

デバイス内蔵カメラを用いた視線入力装置の実装と評価

2018SC010 月東 更紗 2018SC037 勝又 美紀

指導教員：石原 靖哲

1 はじめに

近年 COVID-19 が流行している影響により、オンライン会議やオンライン面会などの機会が増えている。それらの機会が増えているのは重度障がい者にとっても例外ではない。オンライン会議やオンライン面会は、在宅ワークなどオンライン上でのやり取りが色々なところで取り入れられ定着し始めているため、COVID-19 が収束したとしてもなくなると考えられる。オンライン会議などにおいては、ものの操作が必要になる。そういった際にホワイトボード機能などを用い、自分の考えていることを可視化して第三者に伝える必要がある。そのため、重度障がい者もオンライン会議などで容易に意思伝達できるような機材やシステムが必要である。

現段階において肢体不自由者や ALS をはじめとする重度障がい者向けに視線移動を利用して意思を伝達する機材が流通している。しかし、そのような機材は高価であるため容易に手に入れることが出来ない。また、機材の導入の検討をする人や利用環境を整えることが出来る人が身近にいないと導入が難しいなどの問題が存在する。肢体不自由者向けの視線入力装置のさらなる普及のためには、高価な機材がなくてもより簡易な方法で視線を検出する必要があると考えられる。そこで本研究では、デバイス内蔵カメラを用いて縦方向の視線検出が行える方法を提案し、デバイスを操作することが困難な人向けに視線でイラストを描ける視線入力装置の実装を目的とする。

同じ目的の先行研究として鳥居の研究である文献 [1] が存在する。しかし、鳥居が提案した視線検出の方法では縦方向の視線検出が難しいということが挙げられている。文献 [1] が指摘する技術的困難さに対し、本研究では視線入力装置の縦方向の視線検出として、効き目が多いと言われている右目において「ランドマーク検出により求める瞼の開き具合」と「眉毛と上瞼のランドマーク検出により求める瞼の開き具合」の異なる 2 種類の検出方法を提案し、実験により評価する。実験結果より、「ランドマーク検出により求める瞼の開き具合」で縦方向の視線を検出することが 1 つの検出方法の候補と成り得ることが確認できた。

2 関連研究

肢体不自由者や ALS、高齢者向けのコミュニケーションツールの研究と開発を行っている研究として鳥居の研究 [1] が挙げられる。彼の開発したコミュニケーションツールは、デバイス内蔵カメラから画像を取得している。また、視線の検出方法は、OpenCV の Haar-like 分類器を用いることにより常に眼球の中心を求め、黒目の部分を中心点から 4 分割して、それぞれの黒目の見える範囲の変化量を比

較することで視線方向を検出している。この研究において鳥居の提案する視線検出法では、デバイスに内蔵されている 1 つのカメラからの映像で左右の視線の動きは捉えることができるが、上下の視線の動きが捉えにくいことが問題として挙げられている。

3 提案する視線入力装置

提案する視線入力装置の使用する機材は、PC と PC に内蔵されているカメラを使用する。

提案する入力装置は、視線検出部と描画部から成る。視線検出部においては、視線検出の方法が異なる装置を 3 つ提案する。提案する装置全体の流れとしては、初めにデバイス内蔵カメラから撮影されたリアルタイムの動画が画面上に表示される。この画面では、周囲の光量や使用するデバイス内蔵カメラによってしきい値が異なるため予め設定する。しきい値は画面上部に表示されるトラックバーで値を変化させることができる。両目ともに目の内側に緑色の円が安定して表示されたときにスペースキーを押すことで次のキャリブレーションの画面に移動する。次のキャリブレーションの画面では、視線の座標と基準の値の紐づけを行う。ここまですべて視線検出部である。最後にポインタ操作や好きな色を選択して視線移動でイラストを描くことが出来る画面になる。入力時と操作時の切り替えや色の変更などは、指定したキーをタイプすることにより行うことが出来る。ここまですべて描画部である。最後の画面では視線検出部によってホモグラフィ計算された視線の位置が描画部で画面上に印される。

