

高速手指動作認識による 携帯端末向け多指 AR タイピングインタフェース

樋口 政和^{1,a)} 小室 孝^{1,b)}

概要: 本稿では, 携帯端末の背面に設置されたカメラからユーザの手指を含む実画像を取得し, オプティカルフローにより手指の動きを認識することで, 実画像上に重畳した仮想キーボードで多指による空中タイピングを可能にするインタフェースを提案する. このインタフェースを備えたシステムでは, ユーザの視線方向とカメラの向きが同じであり, キーボード上にユーザ自身の手が重畳されるので, 実際にキーボードがそこにあるかのような感覚でタイピングが可能である. また, 手や指の姿勢認識は行わず, その動きのみの認識を行うので, 複雑な処理を必要とせず高速で安定した動作が見込まれる. PC ベースの実験システムでは, 1 フレーム当たり約 8.33 [ms](フレームレートに換算して約 120 [fps]) で手指の動き認識を実現した. 提案インタフェースの性能を評価するため, キーボードアプリケーションを実験システムに実装し, 多指による実時間空中タイピングを実現した.

Multi-finger AR Typing Interface for Mobile Devices Using High-Speed Hand Motion Recognition

HIGUCHI MASAKAZU^{1,a)} KOMURO TAKASHI^{1,b)}

Abstract: In this paper, we propose a new user interface designed to improve the usability of mobile devices. The interface overlays a virtual keyboard on real images captured by a camera attached to the back of a mobile device and recognize user's hand motions by using optical flow information. The user can operate the virtual keyboard with his/her multiple fingers in the space behind the mobile device. In the experimental system equipped with our interface, the user can perform multi-finger typing with his/her hand as if there were a real keyboard because the viewing directions of the camera and his/her eyes are the same and his/her hand is overlaid on the virtual keyboard. The interface recognizes only hand motions and does not recognize poses of a hand or fingers, which enables high-speed stable recognition without complicated processing. The interface recognized hand motions with a processing time of about 8.33 [ms/frame] (about 120 [fps]). We implemented a keyboard application using our interface on the experimental system and evaluated the performance. In the application, we realized real-time multi-finger typing in the air.

1. はじめに

近年, タッチパネルを搭載したスマートフォンなどの携帯端末が広く普及し, 我々の日常に欠かすことのできないものとなっている. スマートフォンの利点は, 文書の編集など, 今まで PC で主に行われてきた作業を携帯端末でも行えることである. しかし, スマートフォンはその携帯性のために画面が小さく, 狭い画面での操作のしづらさが利

便性を損ねている. この問題は特に, 文字入力において顕著であり, その改善のために, 特別な機器を装着しての入力やキーボードを平面上に投影して入力を行うなどの研究が行われている [1], [2]. しかし, 特殊な入力方式による操作訓練の必要性, 機器を身に着ける煩わしさ, キーボード投影のための場所の確保などの問題が挙げられる.

一方, 空中で指の動きを認識し, 特別な機器無しで入力操作を可能とするシステムが開発されている [3]. このシステムでは, 特別な装置を装着することなく, 空中で携帯端末の操作を行う事ができる. しかし, 操作がカーソルを通じ

¹ 埼玉大学大学院理工学研究科 埼玉県さいたま市桜区下大久保 255
^{a)} mhiguchi@mail.saitama-u.ac.jp
^{b)} komuro@mail.saitama-u.ac.jp

