

Bubble Gaze Cursor

崔明根^{†1} 坂本大介^{†1} 小野哲雄^{†1}

概要: 視線入力インタフェースにおいて小さなターゲットを選択することは非常に困難であり、容易にするためにはインタフェースデザインの変更か動作時間の増加を強いられる。本稿では、動作時間やデザインの汎用性を維持した状態で小さなターゲットを選択することを容易にすることを目的として、エリアカーソルの一種であるバブルカーソルを視線操作時のカーソルとして用いる手法を提案する。バブルカーソルが視線入力インタフェースにおいてどのような挙動をするのか確認するため、マウスのバブルカーソルと視線のポイントカーソル、そして視線のバブルカーソルに対してポインティングタスクを行った。その結果、視線カーソルをバブルカーソルに変更したものはポイントカーソルよりも常に高速に動作したうえ、ユーザビリティ評価指標もポイントカーソルよりも有意に高かった。したがって視線カーソルにバブルカーソルを用いる提案手法はポインティングを容易に、かつ高速にするうえで有効な手法だといえる。

1. はじめに

従来の人々の視線の動きを計測するアイトラッキングデバイスやシステムは大きく高価であった。しかし、アイトラッキング技術の発展によってデバイスの小型化が進み、かつ安価に手に入れることが出来るようになったことから、ヒューマンコンピュータインタラクション (HCI) の分野においてアイトラッキング技術を活用した研究が活発に実施されるようになった。特によく研究されているのはユーザの注視点をポインタとして操作するような「注視操作システム」である。このシステムはユーザの視線のみを入力として受け取るため、ALS 患者など、手足を動かすことが出来ない人々のマウスに代わる入力インタフェースとして活用されている。また、マウス操作や音声入力、タッチ操作などに対してユーザの視線データを活用することで、通常のマウスよりも高速なポインティングの実現方法[1][2]や効率的で容易な操作方法[3][4]なども提案されている。

しかし、現在の視線操作にはいくつかの問題がある。例えば視線情報を処理するだけでは、ターゲットを注視によって選択する際にユーザが入力したいのか、単に眺めているだけなのかを判断できないため、意図しない入力が行われてしまう問題が挙げられる。この問題はギリシャ神話の触れるもの全てを黄金に変える力を持つミダスになぞらえ、Midas Touch Problem[5]と呼ばれており、注視型入力を採用する際には避けては通れない問題である。また特に顕著な問題として、小さなターゲットの選択が困難であるという点がある。これは固視微動（人間の意図とは関係なく不随意に起こる非常に微小な眼球運動）や安価なアイトラッカーの低いトラッキング精度によって、ターゲット内に視線を固定し続けることが難しいため起こる。固視微動に関しては人間の生理的な反応でありユーザが意図して取り除くことが出来ないため、視線入力インタフェースとして利用

しやすいように視線データを整形する必要がある。しかし、未だに完全な視線データの制御は達成出来ていない。したがって、「小さなターゲットを選択することが難しい」という問題を解決するには、ズームなどの手法を用いて選択を容易にするよう工夫することや[6][7][8][9]、そもそも大きなターゲットのみが配置されるようなインタフェースデザインを採用すること[10]などが必要になる。しかしこれらの手法は、ポインティングを容易にする代わりに、ポインティング速度やインタフェースデザインの汎用性を著しく損なってしまう。

そこで本稿では、視線入力インタフェースを用いる際に画面デザインを変更せずに、カーソルのデザインを変更することで小さなターゲットの選択を容易にするための手法を提案する。具体的には、ユーザのディスプレイ上の注視点を表す視線カーソルにバブルカーソル[11]を応用することで、小さなターゲットに対する容易なポインティングを達成する。現在、一般的な視線入力インタフェースにおいて、視線カーソルにはポイントカーソルが用いられている。この視線カーソルを、ターゲットを必ず1つだけ含む大きさにカーソルを拡大する手法であるバブルカーソルに変更することで、ターゲットの幅を実質的に拡大し、ポインティングを容易にすることを試みる。本稿では従来手法であるバブルカーソルおよび視線カーソルと、提案手法である Bubble Gaze Cursor を比較する実験を行い、この結果を報告する。

2. 関連研究

2.1 視線における動作時間予測式

HCI の分野において最も有名なポインティングタスクの動作時間を求める予測式は Fitts' Law[12][13]である。Fitts' Law は手を用いたポインティングタスクの動作時間に対し

^{†1} 北海道大学

て、ターゲット間の距離 A とターゲットの幅 W の間に相反的な関係が存在することを示したものである。最も一般的に用いられている動作時間 MT の予測式[14]は、

$$MT = a + b \log_2 \left(\frac{A}{W} + 1 \right) \quad (1)$$

であり、 a 、 b は定数である。

この予測式は手を用いたポインティングタスクに対して成り立つと考えられており、主にマウスによるポインティングタスクに対して用いられることが多く、視線による入力に対してもこの法則が成り立つと結論付ける研究[15][16][17][18]も存在する。しかし、視線を用いたポインティングタスクの動作時間が Fitts' Law と一致しない研究が存在する[1][6][19]ほか、Fitts' Law が成り立つとする論文に対して取り扱っている変数の値が不適切であるといった指摘[20][21]や、そもそも視線は弾道的な動きをするため Fitts' Law は視線の動作時間に対する予測式としては不適切であるといった指摘[22]があり、視線入力および視線情報を利用したポインティングタスクに対して Fitts' Law が成り立つとはいえない。

ゆえに、Zhang ら[20]は視線によるターゲットングにおいて、ターゲット間の距離よりもターゲットの幅の方が大きく動作時間に対して影響を与えるという考えから、視線によるポインティングタスクの動作時間の予測式として以下の式を提案した。

$$MT = a + b \left(\frac{e^{\lambda A}}{W - \mu} \right) \approx a + b \times 2000 \frac{e^{A/2000}}{W - 2\overline{AR}} \quad (2)$$

ここで λ はサッケード（ターゲットに視線を向ける際の高速な眼球運動）を表現する定数であり、 μ はジッター（固視微動によって生じる視線の不意の揺らぎ。マイクロサッケードなど）を、 AR (Average Radius)はターゲット内に視線が入ってから軌跡の中心からの平均の距離を、 \overline{AR} は全ての独立変数に対して得られた AR の平均を表している。また、同論文内で $\lambda = 0.0005$ 、 $\mu = 2\overline{AR}$ と置くことで適切な予測式を得ることが出来るうえに、 A や W の単位を考慮せずに用いることができるようになると主張している。

通常のディスプレイで視線を用いてポインティングタスクを行う際、ターゲット間の最大距離はディスプレイの大きさに依存するため長大な距離になることはない。ゆえに、動作時間に対してターゲット間の距離の増減が大きく影響を与えることはないと考えられる。したがって、視線におけるポインティングタスクの動作時間を減少させるための取り組みとして、ターゲット間の距離を減少させるような手法は効果的ではなく、ジッターを減少させるような手法やターゲットの幅を大きくする手法が効果的であると考えられる。

