

博士学位論文

奥行き情報を利用した顔・髪を検出・追跡の高速化  
と安定化に関する研究

平成 27 年 9 月

和歌山大学大学院システム工学研究科

鈴木 一正

doctoral thesis

# **Fast and Stable Face-Hair Detection and Tracking Using Depth Information**

September 2015

Graduate School of Systems Engineering  
Wakayama University

Kazumasa Suzuki

## 論文概要

画像中の人の顔を検出・追跡する技術は、コンピュータビジョンの中でも重要な研究分野であり、盛んに研究されている。髪領域に関する研究報告は少ないが、髪領域は個人を特徴付ける重要な要素であり、個人識別の特徴として用いたり、髪型を利用したアプリケーションなどの応用が考えられる。また、現代社会のニーズに答える様々な応用システムを開発するために、高速かつ安定なアルゴリズムが要求される。

本論文では、奥行き情報を利用した顔・髪領域やこれらを合わせた頭部領域の検出・追跡システムを提案する。高速かつ安定なシステムを構築するため、奥行き情報を活用することで高速化された顔検出と安定化された追跡手法を提案して組み合わせる。

画像から顔検出を行う場合、位置や大きさが未知の顔を検出するためピラミッドスキャンを行う必要があり、ビデオレートでの処理は難しい。また、ステレオカメラで顔検出を行った場合、複数視点の画像により検出精度の向上が期待できるが、ピラミッドスキャンでは処理コストが倍増する。本論文では、ステレオ画像から得られるスパースな奥行き情報を利用した識別回数の削減によって顔検出の高速化を行う手法を提案する。提案手法では、ステレオカメラを用いることで、検出精度を向上させながら単眼カメラよりも高速なビデオレートでの顔検出を実現する。

顔追跡システムは、改良した色弁別度追跡法と提案する顔検出の組み合わせで構築される。色弁別度による顔検出・追跡システムは、色ベースで顔の検出・追跡を行っているため、予め対象色と背景色を教示する必要があるが、肌色と似た色の物体を誤って検出してしまう。本論文では、顔識別器により顔だけを正確に検出し、検出された顔領域内外の情報を基に色のプロトタイプを自動で選択し、弁別度 LUT を構築する手法を提案する。また、追跡中にも顔検出を行い、対象や背景色の情報を更新することによって周辺環境の変化に対する頑健性を向上させる。さらに、対象までの奥行き情報を用いた制約により顔追跡の安定化を実現する。

髪追跡システムは、改良した K-means Tracker と提案する顔検出の組み合わせで構築される。K-means Tracker は、画素の位置と色情報から構成される 5 次元特徴空間内にて K-means クラスタリングを行い、画像中の画素を対象か背景に分類することで追跡を行う手法である。本論文では、奥行き情報を追加した 6D K-means Tracker を提案し、3次元化された位置情報によって前景と背景の特徴同士の分離性能を高めることで追跡の安定化を行い、色彩情報の乏しく形状変化が大きい髪領域の追跡を実現する。

## Abstract

This paper describe a working system that can detect human head stably and fast. The methods of face detection, face-hair tracking that makes use of depth information effectively are proposed in order to achieve stable and real time performance. These method are combined efficiently to realize a high performance head tracking system.

In general, face detection is carried out through the pyramid search in order to detect faces of any sizes. The computation cost of this approach is too high for video rate processing. There were also methods that used stereo camera to detect faces. Although the improvement of the detection accuracy can be expected with the help of depth information, the face detection in two images and the additional stereo processing make the computation cost more than twice as the case of using a single camera. This paper proposes a fast and stable method for face detection using a stereo camera. We realizes video rate face detection by using the depth information to reduce the number of the classifications significantly. It is not only faster but also gives more accurate detection results than the case of using a single camera.

Our face tracking system is constructed by combining an Improved Color Distinctiveness Tracker and our face detection method. Our method uses face classifier to detect faces in video images, and uses both the inside and outside pixels of the detected face region to train the target and background colors automatically for tracking faces. Moreover, we improve the robustness of the tracking system to the changes in environments by updating target and background colors according to the tracking results and adding the constraint that the 3D size of the same head should be constant to stabilize the face tracking by making use of the depth information.

Hair tracking system consists of the Improved K-means Tracker and our face detection method. K-means Tracker is a method of object tracking by performing K-means clustering in 5D feature space for describing both color and position of a pixel to separate the target and the background. In this paper, we propose a new 6D K-means Tracker by adding the depth as the sixth dimension to the feature to make the position of pixels described in real 3D space, which make the separation of the target and the background more reliable thus improve the stableness and the correctness of the hair tracking significantly.

# 目次

第 1 章	はじめに	1
1.1	研究の背景・目的	1
1.2	従来手法	3
1.2.1	顔検出	3
1.2.2	髪検出	6
1.2.3	対象追跡	8
1.3	提案手法の概要	13
1.4	論文の構成	16
第 2 章	奥行き情報を用いた顔検出の高速化	17
2.1	ステレオカメラ	18
2.2	ステレオカメラを用いた高速顔検出法	22
2.2.1	スパースサンプリングによるステレオ処理の高速化	22
2.2.2	識別回数の削減による検出の高速化	24
2.2.3	ステレオ画像を用いた検出精度の向上	29
2.3	実験	31
2.3.1	実験環境	31
2.3.2	処理速度の実験結果	31
2.3.3	検出精度の実験結果	33
2.3.4	その他の検出結果	38
2.4	まとめ	43
第 3 章	顔検出と色弁別度追跡法を組み合わせた顔追跡システム	44
3.1	色弁別度追跡法	44
3.1.1	弁別性マップの構築	45

3.1.2	弁別度算出の高速化 . . . . .	46
3.1.3	弁別性マップを用いた対象追跡 . . . . .	49
3.1.4	検出・追跡システム . . . . .	50
3.1.5	従来手法のまとめと残っている問題点 . . . . .	51
3.2	適応的カラーモデルを用いた顔検出・追跡システム . . . . .	52
3.2.1	プロトタイプの自動登録による適応的カラーモデルの構築 . . . . .	53
3.2.2	奥行き情報と適応型カラーモデルを用いた追跡の安定化 . . . . .	55
3.2.3	提案システムの構成 . . . . .	56
3.3	実験 . . . . .	58
3.3.1	従来システムとの比較実験 . . . . .	58
3.3.2	追跡中の安定化処理の有無による比較 . . . . .	68
3.3.3	他の追跡手法との比較実験 . . . . .	71
3.4	まとめ . . . . .	76
第 4 章	顔検出と 6D K-means Tracker を組み合わせた髪追跡システム	77
4.1	K-means tracker . . . . .	78
4.1.1	K-means tracker の概要 . . . . .	78
4.1.2	K-means tracker の構成要素 . . . . .	79
4.1.3	K-means tracker の処理手順 . . . . .	82
4.1.4	まとめと 5 次元の K-means Tracker における問題点 . . . . .	84
4.2	FB-Clustering Tracker . . . . .	86
4.3	6D K-means Tracker . . . . .	87
4.3.1	RGB-D カメラ . . . . .	88
4.3.2	6 次元特徴空間でのクラスタリング . . . . .	89
4.3.3	クラスタ中心の更新 . . . . .	90
4.4	髪・顔・頭部領域の検出・追跡システム . . . . .	92
4.4.1	システム構成 . . . . .	92
4.4.2	ターゲットと非ターゲットクラスタ中心の自動初期化 . . . . .	92
4.5	実験 . . . . .	95
4.5.1	従来手法との比較実験 . . . . .	95
4.5.2	他の追跡手法との比較実験 . . . . .	103
4.5.3	様々な髪の追跡結果 . . . . .	108
4.5.4	頭部追跡 . . . . .	108

---

4.5.5	提案システムを用いた応用実験 . . . . .	113
4.6	まとめ . . . . .	117
第 5 章	おわりに . . . . .	118
5.1	本論文のまとめ . . . . .	118
5.2	今後の課題 . . . . .	120
参考文献		122
研究業績		133
付録 A	Viola-Jones 法による顔検出 . . . . .	136
A.1	Haar-like 特徴量 . . . . .	136
A.2	積分画像 . . . . .	137
A.3	カスケード構造 . . . . .	138
A.4	まとめ . . . . .	140
謝辞		141





# 第 1 章

## はじめに

### 1.1 研究の背景・目的

現在，人々の生活は機械に囲まれており，自動車や家電製品など日常的に利用するものから，インフラや産業ロボットなど社会を支えるものまで，あらゆる分野において人と機械は密接な関係を持っている．従来，人と機械の関係は一方向であることが多く，機械が提供する機能を人の判断によって操作する必要があったが，コンピュータの発展にともなって大量の情報処理ができるようになり，知能を持った機械が人を支援するようになってきた．機械が人を補助するためには，人の状態や周囲の環境を認識する必要があり，これらはコンピュータビジョンの技術によって行われている．

コンピュータビジョンとは，人が目で見ていることをコンピュータに行わせる，コンピュータに視覚を与えようという研究である．コンピュータにおける目はカメラであり，カメラから取得された画像を解析することで様々なことを認識・理解し，安心安全な社会づくりを実現している．例えば，デジタルカメラやスマートフォンのカメラに付いている顔検出機能，車に搭載されたカメラで障害物を認識することによって衝突防止などを行う運転支援システム，エアコンやテレビに搭載されたカメラを用いた人物検出による省エネのための機能など，これらは全てコンピュータビジョンの技術を利用しており，人々の生活を支援している．

本研究では，ステレオカメラあるいは Kinect から取得された画像中において，人の状態を認識する上で重要な顔・髪領域からなる頭部領域を決定することを目的とし，撮影された画像から顔を高速かつ高精度に検出できる方法を提案し，検出された顔領域に基づいて，ビデオレートで顔・髪・頭部領域を追跡するシステムを構築する．

コンピュータビジョンの分野では，顔画像処理が 1970 年代頃より始まり，80 年代後半

から 90 年代には盛んに研究されるようになった。90 年代後半にはセキュリティ応用を目的とした、より高速で安定した顔検出・認識手法が提案され、商品化されている。顔画像に関する研究は、顔認識、表情認識、年齢推定、男女判定など多岐にわたる。顔の位置や姿勢、表情など顔が持っている情報をコンピュータで読み取ることができれば、監視カメラやマシンインターフェース、ロボットとの対話など様々な応用システムが考えられる。これらの顔画像処理を行うためには、まず画像中から顔の領域を見つける必要がある。また、継続的に顔領域を得るために、顔検出・追跡は重要な処理であり、実用的なシステムを構築するために高速かつ安定な手法が求められる。一方、髪の色やスタイルは千差万別でありながら、画像内の対応画素には色情報やテクスチャ情報が少なく、モデル化や安定な特徴を抽出することが難しい。そのため、髪領域検出に関わる研究の報告例は少なく、ビデオレートでの追跡を行っている例は見当たらない。しかし、髪領域は重要な個人特徴であり、髪型を個人識別のための特徴として用いたり、様々な応用アプリケーションで顔と髪領域の関係を利用することが考えられる。また、顔領域だけの追跡では頭部とカメラの相対的な姿勢変化によって継続した追跡ができない場合でも、頭部の 360 度という全方位にある髪領域の追跡を行うことで頭部の姿勢に関わらない安定した追跡が可能となる。そこで本研究では、顔と髪をそれぞれ単独対象とした検出・追跡ができるシステムを構築すると同時に、顔と髪からなる頭部領域を対象とした検出・追跡システムを構築する。

単眼カメラから得られるカラー画像は、各ピクセルごとに赤 (Red)、緑 (Green)、青 (Blue) の情報を持っている。1990 年代頃からコンピュータやカメラの進歩に伴い、より多くの情報を解析することでシステムの精度や安定性を向上させるため、複数台からなるカメラシステム (ステレオカメラ) を用いた研究が盛んに行われてきた。ステレオカメラを用いることで、カラー画像に加えて奥行き (Depth) 情報を得ることができ、ロボットビジョンや自動車の運転支援システムといった製品にまで利用されている。また、近年、Kinect などの RGB-D カメラの出現によって、安価で高速に奥行き情報を取得することができるようになり、3 次元データを用いた研究が活性化している。しかし、情報量が増えるにつれ計算量も増えるため、専用のハードウェアを用いた高速化などが行われている。

本研究では、ステレオカメラや Kinect などの RGB-D カメラを用い、奥行き情報を利用することで、処理を安定化させるだけでなく、処理の高速化を行い、提案システムの検出や追跡処理の高速化、安定化を達成する。ここで、「検出」とは、1 枚の画像から対象領域の位置を探す処理であり、「追跡」とは、動画シーケンスに対して対象領域を時間シーケンス上で対応付けていく処理のことである。したがって、「追跡」を開始するためには、初期フレームにおいて、対象領域の初期位置を探す「検出」処理が必要である。提案シス

テムでは、検出・追跡を統合することで自動的に追跡を開始できるシステムを構築する。また一般的に、検出処理は1枚の画像から対象領域を抜き出すため処理コストが高いのに対し、追跡処理は前フレームの情報を利用するため処理コストが低い。本論文では、これら両方の処理に焦点をあて、奥行き情報を利用した、検出の高速化、追跡の安定化について議論する。

