

固視の3次元位置と仮想物体の表面形状 を用いた視線計測器の自動校正

浦宗 龍生^{*1}

池田 聖^{*1}

石塚 裕己^{*1}

大城 理^{*1}

Abstract — 使用者の視野の顕著性マップと視線との整合から視線計測器を自動校正する手法は、バーチャルリアリティ(VR)と親和性が高いが、顕著性マップの収集に時間を要する、高精度な顕著性マップの生成は計算コストが高いという問題がある。本稿では、顕著性マップや能動的な校正動作を必要としない視線計測器の自動校正法を提案する。提案手法は、使用者が頭部に視線計測器を装着し、頭部の位置姿勢、シーン中の物体の3次元形状が取得できる状況を前提とし、物体の表面形状と固視中における注視点群の分散指標を最小化することで校正パラメータを推定する。これにより、使用者が自由に頭部を移動させシーンを見渡している間に視線計測器が自動校正される。VR環境での仮想物体を用いた実験を通して、提案手法により自動校正可能であることを確認し、マーカに対する距離を指標とする従来手法と同程度の精度であることを確認した。

Keywords : 視線計測, 自動校正, 固視検出, 3次元形状

1 はじめに

視線計測は、様々な分野で応用されている。例えば、心理実験や市場調査などでは、視線情報は人の興味や関心、精神状態の解析などに用いられ、視線の入力インタフェースや中心窩レンダリングなどでは、計測した視線を実時間で利用する用途に用いられる。特に、カメラで撮影した眼の画像から使用者の注視点群を計測する方式は、カメラの小型化、低価格化により広く普及し、バーチャルリアリティ(VR)用もしくは複合現実感(MR)用ヘッドマウントディスプレイ(HMD)では、視線計測器は標準搭載されつつある。

一般に、視線計測器は、使用者毎に校正する必要がある。視線計測器の校正は通常、視線計測を始める前、場合によってはその後にも、基準となるマーカを使用者に提示し、使用者にマーカを注視する動作を要する。校正では、注視しているときの視線方向とマーカとの位置関係から各種校正パラメータが推定される。次章で述べるように、様々な視線計測手法が提案されているものの、いずれの校正手法においても、使用者毎に個人差がある何らかの校正パラメータを推定する必要がある。また、校正後も視線計測器の位置が変化すると、視線計測の精度が低下し再校正が必要となる場合もある。自動的に視線計測器を校正することができれば、視線計測後に一括解析する用途においては計測時間の短縮が可能となる。これにより、視線計測器を実時間利用する用途においては使用者が視線計測を意識する必要がなくなる。

これまで、視線計測の自動校正に関する様々な手法

が提案されてきた [1]。自動校正法は大きく分けて、視線情報に加えて使用者が視線計測を意識せずに入力する情報を手掛かりとする手法 [2, 3] と、そのような入力も用いずに視界に写る画像を用いる手法 [4-6] に分けられる。視界画像を用いる手法では、画像から視覚的顕著性の計算モデルにより注視しやすい箇所の確率分布 (以下、顕著性マップと呼ぶ) が予測され、高確率な箇所と視線とそのずれを最小化することで校正パラメータが推定される。しかし、この手法では、多数のフレームで顕著性マップを積算する必要があり、校正には時間を要するとともに、顕著性マップの生成コストが高いという問題がある。

本稿では、顕著性マップとは別の手がかりで、能動的な校正動作を必要としない視線計測器の自動校正法を提案する。具体的には、使用者が頭部に視線計測器を装着し、各時刻に対応する頭部の位置姿勢と、3次元シーン中の物体表面の形状が与えられるとき、使用者が自由にシーンを見渡している間に校正パラメータを推定する。物体表面の3次元形状が与えられていれば、実シーンか仮想シーンかは問わない。提案手法では、上記の問題に対して、物体の表面に分布する固視中の注視点群の3次元位置を指標として校正パラメータを推定する。固視とは、鮮明な視覚情報を得るために、視認対象の像を網膜上の中心窩でとらえる眼球運動であり、固視中の注視点は物体表面上に密集して分布する。視線計測に大きな誤差が含まれるときには、固視中の注視点群の分散が大きくなることを利用して、校正関数のパラメータを最適化する。

