

VRGazeWarp: VRHMDに搭載されたアイトラッカーとVRコントローラを組み合わせたポインティング手法

阿部 広河^{1,a)} 石川 博規² 真鍋 宏幸^{1,b)}

概要: アイトラッカーを搭載したVR(Virtual Reality)向けHMD(Head Mounted Display)に対する注目が高まっている。特に、注視箇所周辺の高画質化とシステム負荷の軽減を両立させるFoveated Renderingという技術が搭載された製品が発表されるなど、アイトラッカーを活用した事例も増えてきている。アイトラッカーは昔から、インタラクション手法への活用が研究されているが、VRHMDに搭載されたアイトラッカーを用いたものに関する研究は、まだ十分に行われていたとは言えない。また、アイトラッカーを用いた手法は、ポインタの微調整が難しいという問題が存在する。本研究では、HMDに搭載されたアイトラッカーとVRコントローラを同時に用いたポインティング手法であるVRGazeWarpを提案し、この問題の解決を試みる。実験の結果、性能では提案手法を超えることはできなかったものの、アイトラッカーを用いた手法の、特有の問題に対して対策を行うことで、SUSスコアが向上することが確認できた。

1. はじめに

一般ユーザー向けのVirtual Reality(VR)市場は、特にエンターテインメントの分野で規模が広がっており、今後も市場規模が拡大すると見られている。VRコンテンツを利用するためのHMDに関する研究も数多く存在しており、近年ではFoveated Renderingという技術[1][2]が注目されている。これは、ユーザーが注視している箇所とその周囲のみレンダリング解像度を向上させ、画面の高画質化とシステム負荷の軽減を両立させる技術である。また、このFoveated Renderingを、VRHMDに組み込んだアイトラッカーを用いて行う研究[3][4]も多く存在しているほか、性能だけではなくVRHMDを用いた際の画面酔いにも効果があるという研究結果[5]が存在する。さらに、PSVR2^{*1}ではアイトラッカーを使用し、既にFoveated Renderingを組み込んでいる例も存在する。今後更にFoveated Renderingが注目されることで、アイトラッカーを搭載する製品も増加すると考えられるが、アイトラッカーは昔からポインティングデバイスとして活用するための研究[6][7]も盛んに行われている。例としてVRHMDに搭載されたアイトラッカーと筋電計を用いたインタラクションに関する研究[8]や、アイトラッカーのみでインタラクションを行う

ための研究[9][10]などが存在する。他にも、コントローラのボタン入力によって、ポインタの移動方法を視線とコントローラで切り替える手法の研究[11]などが存在する。これらの既存研究はあるが、VRHMDに搭載されたアイトラッカーを使用したポインティング手法について十分に研究が行われているとは言えない。

本研究ではVRHMDに搭載されたアイトラッカーとコントローラを組み合わせた手法である、VRGazeWarpを提案する。著者の以前の研究[12]では、同じくアイトラッカーとコントローラを組み合わせた手法を提案している。しかし、この手法はキーボードなどのユーザーインターフェースを操作することに特化した手法であり、ボタン以外の任意の場所にユーザーがポインタを動かすことは難しいという欠点が存在していた。提案手法では、ユーザーが自由にポインタを動かしたいような状況において、小さいターゲットをポインティングしやすくなることを想定している。

2. 提案手法

提案手法であるVRGazeWarpは、ポインタの操作にアイトラッカーとコントローラを使用した手法である。アイトラッカーを用いたポインティングでは、ターゲットサイズが小さい場合にポインティング性能が低下するという問題[13]が存在する。これはVRHMDに搭載されたアイトラッカーを使用する場合でも同じであると考えられる。そのため、VRコントローラを併用することでアイトラッ

¹ 芝浦工業大学

² NTTドコモ

a) ma22009@shibaura-it.ac.jp

b) manabe@shibaura-it.ac.jp

*1 <https://www.playstation.com/ja-jp/ps-vr2/>

カーで行うことが困難な細かいポインティングを効率的に行うことが本研究の目的である。これにより、遠い距離にあるターゲットの高速なポインティングと、近い距離の正確なポインティングを両立させる。具体的には、視線でポインタを遠い距離に素早く移動させ、コントローラでより細かい、小さな距離の調整を行う。この手法は、ターゲットを見て狙うような動作が視線入力と相性が良く、シューティングゲームのような状況で効果的であると考えられる。一方、入力する文字などを確認するために、ターゲットとは異なる位置に視線を動かさなければならない状況が多く発生することから、ソフトウェアキーボードなどを用いた文字入力に適さない。