## 1.2 従来手法

コンピュータビジョン、ロボットビジョンにおいて対象の検出・追跡は基礎的な技術であり、多くの研究が報告されている。本節では、本研究で対象とする顔・髪・頭部領域に関わる検出手法や追跡手法について述べる。

### 1.2.1 顔検出

顔から読み取れる情報は多岐にわたり、性別、年齢、表情、姿勢、視線、そして個人IDなどがある。これら顔から得られる情報は、ロボットとの対話 [1]、ビデオ映像の検索キーとして利用 [2]、マシンインターフェース [3] など、人と機械とのコミュニケーションの高度化や、個人識別による入退室管理 [4]、ATM の認証 [5] など、セキュリティシステムの高度化へ応用されるなど利用価値が高く、顔画像に関する研究が盛んに行われている [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]。顔画像の研究は、顔の位置や大きさを得る「顔検出 (Face detection)」と様々な応用をするため顔の状態を認識する「顔認識 (Face recognition)」に大きく分類することができる。図 1.1 に顔画像処理技術に関する処理の例をあげる。顔検出は、顔画像処理を行う上で最初に行われる重要な処理であり、本論文ではこれを扱うため、ここでは顔検出についての動向を述べる。

1970 年ごろ、Sakai ら [17] や Kanade [18] によって顔検出に対する研究が報告された。初期の顔検出研究では、あらかじめ定義した顔モデルと画像から抽出した特徴を比較して検出するなど経験則に基づく手法が多く、エッジ [19, 20, 21, 22]、テクスチャや色 [23, 24, 25, 26]、顔のパーツの位置関係 [27] などの特徴量とした手法がある。これらの、人が経験的に作成する顔モデルでは、様々な環境に対応した複雑なモデルを作ることは難しく、汎用性の高い顔検出を行うことが困難であった。

1990 年代後半には、コンピュータの処理速度の向上にともなって、大量のデータと高次元の特徴量を処理できるようになってきたため、機械学習による手法 [28, 29, 30] が提案されるようになった。これらの手法では、顔と非顔にラベリングされた大量の画像を学習することで自動的に顔モデルを作成できるため、汎用性の高い識別器を構築することが

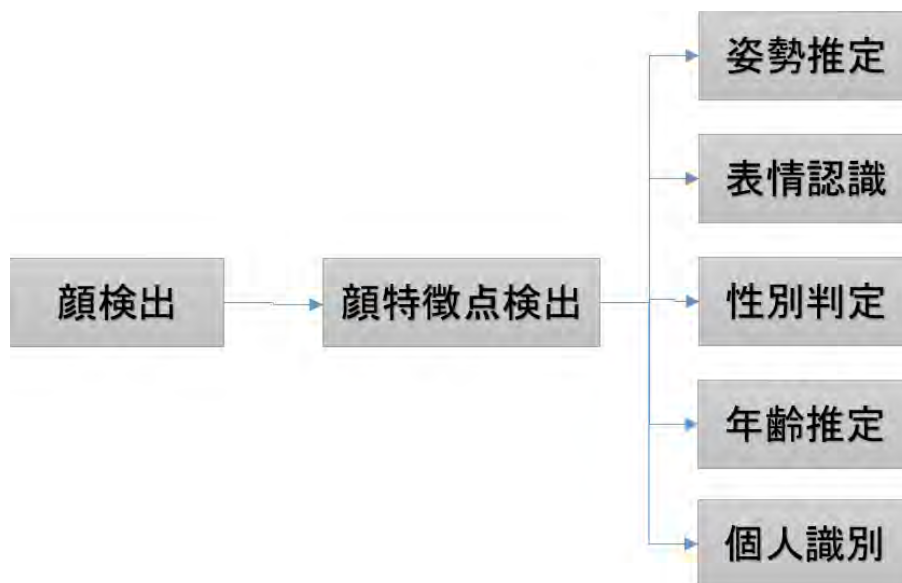


図 1.1 顔画像に関する処理

できる．図 1.2 は，Rowley ら [28] のニューラルネットワーク [31] を用いた顔検出手法の処理フローを表している．まず，画像中から切り出した矩形領域に対して照明環境などによる変動を抑制する処理を行い，これをニューラルネットワークによって構築された識別器に通すことで顔あるいは非顔の判定を行う．これらの検出手法は，基本的に同じような処理フローによって行われており，識別器の構築手段として様々な学習法が用いられている．学習法として他にも，部分空間法 [32]，SVM (Support Vector Machine) [33]，確率密度推定 [34]，AdaBoost[35] を用いた手法などがある．

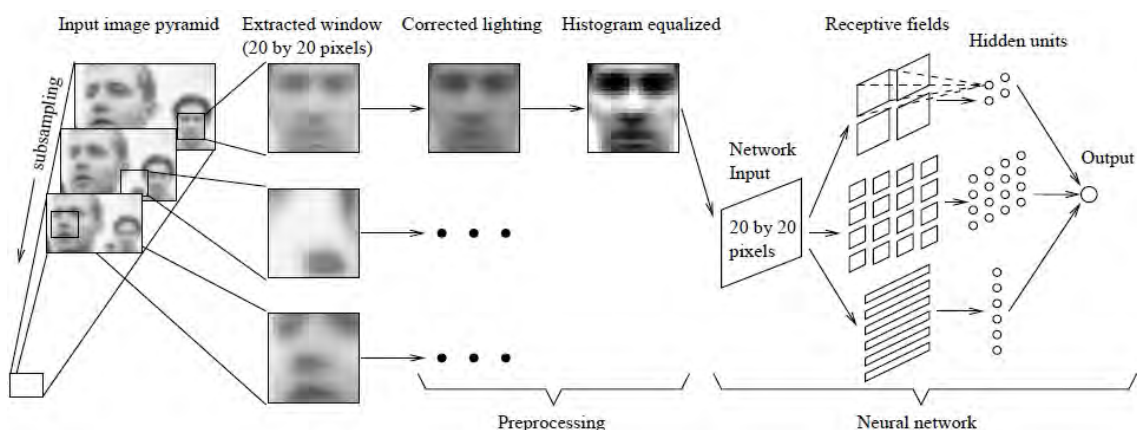


図 1.2 ニューラルネットワークを用いた顔検出例 [28]

中でも Viola ら [30, 36] が提案している AdaBoost で学習した強識別器をカスケードに接続した手法は、ビデオレートでの顔検出の可能性が初めて示され、その高速性と、汎化性能の高さから、単に学術的研究としてだけでなく、デジタルカメラやプリンタ、監視機器など工業製品への組み込みが行われるなど実用的側面からも注目を集めている。

顔検出の研究は識別器に関するものが多いが、中には色情報やフィルタを用いて大部分の背景を除去することで識別回数を削減し、既存の識別手法を高速化しているものもある。勞ら [37] は肌色を顔検出ではなく、明らかに顔が含まれない領域を除去する前処理として使うことで、顔検出の高速化を行っている。Shaick ら [38] は顔モデルを用いたフィルタ処理により画像をセグメンテーションし、背景領域を大幅に削減している (図 1.3)。このような、前処理で背景領域を削減する手法は、誤って顔領域も削減してしまう可能性がある。また、FPGA (field-programmable gate array) [39] や GPU[40] を用いた並列処理によって高速化している手法もあるが、特別なハードウェアが必要となる。

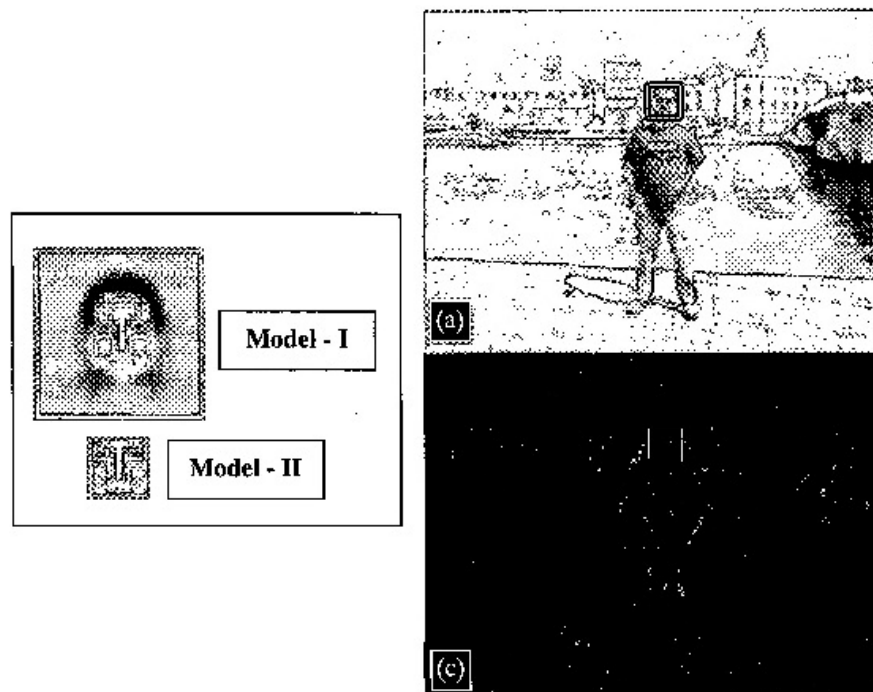


図 1.3 セグメンテーションによる背景領域の削減 [38] (左: 顔モデル, 右: セグメンテーション結果)

近年、奥行き情報を利用した顔検出や認識の研究 [41, 42] も行われており、Microsoft 社から Kinect 用の顔追跡ライブラリ (Face Tracking SDK[43, 44, 45]) も提供されている。R.I. Hg ら [41] は、Kinect を用いて顔の RGB-D データベースを構築し、顔検出シ

システムを構築している。T. Huynh ら [42] は, Kinect から得られた距離情報に基づいて, 顔表面の法線ベクトルを表現する LBP 型の記述子を提案し, 性別の識別を行った。これらは, 奥行き情報を利用した検出精度や認識精度の改善を行っている。

### 1.2.2 髪検出

髪の色やスタイルは個人を特徴付ける重要な要素である。髪領域を顔画像から検出, 追跡できれば, 髪型を利用した男女の識別 [46] や, 人物認証 [47], 髪型を用いたアプリケーション [48] などの応用事例において, 前処理として利用したり, 処理の簡素化, 高速化が期待できる。一方, 髪の色やスタイルは千差万別であり, モデル化が難しい。また, 画像内の対応画素の色やテクスチャの情報が乏しいため, 検出・識別に有効かつ安定な特徴を抽出することが難しい。そのため, 現状のコンピュータビジョンの研究分野では, 顔と比べると髪領域の自動検出に関する研究は少なく, 髪領域をビデオレートで追跡する例は見当たらない。

これまで提案されている髪領域を検出する手法で, 代表的なものを以下にあげる。Y.Yacoob ら [47] は, 顔検出と目検出で得られた幾何学的な位置関係から肌の色と髪の色を取得してモデル化し, 髪色モデルと類似した画素を連結することで髪領域を検出している。K.Lee ら [49] は, 色と画素位置の情報をを用いた, Graph-Cut[50, 51] や belief propagation[52] による顔・髪・背景領域のセグメンテーションを行っている。P.Julian ら [53] は単純な統計髪型モデルを用いて頭部の髪領域を表現し, active contour model[54] と active shape model[55] の組み合わせによって髪の初期領域を検出している。髪の色とテクスチャをこの初期領域から appearance パラメータとして学習し, 得られたパラメータで髪の領域が分割される。図 1.4 に髪モデルを用いた髪領域のセグメンテーション例を示す。これらの手法では, 髪の形状に基づいたモデルを定義しているため正面顔でのセグメンテーションにのみ対応しており, テクスチャ情報を得るために高解像度の画像が必要となる。また, 髪と同色の背景が隣接しないような比較的単純な背景下での検出を行っている。

カラー情報と距離情報の組み合わせ (RGB-D 画像) による静止画を対象とした領域抽出やクラスタリングの研究 [56, 57] も行われており, 距離情報の利用によって精度は向上しているが, 高速な処理は難しく, 処理速度に言及しているものは少ない。

髪領域の検出に関する研究の多くは, 色や形状により定義されたモデルをあらかじめ用意しており, 頭部姿勢の変化を伴う連続的な髪領域の検出を行うことはできない。また, リアルタイム性を求められる動画像へ適用する場合, 処理コストの低い方法が求めら

れる.

