

単眼カメラを用いた簡易な視線方向推定

Simple gaze direction estimation using monocular cameras

山下 和也[†] 平川 正人^{††}

Kazuya Yamashita[†] Masahito Hirakawa^{††}

[†] 島根大学自然科学研究科 ^{††} 島根大学学術研究院理工学系

1 はじめに

近年,AR や VR などの新しいインタフェース技術の盛り上がりと共に, 入力手段としてマルチモーダルなインタフェースが注目されている. AR や VR においては, 既存のキーボードとマウスを用いた入力に縛られない音声認識を用いた入力などが考えられている. ハンドフリーな入力方式として他にも視線追跡を用いたものが挙げられるが, 現状では据え置きハードウェアが流通してはいるが, 移動や柔軟な顔の動きに耐えうるシステムは多くない. また, AR や VR に実装されている視線追跡のシステムは専用の HMD(ヘッドマウントディスプレイ) に組み込まれている. このように現状の視線追跡デバイスは計測可能な距離や角度の制限が大きく, 一般への普及が進んでいない. 今後距離や角度といった制限が緩和できれば, 視線追跡を適応可能な領域が増え, より便利な生活の実現が可能である.

専用の視線追跡デバイスを用いることなく, 単眼のカメラから視線追跡を試みる先行研究もあるが [1][2], より簡易で汎用的な視線の追跡を行うことが期待されている. 筆者らも, 単眼カメラを前提とし, 加えて広範囲に対象者が移動しても顔方向を追跡, 推定するシステムの開発を進めてきている [3]. 目標とするシステムでは図 1 のように複数のカメラで対象者の顔方向と視線方向を捉えることを可能にする. これまでは視線推定が十分ではなかったため, 本研究では PoseNet[4] を用いた実装と, さらに視線を追跡可能にするシステムについて提案する.

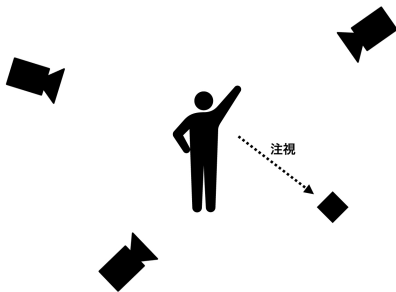


図 1: システムのカメラと対象者の関係イメージ

2 研究内容

2.1 提案

本提案では, 虹彩中心と目周辺の特徴点を結ぶ線分の方向と距離からおおよその視線方向を算出が可能という仮説を立て, 実証のための実装を行う. そのため虹彩中心の推定と目領域周辺の複数の特徴点を求める必要がある. 目領域周辺の特徴点の算出には筆者の前段階の研究で用いたものと同様に, Tensorflow の JavaScript 実装である TensorflowJS を用いる. 特徴点の推定には, 当初は OpenPose[5] を試したがパフォーマンスの問題や本システムが採用する複数台のカメラの切り替えのような実装との連携の難易度が高かったため, より柔軟な Tensorflow の JavaScript 実装上で動作する PoseNet モデルを用いることにした. 虹彩中心の推定は TensorflowJS の推定結果では求める精度になく, 実用には難しかったため, カスケード分類器を用いた画像処理で虹彩中心の座標を求めることにする.

2.2 システムの利点

視線方向の推定にあたっては顔前のディスプレイモニタ周辺にデバイスを設置するタイプのものや, メガネ型のハードウェアを用いたシステムも多くある. 本研究では利用者の周囲にウェブカメラを複数台設置する. このような構成を探る利点は, まずウェブカメラは一般に多く普及しているため入手と設置が容易であることがあげられる. 対象者の周囲にある複数台のカメラから推定を行うため, 対象者が特定の場所に留まる必要はなく, 移動しても適切なカメラに切り替えることで柔軟に追跡が可能になる. さらにメガネ型のハードウェアのように対象者がデバイスを装着する必要がないため, 身体的負担が低くなる. またメガネ型の視線推定システムの場合, ユーザーがデバイスを装着するため, 取得した情報はユーザーに紐づくが, ここでの提案システムではユーザーの情報と紐付かないため, 匿名の情報としてデータを活用することが可能になる. そのため将来的にはユーザーの安心感を維持した上で視線の注視による興味を分析するマーケティングなどへの応用が想定される [6].

