

航空路カーソル：マウス移動方向と中継オブジェクトによる 操作量低減手法

中辻 智裕^{1,a)} 山本 景子^{1,b)} 倉本 到^{1,c)} 辻野 嘉宏^{1,d)}

概要：通常利用されるポイントカーソルは、オブジェクトのサイズが小さい場合やカーソルとオブジェクトの距離が長い場合、選択にかかる時間が長くなる。この問題の解決を目的として、本稿では「航空路カーソル」を提案する。航空路カーソルでは、ほとんどのオブジェクトは、オブジェクトの方向に少し移動してクリックするだけで選択できる。またカーソルから同じ方向にオブジェクトが複数ある場合、遠い方のオブジェクトは、この操作を繰り返すことで選択できる。これにより、マウス操作によるカーソル移動の距離を縮めることを目指す。さらに、操作に十分慣れさせた上で、航空路カーソル、バブルカーソル、Delphian Desktop、ポイントカーソルで比較実験を行った。その結果、航空路カーソルは、ターゲット付近にオブジェクトが密集している場合に Delphian Desktop より有効で、ターゲット付近にオブジェクトが密集していてかつターゲットまでの距離が 600[pixel] 以上遠いときバブルカーソルより有効なことがわかった。また航空路カーソルはマウス操作によるカーソル移動の距離が最も短くなった。

キーワード：エリアカーソル、ポインティング、GUI、Fitts の法則

1. はじめに

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、一般的に用いられているポイントカーソルを使ったとき、選択の操作の対象となるオブジェクトの中から目標となるオブジェクト (以降「ターゲット」) を選択するためには、ユーザはターゲット上の領域までカーソルを移動しなければならない。つまりポイントカーソルでオブジェクトを選択できる領域 (以降「選択域」) はオブジェクト上である。よって画面上のカーソルの位置とターゲットの位置が離れているほど、ユーザはカーソルを長い距離にわたって移動させる必要がある。

Fitts の法則 [1][2] の式 (1) からポイントカーソルには二つの問題があることがわかる。一つ目はカーソルの位置からターゲットまでの距離が長くなるほどターゲットの選択に時間がかかるという問題である。二つ目はターゲットの選択域が小さくなるほどターゲットの選択に時間がかかる問題である。

$$\begin{cases} T_p = a + b \times ID, \\ ID = \log_2 \left(\frac{A}{W} + 1 \right) \end{cases} \quad (1)$$

この式において、 T_p : カーソルの平均移動時間、 ID : ターゲットまでの移動の難しさ、 A : カーソルからターゲットまでの距離、 W : ターゲットの幅あるいは高さ、 a, b : ユーザの熟練度や手法に依存する定数である。

本研究では一つ目の問題を取り上げる。カーソルの位置からターゲットまでの距離が長くなるほどターゲットの選択に時間がかかるのはユーザがマウスを移動させる距離が長くなるからである。そこで、マウスの移動距離を短縮することでターゲットの選択にかかる時間を短くすることを目的とし、カーソルとターゲットの間にあるオブジェクト数が少ないほどマウスの移動距離を短縮できる新しいカーソル「航空路カーソル」を提案する。

以降、2. では従来のカーソルがもつ問題を述べる。3. で航空路カーソルを提案する。4. では航空路カーソルの性能を他のカーソルの性能と比較する実験とその結果を述べる。

2. 既存研究

ターゲットを選択するためには、ターゲットの選択域までカーソルを動かす必要がある。ポイントカーソルでは、ターゲット上がターゲットの選択域に相当する。具体的には、カーソル中のある領域を、ターゲット上の任意の位置

¹ 京都工芸繊維大学
a) nakatsuji@hit.is.kit.ac.jp
b) kei@kit.ac.jp
c) kuramoto@hit.is.kit.ac.jp
d) tsujino@kit.ac.jp

まで動かすことでターゲットを選択できる。

式 (1) より, W を大きくすれば T_p は小さくなるが, オブジェクトが大きくなると画面に対するそのオブジェクトの占める割合が増えるため, 他のオブジェクトを配置できなかつたり, オブジェクト以外のユーザに提示すべき情報を表示する領域が圧迫されるという問題が起きる. 従って, オブジェクトやユーザに提示すべき情報を複数表示する必要のある状況で W を大きくすることは難しい. 同様に式 (1) より, T_p は A が大きくなるほど長くなる. 常に A を短くするためには, カーソルが移動する領域を狭くすることが考えられる. これはオブジェクトが配置されている領域を狭くすることを意味する. しかし領域を狭くすれば必然的に W は小さくなり, T_p を小さくすることは難しい. このように, 単にオブジェクトの配置を変えるだけではターゲットの選択にかかる時間を短くできない.

この問題を解決するための手法は大きく分けて 2 種類ある. 一つ目はマウスの移動距離 (式 (1) における A に相当) を短縮する手法であり, カーソルの位置に一番近いオブジェクトを選択できるバブルカーソル [3] (エリアカーソル [4] の一種) や, 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移動させる Delphian Desktop[5], Drag and Pop[6] などが含まれる. 本研究で提案する航空路カーソルもこれに含まれる. 二つ目はカーソルが勢い余ってターゲットの選択域を通りすぎてからクリックしてしまうエラーを起りにくくする手法であり, Sticky Icon[7][8] や Birdlime Icon[9] などが含まれる. これらは, 実効的に W を大きくする手法である.