2.2 ジッターを減少させる手法

視線はジッターや高速な挙動などの特徴を有している

ため、アイトラッキングデバイスから得られる生の視線データはノイズが多く、データを加工せずに視線入力インタフェースに用いることは難しい。ゆえに、ジッターノイズを抑えるために、視線データに対する平滑化アルゴリズムが数多くの研究で提案されている[23][24]。

最も単純な平滑化アルゴリズム[25]は式 3 のように視点位置 pos_t を、得られた生の視線データ $gaze_t$ とその1つ前の視点位置 pos_{t-1} を用いて平滑化するアルゴリズムである。

$$pos_t = \alpha gaze_t + (1 - \alpha) pos_{t-1} \quad (3)$$

また、最後の視点位置から N 点の視点位置の平均を現在の視点位置とするようなアルゴリズム[26]や、 N 点の視点位置ごとに重みをかけてから平均を取るようなアルゴリズム[27]なども平滑化のために用いられている。また、これらの平滑化アルゴリズムの欠点であるサッケードの遅延を改善するアルゴリズム[28][29][30]も提案されている。Zhang ら[30]はターゲット内に視線が入ったときのみターゲット中心から離れていく視線の動きに対して式 3 のような遅延フィルタをかける手法 iSR(improved Speed Reduction)を導入することで、視線の高速性を保持した状態でジッターを大幅に減少させた。

上記より、ジッターを減少させる取り組みは成功しており、これらの手法を用いることで視線によるポインティングを容易にすることが出来る。しかし、ジッターの幅 μ は0より小さくなることは無いため $(W - \mu)$ は W より大きくなることはない。ゆえに、ターゲットの幅 W がそもそも小さい場合、ジッターを減少させることは動作時間減少のための対策としては不十分だと考えられる。したがって、小さなターゲットに対するポインティングを根本的に改善するためには、ターゲットの幅 W を増大させるような手法を用いる必要があるといえる。

2.3 ターゲットの幅を増大させる手法

ターゲットの幅を増大させる手法としてまず考えられる手法はズームによるターゲットサイズの拡大である。ズームによるターゲット拡張はマウス操作において動作時間を有意に減少させ、かつズーム後のターゲットの幅に対して Fitts' Law が成り立つ[31][32]ことが確認されている。

Skovsgaard ら[7]は、注視点に対して離散的なズームを複数回行うことでターゲットを拡張する手法を提案した。しかし、この手法を視線操作インタフェースに組み込む場合、ズームによって画面を拡大している場合にズーム領域外の情報が損失してしまうことや、ターゲット選択にかかる時間のほかにズーム操作のための時間が必要になることによるタスク達成時間の増加などが起こると考えられる。Ashmore ら[8]はズーム領域外の情報の損失を防ぐために魚眼レンズによる局所的なズームによってターゲット拡張を行う手法を提案した。この手法では、魚眼レンズ内の文字などの情報の損失を防ぐために、視線がある程度固定され

た後に魚眼レンズによる拡大が行われる。しかし、この手法も Skovsgaard ら[7]の手法と同様にズームが発生するまでの時間がターゲット選択のほかに必要となるため、操作時間が大きくなってしまう。

また、ズームを用いずにターゲットの幅を実質的に増大させるような手法も存在する。Špakov ら[9]は視線によってメニューを選択する際に、一定時間凝視したあとにメニューの大きさが拡大されるようなインタフェースデザインを提案した。この手法を用いることでエラーレートを 1/6 に改善することに成功した。しかし、同時に 39%の動作時間の増加が引き起こされた。Minitas ら[6]は視覚的なターゲットの大きさよりも大きな判定領域を用意することで実質的なターゲット拡張を行う手法を提案した。この手法によってポインティングタスクの作業時間は有意に減少したが、この手法は静的な処理であるため、インタフェースのデザインの変更のたびに判定領域を手動で設定する必要がある。また、判定領域内に入力判定のあるオブジェクトを複数配置することが出来ないため、インタフェースデザインに制限がかかってしまう。

上記で触れた手法は、ターゲットサイズを拡大しポインティングを容易にするが、動作時間の増大やデザインの汎用性の損失につながる。特に動作時間に関しては、視線が単一的な入力しか行えないという欠点から、容易なポインティングとトレードオフの関係になっている。ゆえに、本研究では動作時間やデザインの汎用性を維持し、かつ小さなターゲットのポインティングを容易にする手法を提案する。

2.4 視線カーソル

視線カーソルとは、ディスプレイ上におけるユーザの視点を可視化したものであり、様々な形状が用いられている。一般的な視線カーソルの形状は点状のカーソルである。この形状は従来の視線入力によるポインティング手法に多く用いられるほか、カーソル自体が小さいことから、アイタッピング[33]のように密集したターゲット群を選択するインタフェースなどに対して用いられている。他の形状としては、半径の大きな円形のカーソル[3][4]や四角形のカーソル[27]などが用いられている。近年商業的に最も用いられている Tobii 社のアイトラッカーでは、視線カーソルとしてカーソルの動作によって形状が変わるような泡状のカーソルを採用している。この視線カーソルはカーソル自体がある程度の大きさを持つことでアイトラッカーの低い精度を保証している。しかし、カーソルの大きさよりも小さなターゲットが密集している場合、視線カーソル内に複数のターゲットを含んでしまう可能性があるほか、ターゲットが選択されているかどうかを視線カーソルの状態から判別することが出来ないため、視線カーソル内にターゲットが内包されている状態でもターゲットが選択されないことが

起こり得る。

また、単純にディスプレイ上の視線の位置を表すほかに、視線カーソル自体に情報を付加する研究や、インタフェースデザインの 1 つとする研究も存在する。EyeDraw[27]は視線カーソルを絵筆にすることによって絵を描くためのインタフェースを提案し、その中で視線カーソルの色を、ただ見ている場合は緑、線を引いている場合は赤色に視線カーソルの色を変更することで現在の視線カーソルの状態を表現した。ceCursor[34]は GUI を視線で操作するために視線カーソルの形状を上下左右の 4 方向を指し示すような形状にし、アイコンなどの選択を、視線カーソルを離散的に移動させることによって容易にした。

上記のように視線カーソルは様々な実験で用いられているが、視線入力によって達成したい目的がターゲット選択ではないことが多く、視線カーソル自体に焦点を当てるような研究は少ない。

2.5 バブルカーソル

ターゲット選択において、点のカーソルを用いて面のターゲットを選択することと、面のカーソル（エリアカーソル）[35][36]を用いて点のターゲットを選択することは動作時間の観点から同義である。しかし、エリアカーソルはカーソル自体が面積を持つため、カーソルが複数のターゲットを含んでしまう状況が起こり得る。この問題を解消するために提案された手法がバブルカーソル[11]である。バブルカーソルは常にターゲットを 1 つだけ含むような大きさにカーソルの大きさを動的に変更し続ける円形のエリアカーソルである。したがって、エリアカーソルの特性より、バブルカーソルを用いることでターゲットの幅をバブルカーソルの幅に拡大することと同等の効果を得ることが出来る。

