

時間および範囲をもとに認識する凝視に基づく操作手法

磯本 俊弥* 山中 祥太† 志築 文太郎*

概要. 新たな凝視の認識手法による、凝視に基づく操作手法を示す。従来手法における凝視の認識が「対象の中に視線座標が一定時間（以降、凝視時間）以上留まっていること」を条件に行われていたことに対し、本手法では「視線座標がある範囲（以降、凝視範囲）に凝視時間以上留まっていること」を条件に行われる。凝視の認識に凝視時間および凝視範囲を用いることにより得られる利点は、主に、対象内に視線座標が入っているが視線が移動している場合に生じるミダスタッチ問題の軽減である。これにより、例えば、ユーザがウェブページのリンクを読んでいる際や、動画プレイヤーのサムネイルを見ている際に生じるミダスタッチの軽減が可能である。著者を参加者とした実験の結果から、本手法を用いた場合に従来手法と同じ様に対象選択ができ、かつミダスタッチの問題を軽減できる可能性を示した。

1 はじめに

本稿において、視線に基づく操作手法における新たな凝視の認識手法を提案する。従来手法では、視線座標（視線認識機器によって認識されたユーザが見ている位置）がある対象内に一定時間（凝視時間）以上留まっているかを条件に凝視が認識される [7] (図 1a-f)。そして、凝視が認識された際に視線座標が入っている対象へ選択操作（マウス操作での左クリックや、タッチ操作でのタップと同等の操作）が実行される。つまり、従来手法は、対象を選択したいというユーザの意志を、視線座標が対象内に凝視時間以上留まっていることに紐付けている。対して、本手法は、対象の有無に依らず、視線座標のばらつきがある範囲（以降、凝視範囲）に凝視時間以上留まっているかを条件に凝視を認識する（図 1g-l）。

凝視範囲の決定は、ユーザが意図的にある点を見続ける際に生じる視線座標のばらつきにもとづいて行う。この視線座標のばらつきには、人間がある 1 点を見つめている（注視をしている）際に生じるトレモルやドリフト、フリッカリング [1] や、視線認識機器のノイズによって生じるばらつきが含まれる。そして、凝視が認識された際に視線座標群の重心に対して選択操作が実行される。つまり、ユーザの対象を選択したいという意志を、視線座標がある点に凝視時間以上留まるとことに紐付けている。なお、本手法においても、従来手法と同様に凝視を認識するための条件として凝視時間を用いる。

以降、本稿では、まず視線座標のばらつきが凝視範囲に凝視時間以上留まっていることを条件に凝視を認識することの利点を述べる。その後、凝視範囲の決定方法を述べ、著者を参加者とした実験の結果から、本手法が実際に動作することを示す。

2 本手法による利点

凝視を認識する条件を「視線座標がある対象の中に凝視時間以上留まる」から「視線座標が凝視範囲内に凝視時間以上留まる」とすることによる主な利点は、ミダスタッチ問題の軽減である。ミダスタッチ問題とは、凝視に基づく操作時にユーザの意図しない対象選択が生じるという問題である [7]。これは、例えば対象内の文章を読んでいる際や画像を見ている際のように、ユーザに選択する意思はないが視線座標が対象内に入っている際に生じる。

ミダスタッチ問題の解決は、視線に基づく操作の普及に必要なことであり、これまでに多くの研究がされてきた。それらは、凝視時間の調整（例 [11, 16]）、専用の UI の使用（例 [8, 10, 17]）、輻輳効果の使用（例 [9]）、追跡眼球運動の使用（例 [15]）が多かった。ただし、これらの研究では、視線が移動しているかという情報は考慮されてこなかった。また、ミダスタッチの問題の発生に関して、対象内に文字を表示しないことが有効であることも示されている [12]。そのため、これまでの研究での実験における対象は、無地もしくは単語、簡単な画像が表示されたものであり、対象内での視線移動が誘発されない・必要ないような実験設計がされていることが多い。

