

鍵盤奏者のための実時間運指取得システムの設計と実装

Design and Implementation of a Real-Time Fingering Detection System for Pianists

竹川 佳成 寺田 努 西尾 章治郎*

Summary. Fingering is one of important aspects in piano performances because it affects pianists' musicality. On the other hand, we cannot acquire enough samples of actual fingering information. 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も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、優れた運指も無数に存在する [1, 2, 3]。

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など数少ない。これは、運指の記載に多大な時間がかかることが大きな原因となっており、手軽に運指を記録できるソフトウェアが求められている。また、演奏を実際に聞きながら、リアルタイムで演奏者の運指を取得できれば、効率的な演奏の学習や新たな音楽視聴スタイルの創出が期待できる。

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章で関連研究について説明し、5章で今後の展望について述べ、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、コンサートなど演奏者が演奏に集中している状況でも利用できるようにできる限り演

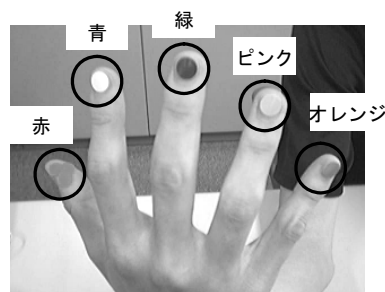


図 1. 指先に付けたカラーマーカ

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。指先に取り付けたボタンや指の関節に取り付けたセンサ、データグローブ [4] 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。カラーマーカはうすいシールであり、装着に違和感が少ない。

(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り、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。

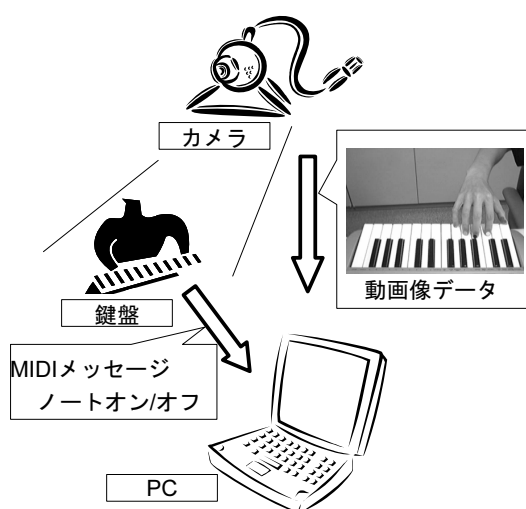


図 2. システム構成

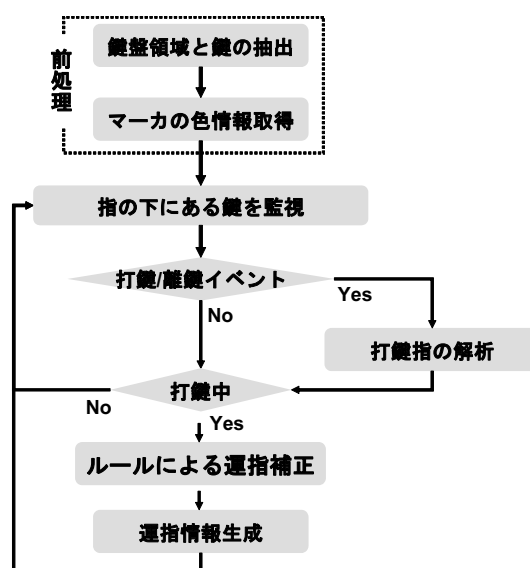


図 3. 運指解析処理のフローチャート

2.1 運指解析処理の概要

運指解析処理部の流れを図 3 に示す。カメラや鍵盤は固定されていると想定しており、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ら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。

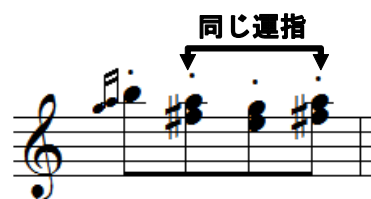


図 4. ルール 2 の例

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位置関係が崩れることがない。すなわち、右手にマーカを付けた場合、左から順に人差指、中指、薬指、小指の順になる。

