

# 鍵盤楽器のための 実時間運指取得システムの構築

竹川 佳成 寺田 努 西尾 章治郎

鍵盤演奏において運指は演奏に影響する重要な要素であるが、同じ楽曲であっても演奏家の身体的特徴や音楽の表現方法によってさまざまな運指が考えられるため、楽曲から運指を一意に決定することは難しい。一方、実際に演奏を聴きながらリアルタイムで演奏者の運指情報を得られれば効率的な演奏学習や奏者と一体感のある音楽視聴が可能となる。そこで、本研究ではカメラベースのシンプルな画像処理と、鍵盤や運指特性から得られた演奏ルールを組み合わせることで、実時間で高精度に運指を取得するシステムを構築する。さらに、鍵盤習熟者による評価実験を行うことで、システムの有効性を検証する。

Fingering is one of important aspects in piano performances because it affects pianists' musicality. If audience, especially pianists, share the real-time fingering of the performer in a concert, they feel sense of togetherness and it helps them for learning professional piano performances. In response to these requirements, the goal of our study is to construct a real-time fingering detection system for pianists. Our system achieves the real-time fingering detection by integrating a simple camera-based image detection and musical rules. We have developed a prototype system, and evaluated its effectiveness by actual use of the system.

## 1 はじめに

演奏家は、楽曲演奏の際に望む音楽表現を実現するために、さまざまな技術を駆使する。その1つに運指(指使い)がある。運指の違いは音楽表現に大きく影響するため、演奏家は、弾きやすさや効率より作品の意図を考えた音楽的な指使いを選ぶ。意図する音楽表現は演奏家各々で異なるため、優れた運指は数多く存在する[4][6][8]。

一方、我々が得られる運指情報は、作曲家や演奏家、教育者によって書き入れられた楽譜や教則本など数少ないため、ピアノレッスンなど教育の現場では、指導者は、生徒の演奏を見ながら適した運指の指導を行っている。しかし、地理的に離れた教師と生徒が

レッスンを行う遠隔レッスン[5]では、質の高いテレビ会議システムの構築はコスト面で難しく、指導者は生徒の演奏状況を十分に確認できない。したがって、リアルタイムに演奏者の運指を取得する技術が求められている。リアルタイムの運指情報が取得できれば、遠隔レッスン以外にも、誤った運指を指摘するなど独習支援や、演奏と運指を同時に見ながらの効率的な演奏学習、奏者と一体感のある音楽視聴が可能となる。

そこで本研究では、鍵盤奏者を対象とした実時間運指取得システムの構築を目的とする。提案システムは演奏者の動作をできるだけ妨げずにカメラを用いて運指を取得する。また、鍵盤演奏の特徴を表す演奏ルールを適用することで認識精度を高めている。

以下、2章でシステムの設計について説明し、3章で実装について述べる。4章で評価と考察について説明し、5章で関連研究について述べ、最後に6章で本研究のまとめを行う。

## 2 運指取得システムの設計

本研究で構築する運指取得システムは、以下の方針

Construction of a Real-Time Fingering Detection System for a Clavier

Yoshinari TAKEGAWA, Tsutomu TERADA and Shojiro NISHIO, 大阪大学大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University

コンピュータソフトウェア, Vol.16, No.5 (1999), pp.78-83.