3. 航空路カーソル

筆者らは, 2 で述べた問題点を踏まえ, マウスの移動距離を短縮する手法であるバブルカーソルと Delphian Desktop よりさらにマウス移動距離を短縮することを目指す新たなカーソル「航空路カーソル」を提案する.

3.1 動作

3.1.1 概要

航空路カーソルは航空路モードとポイントモードを持つ. 航空路モードでは, 図 1 のように, カーソルの位置を中心とした各オブジェクトへの向きに扇状に領域を分け, それぞれのオブジェクトに各領域を割り当てる. この領域をオブジェクトの選択域とすることで移動距離を減らす. 以降では領域を分けたときのカーソルの位置を「焦点」と呼び, 焦点を中心とした円形の領域を「焦点領域」, 扇形の領域を「オブジェクト領域」と呼ぶ.

このとき, オブジェクト領域の扇形の角度が一定以上小さくならないように, カーソルから遠い位置のオブジェクトには選択域を与えない. このときの角度を以降「最小オブジェクト角度」と呼ぶ.

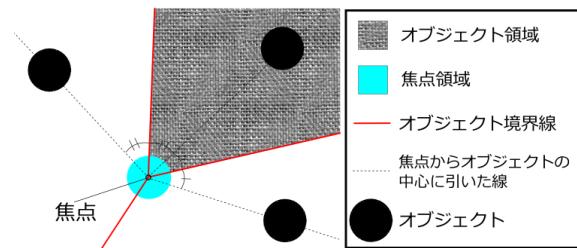


図 1 航空路カーソルの領域の分割方法

一方ポイントモードは, ポイントカーソルと同じ動作をするモードである.

3.1.2 選択域の割り当て

選択域をオブジェクトに割り当てる手順を以下に示す.

- (1) (a) カーソルの位置がオブジェクト上にある場合
 - (i) オブジェクト (O_f) の中心を焦点とする
 - (ii) 焦点を中心とする円形の領域を焦点領域とし, これを O_f の選択域とする
- (b) カーソルの位置がオブジェクト上にない場合
 - (i) カーソルの位置を焦点とする
 - (ii) 焦点を中心とする円形の領域を焦点領域とする
- (2) オブジェクトの集合 S_a, S_b を作る. 最初 S_a は O_f を除く全てのオブジェクトを含み, S_b は空集合である
- (3) S_a に含まれるオブジェクトの中で焦点に最も近いオブジェクトを O_n とし, O_n を S_a から除く
- (4) (a) S_b が空集合でない場合
 - (i) $S_b \cup \{O_n\}$ に含まれるオブジェクト群に対して, 焦点とオブジェクトの中心との間に直線を引き, 隣り合う直線同士が成す角の 2 等分線 (オブジェクト境界線) を焦点から引く
 - (ii) どの隣り合う 2 本のオブジェクト境界線をとっても最小オブジェクト角度以下にならない場合 $S_b = S_b \cup \{O_n\}$ とする
- (b) S_b が空集合の場合
 $S_b = \{O_n\}$ とする
- (5) (a) S_a が空集合でない場合
手順 (3) に戻る
- (b) S_a が空集合の場合
 - (i) S_b に含まれるオブジェクト群に対して, オブジェクト境界線を引く
 - (ii) S_b に含まれるオブジェクトの選択域を, 焦点からそのオブジェクトの中心へ引いた線を間に挟む 2 本のオブジェクト境界線と焦点領域に区切られた焦点領域とする. ただし, 焦点からそのオブジェクトの中心へ引いた線がオブジェクト境界線と重なるとき, つまり焦点から全く同じ方向にオブジェクトが 2 個存在するとき, 図 2 のように焦点から遠い方のオ

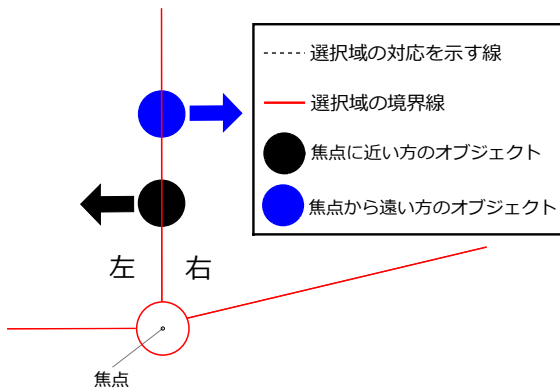


図 2 2 個のオブジェクトが焦点から同方向にあるときのオブジェクト領域の割り振り

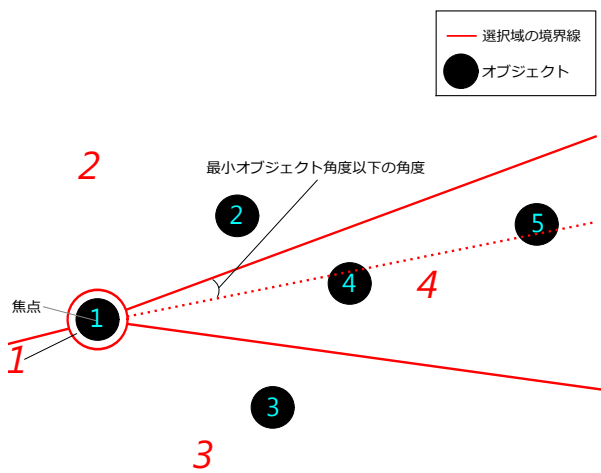


図 3 選択域決定例

オブジェクトの選択域を焦点から見て右側のオブジェクト領域とする

例として、図 3 に示すオブジェクト配置における選択域決定の過程を表 1 に示す。この図の選択域に振られた斜体文字の番号は対応しているオブジェクトを表す。最後の手順 4(a) でオブジェクト 5 を含めてオブジェクト境界線 (点線) を作ると最低オブジェクト角度を下回る角度が発生するので、オブジェクト 5 の選択域は作成されない。

