G-07

組込み Linux を使用した視線認識

The line-of-sight recognition on embedded linux

難波田 ゆう子† Yuko Naniwada　中野秀男†† Hideo Nakano　大西克実† Katsumi Onishi　渡部 健二‡ Kenji Watanabe　岡田 繁‡ Shigeru Okada

1. はじめに

本研究は、カメラの正面にいる人の視線を認識することを目的とする。一般的なパーソナルコンピュータと USB カメラを使用してシステムを構築した上で、シングルボードコンピュータである RaspberryPi と専用のカメラモジュールを使用した環境においても視線認識の精度検証を行い、精度の向上と 2 つの環境での結果を比較する。

2．研究背景

肢体不自由者を対象とした視線入力装置は、カメラ(赤外線を含む)の高性能化、低価格化によっていくつかのメーカーで開発されるようになってきている。しかし、システムとしては依然として高価であることや、思うような動作が得られないため、大きな普及には至っていない。

これまでに視線に関する研究は多数行われており、様々な取り組みがなされている。特に赤外線カメラを使用した視線認識については精度向上が進んでいる。The Eye Tribe Tracker[8]開発者向けの視線検知装置なども発表されている。

●視線認識システム

	マイトビー 画面上の文字を見て会話をしたりインターネットやメールを通して遠くの人とのコミュニケーションが可能になる重度障害者用の視線による意思伝達装置 価格：1,390,000 円

●視線検知システム

	The Eye Tribe Tracker 赤外線照合システムと結合した全く新しい高解像度センサーによりユーザーの視線を検知。接続したパソコンやタブレットの操作が視線だけでできるようになる装置の開発者向けモデル。45cm から 75cm 離れた位置から可能で、本体から 65cm の位置では 40cm 四方のエリアを検知可能 価格：99 ドル

本研究では、赤外線カメラを使用せずに一般的なパソコンと USB カメラを使用して視線入力システムを開発することで、より安価な構成を提案する。

一般的なパソコンと USB カメラを使用して視線入力システムを開発した際には、視線認識の精度がシステム全体の動作に大きく影響するため、本研究では視線認識の向上を目的とし、さらに、名刺サイズ(85.60mm × 53.98mm) のシングルボードコンピュータである RaspberryPi での動作・評価を行う。評価に RaspberryPi を使用することは、小型で汎用性の高い環境での動作の確認を目的としている。

3. 現状における問題点

3.1　視線認識で重要な瞳孔の検出

赤外線カメラを使用した視線認識の場合は、赤外線カメラでの撮影時に生じる角膜反射の位置を基準点として瞳孔の検出を行う。[図 1]しかし、本研究では赤外線カメラを使用しないために角膜反射に相当する基準点が必要になるので、目頭を基準点にする。[図 2]

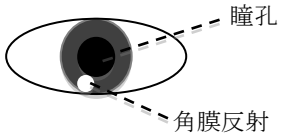

図 1. 赤外線カメラを使用した例

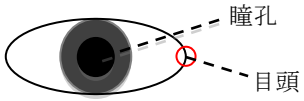

図 2.本研究での検出する例

3.2　自然光における光と影の影響

視線認識を実現する場合、時間や天候による影響が強く、室内であっても窓から差す太陽光の影響が大きく出ることがある。例えば晴天の日中に被験者の横に窓がある場合には、窓側の顔半分が白飛びしてしまい画像処理をうまくできない事が多い。そこで、本研究では大阪市立大学梅田サテライトの教室での室内照明を利用する環境を前提とし、太陽光の影響を受けない状態で実験する。

3.3　キャリブレーション

現在、商品化、研究されている視線に関するシステムの多くはキャリブレーションが必要なものになっている。この場合の、キャリブレーションとは、例えば視線入力システムではコンピュータに接続しているモニタの四隅、または八方向の隅を注視している状態を記憶し、ユーザ

†大阪市立大学大学院，Osaka City University
††帝塚山学院大学，Tezukayama Gakuin University
‡ 電脳・匠工房，Dennou Takumicobo

個別の視線認識の精度を上げるために使用する。しかし、顔の位置にずれが生じると、視線認識にも大きく影響してくるデメリットもある。本研究では、可能な限り簡単で短時間で完了するキャリブレーションを目指し、正面を向いた顔写真を一枚撮影する。また、その写真から被験者の顔の大きさを取得して、カメラとの距離を算出する際にも使用する。

