

ゴーストハンティング：疑似オブジェクト 提示によるオブジェクト選択最適化手法

桑原 智大^{†1} 山本 景子^{†1} 倉本 到^{†1}
辻野 嘉宏^{†1} 水口 充^{†2}

選択対象となるオブジェクトのサイズが小さかったりカーソルとの距離が離れていたりするとポインティングは難しくなる。本研究では、ボロノイ領域を利用してオブジェクトの被選択が判定される領域を拡大し、さらに移動距離が最短となるカーソルの軌跡の終点を提示するオブジェクト選択支援手法「ゴーストハンティング」を提案する。本手法を GUI 環境を模した実験環境で提案手法の性能を他の手法との比較によって評価した。その結果、オブジェクトが垂直一列にならぶ配置では提案手法によってターゲットの選択時間が短くなった。

Ghost-Hunting: Target Selecting Method by Representing Pseudo Objects

KUWABARA CHIHIRO,^{†1} YAMAMOTO KEIKO,^{†1}
KURAMOTO ITARU,^{†1} TSUJINO YOSHIHIRO^{†1}
and MINAKUCHI MITSURU^{†2}

Pointing tasks become difficult when a target is far away from a current cursor position or target size is small. In order to solve the problem, we propose “Ghost-Hunting” method, which 1) extends selectable objects by Voronoi diagram technique, and 2) shows an endpoint of the shortest cursor trail between the cursor and each object. We conducted experiments to evaluate the performance of Ghost-Hunting by comparing it with other selection methods in practical GUI environments. The results show that the Ghost-Hunting method can reduce target selection time in the environment where objects are placed in line vertically.

1. はじめに

一般的なコンピュータのグラフィカルユーザインタフェース (GUI) 環境において、通常使われているポイントカーソルは、カーソルに設定されている、オブジェクトを選択したと判定される点 (ホットスポット) の座標をターゲット領域の内部まで移動させることでターゲットの選択が可能になる。これはポインティングタスクにおいて、カーソルの位置とターゲットの位置が離れているときほど、ユーザがカーソルをより長い距離動かさなければならぬという問題があることを意味する。

これまで、カーソルの移動距離を短縮するために様々な試みが行われてきており、ターゲットまでの距離を予測してカーソルをターゲットの位置までジャンプさせる Delphian Desktop¹⁾ や、ターゲットとなるファイルアイコンをカーソルの周辺に引き寄せるように提示して移動距離を短縮する Drag-and-Pop²⁾ や Vacuum³⁾ が提案されている。移動距離の代わりにユーザのカーソル操作における不正確さを容認し、これを支援することに重点を置いた Birdlime Icon⁴⁾ は、鳥もちのメタファを用いてカーソルの移動速度の調整とターゲットの変形によりポインティング性能を向上させる。これらの手法は距離そのものを短くしたりオブジェクトの被選択が判定される領域を拡大したりすることに注目している。

ターゲット選択の性能を向上させる研究の中でも、エリアカーソル⁵⁾ は、点であるカーソルのホットスポットを、エリアと呼ばれる面へと拡大することで移動距離の短縮を実現する。だがエリアカーソルの場合、エリアの形状によってはカーソルをオブジェクトへ向かう方向へ移動させることが必ずしも最短距離の移動とはならない。にもかかわらず、この移動方向という観点から移動距離の短縮にアプローチする手法は少ない。本研究ではカーソルの移動距離の短縮を目的とする、カーソルの移動方向に着目したオブジェクト選択最適化手法「ゴーストハンティング」を提案する。

2. 従来のカーソルとその問題点

2.1 ボロノイ図に基づくカーソル

エリアカーソルは、エリアと呼ばれる領域を持つカーソルである。エリアの全部または一

^{†1} 京都工芸繊維大学
Kyoto Institute of Technology

^{†2} 京都産業大学
Kyoto Sangyo University

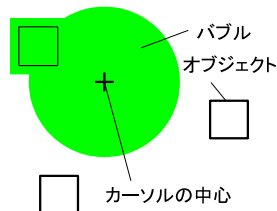


図 1 バブルカーソル⁶⁾
Fig. 1 Bubble cursor⁶⁾.

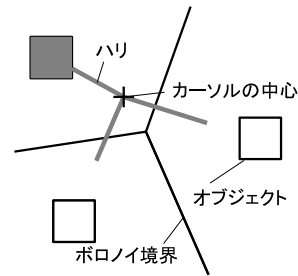


図 2 ハリセンボンカーソル⁷⁾
Fig. 2 Harisenbon cursor⁷⁾.

