

# ターゲット予測インタフェースに関する実験的評価 An Experimental Evaluation of a Target Prediction Interface

古和田 一生

Motoki Kowada

法政大学情報科学部コンピュータ科学科

E-mail: 10k0017@stu.hosei.ac.jp

## Abstract

Sizes and resolutions of displays of personal computers are becoming larger, and it is desirable to shorten times for pointing tasks. Takashima et al. proposed Delphian Desktop, which performs the automatic mouse cursor jump toward a predicted position by detecting the peak velocity during pointing movement. They reported that Delphian Desktop was effective in shortening times for pointing to relatively distant targets. In their experiments, Delphian Desktop was compared with an environment that disabled mouse acceleration. However, most people ordinarily use mouse acceleration. This paper is focused on the experimental comparison of Delphian Desktop with mouse acceleration. An experimental environment for comparing Delphian Desktop and mouse acceleration was implemented by using the Java programming language and the Simple DirectMedia Layer library on top of the Windows operating system. In the experiments that were basically similar to Takashima et al.'s, subjects repeated operations of pointing to targets that appeared one after another. The times for pointing to targets and the rates of pointing failures were compared. The results of the experiments showed that mouse acceleration was more effective for pointing to targets of any distances than Delphian Desktop.

## 1. はじめに

近年、ディスプレイの大画面高解像度化にともない、グラフィカルユーザインタフェース(GUI)環境におけるポインティングタスク(図 1)に必要な時間が増加する傾向にある[1][2]。フィッツの法則によると、ポインティング時間はターゲット距離が長くなるほど増加し、ターゲットが大きくなるほど減少する[3][4]。そのため、ターゲットを拡大させたり[5]、単純にマウスカーソルをターゲットへジャンプさせたり[6]するといったポインティング時間の短縮を図った研究が行われている。

その中で、ポインティング中のマウスの速度から意図するターゲットを予測し、マウスカーソルをジャンプさせる手法を用いたインタフェース、Delphian Desktopが提案された[1][2]。この Delphian Desktop は比較的長いターゲット距離においてポインティング時間の短縮に有効であると報告されている。同研究の評価実験では、ターゲットまでのポインティング時間を比較する際、加速機能

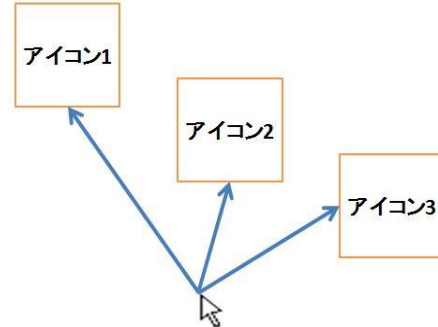


図 1. ポインティングタスク

のないマウスを比較対象として用いていた。しかし、我々が日常的に使用しているマウスは加速機能を有していることが多い。その意味では、同研究の実験結果は妥当であるものの、評価は必ずしも適当ではないと言える。

そこで本研究では、加速機能のあるマウスを比較対象として、純粋にポインティング時間だけに焦点を合わせて Delphian Desktop に関する評価実験を改めて行い、Delphian Desktop の実用性を再検討する。実験の結果、マウスの加速機能がポインティング時間の短縮とポインティング動作が正確さに対してより有効であることを確認した。

## 2. Delphian Desktop

以下に Delphian Desktop とその評価実験の概要を示す。

### 2.1. ピーク速度の抽出

ターゲットを予測するにあたって、マウスの移動速度のピーク値とターゲットまでの距離の間には線形の関係があり、これを式(1)のように 1 次式で近似する。

$$D = a \times V_p + b \quad (1)$$

ここで  $D$  はターゲットの距離、 $V_p$  はマウスカーソル移動時におけるピーク速度、 $a, b$  は被験者ごとに行うデータ取得実験によって得られた値を表す。ピーク速度取得の際、ポインティング時間がターゲットに対して 250 ms 以上となる速度を閾値として設定する。一度閾値を超えた速度を保持し続け、その後一度でも減速した場合、直前の速度をピーク速度とする。また予測の際の方向は、マウスカーソルのスタート位置からピーク速度時のカーソル位置までのベクトルの平均より推定する。

