

# 滑動性眼球運動を用いた視線計測の自動校正

松本 大輝<sup>1,a)</sup> 米谷 竜<sup>1,b)</sup> 佐藤 洋一<sup>1,c)</sup>

**概要:** 視線推定手法の多くは、ディスプレイ上の基準点をユーザに注視させることで視線計測時に発生する視線のずれを補正するキャリブレーションが事前に必要となる。本研究では、一般的な映像を閲覧する際の眼球運動から、キャリブレーションを自動的に実行する手法を提案する。提案手法では、滑動性眼球運動と呼ばれる映像中の動きに沿って視線を滑らかに動かす運動に着目し、目の動きと映像中の動きとの相関や、映像中の動きの顕著性を利用することで、キャリブレーションに利用可能な基準点を自動検出できる。本稿では ASCMN のデータセットを用いて提案手法の有効性を検証する。

## Automatic Calibration of Eye Tracking Systems using Smooth Pursuits

DAIKI MATSUMOTO<sup>1,a)</sup> RYO YONETANI<sup>1,b)</sup> YOICHI SATO<sup>1,c)</sup>

### 1. はじめに

視線推定とは、ユーザがディスプレイ中で目を向けている位置（注視点）を推定する技術であり、広告での視線誘導 [11]、自動車の安全運転支援 [7]、注視に基づくインターフェース [6] といった様々な分野に応用されている。その基本的なアプローチは、ユーザがディスプレイに提示されたコンテンツを閲覧する際の眼球情報（瞳孔の位置や目の見え）をカメラで撮影し解析するというものであり、これまでに数々の手法が提案されている [3]。

多くの視線推定手法では、キャリブレーションと呼ばれる手続きが事前に必要となる。これは、ディスプレイに提示された複数の基準点を注視するようにユーザに教示し、その際の眼球情報を計測することで視線計測時に発生する視線のずれを補正する手続きである。この手続きはユーザごとに必要となるため、ユーザが頻繁に切り替わる状況（例：街中の電子広告）への適用が難しい。

そこで本研究では、ディスプレイに提示された映像を閲覧する際の眼球運動を用いて自動的にキャリブレーション

を行う手法を提案する。提案手法では、滑動性眼球運動と呼ばれる動きに沿って視線を滑らかに動かす運動に着目し、映像中で目をひきつけやすく、かつその動きが眼球運動と相関するような領域を推定する。この領域を前述の基準点として利用することで、ユーザに特別な教示を与えることなくキャリブレーションを行うことができる。

### 2. 関連研究

映像中の動きと眼球運動との相関を用いた注視領域推定やキャリブレーションの手法として [10], [17] が挙げられる。米谷ら [17] は、提示内容が時間的に変化するコンテンツを閲覧する状況において、コンテンツの動きと眼球運動との相関および同期の程度を分析することで注視オブジェクトを推定している。Pfeuffer ら [10] は、一部のオブジェクトが動くような映像を閲覧する状況において、オブジェクトの動きと眼球運動との相関から注視オブジェクトを推定しそれを基にキャリブレーションを行っている。しかし、これらの手法では提案手法が適用できるような映像を新たにデザインする必要がある。

一方、人間は他の領域に比べ顕著な領域を注視しやすいため、顕著性を用いることは注視領域を推定する上で有用である。顕著性を用いたキャリブレーション手法として [1], [14] が挙げられる。菅野ら [14] は、顕著性マップ [5] と呼ばれる、映像中の各位置がどの程度視覚的注意を向け

<sup>1</sup> 東京大学  
The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo,  
153-8505, Japan

a) d-matumo@iis.u-tokyo.ac.jp

b) yonetani@iis.u-tokyo.ac.jp

c) ysato@iis.u-tokyo.ac.jp

られやすいかを数値で表現したマップを用いて確率的に注視点を推定している。Chen ら [1] は、三次元眼球モデルを用いた視線推定において顕著性マップを利用している。

### 3. 提案手法

#### 3.1 問題設定

本研究では入力として、映像とそれを閲覧する際の未校正の注視点データを用いる。ここでの未校正とは、真の注視点データに対してスケールとバイアスがかかっているような状況とし、回転などの非線形変換はされていないものとする。また注視点データは、ディスプレイ上の2次元点系列を想定しており、たとえば事前に獲得された他ユーザのキャリブレーション結果を用いた視線推定によって得ることが可能である。そして、キャリブレーションとは、未校正の注視点データから真の注視点データへの変換パラメータを推定する問題である。