3.1 視線検出部

視線検出部の仕組みは、キャリブレーションに基づく方法で検出を行い、ユーザがどこを見ているか画面上の座標を返す。実際に見ている PC 画面上の座標とデバイス内蔵カメラで取得したユーザの顔画像から求めた値を紐づけることでそれを可能にする。顔の検出のために Haar-Cascade 特徴検出、瞳検出のために Blob 検出、瞳とカーソルの場所を検出するためにホモグラフィを使用する。

視線検出のために 3 つの方法でキャリブレーションを行う。1 つ目は目頭の座標と黒目の x 座標の差の値から推定する。2 つ目は「目頭の座標と黒目の x 座標の差の値」と「目のランドマーク検出により求められる瞼の開き具合の計算」から推定する。3 つ目は「目頭の座標と黒目の x 座標の差の値」と「眉毛と上瞼のランドマーク検出により求められる瞼の開き具合の計算」から推定する。

ランドマーク検出とは、顔の特徴的なポイントを検出することである。検出したポイントの座標から顔の位置や顔の特徴などを調べる。本研究におけるランドマーク検出で

は、目と眉毛のランドマーク検出を行うことにより、利用者が視線を上下に移動させる際にどれだけ瞼を開いているかの計算し、値を返す。ランドマーク検出には、文献 [2] を参考にし、Dlib で顔のランドマーク検出を行って座標を取得する。上記で述べた 2 つ目の目のランドマーク検出により求められる瞼の開き検出は、文献 [3] を参考にし、3 つ目の眉毛と上瞼のランドマーク検出により求められる瞼の開き検出には 2 つ目の計算式の上瞼として処理する式を眉毛に変更し、下瞼として処理する式を上瞼に変更したものである。

3.2 描画部

描画部では予め白いキャンバスとペンの種類や色の種類、白紙に戻す機能を選択できる画面が表示される。視線検出部で座標化された場所に指定したペンの種類、色でペイントしていく。

4 実装

横方向と縦方向の視線の検出において 3.1 節で述べた視線検出方法を用いて視線入力装置を 3 つ作成した。作成する際に EyeDraw [4] の視線検出部に手を加えることによって実装をした。瞼の開き具合の検出は、文献 [3] を参考にした。また、目の目頭の座標を取得するために顔のランドマーク検出を行えるようにした。このランドマーク検出は、文献 [2] の手法を参考にした。

使用した PC は、LEVEL-15FR105-i7-TAZX であり、内蔵カメラの解像度は、100 万画素であり、カメラの位置はパソコン上部の中心にある。環境は表 1 に示す。入力装置 B と入力装置 C のメイン処理は 147 行、GUI は 399 行、関数の定義を行うプログラムは 302 行、視線検出の処理は 95 行のプログラムから成る。

4.1 入力装置 A

入力装置 A は、3.1 節の 1 つ目の視線検出の方法で述べたように目頭の座標と黒目の座標の差の値から視線を推定する視線入力装置である。

入力装置 A を実装したときに PC 画面の最右側や最下側を見ている場合においてポインタがパソコンの中央辺りに表示されることがあり、それより右や下に動かすことが出来ないことが多くみられた。

表 1 使用した環境

使用環境	ver
Python	3.8.11
OpenCV	4.0.1
Dlib	19.22.0
imutils	0.5.4
scipy	1.7.1
NumPy	1.20.2
Anaconda3	4.10.3

4.2 入力装置 B

入力装置 B は、3.1 節の 2 つ目の視線検出の方法で述べたように「目頭と黒目の座標の x 軸の変化量」と「目のランドマーク検出により求められる瞼の開き具合の計算」に変更し、視線を推定する視線入力装置である。

入力装置 B を実装したときに PC 画面の最右側や最下側を見ている場合においてポインタも最も右側や最も下側に表示されることがあり、視線の検出がある程度できていることが確認できた。