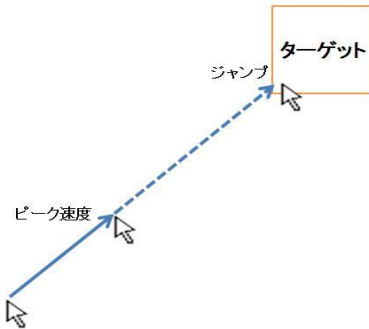


図 2. Delphian Desktop の動作

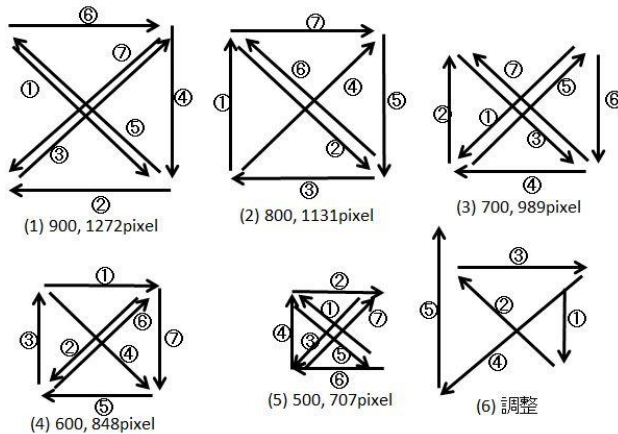


図 3. タスク条件[1]

## 2.2. Delphian Desktop の動作

ピーク速度を取得後、式(1)へそれぞれの値を代入し、求められたターゲット距離へ瞬時にマウスカーソルを移動させる。移動後は被験者の余分なマウス操作を防止するために、マウスカーソルを予測地点に 100 ms 間固定させる。以上の一連の動作を図 2 に示す。

## 2.3. 評価実験におけるタスク

Delphian Desktop の評価実験では、被験者に対してある規則に従って出現するターゲットアイコンを繰り返しポインティングさせたときの各ターゲット距離に対応したポインティング時間が比較された。ターゲットのサイズは一般的に利用されている  $50 \times 50$  pixel とし、各ターゲット距離は、上下左右 4 方向、5 距離条件(500, 600, 700, 800, 900 pixel), 斜め 4 方向、5 距離条件(707, 848, 989, 1131 pixel)である。この条件に従い、計 40 通りのタスク条件を  $1280 \times 1024$  pixel のディスプレイ内に 100 pixel 間隔に配置された候補から 1 つ選択し、表示する。ターゲットアイコンを正しくクリックするまでを 1 タスクとし、その後、直前にターゲットが出現していた箇所を新たなスタート位置として次のタスクを開始する。これは、被験者のカーソル位置の確認を容易にし、また不必要なマウス移動を避けるためであるとされる。計 40 通りのタスク条件を図 3 に示す。それぞれのターゲット距離に対して 7 タスクの実行をするため、各距離で 1 タスクずつ不足する。その不足を補う調整タスクを最後に実行する。また、被験者がタスク順序を予想できることを防ぐため

に、一巡目の 40 タスク条件を実行後、次の一巡は逆で行う。被験者は 120 タスクを練習として初めに行い、1 分間の休憩後 1 タスクあたり 8 回、正順、逆順で 2 回、計 640 回のタスクを行う。さらに、加速機能のない C-D 比 0.5 のマウスで Delphian Desktop 時と同様な実験を行う。C-D 比とはディスプレイ内のマウスカーソル移動距離に対するマウスの移動距離比と定義されている。被験者あたり計 1280 回のタスクを実行することになり、1 時間程度の実験時間を要する。

