

09 Web カメラによる視線トラッキングに関する研究

小松篤志

指導教員 ソソラ

1. はじめに

近年、世界的に非接触の操作デバイスの需要が拡大している。音声認識や体を使ったジェスチャーを画像認識するなど多くの手法がある。

そこで今回私は、眼の動きをトラッキングして何かを操作できるツールを製作したいと考えた。

2. 開発環境

OS	Windows 10
統合開発環境	Visual Studio Code
開発言語	Python 3.8.10
使用ライブラリ	OpenCV MediaPipe faceMesh Pandas Numpy Kivy Concurrent.futures winsound
カメラ	Integrated Camera(標準装備)

3. 研究概要

本研究では、特殊なハードウェア [1] を必要とせずに、WEBカメラを使用してリアルタイムで虹彩、瞳孔、および眼の輪郭を含むキーポイントにより、眼の動きを追跡するアルゴリズムを開発する。

3.1 眼の左右の判定について

提案したアルゴリズムは以下の通りである。

ステップ 1 : MediaPipe の FaceMesh を使用し、眼のキーポイントのうち図 3.1 に示す①～⑤ (①—左目頭、②—右目頭、③—左虹彩の中心、④—右虹彩の中心、⑤—目頭と目頭の中心) のキーポイントの座標を取得する。

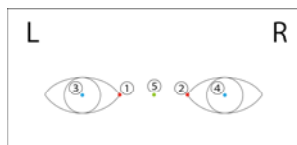


図 3.1 本手法に用いられる眼の

キーポイントの位置情報

ステップ 2 : ①～⑤のキーポイントの座標から異

なるキーポイント同士の距離を求める。ここで、 i 番目のキーポイントの座標 $P_i(x_i, y_i)$ 、 j 番目のキーポイントの座標を $P_j(x_j, y_j)$ とすると i, j の距離 D_{i-j} を式 3.1 により求める。そして、図 3.2 (ア) には前へ向いている初期状態、(イ) には眼球を右に動かす状態の距離をそれぞれ色別で示した。

$$D_{i-j} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (3.1)$$

ステップ 3: 頭を動かさない状態で、眼を右に動かす場合は D_{3-1} と D_{2-4} の値が変化される。

つまり、眼球を右に動かすと D_{3-1} (III) の値がある閾値より小さくなり、 D_{2-4} (IV) の値もある閾値より大きくなる。

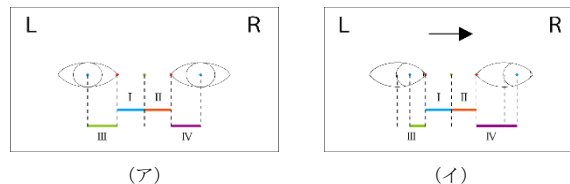


図 3.2 眼を右へ動かしたときの変化の様子

3.2 眼の上下判定について

上を向いた時のアルゴリズムと下を向いた時のアルゴリズムを分けて提案した。

3.2.1 上を向いた時のアルゴリズム

ステップ 1: 新たに図に示す眼のキーポイントの座標を取得する。



図 3.3 上向き判定のキーポイント

ステップ 2: 3.1 ステップ 2 同様にキーポイント同士の距離を取得する。

ステップ 3: 眼を上には動かすと D_{6-8} と D_{7-9} の距離が変化する.これを左右どちらの目でも判定する.
実験中の試行錯誤

左右の判定に比べ, 変化距離が非常に小さく, 誤差値と正常値で区別に難航した.

そこで指数平滑移動平均法 [2] を利用し, データの値を均すことで誤差値と正常値を区別しやすくした.

3.2.2 下を向いた時のアルゴリズム

ステップ 1: 図 3.1 に加えて新たなキーポイントの座標を取得する.

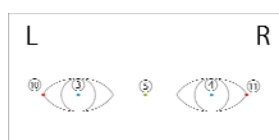


図 3.4 下向き判定のキーポイント

ステップ 2: 線 $6-8$ と点 4 の距離を計算(式)

反対の眼も対応する線と点で計算

ステップ 3: 左右それぞれの眼の面積を計算

ステップ 4: ステップ 2 で算出した距離が線 $6-8$ より下へ移動したことを判定する. 同時に, 眼の面積が閾値より低くなった時に判定結果を返す.

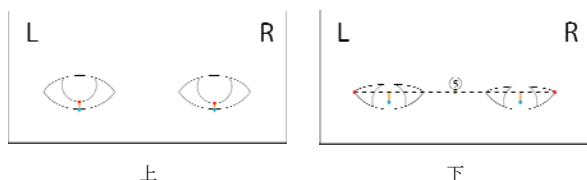


図 3.5 眼を上下へ動かしたときの変化の様子

3.3 視線方向取得を活用したアプリ

作成した視線方向取得ツールを用いて音楽を演奏することのできるデスクトップアプリケーションを作成した.



図 3.6 アプリメイン画面

3.3.1 アプリ仕様

- ・視線方向に合わせて音程が変わる
- ・画面右側が操作オブジェクトである
- ・画面右上の「BPM」欄で BPM(リズム)の変更が可能

各種ボタンの機能

ボタンの名前	機能
mute	演奏の一時停止
play	演奏再開
up, down	音程の切替
set (BPM 欄)	BPM を確定する
?	操作説明
終了	アプリの終了

4. 終わりに

本研究を通して, 眼の動きをトラッキングする専用ハードウェアがなくとも眼の動きを検知することが可能であると確認できた.視線を取得する方法として角膜反射法 [3] がある. しかしこの方法では眼に光を当てなければならず目への負担がどうしても避けられない, しかし本研究では, 外部からの人体への影響を極力減らして実行することが可能となった.しかし, 課題点も残った. 再初期段階で考えていた案は視線で PC のマウスのように画面上を操作できるようなツールだった. 本研究のような, 市販の web カメラから画像認識をして目の動きを取得する方法であると細かい目の動きを検知する精度が出せなかった. 妥協案として方向を取得するだけなら誤差と区別できる明確なデータが取得できると考えた.

今後の課題として左右の判定の精度は十分に作ることができたが, 上下の判定の精度は改善の余地が残されている.

参考文献

- [1] 特殊なハードウェア
<https://www.tobii.com/ja/products/eye-trackers>
- [2] 指数平滑移動平均法
<https://mirukognosis.com/?p=1020>
- [3] 角膜反射法
<https://techblog.nhn-techorus.com/kakumaku>