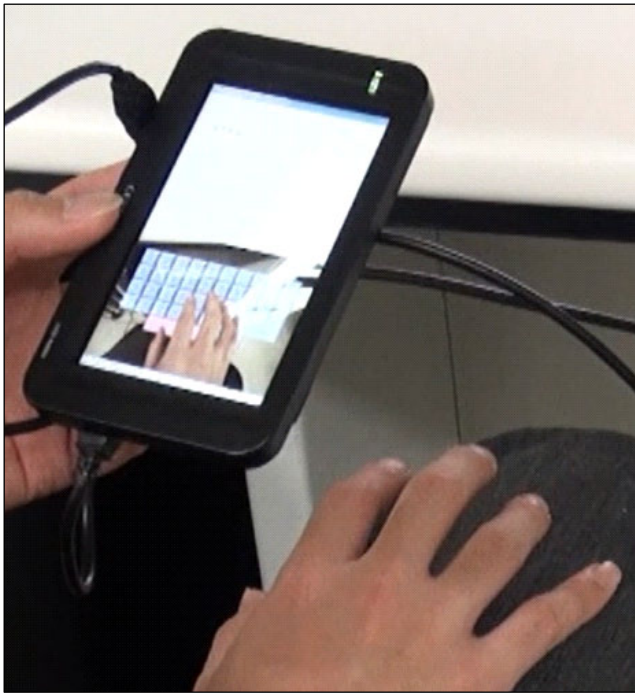


図 1 多指タイピングインタフェースシステムの操作の様子
Fig. 1 Multi-finger typing interface system.

た間接的なものであるため直感的な入力操作が難しいことや、一本指にしか対応していないことが問題点として挙げられる。

この問題を解決するため、携帯端末の背面空間において仮想キーボードからユーザ自身の手指を使ってキー入力可能なシステムが開発された [4]。このシステムでは、携帯端末の背面に設置されたカメラからユーザの手指を含んだ実画像を取得し、拡張現実感 (AR: Augmented Reality) の技術を用いて仮想キーボードとユーザ自身の手指を実画像上に重畳する。そして、オプティカルフローにより手指の動きを認識し、仮想キーボードで多指による空中タイピングを可能にしている。図 1 にこのシステムの操作の様子を示す。ユーザの視線方向とカメラの向きは同じであり、仮想キーボード上にユーザ自身の手が重畳されるので、実際にキーボードがそこにあるかのような感覚でタイピングが可能である。また、手や指の姿勢認識を行わず、その動きのみの認識を行うので、手指姿勢認識の失敗による動作不安定化の問題は回避され、安定した動作が見込まれる。加えて、一本指ではなく、多指での操作が可能であるため、より高速なキー入力が行える。原理上、手を動かすことのできる領域があれば、空中に限らず机上や膝上でも使用が可能である。この利点はユーザーのキー入力の際の疲労を減らすことに繋がる。しかしながら、このシステムでは手指領域全体に対してオプティカルフローを計算しているため、手指の動き認識に時間が掛かっていた。そのため、端末画面上でユーザの手指の動きに遅れが発生し、快適なキータイピングが実現できないという問題があった。

そこで、本稿では [4] で提案されているアプローチを踏襲しつつ、システムが高速で動作するように手指の動き認識手法の大幅な改善を図った。その結果、1 フレーム当たり約 8.33 [ms] の時間 (フレームレートに換算して約 120 [fps]) で手指の動き認識を実現した。また、提案手法の性能を評価するため、キーボードアプリケーションを PC ベースの実験システムに実装した。そのアプリケーションにおいて多指による実時間空中タイピングを実現し、提案手法の有用性を確認した。提案手法を利用することで、端末上でのユーザの手指の動きの遅れが解消され、快適なキータイピングが可能となった。

2. 関連研究

携帯端末の狭い画面上での操作性を向上させる研究はいくつか行われている。[5] では、携帯端末の画面上での操作を工夫することで、操作性の向上を図っている。この手法では、携帯端末の画面上で指を動かす方向に応じて入力を行うことができ、従来のスマートフォン向けの入力方式で起こる目的のキーの隣を誤って押すというリスクを減らすことができる。しかし、特殊な操作方法であるため、入力にはある程度の訓練が必要であることが問題として挙げられる。