[論文] 1999年8月3日受付。

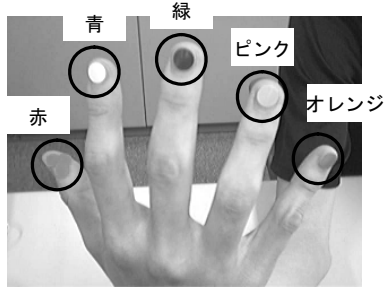


図 1 指先に付けたカラーマーカ

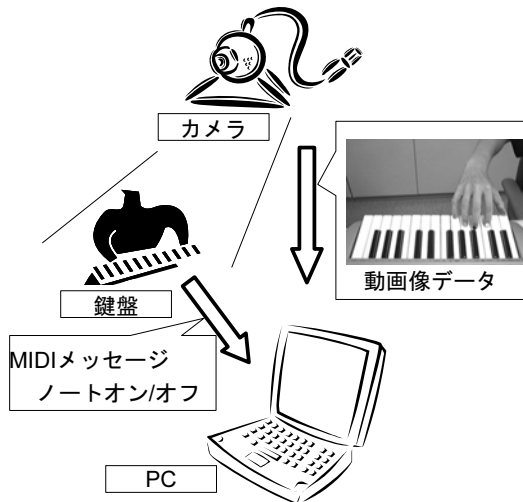


図 2 システム構成

をもとに設計した。

- (1) 演奏を妨げない運指取得
- (2) 演奏ルールを用いた補正処理

(1) に関しては、コンサートやレッスンなど演奏者が演奏に集中していても利用できるようにできる限り演奏を妨げない、つまり指に余計なデバイスをつけずにシステムを実現することを目指した。指先に取り付けるボタンや指の関節に取り付けるセンサ、データグローブ<sup>[10]</sup>といった補助器具を用いると運指は取得しやすいが、指の動きを束縛してしまう。提案システムではカメラを用い、各指の爪に貼り付けたカラーマーカ(図1)の画像処理から運指取得を行う。カラーマーカは薄いシールであり、装着に違和感が少ない。

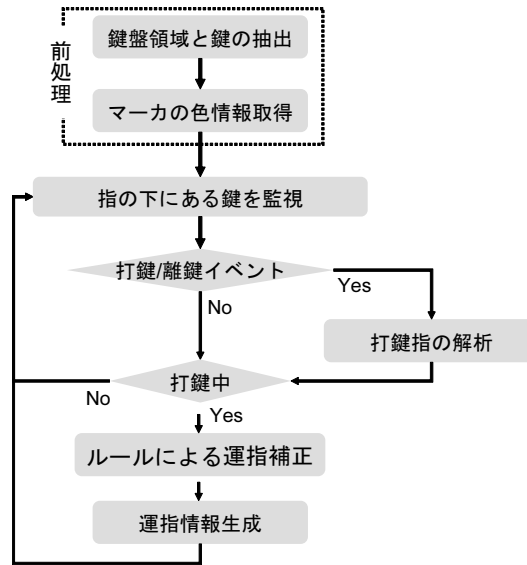


図 3 運指解析処理のフローチャート

(2) に関しては、演奏や鍵盤、手の特性をもとに定義したルールにより運指を補正する。提案システムは、実時間処理を実現するため、画像処理をシンプルにしておき、正確な運指の取得は困難である。また、鍵盤演奏時には、指が複雑に重なりあうため、画像処理だけで正確に指の位置を特定することは難しい。そこで、本システムでは目的を鍵盤演奏に特化し、MIDI鍵盤が生成する打鍵イベント(ノートオン)や離鍵イベント(ノートオフ)の発生と画像処理を連携させることで打鍵位置を特定する。また、鍵盤の運指特性や特徴的なパターンをルールとしてもつことで運指情報の修正を行う。

システムの構成を図2に示す。カメラから鍵盤とマーカを写した動画画像データと鍵盤から得られるMIDIデータをもとにPC上で運指を解析、生成する。

以下、PC上での運指解析処理について説明する。

### 2.1 運指解析処理の概要

運指解析処理の流れを図3に示す。カメラや鍵盤は固定されており、前処理としてカメラ画像から黒鍵の位置関係を解析し、鍵盤領域の抽出および鍵盤領域の各ピクセルと鍵番号との対応をさせておく。以降の処理は、鍵盤領域に対してだけ行われる。また、カラー

表 1 運指ルール

ルール番号	項目
1	親指以外の指の交差は生じない。
2	1 度打鍵した鍵は、離鍵するまで同じ指を使う。
3	同じ高さの音は、同じ運指を使う。
4	音列の構造が類似しているならば、同じ運指列になる。

マーカの色情報もあらかじめ取得しておく。

打鍵していない指の状態は、打鍵中の指を他の指の状態から絞れるなど運指特定において重要であるため、システムは常にカラーマーカを監視し、マーカの下に存在する鍵を検出している。鍵盤から打鍵(離鍵) イベントがあったときは打鍵(離鍵) した指を解析する。また、ルールによる運指補正は、何も打鍵されていないとき以外常に行われる。

