

Bubble Gaze Lens: バブルレンズ法の視線操作への拡張

崔明根¹ 坂本大介¹ 小野哲雄¹

概要: 本論文では、密集した小さなターゲットに対して高速にターゲット選択を行うことができるよう拡張した手法であるバブルレンズ法を、視線操作インタフェースに拡張した手法 Bubble Gaze Lens を提案する。本手法は、サッケードが弾道的な運動と修正的な運動によって構成されていることを利用して、ターゲット付近でレンズを拡大し、視線入力インタフェースにおける小さなターゲットの選択を容易にする。本稿では、既存手法である Bubble Gaze Cursor と提案手法 Bubble Gaze Lens に対してポインティングタスクを行った。その結果、提案手法は既存手法と比較して高速に動作し、エラー率を 54.0%削減した。さらに、ユーザビリティやメンタルワークロードにおいても提案手法の方が既存手法よりも有意に優れていた。

1. はじめに

Bolt [1]が視線入力を用いたマルチモーダルインタフェースの可能性を HCI (ヒューマンコンピュータインタラクション) の分野で提案して以来、視線入力インタフェースは HCI 分野において広く研究されるようになった。視線は遠距離に対するインタラクションが可能である[2]うえに、非常に高速である[3]。このため視線と他の入力を組み合わせることでマウス操作よりも高速な入力や効率的な操作が可能になった。一方で、視線単一の入力手法は何点か課題を抱えており、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問題がある。この問題は固視微動(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)やアイトラッカーの精度の限界[4]などが原因で、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。この問題を解消するため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[5, 6, 7, 8]、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[9]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。

我々はこの既存手法の課題を解決し、かつ小さなターゲットの選択を容易にする手法として、エリアカーソル[10, 11]の一種であるバブルカーソル[12]を視線入力インタフェースに用いる手法 Bubble Gaze Cursor (図 1.a) [13]を提案した。この手法は視線操作インタフェースにおいて最も用いられているポイントカーソルによるターゲット選択よりも有意に高速であり、さらにユーザビリティやメンタルワークロードの面においても有意に優れていた。しかし、バブルカーソルで密集した小さなターゲットの選択することは難しく、Bubble Gaze Cursor のみでは小さなターゲットのポインティングを完全に容易にすることは難しい。

そこで本稿では、バブルカーソルの改善手法であるバブルレンズ[14]を視線入力インタフェースに用いる手法

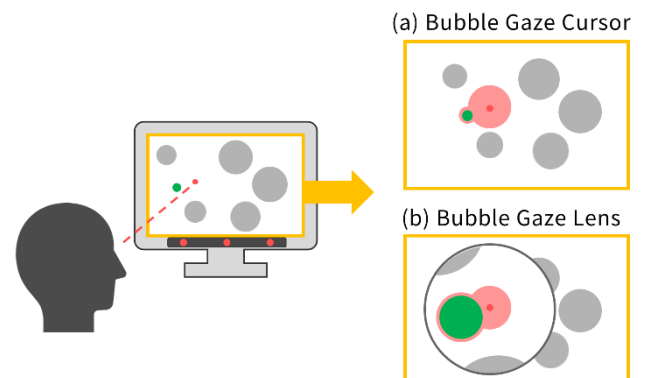


図 1 既存手法 Bubble Gaze Cursor[13]のイメージ (a) と提案手法 Bubble Gaze Lens のイメージ (b)

Bubble Gaze Lens を提案する。バブルレンズはバブルカーソルで取得が困難な、密集した小さなターゲットの選択を容易にするための手法である。バブルレンズは、十分な大きさのターゲットに対してはバブルカーソルとして動作するが、閾値以上に小さいターゲットを選択する場合はターゲット付近で拡大レンズが自動的に開く。この拡大レンズによってターゲットの幅を拡大し、容易なポインティングを達成している。この結果、視線入力インタフェースにバブルレンズを用いることで Bubble Gaze Cursor で困難だった小さなターゲットの選択を容易にすることが出来ると考えられる。本稿においては、既存手法 Bubble Gaze Cursor と提案手法 Bubble Gaze Lens に対して小さなターゲットに対するポインティングタスクを行い、その結果を報告する。

本研究の貢献を下記に要約する。

- 視線操作インタフェースにおける小さなターゲットの選択を容易にすることを目的としてバブルレンズを視線入力に対して用いる Bubble Gaze Lens を提案した。
- Bubble Gaze Cursor と提案手法を比較し、提案手法のパフォーマンスやユーザビリティを確認した。
- 実験の結果、提案手法は既存手法である Bubble Gaze Cursor のエラー率を 54.0%削減した。

¹ 北海道大学
Hokkaido University

2. 関連研究

2.1 視線操作インタフェースに必要なターゲットサイズ

人はターゲットをはっきりと見る時、網膜の中心に存在する中心窩でオブジェクトを捉える必要がある。中心窩は人の高精細な中心視野に寄与している部位であり、視角は約1~2度(眼球中心からディスプレイまでの角度)である。視線操作インタフェースにおけるオブジェクトの大きさは、この視角以上の大きさが必要である[4]。また、ターゲット注視時のディスプレイ上の視点位置は固視微動によって分散が大きく、全ての視点位置を内包する領域は非常に大きくなる。Feitら[15]はターゲットに対して注視しているときの視線データの95%がターゲット内に内包されるようなターゲットサイズを調査した。その結果、75%のユーザにとって十分なターゲットサイズは、生の視線データを扱う場合は $3.28 \times 3.78\text{cm}$ (2.91×3.33 度)、適切な平滑化アルゴリズムを使用する場合は $1.90 \times 2.35\text{cm}$ (1.67×2.07 度)であった。この大きさはマウスやタッチ操作に必要なターゲットの大きさと比較すると非常に大きい[9]。またSchuetzら[16]は、動作時間の減少を目的として、修正サッケード(メインサッケードの後に起こるサッケード)がほとんど発生しない3度以上のターゲットサイズを推奨している。ゆえに、既存のマウスやタッチ操作に適したインタフェースで小さなターゲットを視線操作で選択する場合、ターゲットサイズを拡大する手法が必要だといえる。