3.1.3 ユーザの操作

航空路カーソルでは、以下のボタン 1、ボタン 2 のクリック操作を用いる。なおボタン 1 が通常の選択に用いられるボタンである。

- ポイントモード
 - ボタン 1 をクリック
カーソルの位置がオブジェクト上だった場合、そのオブジェクトを選択する。オブジェクト上にない場合は何も起きない。
 - ボタン 2 をクリック
カーソルの位置に焦点を作り、航空路モードに切り替える。
- 航空路モード

表 1 選択域決定例の過程

| 手順 | O_f | O_n | S_a | S_b |
|---------|-------|-------|------------|---------|
| 手順 1 | 1 | - | - | - |
| 手順 2 | 1 | - | 2, 3, 4, 5 | ϕ |
| 手順 3 | 1 | 2 | 3, 4, 5 | ϕ |
| 手順 4(b) | 1 | 2 | 3, 4, 5 | 2 |
| 手順 5(a) | 1 | 2 | 3, 4, 5 | 2 |
| 手順 3 | 1 | 3 | 4, 5 | 2 |
| 手順 4(a) | 1 | 3 | 4, 5 | 2, 3 |
| 手順 5(a) | 1 | 3 | 4, 5 | 2, 3 |
| 手順 3 | 1 | 4 | 5 | 2, 3 |
| 手順 4(a) | 1 | 4 | 5 | 2, 3, 4 |
| 手順 5(a) | 1 | 4 | 5 | 2, 3, 4 |
| 手順 3 | 1 | 5 | ϕ | 2, 3, 4 |
| 手順 4(a) | 1 | 5 | ϕ | 2, 3, 4 |
| 手順 5(b) | 1 | 5 | ϕ | 2, 3, 4 |

- = 未定義

(1) 焦点領域上でのクリック

- ボタン 1 をクリック

焦点がオブジェクトの中心に設定されていた場合、そのオブジェクトを選択する。オブジェクトの中心に設定されていない場合は何も起きない。

- ボタン 2 をクリック

焦点を消し、ポイントモードに切り替える。

(2) オブジェクト領域上でのクリック

- ボタン 1 をクリック

カーソルはオブジェクト領域に対応したオブジェクトの中心に移動し、焦点をそのオブジェクトの中心に変更する。かつ、そのオブジェクトを選択する。以降このオブジェクト領域上でクリックによるカーソルの移動を「ジャンプ」と呼ぶ。

- ボタン 2 をクリック

ボタン 1 同様、オブジェクトにジャンプし、焦点の座標を変更する。ただし、そのオブジェクトを選択しない。

3.2 中継によるマウス操作量の削減

航空路カーソルではカーソルの位置から同じ方向にあるオブジェクトが増えてくると、図 3 のように目指すオブジェクトに選択域が割り当てられなくなる。このとき一度の操作ではターゲットを選択できない。この場合、ターゲットとの間にある他のオブジェクト (たとえば 4) を何度かジャンプにより経由してからターゲット選択することになる。この動作を以降では「中継」と呼び、中継するオブジェクトを「中継オブジェクト」と呼ぶ。航空路カーソルでは式 (2) で示される距離 D_a だけマウスを動かせばターゲットを選択できる。

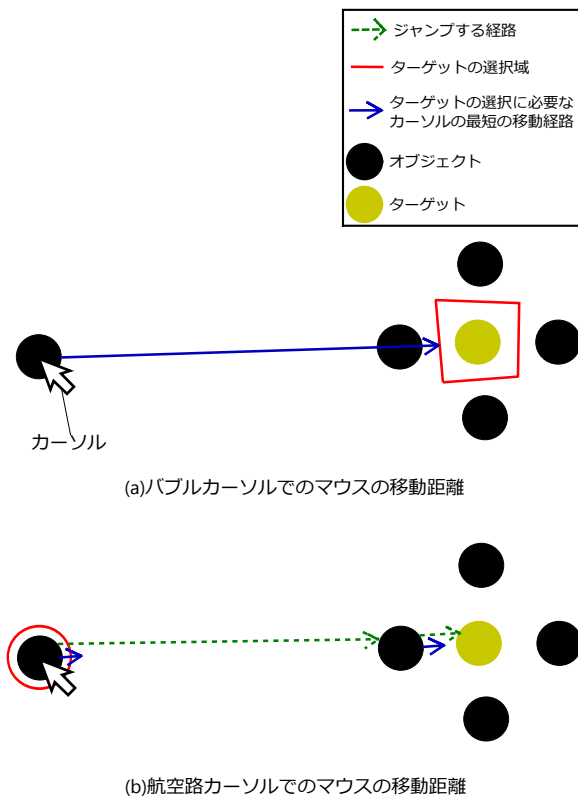


図 4 密集している場合のカーソルの移動距離の比較

$$D_a = (N + 1) \times D_c \quad (2)$$

この式において、 D_a : マウスが移動する必要のある距離、 N : 中継の数、 D_c : 焦点領域の半径である。

D_c はかなり小さく設定してよい。なぜなら、焦点領域はポインティング開始時はカーソルがある領域なので、焦点領域を選択するときの式 (1) のカーソルからターゲットまでの距離 D_a は零と考えて良く、選択域の大きさ W は限りなく小さくてもオブジェクトの選択の難易度は上がらないからである。