部がターゲットと接触すると、カーソルはターゲットの選択が可能になる。つまりエリアカーソルは、ポイントカーソルのホットスポットを面へと拡張したものとみなすことができ、エリアの面積の分、選択は容易になる。

例えば、エリアカーソルの一種であるバブルカーソル⁶⁾はバブルと呼ばれるエリアを持つ。バブルカーソルの模式図を図 1 に示す。バブルはユーザがカーソルを操作するときの目安となる点(ポイントカーソルでいうところのホットスポットの位置。以降「カーソルの中心」)を中心に広がる円の内部領域であり、バブルカーソルではバブルの全部または一部と接触するオブジェクトを選択することができる。バブルはカーソルの中心の座標から最も近い位置にあるオブジェクト(以降「最近傍オブジェクト」)のみを選択できるように、常にその半径を変化させており、カーソルは最近傍オブジェクトをカーソルの移動なしに選択することができる。また、バブルカーソルの応用性は高く、Ninja cursors⁸⁾や Comet and target ghost⁹⁾など、さまざまなポインティング手法において併用されている。

ハリセンボンカーソル¹⁰⁾⁷⁾もバブルカーソルと同様、最近傍オブジェクトを移動なしに選択できるエリアカーソルである。ハリセンボンカーソルの模式図を図 2 に示す。ハリセンボンカーソルではカーソルと最近傍オブジェクトはハリと呼ばれる線分で接続される。また、最近傍オブジェクトではないオブジェクトに対しては、オブジェクトを選択できるようになる位置からオブジェクトまでの距離の長さを示すハリがカーソルから延びる(こちらのハリはオブジェクトと接続されない)。

両カーソルの、移動を伴わずに最近傍オブジェクトを選択できるという機能は、2次元のポロノイ図より説明することができる。2次元のポロノイ図とは、母点と呼ばれる点が定義された平面において、ある点に最も近い母点はどれか、という基準で平面を分割した図のこ

とである。分割された領域は最も近い母点を同じくする点の集合であり、ポロノイ領域と呼ばれる。また、ポロノイ領域間の境界となる線はポロノイ境界と呼ばれる。バブルカーソルやハリセンボンカーソルではオブジェクトの中心座標が母点の座標に相当し、カーソルの選択可能なターゲットはカーソルがどのポロノイ領域に含まれるかで決定される。

本研究ではバブルカーソルやハリセンボンカーソルのように、ポロノイ図からオブジェクトの選択の可能・不可能を説明できるカーソルを「ポロノイ図に基づくカーソル」と呼ぶ。前述の通り、ポロノイ図に基づくカーソルは、オブジェクトの被選択が判定される領域をポロノイ領域に変形・拡大することで、ポイントカーソルを使用したときより移動距離を短くできる。

2.2 ポロノイ図に基づくカーソルの問題点

一般的なポイントカーソルにおいて、カーソルの移動距離を最短にするには、ターゲットとカーソルとを一直線に結ぶ線分に沿ってカーソルを動かせばよい。しかしポロノイ図に基づくカーソルでは、カーソルの移動距離を最短にする軌跡はカーソルからターゲットの含まれる領域の最も近い座標へ至る軌跡に等しく、両者の軌跡は一般には一致しない。図 3 はポイントカーソルとポロノイ図に基づくカーソルの移動のさせ方の違いの例を示している。図中の矢印はターゲット選択を最短距離の移動で行う場合のカーソルの移動距離と移動方向を表したものである。矢印の先端はそれぞれのカーソルにおける移動距離が最短となる軌跡の終点(以降「最短終点」)を示しており、それぞれのカーソルからターゲット選択をする上で最も近い位置である。それぞれのカーソル間では、明らかに移動距離と移動方向が異なることがわかる。

いま、図 3 ではポロノイ境界が可視であるため、ポロノイ図に基づくカーソルの移動方向を推定しやすい。しかしポロノイ境界がユーザに提示されない場合、最短終点を正確に推定することはできないため、ユーザがポロノイ図に基づくカーソルにおいて最短となる移動距離でターゲット選択を行うのは難しいという問題がある。これはカーソルの移動距離を短くするというポロノイ図に基づくカーソルの特徴を最大限発揮できないことを意味している。

例えばバブルカーソルではバブルとなる円の半径は最近傍オブジェクトと 2 番目に近いオブジェクトの位置によってのみ決定され、ポロノイ領域の形状が反映されないため、上記の問題は解決されない。また、ハリセンボンカーソルでは、最近傍オブジェクトではないオブジェクトに対して伸びるハリの長さがポロノイ領域へ移動する際の移動距離の目安となるが、これらはカーソル-オブジェクト間の直線距離を元に算出されており、バブルカーソル同様にポロノイ領域の形状が反映されないため、最短終点の推定を支援することはできない。

問題を解決する最も単純な方法は、ボロノイ境界の線を画面に描画したりボロノイ領域を色分けしたりして、ユーザにボロノイ図そのものを提示することである。しかしこれらの方法はインタフェース設計に干渉し、美観を損ねるなどの問題を引き起こす可能性があり、実際のシステムへの適用が難しい。

