

顔追跡による教材注目情報の取得

田所 龍介[†] 納富 一宏[†]神奈川工科大学[†]

1. はじめに

ICT 活用がされている大学等の講義において、講義で使う資料は、PDF 等のファイル形式で提供されることが多い。教員は講義資料をプロジェクタで投影して解説を行う。このとき、学生側は教員が開設した内容をノート等に記録する。その際、学生が講義中に解説を聞き漏らすようなことがある場合や、疑問に思うことがある場合に、そのことを教員に質問をするというような光景を見ることはあまりない。その結果、講義が教員から学生への一方通行なものになりがちである。そのようなことから、教員側は講義における学生の授業の理解度を把握しづらい。

本研究では、講義での学生の理解度や反応を記録できるシステムを開発している。このシステムは、Web 上で授業資料となる PDF を閲覧できる。そして資料の上に直接付箋のようにメモを貼り付けることができる。また、疑問や理解などの感情を一目でわかるようなスタンプを資料の上に貼ることができるので、資料を見返したときにも、資料のどの箇所でもどのように思ったかをすぐに理解できる。本研究ではスタンプやメモなどの情報を注目情報としている。

新たな注目情報として使用者の視線情報を利用できないかと考えた。視線情報の取得には PC 内蔵のカメラや汎用的な Web カメラを用いて使用者の顔を追跡して情報を取得する。

本稿では動作実験を行い、視線情報を取得可能かどうかについて報告する。

2. 授業分析システム

本システムは、教員がアップロードした PDF ファイルを Web 上で閲覧できる。表示される資料の上に直接メモやスタンプ（注目情報）を図 1 に示すような形で残せるシステムである。直接資料に情報を残せるため、データを Web 上で容易に管理でき、ノートと資料を別々に持つ必要はない。Web 上でデータの管理ができるので、教員は学生の注目情報を見ることで理解度を把握できる。

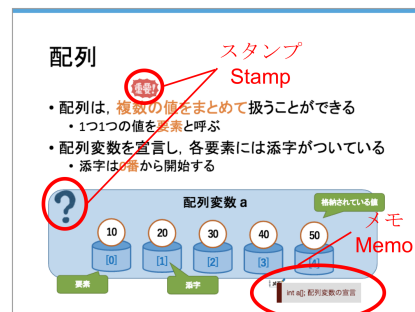


図 1 授業分析システム

スタンプ機能については、LINE や Twitter などの SNS において、コミュニケーションを行う際に画像を用いて行うことが多い。LINE では「スタンプ」と呼ばれるコミュニケーション用のイラストが存在する。テキストのみでは表現しづらい表現を、スタンプを用いることで容易に表現が可能であり、様々な場面で使い分けができる。また、スタンプのみで会話も可能なほどバリエーションが豊富であり、画像を送信するだけで相手に情報を伝えることができるため、採用した。

3. 視線情報取得

3.1 顔追跡

現在、PC 内蔵のカメラや市販されている安価な Web カメラから取得した動画情報から、顔の位置や姿勢をリアルタイムで取得することができる^[1]。そこで、視線の情報を取得できないかと考えた。市販されている視線追跡装置やオープンソースで提供されているもの^[2]は、専用の機器が必要であったり、カメラを目の至近に設置したりしなければならぬため、コストや手間がかかる。本機能では、赤外線機能を有したカメラの導入や動作環境を選ばず、目をカメラに近づける必要無く視線の取得を行う。

顔追跡には「clmtrackr」^[3]という JavaScript ライブラリを使用する。このライブラリでは、画像から顔位置の検出、顔部位（目・鼻・口）の検出を行う。その後、検出した部位の輪郭を抽出した後、その座標値(69点)を返す。

