H-014

可視光カメラを用いた視線検出技法についての一考察

A Consideration on Eye Tracking Technique Using Visible Light Camera

鈴木 拓馬　　松井 章典
Takuma Suzuki　Akinori Matsui

埼玉工業大学　大学院　工学研究科
Department of Electronic and Engineering,
Graduate School of Engineering,　Saitama Institute of Technology

要旨
視線検出では赤外線を用いた手法が多く用いられている．しかし高精度な反面，外部装置が高価で入手が困難であるなど問題点もある．そこで我々は安価で入手が容易である可視光カメラに着目し，画像処理技術による視線検出の手法を提案する．

１．緒言

近年，視線検出技術についての研究が盛んにおこなわれている[1][2]．現在製品化されている視線検出用装置の多くは赤外光照射装置と赤外線カメラを組み合わせた角膜反射法を用いた装置[3]だが，高精度な反面高価であったり設置場所に限りがあったりする事が問題として挙げられる．そこで，本研究では web カメラやスマートフォンのカメラなどの可視光カメラに着目し，市販の web カメラと画像処理技術による視線検出法を提案する．

２．本研究で提案する手法

本研究では画像処理機能を持つ OpenCV を利用して視線検出のための黒目部分の検出を行った．図 1 に本提案手法のフローチャートを示す．ここでは，

①被験者の正面に web カメラを設置し，顔が撮影像内に入るよう撮影を開始する
②映像から 1 フレームを画像として取り出す．
③Haarcascade により取り出した画像から顔を検出する，検出した範囲から更に右目のみを検出して切り出す．
④切り出した画像に対して黒目の検出を行うが，前準備としてノイズ除去を施す．
⑤画像に対して Hough 円検出を行う．
⑥結果を円で描画し，②で取得した画像に描画する．
⑦画像をウィンドウに表示する．
⑧②に戻り処理を繰り返すの手順で処理を行う．

Hough 変換にて得られた円を確認して．黒目を検出できているものとする．以下に各部分について述べる．

2-1　haarcascade[4]

本研究では顔を検出する「haarcascade_frontalcatface」と目を検出する「haarcascade_eye」を使用した．まずカメラから取得した画像より，被験者の顔を検出する．その後検出した顔の範囲から右目のみを検出して切り出す．

2-2　画像処理

切り出した目の画像はノイズが多くそのままでは後述の Hough 円検出に支障をきたすため，画像処理を施してノイズを除去することで検出しやすいようにする．本研究では画像のグレースケール化及び画像の二値化と平滑化によりノイズ除去を行う．

2-2-1 グレースケール化

RGB(Red，Green，Blue)画像をグレースケール画像に変換するため，画素ごとに式(1)を適用する．

$$Gray=Red\times0.30+Green\times0.59+Blue\times03.11 \quad (1)$$

2-2-2　二値化

二値化をすることにより画像中の黒の部分のみを抽出し，黒の範囲でのみ後述の Hough 円検出が行えるように

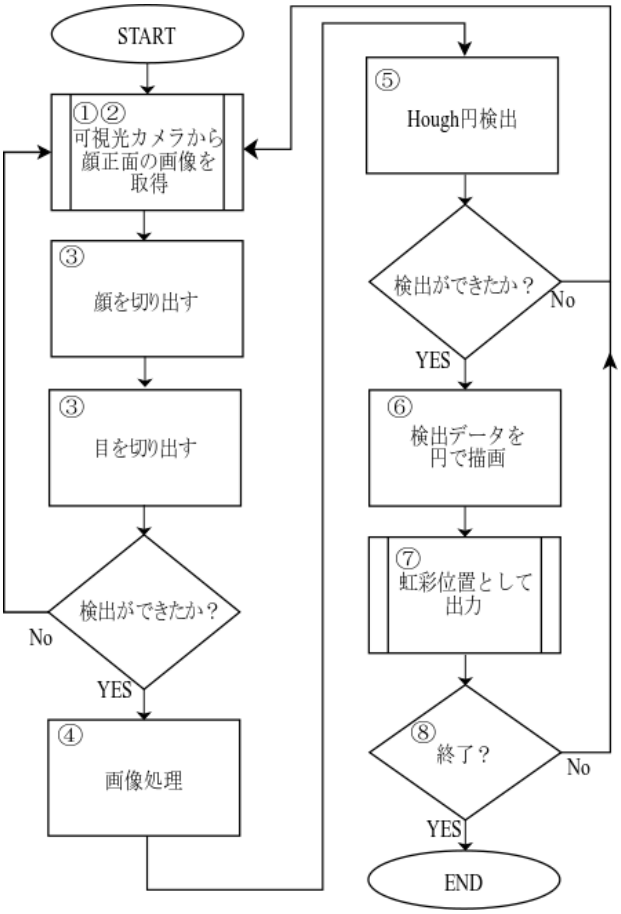

図 1　提案する手法のフローチャート

する．閾値は 80 に設定した．

2-2-3　画像の平滑化(ぼかし)