バブルカーソルによって拡大されるターゲットの実質的な幅 EW (Effective Width) はボロノイ図で表現することが出来る。ボロノイ図とは平面内の各母点の再近隣領域（ボロノイ領域）を分割した図である。つまり図 1 におけるボロノイ領域がバブルカーソルによって拡張されたターゲットの実質的な大きさとなる。

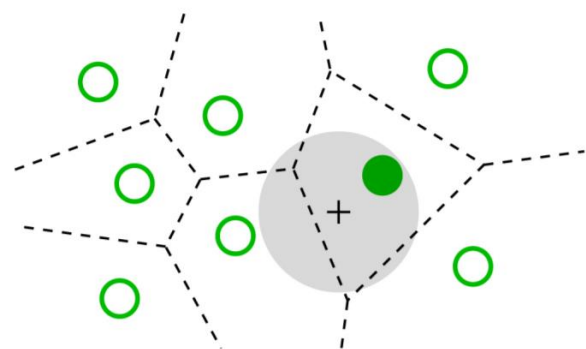


図 1 Grossman ら[11]によるバブルカーソルとボロノイ図

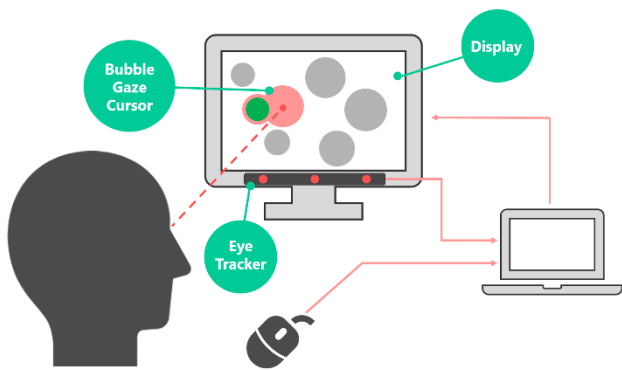


図2 実験システム

3. 提案手法 : Bubble Gaze Cursor

本稿では小さなターゲットに対するポインティングを容易にすることを目的として、視線カーソルにバブルカーソルを用いる手法(Bubble Gaze Cursor)を提案する。

Bubble Gaze Cursor を用いる利点として、ターゲットの幅を増大させる既存手法の難点であった動作時間の増大とインタフェースデザインの制限を回避しつつ、小さなターゲットに対するポインティングを容易にすることが挙げられる。バブルカーソルはターゲットを必ず1つだけ含む大きさに変形するカーソルであるため、カーソルの移動とともにカーソルは常に大きさを変える。したがって、ターゲットの幅を増大させる処理が常に行われている状態でありターゲット拡大の処理に時間がかからないため、動作時間を維持してポインティングを行うことが出来ると考えられる。また、カーソルによる処理であるため、Bubble Gaze Cursor を用いることで視線入力インタフェースに小さなターゲットの配置を可能にすると考えられる。

4. 実験

マウス操作時においてバブルカーソルはポインティングタスクの動作時間を減少させ、かつカーソルの幅を用いて Fitts' Law で定式化することが可能であることが確認されている[11]。しかし、視線操作時において、バブルカーソルが正常に動作するか、また動作時間を予測できるかどうかについては十分に研究されていない。ゆえに、視線カーソルをバブルカーソルに変更することによってどのような影響を及ぼすか検証するために、ポイントカーソル(視線)とバブルカーソル(視線とマウス)を用いてポインティングタスクを行う。

4.1 使用機器と実験参加者

本実験は、2.30Ghz の Intel(R) Core(TM) i5-6200U の CPU を搭載したパソコンで行った。ディスプレイはフル HD の 27 インチ (1920 × 1080 ピクセル; 1 ピクセルあたり約 0.31

mm) のディスプレイを使用した。アイトラッカーは Tobii EyeX を使用し、実験に用いたソフトウェアは C# で作成した。実験システムを図 2 に示す。

実験には主に大学生(男性 14 人、女性 4 人、平均年齢 23.9 歳)の 18 人が参加した。全ての実験参加者はコンピュータやマウス操作を日常的に用いていたが、実験参加者の中でアイトラッキングを実際に使用したことがある者は 4 名で、それ以外はアイトラッキングを使ったことがなかった。また報酬として、実験参加者にそれぞれ 1000 円分の Amazon ギフトカードを渡した。

4.2 実験手順

はじめに、実験参加者をディスプレイから 70 cm ほど離れた位置に座らせ、目がディスプレイの中心に位置する高さに椅子を調整させた。その後、これから行うタスクの内容を紙に印刷された図などを用いて説明した後、視線によるタスクを行う場合はアイトラッカーのキャリブレーション(ディスプレイ上の視点位置を個人に合わせて調整すること)を行った。実験開始前には視線入力やタスクの内容に慣れるために、これから行うタスクと同様のポインティングタスクを、パラメータを実際のタスクとは異なった値に設定し、実験参加者が挙動を理解したと宣言するまで練習させた。そして練習が終了した後に実験タスクを開始した。

全ての操作手法において、まず”START”と印字されたボタンがディスプレイ上に提示される。それをマウスでクリックすることでタスクを参加者自らが開始する。その後、図 3 のように以下の試行を一定回数繰り返す。

- ① 画面上に緑色の円形のターゲットのみが画面に提示されるので、このターゲットを選択する。選択はマウスの場合はクリック、視線の場合はターゲットを注視することによって選択される。全ての操作手法において、このターゲットを選択する時間は計測されておらず、位置合わせを目的としてこの試行を行う。
- ② 試行①直後、試行①のターゲットの位置からディスプレイ中心を通る垂直二等分線に対称な位置にターゲット群が生成される。ターゲット群は緑色の円形ターゲットを四方に囲むように灰色の円形ターゲットが配置されたものである。灰色のターゲットは緑色のターゲットと同じ大きさで、緑色のターゲットに対応するボロノイ領域の幅がバブルカーソルの最大幅と一致するように配置されている。実験参加者はこのターゲット群の中から緑色のターゲットを試行①と同様の方法で選択する。
- ③ 反対側に同様のターゲット群が生成されるので同様に緑色のターゲットを選択する。
- ④ 試行①に戻る。