## 2.2 カラーマーカの抽出

カラーマーカの色抽出は、カメラ画像から得られた RGB 値を輝度変化に強い HSV 値に変換し、各 HSV 値に範囲を設けてその範囲に入っている色を抽出する方法で行う。各 HSV 値の範囲を広くするとマーカ認識率は向上するが、他のマーカの HSV 値と認識されてしまうため、適切な値を設定する必要がある。

## 2.3 ルールによる運指補正

前節で述べたように、本研究ではシンプルな画像処理を用いてマーカの位置検出を行っているため、マーカや運指の認識ミスが生じる。一般に、認識精度を上げるためには特別な画像変換や複雑なパターン認識を用いるが、それらを適用すると計算コストが高くなり、リアルタイム処理が難しくなる。そこで、本研究では鍵盤演奏の運指取得という特徴を活かし、演奏や鍵盤、手の特性をもとに定義されたルールから、運指の修正や運指が認識できなかったときの予測を行う。

表 1 に本システムで適用した運指ルールを示す。以下、各ルールについて説明する。

### ルール 1

親指以外、人差指から小指にかけてはマーカの水平方向の位置関係が崩れることはない。ルール 1 は、マーカの未検出(マーカの存在そのものが検出され

ていない状態) やマーカの誤認識(検出したマーカを誤認識している状態) を検出および修正できる。なおルール 1 は、1 本の指で複数の鍵を打鍵しない、1 つの鍵を複数の指で打鍵しない、という条件下で正しく動作する。ルール 1 を含め以降のルールはいずれも、どのような状況でもうまくいく普遍的なものではない。したがって、各ルールはユーザが選択的に利用する。

ルール 1 の定義より、正しいマーカ順序の組合せは図 4 に示す 5 つの正解マーカ群に限定される。ここで、図 4 における正解マーカ群やシステムが認識したマーカの数字は、1 から順に親指のマーカから小指のマーカに対応し、「a」や「b」の添え字は、1 つしか存在しないはずのマーカが複数検出された場合、それらを区別するために付けた。さらに、1 つの指で複数鍵の同時打鍵は考えないため、マーカが認識できている場合、その下にある鍵も特定できているとする。

以下、ルール 1 によるマーカの修正、鍵盤から得られるノートオン情報から運指を特定する手続きについて説明する。

1. 認識マーカと正解マーカ群を入力とする演算子 MATCH は、認識マーカの要素と正解マーカの要素が並び順を保ったまま最も一致度が高くなるように割り当て、一致したマーカは「1」を、一致しなかったマーカは「0」をマーカ番号順に出力すると定義する。
2. MATCH 演算の結果の中で、「1」の個数もっとも多いものを正解候補とする。対応する正解マーカが運指候補となる。
3. 正解候補が複数あるときは AND 演算をとり 1 つにまとめる。結果において「1」と表されたマーカが誤認識されていない確定マーカとなる。
4. 確定マーカの下にある鍵が打鍵された場合、運指は一意に定まる。一方、確定していないマーカ

【例1 認識マーカ 1, 2, 3, 4, 3】

	正解マーカ群		1, 2, 3a, 3b, 4, 5	⇒	1, 2, 3a, 3b, 4, 5												
(1, 2, 3a, 4, 3b) MATCH	(1, 2, 3, 4, 5)	=	<table border="1"><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	1	1	1	0	1	0	⇒	<table border="1"><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	1	1	1	0	1	0
	1	1	1	0	1	0											
	1	1	1	0	1	0											
	(2, 1, 3, 4, 5)	=	<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	1	0	1	0								
	0	0	1	0	1	0											
(2, 3, 1, 4, 5)	=	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	1	0		確定マーカ → 1, 2, 3a, 4							
0	0	0	0	1	0												
(2, 3, 4, 1, 5)	=	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0									
0	0	0	0	0	0												
(2, 3, 4, 5, 1)	=	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0									
0	0	0	0	0	0												