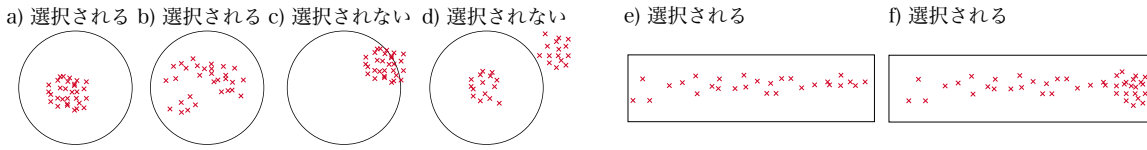
しかし、マウスやタッチでの操作に使われるような UI では、ユーザにとって未知な情報が表示されている対象が多く、また、対象には横長なものもあるため、凝視時間の長さによらずミダスタッチ問題が生じる可能性がある。例えば、ウェブページにおけるリンクや動画プレイヤーにおけるサムネイル（画像もしくは動画）は選択可能対象となっていることが多い。ユーザはこれらの対象群の中から、探索および考え事をしながら 1 つの対象を選択する。この時、何らかの情報を見るために視線が移動しているとしても、視線座標が対象内に留まるため、従来手法ではミダスタッチ問題が生じる。

Copyright is held by the author(s).

* 筑波大学

† ヤフー株式会社

従来手法（凝視時間に基づく凝視の認識）



本手法（凝視時間および凝視範囲に基づく凝視の認識）

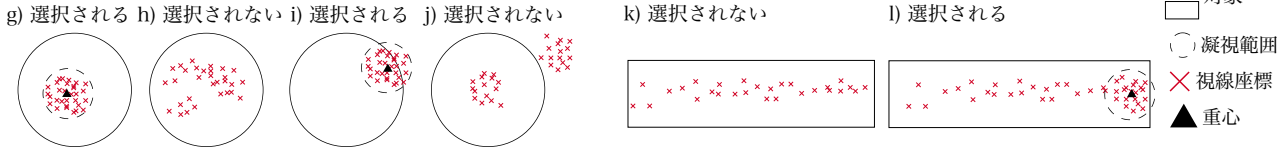


図 1. 凝視の認識方法の例. a-f) 従来手法. 対象内に視線が入り続けた場合に凝視が認識され (a, b, e, f), その対象が選択される. g-l) 本手法. 視線座標のばらつきがある範囲内に留まり続けた場合に凝視が認識され (g, i, l), 視線座標群の重心が含まれている対象が選択される. 対象内に視線座標が入り続けたとしても, 視線座標がばらついていない場合は凝視は認識されない (h, j, k).

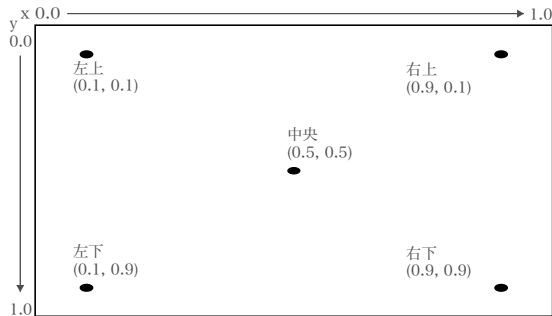


図 2. 凝視範囲を決定するためのキャリブレーションに使用した画面.

対して, 本手法では, 対象内の情報を見るために視線が移動している際に生じるミダタッチ問題を軽減できる. 例えば, 文章を読んでいる際もしくは画像を見ている際はユーザの興味に合わせて視線座標が移動する [13]. 結果として, 視線座標のばらつきが大きくなるためミダタッチ問題は生じない. 加えて, これまでに行われてきた凝視時間の調整を本手法に適用することにより, さらにミダタッチ問題を軽減できる可能性がある.

3 凝視範囲の決定方法

本研究では凝視範囲を, ある点を見ている際の視線座標のばらつきをもとに決定する. ある点を見ている際の視線認識の正確さ (ユーザが実際に見ている場所と視線座標との距離) および精度 (視線座標のばらつき) は, 視線認識機器によって示されている. そして, これらを調査した研究もある (例 [3]). また, ユーザの注視を推定する研究も多く行われている (例 [14]). 凝視範囲の決定方法をこれらを参考に行う.

凝視範囲を決定する1つの方法としては, 視線認識の精度を用いることが考えられる. 例えば, Tobii Pro X3-120 の視線認識の正確さおよび精度は公表されている¹. このように, 使用する認識機器に応じて凝視範囲を決めることにより, 利用環境に適した値を求めることができるだろう.