また本研究では JavaScript でシステムを構築しており, 特定のバックエンドに依存せず多様な環境で利用することが可能である. たとえば GPU を搭載しないコンピュータでも CPU で十分なフレームレートを保ちつつ推定が可能である. Raspberry Pi のような小型の

コンピュータでもウェブカメラさえ接続されていれば利用できるため、設置場所の依存も少ないことが利点にある。

2.3 システム構成

本提案システムを構成する主要な要素を下記に示す。

- ウェブカメラ (Logicoool C270)
- NVIDIA Geforce GTX1080
- Node.js 10.16.3
- TensorflowJS 0.12.4 (JavaScript 版 Tensorflow)
- PoseNet 0.2.2 (TensorflowJS で用いる特徴点推定モデル)

2.4 処理の流れ

最終的な実験環境では複数の単眼カメラを予め設置していることを想定する。但し今回は視線方向の推定にフォーカスするために、使用するウェブカメラ 1 台の正面に対象者が相対している状況での実験を行う。ウェブカメラは PC モニタのベゼル上部の中央に設置し、対象者との距離は 50~60cm とした。

本システムの処理の流れとしては、まずはじめにウェブカメラから入力となる画像を取得し、顔の特徴点の推定を行う。推定結果が有効なスコアの場合、つまり顔の存在を推定した場合は目領域周辺の画像を切り抜き、虹彩中心の座標を求める。得られた虹彩中心の座標をもとの入力画像上での座標に変換し、表示する。最後に虹彩中心の座標と最も短い距離にある目の特徴点との線分を表示しておおよその視線方向を示す。この処理を終了まで繰り返す。

本システムを実行した時の表示画面は図 2 になる。赤い点が虹彩中心の推定結果の座標、白線は推定した目の特徴点の中で最も短い距離となる点に線分を引いており、これが視線方向に相当する。



図 2: システム実行時の様子 1

顔が大幅に横方向を向き、カメラの正面を捉えてない場合は図 3 のように虹彩中心を検出できず、視線方

向の算出が正しく行えない。このような推定結果になった場合は筆者らの前段階の開発 [3] で行った適切なカメラへの切り替えを適用することで、継続して推定できるようにする。



図 3: システム実行時の様子 2

なお, PoseNet は 17 点の特徴点を検出するが、今回は顔周辺の特徴点推定を拡張したモデルを用いる。このモデルでは顔領域について 68 点の推定結果を得られるが、本システムでは主に目領域周辺の情報を用いるため、図 4 のように左右の目それぞれにまぶた上で推定される各 6 点を使用する。

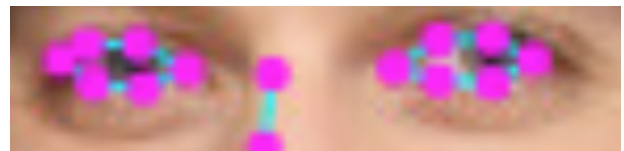


図 4: 取得する目領域周辺の特徴点

虹彩中心の推定については, TensorflowJS で目の特徴点を取得することができるが、これで取得した座標は目領域周辺の座標から外れることが多く、精度良く虹彩中心を求められていなかった。そのため比較的軽量な実装でかつ精度よく虹彩中心の座標を得ることができる関連研究 [7] の論文をベースに実装を行った。

2.5 実験結果

NVIDIA Geforce GTX1080 を搭載した PC 上でシステムを動作させた時には、おおよそ 60FPS を維持しつつ、リアルタイムに顔の特徴点の推定を行い、さらに虹彩中心の座標を求め、それらから視線方向の算出を行えていることが確認できた。