操作領域を実空間に移すことで、携帯端末表面上の操作領域の狭さに関する問題を回避している研究もある。[1] では、ユーザの手や手に持った紙面などに仮想物体を重畳し、ユーザの手指でそれらを操作できる Omnitouch が提案されている。しかし、ユーザの肩に小型プロジェクタと深度カメラを装着する必要があるため、これは現実的ではない。[2] では、赤外線照射モジュールを利用して、任意の平面にキーボードを表示しその平面上で文字入力を可能にする Canesta Keyboard が提案されている。しかし、赤外線モジュールを設置する場所やキーボードを投影するための平面が必要であるという制約がある。

それらに対し、特別な外部機器を必要とせず、携帯端末に搭載されたカメラを利用して指認識を行うことで、その機器を空中で操作可能にするユーザインタフェースシステムが開発されている [3]。このシステムでは、外部機器等の煩わしさは解消され、ユーザへの情報提示は携帯端末のディスプレイに行われるため、使用環境も制限されないが、機器前面に取り付けられたカメラを使用するため、カメラ画像に蛍光灯やユーザの顔などが写り込み指認識が困難になる。また、機器操作は機器ディスプレイ上のカーソルを通して行われるので、操作が直感的でない。加えて、入力は一本指にしか対応していない。

携帯端末の正面ではなく背面のカメラを用いたインタフェースの研究も行われている。[6] では 3 次元的な仮想物体に携帯端末を通して触ることができるシステムが提案されており、その著者らはこのシステムは携帯端末の操作領域を拡張し、仮想物体との直感的なインタラクションも可

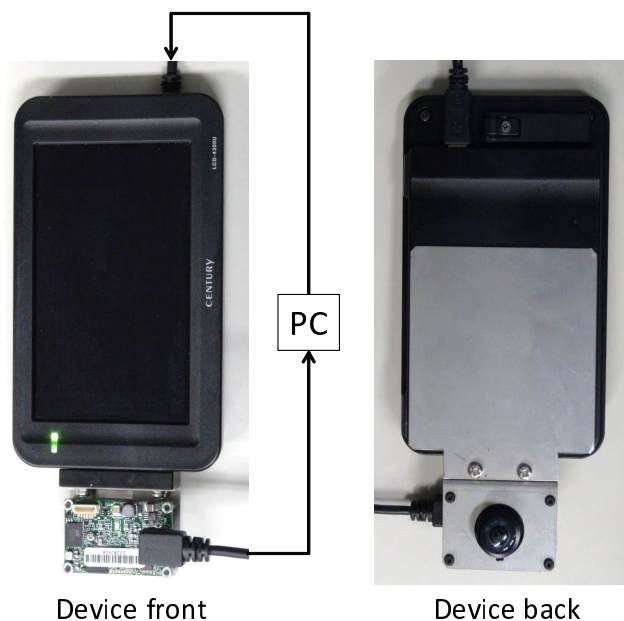


図 2 実験システムの外観
Fig. 2 Photographs of the experimental system.

能にする、と述べている。[7]では、特別な装置を装着せずに多指の指先を認識することができ、実画像上に重畳表示した仮想物体に触れることもできる。[8]では、マーカーを用いることで、1, 2本の指で様々な3次元操作を実現しているが、この種の空中操作は娯楽用途のみにしか活用できない、とその著者らは結論付けている。これらの研究では、拡張現実感 (Augmented Reality: AR) の技術を用いることで、仮想物体の直感的な操作を実現している。しかし、仮想物体とのインタラクションが主な目的となっており、手の高速動作に対応していないなどの問題点もあり、キータイピングに適した手法とは言えない。