また, もう1つの方法として, 視線認識機器へのキャリブレーションと同じ様に凝視範囲をキャリブレーションを通じて決定することも考えられる. 視線認識機器へのキャリブレーションは, 図2のように画面上の複数箇所に表示された点をユーザが見つめることにより行われる. この際に収集される視線座標のばらつき, つまり, ある点を見つめている際の視線座標の標準偏差をもとに凝視範囲を求める. 例えば, 視線認識の正確さと精度から適切な対象の大きさを求めている研究 [3] では, 収集される視線座標の 95%が入るような対象の大きさを式1から求めている.

$$S_{w/h} = 2(O_{x/y} + 2\sigma_{x/y}) \quad (1)$$

式1を用いると, 対象の幅 S_w と高さ S_h を, 表示された点から視線座標 (x および y 座標) までの距離 $O_{x/y}$ およびばらつき (標準偏差) $\sigma_{x/y}$ から求められる. 凝視範囲も同様に, キャリブレーションを通じて収集した視線座標を用いて, $2\sigma_{x/y}$ とすることが1つの案である. なお, 公表されている視線認識機の視線認識の精度を σ として用いた場合, キャリブレーションを行わずに凝視範囲を決定できる.

凝視範囲の大きさは凝視時間と同様に, 操作性およびミダタッチ問題の発生とトレードオフの関係にあるため, 各ユーザもしくは各場面において,

¹ <https://www.tobii.com/pop-ups/accuracy-and-precision-test-report-x3/?v=2.1.7> 最終閲覧 2020 年 11 月 15 日

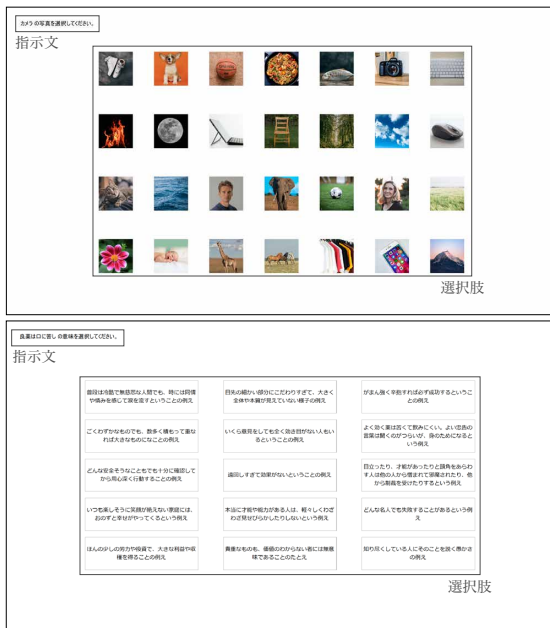


図 3. 実験に用いた UI 設計. (上) 画像タスク. (下) 文章タスク. 両画面における対象の大きさは、キャリブレーションによって決定される

どちらを優先するかによって変更するとよい。例えば、よりミダスタッチ問題の発生に対して堅牢な操作を求める場合は $1\sigma_{x/y}$ を、より誤った操作をある程度許容し簡単な操作を求める場合は $3\sigma_{x/y}$ を採用するなどである。

4 実験

本実験の目的は2つである。1つは、凝視範囲を用いた本手法が従来手法と同じように対象選択が可能であることを示すことである。もう1つは、マウスでの操作に使用されているようなUI設計において、凝視範囲を用いることによりミダスタッチ問題をどれだけ軽減できるかを示すことである。

実験に用いたUI設計は、図3(上)に示すような複数ある画像の中から1つを選択する場面(例えば、動画プレイヤーのサムネイルが表示された画面や、画像検索結果が表示された画面など)、および図3(下)に示すような複数表示された文章の中から1つを選択する場面(例えば、ウェブでの検索結果を示す画面)を想定したものである。そのため、実験に使用した対象は、塗りつぶされた円といった単純な対象ではなく、実験参加者に探索を求めるような、内部に画像および文字列が表示されている対象である。本実験の参加者は本稿の第1著者である。なお、第1著者は視線に基づく操作の研究を4年間行っており、凝視に基づく操作手法に極めて慣れている。

実験に使用した視線認識機器はプロライセンスを付与した Tobii Eye Tracker 4C (サンプリング

レート: 90 Hz (つまり、約 11.11 ms ごとに 1 サンプルの取得)) である。これを 27 インチ (解像度: 3840 × 2160 ピクセル) のディスプレイ下部に取り付け使用した。実験参加者の頭部を固定せず、移動しないようになどの指示は特になかった。実験参加者とディスプレイの距離は約 65 cm である。この条件下において視野角 1° は約 1.1 cm に対応する。