ルール 2

図 4 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。

ルール 3

図 5 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

基本的に 1 本の指では、1 つの鍵しか打鍵できない。例外として、1 本の指で隣り合った 2 つの白鍵を打鍵する場合はあるとしている。本ルールの同時打鍵は、和音だけでなくレガート演奏でも適用される。レガート演奏とは、連続した 2 音において前者の音を離鍵する前に後者の音を打鍵する奏法で、普段の演奏でよく使用される。このとき前の鍵で強い

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	同じ高さの音は、同じ運指を使うことが多い。
3	音列の構造が類似しているならば、同じ運指列になることが多い。
4	隣り合った白鍵を除き、鍵の同時打鍵は異なる指を用いる。
5	基本的に1度打鍵した鍵は、離鍵するまで同じ指を使う。
6	基本形を保ちながら演奏する。



図 5. ルール 3 の例

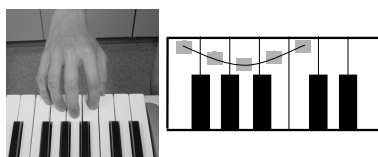


図 6. 基本形

ている指は運指候補から外れる。

ルール 5

基本的に、1度打鍵した鍵は、離鍵するまで同じ指で押される。

ルール 6

手や指の関節に負担をかけずスムーズに演奏をするために、基本形(図6)をできるだけ保ちながら演奏する。

これらのルールを単独あるいは組み合わせて用いることで運指補正を行う。

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Roland社のOXYGEN8を使用した。カメラは、logicool社のQcam(解像度320×240, フレームレート最大30fps)を用いた。マークはNICHIBAN社のマイタックカラーラベル[5]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ク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ク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

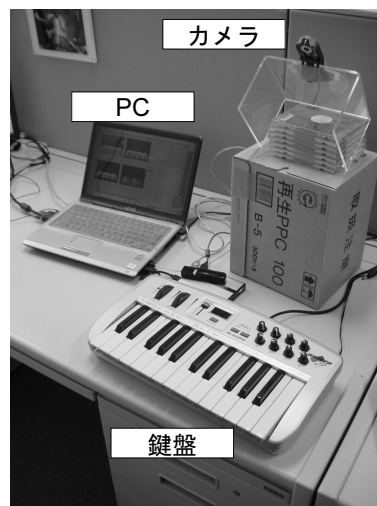


図 7. プロトタイプシステム

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。

3.1 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、抽出したマークを図示する。しかし、照明の輝度変化とシール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ク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。

3.2 ルール適用事例

前節で記述したようにカメラの位置や輝度変化によってマーク認識が不安定になるため、運指を特定できなかったり、運指の誤認識が発生する。本節では、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の事例を述

表 2. 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	ルール 5	ルール 6
1	マーカの認識率向上						
2	運指不特定時の予測						
3	打鍵した指が他の指に隠される問題						



図 8. カメラから取得した画像と画像処理結果



図 9. トルコ行進曲 (一部)

べる．表 2 に適用事例を示し，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る．誤りを起こしたマーカが特定できる場合は，そのマーカを無視するようにし，特定できない場合は全ての指が認識ミスをしているとみなした．

事例 2

打鍵した指のマーカが認識できなかった場合，ルール 1 に基づき，他のマーカの認識状況と鍵盤から得られるノートオン情報を組み合わせることで打鍵した運指を特定できる．例えば，人差指の下にある鍵より低いノートナンバーが打鍵されたときは，親指で打鍵されたことがわかる．また，ルール 4 から運指が特定できなかった鍵の打鍵時や打鍵中に，他の鍵の打鍵や離鍵で使用した指は，運指候補から外される．さらにルール 5 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される．

以上のルールを適用しても離鍵までに運指が特定できなかった音高については，ルール 2 を用い運指不特定音高から 4 音さかのぼった音列中，同じ音高の運指を運指不特定音高の運指として採用する．さらに，ルール 2 を適用しても予測できなかった場合，ルール 3 を用いる．具体的には，運指不特定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．なお．ルール 2 におけるさかのぼる音列数やルール 3 におけるグループ構成音列数の最適な閾値の算出およびルール 2 と

