

タイピングの特性に基づいた文字入力中のコマンド入力方式

A Command Input Method while Texttyping
based on Typing Characteristics

片山 拓也 村尾 和哉 寺田 努 西尾 章治郎*

Summary. 現在, PC 用のキーボードは主に文字入力に用いられ, ファンクションキーやショートカットキーなどを活用することで数多くの機能を実現している. しかし, これらの入力によって高速な操作が可能となる一方, キーと機能の組み合わせを覚えなければならないという問題点がある. そこで本研究では, 従来の文字入力を行う「タイピング」に加えて, キーボードをなぞる「ストローク」およびキーボードのある形状で押す「スタンプ」の 3 種類のコマンドの入力をシームレスに行う方式を提案する. 提案手法はキーボードの入力特性から「タイピング」「ストローク」「スタンプ」を自動的に判定をするため, 特別な装置を用いることなく文字入力中のシームレスなコマンド入力を実現する.

1 はじめに

文字入力に用いられるキーボードは, コンピュータの入力デバイスとして不可欠なものになっており, ある程度習熟した利用者が手軽に特定の機能を実行するために, しばしばショートカットキーと呼ばれる, 利用頻度が高い機能を特定のキーの組合せ入力で行う入力方式が用いられる. しかし, 機能に対応するキーの記憶, ツール毎の使用コマンドの違いなどが存在し, コンピュータの習熟度が低い利用者は使いづらい. これまでにキーボードにおける入力を効率化するための様々な研究が行われ, 多種多様な文字入力デバイスが提案されているが, 現在のキーボードの形状は基本的に万国共通であり, 広く普及しているため, 新たな形状のデバイスを設計することは利用者への敷居が高い.

そこで本研究では, 通常の入力を行う「タイピング」に加えて, キーボードの上をなぞる「ストローク」, およびキーボードの範囲を押す「スタンプ」の 2 種類のコマンド入力によって直観的に特殊機能の入力を行う方式を提案し, 特別な装置を用いることなく, 既存のキーボードのみを用いて, 文字入力中にシームレスにコマンド入力が行える方式を構築する. 本研究ではシステムのプロトタイプを実装し, 利用者評価によって識別精度を調査した. さらに, コマンドと機能を関連付けるインタフェースを構築した.

以下, 2 章で構築したコマンド入力方式について説明し, 3 章で提案システムの評価について述べる. 最後に 4 章で本研究のまとめを行う.

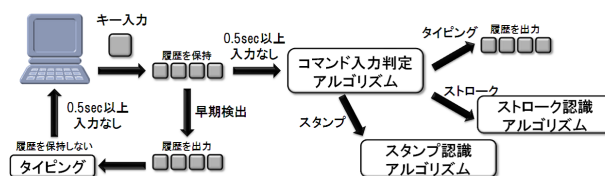


図 1. システムの動作フロー

2 コマンド入力方式の設計

2.1 システム構成

提案システムの動作フローを図 1 に示す. 提案システムでは, ストロークやスタンプが入力された際にキーの入力を無効化してコマンドに置き換える必要があるため, キーイベントが発生したらその履歴をシステムで保持する. そして, キーイベントが一定時間 (初期値は 0.5 秒に設定) 発生しなければ, その履歴がタイピングかコマンドかを判別し, 処理を行う. 一方, キーイベントをただ蓄積すると, キー入力を行ってもそれが画面に反映されないという利用者にとっての違和感を生じさせるため, タイピングに関しては可能な限り早期に検出する必要がある.

2.2 コマンド入力の判定方法

提案システムでは, 新たに特別な装置を用いることなく, キー入力の履歴のみからコマンド入力を識別する.

まず, 入力がアルファベットキーのみで構成される場合, キー入力の履歴がローマ字入力として成立している場合はタイピングと判定する.

次に, コマンド入力の識別のための予備実験として, タイピング, ストローク, スタンプの 3 種類の入力を 30 回ずつ行った場合のキー入力履歴を取り, 「時間範囲 t_{span} : 最初のキーダウンから最後のキー

Copyright is held by the author(s).