^{*1}大阪大学

2 関連研究

2.1 較正手法

眼画像を用いて視線を計測する手法は、モデルベース手法、回帰ベース手法、アピランスベース手法の三つに大別される [1]。モデルベース手法では、画像と眼球の3次元モデルの整合から眼球の位置や姿勢が推定される。3次元眼球モデルとしては、球体、楕円体、回転体 [7,8] などのモデルが提案されている。こうしたモデルを用いても、個人差を吸収するパラメータは必要で、少なくとも光軸と視軸のずれは通常のカメラによる計測では得ることはできない。

回帰ベース手法 [9,10] では、眼画像中の瞳孔中心から視線方向への変換を数理モデルにより表し、数理モデルの回帰分析により較正パラメータが推定される。この変換には、視線検出器と眼球との位置関係や個人差に関わる補正量も含まれている。したがって、このモデルにおいては、視線検出器の使用前だけでなく、使用者が同一であっても視線検出器に位置ずれが生じた場合などには、再較正が必要となる。

アピランスベース手法 [11,12] では、眼画像から直接視線方向を出力する学習器が使用される。学習には、教師データとしてマーカの位置、入力データとしてマーカ注視中の眼画像が必要になる。一般に、アピランスベース手法は、モデルベース手法や回帰ベース手法に比べて精度が劣ると言われている。

2.2 自動較正法

自動較正法は、使用者の視界の顕著性マップを生成する手法と、それ以外の追加情報を用いる手法に大別される。

顕著性マップを用いた手法 Sugano らにより顕著性マップを用いた最初の視線計測器の自動較正が提案された [4]。この手法は、使用者が頭部を固定し平面ディスプレイ上で動画を閲覧する状態が前提となっている。まず眼画像がクラスタリングされ、次に同一クラスタの顕著性マップが積算されて確率マップが生成される。眼画像と確率マップの極大との対応をガウス過程回帰により学習することで、眼画像から画面上の注視位置へのノンパラメトリックな変換が得られる。しかし、頭部固定の前提条件は厳しく、顕著性のマップを積算には一定以上の時間が必要である。

同様に顕著性マップを用いてはいるが、シーンに対して頭部が動く状況が想定された頭部装着型の視線計測器のための自動較正法 [6] も提案されている。この手法では、シーン画像から得た顕著性マップが、3次元眼球モデルから推定された光軸を中心に積算され、視線を顕著性マップの極大方向に補正するための2つの回転パラメータが推定される。この手法においても、

顕著性マップの積算のため十分なフレーム数が必要である。

Shi らは、さらにシーンの3次元形状が既知な仮想空間を対象として、視線計測器搭載のHMDのための自動較正法 [5] を提案した。この手法でも、光軸を中心に顕著性マップが積算され、その極大に視線が補正される。この手法では、HMDの時間経過による装着ずれの補正を目的としており、視線計測カメラの画像上での2つの並進パラメータが推定される。この手法では、シーンカメラがステレオカメラであることを利用して、右目画像、左目画像中の顕著性マップを右から左、左から右にそれぞれ投影する。これにより、両眼による幾何学的拘束が加わり、顕著性マップ積算における極大の抽出効率と精度が改善する可能性がある。

以上のように、顕著性マップを用いた手法では顕著性マップの積算に十分な時間を要することと、顕著性マップを生成する計算コストが高いという問題も存在する。提案手法では、これらの手法のような顕著性マップを全く使用しない点が異なる。また、Shi ら [5] と同様にシーンの3次元形状は既知とする点が共通であるものの、両眼(2視点)の幾何学的整合ではなく、頭部移動時の多視点での整合を考慮している点が異なる。

