

共同注意が計測可能なモバイル視線計測システムの開発 Development of Mobile Eye Tracker to Enable Joint Attention Measurement

今淵 貴志[†] 落合 貴之[‡] 堀江 友祐[†] プリマ オキ ディッキ A.[†] 伊藤 久祥[†]
Takashi Imabuchi Takayuki Ochiai Yusuke Horie Prima O. D. A. Hisayoshi Ito

1. はじめに

視線計測システムは、視線情報の分析を行う心理学やマーケティングなどの分野で広く利用されており、当該システムの開発も活発に行われている[1, 2]。視線計測システムは、大別して非接触型と接触型がある。前者は、低価格化が進んでおり、その多くがカメラユニットのみで販売されている。そのため、計算機ユニットのための PC をユーザが別途で準備する必要がある。後者は、モバイル視線計測システムであるため、カメラユニットと計算機ユニットが統合システムとして販売されていることで、前者と比較して高価である。従来、視線計測システムを利用する場面において、1人の利用者による視線計測を行う事例が多いが、近年、共同注意を計測する研究分野などのように複数の利用者の視線を同時に計測し、取得した被験者らの注視点を重合表示する研究が増えてきており、接触型視線計測システムの需要が増している。

しかしながら、既存の接触型視線計測システムの単価は100万円以上であり、複数人の視線を計測するためには、大きな設備投資が必要である。一方、既存の接触型視線計測システムの多くは、使い易いシステムとはいえない。その原因は、キャリブレーションのために、複雑な操作を要することや、視線を正しく計測できることを確認するための一連の作業に補助員が必要であることが挙げられる。さらに、キャリブレーション終了後から時間が経過すると、視線計測のずれが増大することもあり、複数人の被験者の視線を同時に計測するためには、キャリブレーション作業を短時間で完了する方法が必要である。

上述のように、接触型視線計測システムの課題を解決するための方法も考えられるが、それによって、計算量が膨大になり、その一連の処理を可能である小型計算機ユニットが少ないことから当該視線計測システムの低価格化が困難である。ここ数年において、Raspberry Pi などのように教育向けに種々の小型計算機ユニットが開発されてきており、安価な接触型視線計測システムの開発の可能性が出てきている。しかしながら、視線計測を利用する研究分野において、注視点の他に停留点 (fixation) やサッケード (saccade) を計測する必要があり、視線の計測速度を確保する必要がある。例えば、停留点を求めるためには、200ms以上の視線計測速度が必要であるが、サッケードを求めるためには最低 50ms の視線計測速度が必要である[3]。

そこで本研究では、視線計測を容易に行えることを前提として、共同注意を計測可能な低価格の小型視線計測システムを開発し、当該システムを Raspberry Pi 上で実装する。

表 1 既存の視線計測システムの仕様

項目	Tobii グラス2	NAC EMR-9
サンプリングレート	30 Hz	60 Hz
質量	45 g	150 g
アイカメラ数 ^{※1}	1	2
データ記録時間	60分	120分
視野角	90°	44° 62° 92° 121°
検出方式	暗瞳孔法	
価格 ^{※2}	約500万円	

^{※1} 1つの眼球に対するカメラ数

^{※2} ハードウェアとソフトウェアのセット価格

2. 既存のモバイル視線計測システム

既存の接触型視線計測システムの製品には、Tobii グラス 2 (株式会社トビー・テクノロジー) や NAC EMR-9 (株式会社ナックイメージテクノロジー) などがある[4, 5]。両者は、メガネ型の視線計測デバイスであり、視線計測を行う準備に補助者が必要である点において共通している。表 1 は、それぞれの視線計測システムの仕様を示す。

Tobii グラス 2 では、キャリブレーションを次のように行う。まず、当該機器を装着した利用者の視野内に、補助者が専用の赤外線マーカーを動かしながら、利用者に赤外線マーカーを注視させる。次に、キャリブレーションの結果を専用のディスプレイユニットから確認し、一定以上の精度を確保するまでキャリブレーション作業を繰り返す。NAC EMR-9 は、ほぼ同様なキャリブレーション方法を用いるが、赤外線マーカーの代わりに補助者がレーザーポインタなどを用いて注視すべき地点を提示する。キャリブレーション時に、頭部姿勢が変動すると視線計測の精度が低下するため、利用者に対して、補助者はキャリブレーション中にできるだけ頭部を動かさないように指示する必要がある。