3. 提案手法 -ゴーストハンティング-

3.1 問題解決のアイデア

最短終点はカーソルの位置とオブジェクトの配置から求めることができる。よってボロノイ図に基づくカーソルの問題の解決案のひとつとして、ボロノイ図上の最短終点をユーザに提示することが考えられる。しかし、一般にユーザがどのオブジェクトを選択しようとしているかを知ることはできない。そこで、全てのオブジェクトのボロノイ領域について最短終点の位置情報をユーザに提示するという方法を提案する。ユーザは提示された全ての最短終点の情報からターゲットに対応する最短終点を選び、それへカーソルを動かすことになる。このときターゲットに対応する最短終点の情報をユーザが即座に把握できるようなインタフェースを構築する必要がある。

以下、これらの議論をもとに、オブジェクト選択支援手法「ゴーストハンティング」を提案する。

3.2 ゴーストハンティング

ゴーストハンティングの適用の例を図4に示す。説明の都合上、図ではボロノイ境界を描いてある。ゴーストハンティングを用いたカーソルにおいては、最近傍オブジェクト以外のオブジェクトについて擬似オブジェクト画像「ゴースト」を導入する。ゴーストは各オブジェクトと対応しており、各ゴーストはボロノイ図上の各最短終点の座標に表示される。このとき、ユーザはターゲットに対応するゴーストの位置から最短終点を知ることが可能となる。また、ゴーストの外観は透明度以外は対応するオブジェクトと同じであり、ユーザはオブジェクトとゴーストの対応関係を把握することができる。

ゴーストの挙動の例を図5に示す。この図に示すように、ハリセンボンカーソルと同様に、カーソルと最近傍オブジェクトは線分で接続されている。一方、最近傍オブジェクトではないオブジェクトに対してはゴーストのみが表示されている。ゴーストは線分で接続されていない各オブジェクトと1対1対応しており、カーソルの位置座標が変化するたびに適切な位置に再描画される。つまりカーソルがボロノイ領域内を移動すると、ゴーストは図5(a)のように周囲のボロノイ境界に沿って動く。

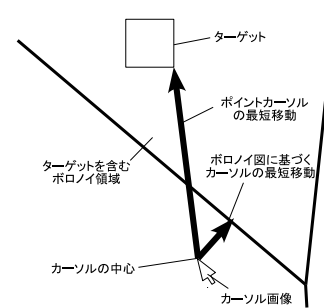


図3 カーソルの移動のさせ方の違い
Fig. 3 The difference of movements between point cursor and Voronoi-based cursor.

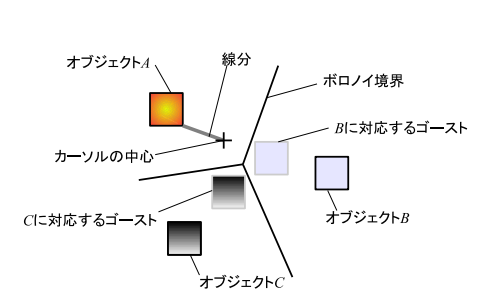


図4 ゴーストハンティング
Fig. 4 Ghost-Hunting.

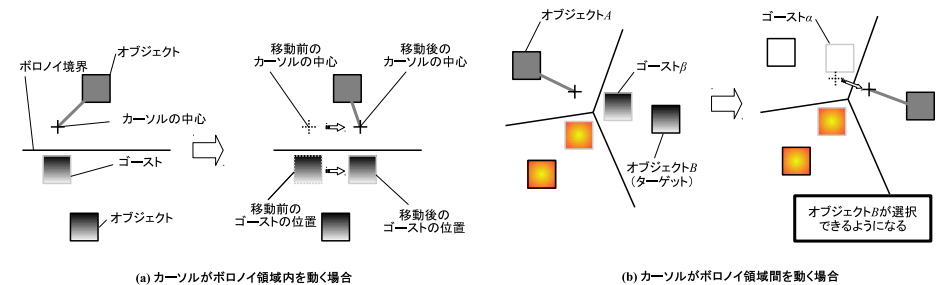


図5 ゴーストの挙動
Fig. 5 Ghost behavior.

図5(b)でボロノイ領域間を移動するときの様子を示す。はじめに、ユーザは画面からターゲットに対応するゴーストを見つける。図ではオブジェクトBがターゲットであるから、ユーザはそれに対応するゴーストβを探すことになる。つぎにユーザは、見つけたβを捕らえるように、カーソルをβのある方向へ向かって移動させる。カーソルがβと重なる位置まで移動するとβは消え、Bとカーソルが線分で接続されるようになる。このとき今まで線分で接続されていたオブジェクトAとの接続は切れ、代わりにAに対応するゴーストαが出現する。

このようにゴーストハンティングを用いたカーソルは常に最近傍オブジェクトを選択する

ことができ、カーソル操作にかかる時間やカーソル操作の難しさを軽減する。加えて、ゴーストによる最短終点の提示により、カーソルの移動距離を最短にすることで移動時間を短くすることが期待される。