特に本研究では、入力映像として複数の領域が同じ動きをしよう一般的な映像、例えば、監視カメラの映像のように複数の人物が同一方向に歩行する映像や、ホームビデオのようにあるイベントをカメラのパン・チルトを行いながら撮影した映像にも適用可能な枠組みの提案を目指す。このとき、キャリブレーション問題を解くにあたって、以下の3つの課題がある。

##### I. 注視領域候補をどう見つけるか

従来研究 [10], [17] ではあらかじめ映像をデザインするため映像中のオブジェクトの動きは既知であった。これに対して、本研究では既存の映像を用いるため映像中にどのような動きが現れるかを推定する必要がある。

##### II. 注視領域候補をどう絞り込むか

複数の人物が同一方向に歩行するといった複数の領域が類似する動きをする場合、眼球運動との相関のみを用いて注視領域を一意に定めることは難しい。

##### III. 滑動性眼球運動の区間をどう見つけるか

映像中の動きと眼球運動との相関を求める際には、人が実際に動くオブジェクトを追視している区間、つまり滑動性眼球運動を行っている区間を検出し、その区間内で相関を求める必要がある。滑動性眼球運動は、従来手法 [8], [9] ではまずサッケードと呼ばれる高速な眼球運動を検出し、次に残りのフレームに対して滑動性眼球運動であるか固視 (fixation) と呼ばれる一定の領域内に視線が留まる運動であるかを判定することによって検出される。つまり、サッケードを正しく検出できなかった場合、例えば本来は長い滑動性眼球運動の区間であったものが間にサッケードが誤って検出されてしまい短く分断されるといったように、滑動性眼球運動の検出にも影響が出る。また、サッケードの検出においては、従来手法 [8] では隣接フレーム間の速度といったように、何らかの尺度に対して閾値を設定して検出する手法が用いられることが多い。この閾値は真の注

視点データに対して正解ラベルがつけられた学習データに基づいて決められる。しかし、本研究のように入力として未校正の注視点データを用いる場合、ユーザごとに真の注視点データとのずれの程度が異なるため、このような学習ベースの手法を用いて閾値を一意に定めることは難しい。

これらの課題を踏まえ、提案手法の流れ (図 1) を説明する。まず、入力の映像から多数の特徴点を抽出し、それらを追跡することによって注視領域候補を生成する (図 1(a))。これによって一つ目の課題を解決することができる。次に、サッケードを検出する際の閾値を複数個設定し、各閾値を用いて入力の注視点データを複数の時区間に分割することによって、複数の視線の分割候補を持つような視線の階層構造を生成する (図 1(b))。これによって三つ目の課題を解決することができる。得られた階層構造の各分節において、その分節内に存在する注視領域候補に対して視線の動きとの相関およびその候補の動きの顕著性に基づき注視領域らしさを求める (図 1(c))。顕著性も考慮することで、注視領域候補を人が視覚的注意を向けやすい領域のみに絞ることができ、二つ目の課題を解決することができる。例えば、カメラのパン・チルトが含まれる映像の場合、背景領域は眼球運動との相関は高くなるものの顕著性は低くなるため候補から除外することができる。そして、生成された階層構造においてどの分節のどの注視領域候補を選べばよいかという問題を整数計画問題に定式化し、それを最適化することによって注視領域を推定する (図 1(d))。最後に、推定した注視領域を用いてキャリブレーションを行う (図 1(e))。また、キャリブレーションは水平方向、垂直方向それぞれ独立に行う。それぞれのステップに対して詳しく説明する。

#### 3.2 注視領域候補の生成

映像中にどのような動きが現れるかを推定するため、注視領域候補の生成には dense trajectory [15] を用いる。すなわち、映像中に現れる多数の特徴点を [13] によって抽出し、オプティカルフロー [2] と呼ばれる映像中の各画素が時間的にどのように動いているかを求めたものを用いてその特徴点を長時間追跡したものを注視領域候補とする。

図 2 に dense trajectory の例を示す。図 2 のように人間や自動車といった動く領域に対してその動きを追跡できていることが分かる。

#### 3.3 視線の階層構造の生成

入力の注視点データから滑動性眼球運動の区間の候補を多数生成するため、本研究では多重解像度解析の枠組み [16] を用いて視線の階層構造を生成する。まず、 $t$  フレーム目の注視位置を  $(x_t, y_t)$  としたときに、隣接フレーム間の視線の移動速度  $v_t = \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}$  を計算する。得られた速度信号  $\mathbf{v} = (v_1, v_2, \dots, v_N)$  を視線の階層構