以上のように、キャリブレーションを利用者 1 人で行うことは困難であり、キャリブレーション中に利用者は頭部を動かすことができない。そのため、キャリブレーションを精度良く行うことは困難であり、複数人での利用を考えた際、複数の補助者が必要であるという問題がある。さらに、前章で述べた通り、キャリブレーションが短時間で完了できなければ、時間経過による視線計測のずれが懸念される。これらのことから、当該システムは複数人の視線を計測する用途には向かない。

3. 提案のモバイル視線計測システム

これまで我々は、キャリブレーション時の頭部変動によるずれを補償するキャリブレーション手法を用いた接触型視線計測システムの開発を行ってきた (Horie et al., 2014)

[†] 岩手県立大学大学院ソフトウェア情報学研究科

Iwate Prefectural University Graduate School of Software and Information Science

[‡] 富士通ビー・エス・シー

Fujitsu Broad Solution & Consulting Inc.

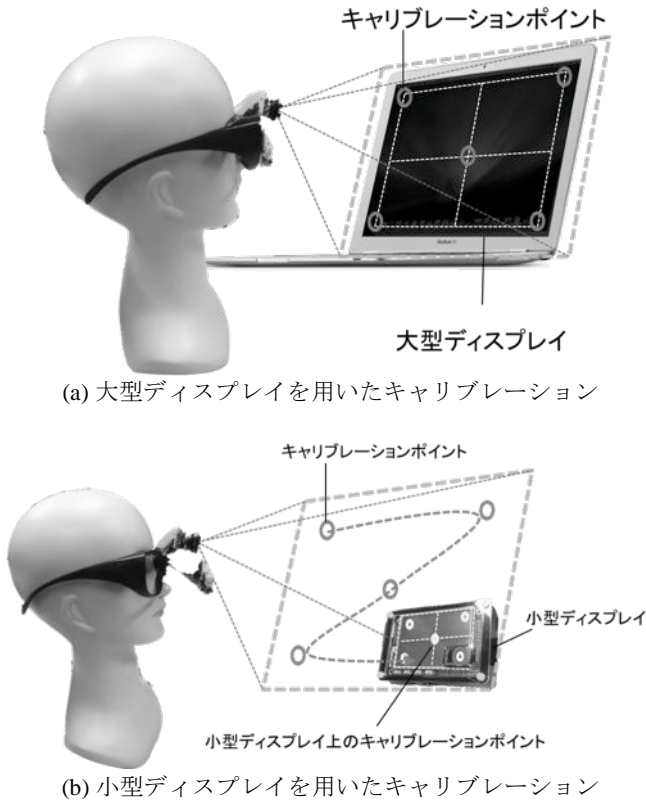


図 1 既存の視線計測システムの仕様

[6]. 当該システムは、カメラユニットと計算機ユニット (PC) から構成されており、PC を除いたコストは約 1 万円と安価である。視線計測では、眼球撮影用の赤外線カメラ (アイカメラ) で瞳孔を撮影する。また、PC の画面をマーカーレス AR として利用し、利用者が見た風景を撮影するための可視光線カメラ (シーンカメラ) で当該画面の位置を追従し、頭部変動に合わせて注視点の座標を補正させることで自然な体勢でキャリブレーションが行える。

本研究では、複数人の視線を同時に計測できるモバイル視線計測システムを実現させるために、以下に示すような小型ハードウェアの構成を決定し、それに対応した各種処理方法を検討する。