4. 実験

4.1 実験目的

本研究では GUI 環境におけるポインティングタスクにおけるゴーストハンティングの効果を、ポイントカーソルおよび他のカーソル手法との性能の比較によって調査する。

4.2 設 計

実験は実験 1 と実験 2 の 2 つからなる。実験 1 では GUI 環境においてよく見られる、スクロールバーやコンテキストメニューなどの垂直方向に並んだ 1 列のオブジェクト群に対するゴーストハンティングを用いたカーソルのポインティング性能を評価する。実験 2 では、デスクトップ画面において縦横に等間隔で配置されたアイコン群のように、タイル状に等間隔で配置された複数行複数列のオブジェクト群に対するゴーストハンティングを用いたカーソルのポインティング性能を評価する。

4.3 手 順

はじめに被験者にはボロノイ図に基づくカーソルに共通する事柄や特徴を記した説明書を配布し、それらについて説明した。

実験は実験 1、実験 2 の順番で行い、ひとつの実験はカーソル手法ごとに行った。実験では被験者にその手法による操作に慣れたと感じるまで練習のタスクセットを行わせた後、本番のタスクセットを 2 回行わせた。各タスクセットの実施の前後では十分に休憩を取らせた。2 回の本番のタスクセットを終えた後はその手法に関するアンケートに答えさせた。

1 回のタスクでは、初めにタスクの開始となるオブジェクト（以降「開始オブジェクト」）を選択し、そのつぎにターゲットを選択することを、全てのタスクが終了するまで被験者に繰り返させた。このとき開始オブジェクトとターゲットは選択すべきタイミングでは赤色の線（RGB(255, 0, 0)、太さ 2[px]）で描き、それ以外のときは白色の線（RGB(255, 255, 255)、太さ 1[px]）で描いた。

4.4 被 験 者

被験者は 9 名で、PC の扱いとマウスの操作に習熟した 20 代の大学生および大学院生であった。全員が普段からマウスを右手で使用しており、実験でも右手でマウスの操作を行った。

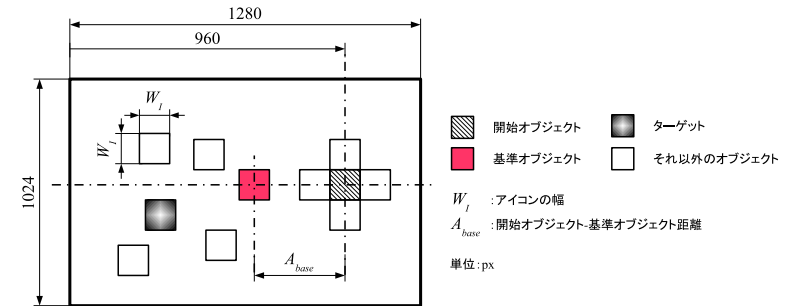


図 6 実験画面

Fig. 6 Display environment.

4.5 画面構成

実験 1 および実験 2 で行われたポインティングタスクの画面構成を図 6 に示す。

実験で用いるオブジェクトは、幅および高さ W_I が 32[px] の正方形とした。画面の右側にある 5 つのオブジェクトは十字をなすように並べられており、中央のオブジェクトが開始オブジェクトである。この配置関係では開始オブジェクトが含まれるボロノイ領域の形状とサイズが開始オブジェクトの形状とサイズに一致するため、ポイントカーソルとそれ以外のカーソルとの間でタスク開始時のターゲットの被選択の判定領域の条件を揃えられる。画面の左側にあるオブジェクト群の中で最も開始オブジェクトに近いオブジェクトの位置を基準オブジェクトと呼ぶ。この基準オブジェクトと開始オブジェクトは画面上で水平に配置されており、両者の間の距離を A_{base} で表す。ターゲットとして指定されるオブジェクトはこちらのオブジェクト群の中に 1 つだけ含まれる（4.7 で詳述）。

4.6 環 境

実験は OS として Microsoft Windows XP を搭載したデスクトップ型 PC で行った。画面を表示する液晶ディスプレイのサイズは対角 48.26[cm]（19 型）、解像度は 1280×1024 [px] であった。カーソルを操作するデバイスにはレーザーマウスを用いた。タスクの成功または失敗を視覚以外の感覚で被験者に知らせるため、ステレオ方式のスピーカを用いた。またマウスの移動量と画面上のカーソルの移動量の比を示す control-display 比は $1/2$ とした。

4.7 独立変数

実験 1 の実験条件となる独立変数を示す。またオブジェクトの配置図を図 7 に示す。

- タスクで用いる手法 CT （7 種類）

- A_{base} (2種類): 256 および 768[px]
- オブジェクト間の距離 D_o (2種類): $W_I (= 32)$ および $3W_I (= 96)$ [px]
- ターゲットの位置 T (3種類): T_1, T_2 および T_3 の位置