4.3 入力装置 C

入力装置 C は、3.1 節の 3 つ目の視線検出の方法で述べたように「目頭と黒目の座標の x 軸の変化量」と「眉毛と上瞼のランドマーク検出により求められる瞼の開き具合の計算」に変更し、視線を推定する視線入力装置である。

入力装置 C を実装したときに入力装置 B とあまり変わらない動作が確認できた。

4.4 事前実験

我々は、視線入力装置で縦方向の視線をより良い精度で検出するために、3 つの入力装置のそれぞれを使用して一点を見つめたときに生じる値のゆらぎの量を比較する。視線入力装置で一点を見つめているときに検出される視線のゆらぎの量は少ないほど視線を安定した取得が可能であり、理想的である。我々はさらに、ゆらぎの量から予想される画面上のカーソル位置の誤差を、画面の高さに対する比として求め、その比較も行う。比較に用いたディスプレイのサイズは、15 インチである。3 つの入力装置は PC 画面の最左上から最左下をつなぐ直線を y 軸とし、PC 画面の最左上から最右上をつなぐ直線を x 軸とする。 y 軸を垂直に 6 つに区切った線と x 座標 1 ずつ x 座標に対して垂直に区切った線の交差点でキャリブレーションを行った。

以下、目頭と黒目の y 座標の差を α とする。入力装置 A で一点を見つめたときに 10 回検出した α の最大値と α の最小値、 α の最大値と α の最小値の差、それらから予想される、画面上での縦方向の誤差をまとめたものを表 2 に示す。入力装置 A は、表 2 より、一点を見つめたときに最大値と最小値の差が 3 ある。予め、入力装置 A を用いて PC 画面の最上から最下を見たときに検出される縦方向を求めるための値の差が 6 であるため PC の画面の高さを 6 とする。従い、PC 画面の高さに対して 50 % の大きさの誤差がゆらぎとして表示されるカーソルに生じることが予想される。

以下、ランドマーク検出により求める瞼の開き具合を β とする。入力装置 B で一点を見つめたときに 10 回検出した β の最大値と β の最小値、 β の最大値と β の最小値の差、それらから予想される、画面上での縦方向の誤差をまとめたものを表 3 に示す。入力装置 B は、表 3 より、一点を見つめたときに最大値と最小値の差が 0.06 ある。予め、入力装置 B を用いて PC 画面の最上から最下を見たとき

に検出される縦方向を求めるための値の差が 0.19 であるため PC の画面の高さを 0.06 とする。従い、PC 画面の高さに対して 38 % の大きさの誤差がゆらぎとして表示されるカーソルに生じることが予想される。

以下、ランドマーク検出により求める眉毛と上脛の開き具合を γ とする。入力装置 C で一点を見つめたときに 10 回検出した γ の最大値と γ の最小値、 γ の最大値と γ の最小値の差、それらから予想される、画面上での縦方向の誤差をまとめたものを表 4 に示す。

入力装置 C は、表 4 より、一点を見つめたときに最大値と最小値の差が 0.282 である。予め、入力装置 C を用いて PC 画面の最上から最下を見たときに検出される値の差が 0.6 であるため、PC 画面の高さに対して 47 % の大きさの誤差がゆらぎとして表示されるカーソルに生じることが予想される。

この比較によって視線を安定して取得できるのは縦方向の視線検出の際に黒目を追うのではなく、脛の開き具合を検出することでより正確になることが確認できた。より視線検出として表示されるカーソルに生じるゆらぎが小さいと考えられる入力装置 B と入力装置 C を用いて実験することで、視線入力装置が縦方向の視線の動きを捉えることが出来ているか確認する。