3. 多指 AR タイピングインタフェース

3.1 ハードウェア構成

本稿は提案インタフェースの操作性の評価が目的であるため、実際の携帯端末の代わりにPCベースの実験システムを用いた。実験システムは、4.3インチの小型ディスプレイ、小型単眼カメラ、PCから構成される。小型ディスプレイはスマートフォンなどの携帯端末と同程度の大きさのものを採用した。カメラはPoint Grey Research社のFirefly MVを用い、小型ディスプレイの背面に取り付けた。レンズの焦点距離は1.68mmであり、カメラのフレームサイズとフレームレートはそれぞれ320×240pixel, 112fpsである。PCはIntel(R) Core(TM) i7-3770 (8M Cache, 3.40 GHz) CPUと4.00 GB RAMを搭載したマシンを用いた。図2に実験システムの外観を示す。カメラで取得した画像はPCで処理され、実画像上に仮想キーボードとユーザの手指を重畳した結果がアプリケーションと共に小型ディスプレイ

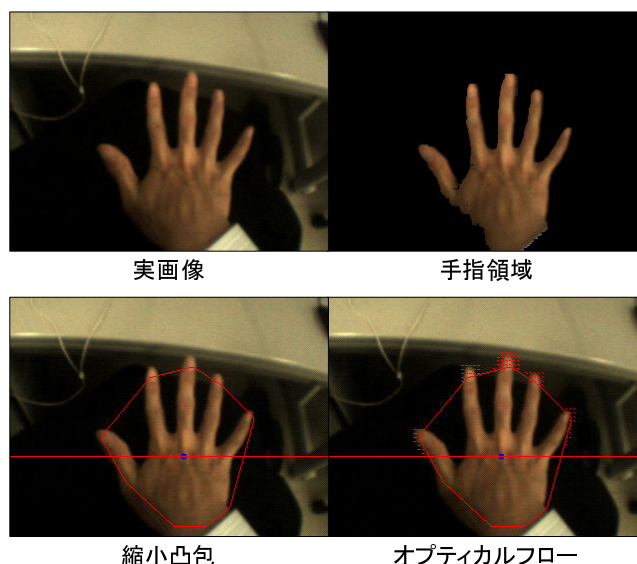


図 3 オプティカルフロー取得までの処理
Fig. 3 The calculation of optical flow.

に提示される。

3.2 オプティカルフローを用いた手指動作の認識

3.2.1 オプティカルフローの取得

まず、HSV色空間を利用して肌色領域を取得することで、カメラ画像から手指領域を取得する。得られた手指領域に対して凸包を計算し、一定の割合で縮小させた凸包内部からはみ出した手指領域で、かつ手指領域の重心より上の領域に対してオプティカルフローを計算する。これにより、指先の領域に対してのみオプティカルフローが計算されることになるので、手指領域全体に対して計算した場合に比べて計算時間は飛躍的に短縮される。一連の処理を図3に示す。オプティカルフローの取得には、Pyramidal Lucas-Kanade法[9]を用いた。これは、他の手法と比べ計算時間が速く、動きの強さに応じたフローがある程度正確に求まり、手の動きを検出しやすいためである。今回、オプティカルフローを計算する点の間隔は4pixelとし、フローの計算に用いるブロックサイズは30×30とした。フロー点を計算する点の間隔を狭くすると計算時間は増え、広くすると計算時間は短縮されるが得られる情報量は少なくなる。また、ブロックサイズを小さくすると、計算時間は短縮されるが、フローの精度は低下する。逆に大きくすると、フローの精度は向上するが、計算時間が増える。

3.2.2 手指動作の認識

手指動作認識のフローチャートを図4に示す。これにより、指の押下動作が認識できる。

まず、オプティカルフローの強度情報を閾値処理することによって動き領域を検出する。ある指がほかの指と比べて大きく動いた場合、その周辺に大きなオプティカルフローが集中する。従って、一定以上の大きさを持つフロー