実験開始前に、視線認識の性能を高く保つために、視線認識機器のキャリブレーションを行った。その後、凝視範囲のキャリブレーションを行い凝視範囲および対象の大きさを決定した後に、図3のUI設計を用いたタスクを行った。

4.1 凝視範囲のキャリブレーション

凝視範囲を決定するためのキャリブレーションを行った。加えて、式1を用いて実験に使用する対象の大きさを決定する。

キャリブレーションには図2に示す画面を用いた。実験参加者は対象へ視線を移した後、スペースキーを押下する。その後、実験参加者は2秒間回転し続ける点を見続け、この間の視線座標が収集される。

各点を見つめている際の視線座標を収集 (2,000 ms / 11.11 ms \approx 180 サンプル) し、冒頭 500 ms (\approx 45 サンプル) を除いた 1,500 ms (\approx 135 サンプル) を用いて各点を見つめている際の視線座標のばらつき (σ') を以下の式から計算した。

$$\sigma' = \frac{1}{m-n+1} \times \sum_{i=1}^{m-n+1} SD([P_i : P_{i+n-1}]), \quad (2)$$

P_i は i 番目の視線座標、 m は計算に用いるサンプル数 (135)、 n は凝視時間 (この例では 1,000 ms) に対応するサンプル数 (90 \approx 1,000 ms / 11.11 ms) である。つまり、90 サンプル分の視線座標の標準偏差 (SD) を 1 サンプルずらしながら計算し、46 区間分の平均を σ' とした。また、各点を見つめている際の視線座標とその点の距離 (O') を、以下の式から求めた。

$$O' = \frac{1}{m-n+1} \times \sum_{i=1}^{m-n+1} Ave.([D_i : D_{i+n-1}]), \quad (3)$$

D_i は i 番目の視線座標と点までの距離、 m および n は式2と同じである。つまり、90 サンプル分の視線座標と点までの距離の平均 (Ave.) を 1 サンプルずらしながら計算し、46 区間分の平均を O' とした。

各点における x 座標、 y 座標ごとの σ' を図4に示す。本実験の目的の1つは、本手法が従来手法と同様に動作することを示すことである。そこで、式1を用いた際に、より簡単に操作が可能となる、より大きな σ' を σ 、より大きな O' を O として用いる。座標軸に依らずもっとも大きな σ' である 0.12° (約 0.14 cm) を σ 、もっとも大きな O' である 1.61° を

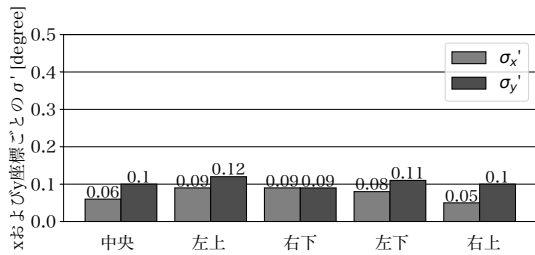


図 4. 図 2 の各点における視線座標のばらつき (σ').

O (約 1.82 cm) とした。これらを式 1 に当てはめ、対象の大きさを 3.70° ($2 \times (1.61 + 0.12 \times 2)$) とし、凝視範囲を 2σ である 0.24° とした。

4.2 実験タスク

実験参加者は、図 3 それぞれの画面において指示された対象を選択するタスクを行う。なお、図 3 (上) の対象の 1 辺を 3.78° とし、図 3 (下) の対象の高さを 3.78° 、幅を表示される文章に合わせて 11.0° (約 12.4 cm) とした。

図 3 (上) を用いたタスク (以降、画像タスク) では、指示文に示された、例えば「カメラを選択する」という指示に従い、「カメラ」が表示されている画像を選択する。指示は 28 種類であり、指示に対応した 28 種類の対象 (選択肢) が表示される。全 28 種類の指示に対して、正誤問わずいずれかの対象を選択すると画像タスクは終了する。

図 3 (下) を用いたタスク (以降、文章タスク) では、指示文に示された、例えば「良薬は口に苦しの意味を選択する」という指示に従い、「良薬は口に苦し」の意味が書かれている文章を選択する。指示は 28 種類であり、それぞれの指示に対して 15 個の対象 (選択肢) が表示される。全 28 種類の指示に対し、正誤問わずいずれかの対象を選択を行うと文章タスクは終了する。指示は全て、ことわざの意味を選択させるものである。

