

# 局所領域の同時多点追跡に基づくピアノ運指認識に関する研究

情報科学研究科 情報科学専攻 H11218M 西山乗

指導教員：橋本学 教授

## 1. はじめに

初心者の演奏学習をサポートするために運指の自動認識が望まれている。しかし、従来の運指認識手法 [1] [2] は指先のセルフオクルージョンによる誤認識が問題となる。問題を解決するための基本アイデアは3つある。複数の指先を同時に追跡し、指先間の相対的な位置関係を考慮した目的関数を用いることで、ミストラップを回避する。また、指先が未検出の場合、他の指先位置との相対位置を用いて指先位置を推定することによって安定した追跡を可能にする。さらに、以上の処理を離散的 DP トラッキング (離散的 DPT) と解析的 DP トラッキング (解析的 DPT) の2段階処理で実現することによって高精度な同時多点追跡を可能とする。これらのアイデアによって、本稿では指先のセルフオクルージョンに頑健な追跡ベース運指認識手法を実現する。

## 2. 提案アルゴリズム

提案手法のデータの流れを図1に示す。提案手法はまず、各画像ごとに指先と手首を検出することによって動画中の候補指先位置を設定する。そして、検出結果と手動によって設定された指先初期位置を元に同時多点追跡を用いて指先追跡経路を算出する。最後にこの追跡経路と打鍵された鍵盤位置を比較することで運指を判定する。

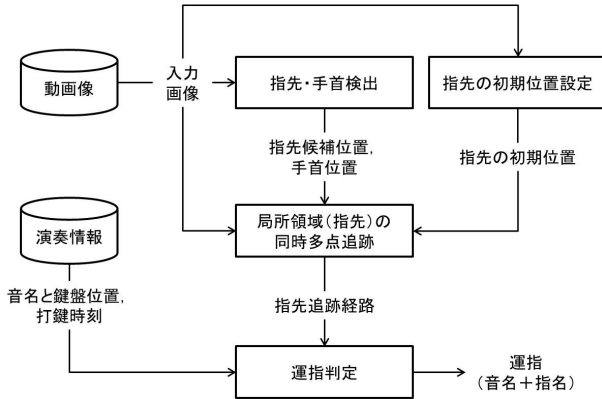


図1 提案アルゴリズムのブロック図

同時多点追跡の1段階めの処理である離散的 DPT では、想定される候補指先位置と全指先の対応付けの中から最適な対応付けを一括処理により算出する。最適な対応付けとは、指先の局所コストと指先位置のフレーム間の変位量、隣接する指先間の相対位置の変位量の3つを評価する。

同時多点追跡の2段階めの処理である解析的 DPT では、各フレームの局所コストを離散的 DPT で得られた追跡位置を中心として2次関数近似することで、探索空間を画像全体とした最適追跡経路を解析的に算出する。

## 3. 離散的 DPT を用いた指先対応付け

本研究は5本の指先と候補指先位置との対応付け問題を時間方向に拡張して解くために離散的 DPT を用いた。本研究は第  $t$  フレームの追跡パラメータを  $\mathbf{w}_t = (\mathbf{w}_{t,1}, \dots, \mathbf{w}_{t,M})^T$ ,  $\mathbf{w}_{t,m} = (x_{t,m}, y_{t,m})^T$  と定義して、目的関数 (1) を最小化する最適経路  $(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_T)$  を求める。ただし、 $n$  は対象物体の番号であり、 $T$  は動画のフレーム数である。

$$F(\mathbf{w}_1, \dots, \mathbf{w}_T) = \sum_{t=1}^T \sum_{m=1}^M d_{t,m}(\mathbf{w}_{t,m}) + \lambda_1 \sum_{t=2}^T \sum_{m=1}^M (\mathbf{w}_{t,m} - \mathbf{w}_{t-1,m})^2 + \lambda_2 \sum_{t=2}^T \sum_{m=2}^M \{(\mathbf{w}_{t,m} - \mathbf{w}_{t,m-1}) - (\mathbf{w}_{t-1,m} - \mathbf{w}_{t-1,m-1})\}^2 \quad (1)$$