なお CT は、ポイントカーソル CT_p , バブルカーソル CT_b , ハリイッポンカーソル CT_1 , ポロノイ図が表示されるハリイッポンカーソル CT_{1v} , ハリセンボンカーソル CT_m , ゴーストハンティングを用いたカーソル CT_g およびポロノイ図が表示されるゴーストハンティングを用いたカーソル CT_{gv} の7種である。 CT_1 はハリ1本のみ表示され最近傍オブジェクトとカーソルが接続されるハリセンボンカーソルのパリエーションである⁷⁾。 CT_g は CT_1 の仕様に加え、最近傍オブジェクト以外のオブジェクトに対応するゴーストを表示する、ゴーストハンティングを用いたカーソルである。 CT_{1v} および CT_{gv} はそれぞれ CT_1 および CT_g の仕様に加え、画面背景にポロノイ境界を表示する。ポロノイ境界は緑色の線 (RGB(0, 255, 0), 太さ 1[px]) で描いた。 CT_m のハリの提示するオブジェクトまでの距離の情報は、文献7) にしたがって実際の距離より 20[px] 長くした。ハリおよび線分は灰色の線 (RGB(89, 89, 89), 太さ 3[px]) で描いた。すべてのカーソルについて、中心座標を表す外観は矢印形でなく、カーソルの中心の座標が明示される十字形を用いた。

カーソルの手法の使用順序は、カーソル外観の類似性による学習効果を考慮して、 CT_{1v} , CT_g および CT_{gv} の間に CT_b , CT_1 および CT_m が挿入されるようにした。また、ポイントカーソルである CT_p は被験者全員が扱いに習熟しており実験結果に影響を与えにくいと考えられるため、順序において最初に使用することとした。

実験1の画面左側のオブジェクト配置を図7(a)に示す。オブジェクト群は垂直方向に並び7個のオブジェクトからなり、図中のオブジェクト T_1, T_2 および T_3 のいずれか1つが目標オブジェクトとなる。以上より実験1の実験条件は $84 (= 7(CT) \times 2(A_{base}) \times 2(D_o) \times 3(T))$ 通りとなり、タスクの総数は84となる。カーソルごとのタスク数は12となり、実験1ではこの12タスクを1つのタスクセットとする。タスクセットにおいては、セットに含まれる全てのタスクがランダムな順序で実施された。

つぎに実験2の実験条件となる独立変数を示す。 CT, A_{base} および D_o については実験1と同じであるが、新たにアイコンの総数 $N (= n_w \times n_h, n_w$: 横方向のオブジェクト数, n_h : 縦方向のオブジェクト数) が加わる。 N は $10 (= 2 \times 5)$ (図7(b)), $20 (= 4 \times 5)$ (図7(c)) および $28 (= 4 \times 7)$ (図7(d)) の3種である。それとともなって T における T_1, T_2 および T_3 の位置もそれぞれ図7(b), (c) および (d) のとおり設定されている。

配置 (図7(b), (c) および (d)) を除き、実験2も実験1同様に行われた。実験2の実験

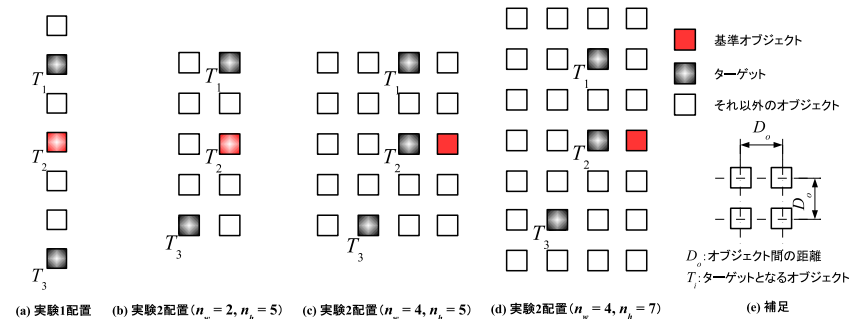


図7 実験のオブジェクト配置
Fig.7 Objects' arrays for experiments.

条件は $252 (= 7(CT) \times 2(A_{base}) \times 2(D_o) \times 3(T) \times 3(N))$ 通りであり、タスクの総数は252となる。カーソルごとのタスク数は36となり、実験2ではこの36タスクを1つのタスクセットとする。

4.8 従属変数

実験1および実験2の従属変数は共通して以下の2つである。

- 選択時間 ST
- エラー率 E

ST はユーザがターゲットを選択するまでにかかる時間のことであり、あるタスクにおいてユーザが開始オブジェクトを選択した時刻と、その後ターゲットを選択した時刻の差である。

本実験におけるエラーとは、ポロノイ図に基づくカーソルでは開始オブジェクトを選択してからターゲットを選択するまでの間に、ターゲット以外のオブジェクトを選択することである。またポイントカーソルを用いた際のエラーとは、上記に加えオブジェクトが存在しない場所で選択操作を行うことを含む。以上よりエラー率 E をつぎのように定義する。

$$E = \frac{N_E}{N_T} \quad (1)$$

ただし N_E はある実験条件において1回以上のエラーを観測したタスクのことであり、 N_T は総タスク数である。

4.9 結果

4.9.1 実験1

カーソルごとの ST の平均を図8に示す。 ST について分散分析を行った結果、 CT ,

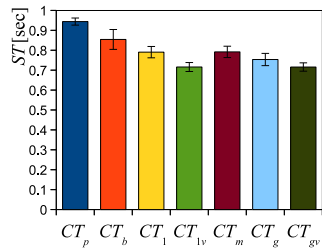


図 8 実験 1 における $ST(CT)$
 Fig. 8 $ST(CT)$ at experiment 1.