* Takuya Katayama, Kazuya Murao and Shojiro Nishio, 大阪大学大学院情報科学研究科, Tsutomu Terada, 神戸大学大学院工学研究科

表 1. タイピングの評価

被験者	打鍵数	誤認識回数	識別アルゴリズム 使用回数	タイムラグ 発生文字数
被験者 1	415	0	9	50
被験者 2	362	0	3	43
被験者 3	314	0	9	47
被験者 4	355	3	18	70
被験者 5	400	0	13	58
被験者 6	358	1	15	70
合計	2204	4	67	338

ダウンまでにかかった時間」「押下時間 t_{push} : 各キーの平均押下時間」「移動距離 c_{dist} : 次のキーまでの平均移動距離」「座標分散 c_{dev} : キー座標の分散」「打鍵数 n : 入力されたキーの数」を調べた。ここで、キー座標 (x,y) は $(0,0) \sim (75,25)$ の範囲であらかじめマッピングしている。

得られた特徴を考慮して、キーイベントが一定時間途切れた際に行う「コマンド入力を識別するアルゴリズム」と、入力と出力のタイムラグを解消するためにキーイベントが発生する毎に行う「タイピング早期検出」を設計した。

2.3 ストロークの認識

システムでストローク入力と判断されたキー入力の履歴は、DP マッチングを用いた文字列比較を改良した手法によって識別する。この手法では、文字列の不一致の原因として「挿入・削除」と「置き換え」の2種類を定義し、それぞれに適当なコストを定めて文字列同士の乖離度をスコア化する。一般的なキーボードのアルファベットキーは周囲のキーと6方向で接しているため、キーの移動方向を6種類の文字に置き換える。そして、挿入・削除のコストを1に設定し、置き換えのコストを軌跡のずれの角度の大きさにより2, 4, 6となるように設定した。

2.4 スタンプの認識

システムでスタンプ入力と判断された履歴は、キー座標の集合から特徴量を抽出して識別する。抽出する特徴量は、履歴のキーの数と、それらの X 座標、Y 座標それぞれの平均、分散、幅の7つである。あらかじめ特徴量と正解のスタンプの組をシステムに学習させておき、未知のスタンプを認識する際は、その未知のスタンプの文字集合の特徴量と学習した特徴量のユークリッド距離を計算し、k 近傍法を用いて入力されたスタンプを判断する。

3 評価

提案システムの評価として、20代の男性、女性の被験者6人から採取したデータを用いて入力の識別精度の解析を行った。

3.1 タイピング

被験者には自由に文章をタイピングしてもらったところ、結果は表1のようになった。

表1より、4回のコマンド入力の誤認識が発生した。4個の文字列のうち、正常なタイピングは前者の2個で、後者の2個はタイピングミスから発生したものである。また、タイピングの早期検出の条件に当てはまらず、コマンド入力の識別アルゴリズムを使用した回数は67回あるが、そのうち46回が一つのキーしかなかった場合であり、タイピングの早期検出は正常に動作したことが分かる。また、ほとんどが2, 3キーが押される間にタイピングの早期検出が発生し、タイムラグは0.3秒以内に抑えられており、利用者に不快感を与えることはなかった。

3.2 ストローク

被験者にはストローク入力として「丸」「四角」「三角」と各々が自由に定義したストロークコマンドの4種類を3回ずつシステムに学習させた後、それぞれのコマンドについて20回ずつ入力してもらってその精度を評価した。全体の認識精度は82.3%となり、高い精度で入力コマンドを区別することができた。

3.3 スタンプ

被験者にはスタンプ入力として、前述した「ゲー」「パー」「チョップ」の3種類のスタンプ入力をそれぞれ20回ずつ入力してもらい、その認識精度を評価した。全体の認識精度が48.3%と低い値になった。さらに、その不正解の履歴の多くがタイピングと誤認識されていた。この問題を解決するためには、キーイベントの履歴以外の要素を使うことが有効になる可能性がある。具体的には、打鍵圧の値などが考えられる。Expressive Typing[1]では、ノートパソコンが落下時にハードディスクを緊急停止させるために用いる内蔵型加速度センサの値を用いて打鍵圧を取得することに成功している。

4 おわりに

本研究では、2種類のコマンド入力を提案し、特別な装置を用いることなく、文字入力中にシームレスにコマンド入力が行える方式を構築した。今後の課題として、認識精度向上のための改良や、提案システムを活用したアプリケーションの実装などを行う予定である。

参考文献

- [1] K. Iwasaki, T. Miyaki, and J. Rekimoto: Expressive Typing: A New Way to Sense Typing Pressure and Its Applications, ACM CHI2009 work in progress, pp.4369-4374 (2009).