画像の高周波成分を消して画像全体をぼかす．本研究においては黒目に光源が反射した白飛びを無くしたり黒目のふちの色を揃えるのに利用する．ここでは正規化された箱型フィルタにより，カーネルの範囲内のある全画素の画素値の平均をとる手法を選択した．カーネルの縦幅と横幅を指定する必要があるため本研究では式(2)の 5×5 の箱型フィルタを用いた．

$$K=\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix} \quad (2)$$

2-3　Hough 円検出

直線や円を検出する Hough 変換のうち，本研究では円を検出する処理を使用する．図２に示すように，直交座標上の点（x，y）を通るすべての円は，円の中心点(p，q)と半径 r で表される． p と q を変化させながら式(3)の関係に基づき r を求める．

$$(x-p)^2+(y-q)^2=r^2 \qquad (3)$$

円検出のための Hough 変換は，画像上すべての点(x，y)をスキャンしていき，目的の円を決定する．本研究ではこの機構を用いて黒目の部分を検出している．

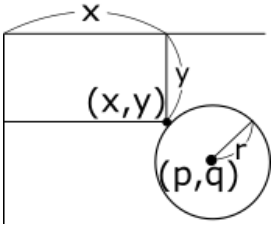

図 2　Hough 円検出

３．実験内容

前項で述べた手法を用いて可視光カメラのリアルタイム映像に対して黒目の検出を行う．最終的な動作状況を想定し，モニター上部に設置した可視光カメラから，正面に着座した被験者の顔を撮影した．カメラから被験者の距離は 70cm 程である．前項 2-2 において Hough 変換を行う準備としてノイズ除去のために画像処理を行うことを示したが，通常 Hough 円検出ではグレースケール化を施した後に平滑化された画像を用いて検出を行う．しかしながら Hough 円検出はノイズの量によって誤検知が増える問題がある，本研究では二値化した画像であれば黒目以外の場所について検出されないと考え，グレースケール化を行った後の画像処理において，平滑化を用いたものを実験１，二値化を用いたものを実験２する．実験環境を表 1 に示す．

PC 仕様		使用 web ｶﾒﾗ	
OS	Windows10 home 64bit	ﾒｰｶｰ	LOGICOOL
CPU	intel Pentium @3. 5GHz×2	型式番号	C270
RAM	16GB	最大解像度	1280×720
実行環境		有効画素数	120
言語	python	ﾌﾚｰﾑﾚｰﾄ	30fps
Python	Ver 3. 5. 4		
OpenCV	Ver 3. 1. 0		
ｺﾝﾊﾟｲﾗ	Anaconda		

表 1　実行環境

４．実験結果と考察

実験１及び実験２の過程と結果で得られた画像を以下に示す (図 3)．それぞれ，(1)RGB 元画像，(2)グレースケール画像，(3)実験１の過程で平滑化した画像，(4)実験１の結果画像，(5)実験２の過程で二値化した画像，(6)実験２の結果である．実験 1(図 3-4)と実験 2(図 3-6)ともに黒目部分が検出できているのがわかる．だが実験１では黒目が瞼に隠れている部分についても推計できているのに対して，実験２は瞼の内側の範囲でしか検出ができていないため，検出した円の大きさに差が出た．これは二値化により睫毛がかかっている黒目上部の境界が失われたためだと考えられる．よって平滑化を施した方がより正

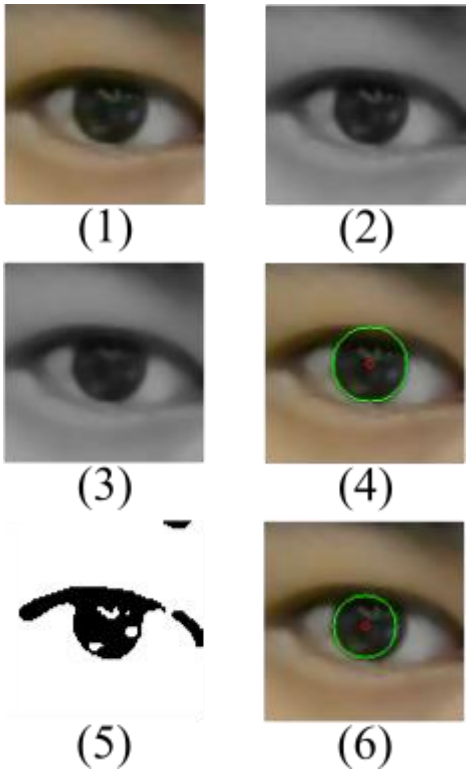

図 3　実験結果

確な結果になった．実際の動作中では，実験１では黒目自体の検出率は高いがこのような誤検知が非常に多くなった．対して実験２では二値化によって黒目周辺以外の対して実験２では二値化によって黒目以外の場所にて誤検知が起こることはなかった．これらの結果より，実験１の平滑化を用いた検出では黒目の検出はほぼ正確に行えて検出頻度も高いが誤検知が多いことも分かった．また，実験２の二値化画像では黒目の検出の正確性と検出頻度が低いものの誤検知がほぼないことが分かった．以上のことから，検出において正確性の高い平滑化画像にて検出を行いながら，誤検知の少ない二値化画像を用いて黒目以外の範囲での検出を防ぐなどで最適化を図ることが出来ると考える．

５．結言

提案する手法を用いてリアルタイムの映像での黒目部分の検出を行った．また，提案する手法のうち，Hough 円検出の前準備に施す画像処理によって検知範囲や誤検知率を操作できることが分かった．今後は誤検知率や処理速度を計測し，最適化をする必要があると考える．また，視線検知には黒目の位置以外に基準となる点が必要であるため，その位置をどう決定するかも検討課題である．

参考文献

[1]伊藤和幸，数藤康雄，伊福部達，”重度四肢不自由者向けの視線入力コミュニケーション装置”，電子情報通信学会論文誌，D-I，Vol. J83-D-I，No. 5，pp495-503，May 2000

[2]Tobii pro，”事例”，https://www.tobiipro.com/ja/fields-of-use/，June2017

[3]Tobii pro，”アイトラッキングとは”，https://www.tobiipro.com/ja/whats_eyetracking/，June 2017

[4]OpoenCV，” Face Detection using Haar Cascades”，https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection，June 2017