次に、提案手法の具体的な動作例を図1を用いて説明する。ステップ0：ポインタは始め、HMDの視界の中心に表示されている。ステップ1, 2：ポインタの動きは、コントローラの回転動作によって制御することが可能である。例えば、コントローラを右に回転させると、ポインタも右に動く。左、上、下への動きも同じようにコントローラで操作することが可能である。また、ポインタの動きは、コントローラの動きに感度を乗算したものが反映される。ステップ3, 4：ユーザーが目で特定の場所を注視すると、コントローラの現在の位置や回転に関係なく、ポインタをその見ている場所へ瞬時にワープ（瞬間移動）させる。ステップ5, 6：ワープ後もコントローラをそのまま使用して、ポインタを微調整することができる。コントローラの回転には相対角度を使用しているため、ポインタの絶対位置に合わせてコントローラを大きく動かす必要はない。以降ではそれぞれのステップについて詳しく説明する。

2.1 コントローラによるポインタの移動 (ステップ0~2)

初期状態では図1のステップ0のように、HMDの正面方向に向かって1.5メートル先の位置にポインタが配置されている。このポインタは、ステップ1からステップ2のようにHMDを中心にして回転させることで、ポインティング先を変更することが可能である。ポインタの動きはコントローラを使って制御され、コントローラが回転する角度と設定された感度に応じてポインタも回転する。例えば、感度が0.5に設定されている場合、コントローラを右に90度回転させると、ポインタは右に45度(90度×0.5)回転する。今回の実装では、感度を0.6に設定した。ここでは2次元空間のオイラー角を用いて説明を行ったが、実際の計算は3次元空間のクォータニオン角を用いて行われる。

2.2 視線によるポインタのワープ (ステップ3, 4)

ユーザーが注視状態となったとき、注視した場所へポインタがワープする。注視状態とは、ユーザーの視線が一定時間移動せずに、その点を見つめていることを指す。ポインタのワープは、ユーザーが任意の場所を注視した際にコ

