

| | |
|--------------|--|
| Title | リモートポインティングと方向キー操作を組み合わせたアイコン選択手法 |
| Author(s) | 藤原, 仁貴; 志築, 文太郎; 田中, 二郎 |
| Citation | 第六回知識創造支援システムシンポジウム報告書: 162-169 |
| Issue Date | 2009-03-30 |
| Type | Conference Paper |
| Text version | publisher |
| URL | http://hdl.handle.net/10119/7983 |
| Rights | 本著作物の著作権は著者に帰属します。 |
| Description | 第六回知識創造支援システムシンポジウム, 主催: 日本創造学会, 北陸先端科学技術大学院大学, 共催: 石川県産業創出支援機構文部科学省知的クラスター創成事業金沢地域「アウェアホームのためのアウェア技術の開発研究」, 開催: 平成21年2月26日~28日, 報告書発行: 平成21年3月30日 |

リモートポインティングと方向キー操作を組み合わせた アイコン選択手法

An icon acquisition technique using combination of remote-pointing and arrow-key operation

藤原 仁貴^{1*} 志築 文太郎^{2†} 田中 二郎^{2‡}
Masaki FUJIWARA¹ Buntarou SHIZUKI² Jiro TANAKA²

¹ 筑波大学第三学群情報学類

¹ College of Information Sciences, the Third Cluster of Colleges, University of Tsukuba

² 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻

² Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba

Abstract: There are a lot of researches of remote-pointing devices for large displays which a user can point at a distant position from the display. However, there are problems when a user does pointing. For example, a user can not keep hotspot at his or her intended point, or the user can not point accurately because of hand jitter. Therefore, it is difficult and intrusive for the user to acquire small target from overlapped icons. In this research, we present an icon acquisition technique using combination of remote-pointing and arrow-key operation. First, the user selects candidates of a target with area cursor. The candidates are shown in a circular pattern. Second, the user selects the target by pressing arrow-key from the candidates. We conducted a preliminary experiment to verify whether our method improves the performance in acquisition of a small target from overlapped icons.

Keywords: large display, target acquisition, remote-pointing, freehand pointing, icon, relocation

1 序論

面積の大きなディスプレイ（以降、大画面と呼ぶ）は複数人が同じ情報を見ながら共同作業を行うのに有効な表示装置である。その大画面向けのポインティングデバイスとして、リモートコントローラ（リモコン）の方向キー、ジャイロマウス、レーザーポインタ等、ユーザが画面から離れてポインティングを行うためのデバイス（以降リモートポインティングデバイスと呼ぶ）が利用される。ユーザはリモートポインティングデバイスを手に持ち、空中で操作を行う事によってポインティングを行う。

しかしこれらのデバイスには、以下に挙げる問題が存在する。

- ユーザがデバイスのボタンを押した時に、デバイスがぶれる

- ポイント位置を一カ所に止めておく事が難しい
- 細かい操作を行う事が難しい

上記の問題のため、リモートポインティングデバイスによるポインティング操作には、以下のような問題が存在する。

- 小さなアイコンや密集し重なりあったアイコンを選択する事が難しい

本研究では、大画面上に表示された、多数の小さく、折り重なって表示されたアイコンの中から、素早く、正確に目的のアイコンを選択するための手法を提案する。

2 リモートポインティングデバイスの分類とその問題点

2.1 連続的リモートポインティングデバイス

スタイラスペン、マウス等、ユーザがデバイスそのものを動かす事によってポインティングを行うデバイ

*fujiwara@iplab.cs.tsukuba.ac.jp

†shizuki@cs.tsukuba.ac.jp

‡jiro@cs.tsukuba.ac.jp

スを、本論文では連続的ポインティングデバイスと定義する。連続的ポインティングデバイスでは、ユーザはデバイスの位置や動かし方によって、それに応じたポインティングを行う事が出来る。連続的リモートポインティングデバイスの例として、ジャイロマウスやレーザーポインタ [6] を挙げる事が出来る。これらのデバイスは、ユーザにとって細かい操作を行いつらく、ユーザが目的の位置にポイント位置を正確に移動させる事が難しい。また、ユーザがデバイスのボタンを押した時に手ぶれの影響を受けるという問題や、ユーザにとってポイント位置を一ヶ所に止めておく事が難しいという問題が存在する。