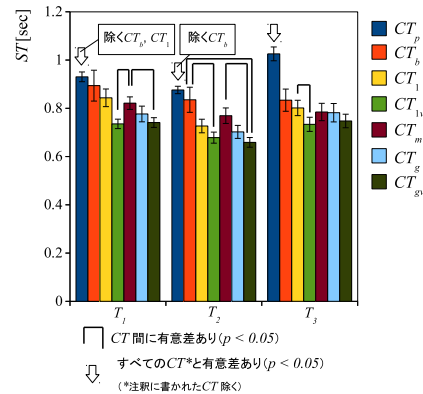


図 9 実験 1 における $ST(CT \times T)$
 Fig. 9 $ST(CT \times T)$ at experiment 1.

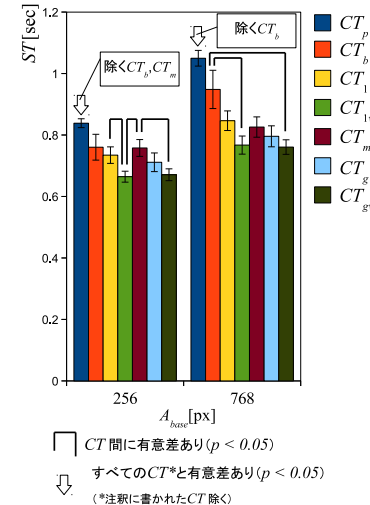


図 10 実験 1 における $ST(CT \times A_{base})$
 Fig. 10 $ST(CT \times A_{base})$ at experiment 1.

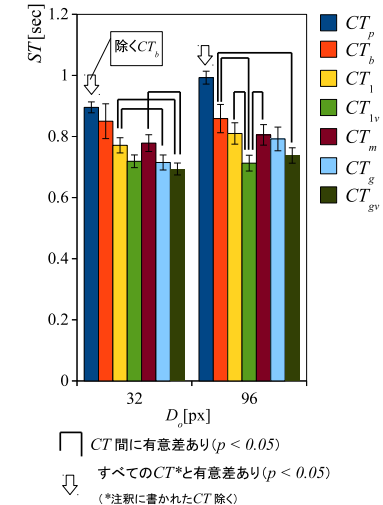


図 11 実験 1 における $ST(CT \times D_o)$
 Fig. 11 $ST(CT \times D_o)$ at experiment 1.

A_{base} , D_o および T の主効果が有意であった (それぞれ $F_{6:102} = 16.81$, $F_{1:17} = 154.9$, $F_{1:17} = 13.40$ および $F_{2:34} = 24.43$. 全て $p < 0.05$). また $CT \times A_{base}$, $CT \times D_o$ および $CT \times T$ における交互作用が有意であった ($F_{6:102} = 6.925$, $F_{6:102} = 3.913$ および $F_{12:204} = 3.471$. 全て $p < 0.05$. 図 9, 図 10 および 図 11 参照).

$CT \times A_{base}$, $CT \times D_o$ および $CT \times T$ について CT の単純主効果を調べたところ, 図のとおりカーソル間で有意差があった (全て $p < 0.05$). 特に CT_g は条件によらず常に CT_p より有意に短く, $D_o = 32$ では CT_1 より有意に短かった.

E はそれぞれ CT_p が 0.014, CT_b が 0.046, CT_1 が 0.023, CT_{1g} が 0.019, CT_m が 0.056, CT_g が 0.028, CT_{gv} が 0.005 であった. E について分散分析を行った結果, CT のみ主効果が有意であり ($F_{6:48} = 4.440, p < 0.05$), CT_m と CT_{gv} の間にのみ有意差があった ($p < 0.05$).

4.9.2 実験 2

カーソルごとの ST の平均を図 12 に示す. ST について分散分析を行った結果, CT , A_{base} , D_o , T および n の主効果が有意であった (それぞれ $F_{6:102} = 9.851$, $F_{1:17} = 454.1$, $F_{1:17} = 44.408$, $F_{2:34} = 91.16$ および $F_{2:34} = 164.5$. 全て $p < 0.05$). 特に CT について, $CT \times A_{base}$, $CT \times N$ における交互作用が有意であった ($F_{6:102} = 2.294$ および $F_{12:204} = 56.16$. 全て $p < 0.05$. 図 13 および 図 14 参照).

$CT \times A_{base}$ および $CT \times N$ について CT の単純主効果を調べたところ, 図のとおりカーソル間で有意差があった (全て $p < 0.05$). 特に $CT \times A_{base}$ における $A_{base} = 768$ で, CT_g が CT_1 および CT_{1v} より有意に長かった. また $CT \times N$ における $N = 28$ では, CT_g が CT_b , CT_1 および CT_{1v} より有意に長く, CT_{gv} が CT_{1v} より有意に長くなった.