2.2 ターゲットを拡大する手法

ターゲットの幅を増大させる手法として考えられる手法にズームによるターゲットサイズの拡大が挙げられる。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[17, 18]が成り立つ[19, 20]ことが確認されている。このズーム機能を視線操作インタフェースに実装した研究はいくつか存在する。まず、ズーム開始のトリガーを視線以外の入力を用いる手法がある。EyePoint[21]はキーボード、Zooming interfaces[22]はスイッチによる入力をトリガーとしてユーザの視点位置をズームする。

一方で視線のみを用いたズーム手法は少ない。なぜなら、視線が単純な入力しか行うことが出来ないためである。Lankford [5]は二段階の滞留時間を用いたズーム手法を提案した。この手法は一定の滞留時間の後に凝視していた領域の拡大領域がポップアップされ、その拡大画面からターゲットを再度凝視し選択するという手法である。しかし、この手法は拡大領域によって画面内の情報損失が発生する。ゆえに、ほかのズーム手法として視線を向けた領域に対して連続的にズームを行う手法が提案されている[6]。この手法は関心領域以外が画面外に移動していくようなズームであるため、関心領域に関する情報損失が少ない。これらの

2つのズーム手法を比較すると、ポップアップ式のズーム手法はヒット率が高く、逆に連続的なズーム手法は高速であった[23]。しかし、どちらも余分な滞留時間が発生することからズーム手法を用いない手法よりも遅い。

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[7]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。Miniotasら[8]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。

2.3 Bubble Gaze Cursor

視線操作インタフェースにおける動作時間を維持した小さなターゲットの選択手法として、我々はBubble Gaze Cursor[13]を提案した。この手法はエリアカーソル手法の一種であるバブルカーソル法[12]を視線操作インタフェースに導入した手法である。バブルカーソルとはターゲットサイズをターゲットのボロノイ領域に常に拡大する手法である(図2)。視線操作インタフェースにおいて最も用いられているポイントカーソルとBubble Gaze Cursorを比較したところ、すべてのターゲットの有効幅に対して有意にBubble Gaze Cursorの方が高速であった。よってバブルカーソルは視線操作インタフェースにおいて正常に動作し、同時にBubble Gaze Cursorは有用な手法であることが、我々の研究によって明らかになった。

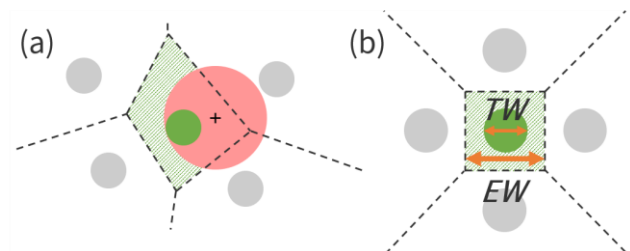


図2 (a) ボロノイ領域とバブルカーソル (b) ターゲットの幅 (TW) とターゲットの有効幅 (EW)

一方で Bubble Gaze Cursor はバブルカーソルがもともと持っていた欠点を引き継いでいる。その欠点とは、密集した小さなターゲットの選択が難しいという点である。バブルカーソルはターゲットサイズをボロノイ領域の大きさに拡大する手法であり、もともとボロノイ領域の大きさが小さいようなターゲットの選択には適していない。特に視線操作インタフェースにおいては非常に大きなターゲットサイズが必要となるため、この欠点は致命的である。

2.4 バブルレンズ

バブルカーソルが有する密集した小さなターゲットの選択が難しい、という問題を解消するために、Mott らはバブルレンズ法を提案した[14]。この手法はバブルカーソルによって十分に拡大されないようなターゲットの付近で拡大レンズを自動的に起動することでターゲットを拡大し、十分なターゲットサイズを確保する手法である。拡大レンズは、カーソルの動作が弾道運動と修正運動から成り立つという Meyer らの Initial impulse model [24]に則った手法であるキネマティックトリガーを用いており、閾値以下の大きさのターゲット付近にてカーソルの速度プロファイルに 2 度以上極小値が検出されたときに起動する。バブルレンズ法をバブルカーソル法と比較したところ、動作時間は 10.2%減少し、エラー率は 37.9%減少した。図 3 にバブルレンズ法での拡大レンズが起動する様子を示す。

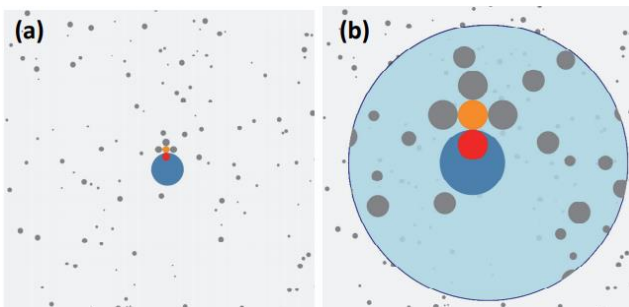


図 3 (a) バブルレンズの拡大レンズが起動する直前
(b) 拡大レンズが起動した画面状態[14]

