

# Presstick: 圧力とスティックで跳ばすポインティング手法

山内啓史<sup>†</sup> 西山卓志<sup>†</sup>  
赤池英夫<sup>††</sup> 角田博保<sup>††</sup>

家庭用ゲーム機のコントローラに標準搭載されている感圧ボタンとアナログスティックとを用いた新しいポインティング手法 Presstick を提案する。本手法によりユーザは自由な姿勢からポインティングを行うことができ、カーソル移動距離に依らず操作手数がほぼ一定で済む。評価実験を実施したところ、被験者 5 人による本手法の平均操作時間はマウス、タッチパッドに劣ったが、熟練した被験者（著者らの一人）の場合は一定距離以上でタッチパッドより操作時間が少ないという結果が得られた。

## Presstick: Pointing method with pressure and joystick control

KEISHI YAMAUCHI,<sup>†</sup> TAKUSHI NISHIYAMA,<sup>†</sup> HIDEO AKAIKE<sup>††</sup>  
and HIROYASU KAKUDA<sup>††</sup>

### 1. はじめに

プレゼンテーション時のような立った状態や椅子に踏ん返り返った状態、机が目の前にない状態など、マウスでポインティングを行うには不都合な場面がしばしば存在する。そのようなときは自由な振る舞いで扱えるようなポインティングデバイスが好ましいであろう。また、近年ではディスプレイの大画面・高解像度化が進展しており、従来のポインティング手法を用いる限り操作手数が増加し続けることになる。

そうした状況を踏まえ、本稿では家庭用ゲーム機のコントローラを用いたポインティング手法 Presstick (プレスティック) を提案する。近年のゲームコントローラには感圧ボタンとアナログスティックとが標準的に搭載されており、本手法はそれら 2 つの機能を利用するものである。本研究の前身として、圧力操作を特徴とするポインティング手法 Preshalor<sup>1)</sup> が存在する。Preshalor は感圧タッチパッドを用いるポインティング手法であり、タッチパッドに加える指の押下圧でカーソルのジャンプ距離を定め、その接地した指の重心移動をジャンプ方向とするものである。本研究は、

それらの操作をゲームコントローラに応用することにより、長距離ポインティングにおける操作手数の低減、およびゲームコントローラのポインティングデバイスとしての実用性を確立することを目標としている。各社のゲームコントローラは既に広く普及しており、それらがそのまま実用的なポインティングデバイスになるということは少なからず有益であると言える。

以下、関連研究の紹介、提案手法の説明、評価実験の手順と結果、そして改良案と述べていく。

### 2. 関連研究

空中で操作するポインティングデバイスとして、加速度センサでカーソル移動を行う空中マウス<sup>☆</sup>や、親指をタッチパッド上で滑らせて操作するリモコン<sup>☆☆</sup>などが製品化されている。

デバイスを指し示した方向に直接カーソルが移動するタイプのもとして、任天堂 Wii<sup>☆☆☆</sup>のポインティングシステムや Direct Pointer<sup>2)</sup> などが存在する。これらは直感的で学習が少なく済むが<sup>3)</sup>、手や腕の運動が大きい<sup>4)</sup>ため疲労の度合いが高い<sup>2)</sup>。また、数 pixel 単位の細かいポインティングに向いていないため、オブジェクトを大きめにするなどの UI 設計が求められる。

<sup>†</sup> 電気通信大学大学院 情報工学専攻

Department of Computer Science, Graduate school of Electro-Communications, The University of Electro-Communications