5 実験方法

実験は、入力装置 B と入力装置 C を利用して行う。実験は 4 人を対象として行い、以下実験に参加してくれる被験者を被験者イ、ロ、ハ、ニとする。

1 つ目の実験としては、予め被験者イ、ロ、ハ、ニにペンタブレットを使用して入力装置 B で描く予定の絵を各々

に手書きしてもらい、その後に被験者に 1 人ずつ PC の画面の前に座ってもらい、このときの PC 画面と机のなす角は 90 度とし、画面の最上部と被験者の目の位置は同じ高さとする。また、被験者イ、ロ、ハ、ニの顔と PC の画面との距離は、45cm となるようにする。この状態で被験者イ、ロ、ハ、ニに入力装置 B を使い、各々がペンタブレットで手書きした絵と同様の内容のものを視線で描いてもらう。これを続けて 3 回ずつ行ってもらい、視線で入力した人以外には、視線で入力されたものを見て何が描かれているのかを予想してもらい、この実験により入力装置 B における入力の精度を確認する。精度の確認としてほかの被験者が描いたイラストを見て何が描かれているのか予測し、どれだけ正解を当てられるのかによって確認する。

2 つ目の実験としては、1 つ目の実験と同様の内容を入力装置 C を用いて行う。

図 1 に実験様子のイメージ図を示す。

6 実験結果

1 つ目の実験である、座った状態で入力装置 B を使用し視線入力をしてもらい、予め手書きした任意のイラストと比較する実験を被験者 4 人に対して手順通りに各々 3 回実行したときの結果を表 5 に示す。表には横にイラストを見て何が書かれているのか予想する予想者、縦に誰が書いたイラストなのか示してあり、正解した場合は○、不正解だった場合を×を記載した。

被験者 4 人に 3 回ずつ実行してもらった結果、視線で入力したイラストから予め手書きした任意のイラストを 7 回特定することが出来た。実際に被験者が入力装置 B で入力したイラストを図 2 に示す。黒い線が入力装置 B で入

表 2 黒目と目頭の y 座標の差

PC 画面の高さ 6	α の最大値	α の最小値	α の最大値と最小値の差	予想される縦方向の誤差
1 回目	-2	-5	3	50 %
2 回目	-3	-6	3	50 %

表 3 目のランドマーク検出により求める脛の開き具合

PC 画面の高さ 0.16	β の最大値	β の最小値	β の最大値と最小値の差	予想される縦方向の誤差
1 回目	0.378	0.327	0.051	32 %
2 回目	0.398	0.338	0.06	38 %

表 4 眉毛と上脛のランドマーク検出により求める脛の開き具合

PC 画面の高さ 0.6	γ の最大値	γ の最小値	γ の最大値と最小値の差	予想される縦方向の誤差
1 回目	0.866	0.584	0.282	47 %
2 回目	0.8	0.61	0.19	32 %

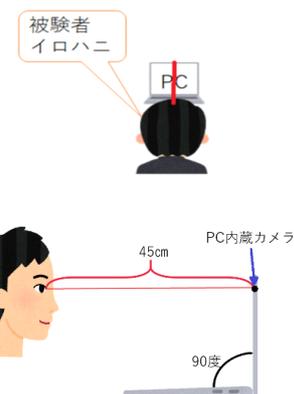


図 1 実験のイメージ図

表 5 実験 1 結果

予想者	イ	ロ	ハ	ニ
イのイラスト		○	○	○
ロのイラスト	○		○	○
ハのイラスト	×	×		×
ニのイラスト	×	○	×	

力した線で赤い線が実際に入力したかった内容である。

このことから座った状態における入力装置 B の精度は 7/12 であるといえる。

2つ目の実験である、座った状態で入力装置 C を使用し視線入力をしてもらい、予め手書きした任意のイラストと比較する実験を被験者 4 人に対して手順通りに各々 3 回実行したときの結果を表 5 に示す表には横にイラストを見て何が書かれているのか予想する予想者、縦に誰が書いたイラストなのか示してあり、正解した場合は○、不正解だった場合を×を記載した。

被験者 4 人に 3 回ずつ実行してもらった結果、視線で入力したイラストから予め手書きした任意のイラストを 4 回特定することが出来た。実際に被験者が入力装置 C で入力したイラストを図 3 に示す。黒い線が入力装置 C で入力した線で赤い線が実際に入力したかった内容である。