3. 提案手法

本稿で提案する Bubble Gaze Lens は視線操作インタフェースにバブルレンズ法を用いた手法である。本手法ではゴールターゲット付近にユーザの視線が近づいたときに、ゴールターゲットを内包する拡大レンズが自動的に起動される。この拡大レンズによって Bubble Gaze Cursor で選択が困難であった密集した小さなターゲットの選択を容易にする。Bubble Gaze Lens のアルゴリズムはバブルレンズ法のアルゴリズムであるキネマティックトリガーに則っている。

3.1 Bubble Gaze Lens の原理

バブルレンズの拡大レンズの起動は、マウスカーソルの動作が単一の弾道運動と複数の修正運動によって成り立つ

という Meyer らのモデル[24]に則っている。サッケード(視線の高速移動)も、マウスカーソルの挙動と同様に単一の弾道運動と修正運動を有している場合がある。サッケードは基本的に単一の弾道運動のみで構成されているが、ターゲット指向のサッケードはターゲットの離心率の 5~10%ほど離れた位置に着地することがあり[25]、それを追加の短いサッケード(二次サッケード)[26,27]が修正することがある。この二次サッケードを修正運動と考えることで、バブルレンズを視線操作インタフェースに導入することが可能であると私たちは考えた。

3.2 Bubble Gaze Lens のアルゴリズム

Bubble Gaze Lens のアルゴリズムはバブルレンズのキネマティックトリガーを参考にしている。キネマティックトリガーはマウスカーソルの速度プロファイルに極大値が 2ヶ所検出されたときに拡大レンズを起動するアルゴリズムである。具体的なキネマティックトリガーの仕組みを以下に記述する。まず、マウスカーソルの速度データを 100Hz でリサンプリングした後にガウシアンカーネルフィルタで平滑化する。そして極大値が 2ヶ所検出され、かつマウスカーソルの直近ターゲットが既定の有効サイズよりも小さい場合に拡大レンズを起動する。

提案手法ではキネマティックトリガーを視線操作作用に拡張した。まず、バブルレンズにおいては速度データをリサンプリングした後に平滑化を行ったが、Bubble Gaze Lens においてはアイトラッカーのサンプリングレートが低い(90Hz)ことから平滑化を行わなかった。さらに、視線はマウスと異なり常に動き続けているため、判定をリセットするためにウィンドウサイズを 50 データに定めた。Bubble Gaze Lens のアルゴリズムを以下に示す(図 4)。

- ① ウィンドウの初めに視線が 150ms 固定されている
- ② 100°/s以上の極大値(メインサッケード)と 30°/s以上の極大値(2次サッケード)を検出
- ③ 2ヶ所の極大値が 50~250ms 離れている
- ④ ①②③が成り立ち、直近 40ms の視線が固定されているときに拡大レンズを起動する

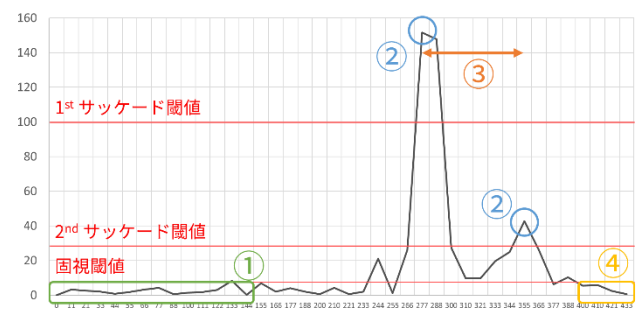


図 4 視線の速度プロファイルに対する Bubble Gaze Lens のアルゴリズムの概略

3.3 パラメタの詳細

Bubble Gaze Lens におけるバブルの最大幅は、バブルレンズに則って 100 ピクセルとした。この最大幅は拡大レンズ内でも維持される。また、拡大レンズによる拡大率は 4 で、拡大レンズの幅は 560 ピクセルである。したがって、70 ピクセルを直径とする円の領域が拡大される。目標ターゲットは、拡大レンズが画面外に表示されないように、画面横端から 200 ピクセル以内と縦端 150 ピクセル以内に生成されないようにした。また、仮に拡大レンズ内に目標ターゲットが含まれていなかった場合、1 秒後に次の試行へ進む。滞留時間は 600ms であり、この滞留時間は拡大レンズが起動したときにリセットされる。

4. 実験

提案手法が有用であるか検討するために、Bubble Gaze Cursor と Bubble Gaze Lens との比較実験を行い、提案手法のパフォーマンスを測定する。

4.1 実験参加者と使用機器

24 名の実験参加者を募ったが、2 名はキャリブレーションのエラーで、2 名は不規則な視線運動が原因で除外した。ゆえに実験参加者として 20 名（女性 4 名、平均年齢 23.0 歳、標準偏差 2.1 歳）が実験に参加した。1 名はアイトラッキングを用いた経験があった。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。実験時間は約 1 時間であった。