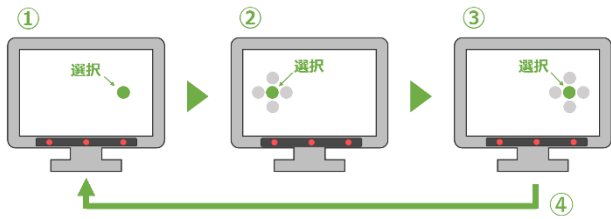


図3 実験タスクの手順

緑色のターゲットの中心には 10 ピクセルの濃緑色の点が描画されている。これは注視すべき凝視点が存在しない場合ターゲットを注視し続けることが難しい[37]ため、配置した。実験参加者にはポインティングタスクの際に視線カーソルではなくこの濃緑色の点を見るように促した。また、バブルカーソルの色は薄い赤色で描画し、マウスのバブルカーソルの中心にはクロスカーソルを、視線のバブルカーソルの中心には 6 ピクセルの濃赤色の点を描画した。ポイントカーソル（視線）は視線のバブルカーソルの中心に描画した濃赤色の点をカーソルとして用いた。

視線入力の際の滞留時間は、今回は単純なポインティングタスクを行うことから、Zhang ら[38]の実験より 600ms とした。また、試行①で表示されるターゲットに対しては、実験の高速化のために滞留時間を 450ms とした。

4.3 実験デザイン

実験は参加者内実験計画を用いて行った。独立変数はカーソルの種類 CT (Cursor Type) (BubbleMouse: マウスカーソルにバブルカーソルを用いた手法, PointGaze: 視線カーソルにポイントカーソルを用いた手法, BubbleGaze: 視線カーソルにバブルカーソルを用いた手法), ターゲット間の距離 A (350, 550, 800 ピクセル), ターゲットの幅 W (50, 65, 80 ピクセル), バブルカーソルの最大の幅 EW (100, 120, 140 ピクセル) とした。 CT の図解を図 4 に示す。独立変数 A , W , EW の値は式 2 の難易度を表す項である ID_{EYE} の値がばらつくように設定した。 ID_{EYE} の値は $2000 e^{A/2000} / (W - 2\bar{AR})$ である。また、従属変数として全てのカーソルタイプに対して動作時間 MT (Movement Time) と精度を求め、視線による入力を行う CT に対してはターゲットにカーソルが触れるまでの時間 EMT (Eye Movement Time) と、Zhang ら[30]のジッターの幅を求める方法を参考にジッターの半径 AR (Average Radius) を求めた。精度は灰色のターゲットを選択した場合と、試行時間に 5 秒以上かかった場合を失敗とし、計算した。

実験全体の拘束時間はおよそ 90 分であり、 CT 条件ごとに 9 ブロックのタスクを行ってもらった。それぞれのブロックごとに、実験参加者はランダムな順に提示される A , W , EW の 27 の組み合わせに対して実験タスクを行った。休憩は図 3 の試行①のタイミングかブロック完了ごとに実験参加者の自己判断で行った。

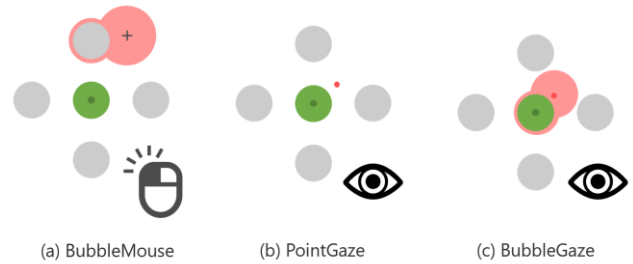


図4 CT の図解

また、それぞれの CT 条件完了後にメンタルワークロードとユーザビリティを計測するために NASA Task Load Index (NASA-TLX) [39] と System Usability Scale (SUS) [40] によるアンケートを実施し、実験終了後に各手法に関して気付いた点をアンケートおよびインタビューで尋ねた。

4.4 実装

バブルカーソルを、滞留時間を用いた視線入力インタフェースに導入するにあたって変更すべき点がある。バブルカーソルはターゲットを必ず 1 つ含むような大きさに変形するカーソルである。ゆえに、視線カーソルに対してそのままバブルカーソルを移植する場合、常にいずれかのターゲットを注視している状態になるため、誤入力が頻発することとなる。この問題の解決のためにバブルカーソルに最大半径を導入し、どのターゲットも選択されていない状態を作った。今回の実験ではバブルカーソルの最大半径を EW とした。

また、サックード時にバブルカーソルの描画が滑らかに行われず、離散的に描画されたため、式 3 の平滑化アルゴリズムを導入した。さらにジッターの抑制と過剰なエラーを抑えるために Zhang ら[30]の iSR を採用した。

4.5 結果

それぞれの従属変数に対して 4 要因分散分析を行った。多重比較には Bonferroni の手法を用いた。

4.5.1 動作時間 (MT)

MT に対して CT ($F_{2,34} = 1590.92$, $p < 0.001$), A ($F_{2,34} = 721.37$, $p < 0.001$), W ($F_{2,34} = 101.86$, $p < 0.001$), EW ($F_{2,34} = 36.76$, $p < 0.001$) がそれぞれ有意であった。また、1 次の交互作用として $CT \times A$ ($F_{4,68} = 8.20$, $p < 0.001$), $CT \times W$ ($F_{4,68} = 81.31$, $p < 0.001$), $CT \times EW$ ($F_{4,68} = 9.19$, $p < 0.001$) が、2 次の交互作用として $CT \times A \times EW$ ($F_{8,136} = 2.30$, $p < 0.05$) が有意であった。すべての CT は A に有意な差が確認された (BubbleMouse ($F_{2,34} = 926.70$, $p < 0.001$), PointGaze ($F_{2,34} = 142.80$, $p < 0.001$), BubbleGaze ($F_{2,34} = 359.81$, $p < 0.001$)) が、 W は PointGaze のみに有意な差が (BubbleMouse ($F_{2,34} = 0.33$, $p = 0.72$), PointGaze ($F_{2,34} = 101.47$, $p < 0.001$), BubbleGaze ($F_{2,34} = 0.71$, $p = 0.50$)),

EWはBubbleMouseとBubbleGazeにのみ有意な差が確認された (BubbleMouse($F_{2,34} = 45.03$, $p < 0.001$), PointGaze($F_{2,34} = 0.27$, $p = 0.77$), BubbleGaze($F_{2,34} = 49.58$, $p < 0.001$)). MTの平均は BubbleMouse が 793.92ms, PointGazeが1529.90ms, BubbleGazeが1269.81msであった. WとEWの組み合わせすべてに対するCTごとのMTを図5に示す.

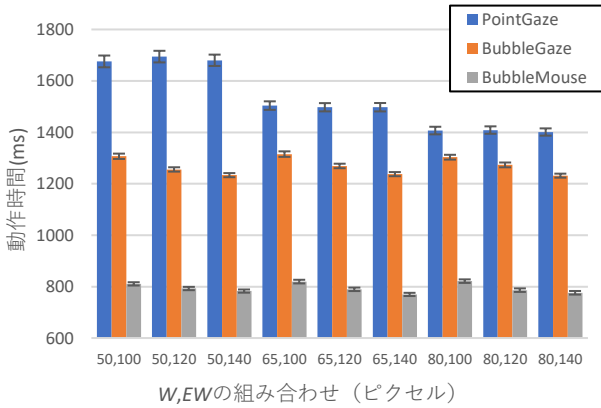


図5 WとEWの組み合わせに対するMT

図6にはPointGazeとBubbleGazeに対して、式2の動作時間予測式に則って、MTの値から線形回帰分析した結果を示した. 図6の横軸 ID_{EYE} (Index of Difficulty EYE)は式2より, $2000 e^{A/2000} / (W - 2\bar{AR})$ とした. PointGazeのモデルの適合度 R^2 の値は0.89であり, BubbleGazeは0.54, PointGazeとBubbleGazeのデータを合わせて線形回帰した場合は0.91であった.