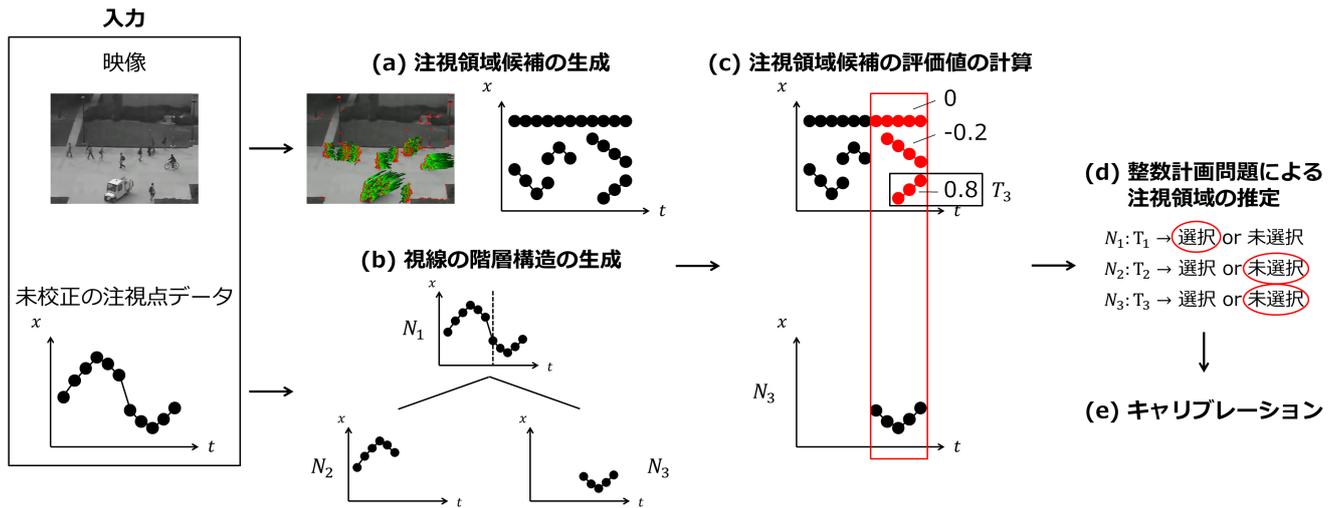


図 1 提案手法の流れ.  
Fig. 1 Proposed framework.

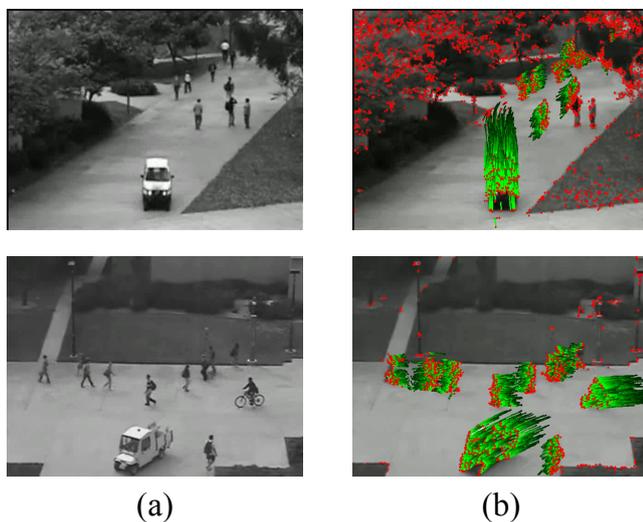


図 2 (a) 映像の 1 フレーム. (b) Dense trajectory. 赤点は特徴点, 緑線はその特徴点の軌跡を表している.  
Fig. 2 (a) Some frames from video clips. (b) Dense trajectories. Red points represent feature points and green lines represent their trajectories.

造を生成する際の入力信号とする. 信号  $v$  に対してスケール  $\sigma_1, \sigma_2, \dots, \sigma_K (\sigma_k < \sigma_{k+1})$  のガウス関数を畳み込むことによって, 信号  $v$  のスケール空間を構築することができる. このとき, 得られた波形集合においてスケールを変化させながら波形の 1 次微分 (ここでは視線の加速度にあたる) のゼロ交差 (変曲点) を追跡することで, ガウス関数の因果性により, スケールが小さくなるにつれて新たな変曲点が現れるような階層構造が形成される. ここでは,  $\sigma_k \rightarrow \sigma_{k-1} (k = 2, 3, \dots, K)$  のスケールの変化により必ず新たな変曲点が現れ, スケール  $\sigma_K$  では変曲点を持たないようにスケールを設定する. そして, 各スケールにおいて新たに現れた変曲点を追跡したときに得られるスケール  $\sigma_1$  での変曲点をそのスケールにおける視線の分割点とする.