2.2 離散的リモートポインティングデバイス

ボタン入力のみにより操作を行うデバイスでは、ユーザによるボタンの押下に応じてポイント位置を移動させる事が出来る。このようなポインティングデバイスを本論文では離散的リモートポインティングデバイスと呼ぶ。例としてリモコンの方向キーを挙げる事が出来る。ポインティングに、デバイスの移動量や位置等の物理量を利用しない。そのため、方向キーは手ぶれの影響を受けないポインティングデバイスであり、方向キーを利用する事により、ユーザは画面の任意の位置を正確にポイントする事が出来る [9]。しかし、方向キーでは、ポイント位置の移動が一操作につき一定であるため、ユーザはポイント位置の移動のために何度も操作を行わなければならない、時間がかかる。また、ポイント位置の移動方向も制限される。このため、現在のポイント位置から離れた場所をポイントする場合、リモコンを用いると大変時間がかかる。また、ユーザが斜め方向にポイント位置を移動させたい場合、縦方向の移動と横方向の移動を組み合わせるポイント位置を移動させなければならない。この操作はユーザにとって煩わしい。

3 密集した小さなアイコンと Fitts の法則

図 1 は、Google マップ¹により、つくば市中心部の飲食店を検索した結果のスクリーンショットである。地図上に、小さなアイコンが折り重なって表示されている。しかし、これらのアイコンの中には、他のアイコンの下に位置し、ほとんど表示されていない物が存在する。手ぶれの影響や操作性の問題のため、リモートポインティングによりこのようなアイコンを選択する事は難しい。

ターゲットを選択するまでのカーソル移動時間 T は、 D をカーソルとターゲットとの距離、 W をターゲットのサイズとすると、Fitts の法則 [1] から、 $T = a +$



図 1: 密集し選択しづらいアイコンの例

$b \log(D/W + c)$ とあらわされる。この式より、ターゲットのサイズが小さくなると、ターゲットの選択に要する時間が大きくなる。他のアイコンの下に位置するアイコンは、そのアイコンの本来のサイズよりも、選択出来る領域が小さくなってしまふ。このため、選択にかかる時間が増大する。さらに手ぶれの影響、操作性の問題が加わるため、その時間はより顕著に増加すると考えられる。また、アイコンが密集している場合、ユーザが選択操作を誤ると、他のアイコンを誤って選択してしまう可能性が高くなる。意図通りにアイコンを選択出来ない事は、ユーザにとって煩わしい。

4 Rough Selecting の提案

我々は、前章で挙げた問題を解決するために、[14] において、リモートポインティングと 4 方向キー操作を組み合わせたアイコン選択手法 Rough Selecting の提案を行った。Rough Selecting は「リモートポインティングを利用してターゲットの候補を大まかに絞込み、その後方向キーを利用してターゲットを選択する」というアプローチを取る。

連続的リモートポインティングは、前述した通り、正確なポインティングが難しい。この問題を解決するために、Rough Selecting では、連続的リモートポインティングの役割を、ターゲットの位置を大まかに絞り込む事とした。ポイントされた位置付近にあるアイコンをターゲットの候補（候補アイコンと定義する）とし、その中から、ターゲットを方向キー操作により選択する。これにより、ユーザはアイコンの選択を確実に行う事が出来る。

¹<http://maps.google.co.jp/>

4.1 エリアカーソルによる候補アイコンの絞り込み

Rough Selecting では、大まかに候補を絞り込むために、エリアカーソル [7, 11] を利用する。図 2 に示すように、エリアカーソルは、円や四角等の形状をしたカーソルである。エリアカーソルを用いる事により、エリアカーソルの内部に存在するアイコンを選択する事が出来る。図 2 では、エリアカーソルに一部分でも含まれているアイコンが、濃い青でハイライトされている。ユーザはエリアカーソルの内部に存在するアイコンを候補アイコンとして確定する。Kabbash らは [7] において、Worden らは [11] において、エリアカーソルがアイコンの選択時における選択時間のパフォーマンス向上に役立つ事、特にターゲットの大きさが小さい時に大きな効果をもたらす事を示した。この事から、折り重なったアイコンの中から、他のアイコンの下に位置し選択出来る領域が小さくなったアイコンをユーザが選択する時、選択時間におけるパフォーマンス向上が期待出来る。

4.2 候補テーブルと候補の再配置

エリアカーソル内に候補アイコンが複数存在する時、ユーザはその中からターゲットを選択する必要がある。ユーザが方向キーを利用して候補アイコンからターゲットを選択出来るように、システム側が候補アイコンを順序付け、並べ替える。Rough Selecting では、一時的に並べ替えた候補アイコンを配置するため、図 3 に示すパイメニュー [5] 型インタフェースを導入する。これを候補テーブルと呼ぶ。ユーザがアイコンの選択操作を行うと、エリアカーソルの中心位置を中心として候補テーブルが表示される。そして、候補アイコンが候補テーブルに再配置される。

パイメニュー型インタフェースを導入する事によって、システムは候補アイコンを円周上に再配置する。このため、システムは、候補アイコンを方向キー操作に適した一次元的な順序付けを行う事が出来る。また、パイメニュー型インタフェースにおける本来のメニュー選択方法は、候補テーブルの中心から見た、カーソルの移動方向による選択方法である。パイメニュー型イ