## 3. Delphian Desktop の再検討

Delphian Desktop では、被験者に、Delphian Desktop と加速機能のない C-D 比 0.5 のマウスを使用させて前述の評価実験を行っていた。しかし、加速機能がないという設定は必ずしも一般的とは考えられず、また C-D 比 0.5 でのマウスカーソルの移動速度も遅い。さらに、ポインティング動作を行う際に動的に C-D 比を変更することの有効性も報告されている[7]。また、Delphian Desktop は被験者のマウス移動時の初動に完全にしがたってマウスカーソルをジャンプさせるため、初動がある程度正確でなければ予測は失敗する。そのため、「被験者は誤操作を避けるためにマウス操作前に熟慮するようになるか、もしくは誤操作を恐れずターゲットへたどり着くまで投機的にワーブさせることを繰り返す戦略をとるようになる」という指摘もある[8]。以上を考慮すると Delphian Desktop の評価実験結果は妥当ではあるものの、評価は必ずしも適当でないと言える。

そこで本研究では、Delphian Desktop を実際に実装し、加速機能ありのマウスを比較対象として、被験者 6 人に対して再評価実験を行う。これは被験者が日常的に使用している環境に限りなく近づけるためである。C-D 比は事前に被験者へのアンケートを参考に 0.7 とした。本研究での評価実験では、タスク条件やタスク試行回数等は、前述した Delphian Desktop と同様の条件で行う。これにより、純粋にポインティング時間だけに焦点をあて、再評価をすることができると考えられる。

## 4. 実装

Delphian Desktop の再評価にあたり、オリジナルの実験環境を可能な限り再現するため、論文[1]の著者にプログラムの提供を依頼したが、入手することができなかった。よって本研究では、[1]で報告されている情報から Delphian Desktop をあらためて実装した。

### 4.1. 実装環境

実験用アプリケーションの実装には、Windows オペレーティングシステム上 Java Simple DirectMedia Layer (JavaSDL)を用いた。[1]では Microsoft Visual C++を用いているが、JavaSDL も C++で用いられる DirectInput と同様な機能によって独自のマウス動作を実装できる。影響はないと考える。ターゲットを予測するための式(1)での  $a, b$  は、[1]では被験者ごとに行ったデータ取得実験によって求められるとされていたが、本研究ではデータ取得実験は行わず、[1]の著者から提供されたある被験者のデータ(2)を用いた。

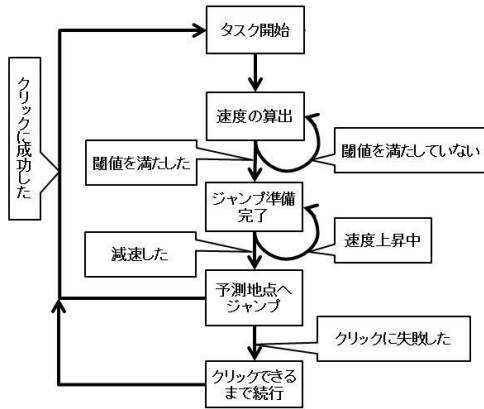


図 4. フローチャート

$$\begin{aligned}
 \text{上: } & y = 0.374433x + 26.2461 \\
 \text{右上: } & y = 0.3865x + 122.537 \\
 \text{右: } & y = 0.348396x + 41.0221 \\
 \text{右下: } & y = 0.304734x + 312.215 \quad (2) \\
 \text{下: } & y = 0.273049x + 151.154 \\
 \text{左下: } & y = 0.37586x + 105.38 \\
 \text{左: } & y = 0.224217x + 308.277 \\
 \text{左上: } & y = 0.500736x - 94.9038
 \end{aligned}$$

式(2)は上下左右斜め 8 方向に応じた式(1)を表し,  $y$  は  $D$  に,  $x$  は  $V_p$  に対応する. 閾値としてはターゲットまでの到達時間が 300 ms 以上になる速度を用いた.

#### 4.2. 実装概要

本研究の実験用アプリケーションでは図 4 に示したような流れでプログラムを動作させる.

- ① タスク開始時に, ターゲットの方向別に与えられた式(1)を適用させる.
- ② ポインティング中のマウスの位置情報を常に監視し, そこから速度を算出する.
- ③ 閾値を超えた場合, ジャンプの準備ができているというフラグを真とする.
- ④ 速度の低下が検出された場合, ①で適応された式(1)に値を代入し, マウスカーソルを移動させる.
- ⑤ フラグを真から偽に戻す.
- ⑥ ターゲットへのクリックが成功した場合, 次のタスクへ移行し①を実行する.