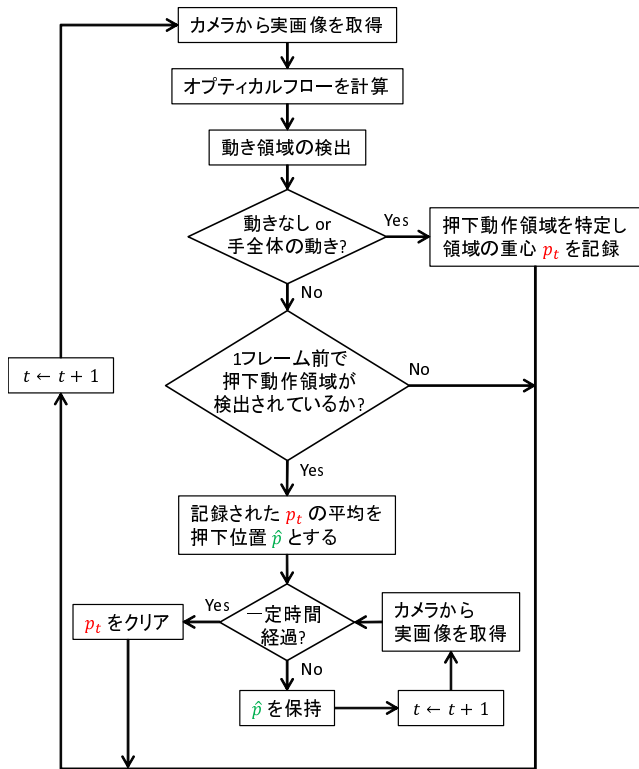


図 4 手指動作認識のフローチャート

Fig. 4 The flowchart of the hand action recognition.

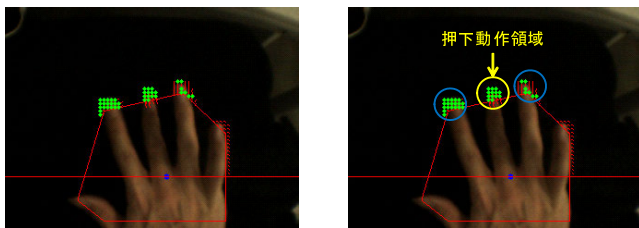


図 5 動き領域とラベリング

Fig. 5 Motion regions and labeling.

点を検出することで手指のどの部位が動いたかが検出できる。手指が動いていないときは動き領域は検出されない。また、手全体が動いているときはほとんどすべてのフロー点が動き領域として検出される。この場合、押下動作の認識は行わないことにする。

動き領域が 1 フレーム内で複数個検出される場合がある。その様子を図 5 の左に示す。この図は、中指を押下したときに同時に左右の人差し指と薬指の領域にも動きが検出された例である。このようなときは、図 5 の右に示されるように、ラベリングにより領域分割を行い、分割された各領域のフロー強度の平均を計算し、強度平均が最大の領域を押下動作領域とする。押下動作を行っている指付近に大きなフローが集中するので、この例では中指の領域が押下動作領域として特定される。押下動作領域が特定されたら、その領域の重心 $p_t = (x_t, y_t)$ を記録する。同時に、押下動作領域のフロー強度平均も記録する。

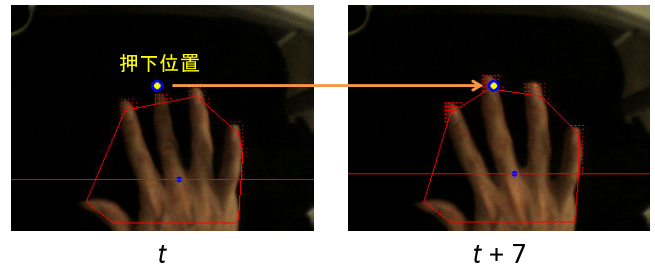


図 6 \hat{p} が数フレームの間保持されている様子

Fig. 6 \hat{p} is kept in sequential frames.