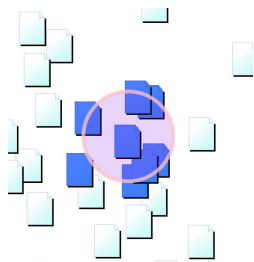


図 2: エリアカーソルによる候補アイコンの絞り込み

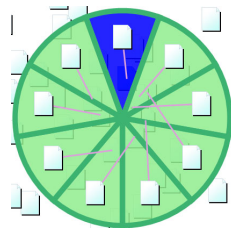


図 3: 候補テーブルへの候補アイコンの再配置

ンタフェースでは、この選択方法により、ユーザは素早いメニュー選択を行う事が出来る。図 4 に、移動方向によるアイコン選択の例を示す。Rough Selecting においても、方向ボタン操作によるターゲット選択の他に、候補テーブルの中心から見たカーソルの移動方向によるターゲットの選択もサポートし、ユーザの好みに応じて使い分けられるようにする。ユーザが円状に配置されたアイコンの中からターゲットを選ぶ場合、選択時間の点で有利であるため [13]、ユーザがリモートポインティングに慣れている場合は、より高速なアイコンの選択を行う事が可能になるためである。

4.3 Rough Selecting によるアイコン選択の手順

本節では Rough Selecting によるアイコン選択の手順について、図 5 に示す例を用いて述べる。この例では、ピンク色にハイライトされたアイコンをターゲットとする。

まずユーザは、リモートポインティングにより、エリアカーソルをターゲットがエリアカーソルの内部に含まれるように移動させる (図 5a b)。ユーザがデバイスのアイコン選択用ボタンを押し下げると、図 5c に示すように、候補テーブルが表示され、その上に候補アイコンが再配置される。この時、指示部分 (図 5c, d に示す、候補テーブル上の青い部分) は 1 つの候補アイコンの下にある。次にユーザは、図 5d に示すように、この指示部分をデバイスの方向キーを用いてターゲットまで移動させる。この操作で用いる方向キーは、左右方向のキーのみである。ユーザが右方向のキーを押すと、図 5d に示すように、指示部分が右回りに進んでいく。左方向キーの場合は、指示部分は左回りに進んでいく。ターゲットまで指示部分を移動させ終わったら、ユーザは今まで押し下げているアイコン選択用ボタンを離す。これでターゲットの選択は完了である。また、前節において述べた通り、ユーザは方向キーを用いずにリモートポインティングのみで、候補テーブルの中心から見たカーソルの方向によるターゲットの選択を行う事も出来る。

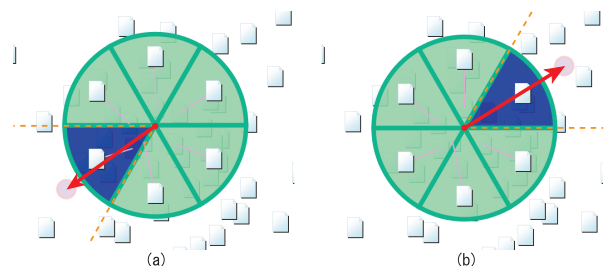


図 4: カーソルの移動方向を利用したターゲットの選択

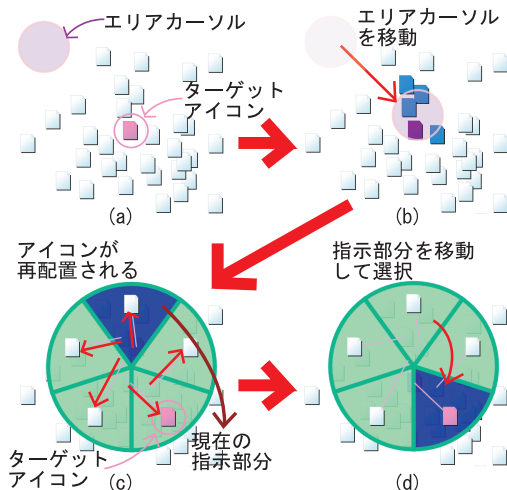


図 5: Rough Selecting によるアイコン選択の手順

4.4 Rough Selecting の応用例

以下では、Rough Selecting の応用例について述べる。
地図アプリケーション

地図アプリケーションでは、店舗情報などを検索した結果がアイコンにより地図上に表示される。これらのアイコンは小さく、また重なり合って表示される事が多い。このようなアイコンの選択に、Rough Selecting は有用であると考えられる。