ただし、 $\lambda_1$  と  $\lambda_2$  は位置の連続性コストに対する重みと相対位置の連続性コストに対する重みを意味し第  $t$  フレームに対象物  $m$  が位置  $\mathbf{w}_{t,m}$  に存在することを仮定した場合の局所コスト  $d_{t,m}(\mathbf{w}_{t,m})$  とする。なお、本稿では局所コストにチャンファーマッチングを用いる。式1を用いたとしても、セルフオクルージョンにより正しい指先位置に候補指先位置が設定されない場合は、正しい指先位置を追跡することが不可能となる。そこで、探索する組み合わせの中に指先位置を同定しないワイルドカードの設定を許容することとした。目的関数の計算時には他の指先位置から推定した指先位置を用いることで評価する。また目的関数の漸化式を式2に示す。

$$f_t(\mathbf{w}_t) = \sum_{m=1}^M d_{t,m}(\mathbf{w}_{t,m}) + \min_{\mathbf{w}_{t-1}} \left\{ f_{t-1}(\mathbf{w}_{t-1}) + \lambda_1 \sum_{m=1}^M (\mathbf{w}_{t,m} - \mathbf{w}_{t-1,m})^2 + \lambda_2 \sum_{m=2}^M \{(\mathbf{w}_{t,m} - \mathbf{w}_{t,m-1}) - (\mathbf{w}_{t-1,m} - \mathbf{w}_{t-1,m-1})\}^2 \right\} \quad (2)$$

式2を用いて各フレームにおける累積コストを第2フレームから第  $T$  フレームまで逐次計算する。その後、第  $T$  フレームの累積コストが最小になる位置  $\mathbf{w}_T$  からバックトレースすることにより、最適経路を決定する。

## 4. 解析的 DPT を用いた高精度指先追跡

離散的 DPT で得られた追跡経路は離散的に得られた指先候補位置のみの限定的な探索空間における最適経路である。したがって、候補指先位置が含む誤差をそのまま引き継いでしまう。このように離散的 DPT の結果により正解位置付近の位置が既知である場合に限り、解析的 DPT は近似的な全探索を高速に実現することができる。

そのために、具体的には局所コストを2次関数で連続的に表現する必要がある。これにより、累積コスト  $f_t(\mathbf{w}_t)$  も2次関数で表現することが可能となる。

$$f_t(\mathbf{w}_t) = \mathbf{w}_t^T \mathbf{A}_t \mathbf{w}_t + \mathbf{b}_t^T \mathbf{w}_t + c_t \quad (3)$$

すると各項は以下のように導出することができる。

$$\mathbf{A}_t = \mathbf{P}_t + \Lambda - \lambda^2 \{\mathbf{A}_{t-1} + \Lambda\}^{-1} \quad (4)$$

$$\mathbf{b}_t = \mathbf{p}_t + \Lambda \{\mathbf{A}_{t-1} + \Lambda\}^{-1} \mathbf{b}_{t-1}$$

$$c_t = r_t - \mathbf{b}_{t-1}^T \{\mathbf{A}_{t-1} + \Lambda\}^{-1} \mathbf{b}_{t-1} / 4 + c_{t-1}$$

$$\mathbf{w}_{t-1} = \{\mathbf{A}_{t-1} + \Lambda\}^{-1} \{\Lambda \mathbf{w}_t - \mathbf{b}_{t-1} / 2\} \quad (5)$$

ただし、

$$\Lambda = \lambda_1 \mathbf{I} + \lambda_2 \mathbf{V}^T \mathbf{V} \quad (6)$$

$$\mathbf{V} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 & -1 \\ 0 & \cdots & \cdots & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

とする。上記までの式を用いた具体的な処理手順は、以下の通りである。まず、各フレームの局所コスト関数を算出する。次に、式4を用いて第1フレームから第Tフレームの順番で累積コスト  $d_t(\mathbf{w}_t)$  を逐次処理で計算する。そして、最終フレームの累積コスト  $f_T(\mathbf{w}_T)$  を最小にする位置を  $\mathbf{w}_T$  に設定し、以降は式5を用いて第Tフレームから第1フレームの順番で位置  $\mathbf{w}_t$  をバックトレース処理で順次計算する。

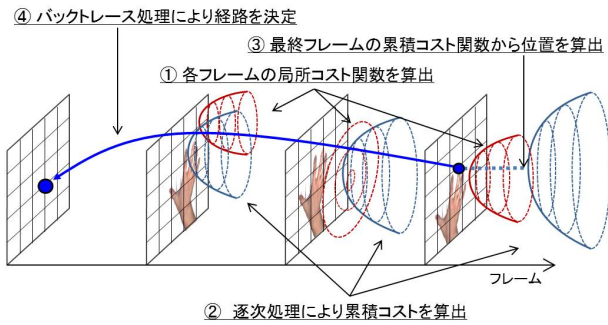


図2 最適経路を求める処理手順のイメージ

## 5. 実験結果と考察

実験に用いた撮影環境を図3に示す。MIDI通信に対応した電子ピアノ（カシオ計算機株式会社のPX-750BK）を使用し、鍵盤直上80mmの高さに撮影用のカメラ（The Imaging Source社のDFK22AUC03）を設置した。また、鍵盤上の照明条件を均一にするためにLED照明器具を設置した。実験には初心者向けの演奏楽曲としてMenuet(J.S.Bach)の1小節目から16小節目を演奏する動画像を使用した。この演奏曲は指先のセルフオクルージョンが多数発生する。