3.1 ハードウェア構成

ここで、モバイル視線計測システムを製作するために、小型計算機ユニットを利用するが、それに伴う周辺機器を次のように選定する。まず、電源ユニットおよび周辺機器の稼働電圧 (電流) が 5V (1A) 以内のものを利用する。これは、一般に入手できるモバイル充電電池の仕様を考慮したものである。次に、アイカメラ・シーンカメラの電源は、小型計算機ユニットから供給できるものとする。最後に、瞳孔の検出やキャリブレーションの結果などを表示できる小型ディスプレイを設置するが、これには小型計算機ユニットの内部電源を利用できるものとする。以上のハードウェア構成によって、提案するモバイル視線計測システムを簡略化・軽量化することが可能であると考えられる。

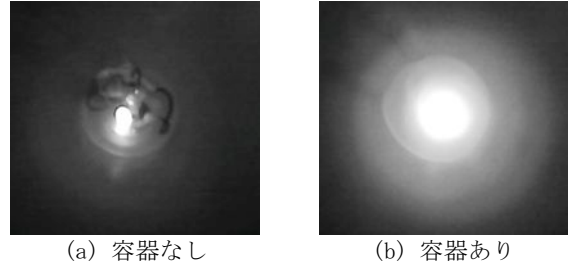


図 2 乳半色のアクリルの有無における赤外線 LED の発光の違い

3.2 新キャリブレーション方法

前章のように、モバイル視線計測システムには、大きな画面が搭載されないため、キャリブレーションにおいて補助者が必要であった。ここで、補助者を必要とせず、自然な体勢でキャリブレーションが行える方法を提案する。図 1 は視線計測システム利用者と計算機ユニットの位置関係を示す。図 1(a)では、大型ディスプレイ上にキャリブレーションポイントを表示し、利用者がそれらのポイントに注視しながらキャリブレーションが行えるが、同様な操作を小型ディスプレイで次のような手順で行えるようにする。

まず、小型ディスプレイ上にシーンカメラの映像とそれに重ねるように固定されたキャリブレーションポイント (5 点) を表示する。図 1(b)のように利用者は、小型ディスプレイ上のキャリブレーションポイントと小型ディスプレイのシーンカメラで捉えた画像内における当該ディスプレイの位置を重ねるよう計算機ユニットを移動させる。利用者はディスプレイ上のポイントを一定時間注視することで、そのポイントに対するアイカメラとシーンカメラの座標対を取得する。すべてのポイントを注視し終えた後、ホモグラフィ変換でキャリブレーションを行う (式 1)。

$$\begin{bmatrix} h_{11} & h_{11} & h_{11} \\ h_{11} & h_{11} & h_{11} \\ h_{11} & h_{11} & h_{11} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (1)$$

ここで、 $h_{11} \sim h_{32}$ はホモグラフィ行列であり、 s はスケールである。また、キャリブレーションで使用した各ポイントの注視点群を有意水準 95% に満たすようにする。

3.3 複数人の注視点の重合表示

共同注意を計測する場合、複数人の注視点を同一座標系で表現しなければならない。ここで、注視者の位置や姿勢などの違いを吸収し、注視者らの注視点の座標を正規化する必要がある。本研究では、共同注意の対象物において、4 つの基準点を設け、これらの基準点を各視線計測システムのシーンカメラで捉え、上記の式 1 を利用して各注視者らの注視点の座標系を正規化する方法を提案する。

a. 赤外線 LED を用いた基準点の設置

基準点として、AR マーカーなどを利用することができるが、注視者らにとって見慣れないマーカーなどがあると自然に注視実験を行うことが困難である。ここで、赤外線 LED を基準点として利用することで注視者らに気づかれることなく、座標の正規化が行えることが可能であり、自然な姿勢で複数人による共同注視実験が行えると考えられる。