他の追加情報を用いた手法 使用者の視線情報以外の入力を用いる自動較正法として、マウスの入力情報を用いた視線計測器の較正法 [2] が提案された。この手法では、PC操作中の使用者は、マウスでクリックした場所を注視するという仮定に基づき、そのときの眼画像および頭部姿勢を取得し較正パラメータを推定する。使用者が能動的な動作が不要なため自動較正法に分類したが、マウスの入力を前提とするため用途は限定される。他にも、第三者の使用者の視線パターンを用いた自動較正法 [3] も提案されている。これは、同一シーンであれば異なる使用者においても、視線パターンが位相幾何学的には類似した構造をもつという仮定に基づいており、新しい使用者の視線パターンが他の使用者の視線パターンに整合するように較正パラメータが推定される。しかし、この手法では、同一シーンにおける複数の使用者の視線データを用意する必要がある。提案手法では、こうした追加情報は仮定せずに、シーンの3次元形状が既知であることを前提としている。シーンの3次元モデルはVR用途であれば必然的に利用可能である。MRにおいても外部センサ不要の自立型HMDであれば、visual SLAM [13] 等により自動的に推定されることが多く、提案手法の利用範囲は広い。

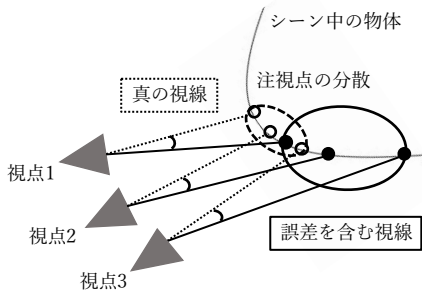


図1 提案手法の原理

3 3次元固視位置を用いた自動較正法

3.1 提案手法の概要

図1に提案手法の原理を示す。視線計測機の較正精度が高いときには、固視中の注視点群は物体表面上の一か所に密集するはずである。しかし、較正精度が低く計測誤差が大きいときには、物体表面上の一か所に密集せずに注視点の分散が大きくなる。この分散の違いは、シーンに対して使用者の視点が並進移動することにより生じるため、提案手法では使用者の視点移動もしくはシーン中の物体が移動することが前提である。

提案手法では、較正モデルは任意であるため、本稿では次のように定義する。較正前後の視線ベクトルをそれぞれ $g, \tilde{g} \in \mathbb{R}^3$ とすると、較正による視線ベクトルの変換関数 f は式(1)として表せる。

$$\tilde{g} = f(g|\theta) \quad (1)$$

ただし、視線ベクトル g, \tilde{g} はシーンカメラ座標系で表されており、 $\theta = \{\theta_1, \dots, \theta_m\}$ は m 個の較正パラメータの集合である。シーンカメラとは、シーンモデルに対する使用者の視野を仮想的なカメラで表したもので、VRの場合はシーンを描画する仮想カメラに対応する。提案手法では、両眼の中央に1台のシーンカメラがあるものとする。

提案手法は、上述の固視中における注視点群の分散の和を評価関数として、評価関数を最小化する較正パラメータ θ を求める問題と考えることができる。次節以降では、固視中の注視点群の検出、注視点群の分散の算出、評価関数の最適化について詳述する。

3.2 固視検出

視線の時系列データから固視を検出する手法は、分散ベース手法と速度ベース手法に大別される[14]。分散ベース手法は、オブジェクト座標系において固視中の注視点どうしの距離が近いことを利用する。代表的なアルゴリズムはI-DT (Dispersion-Threshold Identification) であり、一定時間以上注視点座標が一定の範囲内にあるとき、固視と判定される。速度ベース手法は、シーンカメラ座標系において固視中の注視点の速

度がsaccadeに比べて十分遅いことを利用する。代表的なアルゴリズムはI-VT (Velocity-Threshold Identification) であり、視線ベクトルの角速度が閾値以下のとき、固視と判定する。

本手法の固視検出では、オブジェクト座標で表された固視中の注視点の分散を最小化する性質上、較正前は大きな分散も許容しなければならないため、I-VTアルゴリズムを採用する。具体的には、現時刻 k に取得された視線方向ベクトル g_k が、次式(2),(3)を満たすとき、角速度の閾値を ϕ_{th} として、 g_k を固視と判定する。

$$g_{k-1} \cdot g_k > \cos \phi_{th} \quad (2)$$

$$n_k \geq f_s T_V \quad (3)$$

ただし、 n_k は現時刻 k を含めて式(2)を連続して満たした回数 n_k 、 f_s は視線の取得レートを表す。 T_V は固視と判定可能な最短時間を表し、100 から 200 ms が設定される[14]。