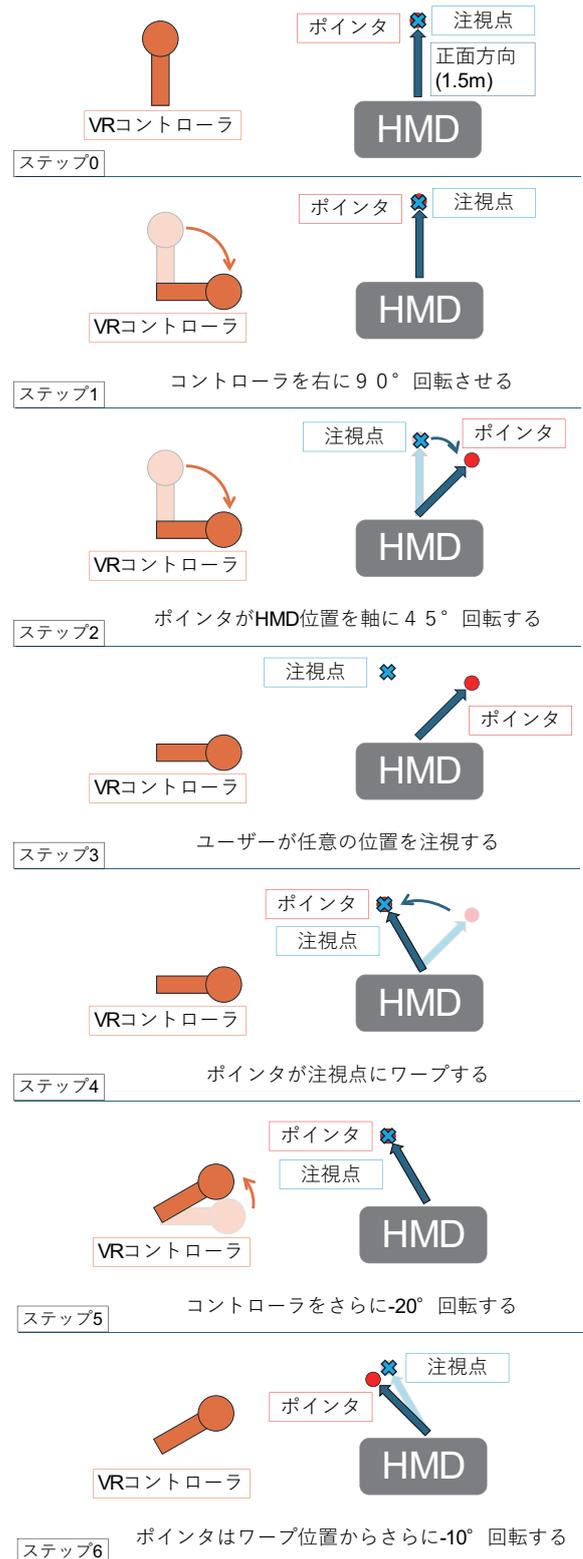


図1 提案手法によるポインタ操作の一連の流れ。

ントローラの現在の回転を打ち消す処理と、注視点に向かって回転させる処理の2つによって行っている。具体的にはまず、現在のコントローラの回転を打ち消して、ポインタがHMDの正面に移動するような反対ベクトルが計算される。次に、ポインタがユーザーが見ている場所に移動するようにポインタを回転させる。これにより、コントロー

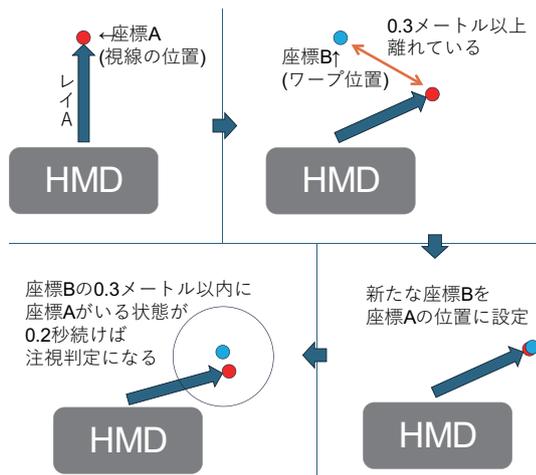


図 2 注視しているかどうかを判定する流れ。

ラがどの方向を向いていても、ユーザーが見ている場所にポインタを瞬時に移動させることが可能となる。また、ステップ5からステップ6のようにワープ後もそこを基準として、コントローラを使ってポインタを移動させることができる。

2.3 視線によるワープの有無の判定

この手法に用いる視線情報は、HMD に組み込まれたアイトラッキング機能を使用する。まず、図2の左の図のようにHMDの位置を基準点として、ユーザーが見ている方向に向かって伸びる仮定の光線(レイA)を考える。ここで、レイA上でHMDから3メートル離れた点を座標A(つまり、座標Aは視線位置)とし、ポインタ位置を座標Bとし、座標Aと直前の座標Bの間の距離を計算する。距離が0.3メートル以上離れていて、かつ注視状態が0.3秒以上継続した場合、視線が移動したとみなして、座標Bを座標Aの位置に更新する。逆に、距離が0.3メートル未満あるいは注視状態が0.3秒未満の場合、視線によるワープは行わず、座標Bは直前の位置を維持する。

3. 1回目の評価実験

この実験では、提案手法がどのような状況で有効に機能するかを調査する。3つの異なる状況で提案手法を試し、どのような状況で一番パフォーマンスが高いかを調べる。また、システムユーザビリティスケール(SUS)スコアを用いて、ユーザーからの主観的な評価も実施する。実験には、提案手法が得意な状況を想定したシューティングタスクと、苦手な状況を想定した文字入力タスクを用いる。

3.1 タスク内容

実験では2つのタスクを被験者に行わせた。

3.1.1 シューティングタスク

1つ目は、ゲームでの利用を想定したタスク(図3左)

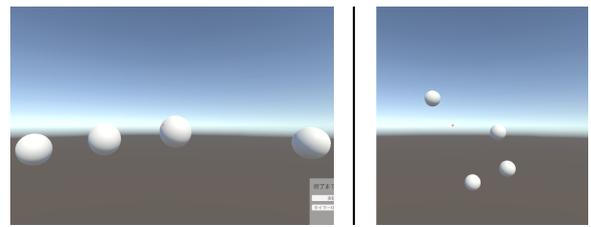


図 3 シューティングタスクのスクリーンショット。大きいターゲットの場合(左)と小さいターゲットの場合(右)。



図 4 文字入力タスクのスクリーンショット。

である。このタスクはシューティングゲームのようなもので、画面に4つの白い球が表示されている。各球の直径は0.3メートルで、被験者はポインタをこれらの球に合わせて、コントローラのボタンを押す。ポインタが球に合っていて、ボタンが押されると、その球はランダムな新しい位置に移動する。新しい位置は被験者の正面方向に2メートル離れた縦3メートル、横2メートルの領域内からランダムに選ばれる。このタスクを3分間の間繰り返す。タスク中にボタンを押したときにポインタが球に合致していた回数と、合致していなかった回数を計測する。さらに、球の大きさが1/2になっている状況でのシューティングタスク(図3右)も追加で行った。