4. 関連研究

4.1 先行研究事例

阿部氏らの研究[2]では、重度肢体不自由患者の頭部移動が小さいことに着目し、室内照明のもとに、視線検出を行う方法が提案されている。提案されたシステムでは、ユーザごとに上中下それぞれの位置の注視画像を記憶し、それらから光強度分布を求め基準データとしてユーザごとに辞書化することで、光強度の誤差を抑えている。

5. 研究方法

5.1 システム構成

本研究における視線認識システムの構成を図 3、図 4 に示す。 開発、試験には一般的なパソコン(Intel Corei3)に DebianLinux をインストールする。モニタの辺には市販の USB カメラが装着されており、被験者を撮影する。最終的に現在安価(35$.2014.07)で手に入る OpenSourceHardware の RaspberryPi(ARM)上に専用のカメラモジュール(25$、2014.07)を取り付けて精度評価を行う。RaspberryPi では Raspbian(Debian Linux の RaspberryPi 用)を使用することで DebianLinux で作成したシステムを移植して精度評価を実施できる。RaspberryPi ではモニタを接続しないため LED で視線認識した場合に点灯させることで確認する。

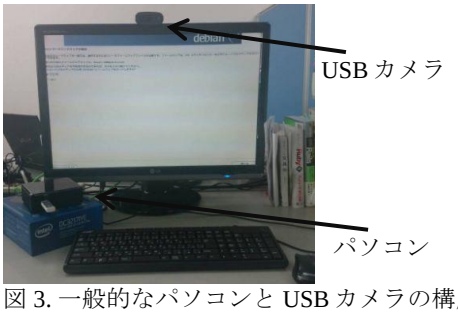

図 3. 一般的なパソコンと USB カメラの構成

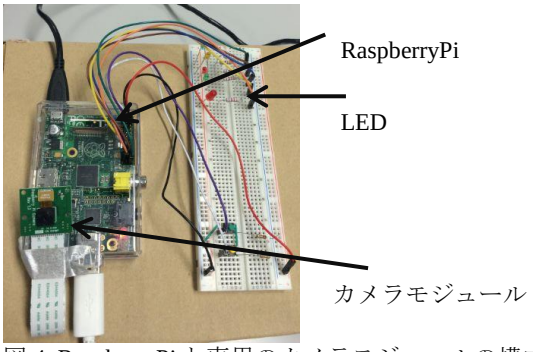

図 4. RaspberryPi と専用のカメラモジュールの構成

5-1 サンプル画像収集

5-1-1 サンプル画像収集の条件

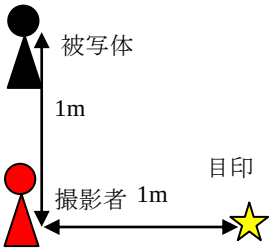

図 5. サンプル画像収集

(1) 真正面に座りカメラに視線を向けて撮影

(2) (1)の状態から視線だけを目印へ移して撮影

(3) 座る位置は変えずに目印が真正面になるように顔を動かし目印に視線がある状態で撮影

(4) (3)の状態から視線だけをカメラに戻して撮影

5-1-2 サンプル画像の収集数

5-1-1 の条件で 4 パターン 100 名から収集する

収集したサンプル画像は被験者とカメラの距離を算出と視線方向の認識に使用する。

5-2 瞳孔と基準点の検出

5-2-1 画像解析による視線計測手法

視線方向の検出は、まず被験者の顔検出を行い、それを基に目の検出を行う。そして、目頭の検出、瞳孔中心の検出を行って、それらを基に視線の計測を行う。

5-2-2 顔検出

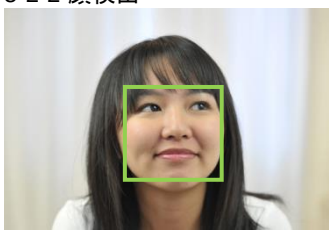
図 6. 顔検出

オープンソースライブラリの顔検出機能を用いて鼻が大凡の中心となる領域の抽出を行う。誤検出防止のため検出位置とサイズの制限を設ける。[図 6]

5-2-3 目の検出

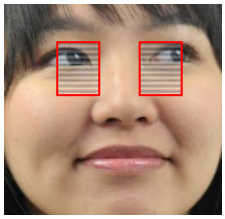
図 7. 目検出

画像の中心を鼻と仮定した場合に目の特徴的な形状の目頭と目の中心付近を含む領域を一般的な顔サイズの比率により限定する。[図 7]

5-2-4 目頭の検出

図 8. 目頭検出

輪郭線の画像に変換し、画像の中心軸から左右両サイド方向に目頭の特徴的な形状の検索を行う。[図 8]

