

20

30

40

50



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。

【0052】

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。

【0053】

そこで、この実施例では、先の非特許文献1でDannenberが提案したDP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考案した。

【0054】

この実施例の演奏位置検出方法では、Dannenber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。

【0055】

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。

【0056】

(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。

【0057】

【数1】

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

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

(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。

【0058】

このことを図6で示すと、ステップS101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S103で“NO”となり、次のステップS105で、コンピュータ20(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S105で“YES”なら、コンピュータ20は、次のステップS107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S109で音数kをインクリメントして先のステップS103に戻る。ただし、この実施例では、ステップS107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。

【0059】

ステップS105で“NO”なら、コンピュータ20は、ステップS111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S113で時点 $t_j$ での音符 $S_k$ の重み $W(S_k, t_j)$ として $W(S_{k-1}, t_{j-1})$

10

20

30

40

50

) - 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。

【0060】

このようなステップS 1 0 1 - S 1 1 5をステップS 1 0 3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。

【0061】

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S 1 0 3で“YES”となり、コンピュータ20は、続くステップS 1 1 7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S 1 1 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、他の適宜の所定値を加算するようにしてもよい。

【0062】

ここで、このステップS 1 1 7での条件判定について詳しく説明する。このステップS 1 1 7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6のS 1 0 7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。

【0063】

そして、このステップS 1 1 7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6の演奏位置判定ルーチンを終了する。

【0064】

(2)もし、 $W(S_{i+1}, t_j) < W(S_i, t_{j-1}) + 1$ であれば、ステップS 1 2 1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S 1 1 7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。

【0065】

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。

【0066】

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、 $t_j$

10

20

30

40

50

$t_j$  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。

【0067】

(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。 10

(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。

【0068】

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

【0069】

(ii)  $S_i$  に最も近い最大値を取る箇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。

【0070】

このことを図6を参照して説明すると、図6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。 20

【0071】

そして、ステップS125において、重み  $W(S_k, t_j)$  ( $k = 1 \sim N$ ) が最大値をとる箇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。 30

【0072】

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  $W(S_k, t_j)$  が最大値を取る箇所が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箇所  $S_k$  が複数存在かするかどうか、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、 $k < i$  の演奏箇所  $S_k$  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。 40

【0073】

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。

【0074】

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。 50

## 【 0 0 7 5 】

次に、図 7 を参照して、演奏音高取得処理を説明する。

## 【 0 0 7 6 】

演奏音高取得処理も、図 6 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。したがって、図 7 の最初のステップ S 3 0 1 で、コンピュータ 2 0 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 S 3 0 3 における一定時間  $T_w$  ミリ秒の経過の後、ステップ S 3 0 5 で、コンピュータ 2 0 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 S 3 0 7 において、コンピュータ 2 0 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 S 3 0 5 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 S 3 0 9 において、コンピュータ 2 0 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。

10

## 【 0 0 7 7 】

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 S 3 0 5 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。

20

## 【 0 0 7 8 】

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 S 3 0 5 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。

30

## 【 0 0 7 9 】

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 7 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拘らず、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。

40

## 【 0 0 8 0 】

一方、セコンドが誤って本来演奏すべきでない箇所で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

50

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。

#### 【0081】

10 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。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。

#### 【0082】

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  
30 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。

#### 【0083】

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  
40 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  
50

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。

【 0 0 8 4 】

続いて、図 8 に示す音高データ差替え処理について詳しく説明する。この図 8 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 S 5 0 1 で、コンピュータ 2 0 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。“ Y E S ” ならそのままこのルーチンを抜けるが、“ N O ” なら、次のステップ S 5 0 3 において、コンピュータ 2 0 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。

【 0 0 8 5 】

具体的に説明すると、図 8 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 1 4 ~ 1 6 の場所がこの場合にあたる。プリモがポジション 1 3 の D 5 を演奏して T w ミリ秒経過すると、セコンドの演奏可能音はポジション 1 4 の A 4 になっている。この状態でセコンドが打鍵するとこの A 4 の音が発音されると同時に、この図 8 のアルゴリズムが動作する。

【 0 0 8 6 】

プリモ側の演奏位置  $S_i$  は 1 3 のままなので、ポジション  $S_i + 1 (= 1 4)$  のプリモの楽譜データを調べ、プリモがポジション 1 4 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 1 4 で演奏するデータがないので、条件判定は“ N O ” となる。この場合、現在の演奏位置を  $S_i = S_i + 1 (= 1 4)$  とし、セコンドの演奏可能音を  $S_i + 1 (= 1 5)$  の G 4 に設定する。この状態でセコンドが再度打鍵すると、G 4 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。

【 0 0 8 7 】

なお、セコンドがまだポジション 1 6 まで演奏し終わっていない状態でプリモがポジション 1 7 の音 E 5 を演奏した場合、演奏位置  $S_i$  は一気にポジション 1 7 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、プリモが間違っただけで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。

【 0 0 8 8 】

ところで、セコンドの打鍵があれば無条件にこの図 8 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 8 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“ Y E S ” 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。