フォトアルバムにおける写真の閲覧

Rough Selecting を利用すると、写真のサムネイルを画面上に大量に表示して写真のブラウジングを行う事が出来ると考えられる。階層構造を平安京ビュー [12] を用いて表現する事により、ページの切り替えを必要としない高速な閲覧が可能になると考えられる。

5 プロトタイプシステムの設計と実装

5.1 入力デバイスに対する要求

Rough Selecting 実現のためには、1つの入力デバイスに、以下の4つの事が要求される。

- リモートポインティングが可能である事
- 方向キーが搭載されている事
- アイコン選択操作機能を割り当てる事が出来るボタンが搭載されている事
- 上記の操作を同時に行う事が出来るボタン配置になっている事



図 6: Wii リモコンと本システムでの利用部分

プロトタイプシステムでは、上記の要求を全て満たすデバイスとして、図 6 に示す Nintendo Wii リモコン²を採用する。Wii リモコンでは、図 6 に示す赤外線センサと後述のセンサーバーを用いる事によって、リモートポインティングが可能である。また Wii リモコンには表側に4方向キーが搭載されている。さらに、デバイスの裏側(4方向キーの裏側)に搭載されているBボタンをアイコン選択操作機能用のボタンとして割り当てる事により、リモートポインティングとアイコン選択操作、方向キー操作とアイコン選択操作を同時に行う事が出来る。

5.2 ハードウェア構成

作成したプロトタイプシステムでは、ハードウェアとしてPC、ディスプレイ、Wii リモコン、及び自作センサーバーを利用した。ハードウェア構成を図 7 に示す。センサーバーは Nintendo Wii³ で利用されるデバイスであり、センサーバーの左右両端に1か所ずつ赤外線 LED が搭載されている。Wii リモコンの赤外線センサを用いてこの LED が発する赤外線からデバイスが向いている方向に関する情報を取得し、その値を用いる事によりポインティングが可能である。センサーバーを自作した理由は、Wii に用いられているセンサーバーの赤外線 LED は、輝度が低く、約 5m ユーザがセンサーバーから離れるとポインティングを行う事が難しくなるためである。プロジェクトとスクリーンを用いた大型の画面を、5m 以上離れて利用する事も視野に入りたいと考えているため、より高輝度な赤外線 LED を用いてセンサーバーを自作した。Wii リモコンとセンサーバーを用いる事によって、ユーザは、リモートポインティングと方向キー入力を併用してアイコンを選択する事が出来る。

PC と Wii リモコンの通信には、Bluetooth アダプタ (Planex BT-Mini2EDRW) を利用した。

5.3 ソフトウェアの実装

Visual C# と Wii リモコン用ライブラリ WiimoteLib⁴ を使用してプロトタイプシステムの実装を行った。

²http://www.nintendo.co.jp/wii/features/wii_remote.html

³<http://www.nintendo.co.jp/wii/>

⁴<http://www.codeplex.com/WiimoteLib>

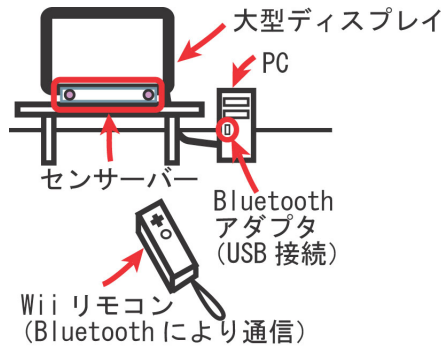


図 7: ハードウェア構成

5.3.1 候補アイコン再配置位置の計算

図 8 に示す例のような座標系における，候補アイコン再配置位置（再配置位置）の計算について述べる．ここでは，プロトタイプシステムのソフトウェアにおける，画面の左上座標 $(0, 0)$ を，この座標系の原点 o とする．

まずソフトウェアは，候補テーブル（円 C ）の中心位置を，エリアカーソルの中心位置に設定する．この位置を， $o'(o'_x, o'_y)$ とする．エリアカーソルにより，ユーザが n 個の候補アイコンを確定したとする．この場合，ソフトウェアは再配置位置を n 個計算する必要がある．再配置位置は， $n = 1$ の時，図 8 に示すように，円 C の中心 o' である．また，ここで k 番目の再配置位置を $P_k(p_{kx}, p_{ky})$ とすると， $n = 2$ の時，再配置位置は，図 9 に示すように，線分 P_1P_2 の両端である． $n \geq 3$ の時，再配置位置は，図 10 に示すように，半径 r の円 C に内接した，正 n 角形の頂点である．この時，再配置先の座標は，以下の式により求められる．但し， $n \geq 1$ とする．また， $k = 1, 2, \dots, n$ とする．

$$p_{kx} = \begin{cases} o'_x & (n = 1) \\ o'_x + r(-1)^{k-1} & (n = 2) \\ o'_x + r \sin(\frac{2(k-1)\pi}{n}) & (n \geq 3) \end{cases} \quad (1)$$

$$p_{ky} = \begin{cases} o'_y & (n < 3) \\ o'_y - r \cos(\frac{2(k-1)\pi}{n}) & (n \geq 3) \end{cases} \quad (2)$$