<sup>††</sup> 電気通信大学 情報工学科

Department of Computer Science, The University of Electro-Communications

<sup>☆</sup> 例えば、<http://buffalo-kokuyo.jp/products/catalog/supply/bomu-w24a02/>

<sup>☆☆</sup> <http://www.sharp.co.jp/i-aquos/products/ax120s/remocon.html>

<sup>☆☆☆</sup> <http://www.nintendo.co.jp/wii/>

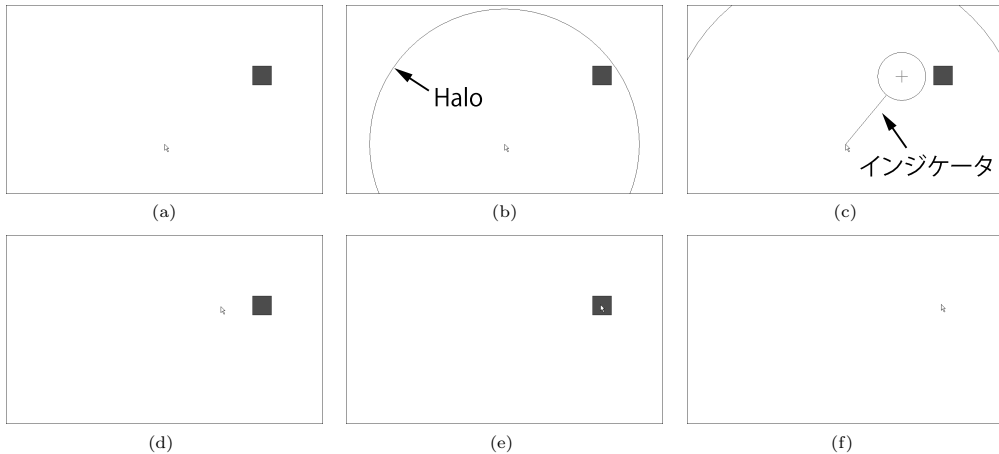


図 1 DUALSHOCK3 による Presstick の操作例 (a) カーソルの初期状態 (b)L2 ボタン押下で Halo 発生 (c) 左スティック入力でインジケータ出現, インジケータの円をターゲット近傍に移動 (d) O ボタン押下でカーソルがインジケータの+の位置にジャンプ (e)L2 ボタン, 左スティックそれぞれの入力を一旦解除した後, 左スティックでカーソル位置を微調整 (f) O ボタンでターゲットをクリック

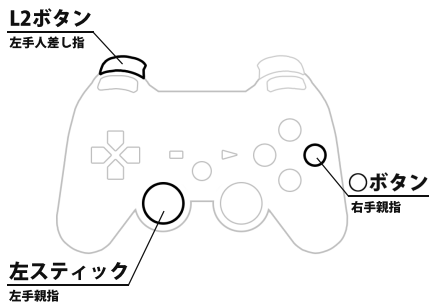


図 2 DUALSHOCK3 で使用する各機能および指の配置

### 3. 提案手法 Presstick

今回, 使用するゲームコントローラとして Sony Computer Entertainment の DUALSHOCK3<sup>☆</sup>を選択した. DUALSHOCK3 は 12 個の感圧ボタン, 2 本のアナログスティック, その他に加速度センサや角速度センサ, Bluetooth 通信機能などを装備する. 感圧ボタンはスイッチに感圧導電ゴムが使用されており, その抵抗値の変化を 256 段階の圧力値として取得可能である. アナログスティックの入力方向は X, Y 方向それぞれ 127~128 の範囲で検知できる.

DUALSHOCK3 による Presstick の操作例を図 1 に, DUALSHOCK3 で使用する各機能およびそれら

の指の配置を図 2 に示す. 感圧ボタンとして L2 ボタン, アナログスティックとして左スティック, マウスクリック用ボタンとして O ボタンをそれぞれ使用する. Presstick のポインティング戦略は基本的に次の 2 段階から成る.

**ジャンプモード** Halo (図 1-b の円弧) とインジケータ (図 1-c の, 中心に+を含む円およびそれとカーソルとを結ぶ直線) とを操作してターゲット近傍にカーソルをジャンプさせる.

**微調整モード** ジャンプ後, カーソル位置を微調整してターゲット上に乗せる.

以下, ジャンプモード, 微調整モードについて詳細を説明する.

#### 3.1 ジャンプモード

初期状態 (図 1-a) から L2 ボタンを押下するとカーソルを中心に Halo が発生し (図 1-b), ジャンプモードとなる. この Halo はカーソルのジャンプ可能範囲 (インジケータの可動範囲) を示すもので, L2 ボタンに加える圧力値  $P$  が高いほど Halo の半径  $R$  は大きくなる. Halo が出ている状態で左スティックを傾けるとその方向にインジケータが現れ (図 1-c), O ボタンを押下するとインジケータのジャンプ点 (+) にカーソルがジャンプする (図 1-d). インジケータは左スティックの任意の方角への任意の傾きに応じて, Halo 内で自由に移動できる. インジケータの操作中も Halo の大きさは調整可能である. インジケータの円の半径は 64px 固定で, ユーザはターゲットをこの円の中に入れ

<sup>☆</sup> <http://www.jp.playstation.com/peripheral/ps3/>

るように Halo とインジケータとを操作することになる。Halo の半径  $R$ ，カーソル位置からジャンプ点までの距離  $J$  はそれぞれ次式 (1)，(2) で算出される。 $d_w$ ， $d_h$  は画面の横，縦の pixel 数， $s_x$ ， $s_y$  は左スティック入力 of X，Y 成分である ( $-127 \leq s_x, s_y \leq 128$ )。

$$R = \sqrt{d_w^2 + d_h^2} \left( 0.1 + 0.9 * \frac{P}{255} \right) \quad (1)$$

$$J = R * \frac{\sqrt{s_x^2 + s_y^2}}{127} \quad (2)$$

( $J > R$  となるときは  $J = R$  とする)