中継がある場合を考える。例えば図 4(a) のようにターゲット付近にオブジェクトが数多くあるとき、バブルカーソルではマウスの移動距離を短縮できない問題があった。それに対して航空路カーソルでは、図 4(b) に示すように中継することにより、バブルカーソルと比べてマウスの移動距離を短くできる。このとき、中継を利用してターゲットを選択するには複数回のクリックが必要なので、式 (1) におけるクリック動作に必要な時間とターゲットへの反応時間 a だけ余分に時間がかかる。しかし中継の有無に関わらずターゲットの位置は変わらず、ジャンプ毎にマウスを動かす方向は容易に予測できる。これより 2 回目以降のクリックに必要な反応時間 a は小さくなると考えられる。結果として、中継があってもターゲットの選択にかかる時間は短くなることが予想される。

4. 評価実験

4.1 目的

航空路カーソルのターゲットの選択にかかる時間を他のカーソルと比較し、航空路カーソルの性能と有効な状況を調べる。ただしポイントカーソルと比べてそれ以外のカーソルは使い方や方針が異なる可能性があるため、ユーザにはそれが身に付くまで長時間練習させて十分に慣れさせた上で比較する。

4.2 方法

以下の 4 種類のカーソルを比較する。

- ポイントカーソル (C_p)
- バブルカーソル [3] (C_b)
- Delphian Desktop[5] (C_d)
- 航空路カーソル (C_a)

被験者は 12 人で、パソコンの GUI 環境の操作に慣れている大学生及び大学院生である。

手順は以下のとおりである。

- (1) 被験者のパソコンに練習用のプログラムを配置し、実験内容を説明し、 C_b , C_a , C_d について使い方を説明する
- (2) 被験者に 3 週間、毎日 10 分程度かかる練習用のタスク (各カーソルで 60 回ターゲットを選択させるタスク) を行わせ、 C_b , C_a , C_d の操作に慣れさせる
- (3) ターゲットを選択するタスクを行わせる (4.3 に詳述)

4.3 タスク

4 種類のカーソル (C_p , C_b , C_d , C_a) それぞれでターゲットを選択するタスクを行わせる。各オブジェクトは円形とし、半径は 20[pixel] とした。オブジェクトを円形にするのは、オブジェクトの中心から異なる方向に等距離ある複数の座標からオブジェクト上の領域までの距離が変わらないようにするためである。タスクを開始するオブジェクトを「開始オブジェクト」と呼びこれを選択すると、一つのオブジェクトの色が変わりターゲットとなる。このターゲットを選択するとタスクは完了となる。

様々な条件のオブジェクト配置で評価するために、ターゲット選択タスクで使うオブジェクトの配置は 12 種類ある。それぞれの配置には 3 種のオブジェクトがターゲットとなる 3 通りのタスクがある。よって合計 36 通りのタスクがある。実験ではカーソルごとに 36 通りのタスクを 2 回ずつ、合計 72 回を順序効果が出ないように行わせる。

オブジェクトの配置パターンは、中継の必要性 (F_1)、中継オブジェクトがターゲットに近い (F_2)、カーソル初期位置から見てターゲットの方向にありターゲットより少し遠くにあるオブジェクト (以降「後方オブジェクト」) の

表 2 配置ごとの特徴

| 配置名 | F_1 | F_2 | F_3 | F_4 |
|---------------|-------|-------|----------------------|-------|
| α_{1S} | 不必要 | × | 無し | 400 |
| α_{1L} | 不必要 | × | 無し | 600 |
| α_{2S} | 不必要 | × | 有り | 400 |
| α_{2L} | 不必要 | × | 有り | 600 |
| β_{1S} | 必要 | 近い | 無し | 400 |
| β_{1L} | 必要 | 近い | 無し | 600 |
| β_{2S} | 必要 | 近い | 有り | 400 |
| β_{2L} | 必要 | 近い | 有り | 600 |
| γ_{1S} | 必要 | 遠い | 無し | 400 |
| γ_{1L} | 必要 | 遠い | 無し | 600 |
| γ_{2S} | 必要 | 遠い | 有り | 400 |
| γ_{2L} | 必要 | 遠い | 有り </td <td>600</td> | 600 |

F_1 =中継の必要性

F_2 =中継オブジェクトがターゲットに近いか

F_3 =ターゲットの後ろのオブジェクトの有無

F_4 =ターゲットまでの距離 [pixel]

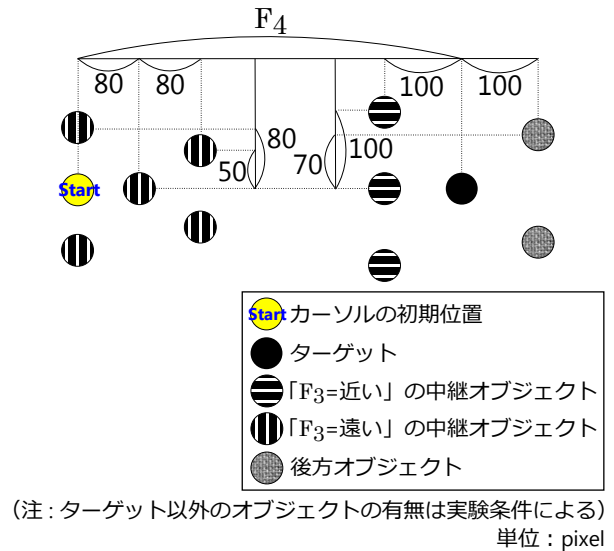


図 5 オブジェクト配置

有無 (F_3), ターゲットまでの距離 (F_4) の 4 特徴により 12 種類に分けられる. F_3 を加える理由は, 航空路カーソルでターゲット後方にオブジェクトがあるとオブジェクト領域が狭まるので, カーソルの操作に影響すると考えたことと, Delphian Desktop で後ろにオブジェクトがないと一定以上速くカーソルを動かせばカーソルがターゲットに移動するため, 操作が容易になると考えたからである.