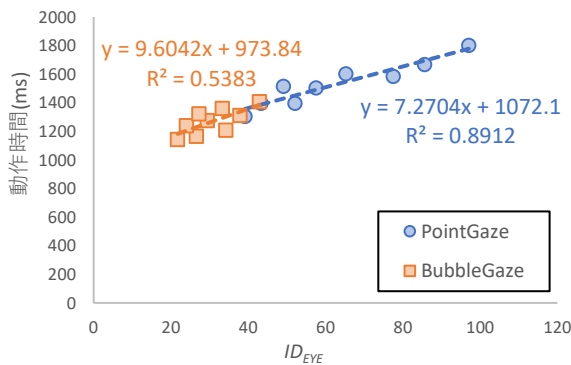


図6 ID_{EYE} に対するMT

4.5.2 ターゲットに触れるまでの時間 (EMT)

EMTに対してCT($F_{1,17} = 922.99$, $p < 0.001$), A($F_{2,34} = 600.17$, $p < 0.001$), W($F_{2,34} = 146.33$, $p < 0.001$), EW($F_{2,34} = 13.86$, $p < 0.001$)が有意であった. また, 1次の交互作用としてCT×A($F_{2,34} = 6.70$, $p < 0.01$), CT×W($F_{2,34} = 141.76$, $p < 0.001$), CT×EW($F_{2,34} = 16.46$, $p < 0.001$), A×EW($F_{4,68} = 2.90$, $p < 0.05$)が有意であった. 両方のCTはAに有意な差が確認された (PointGaze($F_{2,34} = 220.27$, $p < 0.001$), BubbleGaze($F_{2,34} = 933.97$, $p < 0.001$))

が, WはPointGazeのみに有意な差が (PointGaze($F_{2,34} = 162.56$, $p < 0.001$), BubbleGaze($F_{2,34} = 0.80$, $p = 0.46$)), EWはBubbleGazeにのみ有意な差が確認された (PointGaze($F_{2,34} = 0.24$, $p = 0.79$), BubbleGaze($F_{2,34} = 94.45$, $p < 0.001$)). EMTの平均はPointGazeが819.67ms, BubbleGazeが612.15msであった. WとEWの組み合わせすべてに対するCTごとのEMTを図7に示す.

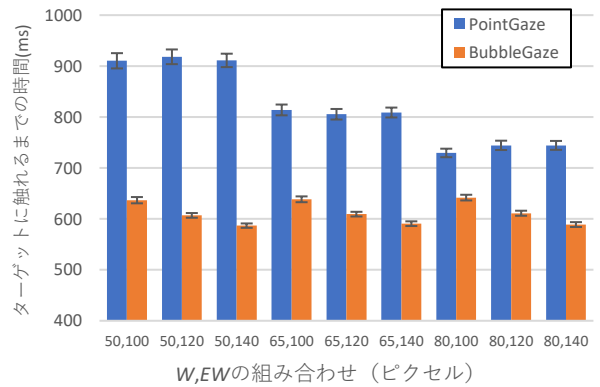


図7 WとEWの組み合わせに対するEMT

図8にはPointGazeとBubbleGazeに対して、式2の動作時間予測式に則って($\mu = 0$ とすることでEMTに対する動作時間予測式を計算することが出来る[20]), EMTの値から線形回帰分析した結果を示した. PointGazeのモデルの適合度 R^2 の値は0.85であり, BubbleGazeは0.59, PointGazeとBubbleGazeのデータを合わせて線形回帰した場合は0.90であった.

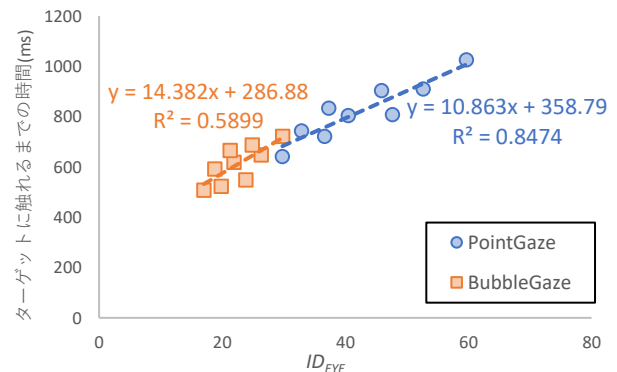


図8 ID_{EYE} に対するEMT

4.5.3 ジッターの半径 (AR)

ARに対してCT($F_{1,17} = 206.73$, $p < 0.001$), A($F_{2,34} = 35.57$, $p < 0.001$), W($F_{2,34} = 3.57$, $p < 0.05$), EW($F_{2,34} = 5.29$, $p < 0.01$)が有意であった. また, 1次の交互作用としてCT×A($F_{2,34} = 6.98$, $p < 0.01$), CT×W($F_{2,34} = 3.46$, $p < 0.05$), CT×EW($F_{2,34} = 8.91$, $p < 0.001$)が有意であった. 両方のCTはAに有意な差が確認された (PointGaze($F_{2,34} = 9.52$, $p < 0.001$), BubbleGaze($F_{2,34} = 26.33$, $p < 0.001$))が, WはPointGazeのみに有意な差が

(PointGaze($F_{2,34} = 7.76, p < 0.01$), BubbleGaze($F_{2,34} = 0.23, p = 0.79$), EW は BubbleGaze にのみ有意な差が確認された(PointGaze($F_{2,34} = 1.82, p = 0.18$), BubbleGaze($F_{2,34} = 5.98, p < 0.01$)). AR の平均は PointGaze が 9.63 ピクセル, BubbleGaze が 15.22 ピクセルであった. ターゲットの幅 (PointGaze は W , BubbleGaze は EW) とターゲット間の距離 A に対する AR を図 9 に示す.