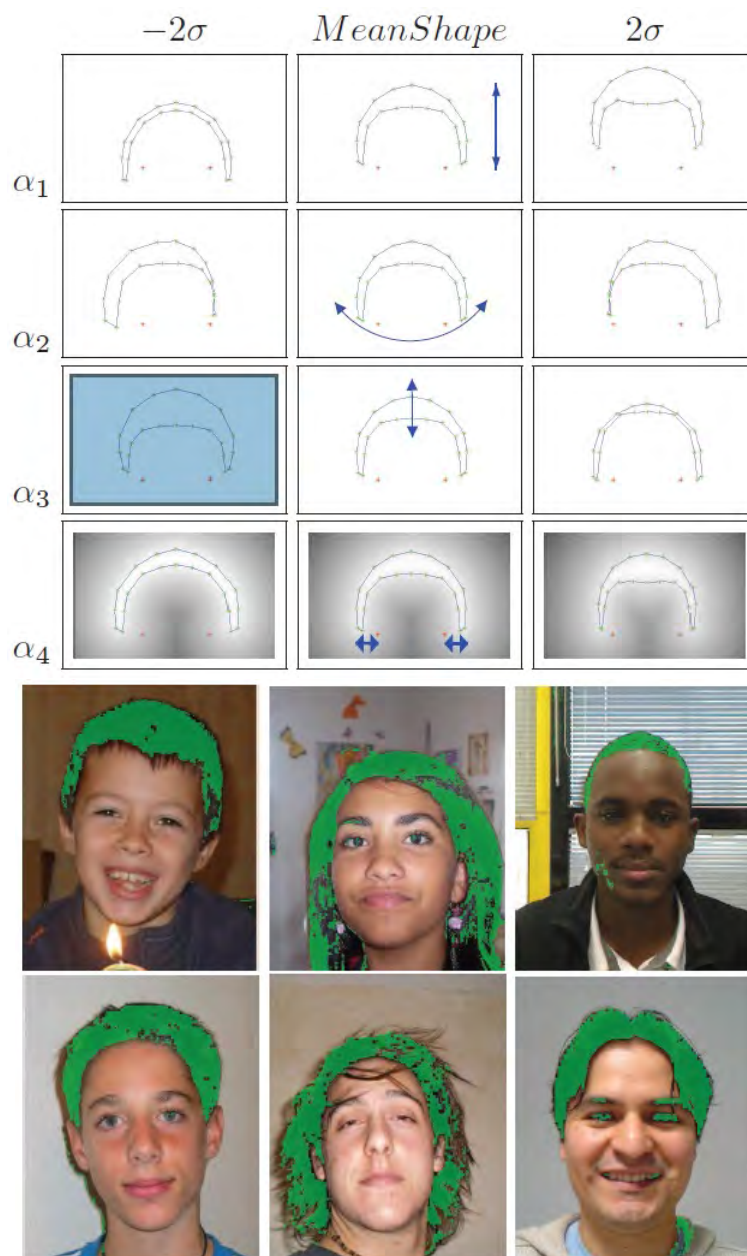


図 1.4 髪領域検出例 [53] (上: 髪モデル, 下: セグメンテーション結果)

### 1.2.3 対象追跡

対象追跡とは，与えられた動画シーケンスから，異なる時刻における対象領域の位置を対応付ける処理である．対象追跡は，セキュリティ，ヒューマンインタラクション，様々なシステムの自動化などその応用は幅広く，これまで多くの手法が提案されてきた．本項では，従来の代表的な追跡手法や本研究で対象とする顔や頭部領域の追跡手法について述べる．

#### テンプレートマッチング法

テンプレートマッチングによる追跡手法 [58, 59, 60, 61] は，最も基礎的なものであり，あらかじめ用意した追跡対象の部分画像と入力画像を照らしあわせることで，入力画像中の対象の位置を決定する．図 1.5 は，テンプレートマッチングを用いた追跡例である．この時用いる評価尺度には，SSD (Sum of Squared Differences)，SAD (Sum of Absolute Differences)，NCC (Normalized Cross Correlation) などがある．SSD は画素値の差を合計したもの，SAD は画素値の差の絶対値を合計したもので，これらが最小となる位置を探す．NCC は，照明環境による影響を抑えるために，テンプレートと入力画像の平均輝度値を用いて補正を行う正規相互相関と呼ばれる尺度であり，この値が最大となる場所を対象位置とする．

基本的に，テンプレートマッチングは入力画像とテンプレートの類似度が最大となる位置を探索するため，対象の回転など，見え方の変化に対応することは困難である．この問題を解決するために，追跡の成功時にテンプレートを更新していく可変テンプレートマッチング [62, 63, 64] が提案されているが，テンプレートへの背景混入による誤差の蓄積や成否の判定自体が難しいといった問題がある．

#### Mean Shift 法

Mean Shift 法 [65] を用いた追跡手法 [66, 67, 68, 69, 70] は，追跡対象を色ヒストグラムで表し，ヒストグラムの類似度を用いた山登り計算により探索範囲を移動させることで追跡を行う．また，Mean Shift を拡張して対象の大きさや姿勢の変化に対応した CAM Shift 法 [71, 72] が提案されている．図 1.6 に Mean Shift を用いた追跡例を示す．この手法は，低い処理コスト，対象の回転といった見え方の変化に対応できるなどの利点があげられるが，単色かつ非ランバートの物体を対象とした場合，色ヒストグラムが鋭くなり，僅かな色の変化によってヒストグラムの重なりが少なくなるなど照明環境の変動に弱いと





図 1.5 テンプレートマッチングを用いた追跡例 [59]

いった問題や、ヒストグラムの幅の変動など対象の大きさの変化に追従しづらいといった問題がある。



図 1.6 Mean Shift を用いた追跡例 [67]

### 点群追跡

計算コストや精度の改善のため、テンプレートマッチングのような画素単位のマッチングではなく、画像中の特徴点群を追跡する手法 [73, 74, 75, 76] が提案されている。これらの手法は、初期フレームで対象上の特徴点を検出 [77, 78, 79, 80] し、Lucas-Kanade アルゴリズム [81] を用いて特徴点群の追跡を行っている。図 1.7 は、Lucas-Kanade アルゴ

リズムを用いて点群を追跡した例である。これらの手法は、フレーム間における対象の移動が少ないという前提条件があり、高速に動く対象を追跡できない。また、背景上に現れる特徴点の影響で追跡が不安定になるといった問題もある。



図 1.7 Lucas-Kanade アルゴリズムを用いた特徴点追跡例 [76]

### MCMC を用いた手法

MCMC (Markov Chain Monte Carlo) [82] を用いた、運動予測に基づく Condensation (Conditional Density Propagation) や Particle Filter と呼ばれる追跡手法 [83, 84, 85, 86] が近年よく利用されている。これらの手法では、ランダムサンプリングによる仮説の生成と各仮説の尤度推定によって事後分布が推定され、追跡対象の状態が確率的に推定される。図 1.8 は、Condensation を用いた追跡例である。MCMC における事後分布を対象の仮説、事前分布を各仮説に対する尤度で表し、各仮説における観測モデルと観測値の類似度が高ければ尤度が高くなり、尤度に基づいて次フレームの仮説が生成される。MCMC を用いた手法は、任意の尤度関数を定義できるため、対象に応じた自由な設計ができるが、仮説を高次元にした場合に計算量が増加する問題や、尤度分布の裾野が狭いと不安定になるといった問題がある。

### 顔追跡

Vadakkapat ら [87] は、色情報を用いた顔の検出と追跡を提案している。この手法は、照明変動の影響が少ない YUV 色空間の UV 平面において肌色領域を定義し、Mean Shift 法をベースとした手法により顔追跡を行っている。このような色情報を用いた顔の追跡手法は、シンプルで高速な処理が可能であるが、あらかじめ肌の色を定義する必要があり、環境にある似た色を誤追跡する可能性がある。また、登録された肌色と異なる人種や大きな照明変動によって追跡が開始できないといったことが生じる。

Microsoft 社の顔追跡ライブラリ [43] では、カラー画像から顔を検出した後、色と距離情報より獲得した 3D の AAM (Active Appearance Model) [88, 89] を利用することに

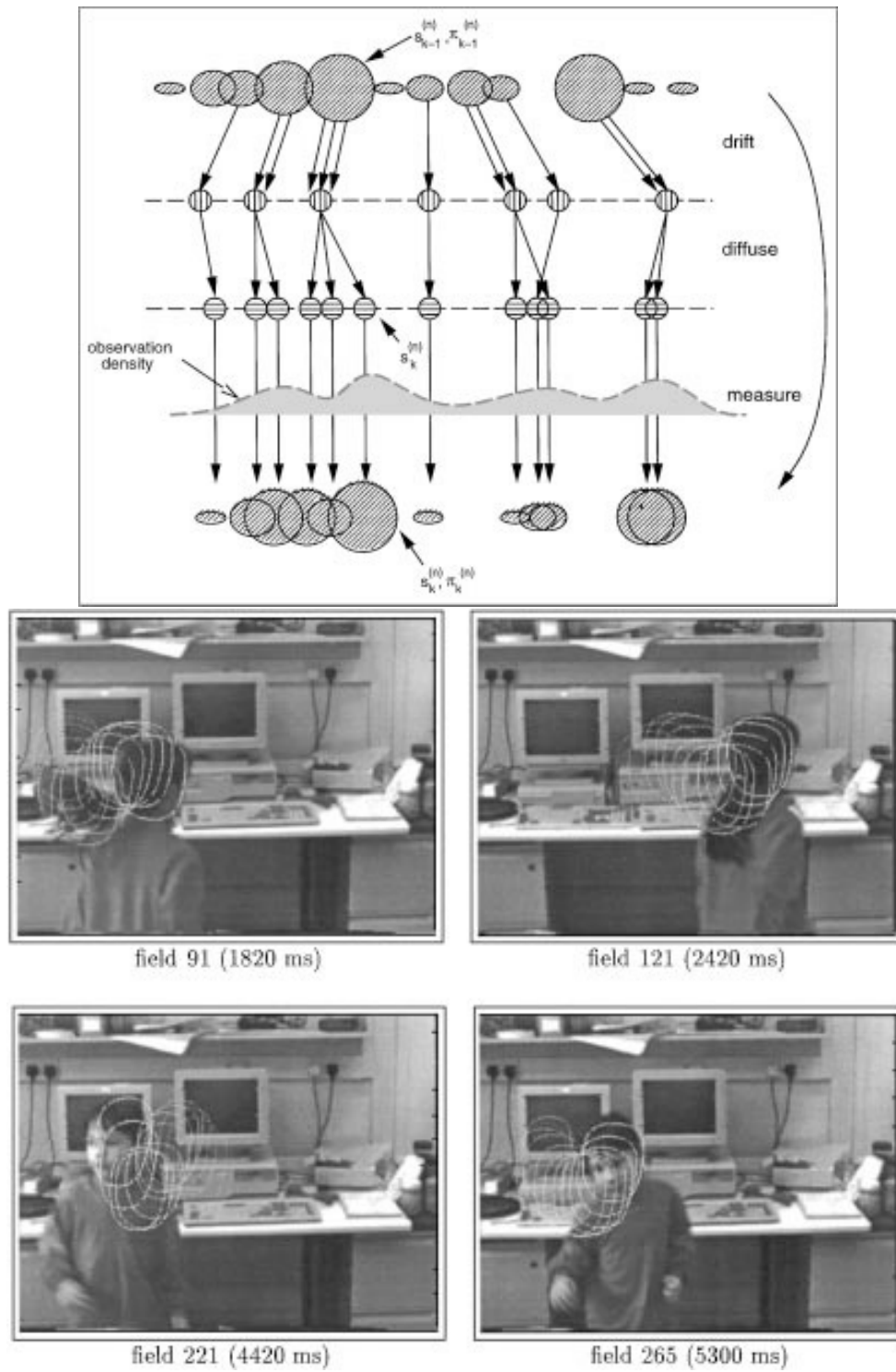


図 1.8 Condensation を用いた追跡 [83] (上: 1 フレームの処理, 下: 頭部位置を追跡した例)

よって安定な顔追跡を実現している．図 1.9 に AAM を用いた追跡例を示す．これは，奥行き情報を用いた特徴記述によって追跡精度を向上させ，顔パーツの 3 次元情報から顔の表情や姿勢を得ることができる．

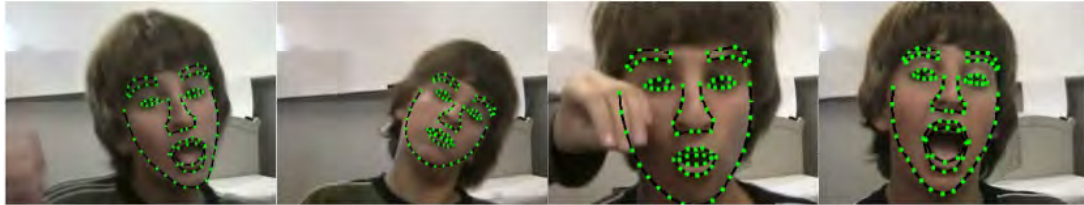


図 1.9 AAM を用いた顔追跡例 [45]

### 頭部追跡

Birchfield[90] は，アスペクト比を固定したスケール可変の垂直楕円で頭部モデルを定義し，楕円輪郭上の勾配強度と楕円内の色ヒストグラムを用いて頭部追跡を行っている．この手法では，頭部の輪郭と色情報を用いることで水平方向の回転やスケール変化，オクルージョンに対して追跡可能な追跡を行っている（図 1.10）．また，頭部を 3D モデルで表して追跡を行う手法 [91, 92] も提案されており，追跡の安定化を実現している．これらの手法は，頭部を楕円または楕円体として定義しているため，顔や髪領域を画素単位で得ることはできない．



図 1.10 楕円を用いた頭部の追跡 [90]