【 0 0 8 9 】

以上のような、図 7 および図 8 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。

【 0 0 9 0 】

発明者等が何組かのペアに頼んで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よって、すぐに合奏（連弾）が可能になった。

【 図面の簡単な説明 】

10

20

30

40

50

【 0 0 9 1 】

【図 1】図 1 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 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位置判定処理の動作を示すフロー図である。

【図 7】図 7 は図 1 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。

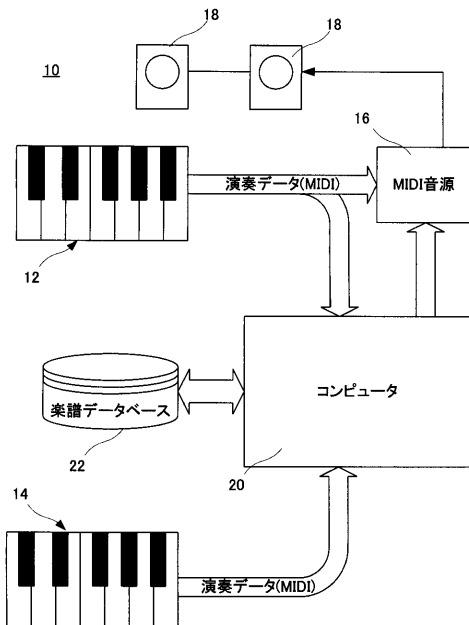
【図 8】図 8 は図 1 実施例における音高データ差替え処理の動作を示すフロー図である。

【符号の説明】

【 0 0 9 2 】

- 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COND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

【 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 3 】



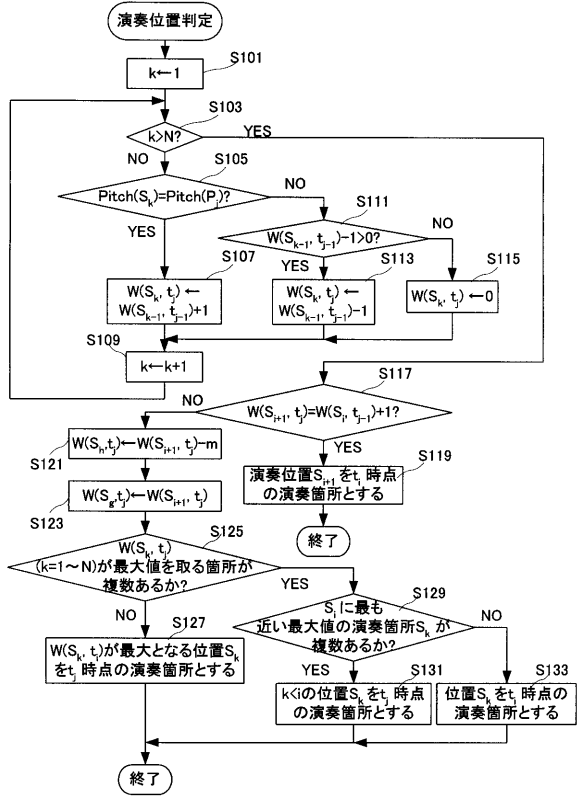
【 図 4 】

\* 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:

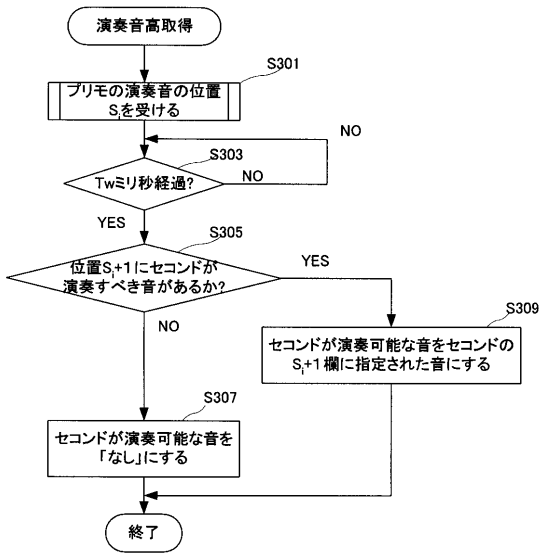
【図5】

ポジション	プリモ	セコンド	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	*

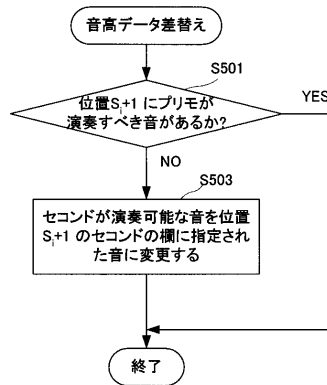
【図6】



【図7】



【図8】





---

フロントページの続き

- (56)参考文献 特開平10-232676(JP,A)  
特開2000-231379(JP,A)  
特開2003-131663(JP,A)

(58)調査した分野(Int.Cl., DB名)

G10H	1/00	-	G10H	7/12
G10G	1/00	-	G10G	7/02
G09B	15/04			