アナログスティックの分解能は X，Y 方向ともに 256 段階しかないため，Halo の半径  $R$  が大きくなるほどインジケータの動作は粗くなってしまふ。 $R$  は必要最小限に抑えることが望ましい。

前述の Preshalor の場合，Halo はジャンプ距離を示すものであったが，圧力操作のみでジャンプ距離を正確に定めるのはやや難があると (私見で) 判断した。そこで Presstick ではジャンプ距離，方向をアナログスティックで制御することにより操作の可用性を高めた。また，圧力の加減およびスティック制御のみでカーソルのジャンプが完結するため，ジャンプ距離に依らず操作手数はほぼ一定で済む。ただし，インジケータのジャンプ点をターゲット上に正確に乗せるのは困難であるため，まずジャンプモードでカーソルをターゲット近傍にジャンプさせ，その後のカーソル移動は次項で述べる微調整モードで行うこととした。

### 3.2 微調整モード

ジャンプモードによるジャンプ後，入力中であった L2 ボタン，左スティックを一旦解除すると微調整モードに切り替わる。微調整モードでは左スティック入力でのカーソル移動を行い (図 1-e)，○ボタン押下でマウスクリックを行う (図 1-f)。このとき，例えば左スティックを  $x=20$ ， $y=20$  に傾けるとカーソルは (ジャンプ点から)  $x=20$ ， $y=20$  の相対座標に移動し，左スティックをニュートラルに戻すとカーソルは元の位置に戻る。つまりカーソルの移動範囲は左スティックの入力範囲と同一であり，おおよそ 256px 四方の正方形<sup>\*</sup>で表される。そのためジャンプ後にカーソルとターゲットが互いにこの範囲内に入っていないと，カーソルがターゲットに届かないことになる。

## 4. 評価実験

提案手法と既存のマウス，タッチパッドとのポインティング性能の比較を行うため評価実験を実施した。被験者は本学の学生 5 人で，全員マウス，タッチパッドの操作に熟練しているほか，アナログスティックを

備えたゲームコントローラの操作経験がある。

### 4.1 実験環境

今回の実験はデスクトップ環境で行った。被験者にはディスプレイの前で椅子に座り，迅速かつ正確にタスクを遂行するよう指示した。

実験用マシンは Dell Precision 390 (Windows XP Professional)，ディスプレイは総画素数 2560 × 1600px の 30inch 液晶ディスプレイを使用し，各手法の入力デバイスとして光学式マウス (Dell MO56UO)，タッチパッド (Lenovo 31P9514)，DUALSHOCK3 それぞれを実験用マシンに USB 接続した。マウス，タッチパッドのカーソル移動速度は OS の設定画面より変更可能であり，被験者ごとに好みの速度に設定させた。

### 4.2 実験デザイン

実験システムは Java 言語で実装した。実験タスクは「画面上のターゲットまでカーソルを移動させてクリックする」を 1 トライアルとし，1 セッション当たり 272 トライアルを連続で行う。マウス，タッチパッドの実験ではそれぞれ 1 セッションずつ，Presstick については 4 セッションを実施した。ターゲットは 50px 四方の正方形とし，実験画面では現在のターゲットが赤色，その次のターゲットが灰色で表示される。ターゲットの出現位置はランダムであるが，偏りが出ないよう自動で調整される。画面の表示画素数は各手法とも 2560 × 1600px とした。

### 4.3 実験結果

Presstick の実験結果は第 4 セッションのデータを採用する。各手法ともフィッツの法則<sup>3)</sup>に従うものと仮定し，各データの  $ID$  (Index of Difficulty)<sup>\*\*</sup>を求めた。全被験者の  $ID$  から求めた回帰直線を各手法ごとに図 3 に示す。熟練した被験者 (著者らの一人) のデータも併載する。クリックミスのあった場合や Presstick のジャンプモードが 2 回以上行われた場合についてはミストライアルとし，データから除外している。

被験者 5 人による Presstick の平均操作時間はマウス，タッチパッドと比較して多くかかるという結果となった。しかし熟練者の場合では回帰の傾きが小さく，一定距離以上からはタッチパッドより操作時間が少なくなっている。5 人の Presstick の操作時間をモード別平均で見ると，ジャンプモードでは 1024ms，微調整モードでは 631ms 所要していた。平均エラー率<sup>\*\*\*</sup>は