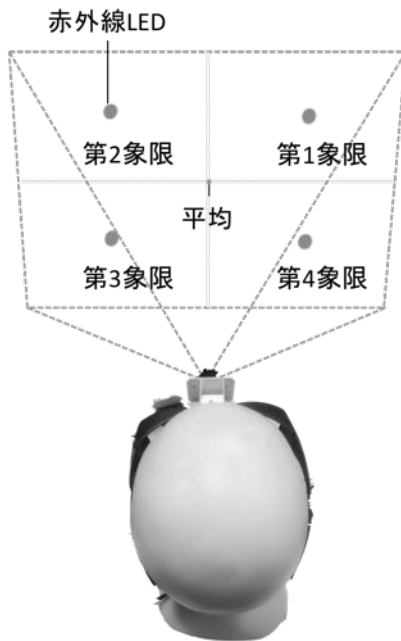


図3 基準点のラベリング

なお、カメラで LED の赤外線光を捉えやすくするために、当該 LED を透過性のある乳白色（乳半色）の亚克力容器に入れて赤外線を拡散させる。これによって、LED から約 5m 離れた場所から、LED の位置を十分に捉えることができる。図 2 は、乳半色の亚克力容器の有無における LED 赤外線光を示す。

b. 基準点の検出

赤外線の基準点を検出するために、シーンカメラとして赤外線光を撮影できるカメラを利用する。注視者らの頭部 roll 角度の変動を左右 45° 以内であると仮定すると、当該カメラで捉えた基準点のラベリングを容易に行える。図 3 は、各基準点のラベリングの概要を示す。4 つの基準点の平均を直交座標系の原点とし、第 1 象限から第 4 象限上に存在する基準点のラベル番号を当該象限番号とする。

c. 注視点座標の正規化

各注視者の注視点座標を上記の基準点でホモグラフィ変換し、同一座標系にする。この処理は、リアルタイムまたはオフラインで行うことができるがリアルタイムの場合は、注視点を正規化するサーバーを設ける必要があり、各視線計測システムとサーバーとの間の座標変換を通信によって実現する。

d. 重合表示

正規化した注視点群を同一座標系のグラフにプロットし、その密度を混合ガウシアンモデル（式 2）で表現する。

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left\{\frac{(x-\mu_x)^2}{\sigma_x^2} - 2\rho\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right\}} \quad (2)$$

ここで、 σ_x と σ_y はガウシアン の x 方向および y 方向の標準偏差、 ρ は相関係数、 μ_x と μ_y は平均値を表すが、注視点を重合表示するために、 σ_x と σ_y を横と縦のマスクの幅、 μ_x と μ_y を眼球の中心座標を決定するものとして扱う。な

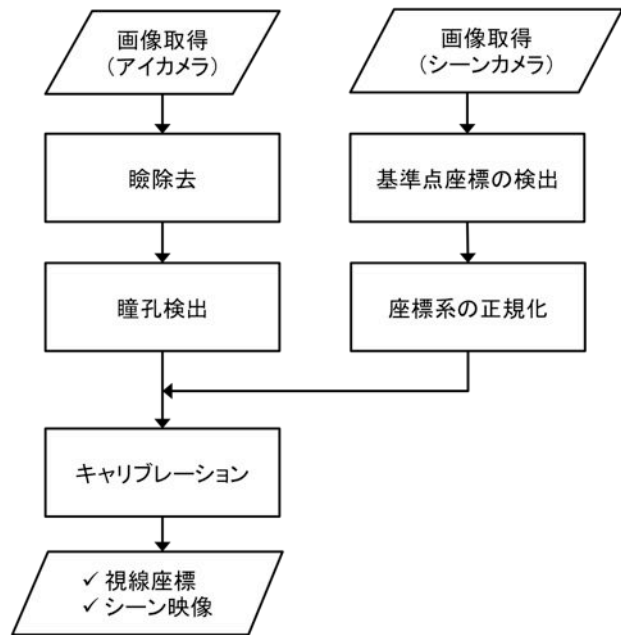


図4 本研究での視線計測アルゴリズム

お、 ρ を 0 とすることでマスクの回転を行わないとする。

3.4 視線計測処理の高速化

図 4 は、提案のモバイル視線計測システムのアルゴリズムを示す。図 5 は、瞳孔中心座標の抽出の様子を示す。視線計測処理を高速化するために、以下のような工夫を施す。