本実験は 3.40Ghz の Intel (R) Core (TM) i7-6700U 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 23 インチ (1920 × 1080 ピクセル; 1 ピクセルあたり約 0.27 mm) のディスプレイを使用した。アイトラッカーは Tobii Eye Tracker 4C (90Hz) を使用し、実験に用いたソフトウェアは C# のウィンドウズフォームアプリケーションで作成した。

4.2 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 4 つ存在し、それぞれ以下のように設定されている。

- カーソルの種類 CT (Bubble gaze cursor, Bubble gaze lens)
- 視線からターゲット中心までの距離 D (図 5) (350, 550, 800 ピクセル) ;
- ターゲットの幅 TW (20, 32, 52 pixels)
- ターゲット間のスペース S (図 6)

(スペースなし: 0 ピクセル, ターゲット半径の半分: $TW/4$ ピクセル, ターゲット半径: $TW/2$ ピクセル)

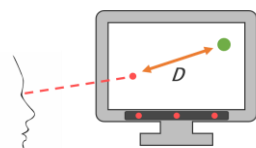


図 5 視線からターゲット中心までの距離 D

実験参加者は CT 条件ごとに 15 のセッションを行った。各セッションはランダムな順序で表示される D 、 TW 、 S の 27 の組み合わせによって構成されている。タスクはターゲット選択であり、ターゲットを選択すると D だけ離れたランダムな位置にターゲットが新しく生成される。実験参加者はそれぞれ 2 CT 条件 × 15 セッション × 27 (D 、 TW 、 S の組み合わせ) = 810 回のターゲット選択を行う。ゆえに、実験参加者 20 名から 16200 のデータを収集した。



図 6 ターゲット間のスペース S

CT に対する従属変数は動作時間とエラー率を求めた。エラーは、間違ったターゲット選択か 5 秒以上ターゲットが選択されなかった場合に計上された。また Bubble Gaze Lens に対してのみ、キネマティックトリガーが失敗した割合を従属変数として扱った。そして、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 System Usability Scale (SUS) [28] と NASA Task Load Index (NASA-TLX) [29] によるアンケートを実施した。

4.3 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、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。

タスクは複数配置された灰色の偽ターゲットの中に 1 つだけ配置された緑の目標ターゲットを選択するというものである。目標ターゲットの周りには、バブルカーソルによって拡大される有効幅を制御するために、偽ターゲットが四方に配置されている。偽ターゲットは 16~40 ピクセルの大きさでランダムに位置に計 350 個生成された。目標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 [30] 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。

4.4 結果

それぞれの従属変数に対してノンパラメトリックな分散分析手法である整列ランク変換 (ART) [31, 32, 33]を行い、混合モデル REML を用いてデータを評価した。多重比較にはテューキーの多重比較検定を用いた。SUS と NASA-TLX によって得られた値はウィルコクソンの符号順位検定を用い、多重比較には Holm 法[34]による補正を用いた。また、データの可視化のためにターゲットの有効幅 (EW) [12]を用いた。ターゲットの有効幅はバブルカーソルによって実質的に拡大されるターゲットの幅であり、TW と S を用いることで計算することが出来る。ゆえに 9 つの EW (20, 25, 30, 32, 40, 48, 52, 65, 78 ピクセル) が得られる。

4.4.1 動作時間

カーソルの種類 CT, 視線からターゲットまでの距離 D, ターゲットの幅 TW, ターゲット間のスペース S を独立変数とし、動作時間を従属変数として分析を行った。動作時間に対して CT ($F_{1,10956}=13.18, p<.01$), D ($F_{2,10954}=44.75, p<.01$), TW ($F_{2,10954}=609.57, p<.01$), S ($F_{2,10953}=59.09, p<.01$) に有意な差が確認された。さらに 1 次の交互作用として CT×TW ($F_{2,10954}=36.72, p<.01$), TW×S ($F_{4,10953}=36.72, p<.01$), 2 次の交互作用として CT×TW×S ($F_{4,10953}=2.95, p<.05$) が確認された。図 7 にターゲットの有効幅に対する CT ごとの動作時間を示す。

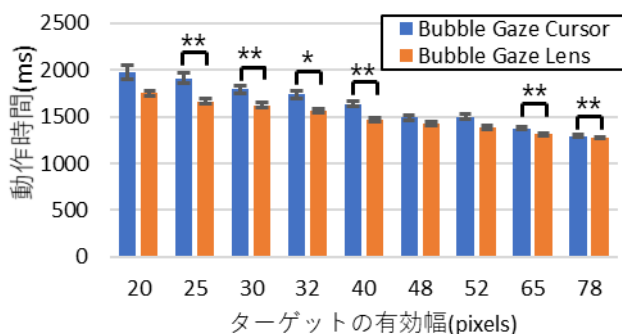


図 7 ターゲットの有効幅に対する動作時間 (有意差: * $p<.05$, ** $p<.01$)

4.4.2 エラー率

カーソルの種類 CT, 視線からターゲットまでの距離 D, ターゲットの幅 TW, ターゲット間のスペース S を独立変数とし、エラー率を従属変数として分析を行った。エラー率に対して CT ($F_{1,16127}=1980.39, p<.01$), TW ($F_{2,16127}=2039.44, p<.01$), S ($F_{2,16127}=304.56, p<.01$) に有意な差が確認された。さらに 1 次の交互作用として CT×D ($F_{2,16127}=45.92, p<.01$), CT×TW ($F_{2,16127}=540.60, p<.01$), D×TW ($F_{4,16127}=6.68, p<.01$), CT×S ($F_{2,16127}=181.49, p<.01$), D×S ($F_{4,16127}=3.75, p<.01$), TW×S ($F_{4,16127}=84.69, p<.01$), 2 次の交互作用として CT×D×TW ($F_{4,16127}=16.10, p<.01$), CT×D×S ($F_{4,16127}=3.45, p<.01$), CT×TW×S ($F_{4,16127}=36.90, p<.01$), D×TW×S ($F_{8,16127}=2.60, p<.01$)