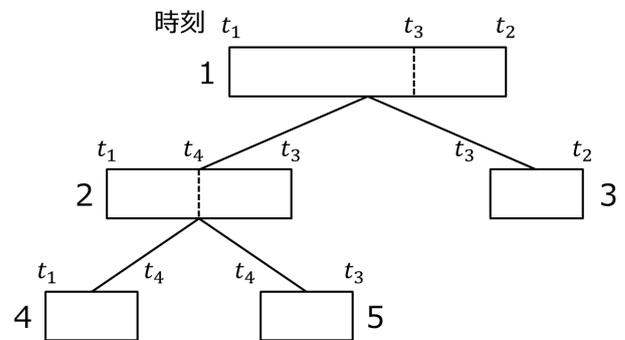


図 3 視線の階層構造の例  
Fig. 3 An example of the hierarchical structure of gaze.

これにより, スケールが小さくなるにつれてより細かく分節化されるような視線の階層構造 (図 3) が得られる.

また, 滑動性眼球運動に注目するため, 明らかに固視を行っている分節を階層構造から除外する. ここでは, 各分節内における注視点がある一定の領域内に収まっている場合, 具体的には注視点の水平方向と垂直方向それぞれの最大値と最小値の差分の二乗和の平方根 ( $D = \sqrt{\{\max(x) - \min(x)\}^2 + \{\max(y) - \min(y)\}^2}$ ) が閾値以下の場合に固視と判定する.

### 3.4 注視領域候補の評価値の計算

3.3 節で得られた階層構造の各分節において, その分節内に存在する注視領域候補に対して注視領域らしさを表す評価値を計算する. 注視領域は眼球運動との相関と自身の顕著性がともに高いと考えられるため, この評価値は注視領域候補の動きと眼球運動との相関と, 注視領域候補の動きの顕著性を掛け合わせたものと定義する. 相関は正規化相互相関とし, 顕著性は注視領域候補の各フレームでの位置の顕著性マップの値を平均したものとする. よって, 視

線を  $\{x_i, y_i\}$ , 注視領域候補を  $\{X_i, Y_i\}$ , フレーム  $t$ , 位置  $(x, y)$  での顕著性マップの値を  $S_t(x, y)$  とすると, 評価値  $(V_x, V_y)$  は次のように表される.

$$V_x = \frac{\sum_i (x_i - \bar{x})(X_i - \bar{X})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (X_i - \bar{X})^2}} \times \frac{1}{N} \sum_i S_t(X_i, Y_i) \quad (1)$$

$$V_y = \frac{\sum_i (y_i - \bar{y})(Y_i - \bar{Y})}{\sqrt{\sum_i (y_i - \bar{y})^2} \sqrt{\sum_i (Y_i - \bar{Y})^2}} \times \frac{1}{N} \sum_i S_t(X_i, Y_i) \quad (2)$$

### 3.5 整数計画問題による注視領域の推定

これまでの手続きで, 入力注視点データは階層的に分節化されており, 各分節には複数の注視領域候補がある. そして, 各注視領域候補にはその分節の視線との相関および自身の顕著性に基づいて注視領域らしさを表す評価値が与えられている. キャリブレーションのためには, ここから注視領域らしい候補を選ぶ必要がある. ただし, その候補を選ぶ上で各時刻においてユーザの視線は一つに向いていて複数に向くことはないという制約を満たしている必要がある.

これを最適化するために整数計画問題で定式化する. 基本的なアイデアは, 選んだ注視領域候補の評価値の和が最大となるような選び方をするというものである. ここで, 各時刻においてユーザの視線は一つに向いていて複数に向くことはないということを考慮するために, 階層構造においてある分節の注視領域候補が選ばれたときに, その分節よりも下の層にある分節からは注視領域候補を選ばないといった制約を入れる. また, 選んだ注視領域候補の評価値の和が最大となるような選び方をした場合, 基本的に多くの分節を選んだ方が評価値の和が大きくなるため, 階層構造において多く分節化された層が選ばれやすくなり, 選ばれる分節に偏りが生じる. そこで, どの分節も均等に選ばれるようにするために, 階層構造において上の層にある分節であるほど注視領域候補の評価値が大きくなるような重みを加える. ここでは, 各分節が自身の下の層で最終的に何個に分節化されているかを調べ, その個数をその分節の重みとする.