3.3 注視点群の分散

固視検出により同一の固視 i と判定された j 番目の視線方向ベクトル $g_{i,j}$ の集合 $\mathbf{G}_i = \{g_{i,1}, \dots, g_{i,n}\}$ (以下、固視クラスと呼ぶ) が得られる。使用者がシーンを自由に見渡すときは、シーン中の様々な箇所を注視する際に固視が発生し、複数の固視クラス $\{\mathbf{G}_1, \dots, \mathbf{G}_N\}$ が得られる。

提案手法では、各固視クラス \mathbf{G}_i に対して3次元的な分散が算出される。固視クラスの分散の算出には視線が物体と衝突する注視点の座標が必要となる。注視点の座標は、固視クラス中の各視線方向ベクトル $g_{i,j}$ を用いて式(4)で表すことができる。

$$e_{i,j} = \pi_{i,j}(g_{i,j}) \quad (4)$$

ただし、 $\pi_{i,j}$ はレイキャスト関数であり、視線と物体との交点をオブジェクト座標 $e_{i,j}$ で算出する関数である。この関数は、シーンカメラの位置姿勢とカメラの内部パラメータおよびシーンの3次元形状により決まる逆投影関数である。レイキャスト関数 $\pi_{i,j}$ により、固視クラス群 $\{\mathbf{G}_1, \dots, \mathbf{G}_N\}$ に対応する、注視点集合族 $\{\mathbf{F}_1, \dots, \mathbf{F}_N\}$ が得られる。

注視点の分散は、固視クラス \mathbf{G}_i に対応する複数の注視点のオブジェクト座標 $e_{i,j}$ および固視クラスの重心位置 $\bar{e}_i = \frac{1}{n_i} \sum_j e_{i,j}$ を用いて次のように表される。

$$\text{Var}(\mathbf{G}_i) = \frac{1}{n_i} \sum_{i,j} (\pi_{i,j}(g_{i,j}) - \bar{e}_i)^2 \quad (5)$$

3.4 評価関数の最適化

評価関数の最適化では、視線方向ベクトル $g_{i,j}$ を変換関数 f により補正した方向ベクトル $\tilde{g}_{i,j} = f(g_{i,j}|\theta)$

を用いて、式 (6) に示す評価関数を最小化することで最適な較正パラメータ $\tilde{\theta}$ を推定する。

$$E(\theta) = \sum_i^N \frac{1}{r_i^2} \text{Var}(\tilde{F}_i) \quad (6)$$

ただし、 $\tilde{F}_i = \{\tilde{e}_{i,1}, \dots, \tilde{e}_{i,n_i}\}$, r_i は i 番目の固視クラスタの重心位置とシーンカメラとの距離である。上記分散は、遠い注視点群の影響が高くなることを避けるために r_i により正規化される。

4 実験手法

実験では、最適化における評価関数の収束性と較正精度を確認する。

4.1 実験環境

HMD 本実験は、シーン中の物体の形状を簡易に取得できる VR 環境下で実施した。HMD には、HTC Vive Pro Eye を用い、視線計測には、HTC Vive Pro Eye 内蔵の視線計測器を使用した。視線データ、頭の位置姿勢は 90 Hz で取得した。視線計測器を使用する場合、HTC Vive Pro Eye のシステム内で能動的な較正が必要である。個人差と位置ずれを較正できることを確認するために、初期値として被験者と異なる人物で、視線計測器の較正を行った。

シーン 曲面、平面 1、平面 2 の 3 種類の仮想物体を Unity 上に構築した。曲面は、図 2 に示すように、奥行方向に 50cm の凹凸がある面であり、理想的な条件下で提案手法の有効性を確認するためにこのような形状を用いた。具体的には、物体の形状が連続かつ滑らかで、較正誤差があるときに固視の分散が大きくなりやすいような様々な法線方向をもつ物体とした。曲面には 9 点の十字マークがあり、マークの格子間隔は平面 1 では 1 m、平面 2 では 0.7 m とした。平面及び曲面は実験参加者正面の約 4 m 前方に配置した。

平面 1 は、4.3 節で詳述する baseline 条件のみでの較正データ取得に使用するシーンである。曲面と同じ大きさで奥行方向には幅をもたない平面を約 4 m 前方に配置した。曲面同様、9 点の十字マークがある。平面 1 はシーンカメラ座標に固定されており、シーンに対して頭部が動かないようにした。