【例2 認識マーカ 1, 1, 3, 4, 5】

			1a, 1b, 2, 3, 4, 5	AND	1a, 1b, 2, 3, 4, 5												
(1a, 1b, 3, 4, 5) MATCH	(1, 2, 3, 4, 5)	=	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	1	0	0	1	1	1	AND	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	0	0	0	1	1	1
	1	0	0	1	1	1											
	0	0	0	1	1	1											
	(2, 1, 3, 4, 5)	=	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table>	0	1	0	1	1	1								
	0	1	0	1	1	1											
(2, 3, 1, 4, 5)	=	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table>	0	0	0	0	1	1		確定マーカ → 3, 4, 5							
0	0	0	0	1	1												
(2, 3, 4, 1, 5)	=	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	0	0	1									
0	0	0	0	0	1												
(2, 3, 4, 5, 1)	=	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0									
0	0	0	0	0	0												

【例3 認識マーカ 2, 3】

			1, 2, 3, 4, 5		1, 2, 3, 4, 5									
(2, 3) MATCH	(1, 2, 3, 4, 5)	=	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0	0						
	0	1	1	0	0									
	(2, 1, 3, 4, 5)	=	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0	0						
	0	1	1	0	0									
	(2, 3, 1, 4, 5)	=	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0	0	AND	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0
0	1	1	0	0										
0	1	1	0	0										
(2, 3, 4, 1, 5)	=	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0	0		確定マーカ → 2, 3					
0	1	1	0	0										
(2, 3, 4, 5, 1)	=	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	1	0	0							
0	1	1	0	0										

図 4 認識したマーカの修正例

の下にある鍵が打鍵されたときや、打鍵された鍵上にマーカが検出できなかったときは、確定マーカと打鍵された鍵の位置関係をもとにマーカ番号を推測する。

以上のアルゴリズムを図 4 の例を用いて具体的に説明する。例 1 では、マーカ候補列が一意に決まり、MATCH 演算結果から認識マーカ 3b の誤認識が判明し、3b → 5 と修正される。また、例 2 では、すべてのマーカが確定しているわけではないが、マーカ 1a あるいはマーカ 1b の下の鍵が打鍵されたときには 1 の指か 2 の指で打鍵されていることがわかる。さらに、例 3 では未検出が多いが、マーカ 2 とマーカ 3 が確定しているため、マーカ 2 より低い鍵や、マーカ 2 とマーカ 3 の間の鍵が打鍵されたときにはその指は 1 であることがわかる。

ルール 2

1 度打鍵した鍵は、離鍵するまで同じ指で押される。なお、本ルールは打鍵中における指の置換を考慮していない。本ルールは、ある打鍵が発生したとき、打鍵した鍵以外の鍵上にあるマーカを運指の候補か

ら外すことで、考えられる運指の候補を削減する。例えば、カメラの位置と実際に打鍵した指の間に他の指がある場合、前の指が後ろの指にあるマーカを隠すため、システムは前の指を打鍵した指であると誤認識してしまう。このとき、打鍵しているはずの指が他の打鍵で使われたり、他の鍵上に移動したりした場合にルール 2 によって運指が修正される。

ルール 3

図 5 に示すように同じ高さの音は、同じ運指になることが多い。しかし、本ルールは想定する楽曲すべてに適用できるルールではない。したがって、ある鍵の打鍵から離鍵においてルール 1 やルール 2 を用いても運指が特定できなかったとき、本ルールを用いて過去の運指データから予測を行う。具体的には、運指が特定できない音高を発見した場合、該当する音高から 4 音さかのぼった音列中、同じ音高の運指を採用する。

ルール 4

図 6 に示すように同じ構造をした音列は同じ運指列になることが多い。ルール 4 はルール 3 を適用し



図 5 ルール 3 の例



図 6 ルール 4 の例



図 7 プロトタイプシステム

でも予測できなかった場合に用いる。具体的には、該当する音高からさかのぼった 8 音でグループを形成し、過去に演奏された音列に対し音高列マッチングを行い、マッチしたグループの運指を採用する。提案手法ではルール 3 適用後にルール 4 を適用するとしているが、ルール 3 とルール 4 の適用順序を逆にした場合および、両者を同時に適用して結果を比較するといった方法も考えられる。これらの手法の評価は今後の課題である。また、ルール 3 におけるさかのぼる音列数や、ルール 4 におけるグループ構成音列数の最適な閾値の算出に関しても今後の評価実験を行い検討を行う予定である。