a. 瞼のマスクング

アイカメラと眼球との位置が変化しないことから、瞼の領域を図 5 (b) に示すようなガウシアンマスクで除去できる。実際の処理では、アイカメラから得た画像とガウシアンマスクの画像を加算することで実現できるため、本瞼除去処理を高速に行える。

b. Blob 検出による瞳孔の検知

瞳孔を検知するには、目領域の画像（図 5(a)）に対しテンプレートマッチングやスターバーストなどの方法が提案されているが[7, 8]、これらの方法よりも高速に実行できる Blob 検出を瞳孔検知に利用する。ここで、検出された Blob に対して楕円近似を行い、楕円の中心座標を瞳孔の中心座標とする。図 5 (c)～(f) は各処理を行ったときのアイカメラの映像を示す。

c. アイカメラ画像の低解像度化

アイカメラの解像度を QVGA (320×240 画素) に設定することで、一連の瞳孔検知処理をより高速に行える。当該システム利用時、アイカメラと瞳孔との距離は 10cm 程度しか離れないと想定されるため、QVGA の解像度で瞳孔を撮影しても、十分に瞳孔の検出が行える。

d. レコーディングの並列化

レコーディングは視線計測と同時に進行する必要があり、各フレーム画像のエンコーディングとエンコーディング後のフレームの書き込みを並列化することで視線計測時の処理速度の低下を抑える。ここで、エンコーディングの処理のために ffmpeg[9]を採用した。

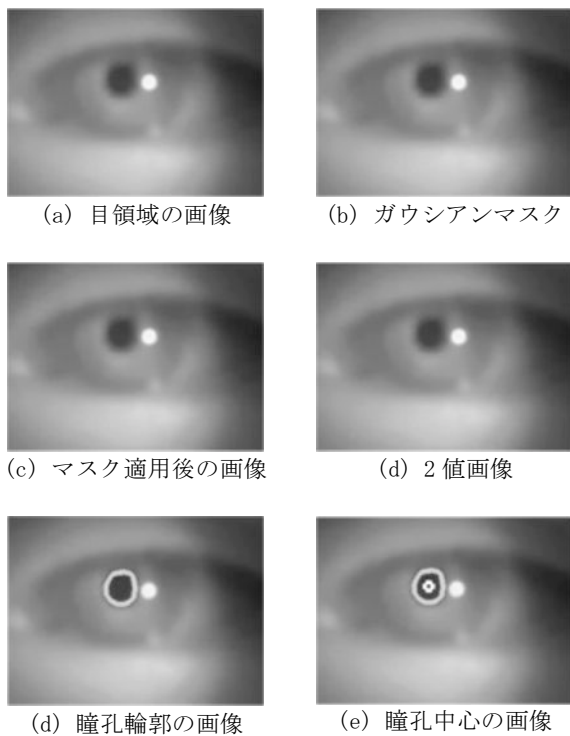


図 5 瞳孔中心座標の抽出

表 2 ハードウェア構成

項目	システム1	システム2
計算機	Raspberry Pi B+	Raspberry Pi 2 B+
OS	Raspbian	
CPU	ARM1176-JZF-S (950Hz)	ARM Cortex-A7 (960Hz)
RAM	512MB	1GB
ディスプレイ	Pi TFT (2.8 inch)	Pi TFT (3.6 inch)
アイカメラ	Pi Noir	USB 2.0 Camera (IR lens)
シーンカメラ	MS LifeCam HD-3000	MS LifeCam HD-3000
その他	KSW-1 タクトスイッチ	

4. 実装

本研究で提案するモバイル視線計測システムを 2 種類のシングルボードコンピュータ上で実装する。表 2 は、2 つの試作システムの構成を示す。ここで、Raspberry Pi B+ の処理能力が低いため、アイカメラとして専用のカメラ (Pi Noir) を採用した。一方、Raspberry Pi 2 B+ では、高速に USB カメラのキャプチャ処理が行えると考えられるため、機器の小型化のために、アイカメラとして USB カメラを採用した。なお、実装機器において、2.8 inch と 3.6 inch のディスプレイを使用したのは、キャリブレーション時の使いやすさを比較するためである。ここで、各システムを約 2 万円で試作できるため、少ない設備投資費用で複数人の視線を計測する実験を行うことができると考える。