平面 2 は、4.3 節で述べる全ての条件を評価するために使用する物体で、平面 1 とは異なる位置にマークを配置した。平面 2 も、平面 1 と同じ位置に固定した。**タスク** 被験者は著者 1 名で、各マークを 3 秒間ずつ見たときの視線情報を取得した。このとき、被験者の頭の位置姿勢は自由に動かすことができる。

自然な姿勢でマークを注視すると、殆どの視線はシーンカメラの画角内で視野角 10 deg 以内に収まったため、着座のまま頭部位置は視点移動が発生するよ

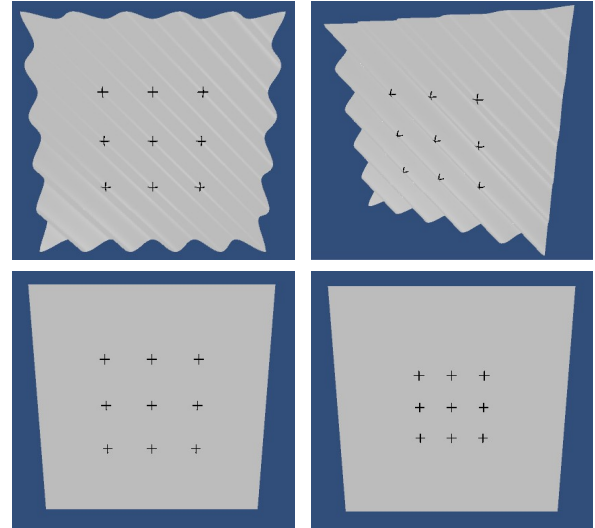


図 2 実験に用いたオブジェクト。(上段) 曲面オブジェクトを正面および右上から見た様子。(左下) baseline 用の平面 1, (右下) 評価用の平面 2 の外観。

うに左右に動かしてデータ (I) を取得し、画角内の周辺にも視線が得られるように、頭部の位置と向きも意図的に動かしてデータ (II) を取得した。

4.2 視線解析手法

視線情報のフィルタリング 本実験では、視線データ取得用のソフトウェアライブラリとして、SRanipal SDK を使用しており両眼の中点を起点とした両眼の視線ベクトルの平均を使用者の視線方向ベクトルとして用いた。また、同 SDK では、瞼の開閉度合いを 0 (完全に閉じている状態) から 1 (大きく開いている状態) の範囲で取得でき、瞬きに対応する 0.7 以下の視線データは排除した。

固視検出 は、シーンカメラ座標系における視線ベクトルの角速度が 30 deg/s ($\phi_{th} = 0.333 \text{ deg}$, $f_s = 90.0 \text{ Hz}$) 以下の視線データが $T_V = 100 \text{ ms}$ 以上連続して得られたとき、その視線データ群を 1 つの固視と判定する。また、固視クラスタの分散が極端に大きな場合 ($\text{Var}(\mathbf{G}_i)/r_i^2 > 6.00 \times 10^{-5}$) は、外れ値として固視クラスタを除いた。この外れ値の閾値は、各固視クラスタの分散を確認した上で経験的に決定した。

較正関数 本実験で使用した較正関数について説明する。シーンカメラ座標における視線方向ベクトル g を、視線と z 軸方向 4m 先に正対する仮想平面との交点座標で表すと、 $g = (x, y, 4)$ と書ける。実験では、この (x, y) を変換する関数を較正関数として、1 次の多項式を使用した。具体的には、較正後の座標を (\tilde{x}, \tilde{y}) とすると、式 (7), (8) で表される関数を用いた。

$$\tilde{x} = \theta_0 + \theta_1 x + \theta_2 y \quad (7)$$

$$\tilde{y} = \theta_3 + \theta_4 x + \theta_5 y \quad (8)$$

ただし、最適化におけるパラメータ $(\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ の初期値は $(0.0, 1.0, 0.0, 0.0, 0.0, 1.0)$ とした。