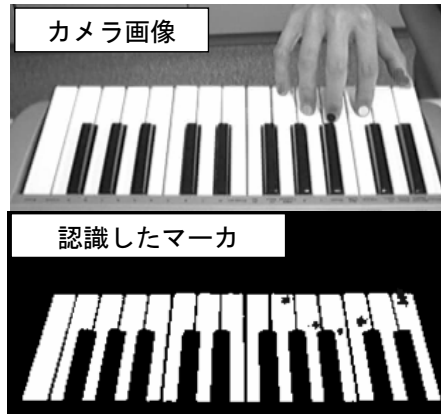


図 8 カメラから取得した画像と画像処理結果

### 3 運指取得システムの実装

2 章で述べた運指取得システムのプロトタイプを実装した。プロトタイプシステムを図 7 に示す。PC としては、SONY 社の VGN-S92PS (CPU 2.13GHz, RAM 2GB) を使用し、MIDI 鍵盤としてフルサイズ 25 鍵盤をもつ M-AUDIO 社の OXYGEN8 を使用した。カメラは、Creative 社の WebCam Live! Motion (解像度 320 × 240, フレームレート 最大 30fps) を用いた。マーカは NICHIBAN 社のマイタック カラーラベル [13] から判別しやすい色を選定し、各爪に貼り付けた (図 1)。カメラは爪に貼ってあるマーカがよく見える位置に 1 台設置した。1 台のカメラで処理する鍵盤領域が広がると相対的にマーカが小さくなり認識率が低くなるため、今回のプロトタイプでは 1 台のカメラによる認識範囲を 2 オクターブとした。最適なカメラ位置から取得した画像と解析する鍵盤領域を図 8 に示す。PC 上のソフトウェアの開発は、Windows XP 上で Visual C++ .NET 2003 と Intel 社の OpenCV ライブラリを用いて行った。

#### カラーマーカの色抽出

各マーカの色抽出は、サンプル HSV 値をあらかじめ取得しておき、それぞれの値から一定範囲に入っている色を抽出する方法で行った。図 8 下に、抽出したマーカを図示する。しかし、照明の輝度変化とシール

表 2 課題曲

楽曲名	作曲者名	テンポ	略称
トロイメライ「子供の情景」より	R. Schumann	四分音符 = 64	曲 A
楽しき農夫	R. Schumann	四分音符 = 107	曲 B
ソナタ K.331 第 3 楽章 トルコ行進曲	W. A. Mozart	四分音符 = 122	曲 C

の光沢による反射から指を立てた状態と寝かせた状態のマーカ両方を完全には認識できなかった。黒鍵の打鍵や音域の広い音程の打鍵時は、指先が寝た状態になるが、通常、鍵盤演奏では指を立てて演奏することが多いため、指を立てた状態の HSV 値をサンプル値とした。

プロトタイプでは片手の運指検出を行い、カラーは、親指から小指にかけて赤、青、緑、ピンク、オレンジの 5 色を選定した。

#### 運指情報の提示

本研究では、リアルタイム性を実現するために複雑な画像処理を避け、ルールを適用することで認識精度を高めている。システムが生成する運指は「マーカが正確に認識されルールを満たすもの」や「マーカが誤認識されルールにより推測あるいは修正されたもの」などさまざまである。また、いずれのルールも普遍性に欠けるため、各状況に応じて生成された運指情報の確度は異なる。したがって、ユーザへの提示方法としては、もっとも確度が高い運指を提示する以外に、確度をもとに順位付けされた複数の運指情報を提示する方法が考えられる。提示方法は、利用アプリケーションによって異なる。例えば、演奏者の誤った運指をシステムが指摘する場合は前者が、レッスン時のように指導者が生徒の運指確認の参考にする場合は後者の方が適している。正確な確度算出は、さまざまな要素が複雑に絡み困難であるため、本研究では、ルールを満たし、かつ、最もマーカ認識率の高い運指をユーザに提示する。確度算出のアルゴリズムおよび実装、評価は今後の課題である。