まず, ある  $i$  番目の分節が選ばれるかどうかを  $p_i \in \{0, 1\}$  で定義する. そして, 変数  $w_i$  を時区間  $i$  における注視領域候補の評価値の最大値と定義すると, 例えば視線の階層構造が図 3 のように表される場合には次式のような整数計画問題で定式化することができる.

$$\max_{p \in \{0,1\}} \mathbf{w}^T \mathbf{p} \quad \text{s.t. } C\mathbf{p} = \mathbf{1} \quad (3)$$

$$\mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix}, \mathbf{w} = \begin{pmatrix} 3w_1 \\ 2w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

図 3 において分節 1 は下の層で分節 3, 4, 5 の 3 個の分節

に分節化されているため, 評価値  $w_1$  に対して重み 3 がかけられている. 同様にして, 評価値  $w_2$  に対しては重み 2 がかけられている. また, 制約条件では, 選ばれるフレームが分節間で重複しない, つまり図 3 では分節の選び方が  $\{1\}, \{2, 3\}, \{3, 4, 5\}$  のみになるように行列  $C$  を設定する. この整数計画問題を解くことによって注視領域を推定する.

### 3.6 キャリブレーション

3.5 節で推定した注視領域とその時刻における注視点データの組を用いてキャリブレーションを行う. 本研究では, 入力注視点データはスケールとバイアスが未校正であるという前提を置いているため, キャリブレーションでは真の注視点データに対してかけられたスケールとバイアスの変換パラメータを最小二乗法を用いて推定する. ここでは, 評価値が高いほどその注視領域が注視されている可能性が高いと仮定し, 評価値を重みとした重み付き最小二乗法を用いる. つまり, 注視点データを  $\{x_i, y_i\}$ , 注視領域を  $\{X_i, Y_i\}$ , 注視領域の評価値を  $\{V_{xi}, V_{yi}\}$  とすると, 変換パラメータ  $(a_x, b_x, a_y, b_y)$  は次式の最小化問題を解くことによって得られる.

$$E_x = \sum_i V_{xi}(X_i - a_x x_i - b_x)^2 \quad (4)$$

$$E_y = \sum_i V_{yi}(Y_i - a_y y_i - b_y)^2 \quad (5)$$

## 4. 実験

### 4.1 提案手法を用いたキャリブレーション

提案手法の有効性を確かめるため, ASCMN データセット [12] から 14 種類の映像<sup>\*1</sup> (図 4) を用いて, キャリブレーションの精度評価を行った. このデータセットには, 図 5 のような人が密集しているような映像もあるが, このような映像は人が視覚的注意を向けやすい領域が画面全体にあり提案手法の適用が難しいため, 本実験では除外した. また, このデータセットには各映像に対して 10 人分のキャリブレーション済みの注視点データがあるため, 本実験では入力注視点データとして, キャリブレーション済みの注視点データに対して線形変換を施し, 擬似的に未校正の注視点データを生成した. このとき, 真の注視点データ  $(p_x, p_y)$  と入力注視点データ  $(p'_x, p'_y)$  の関係は  $p'_x = ap_x + b, p'_y = ap_y + b$  と表され, この係数  $a$  と  $b$  をキャリブレーションによって推定する. 今回の実験では,  $(a, b) = (2, 200)$  とした. 精度評価は, 推定された係数を基に入力注視点データから再現したものと真の注視点データとの平方平均二乗誤差 (RMSE) を各映像のフレームサイズに対する割合に換算した値に基づいて行った.

\*1 フレームレートは 15fps, 10 秒程度の短い映像が 6 種類, 30 秒から 1 分程度の長い映像が 8 種類.



図 4 実験で用いた映像の例.

Fig. 4 Examples of video clips used in the experiment.



図 5 実験で用いなかった映像の例.

Fig. 5 Examples of video clips not used in the experiment.