両タスクにおいて、タスク開始時は指示文のみが表示されている。実験参加者が指示を読んだ後にスペースキーを押下すると対象が表示され、いずれかの対象が選択されると、対象を非表示とし次の指示が表示される。両タスクともに指示の順番はランダムである。各対象の位置は指示ごとにランダムとし、指示に対して選択すべき対象の位置が重複しないようにした。スペースキーが押下され対象が選択されるまでの時間を選択に要した時間とし、また指示に対して誤った対象が選択された回数を記録した。

実験参加者には、ミダスタッチ問題が生じない様に意識し、かつ素早くタスクを行うよう指示した。例えば、第 1 著者は頻繁に視線を動かすよう意識することにより、ミダスタッチが生じないよう工夫した。実験を行った順番は、画像タスク (本手法 → 従来手法)、文章タスク (本手法 → 従来手法) で

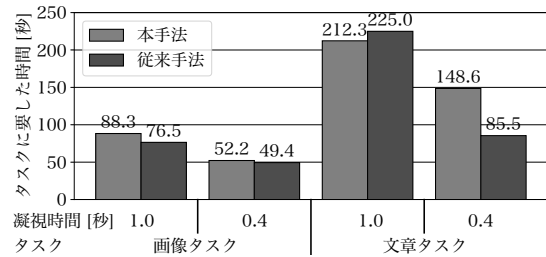


図 5. 選択に要した時間.

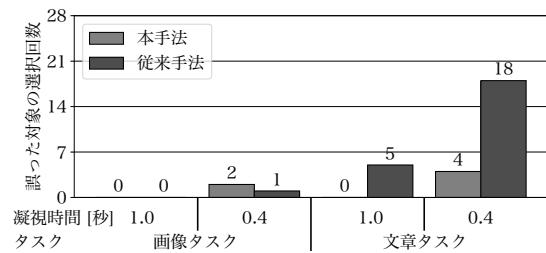


図 6. 誤った対象が選択された回数.

ある。凝視時間を 1 秒とした実験を行った後に、同一の手順にて凝視時間を 0.4 秒とした実験を行った。なお、0.4 秒は凝視時間の調整する研究において下限の値として用いられている時間 [11] であり、さらに長い時間として 1 秒での実験も行った。

4.3 実験結果

実験結果を図 5 および図 6 に示す。画像タスクにおいては凝視時間を 1.0 秒、0.4 秒とした場合において、選択に要した時間および誤った対象が選択された回数に大きな差は見られなかった。対して文章タスクにおいて、誤った対象が選択された回数は凝視時間を 1.0 秒とした時に本手法では 0 回であり従来手法では 5 回、凝視時間を 0.4 秒とした時に本手法では 4 回であり従来手法では 18 回であった。また、文章タスクでの凝視時間が 0.4 秒の場合、選択に要した時間は本手法では 148.6 秒であるのに対し、従来手法では 85.5 秒であった。この結果において従来手法の選択に要した時間が小さくなった理由は、従来手法では誤った対象が選択された回数が多いためであると考えられる。つまり、ミダスタッチ問題が発生し誤った選択が生じたことにより、実験参加者が適切な回答を探索し終わる前にタスクが終了することが多く生じたために、選択に要した時間が小さくなったと考えられる。

また、文章タスクでは、誤った対象が選択された回数が本手法の方が少なくなった。文章タスクでは画像タスクと比べ、実験参加者は適切な対象を選択するためにより長い時間の探索が求められる。そのため、選択に要した時間は大きくなる。また、指示に対して、回答を考えながら文章を読むため、対象

内に視線座標が入り続ける時間は増える。従来手法では、対象内に視線座標が入っている時間をもとに凝視を認識するため、ミダスタッチ問題が多く発生したと考えられる。対して本手法では、文章を読んでいる際、つまり対象内を視線座標が移動している際は凝視が認識されないため、ミダスタッチ問題の発生が少なくなったと考えられる。