それぞれの配置の特徴を表 2 に示し, オブジェクト配置を図 5 に示す. この図のオブジェクト配置でターゲットが右にあるが, 他にターゲットが下および左にあるのパターンがある. 配置名は X_{YZ} の形で, X は F_1 と F_2 を示し, Y は F_3 , Z は F_4 を示す. X の各記号は, α が「 F_1 = 不必要」, β が「 F_1 = 必要」かつ「 F_2 = 近い」, γ が「 F_1 = 必要」かつ「 F_2 = 遠い」を示す. Y の各記号は, 1 が「 F_3 = 無し」, 2 が「 F_3 = 有り」を示す. Z の各記号は, L が「 $F_4 = 600[\text{pixel}]$ 」, S が「 $F_4 = 400[\text{pixel}]$ 」を示す.

4.4 実験環境

使用機材は以下の通りであり, 図 6 のように配置した.

- レーザ式マウス (SANWA SUPPLY MA-NOLS5 W)
- パソコン (解像度 1280×800 [pixel])
- プロジェクタ

実験に使う OS は Microsoft Windows 7 64bit である. 画面を横幅 $157 \times$ 縦幅 $91[\text{cm}]$ の大きさで投影した. スクリーンから机の手前の縁までの距離は $110[\text{cm}]$ とした.

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 比を 2.5 とし, マウスの移動量に対してカーソルの移動量が線形に増加するように設定した. これはカーソルからターゲットまでの距離が遠くてもマウスの移動距離を短縮できる航空路カーソルの特性を評価するためである. そのためには大きなディスプレイの環境で評

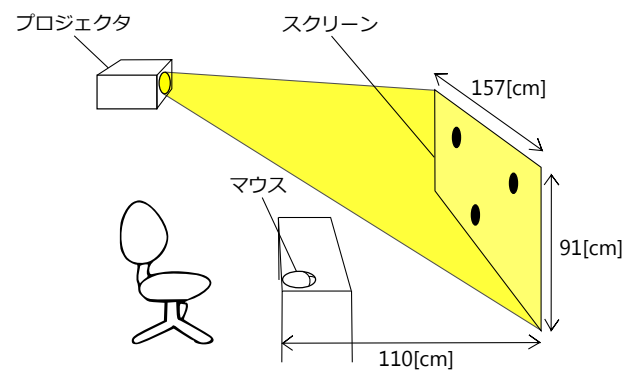


図 6 実験環境

価することが好ましい. そこで 4K ディスプレイ ($3840 \times 2160[\text{pixel}]$) で, control/display 比を 1/1 とした環境を想定し, このような設定とした.

またオブジェクトを選択したときにターゲット以外のオブジェクトを選択したこと (不正解) をユーザに知らせるため, スピーカを用いて不正解音を鳴らす.

4.5 評価尺度

実験条件となる独立変数を示す.

- カーソルの種類
 - 特徴 $F_1 \sim F_4$
 - 被験者
- 実験によって取得する従属変数は以下の通りである.
- タスク遂行時間 T
 - エラー率 E
 - カーソル操作距離 D

タスク遂行時間 T を, 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時間と定義する. なお途中でターゲット以外のオブジェクトを誤って選択 (以降「エ

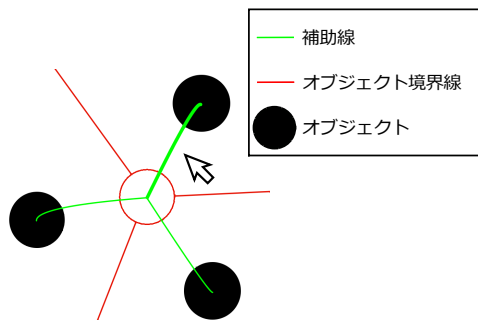


図 7 航空路カーソル表示方法の例

ラー)した試行は T の分析から除外する。
またエラー率 E を以下のように定義する。

$$E = \frac{m}{n} \quad (3)$$

この式において、 m : ターゲットでないオブジェクトを選択したタスクの数、 n : タスク数である。ここで選択とはボタン1をクリックすることとする。

またカーソル操作距離 D は、マウス操作によるのみ動いたものとし、ジャンプによる移動の距離は含まれない。

4.6 航空路カーソルの実装

今回実験で使った航空路カーソルの実装を説明する。ボタン1にはマウスの左ボタンを、ボタン2にはマウスの右ボタンを割り当てる。また、タスク開始時は航空路モードから始まるようにしている。