3.1.2 文字入力タスク

2つ目のタスクは文字入力である(図4)。実験中は、英語配列のQWERTYソフトウェアキーボードと英語の例文が、画面上に表示されている。この例文は、予め用意された41個のパングラムの中からランダムに一つ選ばれたものである。パングラムとは、アルファベットすべての文字を含み、できるだけ重複がないように構成された文章のことである。

被験者はソフトウェアキーボードを用いて、表示された例文の通りに文章を入力する。文章を入力し終えた後、returnキーを押すことにより文章の入力を確定する。その後、新しいランダムなパングラムが表示され、被験者はその表示された文章を入力するタスクを、制限時間の間繰り返す。

3.2 実験内容

6名の被験者に対して前述の2つのタスクと、SUSスコアを計測するための実験後アンケートを実施した。6名は

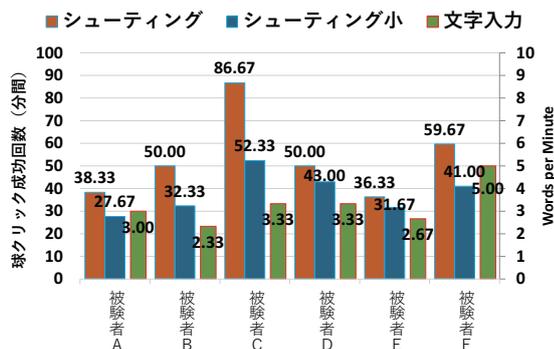


図 5 6名の被験者ごとの1分間の球クリック回数（左目盛り）と Words per Minute（右目盛り）

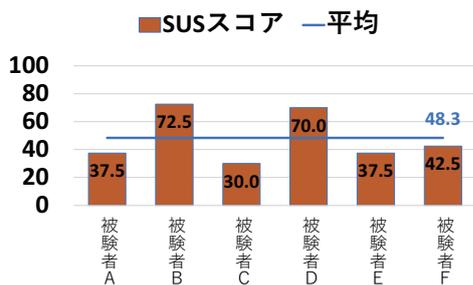


図 6 6名の被験者ごとの SUS スコアとその平均

全員が学生または大学院生であり、いずれも英語のネイティブスピーカーではない。実験では各タスクの前に、2分間の練習時間を設け、その後3分間の実験を行った。タスクの順序は、学習効果を考慮してランダムに決定した。

3.3 実験結果

それぞれのタスクにおけるクリックの成功回数と Words per Minute (WPM) を表したものが図 5 である。まず、大きいターゲットを用いたシューティングタスクにおけるクリックの成功回数は、1分間でおおよそ 30 回から 90 回ほどであった。小さいターゲットを用いた場合では1分間でおおよそ 20 回から 50 回ほどであり、被験者全員は小さいターゲットより大きいターゲットの方が成功回数が多かった。また、文字入力タスクでは被験者全員が3分間で3つ目の文章まで入力することができず、WPM は3から5ほどを記録した。次に、SUS スコアには被験者毎にとっても大きいばらつきがあり、最低 30 点で最高 72.5 点、平均値が 48.33 点という、かなり低い点数を記録した (図 6)。

3.4 追加実験

SUS アンケートにおいて、筆者の予想よりも「この入力手法をもっと頻繁に使いたい」という回答が少なかったことが確認できた。さらに、6人全員が文字入力タスクで手法が使いづらいと回答していた。

実際に、被験者全員は文字入力タスクにおいて WPM が最大で5という、とても低い結果になっている。そのため、

文字入力タスクでは手法がとても使いづらく、その印象が SUS スコアに影響を及ぼしているのではないかと考えた。そこで、文字入力タスクを行わない追加の実験を行った。この追加実験は3名の被験者に対して行い、文字入力タスクが除外されたこと以外は先の実験と同じ条件で実験を行った。