5-2-5 瞳孔の検索

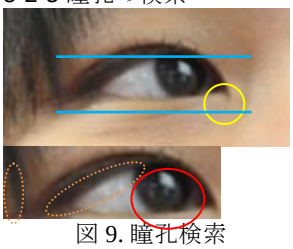

図 9. 瞳孔検索

目頭を基準に目の領域の下限と上限を肌色との輝度境界より限定する。
目尻の特定は困難なため平均的なサイズと仮定する。[図 9 上]

画像の平均輝度に応じて特定の閾値で暗い部分を抽出する。
抽出結果より形状と大きさから黒目を特定し重心点を瞳孔とする。[図 9 下]

5-3 視線認識

5-3-1 カメラと被験者との距離を測る

人の体の寸法には、ある程度の平均値というものが存在する。頭部寸法についてデータをまとめている”日本人頭部寸法データベース(2001)”を使用し、カメラの焦点距離、画角、被写体の画像を顔検出したサイズから、撮像距離を算出する。

また、サンプル画像に対してオープンソースライブラリを使用して顔検出を行うことで、1m の距離から撮影した場合の顔サイズを取得し、データ化することで、更に誤差を少なくする。

5-3-2 視線方向認識

収集したサンプル画像を用いて、カメラと人の距離が1mの状態である時、真正面を向いてカメラに視線がある場合を0°、目印を見ている状態を45°とし、視線の位置を顔の向きを変えた時にカメラに視線がある場合の目頭と瞳孔の位置関係を収集し、間を補完してデータベースを構築する。
データベースと5−3−1で求めた、被験者とカメラとの距離データを組み合わせて、現在の視線方向を認識する。
[図10]

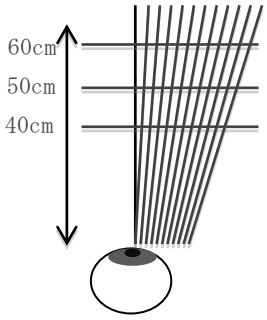

図 10. 注視点の角度によって瞳孔の変化イメージ

6. 検証方法

カメラと被験者との距離を xcm、被験者が注視する目印の角度を y°とする。[図 11]

x、y の組み合わせパターンを考え、視線がカメラに向いているのか、さらに目印を注視している位置を認識出来ているか検証を行う。

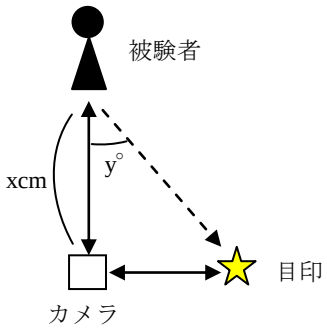

図 11. 実験イメージ

5. 今後の課題

2014 年 10 月にプロトタイプが完成予定であるが、その時点ではまだ RaspberryPi での検証は行っていないので、RaspberryPi へ移行して検証、精度の向上を繰り返す必要がある。

また、被験者の人数や、どのような人に被験してもらうかも含め検討する必要がある。

本研究では、精度の検証は常に同じ場所(室内)で行うが、実際に自然光により大きな影響を受けることが分かっているため、場所やカメラの向きによってある程度パラメータで吸収できるような手法が必要になる。

顔の向き、角度をの検出することについては本研究では行わないが、今後必要になると考えている。

8. 文献

[1] 阿部清彦、佐藤寛修、大山実、大井尚一、視線による重度肢体不自由者向けコンピュータ操作支援システム、映像情報メディア学会、Vol.60、No.12、pp.1971-1979、2006.

[2] 阿部清彦、大山実、大井尚一、自然光下における画像解析を用いた多指標視線入力システム視線、映像情報メディア学会、Vol.58、No.11、pp.1656-1664、2004.

[3] 画像処理アルゴリズムの基本と仕組み(2012.10 長尾智晴)

[4] 視線検出とプレイヤー動向(2014.06.14 URL:http://so-ti.com/report/detail151.html)

[5] 顔検出用のプログラムを動かすまでのメモ (2014.06.14 http://qiita.com/Jey/items/334d143767aabfdc4145)

[6] 日本人頭部寸法データベース2001 (2014.06.14 https://www.dh.aist.go.jp/database/head/
河内まき子・持丸正明, 2008：日本人頭部寸法データベース2001, 産業技術総合研究所H16PRO-212.
Makiko Kouchi and Masaaki Mochimaru, 2008: Anthropometric Database of Japanese Head 2001, National Institute of Advanced Industrial Science and Technology, H16PRO-212.

[7] マイトビー (2014.07.23 http://www.tobiiatj.com/jpn/c15_top.html)

[8] The Eye Tribe Tracker (2014.07.23 https://theeyetribe.com/)