4.6.1 定数設定

焦点領域の半径は、実験タスクのオブジェクトの円の半径と同じ 20[pixel]、最小オブジェクト角度は 15[deg] である。

4.6.2 表示方法

(1) ポイントモードのとき

カーソルの現在位置に矢印が表示される。

(2) 航空路モードのとき

図7のように、カーソルの現在位置に矢印が表示されるとともに、選択域の境界線を赤色の直線で表示する。また、オブジェクト領域がどのターゲットに対応しているかを表すフィードバックとして、焦点とオブジェクトの中心の間には緑色の曲線を表示する。この緑色の曲線を補助線と呼ぶ。クリックすると選択できるオブジェクトを示すために、カーソルがオブジェクト領域の中にいるときは、対応するオブジェクトへの補助線は別のオブジェクトへの補助線より太く表示される。

補助線を曲線にする理由は、オブジェクト境界線とオブジェクトへの角度が近接している場合、補助線が直線だと見づらくなってしまふからである。補助線は2次元ベジェ曲線で描き、制御点は図8のように、対応したオブジェクト領域の境界線の一部である2本のオ

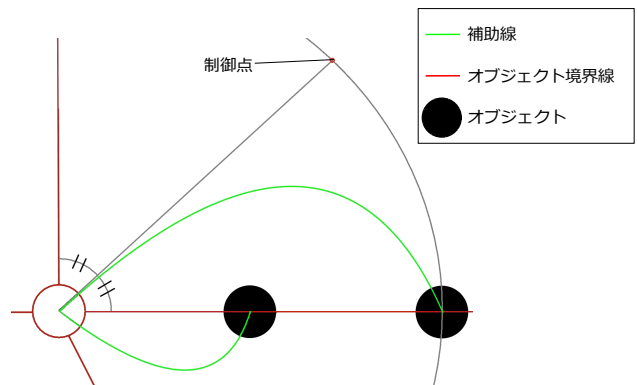


図 8 補助線における二次元ベジェ曲線の制御点の求め方

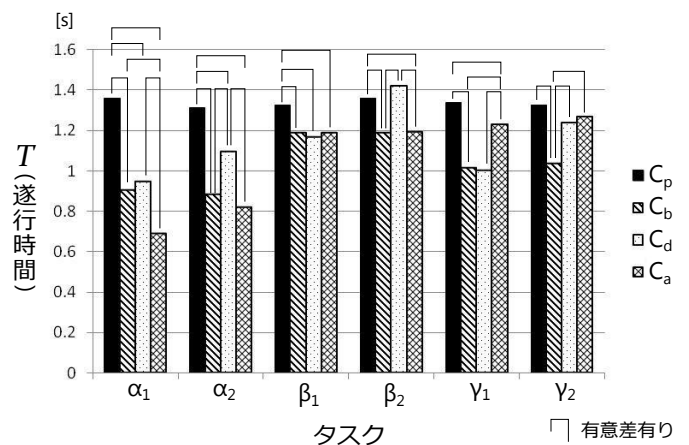


図 9 T カーソル間比較 (F_1, F_2 および F_3 による分類)

ブジェクト境界線が成す角度を2等分した方向に焦点からオブジェクトの間の距離だけ動いた座標に設定している。ただし、求めた制御点がディスプレイから外れる場合は補助線が切れてしまうので、求めた座標に最も近いディスプレイ内の座標に制御点を設定する。

4.7 結果と考察

結果における有意水準を 0.05 とする。なお結果において配置名のうち Y と Z は F_3 や F_4 によって分類せずまとめるとき、省略して表記する。例えば図9では、 F_4 によって分類しないので Z を省略し、 α_1 のように表記している。

4.7.1 航空路カーソルの特徴

図9より、中継の必要なタスク (α_1, α_2) では航空路カーソルが最も速くターゲットを選択できることがわかる。これは図10より、航空路カーソルは、マウス操作によるカーソルの移動距離が中継の有無に関わらずどのカーソルよりも少ないことが理由であると考えられる。

次に中継の必要なタスクに着目する。図9より、ターゲットの手前の近い位置にオブジェクトが存在するとき (β_1)、中継がいらないとき (α) と比較して、 C_a の T は C_b および C_d と有意差が見られないほど大きくなっている。これは式(2)より、中継回数が増えるとその分航空路カーソルはターゲット選択に時間がかかるからだと考えられ

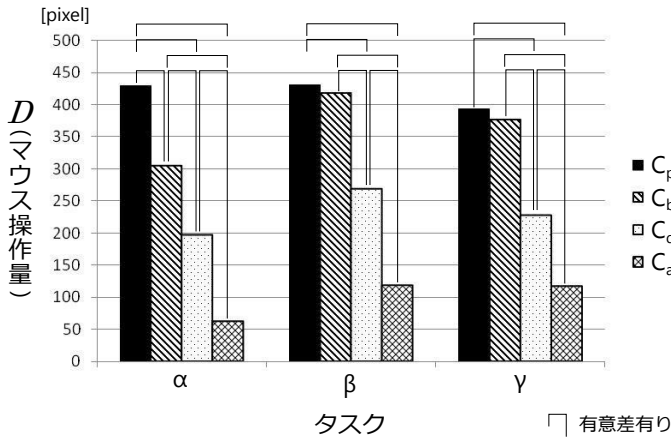


図 10 D カーソル間比較 (F_1 および F_2 による分類)