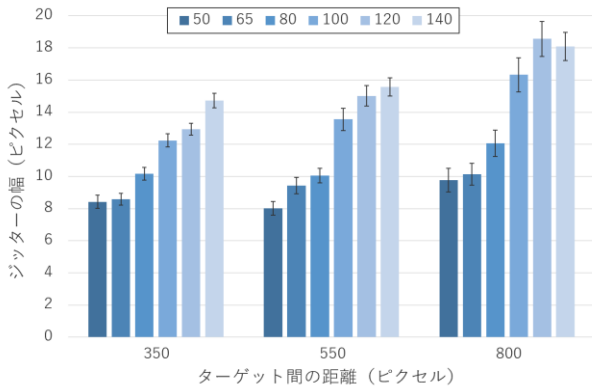


図 9 ターゲットの幅(W,EW)とターゲット間の距離 A に対する AR

4.5.4 精度

精度に対して W ($F_{2,34} = 7.48, p < 0.01$), EW ($F_{2,34} = 10.08, p < 0.001$)が有意であった. また, 1 次の交互作用として $CT \times A$ ($F_{4,68} = 9.13, p < 0.001$), $CT \times W$ ($F_{4,68} = 23.99, p < 0.001$), $CT \times EW$ ($F_{4,68} = 4.70, p < 0.01$), $A \times EW$ ($F_{4,68} = 2.86, p < 0.05$)が, 2 次の交互作用として $CT \times A \times W$ ($F_{8,136} = 3.41, p < 0.01$), $CT \times A \times EW$ ($F_{4,68} = 2.87, p < 0.01$)が, 3 次の交互作用として $CT \times A \times W \times EW$ ($F_{16,272} = 2.26, p < 0.01$)が有意であった. 精度の平均は BubbleMouse が 98.67%, PointGaze が 98.15%, BubbleGaze が 98.27%であった.

4.6 NASA-TLX および SUS

NASA-TLX の平均スコアは BubbleMouse が 25.56, PointGaze が 64.15, BubbleGaze が 39.44 であった. これらの値には有意な差が存在し($F_{2,34} = 25.43, p < 0.001$), それぞれの CT 間にも有意な差が存在した (BubbleMouse-PointGaze ($p < 0.001$), PointGaze-BubbleGaze ($p < 0.001$), BubbleGaze-BubbleMouse ($p < 0.05$)).

SUS の平均スコアはそれぞれ, BubbleMouse が 78.19, PointGaze が 55.56, BubbleGaze が 69.44 であった. これらの値には有意な差が存在し($F_{2,34} = 12.24, p < 0.001$), PointGaze と他の CT に有意な差は存在したが (BubbleMouse-PointGaze ($p < 0.01$), PointGaze-BubbleGaze ($p < 0.01$)), BubbleMouse と BubbleGaze の間に有意な差は存在しなかった ($p = 0.062$).

5. 議論

本稿では, 小さなターゲットのポインティングを容易にする手法として視線カーソルにバブルカーソルを用いる手法を提案した. バブルカーソルが視線入力インタフェースにおいて正常に動作するか検証するため, ポイントカーソル (視線) とバブルカーソル (視線とマウス) を用いてポインティングタスクを行った. その結果, 以下の点が明らかになった.

1. 動作時間 (MT) は全ての条件の組み合わせに対して, BubbleMouse, BubbleGaze, PointGaze の順に速く, それぞれの MT の平均は, 793.92ms, 1269.81ms, 1529.90ms であった.
2. ターゲットに触れるまでの時間 (EMT) (PointGaze, BubbleGaze に対して計測) は全ての条件の組み合わせに対して PointGaze よりも BubbleGaze の方が速く, EMT の平均は PointGaze が 819.67ms, BubbleGaze が 612.15ms であった.
3. MT に対する PointGaze のモデルの適合度 R^2 の値は 0.89 であり, BubbleGaze の R^2 の値は 0.54, PointGaze と BubbleGaze のデータを合わせて線形回帰した場合の R^2 は 0.91 であった. また, EMT に対する PointGaze の R^2 の値は 0.85 であり, BubbleGaze の R^2 の値は 0.59, PointGaze と BubbleGaze のデータを合わせて線形回帰した場合の R^2 の値は 0.90 であった. 計算結果より PointGaze の適合率は高かったが, BubbleGaze の適合率は低かった.
4. ジッターの半径 (AR) の平均は PointGaze が 9.63 ピクセル, BubbleGaze が 15.22 ピクセルであった. また, 精度の平均は BubbleMouse が 98.67%, PointGaze が 98.15%, BubbleGaze が 98.27%であった.
5. NASA-TLX の平均スコアは BubbleMouse が 25.56, PointGaze が 64.15, BubbleGaze が 39.44 であった. SUS の平均スコアは BubbleMouse が 78.19, PointGaze が 55.56, BubbleGaze が 69.44 であった.

5.1 実験結果に関する考察

5.1.1 MT および EMT について

実験の結果より, 全ての実験の組み合わせにおいて動作時間 (MT) は BubbleMouse, BubbleGaze, PointGaze の順に速く, ターゲットに触れるまでの時間 (EMT) は BubbleGaze, PointGaze の順で速かった. したがって視線操作において BubbleGaze は PointGaze よりも常に高速に動作するといえる. また BubbleGaze の MT , EMT は EW に対する有意差が存在したが, W に対する有意差は存在しなかった. これは BubbleMouse の MT に対しても同様であることから, 視線カーソルにバブルカーソルを用いることによってターゲットの幅が拡大されていることが分かる. したがって Bubble

Gaze Cursor は、小さなターゲットに対するポインティングを改善するために有効な手法であるといえる。

一方で BubbleGaze は BubbleMouse よりも動作時間が大きかった。MTの平均で比較すると BubbleMouse は BubbleGaze よりもおおよそ 470ms 高速に動作した。この差は、今回設定した滞留時間 600ms よりも小さい。したがって滞留時間以外の入力手法（筋電[2]，ボタンクリック[15]など）の手法を用いることでマウス操作の動作時間に迫ることが出来ると考えられる。また、視線はマウスよりも非常に高速であるためターゲット間の距離が開けば開くほど、より視線操作が高速に動作する。ゆえに、本実験で用いた変数よりもターゲット間の距離が大きい場合、BubbleGaze と BubbleMouse との差は小さくなると考えられる。

5.1.2 ARについて

実験の結果より、ジッターの半径（AR）の平均は PointGaze が 9.63 ピクセル、BubbleGaze が 15.22 ピクセルであり、BubbleGaze の方が PointGaze よりもARが大きいという結果となった。これは図9から読み取れる通り、ターゲットの幅（PointGaze の場合は W ，BubbleGaze の場合は EW ）が大きくなるほど、ARも同様に大きくなるからである。ARがターゲットの幅に比例して大きくなるという特性は先行研究[20]でも指摘されており、ターゲットが大きくなるほどターゲットに視線を固定しようという注意が散漫になり、ARが大きくなると考察している。しかし BubbleGaze の場合、ターゲットの実質的な幅が可視化されていないが、ARとEWは有意な相関が存在し、EWが大きくなるほどARが大きくなっている。これはユーザがターゲット群の配置からターゲットの実質的な幅を推測しているためだと考えられる。今回の実験に用いたターゲット群の配置は非常に単純であり、バブルカーソルによって拡大されるターゲットの実質的な大きさは直感的に理解できる。一方で、ターゲットの実質的な大きさが不明瞭であるようなターゲット配置である場合、実質的なターゲットサイズをユーザが取り違え、視線がターゲットの外に出してしまうことが起こると考えられる。したがって、実環境に対する実験を通して Bubble Gaze Cursor の挙動を確認する必要がある。