#### 4 評価と考察

実装したプロトタイプの有効性を示すために、鍵盤習熟者 3 名に実際に演奏してもらい運指正答率を調査した。また、ルールを適用した場合とそうでな

い場合とで比較しルールの有効性について検証した。以下に、実験の詳細について述べる。

##### 4.1 実験の手続き

###### 課題曲

表 2 に実験で演奏してもらった課題曲とプロファイルを示す。また、実験では、全ての楽曲において最初から 20 小節までを右手で弾く部分のみ演奏してもらった。

曲 A は課題曲の中でもっともテンポが遅い楽曲である。しかし、楽譜にはスラーが記譜されているため、演奏者は親指と他の指との交差を行いながら音をつなげていく。また、曲 B は 9 割が 3 和音や 2 和音で構成され、ほとんどの和音はスタカートで演奏されるため、システムは瞬時に複数の運指を認識しなければならない。さらに、曲 C は課題曲の中でもっともテンポが速く、16 分音符の連続や装飾音、指の交差も多い。

###### 被験者

課題曲を楽譜に指定されたテンポで初見で十分に弾きこなせる音楽大学ピアノ科学生 2 名と鍵盤暦 22 年の大学生 1 名に実験してもらった。

###### システム構成

プロトタイプシステムを用い、MIDI ノートオンおよびノートオフ情報、打鍵および離鍵時刻、ルールを適用したときの運指とルールを適用しなかったときの運指を記録した。また、被験者の手元をデジタルビデオカメラで撮影することで正解運指列を得た。

###### 実験の手続き

被験者には曲 A から順にすべての曲を 2 回ずつ弾いてもらった。各被験者には、弾きやすい運指で楽譜に指示されたテンポで弾くようにと指示した。

表 3 ルール適用事例

事例番号	項目	ルール 1	ルール 2	ルール 3	ルール 4
1	マーカの認識率向上				
2	運指が不特定できない時の予測				
3	打鍵した指が他の指に隠される問題				

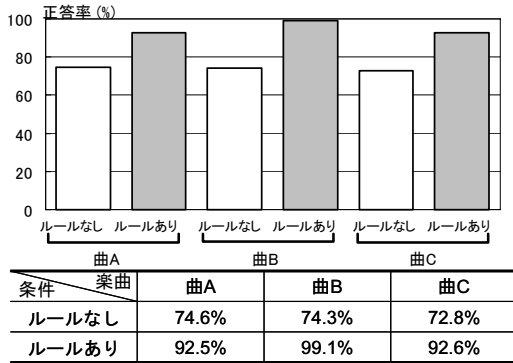


図 9 認識正答率

#### 4.2 評価結果

図 9 に各楽曲のルール適用時と未適用時の運指認識正答率を示す。ルール適用時の認識正答率は平均して 95%と高く、指の交差、複数の指の同時判定、高速で複雑な演奏の追従を高い精度で行えた。また、これらは有意水準 5%でルール未適用時(平均正答率 74%)との有意差も確認できた。

以下、2.3 節で定義したルールを適用することで、どのように誤認識が解決されたかについて事例をもとに述べる。表 3 に事例を示し、問題解決に貢献したルールには、印を示した。

##### 事例 1

マーカ認識の不安定さから、人差指、中指、薬指、小指の並び順が崩れた状態で認識されてしまう場合がある。これは、ルール 1 を適用することでありえない交差を検出することで防止できた。

##### 事例 2

小指や親指は、他の指に隠されることが多くマーカ検出がうまくできなかった。したがって、ルール未適用時は運指が特定できなかったため認識ミスの大きな原因となっていた。例えば、図 10 において「曲

A の 2 小節目の E 音」、曲 B のスタッカート付和音、「曲 C の 1 小節目の A 音」、「曲 C の 2 小節目の最後の C 音」などで頻繁に親指の検出ミスが生じていた。