まず、SUS スコアは文字入力タスクを行った場合と比較しても、変わらず低い水準であるという結果になった。被験者は全員、1分間で球を50個以上クリックすることはできず、ターゲットが小さい場合はその半数以下の回数になってしまう結果も確認できた。

3.5 手法の問題点

追加実験の結果から、得意な状況と考えていたシューティングタスクでも手法に問題があることが分かった。だがこの問題は、文字入力タスクのときは異なる問題なのではないかと考えた。まず、文字入力タスクでは、文字を入力している間に例文や入力文を一瞬見た際に、その箇所にポインタが移動してしまうという問題が考えられた。これは実際に、実験中では文字の誤入力を起こした後に、バックスペースキーを押しているタイミングで確認できた。具体的には、バックスペースキーで文字を消しながら、入力文見てどこまで消したのかを確認するタイミングで、ポインタがワープしてしまうという状況である。それとは異なり、シューティングタスクでは、視界の端でポインティングしようとする場合に、ポインティング精度が低下してしまうという問題が確認できた。シューティングタスクは文字入力タスクと異なり、視界の端にターゲットが存在している状況が多く、その場合にうまくポインティングできないことが SUS スコアが低くなってしまった問題の原因なのではないかと考えた。

4. 手法の改良

実験結果を踏まえ、手法の改良を行った。まず、視線の滞留時間に基づく注視判定の基準を、0.3秒から0.15秒へと変更した。これは、被験者のポインタの動きを実験中に観察した結果、ポインタがワープするまでの時間が想定よりも長いことに気がついたためである。さらに、ポインタをワープさせる際に、視線の移動の方法が被験者毎に異なっていることと、主にそれが2種類あることが分かった。一つ目は頭ごと視界を動かす方法、もう一つは頭をほとんど動かさずに、視線だけを動かす方法である。だが後者の方法では、視線が視界の端にある場合に視線が大きく揺れることがあり、それが原因で注視判定の範囲から視線が外れてしまっていた。そのため、HMDの正面方向から視線が離れていれば離れているほど、注視判定の範囲を大きくする改良を加えた。まず、HMDから正面方向に3メートル離れた点を新たに座標Cとする。この座標Cは常に

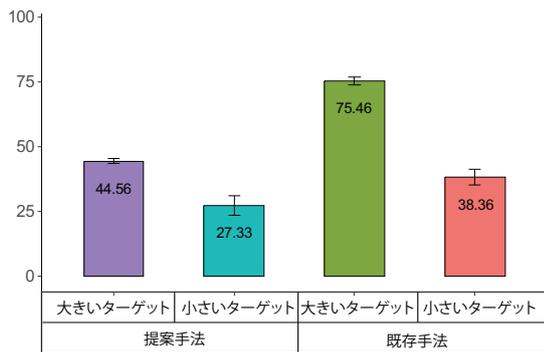


図 7 それぞれの手法、タスクにおけるターゲットのクリック成功回数の 1 分間における平均 (エラーバーは標準偏差)。

HMD の正面方向に固定されており、視線やコントローラで動かすことはできない。この座標 C と視線方向にある座標 A の距離を Vector3.Distance 関数で計算し、その絶対値を注視判定の範囲に乗算する。これにより、視線が HMD の正面方向から離れるほど注視判定の範囲が大きくなる。また、注視判定の範囲が 0.3 より小さくなる場合は 0.3 を最低値とし、それより小さくはならないようにした。

5. 2 回目の評価実験

5.1 実験内容

2 回目の評価実験は、1 回目の実験と同様のシューティングタスクを行った。ただし、手法の有効な場面とそうでない場面をより明確にするために、タスクの内容に一部変更を加えた。それは、シューティングタスクの中でも小さい的を用いたタスクにて、前回は球のサイズを 1/2 に設定していたものを、今回は 1/4 のサイズに変更するという内容である。被験者は小さい的でのシューティングタスクは、より詳細にポイントを動かす必要がある。また、提案手法のポインティング性能の比較も行うため、VR では一般的な手法である、レイキャストによるコントローラポインティング法も同条件で実験を行った。

被験者は 6 名の学部学生または大学院生であり、一度目の実験にはいずれも参加していない。この 6 名の被験者に対して前述したタスクと、SUS スコアを計測するための実験後アンケートを実施した。実験では各タスクの前に、2 分間の練習時間を設け、その後 3 分間の実験を行った。タスクの順序は、学習効果を考慮してランダムに決定した。