が確認された。図 8 と表 1 にターゲットの有効幅に対する CT ごとのエラー率を示す。

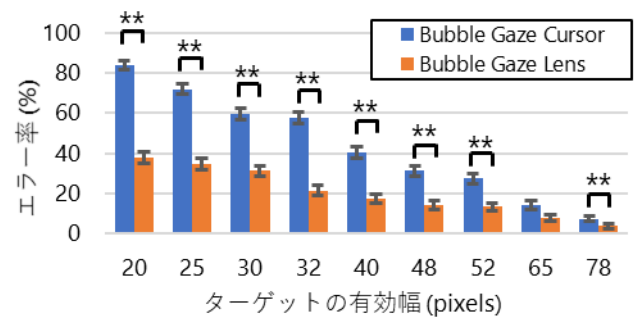


図 8 ターゲットの有効幅に対するエラー率 (有意差: * $p<.05$, ** $p<.01$)

		ターゲットの有効幅 (EW)					
		20	25	30	32	40	
		tasks	error	tasks	error	tasks	
Cursor		142	84%	252	72%	363	60%
Lens		560	38%	589	35%	619	31%

		ターゲットの有効幅 (EW)			
		48	52	65	78
		tasks	error	tasks	error
Cursor		619	31%	653	27%
Lens		774	14%	781	13%

表 1 EW に対する CT ごとの成功タスク数とエラー率 (総タスク数は 900)

Bubble Gaze Lens におけるエラーがいつ発生したか調べたところ、全体のエラーの 72.79%が、拡大レンズが起動しなかった場合に発生していた。ゆえに Bubble Gaze Lens のエラーの多くはキネマティックトリガーが正しく動作しなかったことに起因しているといえる。また、レンズ内に目標ターゲットを含まない位置で拡大レンズが開いた割合は全試行の 0.95%であり、27.21%のエラー試行のうち 3.49%であった。図 9 と表 2 にターゲットの有効幅に対する、レンズが開いたときのエラーとレンズが開かなかったときのエラーの割合を示す。

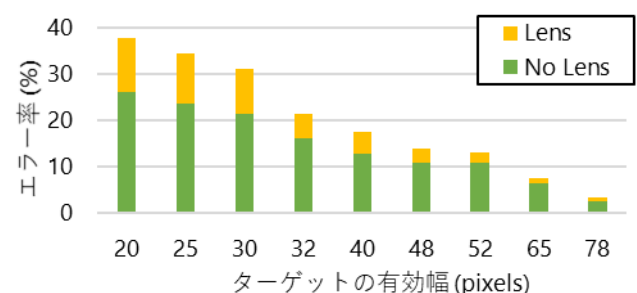


図 9 レンズが開いた時のエラー (Lens) とレンズが開かなかった時のエラー (No Lens) の割合

	ターゲットの有効幅 (EW)									
	20	25	30	32	40	48	52	65	78	
Error rate (%)	37.78	34.56	31.22	21.44	17.44	14.00	13.22	7.67	3.56	
Activation error rate (%)	11.75	10.94	9.66	5.21	4.51	2.98	2.20	1.08	0.86	
Non-activation error rate (%)	26.03	23.62	21.56	16.23	12.93	11.02	11.02	6.59	2.70	

表2 ターゲットの有効幅に対するエラー率とレンズが開いた時のエラー率とレンズが開かなかった時のエラー率

4.4.3 キネマティックトリガーが失敗した割合

視線からターゲットまでの距離 D, ターゲットの幅 TW, ターゲット間のスペース S を独立変数とし, キネマティックトリガーが失敗した割合を従属変数として分析を行った. キネマティックトリガーが失敗した割合に対して D ($F_{2,7977.1}=44.31, p<.01$), TW ($F_{2,7977.0}=9.77, p<.01$), S ($F_{2,7977.0}=28.91, p<.01$) に有意な差が確認された. さらに 1 次の交互作用として D×TW ($F_{4,7977.0}=3.86, p<.01$), D×S ($F_{4,7977.0}=8.95, p<.01$), TW×S ($F_{4,7977.0}=5.79, p<.01$), 2 次の交互作用として D×TW×S ($F_{8,7977.0}=4.64, p<.01$) が確認された. 表 3 に D に対する失敗した割合, 失敗した数, 全試行数を示す. キネマティックトリガーが失敗した平均割合は 28.37%であった.

D	失敗した割合	失敗した数	全試行数
350	40.04%	1075	2685
550	30.40%	816	2684
800	14.51%	385	2654

表3 D に対する失敗した割合, 失敗した数, 全試行数

4.4.4 System Usability Scale (SUS)

得られたスコアに対してウィルコクソンの符号順位検定を用いて分析した. Bubble Gaze Cursor と Bubble Gaze Lens の平均 SUS スコアはそれぞれ 60.88 と 70.13 であり, 有意な差が確認された ($Z=3.11, p<.01$). SUS の質問項目ごとの結果を図 10 に示す.