4.3 実験条件

較正精度の比較のための実験条件について詳述する。全ての条件において、求めた較正パラメータで較正し平面 2 により評価した。評価では、較正後の視線と平面 2 の交点と各マーカとの距離の平均を視点からの角度単位で表したものを誤差とした。各条件の詳細は次の通りである。

raw では、未較正のまま評価した。

baseline では、平面 1 によりマーカと注視点との距離を指標に較正パラメータを推定し評価した。

variance1 では、曲面のシーンの視線データを用い、計測開始から終了までの全ての固視クラスタを用いた。較正パラメータは、式 (6) の分散指標を用いて求めた。

variance2 では、マーカ出現 1 秒後から 2 秒後までの固視クラスタのみを抽出して、variance1 と同じ方法で較正パラメータを求めた。

distance では、variance2 と同一のデータを用い、較正パラメータ最適化の評価関数に注視点とマーカ間の距離の二乗和を用いた。

4.4 評価関数の収束性確認手法

データ (II) を用いた variance1 条件下で較正パラメータ空間における式 (6) の評価関数値の分布を確認した。具体的には、variance1 条件で得られた較正パラメータを中心に、 θ_1 および θ_5 をを変化させたときの $E(\theta)$ を調べた。 θ_1 および θ_5 を選択した理由は、 \hat{x} および \hat{y} に与える影響が大きいと考えたためである。

5 実験結果と考察

5.1 評価関数の収束性

図 3 に評価関数値 $E(\theta)$ の分布を示す。●印は初期値 (raw), ▲印は提案手法 (variance1), ▼印は能動的な較正を必要とする一般的な較正手法 (baseline) により定めた較正パラメータを示す。図 3 より、推定値付近が極小になり、周囲が増加する形状であった。このため初期値が別の値であっても、安定して推定値付近に収束可能であることがわかる。一部、局所最適解が存在するものの初期値付近から発散せず、局所的にはおおよそ下向きに凸の形状をしていることを確認した。

5.2 精度評価結果

図 4 に各条件の平均誤差を示す。図中の左下がり斜線はデータ (I) を表し、右下がり斜線はデータ (II) を表す。ノンパラメトリック多重比較法として Steel-Dwass 検定 [15,16] を行った結果、raw と variance1+データ (I), baseline と variance1+データ (II), baseline と variance2+データ (II), variance1+データ (I) と distance+データ (II) の間には有意な差 ($p < 0.05$)

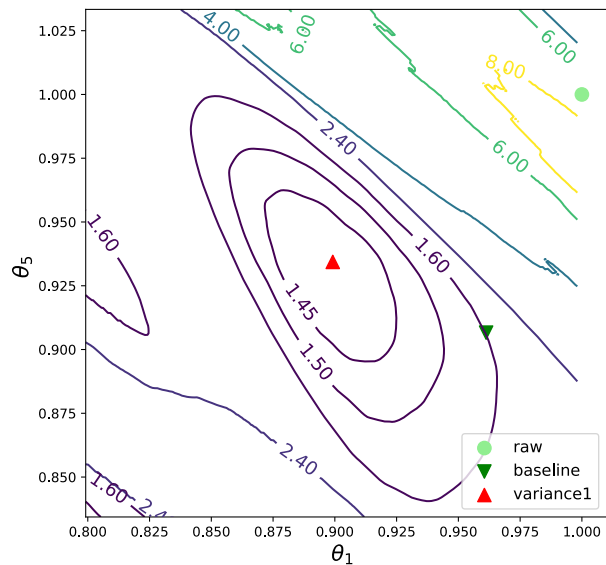


図 3 評価関数値の分布 ($E \times 10^3$)

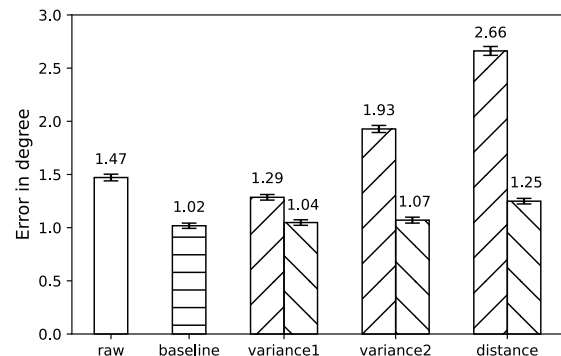


図 4 精度評価結果

が確認されず、それ以外のいずれの 2 組の間でも有意な差 ($p < 0.01$) が確認された。

5.3 考察

較正前の raw よりも提案手法である variance1+データ (II) の精度が向上し、基準である baseline と同等の精度 (約 1 deg) になった。この結果より、提案手法により自動較正が可能であることが確認された。また、variance1+データ (I) においては、distance よりも高精度な較正パラメータが推定された。この原因は、distance 条件では、マーカと注視点の距離が利用されるが、マーカを中心として対称に注視点分布しない場合は、偏った位置に視線が補正されると考えられる。提案手法では、マーカを使用していないためそのような問題がなかったものと考えられる。