3.2 視線情報取得手法

使用者がどこに視線を向けているかという情報を取得するために、最初にキャリブレーション

ンを行う。キャリブレーションは図2に示すように、①使用者にはディスプレイの四隅に視線を向けてもらい、その時の瞳の座標値 $R(rx_{ij}, ry_{ij})$ を記録する。ここで得られた4点の座標値から表される矩形領域(図2:青色で塗られた領域)は、使用者がディスプレイを見る際の瞳の可動域となる。矩形領域が決定されると、②瞳の初期位置を設定する。初期位置はディスプレイの中心に使用者は視線を向けてもらい、その座標値 $M(mx, my)$ を記録する。キャリブレーションでは①、②を行う。点Mは使用者がディスプレイの中心に視線を向けているときの値であるので、矩形領域の中心点に点Mを設定する。現在、キャリブレーションを行う際には使用者は顔を動かすことができるが、キャリブレーションを終えた後は、カメラと顔の距離、振り向き動作などを行うと視線の取得の精度が落ちる。矩形領域をディスプレイと見立てることで、矩形領域内で検出される瞳の座標値 $EYE(eyex, eyey)$ とディスプレイの縦の大きさH、横の大きさWから、式1を用いることで座標値を実際のディスプレイに対応した座標 $E(ex, ey)$ を求める。

$$ex = \frac{W(eyex - rx_{11})}{(rx_{12} - rx_{11})} \dots\dots\dots (1)$$

$$ey = \frac{H(eyey - ry_{11})}{(ry_{12} - ry_{11})}$$

4. 動作検証

4.1 方法

3人の被験者に検証を行った。被験者の1人は眼鏡を装着していたが、装着・見装着は区別切に検証を行った。検証の流れについて述べる。Webカメラはディスプレイの上部に設置し、被験者の目の位置と平行になるように高さを設定した。検証用のWebページを開くと、図3に示すようにブラウザ上に順番に指示される10箇所(図3:青色の枠)の位置に顔を動かして視線を向けてもらう。その際、10箇所全ての位置へ移動させるまでの視線が辿った軌跡を取得する。表1に検証に使用した環境を示す。

4.2 結果・考察

被験者Aの軌跡情報を図4に示す。図4の結果から、下方向への視線情報の取得において精度が落ちているが、横方向と上方向においては視線の取得ができていたことがわかった。赤外線機能を有したカメラや目の至近にカメラを設置する必要なく、視線の取得が行えることが確認できた。



図2 瞳の可動域

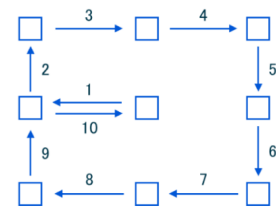


図3 視線移動順番

表1 動作検証環境

名称	バージョン等
Webブラウザ	Google Chrome 55.0.2883.87m
画面サイズ	2064×1080
Webカメラ	Logicool HD Pro Webcam C920

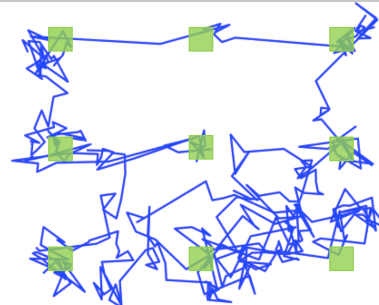


図4 被験者Aの軌跡情報

しかし、視線をディスプレイの外側に向けるにつれて、視線の取得精度が落ちることが検証からわかった。どの位置においても視線取得の精度が高くなるように今後改良する必要がある。また、現在はキャリブレーションを行った後は、顔を動かすと精度が落ちてしまう問題がある。講義などの場面において利用するならば、顔を動かしても視線の取得を可能にする必要があるため、今後改良していく。

5. まとめ

本稿では、授業分析システムの注目情報として新たに視線を利用できないかという目的のもと、視線追跡を汎用的なWebカメラで行う機能を試作した。動作検証から、視線追跡のある程度の精度は確認できたが、さらに精度を高める必要がある。

参考文献

[1]Shaoqing Ren, Xudong Cao, Yichen Wei, Jian Sun: Face Alignment at 3000 FPS via Regressing Local Binary Features, Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1685-1692, 2014.
 [2]Animouse, Animouse, <http://www.animouse.org/> [Accessed 28 December 2016].
 [3]auduno, clmtrackr, <https://github.com/auduno/clmtrackr> [Accessed 10 December 2016].