## 1.3 提案手法の概要

機械が人の状態を認識する上で、画像中の顔位置を特定することは最も基礎的で重要な処理である。また、髪の色やスタイルには重要な個人特徴が含まれているため、顔領域に加えて髪領域を取得することで、個人識別の特徴として用いたり頭部の姿勢推定に利用できる。さらに、頭部姿勢の変化に関わらず頑健な追跡が可能となる。そこで本論文では、画像中から顔・髪・頭部領域の決定を高速かつ安定に行うことを目的とし、毎秒 30 フレームのビデオレートで撮影された動画像において顔や髪領域を自動的に検出・追跡するシステムを提案する（図 1.11）。提案システムは、識別器による顔検出と色情報を用いた追跡手法の組み合わせで構成されており、顔検出で得られた領域から追跡に必要な情報をオンラインで取得することによって、その時の照明環境に応じたターゲット・背景モデルを構築し、自動的に追跡を開始できるシステムとなっている。本論文で提案する手法は以下の 3 点であり、検出や追跡において奥行き情報を利用することで処理の高速化や安定化を実現する。

提案手法 1 奥行き情報を用いた顔検出の高速化

提案手法 2 顔検出と色弁別度追跡法を組み合わせた顔追跡システム

提案手法 3 顔検出と 6D K-means Tracker を組み合わせた髪追跡システム

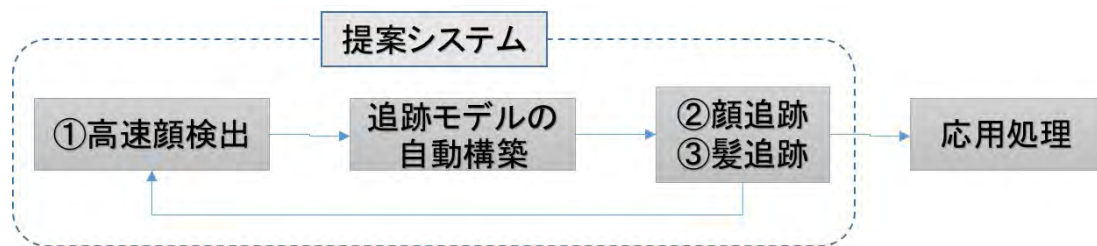


図 1.11 提案システムの全体像

奥行き情報を用いた顔検出の高速化（図 1.11①）

識別器を用いた顔検出は、大量の学習画像から識別器が構築されるため、汎用性が高く識別性能は良い。単眼カメラで撮影された画像から識別器を用いて検出する場合、画像内の顔の位置や大きさは未知であるため、画像内から位置とスケールを変えて切り出した矩形領域を識別器に通し、顔か非顔かの判別を行う必要がある。そのため、識別回数は数万

回にもなりビデオレートでの処理が困難となる。また、ステレオカメラで顔検出を行った場合、視点や視野の増加によって検出精度の向上が期待できるが、単純に複数枚の画像に対して顔検出を行うと処理コストが倍増していく。そのため、これまでは高い識別率を持ちながら高速に顔検出できる、識別器の構築法に関する研究が多かった。

本論文では、奥行き情報を利用した探索方法の改善によって既存の識別器を用いた顔検出の高速化を行う手法を提案する。基本アイデアとして、ステレオカメラによって得られる奥行き情報に基づいて画像上での顔サイズを推定することで、画像中の各領域の奥行きに応じたスケールのみでの探索を行う。限定された領域とスケールの探索によって識別回数が大幅に削減され、識別回数の削減にともなって誤識別数も削減することができる。さらに、ビデオレートでの顔検出を実現するために、ステレオ画像に対してスパースに奥行き情報を求める手法を提案する。このようにして、ステレオカメラを用いることで顔検出の精度を向上させるだけでなく、単眼カメラで行うよりも高速な顔検出が可能となる。また、本研究では識別器として、Viola らが提案している手法（付録 A）をベースとした顔識別器 [93] を用いる。

#### 顔検出と色弁別度追跡法を組み合わせた顔追跡システム（図 1.11②）

顔検出を毎フレーム行うことで顔領域を追跡することは可能であるが、正面顔のみの学習画像で構築された識別器では顔の姿勢変化に対応できない。我々の研究グループはこれまでに、色弁別度を用いた実時間ステレオ対象検出・追跡の手法 [94, 95, 96, 97, 98, 99] を提案している。先行研究 [98] では、任意の対象について追跡を行っており、可変テンプレートマッチング法や Mean Shift 法に比べ安定した追跡を行えることが示されている。

先行研究 [98] による顔検出・追跡システムでは、色に基づいた画素単位で行う最近傍識別による色ターゲット検出法 [100] を用いて顔の検出を行っており、予めターゲット色として肌色を教示し、画像中から肌色領域を検出することで顔の初期領域を取得していたため、画像内に手など教示したターゲット色と似た色が入ってくると誤って検出してしまうという問題や、教示した色と肌の色が異なっていると検出できないといった問題がある。提案システムでは、より正確に顔だけを検出するために提案手法 1 を組み合わせる。さらに、追跡途中で継続的に顔検出を行い、検出に成功した際に得られる顔領域に基づいて顔と背景の色ヒストグラムを構築し、背景に少なく追跡しやすいターゲット色を選びながら自動的にターゲットと背景色の教示を行う。このように、顔と非顔の色モデルのアップデートを行うことで、様々な人種、照明環境や周辺環境の変動に強い追跡手法となる。

先行研究 [98] に残された追跡中の問題点として、背景に対象と似ている色が現れ対象と隣接した場合に追跡範囲である楕円が広がりすぎてしまう場合があるが、対象までの距離



と楕円サイズを対応付けることで解決できることが示唆されている。本研究では、奥行き情報を利用して画像上での顔サイズを推定し、楕円サイズを制限することでこの問題を解決する。

提案システムでは、顔識別器が正面顔だけを検出するという特性を活かし、本システムに興味のある人だけを検出・追跡するといったインタラクションを実現している。また、ステレオカメラがPTU(Pan-Tilt Unit)の上に搭載されており、PID制御[101, 102, 103]により、顔を画像中央に捉えるように自動的に追尾し、広範囲の追跡を行うことができる。

### 顔検出と 6D K-means Tracker を組み合わせた髪追跡システム (図 1.11③)

髪領域は、色やテクスチャの情報が乏しく形状の個人差が大きいため、モデル化や安定な特徴を抽出することが難しい。そのため、髪領域の検出に関する報告例は少なく、追跡を行っている例は見当たらない。しかし、髪領域は個人を特徴付ける重要な要素であり、また、髪領域を追跡することで、顔領域が見えなくなるような頭部の姿勢変化があった場合でも継続して追跡することが可能となる。

東洋人の場合、髪の色は基本的に黒・灰・白色であるため、色彩情報が乏しいと同時に背景に同じ色が存在する可能性が高い。したがって、提案手法2で利用した色弁別度追跡法をそのまま適用しても安定に追跡することが難しい。背景内の同色から判別を行うには奥行き情報の利用が不可欠であるが、色弁別度追跡法ではターゲットと背景を色情報のみで表現してテーブル化しているため、常に変化する奥行き情報を導入するのが困難である。また、髪スタイルは様々であり、姿勢による変化も大きいため画素単位の判別法が望ましい。

本研究グループが提案している K-means Tracker[104, 105, 106, 107, 108, 109, 110, 111] は、3次元の色情報と2次元の画素位置情報からなる5次元特徴空間において、画素単位のクラスタリングを行いビデオレートでの非剛体物体の追跡に成功している。また、1次元の奥行き情報を加えた FB-Clustering Tracker[112] が提案されており、追跡の安定化が実現している。本論文では、2次元の画素位置情報と1次元の奥行き情報を統合した3次元空間として位置情報を取り扱うことで、奥行き情報の導入による安定化を実現する。さらに、背景に応じた非ターゲットクラスタ中心の配置法[113]を用い、この特徴ベクトルにも奥行き情報を導入することで、次元拡張に伴う処理コストの増加を抑制することができる。

提案システムでは、Kinectにより必要な RGB-D の情報を獲得し、顔検出と提案する 6D K-means Tracker を組み合わせることで髪の追跡を実現する。提案手法1で検出された顔領域より、追跡する人や照明環境に応じた初期のターゲット・背景モデルを自動的に

構築する。

## 1.4 論文の構成

本論文は，下記の 5 つの章から構成される。

### 第 1 章：はじめに

本章では，研究背景や目的などを述べる．また，本研究で扱う検出や追跡手法について，これまでに報告されてきた関連手法を概説し，提案手法の概要を述べる．

### 第 2 章：奥行き情報を用いた顔検出の高速化 [研究業績 3, 4]

本章では，ステレオカメラによって得られるスパースな奥行き情報を利用し，既存の識別器 [36, 93] を用いた顔検出を高速かつ高精度に行う手法について説明する．

### 第 3 章：顔検出と色弁別度追跡法を組み合わせた顔追跡システム [研究業績 5, 6]

本章では，色弁別度追跡法について概説し，先行研究において構築された顔検出・追跡システムに残された課題を解決することで提案システムの構築を行う方法を説明する．第 2 章で提案した高速な顔検出法と色弁別度追跡法を組み合わせ，検出に成功した際に随時，ターゲットと背景モデルのアップデートを行うことで，照明や周辺環境の変化に強い追跡手法となる．

### 第 4 章：顔検出と 6D K-means Tracker を組み合わせた髪追跡システム [研究業績 1, 2, 8, 9]

本章では，従来の K-means Tracker について概説し，奥行き情報を用いて追跡を安定化させる手法について説明する．また，第 2 章で提案した高速な顔検出法との組み合わせにより自動的に追跡が開始できるシステムについて説明する．髪領域を追跡できるようになることで，顔領域が見えなくなるような頭部の姿勢変化に対しても継続した追跡が可能となり，顔と髪領域からなる頭部を追跡することで，頭部の姿勢推定や，髪色・スタイル変化などの応用アプリケーションに貢献できる．

### 第 5 章：おわりに

最後に，本研究の成果を総括し，今後の課題について議論する．



## 第 2 章

# 奥行き情報を用いた顔検出の高速化

本章では，Haar-like 特徴と AdaBoost によって構築された顔識別器（付録 A）を使用し，探索方法に着目することで顔検出を高速化かつ高精度化する手法を提案する．

一枚の画像から顔の位置を検出する場合，画像内の顔の位置や大きさは未知であるため，位置やスケールを変えて切り出した数万ものサブウィンドウを識別器に通して，顔か否かの判別を行う必要がある．図 2.1 は，探索の様子を表しており，画像からサブウィンドウを切り出し，それを識別器に通すことで顔あるいは非顔と判断する．また，識別器を構築する際に学習で用いる画像サイズは一定のため，識別器に通すサブウィンドウの大きさは決まっている．そのため，まずオリジナルサイズの入力画像から識別器のウィンドウサイズで画像中を走査し，続いて入力画像を縮小した画像に対して探索を行う．これを繰り返し，画像内のあらゆる位置やスケールに対応したサブウィンドウについて探索（ピラミッドスキャン）を行う必要がある．このようにピラミッドスキャンを行うと識別回数は数万回にもなり，実時間での顔検出は難しくなる．また，ステレオカメラを用いて顔検出を行った場合，異なる視点で撮影された複数枚の画像を探索することで検出精度の向上が期待できるが，処理コストが倍増してしまう．

このように，識別回数と検出速度は比例関係にあり，識別回数を削減することができれば大幅な速度の改善が期待できる．提案手法では，ステレオカメラを用いて得られる奥行き情報を活用して探索範囲やスケールを限定し，識別回数の削減によって顔検出の高速化を実現する．ステレオカメラを利用することで，検出精度を向上させるだけでなく，単眼カメラよりも高速な顔検出を行うことができる．また，識別回数の減少にともなって誤識別を減少することもできる．さらに，スケールを限定した探索によって，写真やカレンダーに写っている顔など実物の顔と大きさが異なる顔を検出しないといった効果も得られる．