ルール 3 の優先度に関する考察については今後の課題である．

事例 3

カメラの位置から打鍵した指の前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してしまい，前の指を打鍵した指であると誤認識してしまう．そこで，本研究では常に鍵盤上にあるマーカを監視し，打鍵中の鍵盤上にある運指候補をルールから補正する．具体的には，ルール 4 やルール 5 から，打鍵しているはずの指が，他の打鍵で使われたり，他の鍵盤上にあると認識されたりした場合に修正される．また，ルール 6 から中指，人差指と薬指，小指，親指の順で後方になる．すなわち，中指は他の指を隠しやすく，親指はもっとも隠れやすい．したがって，前方にある指から後方の隠れていた指に変わることはあっても，後方の指が前方の指を隠すことはない．

3.3 評価

実装したプロトタイプの有効性を示すために，図 9 に示すモーツアルトのトルコ行進曲 (一部) を演奏し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．トルコ行進曲のテンポは Allegro と速く，白鍵だけでなく黒鍵も使用する．運指認識ミス数を図 10 に示す．図 10 において，ルール適用時と未適用時それぞれ 10 回演奏したときの運指認識ミス数の平均を棒グラフ，平均値 ± 標準偏差の幅を線分で示している．また，図 9 において音符上の数字は，ルール適用時にシステムが認識した運指の出力結果の紹介である．数字は，親指から順に 1 からナンバリングした番号を示す．網掛けになっている数字は誤認識した運指を，上の数字は正しい解答を表す．

評価結果

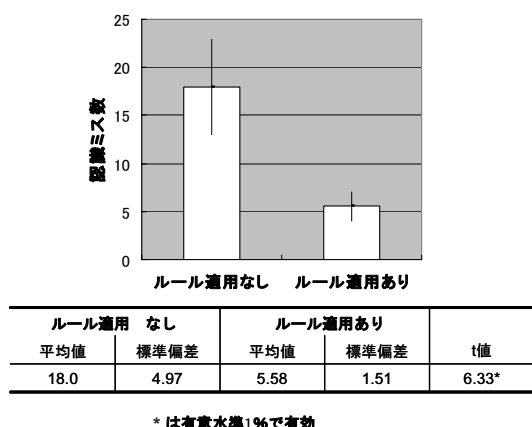


図 10. 認識ミス数の平均と標準偏差

ルール適用時の認識正答率は平均して 95%と高かった。また、ルール未適用時との有意差も確認できた。以下にルールや事例が適用された場面について説明する。

事例 1 手の形や指の向きから小指のオレンジ色マーカの一部が親指の赤色マーカと認識されてしまった。すなわち親指のマーカが 2 箇所 (親指のマーカと小指のマーカの一部) で検出されていた。事例 1 から親指のマーカと小指のマーカが隣接することはないため、小指の一部において認識された親指マーカを無視することで適切なマーカ認識ができていた。

事例 2 小指や親指は、他の指に隠れてしまう可能性が高くマーカ認識がうまくできなかった。それゆえ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原因となった。ルール適用時は、ルール 1 の貢献が大きく人差指や薬指の運指認識状況と MIDI ノートナンバーから運指を補正できた。また、11 小節目の最初の E と G の和音は、G が認識できていなかった。次の E と G の和音打鍵時はうまく認識でき、このときルール 2 から運指補正が行われた。

ルール 4 譜面上のスラーが記譜されている箇所や和音は、同時打鍵が多く生じているため、ルール 4 から打鍵時にその他の鍵を打鍵している指は運指候補から外した。

ルール 5 3 小節目最後の E 音を打鍵している親指は、打鍵時、1 つ前の Dis 音を打鍵している人差指に隠れてしまっているが、打鍵中から離鍵時にかけて 4 小節目の第 1 音 H を打鍵するために手全体が移動したため、親指が出現し、ルール 5 から正しい運指を得られた。

一方、ルール適用時にうまく認識できなかった箇所について説明する。網掛け部分の認識ミス原因は、人差指が中指に隠れているために生じた。ルール 4 では正しい指を隠している指が他の鍵へ動くこと