なお， $n = 1$ の時もアイコンを再配置する事にした理由は， $n \geq 2$ の時とユーザに対するフィードバックを統一するためである． $n \geq 2$ の時は，どの候補アイコンをどの再配置先へ移動させるかを定める必要がある．

まず，図 9, 10 に示すように，線分 $o'P_1$ を基準として時計回りに， o' と各候補アイコンの中心 $i_k(i_{kx}, i_{ky})$ を結ぶ線分 $o'i_k$ の角度 θ_k を求める． x_k, y_k, z_k を図 11

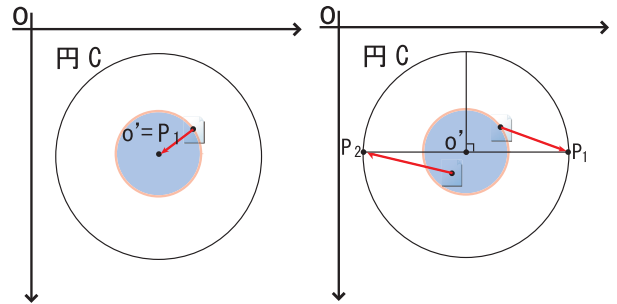


図 8: 候補再配置位置の計算 ($n = 1$ の時) 図 9: 候補再配置位置の計算 ($n = 2$ の時)

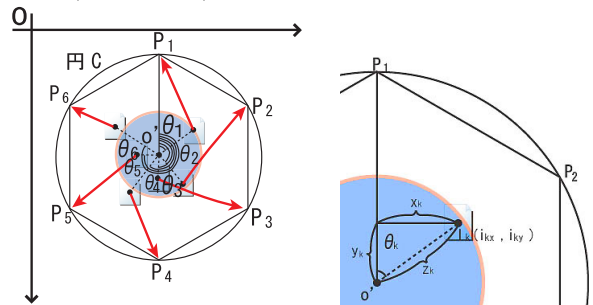


図 10: 候補再配置位置の計算 ($n \geq 3$ の時)

図 11: θ_k の計算

のように定めると，これらは以下の式によって計算出来る．

$$x_k = i_{kx} - o'_x \quad (3)$$

$$y_k = o'_y - i_{ky} \quad (4)$$

この式 (3), (4) より，

$$z_k = \sqrt{x_k^2 + y_k^2} \quad (5)$$

従って， θ_k は以下の式により求める事が出来る．

$$\theta_k = \arccos\left(\frac{z_k^2 + y_k^2 - x_k^2}{2z_k y_k}\right) \quad (6)$$

各アイコン全ての θ_k が求めれば，これらを基準として小さい順に，各アイコンの再配置先を， P_1 から P_n まで順に割り当てていく．

6 予備実験

本実験では，リモートポインティングにより得られたポイント位置をそのままアイコンの選択に利用する場合と，Rough Selecting を利用する場合とを比較する．ユーザが密集し重なり合ったアイコンを選択する場合，選択にかかる時間の点においてどちらが有用であるかを調べる．

6.1 被験者

被験者は、筆者自身と、同じ研究室の学生の計2名であった。年齢は22歳及び23歳であった。また、両人とも右利きであった。筆者はプロトタイプシステムを使い慣れていた。もう一人の学生は、プロトタイプシステムの使用経験が1,2度あった。

6.2 実験環境

本実験では、ディスプレイとして50V型プラズマディスプレイ (Panasonic TH-50PH50, 1280×1024 pixel) を使用した。また、Windows Vista Business Service Pack1 がインストールされた、Intel(R) Core(TM)2 Duo E6550(2.33GHz×2) 及び4GBのメモリを搭載したPCを使用した。センサーバーの赤外線LED間隔は15cmとした。

6.3 実験のデザイン

図12に示す手順によって32x32 pixelのアイコンを9個並べる。この重なり合った9個のアイコンをアイコンセットと呼ぶ。アイコンセットは図13のように、縦横それぞれ244 pixel間隔で画面上に9セット表示される。

独立変数は2種類である。1つ目は、アイコンの面積に対する、他のアイコンと重ならずに見えている面積の率(1/16,1/8,1/4,1/2,1の5通り: 図14)である。これを面積率と呼ぶ。2つ目は、下記のアイコン選択手法(P, AP, RS, ARSの4通り)である。