#### 4.2 比較手法

提案手法との比較手法として以下の 2 種類の手法を用いた。

比較手法 1 注視領域候補の評価値が眼球運動との相関のみの場合。

比較手法 2 特定のスケールで注視点データを分節化した場合。

提案手法と比較手法 1 を比較することで顕著性を用いることの有効性を検証し、提案手法と比較手法 2 を比較することで視線の階層構造を用いることの有効性を検証した。

#### 4.3 実装

3.3 節での視線の階層構造の各分節が固視であるかの判定は、入力の注視点データを平均 0, 分散 1 になるように正規化した状態で行い、判定の際の閾値は 0.3 とした。また、3.4 節での顕著性マップの生成には Harel らの手法 [4] を用いた。

#### 4.4 実験結果と考察

図 6 に推定精度の比較結果を示す。また、比較手法 2 での結果は、精度が最も良かった場合と最も悪かった場合をデータごとに手動で選択し、それぞれを平均した結果を示

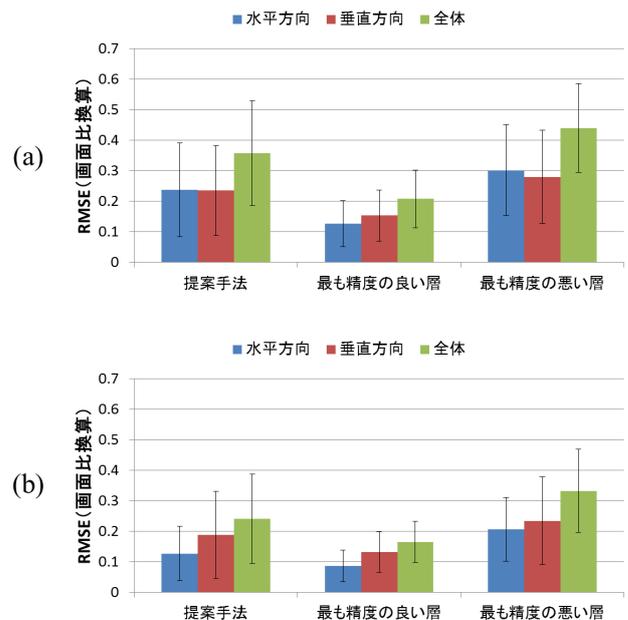


図 6 各手法でのキャリブレーション結果。(a) 評価値が相関のみの場合での結果。(b) 評価値が相関と顕著性を掛け合わせたもの場合での結果。

Fig. 6 Calibration results. (a) Results when using only the correlation as the score. (b) Results when using the correlation and saliency as the score.

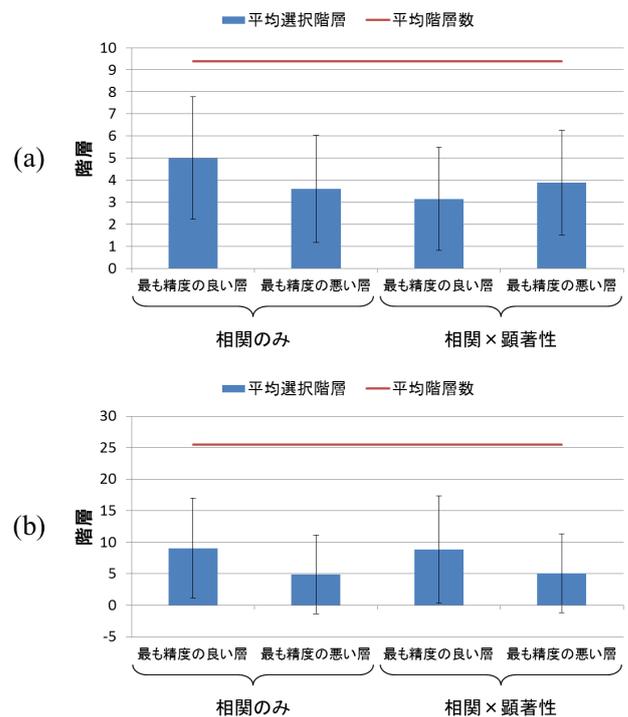


図 7 比較手法 2 を用いた場合の視線の階層構造における平均選択階層。(a) 短い映像での結果。(b) 長い映像での結果。

Fig. 7 The average number of the layer in the hierarchical structure of gaze selected when using the comparative method 2. (a) Results of short video clips. (b) Results of long video clips.