で、隠している指がミスであることに気付く。この場合、完全に網掛けの指が前方にあったため事例 3 が適用されなかった。プロトタイプでは、ルール 3 は運指不特定時のみ用いていたが、それ以外の状況においてルール 3 を適用することで解決できる。例えば 2 番目の網掛け部分は、11 小節目、13 小節目、15 小節目の運指傾向から修正できる。しかし、演奏家は音楽表現のためにあえて異なった運指をしたという見方もあるため、マーカ認識率などから確信度を算出し、選択的に適用する必要がある。

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や広角レンズ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するようにアルゴリズムを変更する予定である。こうすることで、マーカ認識ができていなくても指の存在だけは把握でき、マーカ認識を補正できる。

4 考察

4.1 応用

本システムの応用例として運指情報を用いた伴奏システムが考えられる。これは人 (演奏者) の演奏に合わせてコンピュータが伴奏するシステムである [13]。従来の伴奏システムは、あらかじめ音高や楽曲の構造といった楽曲情報をもとに人が演奏している位置を特定していた。ある楽曲を習得してから、運指は大幅に変化しないという特性から、新たに運指情報を加えることでよりロバスタな追跡ができる。例えば、音高情報と運指情報と双方のマッチングをとることで、演奏位置の確信度が高まる。また、演奏中に楽譜と異なる鍵を演奏者が打鍵した場合、意図的に違う音を打鍵したか、誤って打鍵してしまったかを運指から判断できる。

その他の応用例として学習支援が考えられる。冒頭で述べたように、運指は音楽表現に影響を与える。したがって、コンサートや公開レッスンといった場で、演奏者の運指を学ぶことができれば効率的な学習が期待できる。また、薬指や小指は他の指に比べてコントロールしにくいいため、演奏者はさまざまな練習曲集を用いて各指をスムーズに動かせるように

鍛錬する。各指の打鍵タイミングのずれやペロシティの乱れを計測することで、演奏者は苦手なフレーズとそのときの指の組合わせを認識でき、効率的にトレーニングできる。さらに、手が小さい、指が短いといった身体的な特徴からペロシティの問題が克服できない場合、システム側でペロシティを調節するという演奏支援ができる。

4.2 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 [4] や FingerRing [6], Lightglove [7] 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 [8] 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 [9] や手の移動距離 [10]、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 [12] 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。

5 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、速い演奏を行っても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、演奏者ごとに運指特性をルール化する試みや、さまざまな楽曲、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。ここに記して謝意を表す。

参考文献

- [1] 井上直幸：“ピアノ奏法,” pp. 51, 春秋社 (2003).
- [2] 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,” pp. 31-53, 春秋社 (2003).
- [3] セイモア・パースタイン：“心で弾くピアノ,” pp. 63-65, 音楽之友社 (2002).
- [4] データグローブのホームページ.
<http://www.vrealities.com/glove.html>
- [5] 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>
- [6] 福本雅明, 外村佳伸：“指鉤”:手首装着型コマンド入力機構,” 情報処理学会誌, Vol.40, No.2, pp.389-398 (1999).
- [7] Bruce Howard, and Susie Howard:“Lightglove: Wrist-Worn Virtual Typing and Pointing,” Proc. of the Fifth IEEE International Symposium on Wearable Computers (ISWC '01), pp. 172 - 173 (2001).
- [8] 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,” 日本ソフトウェア科学会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論文集 (2000).
- [9] M. Hart and E. Tsai:“Finding Optimal Piano Fingerings,” The UMAP Journal 21(2), pp. 167 - 177 (2000).
- [10] 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,” 情報処理学会誌, Vol.40, No.6, pp. 2827 - 2837 (1999).
- [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, pp. 203 - 204 (2003).
- [12] Mathias Kolsch, Matthew Turk, Tobias Hollerer and James Chainey:“Vision-Based Interface for Mobility,” Proc. of the 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004) (2004).
- [13] 長嶋洋一, 橋本周司, 平賀譲, 平田圭二：“コンピュータと音楽の世界-基礎からフロンティアまで,” pp. 252-269, bit 別冊, 共立出版 (1998).