variance1 のデータ (I) よりも (II) の方が精度が高

かった原因は、今回の実験データ (I) では視線方向が視界の中央に集まり過ぎており、較正パラメータが十分な精度で推定できなかったことが考えられる。新しい視線が生じるたびに定期的に提案手法により較正されれば、視線が集中することは特に問題にならないと考えられる。

今回シーンとして連続かつ滑らかな曲面を用いたが、不連続箇所を含むシーンでは単純な最適化方法では最適解が求められないことが想定できる。これについては今後の課題である。

6 おわりに

本稿では、顕著性マップとは異なる手がかりとして、固視中の3次元注視位置の分散を指標とした視線計測器の自動較正法を提案した。提案手法は、固視中の注視点は物体表面上で密集して存在すると仮定し、シーンに対して頭部が移動する状況下では、視線計測器の誤差により注視点群の分散が大きくなることを利用した手法であった。実験では、VR環境において連続でなめらかな曲面を対象に、提案手法の収束性能があることと、較正パラメータの推定精度が約1degであることを確認した。

今後は、不連続な箇所（オクルージョン）が存在するより複雑なシーン形状や、様々な較正モデルの利用可能性、較正に必要な時間などを明らかにする予定である。

謝辞 本研究の一部は科研費（19K22873）の支援による。本研究における被験者実験は、大阪大学大学院基礎工学研究科人を対象とした研究に関する倫理委員会の承認（R2-9）を得て実施した。

参考文献

- [1] D.W. Hansen and Qiang Ji. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010.
- [2] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike. Appearance-Based Gaze Estimation With Online Calibration From Mouse Operations. *IEEE Transactions on Human-Machine Systems*, 45(6):750–760, 2015.
- [3] Fares Alnajar, Theo Gevers, Roberto Valenti, and Sennay Ghebream. Auto-Calibrated Gaze Estimation Using Human Gaze Patterns. *International Journal of Computer Vision*, 124(2):223–236, 2017.
- [4] Y. Sugano, Y. Matsushita, and Y. Sato. Appearance-Based Gaze Estimation Using Visual Saliency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):329–341, 2013.
- [5] Peiteng Shi, Markus Billeter, and Elmar Eisenmann. SalientGaze: Saliency-based gaze correction in virtual reality. *Computers & Graphics*,

- 91:83–94, 2020.
- [6] M. Liu, Y. Li, and H. Liu. 3d gaze estimation for head-mounted eye tracking system with auto-calibration method. *IEEE Access*, 8:104207–104215, 2020.
- [7] Jian-Gang Wang, E. Sung, and R. Venkateswarlu. Eye Gaze Estimation from a Single Image of One Eye.
- [8] Takashi Nagamatsu, Yukina Iwamoto, Junzo Kamahara, Naoki Tanaka, and Michiya Yamamoto. Gaze Estimation Method Based on an Aspherical Model of the Cornea: Surface of Revolution about the Optical Axis of the Eye. page 255–258, 2010.
- [9] Carlos H. Morimoto and Marcio R.M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1):4–24, 2005.
- [10] O. Williams, A. Blake, and R. Cipolla. Sparse and Semi-supervised Visual Mapping with the S³GP. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:230–237, 2006.
- [11] E. Wood, T. Baltruaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. In *2015 IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.
- [12] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):162–175, 2019.
- [13] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, 2017.
- [14] Dario D. Salvucci and Joseph H. Goldberg. Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, page 71–78, 2000.
- [15] Robert G. D. Steel. A rank sum test for comparing all pairs of treatments. *Technometrics*, 2(2):197–207, 1960.
- [16] M. Dwass. Some k-sample rank-order tests. *Contributions to Probability and Statistics*, 1960.