5.2 実験結果

提案手法と既存手法をシューティングタスクで用いた比較実験の結果が、図 7 である。提案手法では、大きいターゲットを用いたタスクでは 1 分間で平均 44.58 回の成功クリック回数、小さいターゲットを用いたタスクでは 1 分間で平均 27.33 回の成功クリック回数を記録した。また既存

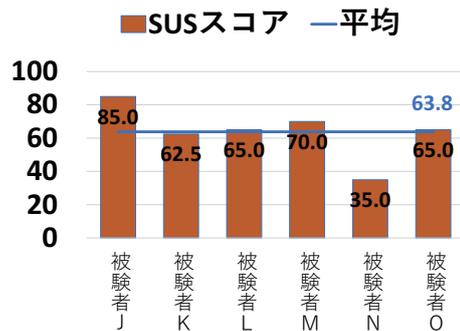


図 8 各被験者ごとの SUS スコアと平均

手法では、大きいターゲットを用いたタスクでは 1 分間で平均 75.46 回の成功クリック回数、小さいターゲットを用いたタスクでは 1 分間で平均 38.36 回の成功クリック回数を記録した。また、ANOVA を用いて有意差の確認を行い、有意差が確認できたため、Tukey-Kramer 法を用いた多重比較を行ったところ、全てがその他に対して有意差があることが確認できた。

提案手法のみを対象とした SUS スコア (図 8) は、改良前と比べて向上していることが確認できた。平均スコアは 63.8 点であり、1 回目の実験と比較した場合は約 15 点ほど、2 回目の実験と比較した場合は約 25 点ほど上昇していた。

6. 議論

6.1 SUS スコア

SUS スコアは、手法の改良によって向上した。考えられる理由として、視界際での注視判定にあいまいさを増やしたことが挙げられる。まず、改良前の手法では視界際では注視によるワープが行いづらいう問題点が存在した。そのためユーザーは、一定時間ターゲット付近を注視した後、カーソルがワープしないことを確認した後、感度の低いコントローラ操作でカーソルを長距離、移動させなければいけない状況が発生していた。だが、視界際での注視判定にあいまいさを増やす改良によって、ワープが視界際でも発生させやすくなったことから、SUS スコアが向上したのではないかと考えられる。しかし、改良後もまれに注視によるワープが行われないことがあるため、より注視判定の精度を向上させ、ユーザーが意図したタイミングで正確にカーソルをワープすることができれば、さらに SUS スコアを向上させることは可能なのではないかと考える。ただし、SUS スコアはさらに向上させる必要がある。そのためどのような改善を行うのか、今後の検討が必要である。

6.2 クリック成功回数

クリック成功回数で手法の性能を比較すると、大きいターゲットの場合でも小さいターゲットの場合でも提案手

法は既存手法を超えることはできなかった。しかし、提案手法は既存手法と比較して、小さいターゲットにおいて性能の低下がしばらくということが確認できた。既存手法では、大きいターゲットの場合と比較して、クリック成功回数が約 49 %ほど減少しているのに対し、既存手法では約 38 %ほどの低下となっている。これは、コントローラによる微調整で、小さいターゲットに対してより素早くポインティングできたことが影響しているのではないかと考える。また、クリック成功回数は改良前と改良後であまり違いが見受けられなかった。考えられる理由の一つとして、コントローラで操作できる範囲が無制限であることが挙げられる。本来、コントローラでのポインタの操作は感度が低く設定されているため、ポインタを長距離移動させるためには、既存手法と比べてもより大きくコントローラを動かさなければならない。しかし、ユーザーはカーソルのワープを使用せずに、コントローラを大きく動かすことでターゲットへポインティングしようとしている状況を、実験中に何度か確認した。これによりポインティング速度が低下し、ワープを行いやすくしてもクリック成功回数の増加に結びつかなかったのではないかと考えた。この問題を解決するためには、そのような状況がユーザー自身の意図で行ったものなのか、手法の不具合で本意ながら行ったことなのかを明確にする必要があると考える。もし前者が主な要因な場合、コントローラで操作できる範囲を、視線の一定範囲内などに制限することで、視線による操作の重要性をより向上し、長距離移動の速度とコントローラによるポインタの微調整の精度をより向上できると考えられる。