また, 同プログラムでは1回のタスクでのジャンプ回数を1回までに制限している. これにより, 無意味なジャンプによってマウスカーソルがターゲットを超過し, かねてポインティング時間が増加することを防止できる. 実装した実験用アプリケーションを図 5 に, 実際に操作している様子を図 6 に示す. 本研究の実装ではマウスカーソルを三角形で, ターゲットを正方形で表現している. また, 被験者に分かり易くするため, 明確な色を使用している.

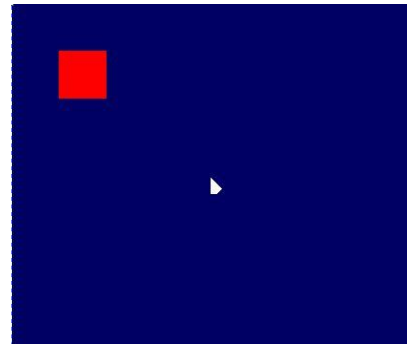


図 5. 実験用アプリケーション



図 6. 操作の様子

#### 5. 評価実験

本研究では, 2 節で述べたタスク条件で[1]と同様に 6 人の被験者(平均年齢 21.8 歳の男 4 人, 女 2 人)に対して反復実験を行った. [1]では休憩時間を 5 分間にしていたのに対し, 本実験では 10 分間とした. これは 1 人あたりの実験時間の多さによる疲労の問題を考慮したためである. また, 被験者がタスク時にターゲットを予測することを防止するために, 1 人が一巡したら次の被験者に交代させた. 実験環境には, 解像度  $1280 \times 1024$  pixel としたディスプレイを使用し, 入力デバイスとして日常的に利用されているマウスと, 十分な大きさのマウスパッドを使用した.

評価実験を行った結果を図 7, 8 に示す. 図 7 はターゲットの距離条件ごとのポインティング時間の比較, 図 8 はターゲットへの方向条件ごとのポインティング時間の比較を示したものである. 検定の結果, 予測の有無 ( $F = 8.612, p < .01$ ), ターゲット距離 ( $F = 20.586, p < .01$ ) で有意差が確認された. さらに実験の結果, ターゲットへのクリック失敗の割合が, Delphian Desktop が 5%なのに対し, 加速ありのマウスが 3%とより正確であることがわかった.