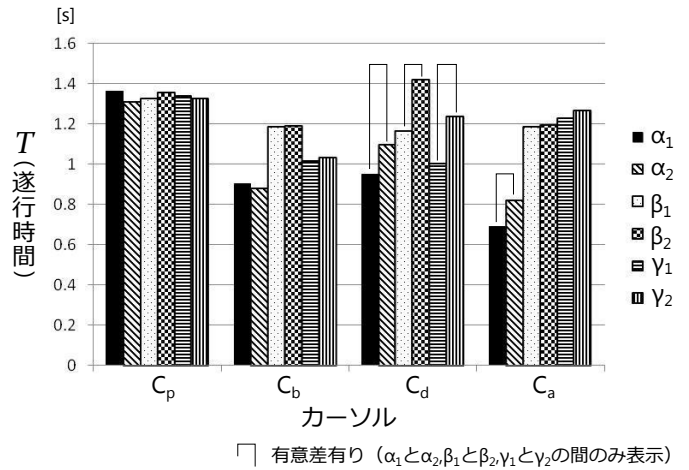


図 12 T タスク間比較 (F_1 , F_2 および F_3 による分類)

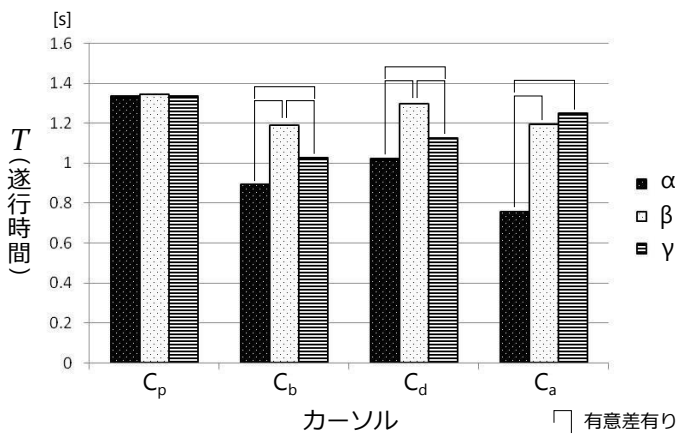


図 11 T タスク間比較 (F_1 および F_2 による分類)

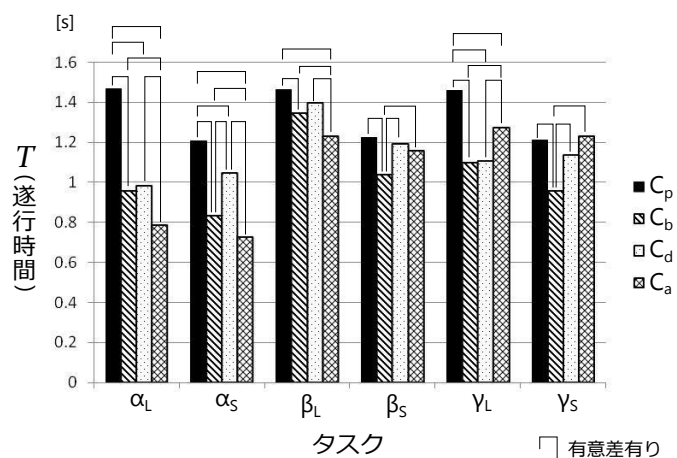


図 13 T カーソル間比較 (F_1 , F_2 および F_4 による分類)

る。このことは、図 11 より、 C_a において、タスク β と γ で T に有意差がない ($p = 0.12$) が、 β および γ は α より有意に T が大きいことから示される。

4.7.2 Delphian Desktop との比較

図 11 より、ターゲットの前後両側にオブジェクトが存在するとき (β_2)、航空路カーソルの方が Delphian Desktop よりも速くターゲットを選択できることがわかる。この理由は、手前か後ろかに関わらず、ターゲットの付近にオブジェクトが増えれば、それら周辺のオブジェクトにジャンプで移動する確率が高くなり、その分遂行時間がかかるという Delphian Desktop の性質によるものと考えられる。

ここでターゲット付近にオブジェクトが増えればその分遂行時間がかかること述べたが、それは以下の結果から示される。まず、図 11 の C_d における β と γ の比較より、Delphian Desktop ではターゲット側にオブジェクトが配置されているとターゲット選択が有意に遅いことがわかる。次に、図 12 の C_d における F_3 の条件間に T の有意差があることより、Delphian Desktop では後方オブジェクトが配置されているとターゲット選択に時間がかかることがわかる。

4.7.3 バブルカーソルとの比較

図 13 より、ターゲット側の手前にオブジェクトが存在する配置で、ターゲットまでの距離が 600[pixel] のとき (β_L)、 C_a は C_b より T が有意に小さいことがわかる。それに対し、ターゲットまでの距離が 400[pixel] のとき (β_S)、 C_a は C_b より T は有意に大きいことがわかる。

これは、バブルカーソルはターゲットが遠い方がターゲットを選択できる領域までの距離が長くなり、ターゲット選択に時間がかかるためであると考えられる。このことは、図 14 より C_b ではいかなる条件においてもターゲットが 600[pixel] 離れているときの方がターゲットが 400[pixel] 離れているときより有意に T が大きいことからわかる。よって、距離が遠くなればその分相対的に航空路カーソルがバブルカーソルより速くなると考えられる。

しかし、図 13 より、ターゲットとの間のカーソル初期位置に近いところにオブジェクトがあるとき (γ)、ターゲットまでの距離が遠い 600[pixel] の場合でも C_b の T が C_a より有意に小さいことがわかる。これは、 β ではターゲットの近くにあったオブジェクトが、 γ ではカーソルの初期位置に近くなっていることから、そのオブジェクトを越え