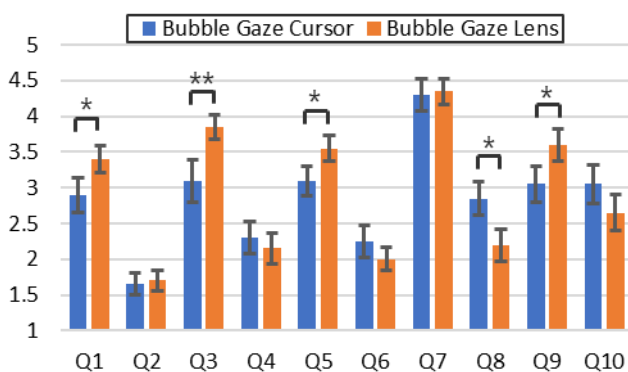


図10 SUS スコア (有意差: * $p<.05$, ** $p<.01$)

4.4.5 NASA Task Load Index (NASA-TLX)

NASA-TLXによって得られた6つの項目に対するスコアと Overall スコアに対して, ウィルコクソンの符号順位検定を用いて分析した. Bubble Gaze Cursor と Bubble Gaze Lens の平均 Overall スコアはそれぞれ 46.10 と 34.57 であった. また CT に対して Mental ($Z=3.04, p<.01$), Physical ($Z=2.47, p<.05$), Overall workload ($Z=2.92, p<.01$) に有意な差が確認された. NASA-TLX の項目ごとの結果を図 18 に示す.

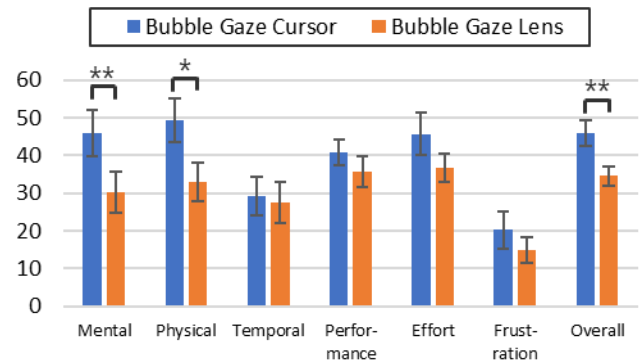


図11 NASA-TLX スコア (有意差: * $p<.05$, ** $p<.01$)

4.5 実験結果のまとめ

Bubble Gaze Lens は Bubble Gaze Cursor よりも全ての EW に対して高速であった. しかし, EW=20 のときに有意に高速ではなかった. これは Bubble Gaze Cursor のエラー率が高く (84%), 十分なデータ数が確保できなかった (表 1: 142/900) ためだと考えられる. また, Bubble Gaze Lens はエラー率を 54.0%削減した. ただし, 平均エラー率は 20.10%, 最も高いエラー率 (EW=20) は 37.78%であった. このエラーの原因の大半はキネマティックトリガーが正常に動作せず, 拡大レンズが起動しなかったためである. キネマティックトリガーの平均エラー率は 28.37%, 最も高いエラー率 (D=350) は 40.04%であった. 最後に, ユーザビリティとメンタルワークロードの両方において, Bubble Gaze Cursor よりも Bubble Gaze Lens の方が優れていた.

5. 議論

5.1 得られた従属変数に対する考察

5.1.1 動作時間

実験結果より, 全ての EW に対して Bubble Gaze Lens の方が Bubble Gaze Cursor よりも高速に動作した. 一方で, EW が大きくなるほど動作時間の差が小さくなった. Bubble Gaze Lens は拡大レンズの起動による画面レイアウトの変更が原因で, 最善な軌道における通常のポインティングよりも動作時間が大きくなる. なぜなら, 画面レイアウトが変わった瞬間にサックード計画が必要になり, 200ms 程度視線の動きが止まる (潜時) ためである. ゆえに, Bubble Gaze Lens においてもバブルレンズと同様に拡

大レンズを用いない方が早くなる閾値が存在すると考えられる。実験結果より、この閾値はおよそ 80 ピクセルだと考えられるが、ユーザごとに視線ポインティングの精度に差があるため、個別に設定を行う必要があると考えられる。

5.1.2 エラー率

実験結果より、Bubble Gaze Lens は大幅にエラー率を改善した。しかし、実用に耐えうるほどの高精度を達成することは出来なかった。エラー率の主な原因は、キネマティックトリガーが上手く動作しなかったことが挙げられる。一方で、レンズが起動した場合においても、エラー率は EW が小さい場合において高くなっている。拡大倍率が 4 であるため、EW=20 のときターゲットは拡大レンズによって 80 ピクセルまで拡大される。しかし EW=20 におけるレンズが拡大されたときのエラー率は EW=78 のときのエラー率よりも高い値が記録されている(表 2)。これは潜時による拡大レンズ起動時の視線の固定が原因であると考えられる。拡大レンズが起動すると、視線座標の直近のターゲットがバブルカーソルの最大幅より近い場合、直近のターゲットが選ばれている状態で潜時が発生する。ここで選択されているターゲットが目標ターゲット以外のターゲットだとすると、200ms 程度、間違ったターゲットに視線を固定した状態から視線の移動が始まる。そして、4 倍に拡大されたターゲット領域から視線を外そうとする途中で、間違ったターゲットが選択されることが起こり得る。この現象がレンズ起動時のエラー率の高さに繋がっていると考えられる。