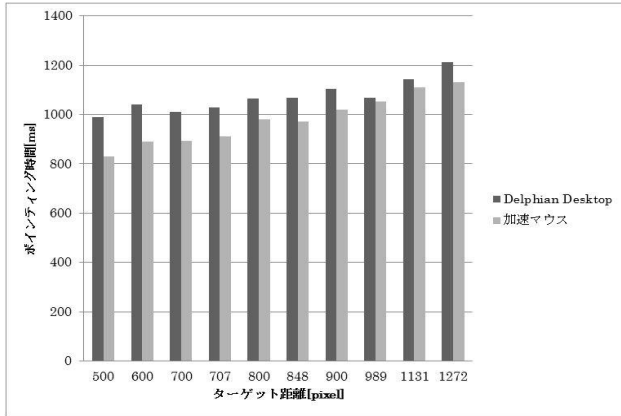


図 7. ターゲット距離に対するポインティング時間

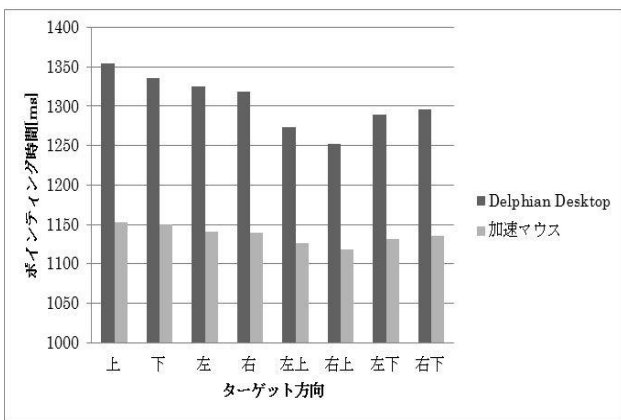


図 8. ターゲット方向に対するポインティング時間

## 6. 考察

図 7, 8 及び検定結果より Delphian Desktop が, 比較的長いターゲット距離に対するポインティングタスクでポインティング時間の短縮に成功していたのに対し, 加速機能ありのマウスではどのターゲット距離に対してもポインティング時間の短縮がみられた. また, 加速機能ありのマウスにおいて, 距離の長い斜め方向に対してのポインティング時間が距離の短い上下左右方向とさほど変わらないという結果も見られた. これは加速機能が距離に比例して上昇するためであると考えられる. さらにクリック失敗の割合から加速ありのマウスがより正確であるという結果から, 被験者の日常的状況下に限りなく近づくことができたと考えられる.

また実験後のアンケートでは 6 人中 4 人の被験者から, 次のターゲットの場所が予測できる時があったとの意見を得た. よって, [1]での評価実験が適当でなかった可能性を示すことができたと考えられるが, 同環境で行った再評価実験でのポインティング時間の比較は Delphian Desktop の実用性を問う判断材料になりえることも事実である. また, 同アンケートでは Delphian Desktop が手の移動量が少なく, ポインティング動作が比較的容易で楽だったという意見も得られた. このことから Delphian Desktop が少なからず被験者の操作負担軽減に役立っている可能性を示すこともできたと考えられる. さらに, 本

研究では, ターゲットを予測するための式(1)での  $a, b$  に関しては, データ取得実験を行わずに, [1]の著者から提供されたものをそのまま用いた. このようにして実装した Delphian Desktop がオリジナルに対してどれだけ近づけたかという問題も残された.

## 7. おわりに

本論文では, ポインティング中のマウスの速度から意図するターゲットを予測し, マウスカーソルをジャンプさせる手法を用いたインタフェースとして提案された Delphian Desktop について, その評価方法への問題提起を行い, 新たな比較対象を用いた再評価実験を行った. その結果, 加速機能ありマウスがどのターゲット距離, 方向に対してもポインティング時間の短縮に有効であり, ポインティングが正確であること, そして Delphian Desktop の評価方法が十分でないことが確認された. しかし, ポインティング時における被験者の操作負担軽減の可能性を示すことができたというのもまた事実である. 今後の課題としては, Delphian Desktop の実装の再現度を高めることが挙げられる.

**謝辞** 本研究の一部のデータや値は, 高嶋和毅氏の提供による.

## 文献

- [1] 高嶋 和毅, 浅野 岳史, Ehud Sharlin, 北村 喜文, 岸野 文郎, “ポインティングタスク中のピーク速度を用いたターゲット予測インタフェースの提案”, 情報処理学会論文誌, vol.48, no.2, February 2007.
- [2] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, “Predictive Interaction using the Delphian Desktop”, Proc. ACM symposium on User interface software and technology (UIST’05), pp.133-141, October 2005.
- [3] Johnny Accot, Shumin Zhai, “Refining Fitts’ law models for bivariate pointing”, Proc. Conference on Human Factors in Computing Systems (CHI’03), pp.193-200, 2003.
- [4] Paul M. Fitts, “The information capacity of the human motor system in controlling the amplitude of movement”, Journal of Experimental Psychology, 47, pp.381-391 1954.
- [5] Shumin Zhai, Stephane Conversy, Michel Beaudouin Lafon, Yves Guiard, “Human on-line response to target expansion”, Proc. Conference on Human Factors in Computing System (CHI’03), pp.117-284, 2003.
- [6] Yves Guiard, Renaud Blanch, Michel Beaudouin, “Object pointing: a complement to bitmap pointing in GUIs”, Proc. Graphics Interface Conference ’04, pp.9-16, 2004.
- [7] 朝日 元生, 高嶋 和毅, 築谷 嵩之, “マウスポインティング時の C-D 比が運動特性に与える影響”, 情報処理学会論文誌, vol.49, no.12, December 2008.
- [8] 栗原 一貴, “マイクロスリップの Human Computer Interaction 研究への応用”, Journal of Ecological Psychology, vol.4, no.1, pp.7-13, 2009.