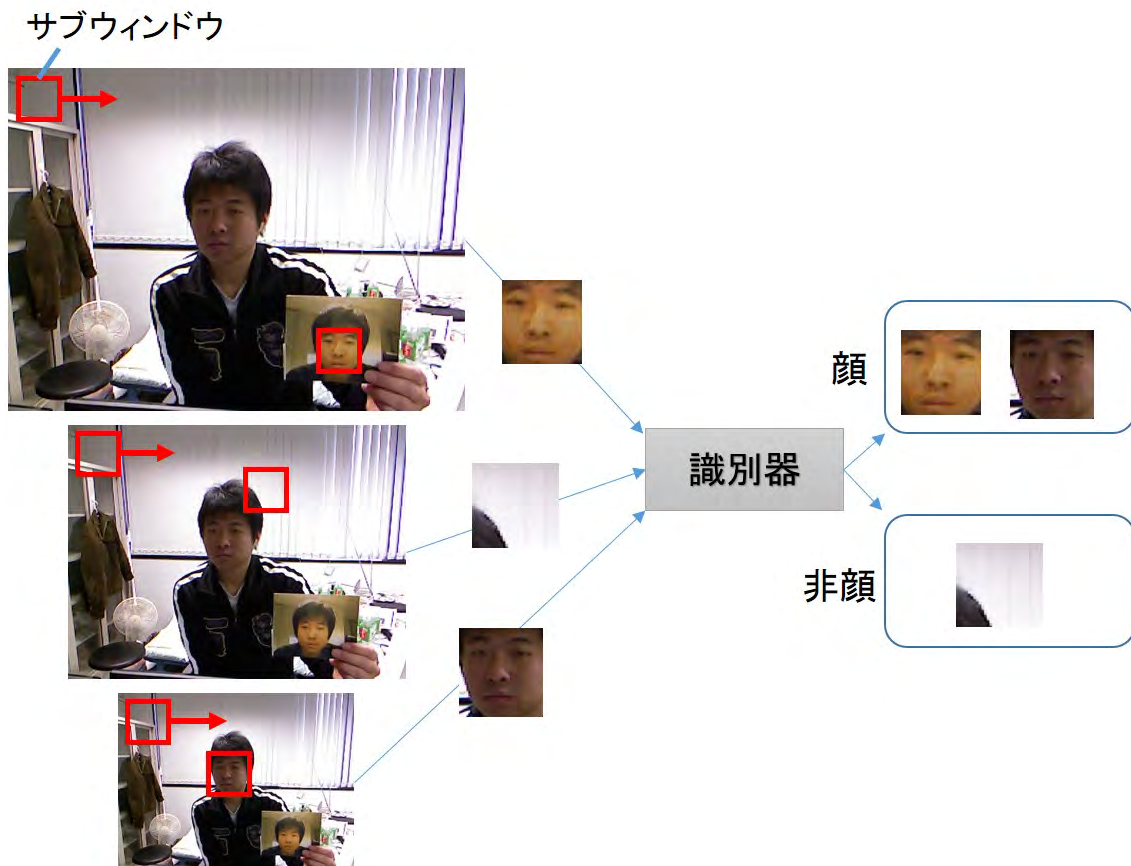


図 2.1 ピラミッドスキャン

ステレオカメラを用いて奥行き情報を取得するためには処理コストの高いステレオ処理が必要となるが、ビデオレートでの顔検出を実現するために、全画素ではなく検出できる最小顔サイズに合わせた画素間隔でスパースに奥行き情報を求める手法を提案する。

## 2.1 ステレオカメラ

提案手法では、視点の拡大と奥行き情報の取得を行うためにステレオカメラを用いる。ステレオカメラとは、複数のカメラで構成されたカメラシステムであり、異なる視点から撮影された対象物の画像中におけるずれ（視差）から、三角測量に基づいたステレオ処理によってその対象物の3次元的位置や形状を取得することができる。ステレオカメラでは、正確なステレオ処理を行うために複数カメラの同期や位置関係が必要であり、処理を簡便化するため光軸が並行になるように2つのカメラが配置された、1台で両眼視差を再現するような2眼の並行ステレオカメラが一般的となっている。

本手法では，ステレオカメラとして Point Grey Research 社の Bumblebee Stereo Vision Camera（図 2.2）を用いる．ステレオ処理は，エリアベースの手法で行われており，画像中の各画素ごとに奥行き情報を取得することが出来る．



図 2.2 Point Grey Research, Inc. Bumblebee Stereo Vision Camera

ステレオ処理とは，左右カメラからの画像中における対象物の位置の違い（視差）によって対象までの距離を求める方法である．近くにある物体ほど視差は大きくなり，遠くなるほど視差は小さくなる．カメラ校正済みの左右に配置された 2 台のカメラで撮影された図 2.3 のような 2 枚 1 組のステレオ画像を用いて，例えば，右画像について距離を求めるときは，まず右カメラの画像内のある点が左カメラの画像のどの点と対応するかを相関の計算により求める．次にその対応関係を使った三角測量により，各点の 3 次元的位置を計算する．

並行ステレオカメラの場合，画像中の任意の点におけるエピポーラ線は水平となるため，各画素における奥行き計算は次のように行われる．まず，次式 (2.1) を用いて右画像と左画像の点の対応付けを求め，

$$d = \min_{d=d_{min}}^{d_{max}} \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} |I_{right}[x+i][y+j] - I_{left}[x+i+d][y+j]| \quad (2.1)$$

対応付けられた点の視差  $d$  から式 (2.2) により奥行き  $Z$  を求める．

$$Z = \frac{fB}{d} \quad (2.2)$$

ここで， $d_{max}$ ， $d_{min}$  はそれぞれ最大の視差と最小の視差， $m$  はマスクサイズ， $I_{right}$ ， $I_{left}$  はそれぞれ右画像と左画像， $f$  は焦点距離， $B$  はカメラ間距離である．このように，



(a) 左カメラで撮影された画像

(b) 右カメラで撮影された画像

図 2.3 ステレオ画像

左右画像の点のマッチングには SAD (Sum of Absolute Difference) が用いられ, SAD の値が小さいほど, 似ている位置となる.

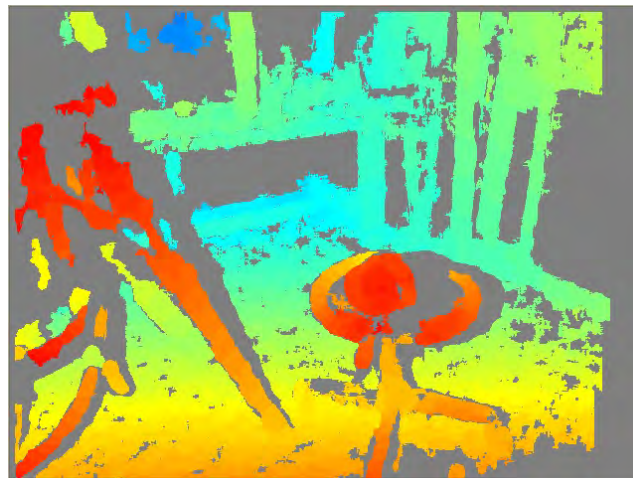


図 2.4 視差画像

この対応付け処理を画像内の全ての点に対して行うことで, 全画素に対応した距離画像を構築することができる. 図 2.4 は, 図 2.3 のステレオ画像において左右画像から点のマッチングを行い, 各画素の視差を求めて可視化した視差画像である. 図 2.4 では, 推定された視差を疑似カラーで表している. この視差を求めるマッチング処理は, 多くの処理時間を要してしまい, 顔検出などを含めた処理全体をビデオレートで行うことは困難となる. そこで, 本研究では, 入力画像の全画素ではなくサンプリングされた点のみで距離計

算を行うことによって，検出精度を保証した上でステレオ処理の時間を抑えるようにした．次章でこの方法を詳しく説明する．

## 2.2 ステレオカメラを用いた高速顔検出法

本手法ではステレオ処理によって得られた距離情報を用いることで、探索する領域とスケールを特定し、識別回数を減少させることで、ビデオレートでの顔検出を実現する。基本アイデアは、

1. スパースサンプリングによるステレオ処理の高速化
2. 識別回数の削減による検出の高速化
3. ステレオ画像を用いた検出精度の向上

という3点である。以下それぞれの点について詳しく述べる。

### 2.2.1 スパースサンプリングによるステレオ処理の高速化

前節で述べたように、入力画像の全画素に対してステレオマッチングを行うと、多大な計算時間となり、ビデオレートでの処理は難しい。そこで、一定間隔でサンプリングした画素に対してのみステレオマッチングを行うことで、ステレオ処理にかかる時間を抑えることが可能となる。このときサンプルが少なすぎると、人が遠くに映っているなど画像上での顔が小さくなった場合に、顔がサンプル点の間に位置すると顔までの距離が計算できなくなるという問題がある。これは、サンプル数が十分多ければ問題ないが、サンプル数を増やすほど処理時間も多くなる。

本節ではこのような問題を考慮したサンプリングの方法について述べる。以下に述べるようなサンプリングを行うことで、ステレオ処理に要する時間を  $320 \times 240$  の画像において 10msec 程度に抑えることができる。また、ステレオ処理時の各種フィルタについて説明する。

#### スパースサンプリング

ここでは、処理時間とサンプル数を考慮したサンプリングの方法について述べる。まず、使用している識別器では、顔として識別できる最小サイズが  $20 \times 20$  ピクセルであるため、図 2.5 の様にサンプル点の間隔を 20 ピクセルとする。予備実験により、このままのサンプル点に対して奥行き情報を求めると約 20msec の処理時間を要し、検出処理を含めた全処理をビデオレートで行うことが困難である。そこで、図 2.5 のようにサンプルを赤と青の2つのグループに分ける。これら2種類のサンプル配置をフレームごとに切り替えてステレオ処理を行う。以上のようにすることで2フレームに1回は最小サイズの

顔を検出できることを保証している。

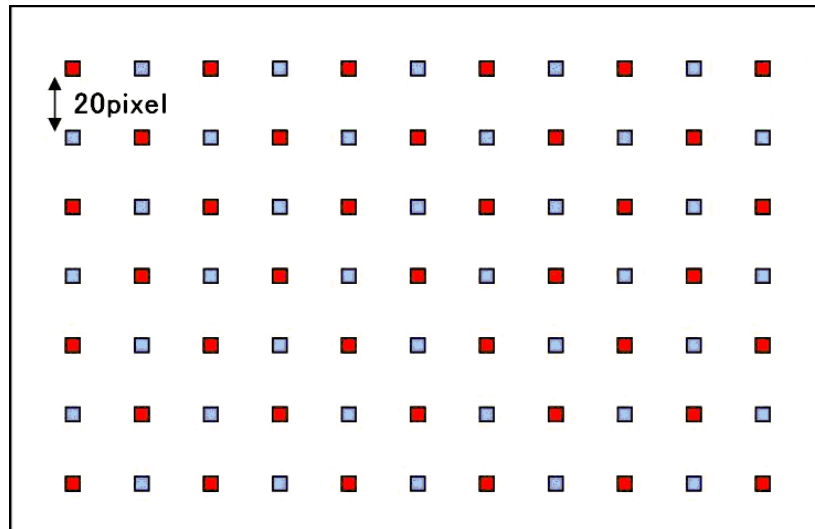


図 2.5 サンプル点の配置

### 重点サンプリング

上述のようなスパースサンプリングをすることで、あるフレームでは最小サイズの顔を検出できても、次のフレームでは顔上にサンプルが配置されず検出できない場合がある。また、画像上での顔が小さい場合、顔上のサンプル数が少なくなるので、距離が正しく計算できなかった場合に検出できないことがある。一方、画像上での顔が大きい場合必要以上のサンプル数となる。このような問題を軽減するために重点サンプリングを行う。

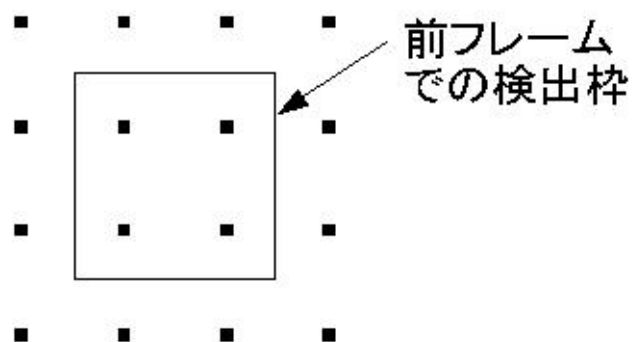


図 2.6 重点サンプリング

重点サンプリングは前フレームで顔が検出された付近で行い、図 2.6 のようなサンプル



配置となる。前フレームで図の矩形のように顔が検出された場合、その付近にサンプルを16点配置する。こうすることで、小さい顔が検出されたところではサンプル間隔が細かく、大きい顔が検出されたところではサンプル間隔が粗くなる。

### ステレオ処理時のフィルタ

ここでは、実際の入力画像に対してサンプリングによるステレオ処理を行った例と、ステレオ処理時のノイズ低減や安定性を高めるためのフィルタについて述べる。

図 2.7 は入力されたステレオ画像の右画像である。図 2.8 はサンプル配置を示している。ここでは、前フレームで顔が検出されていたため、その付近に重点サンプリングが行われている。このような入力画像、サンプル配置でステレオ処理を行った結果、得られた距離画像が図 2.9 である。緑色の点の濃さは距離を表しており、濃いほど距離が近いことを示している。結果を見ると、距離値が出ていないところがある。このように距離値がないのは、得られた距離が識別出来る顔サイズの範囲外、つまり遠すぎたり近すぎたりといった場合、あるいは以下で述べるようなノイズを低減するためのフィルタによって除去された場合である。

ステレオ処理時のサンプル点は 4x4 画素となっている。算出された各画素の距離値には、多くの場合ノイズが含まれており、これを低減するためにサーフェイスバリデーションフィルタを用いている。サーフェイスバリデーションフィルタとは、ある点において、そこで計測された距離値がその近傍の点の距離値と大きく異なっている場合に、その点を除去するものである。すなわち、計測された距離を一連のサーフェイス（物体表面）として認識し、認識できなかった点を除去するフィルタである。

また、ステレオ相関演算の精度を上げるためにエッジフィルタを用いている。エッジ画像上でステレオ相関演算を行うことで、両カメラ間で対象の明るさが異なっている場合でも安定した計測値を得ることが出来る。

## 2.2.2 識別回数の削減による検出の高速化

画像上の各領域の奥行き情報に基づいて、その領域の探索スケールを決定し、探索する領域とスケールを限定することで識別回数を削減することができる。探索する領域とスケールの決定は図 2.9 で求めた距離画像を元に行う。基本的な流れは次のようになっている。