5.1.3 キネマティックトリガーが失敗した割合

本実装におけるキネマティックトリガーは、二次サッケードの検出を行うアルゴリズムである。二次サッケードは視線とターゲット中心間の距離が大きくなればなるほど発生する割合が大きくなるのが以前の研究で明らかになっており[35]、本実験の結果とも一致する(表 3)。しかし、全てのサッケードに二次サッケードが含まれることはない。これは二次サッケードをトリガーとすることの限界が表れている。特に近距離に対するトリガーとしては、キネマティックトリガーは不十分だといえる。

また、キネマティックトリガーのアルゴリズムの改善が必要だと考えられる。現在のアルゴリズムの二次サッケード検出は速度閾値を用いる手法[36]であり、検出精度は低い。二次サッケードの検出手法は様々提案されているが、基本的にオフラインにおけるデータ分析を前提とした手法である。ゆえに動的な二次サッケードの検出手法を検討していく必要がある。

5.2 制限

本研究で用いたアイトラッカーは非常に安価(約 2 万円)であり、サンプリングレートは 90Hz である。このサン

プリングレートはマイクロサッケード(短いサッケード)を取得するには非常に小さい値である[37, 38]ため、二次サッケードの検出に適していない。二次サッケードのピーク速度を正確に検出するためには 240Hz 以上のサンプリングレートを有したアイトラッカーを用いる必要がある[38]が、そういったアイトラッカーは非常に高価である。

また、サンプリングレートを補う手法として視線データにアップサンプリングを行う手法が提案されている[38]。この手法によって低いサンプリングレートであっても二次サッケードの検出精度を向上させることが出来るが、この手法はオフラインを前提とした手法であり、リアルタイムでの適用は難しい。ゆえに本研究では用いなかった。

5.3 今後の課題

今後の課題として、まずキネマティックトリガーのアルゴリズムの改善が挙げられる。確実に二次サッケードが検出できるようになった場合における提案手法の有用性を調べることによって、二次サッケードを利用したインタラクティブ手法の可能性がより明らかになると考えられる。次に、二次サッケード以外のトリガーの検討が挙げられる。二次サッケードをトリガーとすることに限界がある以上、別の入力等を組み合わせることによって、よりレンズ起動の頻度や精度を向上させることを目指す必要がある。ターゲット選択時における瞳孔情報[39]などが 1 つの候補である。最後に、私たちはエリアカーソル法の一つであるバブルレンズを視線操作インタフェースに適用したが、ほかにも検討すべきエリアカーソル法は存在する[40]。ゆえに、それらが視線入力に適しているかどうか検討が必要だといえる。

6. おわりに

本稿では視線操作インタフェースにおける、密集した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、バブルレンズを視線操作インタフェースに用いる手法 Bubble Gaze Lens を提案した。既存手法である Bubble Gaze Cursor と Bubble Gaze Lens を比較するポインティングタスクを行ったところ、Bubble Gaze Lens は気損手法 Bubble Gaze Cursor よりも動作時間、エラー率、ユーザビリティ、メンタルワークロードの点において優れていた。一方でエラー率は未だに高く、現状の提案手法は実用に適してはいない。したがって、高いサンプリングレートのアイトラッカーを用いることや、キネマティックトリガーのアルゴリズムの改善などによって、改善を試みていく必要がある。