P(Pointer) リモートポインティングにより得られたポイント位置を、そのままアイコンの選択に利用する手法である。一般的な矢印型マウスカーソルを被験者がポインティングを行ってターゲットの上に移動させ、アイコン選択用ボタンを押す事によってアイコンを選択する。

AP(ArrowKey Pointer) Wiiリモコンの4方向キーを用いた相対ポインティングにより得られたポイ

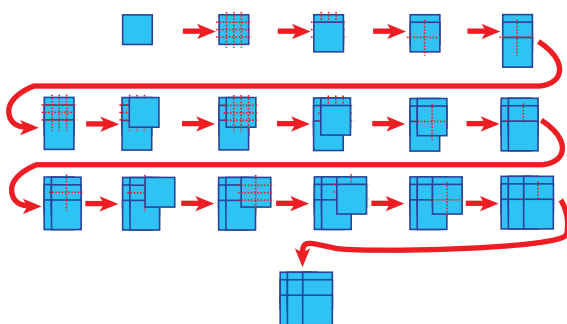


図12: アイコンセットの作成の流れ

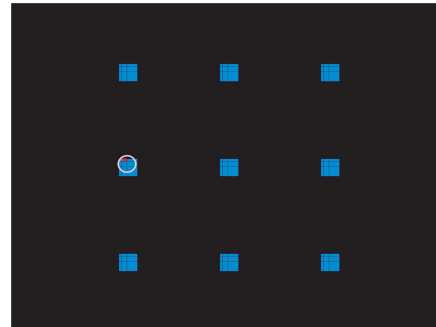


図13: デザインに基づいて配置されたアイコンセットと指示されたターゲット

ント位置を、そのままアイコンの選択に利用する手法である。APは、ポインティングの手法以外はPと同様である。

RS(RoughSelecting) アイコンの選択に Rough Selecting を利用する手法である。RSでは、被験者は、パイメニューにおける項目選択のように、候補テーブルからのターゲット選択を、候補テーブルの中心から見たカーソルの移動方向により行う。なお、エリアカーソルの半径は50 pixelとした。

ARS(ArrowKey RoughSelecting) アイコンの選択に Rough Selecting を利用する手法である。RSとの違いは、候補テーブルからのターゲット選択に、方向キーを用いる点である。この手法でも、エリアカーソルの半径は50 pixelとした。

6.4 実験の手順

実験では、被験者がアイコンを選択するのにかかる時間を測定する。被験者は、4通りのアイコン選択手法それぞれを用いてアイコンを指示された順番に1個ずつ選択していく。

実験は画面中央に紫色の矩形が表示されている状態が始まる。被験者がその矩形上にポイント位置を合わせ、

| | | |
|----------------|---------------|---------------|
| $\frac{1}{16}$ | $\frac{1}{8}$ | $\frac{1}{4}$ |
| $\frac{1}{8}$ | $\frac{1}{4}$ | $\frac{1}{2}$ |
| $\frac{1}{4}$ | $\frac{1}{2}$ | 1 |

図14: アイコンセット内の各アイコンの面積率

そこでアイコン選択用ボタンを押すと、最初のターゲットが指示され、時間の測定 (msec 単位で測定) が開始される。同時に画面上から紫色の矩形が消える。ターゲットは図 13 に示すように赤く表示され、かつ円で囲まれて強調される。被験者は、カーソルを移動し、ターゲットを選択する。被験者が正しくターゲットを選択出来れば、最初の測定が終了する。そして、他のアイコンにターゲットが移り、このアイコンを選択するまでの時間の測定が開始される。被験者は同様に選択操作を繰り返していく。被験者が 81 個 (9 アイコンセット=81 アイコン) のアイコンを 1 回ずつ選択し終わると、実験は完了である。ターゲットの指示される順番は、どのアイコン選択手法においても同じである。その順番は、連続して同一のアイコンセット中のアイコンがターゲットにならないよう制約を加え、疑似乱数により決定した。同一のアイコンセットから連続してターゲットを選ぶと、ユーザはエリアカーソルを移動させずにアイコンを選択出来てしまうからである。

実験を開始する前に、被験者には操作に慣れるまで実験の練習を行ってもらった。また、被験者には、ディスプレイから 10 フィート離れた所に設置された椅子に、自由な姿勢で座ってもらった。デバイスの持ち方については特に指示しなかった。