5.2 Bubble Gaze Cursor における動作時間予測式

実験より、PointGaze と BubbleGaze のどちらも式2のMT、EMTに対する予測式に対する適合率が高くなかった。特に BubbleGaze の R^2 の値は低く、MTは 0.54、EMTは 0.59であった。予測式との適合率が低い理由として、式3の平滑化アルゴリズムを導入したことが考えられる。式3の平滑化アルゴリズムは高速な視線データを遅延させるアルゴリズムであり、この遅延によってジッターを減少させるが、同時にサッケードなどの視線の高速移動も遅延させてしまう。

ゆえに視線の速度が遅くなり、ターゲット間の距離Aが通常の視線操作よりも大きく影響を及ぼすようになったため、式2に適合しなくなると考えられる。実際、式2を提唱した Zhang らの論文[20]では、ターゲット内部でのみ機能し、ジッターを減少させるアルゴリズムである iSR[30]を除いて、視線データに対して平滑化アルゴリズムを利用することは一切していない。したがって、今後実験デザインを変更して追実験を行う必要があるといえる。

一方で、PointGaze と BubbleGaze の両方のデータを合わせて線形回帰した場合の R^2 は、MTは 0.91、EMTは 0.90であった。この値は PointGaze や BubbleGaze 単体の適合度よりも高い。したがって、平滑化アルゴリズム適用時においても、PointGaze から得られる動作時間予測式から BubbleGaze の動作時間をある程度予測することが可能であると考えられる。

5.3 提案手法の将来課題

バブルカーソルは視線操作においても正常に動作するか検証することを目的として、本稿では1次元ポインティングタスクを行った。しかし、2次元環境におけるポインティングや実環境における挙動を検証していないほか、Midas Touch Problem[5]の回避のために設定した、バブルカーソルの最大半径の最適な値の検証も必要であると考えられる。最大半径に関しては、高密度なターゲット群の中で Midas Touch Problem を回避することを考慮すると、ターゲットの距離や密度に応じて動的に変更することが必要だと考えられる。したがって、提案手法に対する検討はまだ必要であり、提案手法が実用に耐え得るか、引き続き実験していくべきだといえる。

5.4 視線カーソルの将来課題

バブルカーソルの抱える問題として、高密度のターゲット群に対しては効果を十全に発揮することが出来ないことが挙げられる[41]。なぜならバブルカーソルの大きさは最も近いターゲットとその次に近いターゲットの距離に依存するからである。したがって、GUIのように高密度にターゲットが集まっているようなデザインに対してバブルカーソルを用いても、ポインティングを容易かつ高速にすることは難しい。また、バブルカーソルの他の問題として急激なサイズ変更による視覚的注意散漫も挙げられる。常時カーソルの大きさを変える特性上、頻繁に大きさが変わる場合があり、それがユーザの集中を削ぐ可能性が存在する。

しかし、これらの難点の解消を目的にバブルカーソルの改善手法が提案されている。Mott ら[41]はバブルカーソル使用時に、ターゲットが小さく高密度である場合にズームすることで、バブルカーソルが機能しない状況で精度と動作時間を改善することに成功している。また、ズームの問題点である動作時間の増大を、カーソルの速度を参考にズ

ームを行うことで回避している。DynaSpot[42]はバブルカーソルの欠点であるカーソルの急激なサイズ変更を解決するために提案された、カーソルの速度に応じてカーソルの大きさを変えるエリアカーソルである。ターゲットとの距離を参考にカーソルの大きさを変更しないため、急激なサイズ変更がなくなり、視覚的注意散漫を減少させるほか、DynaSpot は止まっているときは基本的にポイントカーソルと同じ大きさになるため、仮に視線カーソルにこのカーソルを用いた場合、最大半径を設定せずに利用することが出来る。ゆえにこれらのエリアカーソルを視線カーソルに用いた場合、バブルカーソルを用いるよりも、より良い結果が得られる可能性があるといえる。

視線カーソルに対して、用途に応じて最適な形状やアルゴリズムが存在すると考えられる。しかし、現状視線カーソルに焦点を当てた研究は少ない。ゆえに今後、最適な視線カーソルを探求していく必要があるといえる。

6. おわりに

本稿では視線入力インタフェースでの小さなターゲットに対するポインティングを容易にすることを目的として、視線カーソルを、既存のポイントカーソルからバブルカーソルに変更する提案を行った。視線ポイントカーソルと視線バブルカーソルを比較するポインティングタスクを行ったところ、視線バブルカーソルは視線ポイントカーソルよりも常に高速で動作するうえ、ユーザビリティ評価指標やメンタルワークロード指標も視線ポイントカーソルより優れていることが確認された。

参考文献

- [1] Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99 (pp. 246–253). New York, New York, USA: ACM Press. <https://doi.org/10.1145/302979.303053>
- [2] Mateo, J. C., San Agustin, J., & Hansen, J. P. (2008). Gaze beats mouse. In Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08 (p. 3039). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1358628.1358804>
- [3] Pfeuffer, K., Alexander, J., Chong, M. K., & Gellersen, H. (2014). Gaze-touch. In Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14 (pp. 509–518). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2642918.2647397>
- [4] Pfeuffer, K., Alexander, J., Chong, M. K., Zhang, Y., & Gellersen, H. (2015). Gaze-Shifting. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15 (pp. 373–383). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2807442.2807460>
- [5] Jacob, R. J. K., & K., R. J. (1991). The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems*, 9(2), 152–169. <https://doi.org/10.1145/123078.128728>
- [6] Miniotos, D., Špakov, O., & MacKenzie, I. S. (2004). Eye gaze interaction with expanding targets. In *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04* (p. 1255). New York, New York, USA: ACM Press. <https://doi.org/10.1145/985921.986037>
- [7] Skovsgaard, H., Mateo, J. C., Flach, J. M., & Hansen, J. P. (2010). Small-target selection with gaze alone. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10* (p. 145). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1743666.1743702>
- [8] Ashmore, M., & Duchowski, A. (2005). Efficient eye pointing with a fisheye lens. In *Proceedings of Graphics Interface - GI '05*(pp. 203–210). Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society School of Computer Science, University of Waterloo. Retrieved from <https://dl.acm.org/citation.cfm?id=1089542>
- [9] Špakov, O., & Miniotos, D. (2005). Gaze-based selection of standard-size menu items. In *Proceedings of the 7th international conference on Multimodal interfaces - ICMI '05* (p. 124). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1088463.1088486>
- [10] Raudsandmoen, H., & Rødsjø, B. (2012). Empirically Based Design Guidelines for Gaze Interaction in Windows 7. 202. Retrieved from <https://brage.bibsys.no/xmlui/handle/11250/253070>
- [11] Grossman, T., & Balakrishnan, R. (2005). The bubble cursor. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05* (p. 281). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1054972.1055012>
- [12] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6), 381–391. <https://doi.org/10.1037/h0055392>
- [13] MacKenzie, I. S., & Scott, I. (1992). Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction*, 7(1), 91–139. https://doi.org/10.1207/s15327051hci0701_3
- [14] MacKenzie, I. S., Sellen, A., & Buxton, W. A. S. (1991). A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91* (pp. 161–166). New York, New York, USA: ACM Press. <https://doi.org/10.1145/108844.108868>
- [15] Ware, C., Mikaelian, H. H., Ware, C., Mikaelian, H. H., Ware, C., & Mikaelian, H. H. (1987). An evaluation of an eye tracker as a device for computer input. In *Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface - CHI '87* (Vol. 17, pp. 183–188). New York, New York, USA: ACM Press. <https://doi.org/10.1145/29933.275627>
- [16] Miniotos, D., & Darius. (2000). Application of Fitts' law to eye gaze interaction. In *CHI '00 extended abstracts on Human factors in computing systems - CHI '00* (p. 339). New York, New York, USA: ACM Press. <https://doi.org/10.1145/633292.633496>
- [17] Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00* (pp. 281–288). New York, New York, USA: ACM Press. <https://doi.org/10.1145/332040.332445>
- [18] Vertegaal, R., & Roel. (2008). A Fitts Law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th international conference on Multimodal interfaces - IMCI '08* (p. 241). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1452392.1452443>
- [19] Surakka, V., Illi, M., & Isokoski, P. (2004). Gazing and frowning as a new human-computer interaction technique. *ACM*