E はそれぞれ CT_p が 0.037, CT_b が 0.034, CT_1 が 0.045, CT_{1g} が 0.034, CT_m が 0.035, CT_g が 0.049, CT_{gv} が 0.031 であった. E について分散分析を行った結果, 要因間に主効果はなかった.

4.10 考察

4.10.1 実験 1

選択時間について, ゴーストハンティングは他のカーソル手法と比べ, オブジェクトの配置や距離の条件によらず効率的なポインティングを行えることがわかった. これはゴーストハンティングがパブルカーソルやハリセンボンカーソルと異なり, ゴーストの表示によって常に正確な最短終点の提示を行ったためだと考えられる.

また D_o の違いにより, ボロノイ図を表示しないゴーストハンティングを用いたカーソル

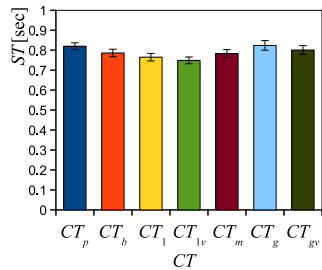


図 12 実験 2 における $ST(CT)$
Fig. 12 $ST(CT)$ at experiment 2.

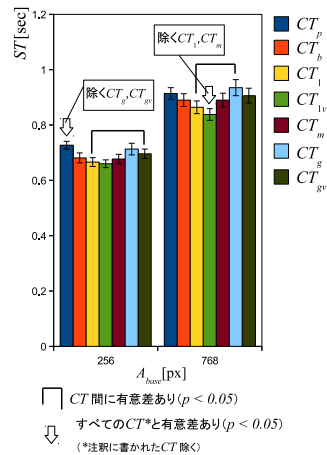


図 13 実験 2 における $ST(CT \times A_{base})$
Fig. 13 $ST(CT \times A_{base})$ at experiment 2.

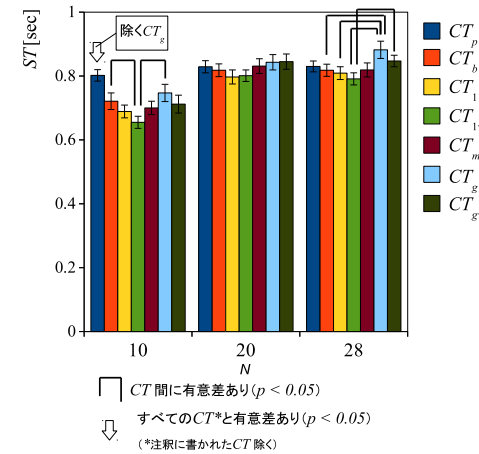


図 14 実験 2 における $ST(CT \times N)$
Fig. 14 $ST(CT \times N)$ at experiment 2.

とハリイボンカーソルの間に有意な差が生じている。これはゴーストが狭い範囲に点在する環境ではオブジェクトとゴーストがより接近して表示されるためだと考えられる。

ところで実験においては、ゴーストやボロノイ図を表示しないカーソル（以降「ガイドのないカーソル」）が、ゴーストやボロノイ図を表示するカーソル（以降「ガイドのあるカーソル」）を事前に扱ったことによる学習効果で、実際の性能よりも良い性能を発揮した可能性がある。タスク成功時におけるカーソルの選択座標位置の分布を見ると、ガイドのないカーソルを使用する際に、最短終点ではない点を狙ってターゲットを選択していると考えられる被験者もいたが、一方でガイドがないにも関わらず最短終点を狙ってターゲットを選択していると考えられる被験者もいた。後者の被験者は、実験において、ガイドのないカーソルを使用する以前にガイドのあるカーソルを使用したことによって、最短終点の推定の学習を行っていた可能性がある。この仮説が正しければ、その学習効果がガイドのないカーソルではたらく、実際に発揮するであろう性能より良くなったと考えられる。

一方、最短終点ではない点を狙ってターゲットを選択していると考えられる被験者の選択座標位置の分布は、ゴーストの有用性を示唆している。図 15 に、最短終点とは関係のない点を狙ってターゲットを選択していると考えられる被験者の CT_{1v} 、 CT_g および CT_{gv} 使用

における選択位置の分布の実例を示す^{*1}。図中の選択位置は、 CT_{1v} と CT_g の間では、平均的に CT_g の方がより最短終点に近い位置に分布しており、また CT_{1v} と CT_{gv} の間では、平均的に CT_{gv} の方がより最短終点に近い位置に分布している。このことからゴーストは、単純にボロノイ図を表示するよりも的確に最短終点をユーザに提示でき、選択座標位置をより最短終点に近づける効果があると考えられる。アンケートの結果においても、ゴーストが「カーソルの動かし方の目安になった」旨を答えた被験者が 6 名 (66.7%) おり、ユーザはゴーストが役に立つと考えていることがわかった。