図 6 は、実際に試作した機器を示す。図 6(a)において、2.8 inch のディスプレイに GPIO スイッチを配置し、視線計測に必要な操作をこれらのスイッチで行えるようにした。



(a) システム 1



(b) システム 2

図 6 本研究で試作した 2 つのモバイル視線計測システム

一方、図 6(b)において、より大きいディスプレイを採用することから、GPIO スイッチをディスプレイ周辺に配置しないが、GPIO スイッチの代わりに Bluetooth で接続可能なキーボードを利用した。

5. 評価

実装した試作システム 2 つにおける、キャリブレーションに要する時間や、視線計測の精度とそのサンプリングレート、バッテリーによる連続稼働時間について評価を行う。

5.1 キャリブレーションに要する時間

試作システムでは、5 点によるキャリブレーションを行っており、各点の注視時間を 2 秒に設定した。上述のように、1 つのキャリブレーションポイントから次のポイントへの注視において、利用者自身でその注視開始を指定することができるため、システムに慣れた被験者がキャリブレーションを行った場合に要する時間を約 15 秒で完了できることを確認した。なお、試作した 2 つのシステムのディスプレイサイズが異なるが、そのサイズによるキャリブレーションに要する時間において顕著な差がみられなかった。

5.2 視線計測の精度とそのサンプリングレート

試作システムの視線計測精度について、著者らによる予備実験によって検証を行った。実験の際、姿勢やキャリブレーション時間の制約を設けずに視線計測を行った結果、約 1 度の視線計測精度を確保できることを確認した。これによって、複数人の注視点を約 5m 離れたスクリーンに投影したとき、その誤差を 10cm 未満に確保することができる。プレゼンテーション中のスライドに対する複数人の視線計測による評価などの利用においても十分である。

表 3 サンプリングレート (システム 1)

	描画あり		描画なし	
	最大	平均	最大	平均
場面1	14.8 fps	13.7 fps	18.9 fps	18.1 fps
場面2	8.1 fps	7.5 fps	13.7 fps	13.3 fps
場面3	8.6 fps	8.3 fps	13.4 fps	12.9 fps
場面4	11.3 fps	10.8 fps	13.5 fps	13.1 fps
場面5	7.4 fps	7.1 fps	8.1 fps	7.7 fps

表 4 サンプリングレート (システム 2)

	描画あり		描画なし	
	最大	平均	最大	平均
場面1	30 fps	30 fps	30 fps	30 fps
場面2	30 fps	30 fps	30 fps	30 fps
場面3	30 fps	30 fps	30 fps	30 fps
場面4	30 fps	30 fps	30 fps	30 fps
場面5	30 fps	30 fps	30 fps	30 fps

次に、試作システムの処理速度について 5 つの処理場面におけるサンプリングレートをもとに検証する。表 3 と表 4 は、それぞれの視線計測機器の各処理で計測したサンプリングレートを示す。ここで、場面 1~5 はそれぞれ「アイカメラとシーンカメラから映像を取得する場面」、「瞳孔の中心座標を検出する場面」、「キャリブレーションを行う場面」、「視線情報を算出する場面」、「視線情報の録画を行う場面」である。また、各処理場面において、ディスプレイに映像を描画する場合と、描画しない場合のサンプリングレートを検証した。

表 3 の場面 1~4 において、描画を行った場合と行わなかった場合のフレームレートに大きく差が見られた。これは、Pi FTB を GPIO で接続しているため、映像の出力に GPU を使用することができず、CPU のみで描画処理を行っているためである。一方、表 4 では、どの場面においても 30 fps で処理を行うことができていることを確認した。この処理速度は、試作システム 2 で使用したカメラの映像取り込みの最大処理速度であり、これらのカメラよりも高速なカメラを利用すれば、さらに視線計測のサンプリングレートを向上させることができると考えられる。