そのためには、どのぐらいの距離ならカーソルをワープさせるのが適切なのか、またどのぐらいの範囲ならコントローラで動かしても性能や SUS スコアに影響が出ないかといったパラメータ調整を適切に行う必要がある。後者が主な要因の場合は、SUS スコアを向上させる場合と同じで、ワープのための注視判定の精度をより向上させ、ユーザーが狙ったタイミングで的確にワープを行わせるようにさらなる改良を行う必要がある。

7. おわりに

本研究では、VRHMD に搭載されたアイトラッカーとコントローラを組み合わせた、VR コンテンツ向けの新しいポインティング手法について提案した。改良前の提案手法では性能、SUS スコア共に低い水準だったが、改良により SUS スコアの向上が確認できた。

参考文献

- [1] Meng, X., Du, R., JaJa, J. F. and Varshney, A.: 3D-Kernel Foveated Rendering for Light Fields, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 27, No. 8, pp. 3350–3360 (online), DOI: 10.1109/TVCG.2020.2975801 (2021).
- [2] Tariq, T., Tursun, C. and Didyk, P.: Noise-Based Enhancement for Foveated Rendering, *ACM Trans. Graph.*, Vol. 41, No. 4 (online), DOI: 10.1145/3528223.3530101 (2022).
- [3] Matthews, S. L., Uribe-Quevedo, A. and Theodorou, A.: Rendering Optimizations for Virtual Reality Using Eye-Tracking, *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pp. 398–405 (online), DOI: 10.1109/SVR51698.2020.00066 (2020).
- [4] Meng, Xiaoxu and Du, Ruofei and Zwicker, Matthias and Varshney, Amitabh: Kernel Foveated Rendering, *Proc. ACM Comput. Graph. Interact. Tech.*, Vol. 1, No. 1 (online), DOI: 10.1145/3203199 (2018).
- [5] Pai, Y. S., Tag, B., Outram, B., Vontin, N., Sugiura, K. and Kunze, K.: GazeSim: Simulating Foveated Rendering Using Depth in Eye Gaze for VR, *ACM SIGGRAPH 2016 Posters*, SIGGRAPH '16, (online), DOI: 10.1145/2945078.2945153 (2016).
- [6] Zhai, S., Morimoto, C. and Ihde, S.: Manual and Gaze Input Cascaded (MAGIC) Pointing, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, New York, NY, USA, Association for Computing Machinery, p. 246–253 (online), DOI: 10.1145/302979.303053 (1999).
- [7] Heikkilä, H. and Rähkä, K.-J.: Simple Gaze Gestures and the Closure of the Eyes as an Interaction Technique, *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, p. 147–154 (online), DOI: 10.1145/2168556.2168579 (2012).
- [8] Pai, Y., Dingler, T. and Kunze, K.: Assessing hands-free interactions for VR using eye gaze and electromyography, *Virtual Reality*, Vol. 23, No. 2, pp. 119–131 (online), DOI: 10.1007/s10055-018-0371-2 (2019).
- [9] Piumsomboon, T., Lee, G., Lindeman, R. W. and Billingham, M.: Exploring natural eye-gaze-based interaction for immersive virtual reality, *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 36–39 (online), DOI: 10.1109/3DUI.2017.7893315 (2017).
- [10] Mohan, P., Goh, W. B., Fu, C.-W. and Yeung, S.-K.: DualGaze: Addressing the Midas Touch Problem in Gaze Mediated VR Interaction, *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 79–84 (online), DOI: 10.1109/ISMAR-Adjunct.2018.00039 (2018).
- [11] 柿沼 育, 小宮山 撰: VR 空間における視線ポインティングにコントローラを併用する効果, *ヒューマンインタフェース学会論文誌*, Vol. 23, No. 1, pp. 89–100 (オンライン), DOI: 10.11184/his.23.1.89 (2021).
- [12] Abe, Kouga and Ishikawa, Hironori and Manabe, Hiroyuki: A Hybrid Method Using Gaze and Controller for Targeting Tiny Targets in VR While Lying down, *HCI International 2023 Posters*, Cham, Springer Nature Switzerland, pp. 139–146 (2023).
- [13] Ware, C. and Mikaelian, H. H.: An Evaluation of an Eye Tracker as a Device for Computer Input2, *SIGCHI Bull.*, Vol. 17, No. SI, p. 183–188 (online), DOI: 10.1145/30851.275627 (1986).