一方、ルール適用時は、ルール 1 の貢献が大きく他のマーカや MIDI ノートナンバーから運指を補正できた。また、ルール 2 から特定できなかった鍵の打鍵中に、全ての指を監視し、鍵間を動きまわっている指は候補から外すことで運指候補を削減できた。このように、双方のルールを適用することで運指の特定率を高めた。さらに、曲 B のスタッカート付和音ではルール 1 やルール 2 を用いて認識できない場合があった。このとき、ルール 3 による過去の運指データの予測により補正が行われた。

##### 事例 3

図 10 に示す「曲 A の 2 小節目の最初の C 音」では、カメラから見て親指の前にある薬指で打鍵しているとシステムは誤認識していた。また、「曲 C の 4 小節目の最後の E 音」では打鍵時、親指が認識されず、その後、親指の前に現れる人差指や中指で打鍵したと誤認識してしまった。

ルールを適用した場合、曲 A では打鍵直後は薬指と認識されたが、F 音を小指で弾くために薬指が移動し親指が出現することで、ルール 2 から薬指は運指候補から外され親指が認識された。また、曲 C では E 音打鍵時にルール 1 から親指で打鍵されていると認識され、親指の前に覆いかぶさる人差指や中指は、親指打鍵時に他の鍵上にあるため、運指候補から外され正しく認識された。

ルール適用時にうまく認識できなかった箇所

図 10 の「曲 A の 5 小節目の 2 音目の F 音」は人差指で打鍵していたが中指で打鍵していると誤認識してしまった。これは、事例 3 においてルール 2 がうまく機能しなかった例である。ルール 2 では前に

【曲A】

正解運指 1 2 1 2 3 1 4 4 4 3 2 3 5 1 2 1 3

誤認識

事例2 事例3 事例3

【曲B】

正解運指 全ての和音の運指は「1,3,5」

【曲C】

正解運指 4 3 2 1 3 4 3 2 1 3 3,5 1,3 4 3 2 1 2,4 1

誤認識

事例2 事例2 事例3

図 10 認識結果の詳細

ある指が他の鍵に移動することで運指を修正しているが、F音が打鍵されてから離鍵されるまで中指が動かなかったため誤認識してしまった。

#### 処理速度

画像処理をシンプルにし、ルールにより補正することで、リアルタイムで運指認識ができた。1フレームあたりの処理時間は平均20msecとカメラのフレームレート内で画像処理を完了できる。今後は、1台のPCでより広い音域を認識するために複数台のカメラの使用を考えている。また、状況に合わせて適用するルールを動的に選択し処理の軽減を行う予定である。

#### マーカ認識率の向上

今回マーカとして市販されているシールを用いたが、色が限定されており光沢感もあったため光に反射してマーカ認識率が下がるという問題があった。今後は、マーカ認識率を向上させるために、光の反射を

カットする偏光フィルタをカメラに取り付けたり、光沢のないマーカを使用するといった工夫を行う予定である。また、爪先のマーカだけでなく指の肌色部分も認識させることで、手の可動領域を考慮した認識率の向上および計算量の削減を行う。

#### 5 関連研究

鍵盤の運指取得を主目的とした研究事例は少ない。データグローブ[10]やFingerRing[12]、Lightglove[2]から出力されるデータを用いれば運指を取得することが可能であるが、これらのデバイスは鍵盤演奏を使用目的として考えていないため、指にまとわりつくセンサや手首に設置する回路、配線などで演奏を妨げてしまう。また、バーチャルキーボード[14]は本研究と同様のアプローチでカメラを用いて実時間指先検出を実現している。しかし、画像処理ベースで認識を行い、使用用途はPC用キーボードを想定しているため指が交差するといった状況に対応できない。