7 予備実験の結果と議論

実験結果を基に、平均選択時間について分析を行い、また問題点について考察を行った。

7.1 平均選択時間

図 15 は、実験結果から求めた、選択手法及び面積率毎のアイコンの選択時間の平均と標準偏差である。

アイコン選択手法と面積率を要因として二元配置分散分析を行った結果、アイコンの選択手法 ($F(3, 51) =$

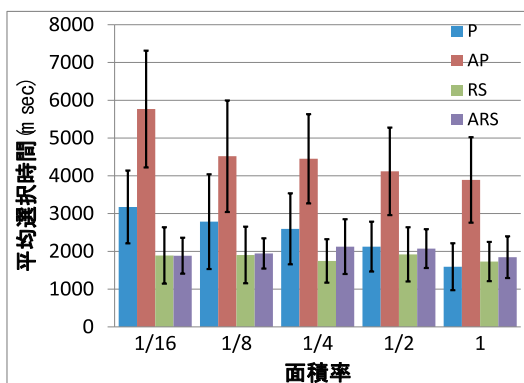


図 15: 選択手法及び面積率毎のアイコン選択時間の平均と標準偏差

$196.20, p < 0.01)$ 、面積率 ($F(4, 68) = 8.32, p < 0.01)$ 及びアイコン選択手法と面積率の交互作用 ($F(12, 204) = 4.06, p < 0.01)$ において有意に差が見られた。

アイコン選択手法の単純主効果を検定したところ、面積率 $1/16$ ($F(3, 204) = 60.84, p < 0.01$)、 $1/8$ ($F(3, 204) = 24.13, p < 0.01$)、 $1/4$ ($F(3, 204) = 31.45, p < 0.01$)、 $1/2$ ($F(3, 204) = 31.52, p < 0.01$)、 1 ($F(3, 204) = 36.18, p < 0.01$) 全てにおいて有意に差があった。LSD 法を用いた下位検定を行った結果、面積率 $1/16$ では、AP は P よりも平均選択時間が有意に大きく、P は RS, ARS よりも有意に大きかった。RS, ARS の間には有意な差は見られなかった ($MSe = 989977.95, p < 0.05$)。面積率 $1/8$ ($MSe = 1065700.88, p < 0.05$)、 $1/4$ ($MSe = 700983.78, p < 0.05$)、 $1/2$ ($MSe = 490709.23, p < 0.05$)、 1 ($MSe = 590163.25, p < 0.05$) は、P, RS, ARS の間には有意な差は見られず、これらと AP の間では、AP の選択時間が有意に大きかった。以上の分析結果と図 15 より、RS, ARS は、面積率のどのような条件下においても、アイコン選択時間が P, AP よりも有意に小さいか、ほぼ等しいという事が分かった。Rough Selecting は、一度候補アイコンを確定し、その中からターゲットを選択するという 2 段階の選択手法であるため、RS, ARS は、P よりもアイコンの選択に要求される操作の回数が多い。そのため当初は、面積率が大きい時 RS, ARS は P よりも平均選択時間が大きいのではないかと予測をしていた。しかしアイコンの面積率が 1 の時も、P と RS, ARS の間で有意な差が見られなかった事から、面積率が大きい時も Rough Selecting が利用出来る事が分かった。これは、Rough Selecting で用いているエリアカーソルの効果が大きいと考えられる。

面積率の単純主効果を検定したところ、P ($F(4, 68) = 10.00, p < 0.01$)、AP ($F(4, 68) = 4.79, p < 0.01$)、ARS ($F(4, 68) = 4.78, p < 0.01$) には有意差が見られた。しかし、RS では有意差が見られなかった。このことから、RS では面積率とアイコン選択時間との間に関係がなく、アイコン選択時間の点で安定した手法であると言える。当初は、ARS も面積率の単純主効果に有意差は現れないと考えていたが、実際には現れた。予想と反した結果になった原因は、Wii リモコンの方向キーと、B ボタン (アイコン選択用ボタン) の位置関係にあると思われる。被験者から「B ボタンを押しながらの方向キー操作は行いづらい」という意見が得られた。実験中被験者を観察していると、被験者が方向キー操作を行っている時、誤って B ボタンを離してしまう事があった。ARS の方が標準偏差が低い時と高い時があり、安定していない事も、この事が理由であると考えられる。この問題を解決するためには、デバイスのボタン配置を改良すればよいと考えられる。

8 関連研究

大画面上における、指やスタイラスペンを用いた直接ポインティングでは、ユーザの腕や手を動かす量が大きくなり、ユーザの負担が増加するだけでなく、ターゲットの選択にかかる時間が増加する。また、高解像度ディスプレイにおいて、ユーザが画面の端にあるカーソルを、画面の反対側にあるターゲットへ移動させるような場合、カーソルの移動時間が増加する。このような問題を解決するため、Fittsの法則に基づき、 D を減少させたり、 W を増加させる事によりターゲット選択時間の減少を図る様々な研究がなされている [2, 3, 4, 8]。