謝辞 本研究はノーステック財団「研究開発助成事業」の助成を受けたものです。

参考文献

- [1] Richard A. Bolt. Eyes at the interface. In *Proc. CHI '82*, 1982, p. 360-362.
- [2] Vildan Tanriverdi and Robert J. K. Jacob. Interacting with eye movements in virtual environments. In *Proc. CHI '00*, 2000, p. 265-272.
- [3] Linda E. Sibert and Robert J. K. Jacob. Evaluation of eye gaze interaction. In *Proc. CHI '00*, 2000, p. 281-288.
- [4] I. Scott MacKenzie. An eye on input: research challenges in using the eye for computer input control. In *Proc. ETRA '10*, 2010, p. 11-12.
- [5] Chris Lankford. Effective eye-gaze input into Windows. In *Proc. ETRA '00*, 2000, p. 23-27.
- [6] Dan Witzner Hansen, Henrik H. T. Skovsgaard, John Paulin Hansen, and Emilie Møllénbach. Noise tolerant selection by gaze-controlled pan and zoom in 3D. In *Proc. ETRA '08*, 2008, p. 205-212.
- [7] Oleg Špakov. Comparison of eye movement filters used in HCI. In *Proc. ETRA '12*, 2012, Stephen N. Spencer (Ed.), p. 281-284.
- [8] Darius Miniotas, Oleg Špakov, and I. Scott MacKenzie. Eye gaze interaction with expanding targets. In *Proc. CHI EA '04*, 2004, p. 1255-1258.
- [9] Håkon Raudsandmoen and Børge Rødsgjø. 2012. Empirically Based Design Guidelines for Gaze Interaction in Windows 7. Master thesis. Norwegian University of Science and Technology, 2012.
- [10] Paul Kabbash and William A. S. Buxton. The “prince” technique: Fitts' law and selection using area cursors. In *Proc. CHI '95*, 1995, Irvin R. Katz, Robert Mack, Linn Marks, Mary Beth Rosson, and Jakob Nielsen (Eds.), p. 273-279.
- [11] Aileen Worden, Nef Walker, Krishna Bharat, and Scott Hudson. Making computers easier for older adults to use: area cursors and sticky icons. In *Proc. CHI '97*, 1997, p. 266-271.
- [12] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI '05*, 2005, p. 281-290.
- [13] 崔明根, 坂本 大介, 小野 哲雄. Bubble Gaze Cursor. インタラクション 2019 論文集, 2019, p. 20-29.
- [14] Martez E. Mott and Jacob O. Wobbrock. Beating the bubble: using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proc. CHI '14*, 2014, p. 733-742.
- [15] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design. In *Proc. CHI '17*, 2017, p. 1118-1130.
- [16] Immo Schuetz, T. Scott Murdison, Kevin J. MacKenzie, and Marina Zannoli. An Explanation of Fitts' Law-like Performance in Gaze-Based Selection Tasks Using a Psychophysics Approach. In *Proc. CHI '19*, 2019, p. 535-547.
- [17] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 1954, vol. 47, no. 6, p. 381-391.
- [18] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human Computer Interaction*, 1992, vol. 7, no. 1, p. 91-139.
- [19] Shumin Zhai, Stéphane Conversy, Michel Beaudouin-Lafon, and Yves Guiard. Human on-line response to target expansion. In *Proc. CHI '03*, 2003, p. 177-184.
- [20] Michael McGuffin and Ravin Balakrishnan. Acquisition of expanding targets. In *Proc. CHI '02*, 2002, p. 57-64.
- [21] Manu Kumar, Andreas Paepcke, and Terry Winograd, Terry Winograd. EyePoint: practical pointing and selection using gaze and keyboard. In *Proc. CHI '07*, 2007, p. 421-430.
- [22] Richard Bates and Howell Istance. Zooming interfaces!: enhancing the performance of eye controlled pointing devices. In *Proc. Assets '02*, 2002, p. 119-126.
- [23] Henrik Tomra Skovsgaard Hegner Jensen, John Paulin Hansen, and Julio C. Mateo. How Can Tiny Buttons Be Hit Using Gaze Only?. In *Proc. COGAIN 2008*, 2008, p. 38-42.
- [24] David E Meyer, J. E. Keith-Smith, Sylvan Kornblum, Richard A. Abrams, and Charles E Wright. Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. *Attention and Performance 13: Motor representation and control*. Lawrence Erlbaum Associates, Inc. Hillsdale, 1990, p. 73-226
- [25] David B. Henson. Corrective saccades: Effects of altering visual feedback. *Vision Research*, 1978, vol. 18, no. 1, p. 63-67.
- [26] Wolfgang Becker and Albert F. Fuchs. Further properties of the human saccadic system: eye movements and correction saccades with and without visual fixation points. *Vision research*, 1969, vol. 9, no. 10, p. 1247-1258.
- [27] Ronald B. Weber and Robert B. Daroff. Corrective movements following refixation saccades: type and control system analysis. *Vision research*, 1972, vol. 12, no. 3, p. 467-475.
- [28] John Brooke. SUS- A quick and dirty usability scale. *Usability evaluation in industry*, 1996, p. 189-194.
- [29] Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index) : Results of Empirical and Theoretical Research. Human Mental Workload. *Advances in Psychology*, 1988, vol. 52, p. 139-183.
- [30] Xinyong Zhang, Wenxin Feng, Hongbin Zha. Effects of Different Visual Feedback Forms on Eye Cursor's Stabilities. In *Proc. IDGD '11*, 2011, Lecture Notes in Computer Science, vol 6775, p. 273-282.
- [31] James J. Higgins and Suleiman Tashtoush. An aligned rank transform test for interaction. *Nonlinear World*, 1994, vol. 1, no. 2, p. 201-211.
- [32] Salter, K. C., and Fawcett, R. F. The art test of interaction: a robust and powerful rank test of interaction in factorial models. *Communications in Statistics - Simulation and Computation*, 1993, vol. 22, no. 1, p. 137-153.
- [33] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proc. CHI '11*, 2011, p. 143-146.
- [34] Sture Holm. A simple sequentially rejective Bonferroni test procedure. *Scandinavian Journal of Statistics*, 1979, vol. 6, p. 65-70.
- [35] Ronald B. Weber and Robert B. Daroff. The metrics of horizontal saccadic eye movements in normal humans. *Vision Research*, 1971, vol. 11, no. 9, p. 921-928.
- [36] Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proc. ETRA '00*, 2000, p. 71-78.
- [37] Roel Wierds, Maurice J. A. Janssen, and Herman Kingma. Measuring Saccade Peak Velocity Using a Low-Frequency Sampling Rate of 50 Hz. *IEEE Transactions on Biomedical Engineering*, 2008, vol. 55, no. 12, p. 2840-2842.
- [38] David J. Mack, Sandro Belfanti, Urs Schwarz. The effect of sampling rate and lowpass filters on saccades – A modeling approach. *Behavior Research Methods*, 2017, vol. 49, no. 6, p. 2146-2162.
- [39] Christoph Strauch, Jan Ehlers, and Anke Huckauf. Pupil-Assisted Target Selection (PATS) . In *Human-Computer Interaction – INTERACT 2017*, 2017, Lecture Notes in Computer Science, vol 10515, p. 297-312.
- [40] Olivier Chapuis, Jean-Baptiste Labrune, and Emmanuel Pietriga. DynaSpot: speed-dependent area cursor. In *Proc. CHI '09*, 2009, p. 1391-1400.