一方、あらかじめ入力しておいた楽曲データをもとに運指を自動生成する試みは多数存在する。運指生成のアプローチとして、ある2本の指が打鍵可能な音域を求め最適な運指を見つける方法[1]や手の移動距離[9]、楽曲の構造に着目[11]した研究などがある。これらは、手や指に負担をかけない模範的な運指を生成することを目的としており、実際に演奏者が弾いている楽曲の運指をリアルタイムで取得することを目的としている本研究とは異なる。また、これらの方式では1章で述べたような音楽的な指使いを生成することは難しく、片手かつ単旋律でしか運指を生成できないなど実用性に問題がある。ただし、これらの研究はいくつかの運指ルールを含んでおりこれらのルールを取り込むことで提案方式の精度を高められる可能性がある。

ハンドジェスチャの研究として、RGB出力のカメラや赤外線カメラを用いユーザの指先を検出しインタラクションを行う研究[7][3]がある。これらは、いずれも複数のカメラを用いるなどカメラベースでジェスチャ認識を行っており、本研究のように画像処理以外の特性を用いて認識率を向上させるといった工夫は行っていない。

## 6 おわりに

本研究では、爪に貼り付けたカラーマーカをカメラで認識することで実時間で運指取得するシステムを構築した。提案システムは実時間処理を実現するために複雑な画像処理を避け、演奏や手、鍵盤の特性をもとに定義したルールから運指認識率の向上を図った。プロトタイプシステムの評価結果より、運指を高い精度で取得できることがわかった。

今後の課題としては、これまでの章で述べたもの以外に、両手への拡張や、さまざまなレベルのピアノ習熟者による評価実験を行う予定である。

謝辞 本研究の一部は、文部科学省21世紀COEプログラム「ネットワーク共生環境を築く情報技術の創出」によるものである。ここに記して謝意を表す。

### 参考文献

- [1] Hart, M. and Tsai, E.: Finding Optimal Piano Fingerings, *The UMAP Journal*, Vol.21, No.2 (2000), pp. 167-177.
- [2] Howard, B. and Howard, S.: Lightglove: Wrist-Worn Virtual Typing and Pointing, *Proc. of the Fifth IEEE International Symposium on Wearable Computers (ISWC '01)*, 2001, pp. 172-173.
- [3] Kolsch, M., Turk, M., Hollerer, T. and Chainey, J.: Vision-Based Interfaces for Mobility, *Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous '04)*, 2004, pp. 86-94.
- [4] 井上直幸: ピアノ奏法, 春秋社, 2003, pp. 51.
- [5] 遠隔レッスンシステムのホームページ.  
<http://www.yamaha-mf.or.jp/onken/soft/theme5.html>
- [6] 岡田暁生, 伊東信宏, 近藤秀樹, 大久保賢, 小岩信治, 大地宏子, 筒井はる香: ピアノを弾く身体, 春秋社, 2003, pp. 31-53.
- [7] 小林貴訓, 佐藤洋一, 小池英樹: Enhanced Deskのための赤外線画像を用いた実時間指先認識インターフェース, 第7回日本ソフトウェア科学会インタラクティブシステムとソフトウェアに関するワークショップ(WISS '99), インタラクティブシステムとソフトウェアVII, 近代科学社, 1999, pp. 49-54.
- [8] セイモア・バーンスタイン: 心で弾くピアノ, 音楽之友社, 2002, pp. 63-65.
- [9] 関口博之, 英保茂: 計算機におけるピアノ演奏動作の生成と表示, 情報処理学会論文誌, Vol.40, No.6, 1999, pp. 2827-2837.
- [10] データグローブのホームページ.  
<http://www.vrealities.com/glove.html>
- [11] 林田教裕, 水谷哲也: 楽曲構造に基づくピアノ運指ルールの論理表現, 情報処理学会第65回全国大会講演論文集, 第2分冊, 2003, pp. 203-204.
- [12] 福本雅明, 外村佳伸: 指釦:手首装着型コマンド入力機構, 情報処理学会論文誌, Vol.40, No.2 (1999), pp. 389-398.
- [13] マイタック カラーラベルのホームページ.  
<http://www.nichiban.co.jp/stationery/product/product/b-9.htm>
- [14] 松井望, 山本喜一: バーチャルキーボード: ビデオ画像からの頑健な実時間指先検出の実現, 第3回プログラミングおよび応用のシステムに関するワークショップ(SPA '00) 論文集, 2000.