しかし、これらの手法は、リモートポインティングを利用する事が想定されていない。そのため、リモートポインティングデバイスにこれらの手法をそのまま適用しても、デバイスの操作性の問題や、手ぶれの影響の問題を大きく受ける事には変わりはないものと考えられる。リモートポインティングを想定に入れたターゲット選択手法は数が少ないのが現状である。数少ない例として、リモートポインティング時の操作性の問題や、手ぶれの影響を考慮に入れた、Tseらの [10] が挙げられる。Tseらは、リモートポインティングによるアイコンの選択手法として、バブルカーソル [4] を用いて候補を確定し、ターゲットに付加された色情報を発声する事により選択を行う手法の開発と評価を行った。しかし、特徴となる情報をすべてのアイコンに付加しなければならぬという問題や、ターゲットの特徴を発声する必要がある事から、利用出来る場面や場所が限られるという問題がある。これに対し我々の手法は、アイコンに特徴となる情報を付加する必要がない。また、全ての操作をユーザの手により行う。このため、大画面環境ならばどのような環境でも利用する事が出来る。

9 まとめと今後の課題

本研究では、リモートポインティングと方向キー操作を組み合わせた大画面向けアイコン選択手法「Rough Selecting」の提案を行った。これにより、ユーザに要求されるリモートポインティングの精度が低くなるため、特にターゲットのサイズが小さい又は密集して重なり合っている時、ターゲット選択タスクの精度が向上し、またターゲット選択時間のパフォーマンスを向上出来る。また、Wii リモコンを入力デバイスとして採用したプロトタイプシステムの設計と実装を行った。さらに、予備実験を行い、パフォーマンス向上に有益である可能性を見出した。今回行った実験環境では、アイコンが小さい程、又は他のアイコンの下に位置し見えている面積が小さい程ターゲットの選択にかかる時間のパフォーマンスの点において有益である事を示した。

今後は、候補テーブルの改良を行いたいと考えている。また方向キーの上下方向、斜め方向も利用したより素早く確実なターゲット選択を手法へ改良していきたい。また、今回行った実験の結果を基に、本格的な被験者実験を行い、本手法の有用性を確かめていく予定である。

参考文献

- [1] Accot, J. and Zhai, S.: Beyond Fitts' law: models for trajectory-based HCI tasks, *Proc. of CHI '97*, pp. 295-302 (1997).
- [2] Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F.: Predictive Interaction using the Delphian Desktop, *Proc. of UIST '05*, pp. 133-141 (2005).
- [3] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B. and Zierlinger, A.: Drag-and-Pop and Drag-and-Pick: techniques for accessing remote screen content on touch- and pen-operated systems, *Proc. of Interact '03*, pp. 57-64 (2003).
- [4] Grossman, T. and Balakrishnan, R.: The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of The Cursor's Activation Area, *Proc. of CHI '05*, pp. 281-290 (2005).
- [5] Jack Callahan, D. H., Weisert, M. and Shneiderman, B.: An empirical comparison of pie vs. linear menus, *Proc. of '89*, pp. 95-100 (1989).
- [6] Jr., D. R. O. and Nielsen, T.: Laser pointer interaction, *Proc. of the CHI '01*, pp. 17-22 (2001).
- [7] Kabbash, P. and Buxton, W.: The "Prince" Technique: Fitts' Law and Selection Using Area Cursors, *Proc. of CHI '95*, pp. 273-279 (1995).
- [8] Kobayashi, M. and Igarashi, T.: Ninja cursors: using multiple cursors to assist target acquisition on large screens, *Proc. of CHI '08*, pp. 949-958 (2008).
- [9] Lim, S. C., Han, J. H., Jo, M. Y., Jeon, E. M., Choi, W. S., Song, C. G., Sim, S. Y. and Kim, S. W.: Effective interaction techniques for moving cursor using a remote control, *CHI '04 extended abstracts on Human factors in computing systems*, p. 1542 (2004).
- [10] Tse, E., Hancock, M. and Greenberg, S.: Speech-Iterated Bubble Ray: Improving Target Acquisition on Display Walls, *Proc. of ICMI '07*, pp. 307-314 (2007).
- [11] Worden, A., Walker, N., Bharat, K. and Hudson, S.: Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons, *Proc. of CHI '97*, pp. 266-271 (1997).
- [12] 伊藤貴之, 小山田耕二: 平安京ビュー ~ 階層型データを基盤状に配置する視覚化手法, 可視化情報学会第9回ビジュアライゼーションカンファレンス (2003).
- [13] 越澤勇太, 日浦慎作, 佐藤宏介: ポインティングによる多数項目からの選択インタフェースの設計と評価, インタラクション 2008, pp. 101-108 (2008).
- [14] 藤原仁貴, 志築文太郎, 田中二郎: Rough Selecting: 直接指示とボタン入力を組み合わせたアイコン選択手法 (2009年3月発表予定), 情報処理学会 第71回全国大会 (2009).