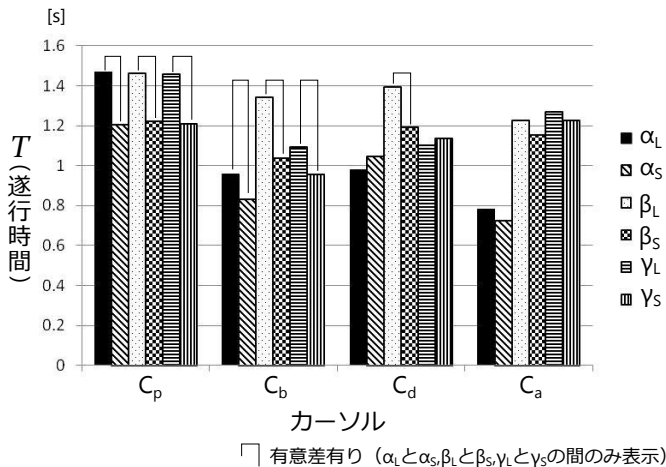


図 14 T タスク間比較 (F_1 , F_2 および F_4 による分類)

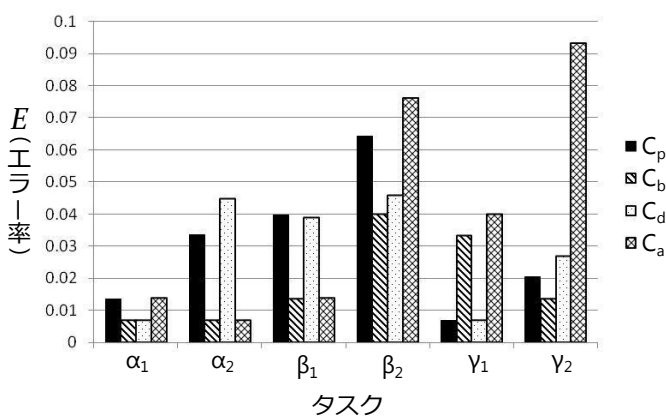


図 15 E カーソル間比較 (F_1 , F_2 および F_3 による分類)

た後のターゲットとカーソルの間にオブジェクトがない領域が広くなり、バブルカーソルに有利になっているためだと考えられる。ただしこの場合も先ほどのバブルカーソルの性質より、ターゲットまでの距離が一定以上離れば航空路カーソルの方が速くなることが期待される。

4.7.4 エラー率

図 15 より、 β と γ では航空路カーソルは後方オブジェクトがあることでエラー率が大きくなるのがわかる。これは後方オブジェクトがあることでオブジェクト領域の角度が小さくなっていることが原因として考えられる。

4.7.5 まとめ

ターゲットが離れていて、かつ周りにオブジェクトが密集しているとき、航空路カーソルは特に有効なことがわかる。これは、バブルカーソルや Delphian Desktop を有効に使おうとするとオブジェクトが密集しないように配置する必要があるが、航空路カーソルでは一部にまとまってオブジェクトが配置してあっても有効であるという利点があることを示している。インタフェースデザインで画面上の広い範囲に均等に選択対象となるボタンなどを配置することは少なく、まとめて配置されているのが一般的であることを考えると、提案手法は有用だと考えられる。

5. おわりに

本研究では、ポイントカーソルの問題点を踏まえ、ターゲットを選択するのに必要なマウスの移動距離を短縮する手法である「航空路カーソル」を提案した。航空路カーソルでは、カーソルの現在位置からの方向でオブジェクトの選択ができる領域を分けて選択域を大きくし、最終的なマウスの移動距離を縮めることができる。その結果としてターゲット選択にかかる時間を短くすることを目指している。ただし、同じ方向に複数オブジェクトが存在するときは、複数回のカーソルボタンのクリックが必要である。

航空路カーソルと既存手法を用いて、GUI 環境上でターゲットを選択するタスクを行わせる比較実験を行った。その結果、ターゲットが離れていてその周りにオブジェクトが密集しているとき航空路カーソルは特に有効なことがわかった。

謝辞 Delphian Desktop に関する貴重なデータをご提供頂きました、東北大学電気通信研究所 高嶋和毅助教に深く感謝致します。

参考文献

- [1] Fitts, P. M.: The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, Vol. 47, No. 6, pp. 381–391 (1954).
- [2] MacKenzie, I. S.: Fitts' Law as a Research and Design Tool in Human-Computer Interaction, *Human-Computer Interaction*, Vol. 7, No. 1, pp. 91–139 (1992).
- [3] 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).
- [4] Kabbash, P. and Buxton, W.: The “Prince” Technique: Fitts' Law and Selection Using Area Cursors, *CHI*, pp. 273–279 (1995).
- [5] 高嶋和毅, 浅野岳史, エフッドシャーリン, 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, pp. 929–938 (2007).
- [6] Collomb, M., Hascoet, M., Baudisch, P. and Lee, B.: Improving drag-and-pop on wall-size display, *Proc. Graphics Interface'05*, pp. 25–32 (2005).
- [7] Cockburn, A. and Brock, P.: Human on-line response to visual and motor target expansion, *Proc. of GI'06*, pp. 81–87 (2006).
- [8] Mandryk, R. L. and C.Gutwin: Perceptibility and utility of sticky targets, *Proc. of GI'08*, pp. 65–72 (2008).
- [9] 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変形するポインティング支援手法の提案, *日本ソフトウェア科学会論文誌*, Vol. 28, No. 2, pp. 140–152 (2011).