このことから座った状態における入力装置 C の精度は 4/12 であるといえる。図 2 と図 3 を比較すると、入力装置 B のイラストのほうがより正確な縦方向のペイントが行えていることが分かる。図 2 と図 3 を描いた被験者からは「入力装置 B で入力したほうが縦方向の動きがよく反映された。」「入力装置 C は縦方向に動かしたペンがあまり動かしにくかった。」との使用感のコメントを受けた。実験 1, 2 の実験結果と被験者にインタビューし、得た使用感を比較した結果入力装置 B の方が縦方向の視線移動に合わせイラストを描くことが出来る割合が高いと言える。そ



図 2 入力装置 B で入力したイラスト

表 6 実験 2 結果

予想者	イ	ロ	ハ	ニ
イのイラスト		○	○	○
ロのイラスト	○		×	×
ハのイラスト	×	×		×
ニのイラスト	×	×	×	

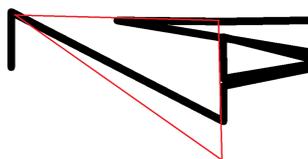


図 3 入力装置 C で入力したイラスト

のため、描きたかったイラストの特定にもつながったと考える。

7 まとめ

本研究では、デバイス内蔵カメラを用いて縦方向の視線検出が行える方法を提案し、デバイスを操作することが困難な人向けに視線でイラストを描ける視線入力装置の実装を目的とした。

実験結果より入力装置 B の方が縦方向の視線検出に合わせイラストを描きやすいといえる。このことより縦方向の視線検出において黒目を追跡する以外の方法として「顔のランドマーク検出により算出する顔の開き具合で視線検出を行う」が有効性があるといえる。これは縦方向に視線を動かしたときの目の様子を観察すると上瞼以外にも下瞼に変化がある。下を見たときに上瞼は小さく閉じて、下瞼は大きく開く。しかし、入力装置 C は眉毛と上瞼のみで開き具合を算出している。これは縦方向に視線を動かしたとき下瞼にも変化があるが算出時に考慮されていない点が正確性を低下させる要因と考えられる。

今後の課題は 2 点ある。課題の 1 つ目は、ゆらぎが生じたり、入力時に線が重複して書かれるなどの視線入力装置の精度が低いことである。1 つ目の原因はキャリブレーションで紐づけされた座標の値の変化が小さいことだと考えられる。もっと値に変化があるようなキャリブレーションの仕方を検討するべきである。2 つ目の原因は、黒目やランドマーク検出の方法の可能性が考えられる。黒目やランドマーク検出するとしても別の検出方法の検討をするべきである。課題の 2 つ目は、操作モードと入力モードの切り替えなどが現時点では、任意のキーをタイプすることにより切り替え可能だが、より重度障がい者のユーザビリティや負担の軽減などを考慮し、瞬きや注視などの動作で切り替え可能にすることが課題となる。

参考文献

- [1] 鳥居一平. 肢体不自由児や ALS, 高齢者のための視線方向によるコミュニケーションツールの研究と開発. 第 8 回 児童教育実践についての研究助成<継続助成>研究成果報告書 (要約), 2006.
- [2] Atom. OpenCV、Dlib で顔のランドマーク検出とまぶたを閉じる検出をやってみた. <https://iatom.hatenablog.com/entry/2020/11/01/152334>. (2021-09-09 参照).
- [3] ウルシステムズ株式会社. 眠気を判定! 目のまばたき検知を Dlib と OpenCV を組み合わせて数十行で作る. <https://qiita.com/mogamin/items/a65e2eaa4b27aa0a1c23>.
- [4] fabridigua. Eye Draw — コンピュータービジョンを使用して目で描く方法. <https://ichi.pro/eyedraw-konpyu-ta-bijon-o-shiyoshite-me-de-egaku-hoho-37808988233322>. (2021-05-26 参照).