凝視時間を0.4秒とした場合、本手法においても誤った対象選択は生じた。この誤った対象選択が生じた際の視線移動の軌跡を調べたところ、文章を読んでいるように視線は移動していた。文章を読んでいる際の注視に要する時間は平均0.25秒であり0.1–0.5秒内に収まる[13]。そのため、凝視時間を0.5秒以下とした場合は、凝視範囲を用いたとしても、必然的に誤った対象選択は生じる。対して、凝視時間を1.0秒とした場合、本手法では両タスクにおいて誤った対象選択は生じていない。

これらの結果から、画像タスクのように視線が大きく移動しないような場合では、本手法と従来手法は同様に対象選択が行え、文章タスクのように視線が大きく移動するような場合では、従来手法においてある程度長い凝視時間（1秒）を用いても軽減できなかったミダスタッチ問題を、本手法は軽減できることがわかった。しかし、今回行った実験では著者のみが実験参加者であったため、今後さらに実験規模を大きくした調査を行う必要がある。

5 今後の発展

本手法の今後の発展に関して述べる。

5.1 ミダスタッチ問題のさらなる軽減

本手法は凝視時間の調整を行わず、凝視範囲を用いることによりミダスタッチ問題を軽減している。そこで、本手法においても凝視時間の調整（例[11, 16]）を行うことにより、さらなるミダスタッチ問題の軽減を試みる。なお、凝視範囲を用いることにより、凝視範囲の調整を行ってきた研究における選択対象であった、無地もしくは単語、簡単な画像が表示された円や正方形以外の、より一般的に使用されているようなUI設計での対象選択への調査も可能となる。

5.2 文脈に沿った操作の実現

本手法では凝視の認識は対象の外でも行える。そのため、対象選択以外の操作を凝視により可能となる。例えば、対象外にて凝視が認識された際に、メニュー表示（マウスでの操作時における右クリックに近い操作）や、表示したメニューの非表示ができる。このように、操作の文脈によって凝視により可能な操作を変えることは、視線ジェスチャの開始点の決定[6]や、視線での描画時の開始点および終了点の決定[4]においても行われてきた。しかし、これらの研究では対象の選択は行われていない。

ただし、対象外への操作を凝視により行う場合は、ミダスタッチ問題の発生を防ぐための工夫が必要である。例えば[6]では視線ジェスチャが完了するまで操作を行わないようにしており、[4]ではミダスタッチ問題が発生しないように音声や視覚的なフィードバックを用いて視線を移動するよう促している。まずミダスタッチ問題の解決をする必要があるが、凝視範囲を用いることにより文脈に沿った操作が可能となることは、視線に基づく操作への大きな貢献となるだろう。

5.3 操作性能の向上

ミダスタッチ問題の軽減に加え、凝視に基づく操作性能の向上を目的とした手法（例[2, 18]）を取り入れることもできる。これらの多くは、マウスでの操作性能を向上させる手法を従来手法に取り入れたものである。従来手法と比べ用途がマウスでの操作に似ている本手法においては、これらの研究で得られた知見を特に享受しやすいだろう。

フィルタを適用することにより、視線認識時に生じるマイクロサッカードなどを抑えることができる[3]。そのため、視線認識を行う機器や人、環境が与える、ある点を見つめている際の視線座標のばらつきへの影響を少なくすることができるだろう。しかし、フィルタの種類によっては遅延が生じる。そのため、適切なフィルタを選択する必要がある。

5.4 短い凝視時間の使用

凝視時間を0.4秒としたところ、本手法においても誤った選択が生じた。本手法、従来手法ともに、0.4秒以下の短い凝視時間を用いる場合、専用のUIの使用（例[10, 17]）、輻輳効果の使用（例[9]）、追跡眼球運動の使用（例[15]）が必要となるだろう。もしくは、対象へ視線を移す時の視線移動の軌跡を観察すること（例[2, 5]）が考えられる。

6 まとめ

本稿において、新たな凝視の認識手法による、凝視に基づく操作手法を示した。従来手法が「対象の中に視線座標が凝視時間以上留まっていること」を条件に凝視を認識していたことに対し、本手法では「視線座標のばらつきが凝視範囲に凝視時間以上留まっていること」を条件とする。本手法によって主に、対象内に視線座標が入っているが視線が移動している場合に生じるミダスタッチ問題の軽減できる。例えば、ウェブページのリンクを読んでいる際や、動画プレイヤーのサムネイルを見ている際に生じるミダスタッチ問題の軽減が可能である。本稿では、著者による実験を通じて、本手法は従来手法と同じ様に操作ができ、対象内に視線座標が入っているが視線が移動している場合に生じるミダスタッチ問題を軽減できる可能性を示した。