1. 距離値から画像上での顔サイズを推定する
2. 推定された顔サイズから探索領域と識別スケールを決定する





図 2.7 入力画像

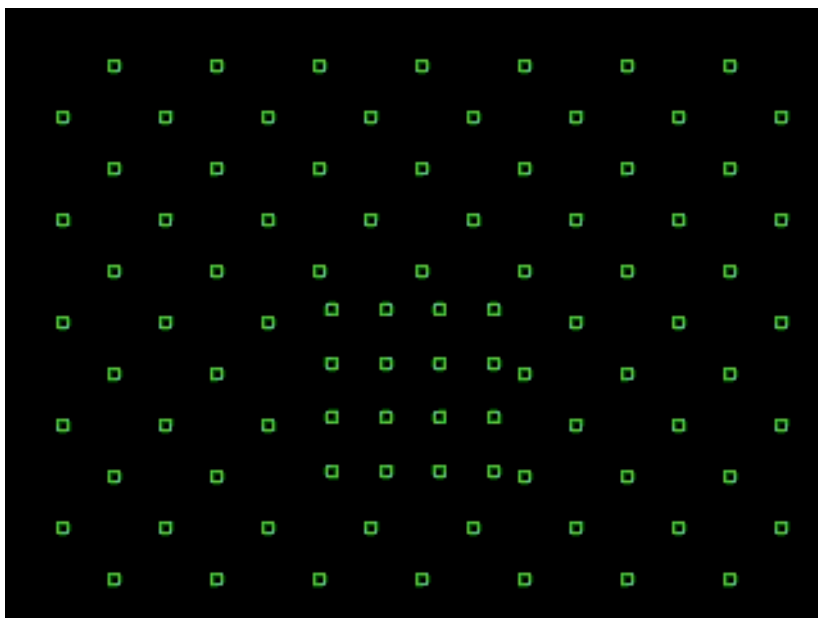


図 2.8 サンプル配置

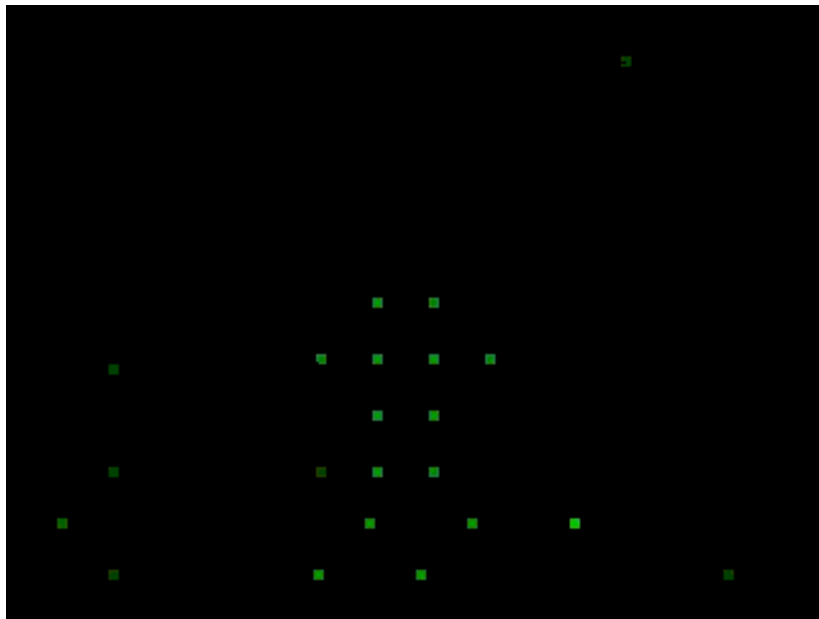


図 2.9 サンプルングによるステレオ処理の結果例

### 3. 探索領域の統合を行う

以下それぞれについて詳しく述べる.

#### 画像上における顔サイズの推定

顔の大きさには個人差があるが、それほど大きな違いがない. そのため、ワールド座標系における実際の顔の大きさをあらかじめ決めておくことで、カメラから顔までの距離が分かれば画像上での顔の大きさを求めることができる.

図 2.10 に示すように、実際の顔の大きさを  $W_{size}$  とすると、画像上の顔の大きさ  $I_{size}$  は次式 (2.3) により推定できる.

$$I_{size} = \frac{f}{Z} W_{size} \quad (2.3)$$

ただし、 $f$  はピクセル単位の焦点距離、 $Z$  はワールド座標におけるカメラか物体までの距離である.

図 2.11 は入力画像と距離画像の例である. 同図のように得られた距離値から式 (2.3) により画像上の顔サイズを推定し、それぞれの距離に応じた顔サイズと同じスケール (図中の青色の矩形) のみの探索を行う. 距離が近いところは大きなスケールの矩形で、距離が遠いところは小さいスケールの矩形での探索が行われる. 提案手法では、図 2.9 のサン

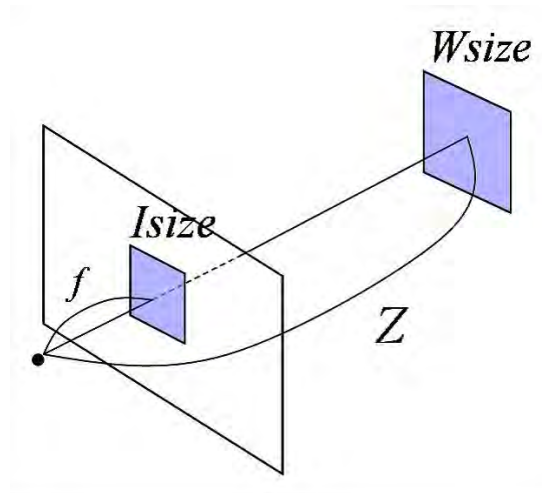


図 2.10 画像上での顔サイズの推定

プリングされた距離画像より，探索領域とスケールを決定した上で探索を行う．以下処理の流れを説明する．

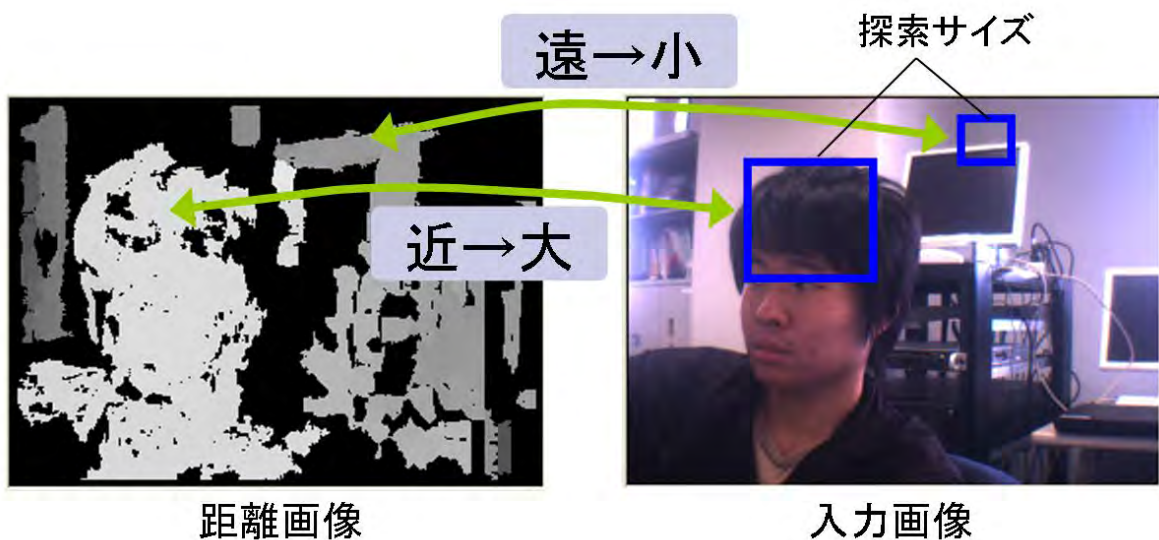


図 2.11 距離に応じた探索スケールの決定

### 探索領域とスケールの決定

図 2.9 の各サンプル点において，式 (2.3) により求められた顔サイズを用いて探索領域とスケールを決定する．探索領域は，図 2.12 に示すようにサンプル点から上下左右に顔

サイズ分だけ離れた矩形領域とする．あるサンプル点において顔までの距離が得られたとき，そのサンプル点が顔のどこに位置しているかわからない．そのため，上述のような探索領域とすることで，サンプル点が顔上のどこに位置していても，探索領域内に顔が含まれるようになる．

探索するスケールは，識別器に通す際のサブウィンドウの大きさが式 (2.3) で求めた顔サイズと同じになるようにする．また，顔の大きさには多少個人差があるので少し大きくしたスケールや，小さくしたスケールでも探索する必要がある．

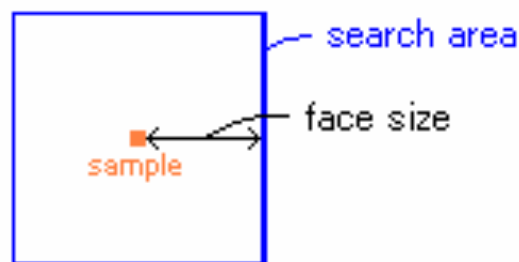


図 2.12 各サンプル点における探索領域

### 探索領域の統合

図 2.9 の距離画像における全てのサンプル点について，前述のようにして探索領域を決めると図 2.13 のようになる．図 2.9 を見ると，画像中にある物体付近では，距離値のほとんど変わらないサンプル点が集まっていることがわかる．そのため，図 2.13 のようにスケールの近い探索領域が重なり，同じ位置を同じようなスケールで何度も探索するという無駄な処理が起こる．これを防ぐために，探索領域が重なるサンプル点を距離値が近いときに限りひとまとまりにし，一つの探索領域とする．また，この領域内を探索する際のスケールは，まとめたサンプル点の距離値の平均から決める．

図 2.13 に対して，上述の処理を行った最終的な探索領域は図 2.14 のようになる．図中のそれぞれの矩形領域内を求められた特定のスケールのみで探索する．

さらに，ノイズによる無駄な探索領域を増やさないために，各領域において求められた顔サイズとサンプル数から，その領域を探索するか否かを決定するようにした．例えば，ある領域で，顔サイズが大きいほどサンプル点数も多くなるはずである．このとき，サンプル点数が少なすぎればノイズによる探索領域だと判断して探索しない．同様に，ある領域のサンプル数が 1 つの場合もノイズとする．顔が識別できる最小サイズのような場合であっても，顔上で得られたサンプルと体上で得られたサンプルが統合されて 1 つの探索領

域となる．このように，顔や物体がある付近では，ほとんどの場合複数のサンプルが統合される．つまり，どのサンプルとも統合されないような独立したサンプルはノイズである可能性が極めて高いためこれを除去する．

また，領域同士が重なる位置では，サンプル数が多いほうの領域に対応するスケールのみで探索する．

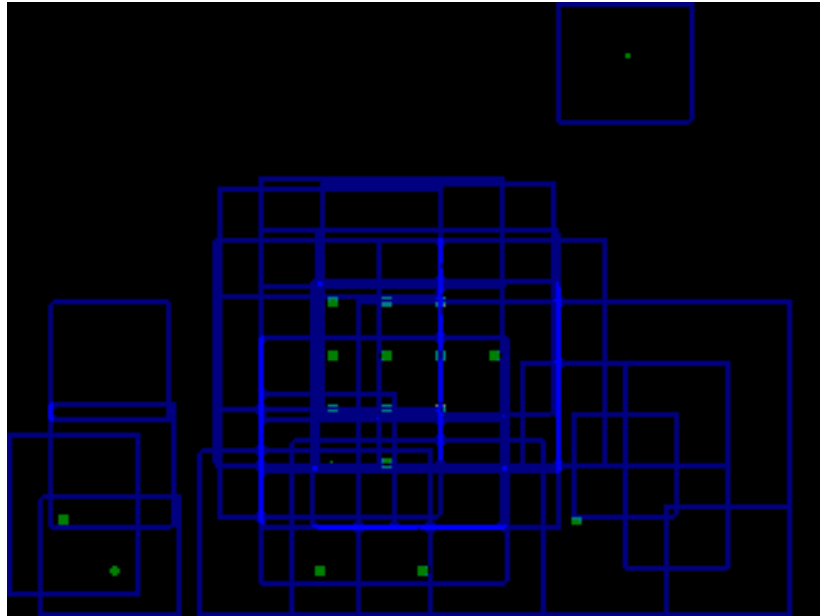


図 2.13 探索領域を決定した例

### 2.2.3 ステレオ画像を用いた検出精度の向上

本研究で顔検出時に用いている識別器は，正面顔画像のみによる学習で構築されている．そのため，ある程度横を向いた画像などでは顔と識別することができない．本研究で用いるステレオカメラでは，視点の異なる左右 2 枚の画像が得られる．このような 2 枚の画像での探索を行うと，一方の画像では少し横を向いていて顔と判断できない場合でも，もう一方の画像では視差があるため正面に近く顔と判断できるケースがある．このように何らかの影響で一方の画像では検出できなかった顔がもう一方の画像では検出できるということがあるため，ステレオ画像を探索することによって検出率の向上が期待できる．

また，全探索（ピラミッドスキャン）ではあらゆる位置とスケールを探索するのに対して，提案手法ではステレオ画像によって得られる奥行き情報に基づいて，探索する位置とスケールを制限することで識別回数を削減しているため，識別回数の減少に伴って必然的