している。図 6 から比較手法 1 よりも提案手法の方が精度が良いことが分かり、顕著性を用いることの有効性が確認できる。また、視線の階層構造において最も精度の良い層を選んだ場合の結果が提案手法よりも良いものの、図 7 に示すように最も精度の良い層が何層目にあるかはユーザごとに大きなばらつきがあることが分かる。よって、単一のスケールで視線を分割するよりも複数のスケールで視線の階層構造を生成する方が良いことが分かる。

また、図 8 に具体的なキャリブレーション結果の一例を示す。図 8 の上の結果は、正しくキャリブレーションできた結果を表しており、映像中央にある自動車を注視領域として推定できていることが分かる。また、図 8 の下の結果は、キャリブレーションに失敗した結果を表しており、印をつけた部分のように真の視線に比べ速度が全く異なる注視領域が選ばれていることが分かる。これは、注視領域候補の評価値で用いている眼球運動との相関は正規化相互相関であるため、注視領域候補の動きと眼球運動の方向が合っていれば相関が高くなり全体の評価値が高くなるのが原因であると考えられる。よって、このような動きの速度も考慮するように注視領域候補の選び方を工夫する必要がある。これについては今後の検討課題とする。

## 5. 結論

本研究では、視線計測におけるキャリブレーションの自動化を目的として、眼球運動と映像中の動きとの相関および映像中の動きの顕著性から注視領域を推定しキャリブレーションを行う手法を提案した。また、擬似的ではあるが未校正の注視点データに対して提案手法を用いてキャリブレーションを行い、その有効性を実証した。

今後は実際にユーザが映像を閲覧する際の未校正の注視点データを集め、提案手法を用いてキャリブレーションを行うことで実データでの提案手法の有効性を検証する。

謝辞 本研究の一部は、JST CREST および文部科学省科研費 15K16018 の助成により行った。

## 参考文献

- [1] Chen, J. and Ji, Q.: Probabilistic Gaze Estimation Without Active Personal Calibration, *Computer Vision and Pattern Recognition*, pp. 609–616 (2011).
- [2] Farnebäck, G.: Two-Frame Motion Estimation Based on Polynomial Expansion, *Proceedings of the 13th Scandinavian Conference on Image Analysis*, pp. 363–370 (2003).
- [3] Hansen, D. and Ji, Q.: In the Eye of the Beholder: A Survey of Models for Eyes and Gaze, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 3, pp. 478–500 (2010).
- [4] Harel, J., Koch, C. and Perona, P.: Graph-based visual saliency, *Advances in Neural Information Processing Systems*, Vol. 19, pp. 545–552 (2006).
- [5] Itti, L., Koch, C. and Niebur, E.: A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 11, pp. 1254–1259 (1998).
- [6] Jacob, R.: The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get, *ACM Transactions on Information Systems*, Vol. 9, No. 3, pp. 152–169 (1991).
- [7] Ji, Q. and Yang, X.: Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance, *Real-Time Imaging*, Vol. 8, No. 5, pp. 357–377 (2002).
- [8] Komogortsev, O. and Karpov, A.: Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades, *Behavior Research Methods*, Vol. 45, No. 1, pp. 203–215 (2013).
- [9] Larsson, L., Nyström, M., Andersson, R. and Stridh, M.: Detection of fixations and smooth pursuit movements in high-speed eye-tracking data, *Biomedical Signal Processing and Control*, Vol. 18, pp. 145–152 (2015).
- [10] Pfeuffer, K., Vidal, M., Turner, J., Bulling, A. and Gellersen, H.: Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible, *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pp. 261–270 (2013).
- [11] Rayner, K., Rotello, C. and Stewart, A.: Integrating Text and Pictorial Information: Eye Movements When Looking at Print Advertisements, *Journal of Experimental Psychology*, Vol. 7, No. 3, pp. 219–226 (2001).
- [12] Riche, N., Mancas, M., Culibrk, D., Crnojevic, V., Gosselin, B. and Dutoit, T.: Dynamic saliency models and human attention: a comparative study on videos, *Proceedings of the 11th Asian Conference on Computer Vision*, No. 586–598 (2012).
- [13] Shi, J. and Tomasi, C.: Good features to track, *Computer Vision and Pattern Recognition*, pp. 593–600 (1994).
- [14] Sugano, Y., Matsushita, Y. and Sato, Y.: Calibration-free gaze sensing using saliency maps, *Computer Vision and Pattern Recognition*, pp. 2667–2674 (2010).
- [15] Wang, H., Klaser, A., Schmid, C. and Liu, C.-L.: Action Recognition by Dense Trajectories, *Computer Vision and Pattern Recognition*, pp. 3169–3176 (2011).
- [16] Witkin, A.: Scale-space Filtering, *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, No. 1019–1022 (1983).
- [17] Yonetani, R., Kawashima, H., Hirayama, T. and Matsuyama, T.: Gaze Probing: Event-Based Estimation of Objects Being Focused On, *IEEE International Conference on Pattern Recognition*, pp. 101–104 (2010).