これまで従来手法におけるミダタッチ問題の軽減や性能向上を目的とした手法は、本手法においても適応可能である。さらに、本手法はマウスやタッチでの操作と似た操作であるため、従来手法に比べよりこれらの操作に関する研究にて得られてきた知見を享受することができるだろう。著者らは、凝視範囲を用いた凝視に基づく操作手法により、視線に基づく操作の可能性がさらに広がり、より視線に基づく操作手法が普及すると考えている。

謝辞

本研究は、公益財団法人立石科学技術振興財団の2020年度研究助成(C)[博士後期課程対象]を一部受けたものです。

参考文献

- [1] W. Abd-Elmageed, M. S. Fadali, and G. Bebis. A Non-intrusive Kalman Filter-based Tracker for Pursuit Eye Movement. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, Vol. 2, pp. 1443–1447. IEEE, 2002.
- [2] M. Choi, D. Sakamoto, and T. Ono. Bubble Gaze Cursor + Bubble Gaze Lens: Applying Area Cursor Technique to Eye-Gaze Interface. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20, pp. 11:1–10. ACM, 2020.
- [3] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. Morris. Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 1118–1130. ACM, 2017.
- [4] A. J. Hornof and A. Cavender. EyeDraw: Enabling Children with Severe Motor Impairments to Draw with Their Eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pp. 161–170. ACM, 2005.
- [5] T. Isomoto, T. Ando, B. Shizuki, and S. Takahashi. Dwell Time Reduction Technique Using Fitts' Law for Gaze-based Target Acquisition. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pp. 26:1–26:7. ACM, 2018.
- [6] T. Isomoto, S. Yamanaka, and B. Shizuki. Gaze-based Command Activation Technique Robust Against Unintentional Activation using Dwell-then-Gesture. In *Proceedings of the 46th Graphics Interface Conference on Proceedings of Graphics Interface 2020*, GI '20, pp. 24:1–11. CHCCS, 2020.
- [7] R. J. K. Jacob. What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proceedings of the 1990 CHI Conference on Human Factors in Computing Systems*, CHI '90, pp. 11–18. ACM, 1990.
- [8] Y. Kammerer and W. Beinbauer. Gaze-Based Web Search: The Impact of Interface Design on Search Result Selection. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pp. 191–194. ACM, 2010.
- [9] S. Kudo, H. Okabe, T. Hachisu, M. Sato, S. Fukushima, and H. Kajimoto. Input Method Using Divergence Eye Movement. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pp. 1335–1340. ACM, 2013.
- [10] C. Lutteroth, M. Penkar, and G. Weber. Gaze vs. Mouse: A Fast and Accurate Gaze-Only Click Alternative. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pp. 385–394. ACM, 2015.
- [11] A. Nayyar, U. Dwivedi, K. Ahuja, N. Rajput, S. Nagar, and K. Dey. OptiDwell: Intelligent Adjustment of Dwell Click Time. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI '17, pp. 193–204. ACM, 2017.
- [12] A. M. Penkar, C. Lutteroth, and G. Weber. Designing for the Eye: Design Parameters for Dwell in Gaze Interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pp. 479–488. ACM, 2012.
- [13] K. Rayner. Eye Movements and Cognitive Processes in Reading, Visual Search, and Scene Perception. In J. M. Findlay, R. Walker, and R. W. Kentridge eds., *Eye Movement Research*, Vol. 6 of *Studies in Visual Information Processing*, pp. 3–22. North-Holland, NLD, 1995.
- [14] D. D. Salvucci. Inferring Intent in Eye-Based Interfaces: Tracing Eye Movements with Process Models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 254–261. ACM, 1999.
- [15] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pp. 439–448. ACM, 2013.
- [16] O. Špakov and D. Miniotas. On-Line Adjustment of Dwell Time for Target Selection by Gaze. In *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction*, pp. 203–206. ACM, 2004.
- [17] C. Ware and H. H. Mikaelian. An Evaluation of an Eye Tracker As a Device for Computer Input. In *Proceedings of the 1987 CHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pp. 183–188. ACM, 1987.
- [18] 崔 明根, 坂本 大介, 小野 哲雄. Bubble Gaze Cursor : バブルカーソル法の視線操作への適用. 情報処理学会論文誌, 61(2):221–232, 2020.