虹彩中心に関して、座標の推定は概ね精度良く求められているが、頭髮等で眉が隠れている場合やメガネのレンズで外光が強く反射しているような場合に推定がうまく行えず、精度が低下する場面が見受けられた。PoseNet による機械学習を用いた特徴点の推定とは

異なり画像処理で算出を行うため、まぶたの形状などを考慮した補正 [8] を行う必要がある。

PoseNet では身体全体の特徴点を推定するが、本システムでは入力となる映像に顔領域が写っていれば特にキャリブレーションは必要ない。しかし、本システムでは特徴点推定のために用いるモデルやカスケード分類に用いるデータの取得を実行の最初に行っているため、初回起動時には実際に動作するまで 1 秒ほどの待機時間がある。そのため将来考えている複数台のカメラを用いるような実稼働では、初期使用カメラの決定などの処理と合わせて簡易的なキャリブレーションを行うことを含めて、対応が必要である。

3 今後の展望

現状ではおおよその視線方向を求めることはできたが、具体的な角度の算出までが行えていない。そのため今後は虹彩中心と目領域周辺の特徴点を結ぶ線分の長さから角度の対応付けを行うなどして、視線方向の角度を算出する必要がある。さらにその数値に対して定量的な精度を評価する。今後の実験としては被験者を増やし、複数人からデータを取得する。また、前段階の実験で行った複数台のカメラでの動作にも対応させ、対象者が動いていても顔方向や視線方向を追跡できる柔軟なシステムを目指す。

虹彩中心の座標はかなり精度良くとれているが、目領域周辺の特徴点の数が少ないため、視線方向の算出が粗くなっている。今後より多くの目の周辺について特徴点を推定できるモデルへの切り替えや実装を検討する。

4 おわりに

本研究では、前段階で行った広範囲の視線の追跡を目指した広角度の顔方向推定を行うシステムの実用化に向けて、視線方向の追跡を行うシステムの開発を行った。現状ではある程度の視線方向は取得できるが十分な精度では角度を算出できていないため、更に検討が必要である。また複数台カメラでの適切なカメラへの切り替えなどとの連携も確認する必要がある。将来的には本システムで単眼のカメラから顔方向や視線方向を追跡できるようすることで、簡便でより柔軟かつ強力な入力手段となり得る。また今後は既存の類似システムとの比較を行い、精度評価や適応可能なアプリケーションの実装を目指す。

参考文献

- [1] 岡本 康太郎, 内海 章, 萩田 紀博, 高橋 和彦, "複数カメラを利用した広域遠隔視線計測", 情報処理学会インタラクシオン 2012, IEXB-23, 2012
- [2] 山添 大丈, 内海 章, 安部 伸治, "顔特徴点追跡による単眼視線推定", 映像情報メディア学会誌 61, 12, pp.1750-1755, 2007
- [3] 山下 和也, 平川 正人, "複数の単眼カメラを用いた広角度の顔方向推定", HISS2018 論文集, pp.108-110, 2018
- [4] I.Alvarado and A.Gallo, "Real-time Human Pose Estimation in the Browser with TensorFlow.js", <https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5>, 2018
- [5] Z.Cao, G.Hidalgo, T.Simon, S.Wei, Y.Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", <https://arxiv.org/abs/1611.08050>, 2016-2017 Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.7291-7299, 2017
- [6] 里村 卓也, "視線計測による消費者行動の理解", 画像ラボ, 画像ラボ編集委員会編, 29, pp.71-79, 2018
- [7] N,Markus, M.Frljak, I.S.Pandzic, J.Ahlberg, R.Forchheimer, "Eye pupil localization with an ensemble of randomized trees", Pattern Recognition, 47, 2, pp.578-587, 2014
- [8] 平野 公太, 杉原 真, 志堂寺 和則, "虹彩とまぶたの形状を考慮した Hought 円検出による虹彩位置推定手法の提案", 情報処理学会研究報告, 2015-SLDM-170, 52, 2015