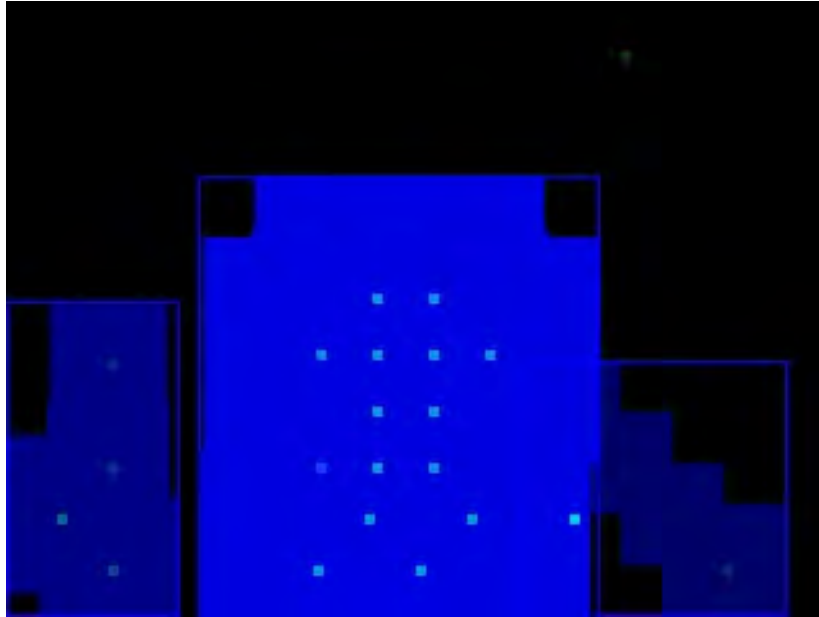


図 2.14 探索領域の統合

に誤検出の数も減少する。

以上のことより従来の全探索に比べ提案手法では検出精度が向上すると考えられる。

## 2.3 実験

### 2.3.1 実験環境

提案手法の処理速度と検出精度を確認する実験を行うにあたり 100 フレームほどの実画像系列（ステレオ画像  $320 \times 240$  画素）を数種類用意した。これらの画像系列に対し全探索（ピラミッドスキャン）と提案手法による探索の比較実験を行った。

全探索では、識別できる最小顔サイズを  $20 \times 20$  とし、倍率を 1.25 倍としている。一般的に、物体検出ではスケールを細かく見るほど検出率は上がるが、使用ライブラリのデフォルト値を用いている。

提案手法では次のようにしている。式 (2.3) のワールド座標系における顔の大きさ *RealFace* を 14cm と設定する。1 スケールだけの探索では個人による顔の大きさの違いや距離計算時の誤差などがあるため十分ではない。そのため、各探索領域で 2 スケール分を探索するようにする。

表 2.1 探索サイズ

	全探索	提案手法
開始サイズ	$20 \times 20$	式 (2.3) の $I_{size}$
スケール倍率	$1.25^N (N = 0, 1, 2, \dots)$	$1.25^N (N = 0, 1)$

実験環境は次のようになっている。PC は Windows XP, Pentium(R) 4 CPU 3.0GHz (2 CPUs), 1024MB RAM のものを使用し、顔識別器は OpenCV[93] のライブラリ関数を利用している。

### 2.3.2 処理速度の実験結果

用意した 2 種類の画像系列に対し、提案手法による顔検出実験を行った。系列 1 は 1 人が映っているもので、系列 2 は 2 人が映っているものである。ステレオ画像のうち一方の画像では各探索領域において式 (2.3) で求められた  $I_{size}$  で探索し、もう一方はその 1.25 倍のサイズで探索している。

図 2.15, 図 2.16 に提案手法を用いて系列 1 (1 人), 系列 2 (2 人) に対し顔検出を行った際の処理時間のグラフをそれぞれ示す。横軸はフレーム番号で、縦軸は処理時間 (msec)

である。緑色 (◆) のグラフは検出にかかる時間、オレンジ色 (■) のグラフはステレオ処理にかかる時間であり、青色 (▲) のグラフは1フレーム全体の処理時間となっている。

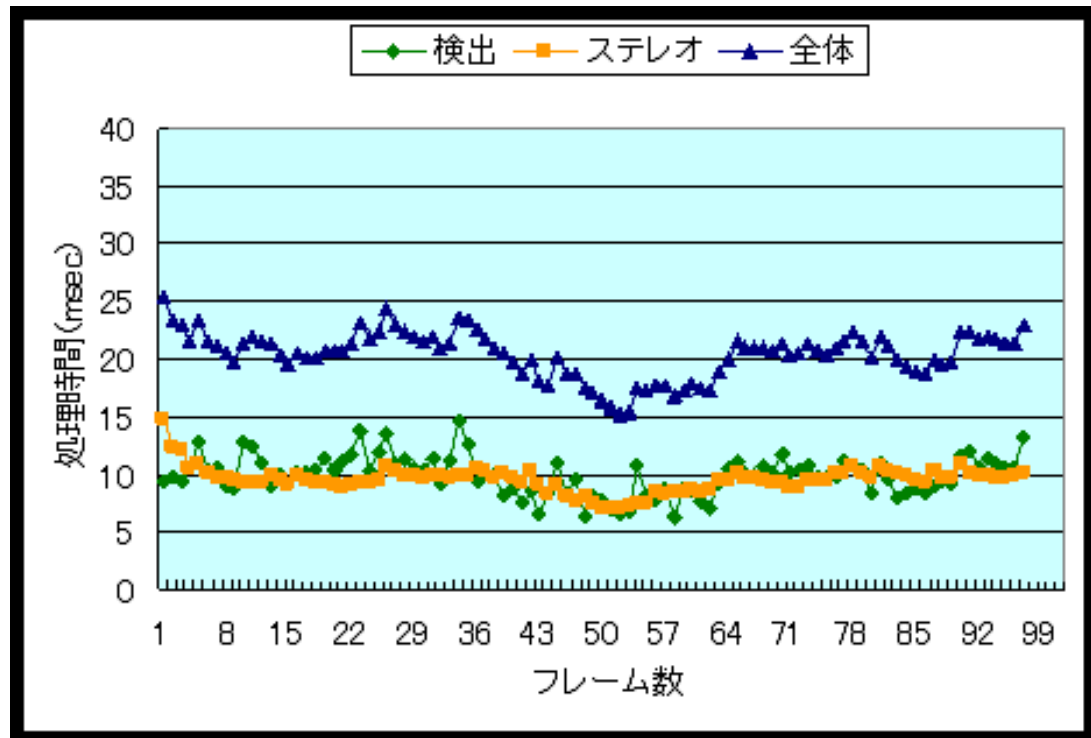


図 2.15 処理速度 系列1 (1人)

図 2.16, 図 2.15 より, ステレオ処理にかかる時間はほぼ一定で 10msec 程度となっており, 検出にかかる時間によって全体の処理時間が変化しているが, ほとんどのフレームにおいて 33msec 以内で全ての処理が終わっている。系列 2 では 2 人の人が映っているため, 系列 1 に比べて探索領域が増え, 検出に要する時間が全体的に多くなっている。

国際会議や国内大会において, 様々な環境で提案手法を用いた顔検出のデモを行ったが, 提案手法での 1 フレームに要する平均処理時間はおよそ 25msec となっている。また, 片方のカメラ画像に対してのみ全探索を行った場合, 識別回数に変化はないため処理時間はほぼ一定で, 1 フレームに要する処理時間は約 70msec であり, 左右カメラ画像に対して単純に全探索を行うと処理時間は約 140msec となる。デモ時は, 図 2.17 のように検出結果と処理時間のグラフをリアルタイムで表示している。処理時間のグラフでは, 縦軸が処理時間をミリ秒で表し, 横軸は処理時間の履歴を表しており, 右端が現フレームの処理時間である。

実験やデモの結果より, 2 枚のステレオ画像では, 提案手法によって全探索に比べ 5 倍



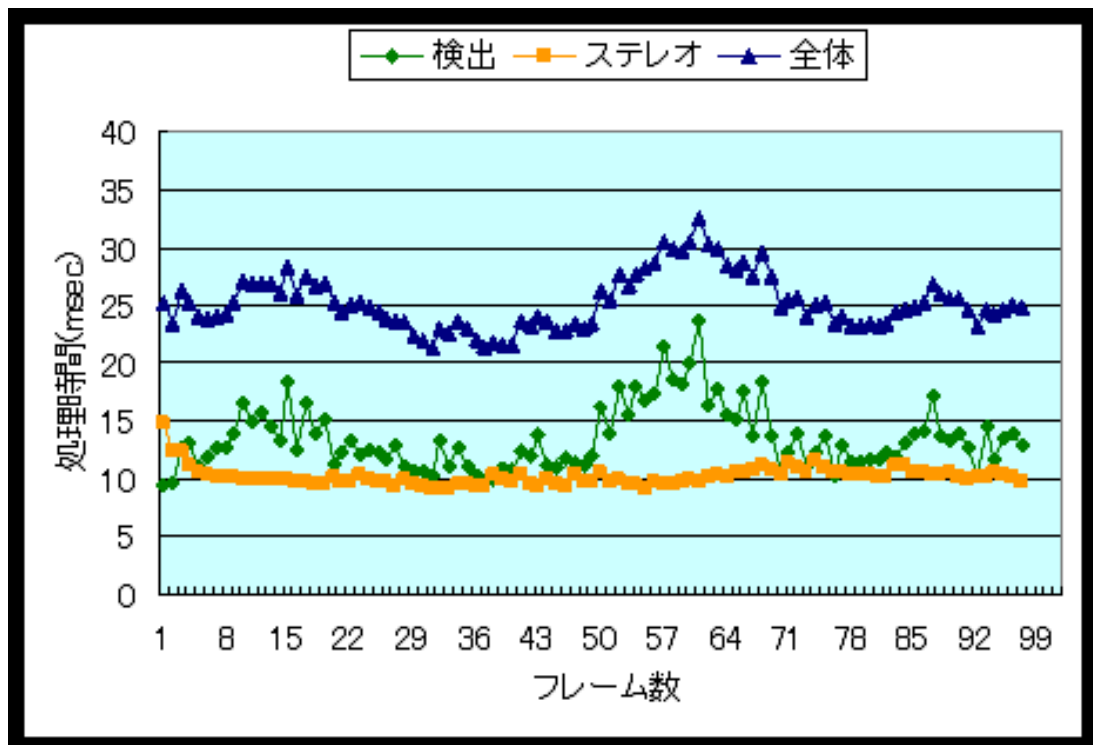


図 2.16 処理速度 系列 2 (2 人)

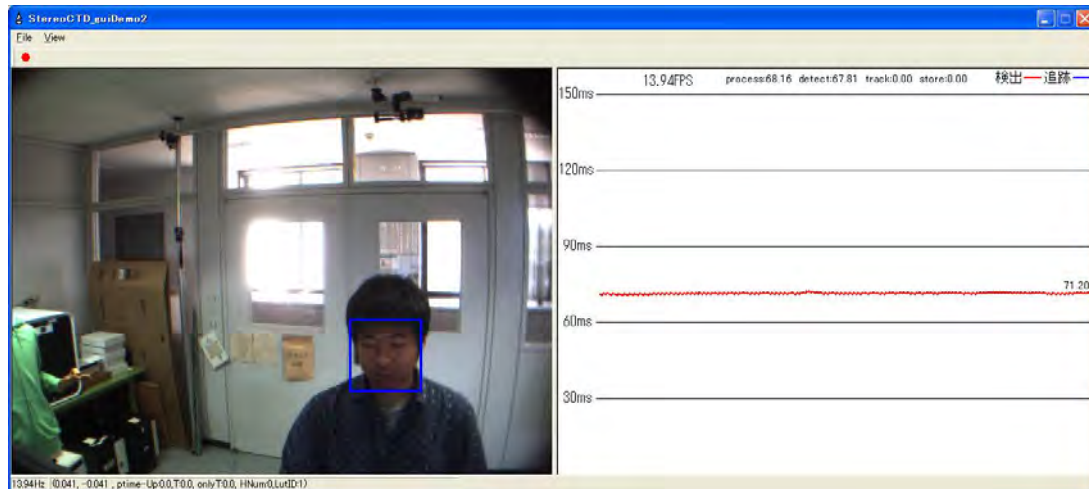
以上の高速化が行える。単眼カメラからの全探索と比べた場合、提案手法ではステレオ画像を探索しながら 3 倍近く高速化された顔検出が行え、ビデオレートでの処理が可能となった。