<sup>\*</sup> 角が丸い正方形のイメージ

<sup>\*\*</sup> ポインティング距離を  $D$ ，ターゲットの幅を  $W$  とすると

$ID = \log_2 (D/W + 1)$

<sup>\*\*\*</sup> ミストライアル数 ÷ 全トライアル数

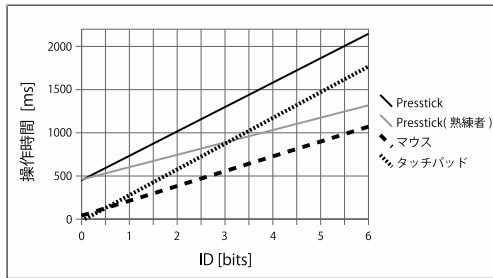


図3 各手法ごとのID-操作時間の回帰直線

マウスが8.2%、タッチパッドが7.7%、Presstickが17.5%（熟練者17.3%）で、Presstickのエラーの内訳は、ジャンプモードを2回以上行ってしまったエラー率（ジャンプエラー率）が9.1%、クリックミスによるエラー率（クリックエラー率）が8.4%であった。

事後アンケートより、インジケータがぶれることが多いという意見が全被験者から聞かれた。

#### 4.4 考察

実験結果から、インジケータのぶれや微調整モードの存在といった要因により、操作時間、エラー率が增大しているという問題点が明らかとなった。以下に改良案を挙げる。

##### ● Rの算出法の改良

Haloの半径Rが無意味に大きくなりすぎたり、小刻みに増減を繰り返したりするとインジケータの操作精度が悪くなるため、Rの算出法を見直す必要がある。今回用いた式(1)では係数を画面の対角pixel数で固定したが、例えばカーソルが画面中央に位置する場合、係数はその半分でも十分である。そこでこの係数を、カーソルの現在座標から画面端までの距離やインジケータの操作方向などに応じて動的に変化させることで、Haloの無駄な拡大を切り捨てることができると考えられる。また、式(1)では圧力Pの増減がRに対してリニアに反映されるが、他の文献によれば圧力をリニアに反映するとユーザは狙った圧力値に定めにくいという実験結果が示されている<sup>4)</sup>。その代替となる圧力制御手法としてPressureFish<sup>4)</sup>などが提案されているため参考にした。

##### ● インジケータ操作時のマウスクリックの実現

微調整モードは操作時間631ms、クリックエラー率8.4%と操作の手間を大きく増やす要因となっている。もしジャンプモードのインジケータ操作のみでマウスクリックが完結してしまえば、微調整モードに要していた時間が大幅に削減されることが見込まれる。例えばインジケータの円の部分をそのままBubble Cursor<sup>5)</sup>

にしてしまえばそのようなことも可能である。しかし、これを実現するには先のインジケータのぶれ問題を解決する必要がある。Rの算出法の改良によりある程度の緩和は期待されるが、最も確実な解決方法はターゲットへの吸い付きを実装することであろう。もちろん吸い付きアルゴリズムはオブジェクトの密集度に左右されるものであるが、DUALSHOCK3には今回使用していないボタンが多く備わっているため、例えば吸い付きのON/OFF切替ボタンを設ければ特に問題は無いと考える（あまり賢い方法ではないが）。

## 5. おわりに

本稿ではゲームコントローラの感圧ボタンとアナログスティックとを用いたポインティング手法Presstickを提案した。評価実験を実施して結果の解析を行い、得られた知見をもとに改良案について述べた。また、本手法に熟練した被験者による実験では、一定距離以上のポインティングにおいてタッチパッドより操作時間が少ないという結果が得られている。これより、本手法は学習を積むことで実用的なポインティング性能に達するものと考えられる。

今後はシステムの改良を行い、より現実的なタスクや様々な操作環境（リビングなど）を設定した上で実験を実施し、詳細な評価を行っていきたい。さらにGoogleマップなどのアプリケーションへの応用も検討する。

## 参考文献

- 1) 織田恵太, 赤池英夫, 角田博保: Preshalor: 圧力で跳ばずポインティングシステム, インタラクシオン2008論文集, pp.183-184 (2008).
- 2) Jiang, H., Ofek, E., Moraveji, N. and Shi, Y.: Direct Pointer: Direct Manipulation for Large-Display Interaction using Handheld Cameras, CHI 2006, ACM, pp.1107-1110 (2006).
- 3) Fitts, P. M.: The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, Journal of Experimental Psychology, Vol.47, pp.381-391(1954).
- 4) Shi, K., Irani, P., Subramanian, S. and Gustafson, S.: PressureFish: A Method to Improve Control of Discrete Pressure-based Input, CHI 2008, ACM, pp.1295-1298 (2008).
- 5) Grossman, T. and Balakrishnan, R.: The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area, CHI 2005, pp.281-290 (2005).