押下動作領域が検出されないときは、1 フレーム前で押下動作領域が検出されているかをチェックし、検出されていたら記録された p_t の点列の平均を押下位置 \hat{p} とする。また、記録されたフロー強度平均の平均を \hat{p} における押下強度とする。 \hat{p} が決定されたら、一定時間その位置を保持し、その後記録されていた p_t のクリアを行う。

指が押下されたとき、指が動いている最中は p_t が記録され、押下中心で一瞬動きが止まるので、そこで \hat{p} が決定される。そして、その位置を一定時間保持することで、押下を終えた指が元の位置に戻るまでの動作を無視できる。このようにすることで、押下位置が振動することを防ぐことができる。1 フレーム毎に押下位置を検出する方法では、キーボードアプリケーション側で押下位置が振動し所望のキーを押すことが困難となる。また、押下を終えた指が元の位置に戻るまでの動作で検出されることで同じキーが連続で押されてしまうことを防ぐこともできる。 \hat{p} が決定され指が元の位置に戻るまでの間、その位置が保持されている様子を図 6 に示す。

[4] で用いられている手指動作認識手法は、オプティカルフローを手指領域の左上から走査し、最初に一定以上の大きさのフローを持つ点を検出している。そして、その点を基準点とし、そこから探索範囲を定め、その探索範囲内で最も大きなフローを持つ点を検出し、その点のフローの大きさが予め設定した閾値よりも大きければ、その点の付近で押下が発生したと認識する。しかし、この手法では手が傾いているときには対応できない。また、静止状態のとき押下動作が検出されることがあるなどの問題もあり、押下動作の認識率が低くなっていた。今回、改善された手法では、手が傾いた状態にも対応でき、また静止状態では押下動作は検出されない。

4. AR キーボード

4.1 実装

提案した手指動作認識手法を利用して、キーボードアプリケーションを開発した。スナップショットを図 7 の左に示す。このアプリケーションでは AR 技術によりカメラ画像に仮想キーボードを重畳し、その上にユーザの手を重畳

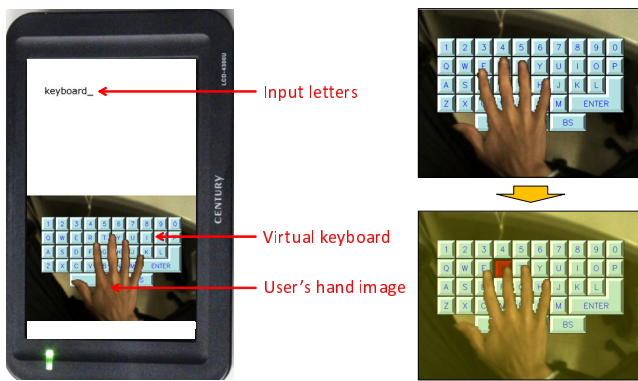


図 7 キーボードアプリケーションとキー押下の様子

Fig. 7 Screenshot of our keyboard application and key pressing in the application.

する。これにより、実際にキーボードが手元にあるような感覚でキー入力を行うことができる。手指動作認識により押下動作が検出されたとき、その動作発生位置にあるキーが押されたと見なす。空中に限らず、机の上や膝の上でも操作を行うことが可能であり、携帯端末の背面のあらゆる空間で入力を行うことができる。キーが押された時は、そのキーは赤く表示され、押下強度に応じてカメラ光軸に沿って奥に押し込まれ、一定時間経過後、元の位置に戻る。仮想キーボードは3次元空間上に配置され透視投影により2次元のディスプレイ面に表示されており、実際の物理キーを押したときと同じ視覚効果がユーザに提示される。キーが押されている間は画面全体を光らせている。あるキーを押下している様子を図7の右に示す。仮想キーボードのキー配置は一般的な物理キーボードに用いられる QWERTY キーボードの配置を用いている。この配置にした理由は、ユーザが普段使い慣れているキーボードに近い配置であれば、特別な訓練も必要とせず、直感的な入力操作を行うことができると考えられるからである。このアプリケーションを用いて、連続で文字入力を行っている様子を図8に示す。図の左から順に、親指、人差し指、中指、小指、薬指を用いた文字入力を行っており、多指でのキータイピングが可能となっていることが分かる。