第 1 章で述べた通り、停留点を求めるためには、200ms 以上不動な注視点を計測する必要があるが、不動な注視点数を 5 点とした場合、25 fps 以上の処理速度が必要である。このことから、試作システム 1 は、利用者の注視点を計測し、他の利用者の注視点と重合表示することが可能であるが、停留点を求めるためには、試作システム 2 の仕様が必要不可欠である。また、高速なサッケードを分析するためには、さらに高速な処理ユニットが必要である。

5.3 電池による連続稼働時間

提案した 2 つのシステムを 5200mAh の充電電池で稼働させ、その連続稼働時間を測定した。その結果、約 2 時間の

視線計測が行えることを確認した。使用した Raspberry Pi の電源は、5V (2A) であるため、期待通りの稼働時間であるといえる。このことから、共同注意実験において十分な稼働時間を確保できていると考えられる。

6. おわりに

本研究では、オンボードコンピュータを利用して複数人の視線を計測し、共同注意を計測可能であるモバイル視線計測システムを開発した。また、従来のキャリブレーション方法の複雑さを解決するために、視線計測の計算機ユニット自体を利用者が動かすことで利用者単独によるキャリブレーションを可能にするキャリブレーション方法を提案した。さらに、ハードウェアの構成を安価に実現したことで、少ない設備投資費用で複数人の視線を同時に計測する実験を可能とし、提案システムを教育や画像・映像評価などの諸研究分野において大いに活用できると考える。

評価の結果から、利用者単独によるキャリブレーションに要する時間は約 15 秒であり、さらに視線計測精度が約 1 度であることが確認できたため、通常の視線計測において十分に実利用が可能であると考えられる。また、Raspberry Pi 2 B+を用いた試作システム 2 において、一連の処理を通して、30 fps のサンプリングレートで視線計測を実現したため、停留点を求めることが可能である。今後、視線計測精度、キャリブレーションに要する時間、複数人の視線の重合表示の精度に関して詳細な実験を行う必要がある。

今後、より高速処理可能な小型コンピュータと高速なカメラを利用し、提案システムの計測速度を改良し、高速なサッケードを分析可能にする予定である。また、本システムの各アルゴリズムに改良を加えるとともに GUI を作成することで、利用者にとって使いやすいシステムを実現し、より広い分野においての利用が望めると考えられる。

参考文献

- [1] Li, D., Babcock, J., and Parkhurst, D.J., "openEyes: a low-cost head-mounted eye-tracking solution", ETRA '06 Proceedings of the 2006 symposium on Eye tracking research & applications, pp.95-100 (2006).
- [2] Broz, F., Lehmann, H., Nehaniv, C.L., and Dautenhahn, K., "Mutual gaze, personality, and familiarity: Dual eye-tracking during conversation", RO-MAN, 2012 IEEE, pp.853-864 (2012).
- [3] Fischer, B., Ramsperger, E., "Human express saccades: Extremely short reaction times of goal directed eye movements", Experimental Brain Research, Vol.57 (1984).
- [4] トビー・テクノロジー社 Tobii グラス 2
<http://www.tobii.com/ja-JP/eye-tracking-research/japan/products/hardware/tobii-glasses-eye-tracker/>
- [5] ナックイメージテクノロジー社 モバイル型アイマークレコーダ EMR-9 http://www.eyemark.jp/product/emr_9/index.html
- [6] Horie, Y., Prima, O.D.A., Imabuchi, T., Ito, H., "Unrestricted Head Movement Calibration for a Head-mounted Eye Tracker", The Fourth IEEE International Workshop on Image Electronics and Visual Computing, 5p, (2014).
- [7] Zhao, Z., Fu, S., and Wang, Y., "Eye Tracking Based on the Template Matching and the Pyramidal Lucas-Kanade Algorithm", CSSS '12 Proceedings of the 2012 International Conference on Computer Science and Service System, pp.2277-2280 (2012).
- [8] Li, D., Babcock, J., and Parkhurst, D.J., "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches", CVPR '05 Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol.3, pp.79 (2005).
- [9] FFMPEG, <http://ffmpeg.org>, (2015 年 4 月 15 日に閲覧).