4.10.2 実験 2

選択時間について、ゴーストハンティングを用いたカーソルはボロノイ図の表示に関わらず他のカーソルと比べ、同等、あるいは有意に長くなる場合が多かった。理由として、オブジェクトの配置がタイル状であるため、各々のボロノイ領域が小さくなってしまい、ターゲットとゴーストの位置がほぼ同じになり、大きな移動距離および方向の差異がなかったことが挙げられる。

*1 T について、 T_1 、 T_2 および T_3 に関する分布を同時に描いていることに注意せよ。

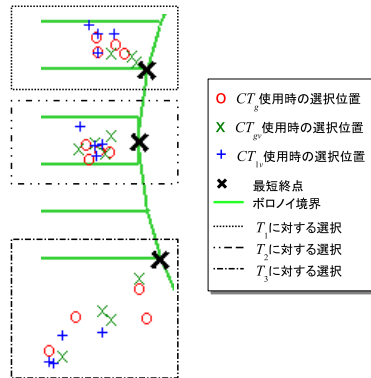


図 15 2名の被験者のカーソル使用時の選択位置分布 ($A_{base} = 256, D_o = 32$ の場合)
Fig. 15 Distribution map using the cursors on the experiment 1 for two participants(in $A_{base} = 256, D_o = 32$).

またアンケートにおいて、3名(33.3%)の被験者が C_g と C_{gv} の両方で「ゴーストの数が増えたことで画面が見辛くなった」旨を挙げていたことから、オブジェクトとゴーストの数の増加により増大したユーザへの認知負荷が性能に大きな影響を与えた可能性がある。

5. おわりに

本研究ではボロノイ図に基づくカーソルの抱える、移動距離が最短となる終点の推定が難しいという問題に対し、ゴーストによってこれを提示するオブジェクト選択支援手法、ゴーストハンティングを提案し、GUIを模した環境において他のカーソルとの比較によって性能を評価した。

その結果、ゴーストハンティングによってカーソルの移動距離が最短となるようにカーソルを動かせることがわかった。この効果はゴーストとオブジェクトが接近している条件下で特に発揮され、オブジェクトの選択時間が短くなった。ただしオブジェクトがタイル状に2列以上配置された環境では選択時間の改善はほとんど見られなかった。この配置ではオブジェクトとゴーストの位置がそれほど変わらなかったことや、ゴーストの数の多さによるユーザへの認知負荷の増大が原因として考えられる。特に後者の原因についてはゴーストやオブジェクトの表示に更なる改良(たとえばゴーストの表示を一部のオブジェクトに対してのみ行うなど)が必要と考えられる。

今回はオブジェクトがタイル状に配置されている環境において性能を評価したが、今後はランダムに配置されているなどの環境における性能も評価したい。

謝辞 研究を行うにあたりご協力いただきました京都市交通局 重森 晴樹氏と被験者の方々に感謝します。

参考文献

- 1) Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F.: Predictive Interaction using the Delphian Desktop, *Proceedings of the 18th annual ACM symposium on User interface software and technology*, New York, NY, USA, ACM, pp. 133-141 (2005).
- 2) 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, *In Proceedings of Interact 2003, Zurich Switzerland*, pp.57-64 (2003).
- 3) Bezerianos, A. and Balakrishnan, R.: The vacuum: facilitating the manipulation of distant objects, *Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, ACM, pp.361-370 (2005).
- 4) 築谷 喬之, 高嶋和毅, 朝日 元生, 伊藤 雄一, 北村 喜文, 岸野 文郎: Birdlime Icon: 動的にターゲットを变形するポインティング支援手法, *コンピュータ ソフトウェア*, Vol.28, No.2, pp.140-152 (2011).
- 5) Kabbash, P. and Buxton, W.: The “Prince” Technique: Fitts’ Law and Selection Using Area Cursors, *CHI*, pp.273-279 (1995).
- 6) Grossman, T. and Balakrishnan, R.: The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor’s activation area, *CHI’05*, pp.281-290 (2005).
- 7) 入江 健一, 重森 晴樹, 倉本 到, 渋谷 雄, 辻野 嘉宏: エリアカーソルの表示領域がポインティング操作に与える影響, *電子情報通信学会論文誌*, Vol.91, No.12, pp.1203-1212 (2008).
- 8) Kobayashi, M. and Igarashi, T.: Ninja cursors: using multiple cursors to assist target acquisition on large screens, *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, New York, NY, USA, ACM, pp.949-958 (2008).
- 9) Hasan, K., Grossman, T. and Irani, P.: Comet and target ghost: techniques for selecting moving targets, *Proceedings of the 2011 annual conference on Human factors in computing systems*, New York, NY, USA, ACM, pp.839-848 (2011).
- 10) 重森 晴樹, 入江 健一, 倉本 到, 渋谷 雄, 辻野 嘉宏: バブルカーソルの GUI 環境への適用と拡張, *インタラクション 2006 論文集*, pp.21-22 (2006).