- Transactions on Applied Perception, 1(1), 40–56.
<https://doi.org/10.1145/1008722.1008726>
- [20] Zhang, X., Ren, X., & Zha, H. (2010). Modeling dwell-based eye pointing target acquisition. In Proceedings of the 28th international conference on Human factors in computing systems - CHI '10 (p. 2083). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/1753326.1753645>
- [21] Gori, J., Rioul, O., Guiard, Y., & Beaudouin-Lafon, M. (2018). The Perils of Confounding Factors. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18 (pp. 1–10). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/3173574.3173770>
- [22] H. Drewes. 2013. A Lecture on Fitts' law.(2013).
[http://www.cip.ifi.lmu.de/~drewes/science/fitts/A Lecture on Fitts Law.pdf](http://www.cip.ifi.lmu.de/~drewes/science/fitts/A%20Lecture%20on%20Fitts%20Law.pdf)
- [23] Špakov, O., & Oleg. (2012). Comparison of eye movement filters used in HCI. In Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12 (p. 281). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2168556.2168616>
- [24] Feit, A. M., Williams, S., Toledo, A., Paradiso, A., Kulkarni, H., Kane, S., & Morris, M. R. (2017). Toward Everyday Gaze Input. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17 (pp. 1118–1130). New York, New York, USA: ACM Press. <https://doi.org/10.1145/3025453.3025599>
- [25] Schenk, S., Dreiser, M., Rigoll, G., & Dorr, M. (2017). GazeEverywhere. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17 (pp. 3034–3044). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/3025453.3025455>
- [26] Hornof, A. J., & Cavender, A. (2005). EyeDraw. In Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05 (p. 161). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/1054972.1054995>
- [27] J Jimenez, D Gutierrez, and P Latorre. 2008. Gaze-based Interaction for Virtual Environments. Journal of Universal Computer Science 14 19 (2008), 3085--3098.
<http://giga.cps.unizar.es/>
- [28] Kumar, M., Klingner, J., Puranik, R., Winograd, T., & Paepcke, A. (2008). Improving the accuracy of gaze input for interaction. In Proceedings of the 2008 symposium on Eye tracking research & applications - ETRA '08 (p. 65). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1344471.1344488>
- [29] Casiez, G., Roussel, N., & Vogel, D. (2012). 1 € filter. In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12 (p. 2527). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/2207676.2208639>
- [30] Zhang, X., Ren, X., & Zha, H. (2008). Improving eye cursor's stability for eye pointing tasks. In Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08 (p. 525). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/1357054.1357139>
- [31] McGuffin, M., & Balakrishnan, R. (2002). Acquisition of expanding targets. In Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02 (p. 57). New York, New York, USA: ACM Press. <https://doi.org/10.1145/503376.503388>
- [32] Zhai, S., Conversy, S., Beaudouin-Lafon, M., & Guiard, Y. (2003). Human on-line response to target expansion. In Proceedings of the conference on Human factors in computing systems - CHI '03 (p. 177). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/642611.642644>
- [33] Majaranta, P., & Rähä, K.-J. (2002). Twenty years of eye typing. In Proceedings of the symposium on Eye tracking research & applications - ETRA '02 (p. 15). New York, New York, USA: ACM Press. <https://doi.org/10.1145/507072.507076>
- [34] Porta, M., Ravarelli, A., & Spagnoli, G. (2010). ceCursor , a contextual eye cursor for general pointing in windows environments. In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10 (p. 331). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/1743666.1743741>
- [35] Kabbash, P., & Buxton, W. A. S. (1995). The “prince” technique. In Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95 (pp. 273–279). New York, New York, USA: ACM Press. <https://doi.org/10.1145/223904.223939>
- [36] Worden, A., Walker, N., Bharat, K., & Hudson, S. (1997). Making computers easier for older adults to use. In Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97 (pp. 266–271). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/258549.258724>
- [37] Penkar, A. M., Lutteroth, C., & Weber, G. (2012). Designing for the eye. In Proceedings of the 24th Australian Computer-Human Interaction Conference on - OzCHI '12 (pp. 479–488). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/2414536.2414609>
- [38] Zhang, X., Xu, P., Zhang, Q., & Zha, H. (2011). Speed-accuracy trade-off in dwell-based eye pointing tasks at different cognitive levels. In Proceedings of the 1st international workshop on Pervasive eye tracking & mobile eye-based interaction - PETMEI '11 (p. 37). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/2029956.2029967>
- [39] S. G. Hart and L. E. Staveland. Development of NASATLX (Task Load Index): Results of Empirical and Theoretical Research. Human Mental Workload, North Holland Press, Amsterdam, page 46, 1988. P. A. Hancock and N. Meshikati (eds.).
- [40] John Brooke. SUS- A quick and dirty usability scale. Usability evaluation in industry, pp.189-194, 1996.
- [41] Mott, M. E., & Wobbrock, J. O. (2014). Beating the bubble. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14 (pp. 733–742). New York, New York, USA: ACM Press.
<https://doi.org/10.1145/2556288.2557410>
- [42] Chapuis, O., Labrune, J.-B., & Pietriga, E. (2009). DynaSpot. In Proceedings of the 27th international conference on Human factors in computing systems - CHI 09 (p. 1391). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1518701.1518911>