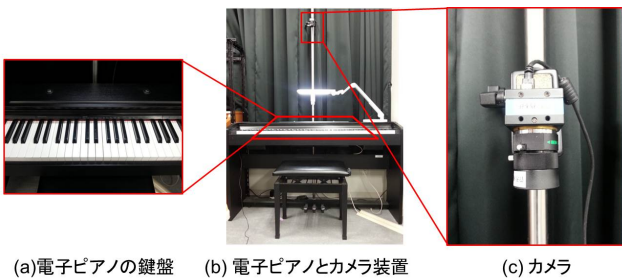


図3 撮影環境

まず、指先追跡の比較実験の結果を示す。比較手法は色ヒストグラムを用いたMeanShift追跡(MST)[4]と離散的DPTのみを用いた追跡手法、離散的DPTと解析的DPTを用いた提案手法の3手法とする。各手法に対する評価方法は追跡成功率と処理時間を用いた。この追跡成功率は追跡結果と正解位置とのユークリッド距離がしきい値以下であった割合とした。指先追跡の比較結果を表1に示す。Mean-Shift追跡の追跡成功率が低い理由は、セルフオクルージョンや指の同士の密着が起こるごとに他の指へミストラップされてしまったからである。これに対して、提案手法はミストラップすることなく、安定した追跡が可能であった。また、離散的DPTに比べて解析的DPTを用いた追跡手法は、指先検出で未検出のフレーム数が多く発生したR1(親指)に対して追跡成功率の向上が見られた。指先検出結果と提案手法の指先追跡結果例を図4に示す。

ラップすることなく、安定した追跡が可能であった。また、離散的DPTに比べて解析的DPTを用いた追跡手法は、指先検出で未検出のフレーム数が多く発生したR1(親指)に対して追跡成功率の向上が見られた。指先検出結果と提案手法の指先追跡結果例を図4に示す。

表1 指先追跡手法の比較

追跡手法	追跡成功率 [%]					合計	処理時間 [sec]
	R1	R2	R3	R4	R5		
MST	55.0	29.9	1.7	0.93	6.3	18.8	14.7
離散的DPT	43.4	98.9	97.3	99.0	92.4	86.2	50.1
提案手法	82.5	96.9	97.4	98.4	96.8	94.4	56.3

(CPU: Intel®CORE™i7-3.40GHz, メモリ: 16GB)

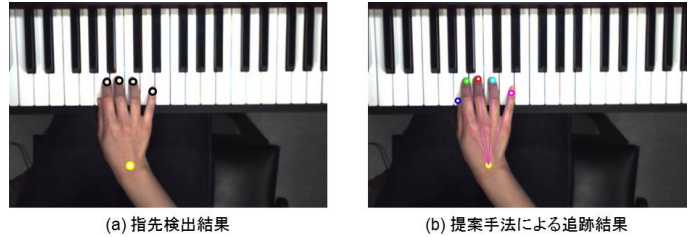


図4 候補指先位置と提案手法による指先追跡結果

次に、運指認識性能を実験により示す。運指認識成功の定義は、認識結果の運指と実際の演奏で用いた運指が一致した割合とした。今回使用した演奏データに対して、提案手法の認識成功率は100.0%(64/64)であった。また、運指判

表2 運指認識成功率と処理速度

ベースの追跡手法	認識成功率 [%]	処理時間 [sec]
MST	40.6	14.7
離散的DPT	98.4	50.1
提案手法	100.0	56.3

(CPU: Intel®CORE™i7-2.93GHz, メモリ: 8GB)

定に関する処理自体はテーブルの参照と小規模な多数決演算処理のみで構成されていることから、処理時間はほぼ追跡処理と同等であった。

## 6. おわりに

高精度な指先追跡が可能な同時多点追跡を運指認識に適用することで、指先のセルフオクルージョンに頑健な運指認識手法を実現した。運指認識成功率100.0%(64/64)、処理時間56.3secを実現した。

## 参考文献

- [1] 子安大士ら, “ピアノ演奏動作解析のための3次元手指追跡,” 情報科学フォーラム2011, pp.171-172(2011).
- [2] 有賀治樹ら, “連続デプス画像解析に基づく仮説検証型ピアノ運指認識手法,” 情報処理学会研究報告, Vol.2013-MUS-100, No.11, pp.1-6(2013).
- [3] 藤村一行ら, “大局的最適化に基づくトラッキング-DPトラッキング,” 信学論(D), Vol.J92-D, No.8, pp.1279-1288(2009).
- [4] D. Comaniciu, et al., “Real-Time Tracking of Non-Rigid Object using Mean Shift,” Computer Vision and Pattern Recognition(CVPR), Vol.2, No6, pp.142-149(2000).