4.2 操作性の評価

提案インタフェースの性能評価のための第一ステップとして小規模な動作テストを行った。このテストは、“the quick brown fox jumps over the lazy dog” という文章の入力を一定回数だけ繰り返すというものである。この文章は英語のパングラムの一つであり、コンピュータのキーボードの試験などによく用いられることから、この文章を採用した。1回の実験で入力に掛かった時間を計測した。なお、間違ったキーを押した場合は文字入力はされないようにした。これにより、修正操作の必要はなくなり、ミスタイプに依存せずに入力時間を評価できる。今回のテストは一

人の被験者で行った。5回の実験の結果、試行回数を重ねるごとにタイピング時間は短くなることが分かった。これは、本インタフェースの操作に十分慣れることで高速タイピングが可能になることを示唆している。なお、5回とも入力操作は空中で行った。動作テストから得られたタイピング時間の平均を1分あたりの入力単語数 (Words Per Minute: WPM)[10] に換算したところ、9.23[word/min]であった。本インタフェースはまだ従来システムと比較できる段階ではないが、参考のため一般に広く普及している物理キーボードとスマートフォンなどに搭載されているソフトキーボードの WPM を述べると、前者は 59.27[word/min]、後者は 21.79[word/min] である [11]。両手 10 本指を用いる物理キーボード並みの WPM を本インタフェースを搭載したシステムで達成することは困難であるが、1 本指で使用するの多いソフトキーボードの WPM に関しては、本インタフェースの多指操作という利点を活かせば、それを超えることも可能であると考えられる。

また、操作領域は空中なので端末カメラからユーザの手までの距離を変化させ、ユーザの好みに合わせて手の大きさを仮想キーボードをタイピングするのに適した大きさに調整することができるという点も本インタフェースの利点である。

5. まとめ

本稿では、携帯端末の背面に設置されたカメラからユーザの手指を含む実画像を取得し、オプティカルフローにより手指の動きを認識することで、実画像上に重畳した仮想キーボードで多指による空中タイピングを可能にするインタフェースを提案した。このインタフェースを備えたシステムでは、狭い操作空間に制約されることなく仮想キーボード上でタイピングを行うことができ、キーボード上にユーザ自身の手が重畳されるので、実際にキーボードがそこにあるかのような感覚でタイピングを可能とした。手指動作認識手法は従来のアプローチを踏襲しつつ、システムが実時間で動作するように手指の動き認識手法の大幅な改善を図った。その結果、認識の速度と精度の問題を解決し、1フレーム当たり約 8.33 [ms] の時間 (フレームレートに換算して約 120 [fps]) で手指の動き認識を実現した。そして、提案手法の性能を評価するため、キーボードアプリケーションを実験システムに実装した。そのアプリケーションにおいて多指による実時間空中タイピングを実現し、提案手法の有用性を確認した。提案手法を利用することで、端末上でのユーザの手指の動きの遅れは解消され、快適なキータイピングが可能となった。

今後の課題としては、押下動作認識の精度向上が挙げられる。現在の手法では、やや強めに押下動作を行わないと、動作を認識できない。この問題は、動き領域検出時の閾値を調整することで対応可能であるが、判定を緩めにする