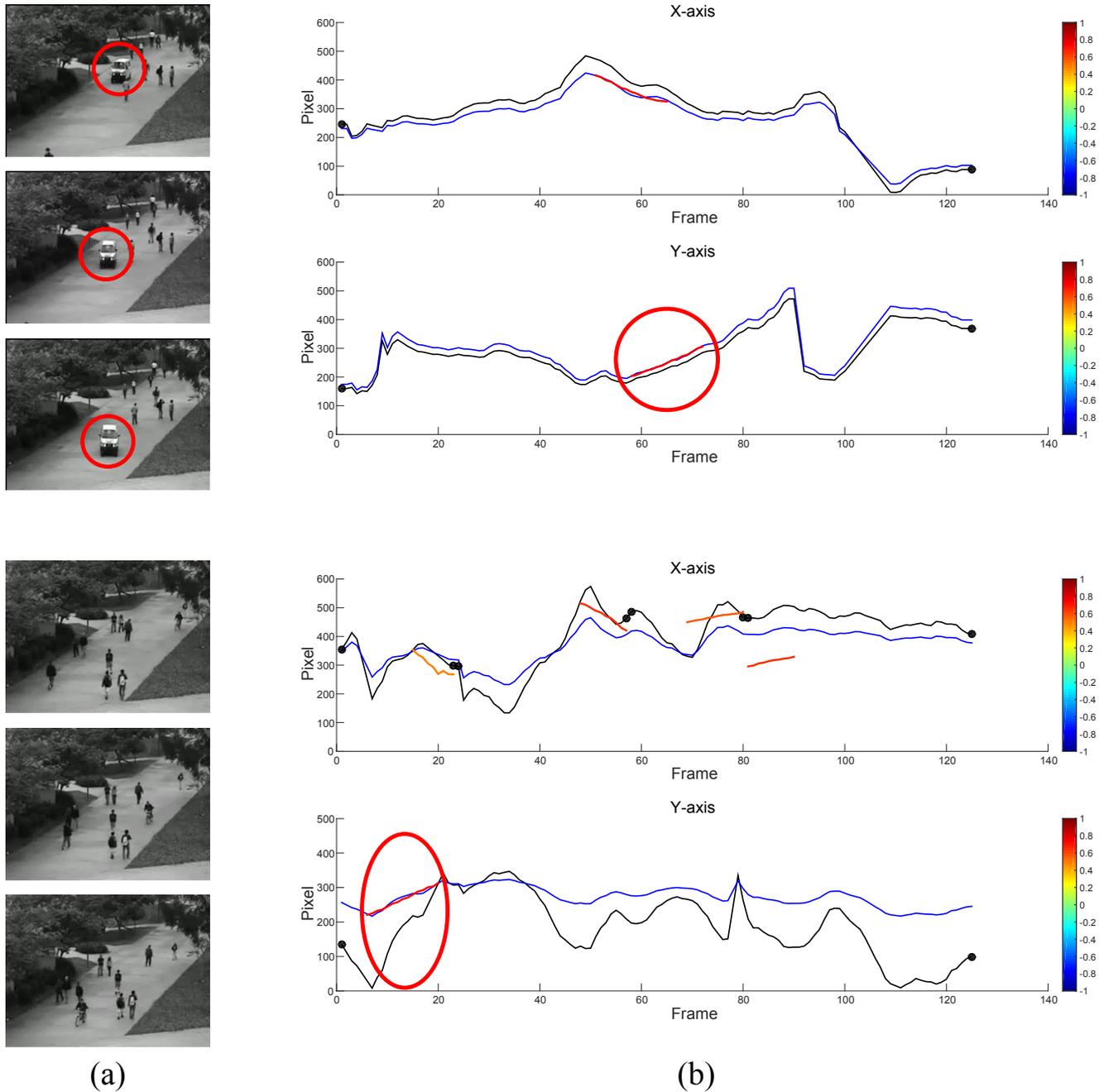


図 8 ユーザごとのキャリブレーション結果。(a) 映像の数フレーム。(b) キャリブレーション結果。黒線が真の注視点データ、青線は推定した変換パラメタを基に再現した注視点データ、色のついた短線はキャリブレーションに用いた注視領域を表している。また、黒点に囲まれた時区間が整数計画問題によって選ばれた分節を表している。

Fig. 8 Calibration results of each user. (a) Some frames from video clips. (b) Calibration results. Black lines represent the original gaze data and blue lines represent the estimated gaze data and colored short lines represent the gaze areas used in calibration. The frames surrounded by black spots represent the time sections selected by solving the integer programming problem.