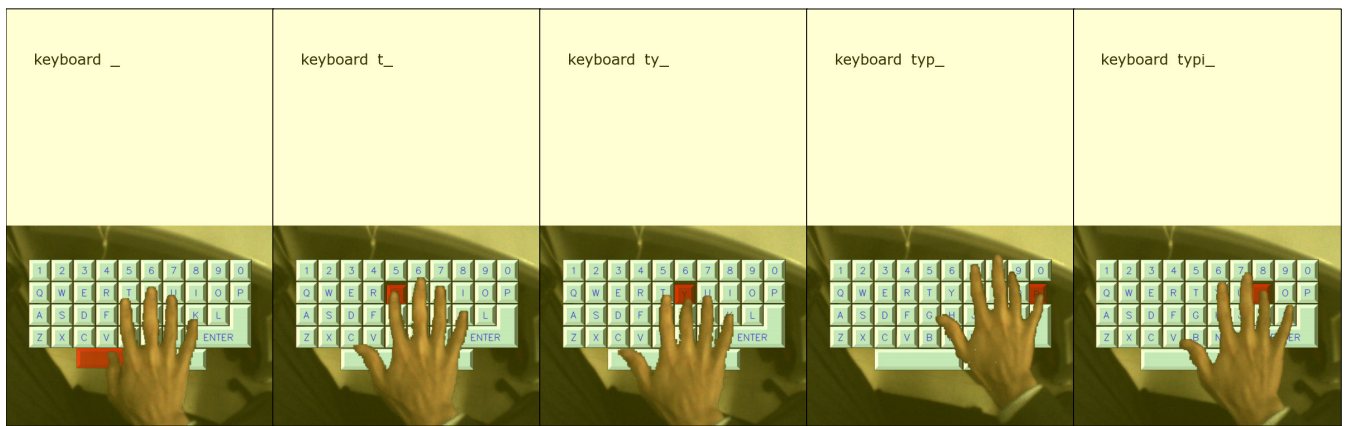


図 8 キーボードアプリケーションでの連続キータイピングの様子

Fig. 8 Sequential key typing using our keyboard application.

指の僅かな動きでも押下動作と認識してしまい、この調整は現段階では難しい。また、手の並進動作の際に押下動作と判定されてしまうことが比較的多いので、この点も改善を図る必要がある。これらの問題を解消し、より快適で高速な入力が行えるインタフェースの開発を目指す。

参考文献

- [1] C. Harrison, H. Benko, A. D. Wilson, Omnitouch: Wearable multitouch interaction everywhere, *Proc. the 24th annual ACM symposium on User interface software and technology*, pp.441–450 (2011).
- [2] H. Roeber, J. Bacus and C. Tomasi, Typing in thin air: The canesta projection keyboard – a new method of interaction with electronic devices, *Ext. Abstracts CHI2003*, pp.712–713 (2003).
- [3] T. Niikura, Y. Hirobe, A. Cassinelli, Y. Watanabe, T. Komuro, and M. Ishikawa, In-air Typing Interface for Mobile Devices with Vibration Feedback, *ACM SIGGRAPH 2010 Emerging Technologies Article*, 15 (2010).
- [4] S. Sagara, M. Higuchi, T. Komuro, Multi-Finger AR Typing Interface for Mobile Devices, *Companion Publication of the 19th International Conference on Intelligent User Interfaces (IUI 2014)*, pp.13–16 (2014).
- [5] M. Klima and P. Slavik, Vector Keyboard for Android Platform-Based Devices, *Proceedings of the 2011 international conference on Human interface and the management of information*, Volume Part I, pp.435–442 (2011).
- [6] M. L. Caballero, T. R. Chang, M. Menendez and V. Occhialini, Behand: augmented virtuality gestural interaction for mobile phones, *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pp.451–454 (2010).
- [7] M. Baldauf, S. Zambanini, P. Frohlich, and P. Reichl, Markerless Visual Fingertip Detection for Natural Mobile Device Interaction, *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp.539–544 (2011).
- [8] W. Hurst and C. V. Wezel, Gesture-based interaction via finger tracking for mobile augmented reality, *Multimedia Tools and Applications*, Vol.62, No.1, pp.233–258 (2013).
- [9] J. Y. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature Tracker, Intel Corporation, Microprocessor Research Labs (2000).
- [10] A. S. Arif and W. Stuerzlinger, Analysis of Text Entry Performance Metrics, *Science and Technology for Humanity*, pp.100–105 (2009).
- [11] P. Bao, J. Pierce, S. Whittaker and S. Zhai, Smart Phone Use by Non-Mobile Business Users, *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp.445–454 (2011).