### 2.3.3 検出精度の実験結果

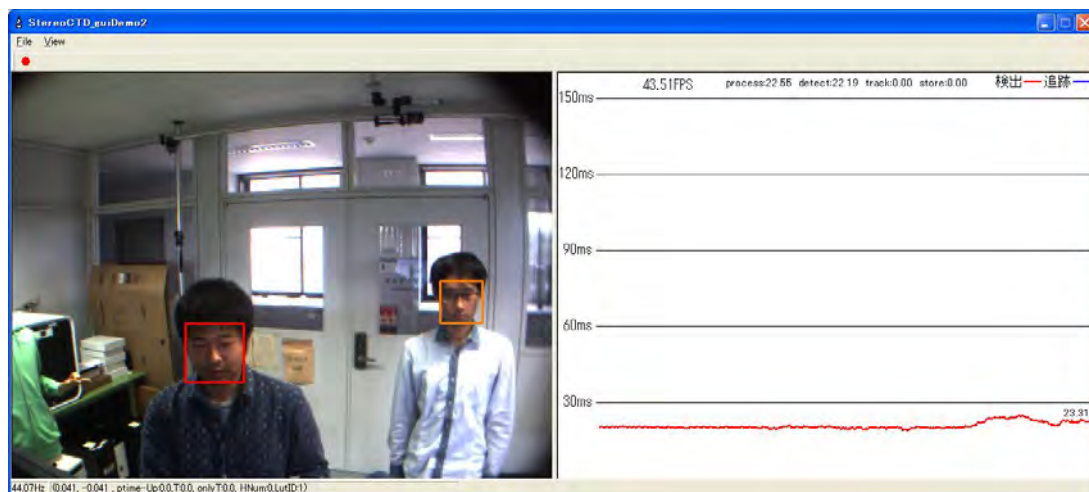
用意した 3 種類の画像系列に対して次の手法、

- 単眼画像からの全探索（右画像を使用）（全探索）
- 提案手法で右画像のみの探索（提案手法 1）
- 提案手法で左右の画像を用いた探索（提案手法 2）

の計 3 通りでの検出実験を行った。系列 3 は一人の人物が移動しているようなもので、系列 4 は映っているのは 1 人で動きが早いもの、系列 5 は 2 人の人物が映っているようなものを用いた。提案手法 1 の場合は、各探索領域において式 (2.3) で求めた  $I_{size}$  とその 1.25 倍の計 2 スケールを探索、提案手法 2 の場合は、右画像で  $I_{size}$ 、左画像でその 1.25 倍の計 2 スケールを探索した。比較実験の結果を表 2.2 に示す。表の数値は 101 フレーム



(a) 片側カメラからの全探索



(b) 提案手法

図 2.17 デモ時のアプリケーション画面

中，顔を検出できたフレーム数を表している．（）内の数値は 101 フレーム中にあった誤検出の合計数を表している．系列 5 では 2 人の人物が映っているため，人物 a，人物 b に分けて検出数を数えた．

表 2.2 より，全ての場合に単眼画像からの全探索に比べると提案手法の検出数が多くなっていることが分かる．単眼画像からの全探索では検出できているのに提案手法で検出できなかった例としては，ステレオ画像から顔までの距離を正確に求められず（例えば，画像の右端の方は視差の関係で正確な距離が出せない）顔を含むような探索領域が正確

表 2.2 手法による検出数の比較

画像系列	3	4	5a	5b
全探索	82(8)	52(20)	54(22)	83(22)
提案手法 1	87	77	55	92
提案手法 2	87(2)	87(7)	55(2)	92(2)

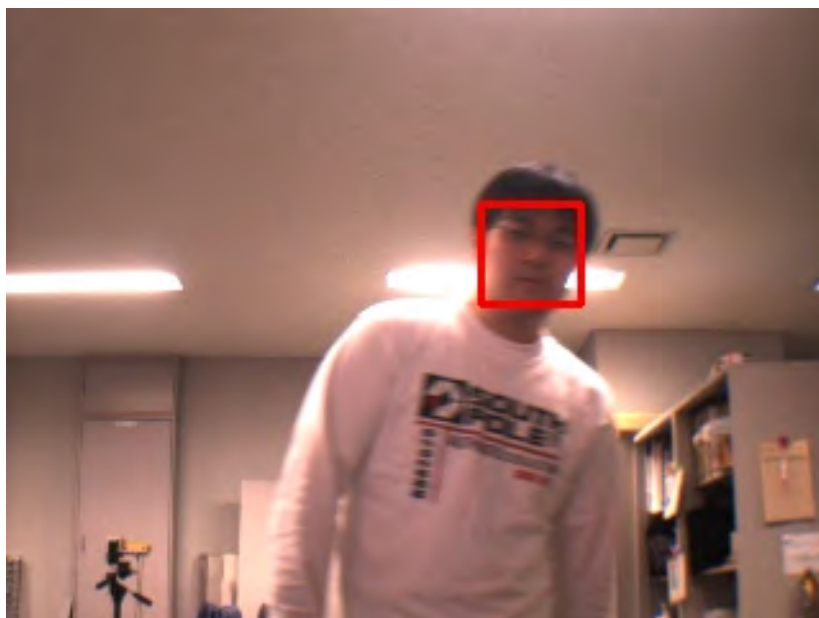
に設定されていない場合などがあった。また、誤検出については提案手法の方が大幅に減っているが、識別回数が削減されたため、それにともなって誤検出も減少している。提案手法 2 は、全探索に比べ検出数が平均して約 20% 向上しており、全探索の誤検出を約 80% 削減できている。

全探索、提案手法 1 を比べると、同じ右画像のみによる探索であるが、提案手法の検出数が多くなっている。単眼画像からの全探索の場合、探索サイズを 1.25 倍ずつ大きくしているため、顔の大きさが 2 つのスケールの間であるような場合に検出されにくいといったことが起こる。一方、提案手法では距離に応じたサイズの探索をしているため、そのときの顔の大きさにあったスケールでの探索となり、全探索のような問題は起こらない。そのため、片方の画像だけでの探索でも、的確なスケールを探索しているため、提案手法では全探索よりも検出率が向上している。

提案手法 1, 2 の検出数を比べると、系列 3, 5 では違いが見られなかったが、系列 4 については検出数が多くなっている。系列 3, 5 では映っている人がほぼ正面を向いていて、動きも早くなく、比較的検出されやすい状況であったため、差が見られなかったと考えられる。一方、系列 4 では検出数が 1 割以上増加しているが、次のような理由が考えられる。図 2.18, 図 2.19 は系列 4 に対してステレオ画像において探索（提案手法 2）を行った結果で、右画像で検出できなかった顔が左画像で検出できた例を示している。まず、図 2.18 については、左右カメラの特性の違いによるものだと考えられる。右画像では顔の後ろに映っている蛍光灯の明かりが膨らんでおり、顔が検出できにくくなっているが、左画像ではそのような現象がなく、検出出来ている。次に、図 2.19 の結果については、左右カメラの視差によるものだと考えられる。左画像では、視差によって右画像よりも、顔がより正面を向いている。そのため、右画像で検出できなかった顔が左画像では検出できている。このように、一方の画像で検出できなかった顔がもう一方の画像では検出できるケースがあるため、ステレオ画像を用いた探索が有効であるといえる。



(a) 右画像



(b) 左画像

図 2.18 ステレオ画像の検出結果例 1



(a) 右画像



(b) 左画像

図 2.19 ステレオ画像の検出結果例 2

### 2.3.4 その他の検出結果

#### 回転した顔

最も単純な顔検出は正面かつ正立の顔の検出であるが、実アプリケーションの中では、顔の回転や向きへの対応が要求される。ここでは、回転した顔に対応した検出実験について述べる。提案手法により検出速度が向上したため、正面だけでなく画像平面内で回転したような顔についても、画像を回転させながら探索することで検出することが可能となる。

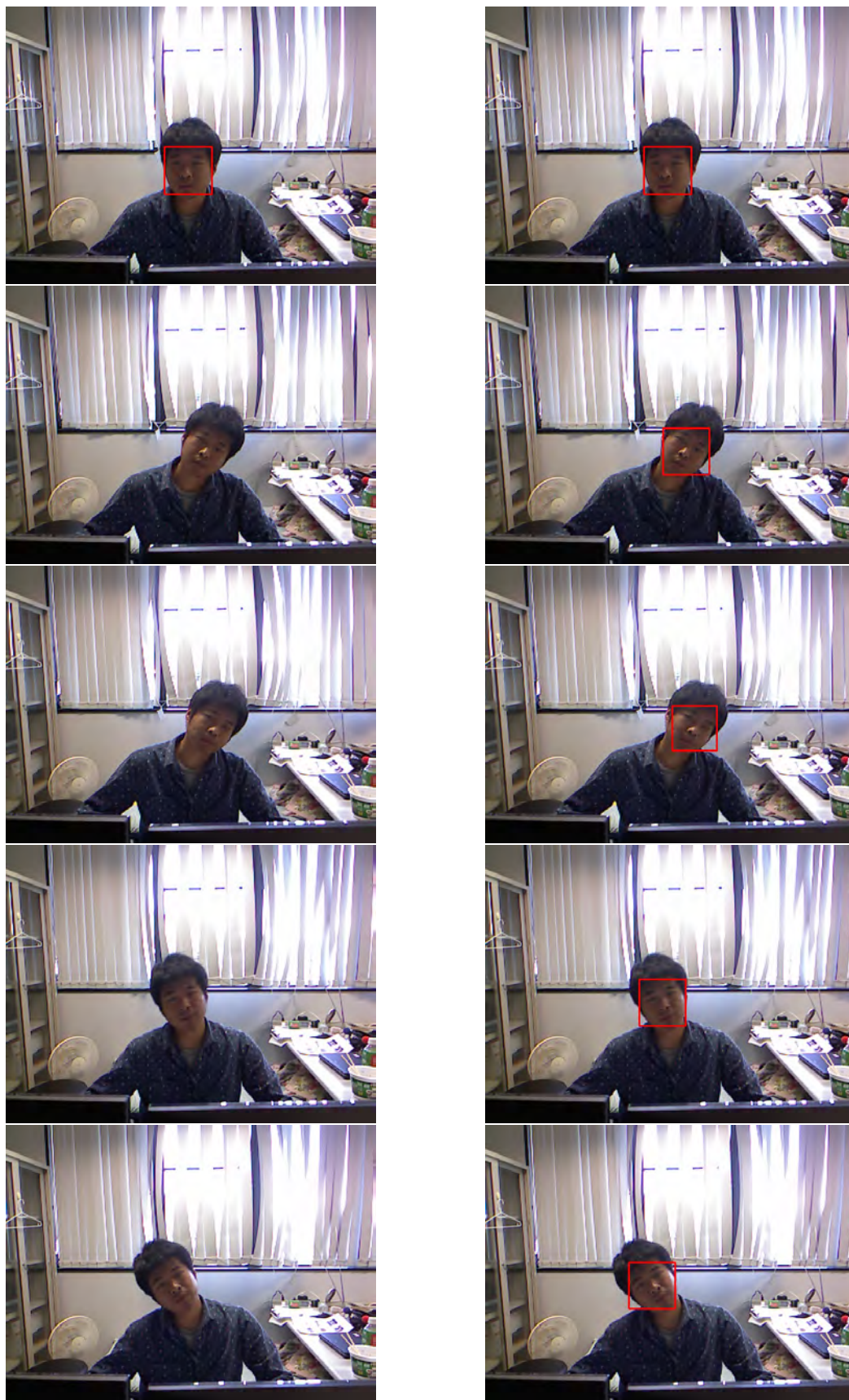
画像平面内で回転した顔を検出する方法には、画像を回転する方法と回転顔に対応した識別器を構築する2通りがある。一般的に画像を回転させる方法では、識別器を1つ構築すればいいという利点があるが、計算量が大きく、誤検出率が高いという問題点がある。提案手法で用いている識別器は正面顔画像のみによって構築されているため正面顔しか検出することができない。しかし、提案手法では計算量、誤検出ともに大幅に削減することが出来たことによって、入力画像を平面回転させる方法で顔の回転対応が可能となった。

画像平面内で顔が回転するような300フレームの画像列を用い、回転対応を行った場合と行わなかった場合の検出結果の一部を図2.20に示す。実験では、回転対応を行う場合、正面と左右20度回転させた画像での探索を行った。また、処理時間を早くするため、1枚の画像を1スケールのみの探索としている。画像を回転する処理を含む全体での処理時間は、1フレームにつき40~50msec程度となっており、ビデオレートでの検出は困難であった。表2.3に300フレーム中の検出数と誤検出数を示す。回転対応により検出数が大幅に増えているが、同時に誤検出数も増えてしまっている。このように、回転対応することで検出できる顔を大幅に増やすことができるが、誤検出数や処理コストも増えてしまうため、これらのバランスを考慮する必要がある。

表 2.3 回転対応時の検出数

	回転対応なし	回転対応あり
検出数	90	238
誤検出数	0	9





(a) 回転対応なし

(b) 回転対応あり

図 2.20 回転顔の検出結果例

### 複数人物の検出

図 2.21 は、3 人の人物が写っている動画シーケンスに対して、提案手法による顔検出を行った結果例である。図では、左側に右カメラからの入力画像と検出結果、右側に入力画像に対する距離画像と探索領域を示している。図のように 3 人の人物が写っているような場合では、それら人物の部分にそれぞれ探索領域が設定されていることがわかる。

### 実物と写真

図 2.22 は、実物と顔写真が写っている動画シーケンス（100 フレーム）に対し、全探索と奥行き情報を利用して探索を行う提案手法を用いて検出を行った比較結果である。

全探索では、顔の大きさにかかわらず検出されているのに対し、提案手法では、実物の顔のみが検出されていることがわかる。顔写真の位置はカメラに近く、この領域で算出された距離値より求められた探索スケールと顔写真のスケールは大きく異なっているため、顔写真は検出されない。提案手法では、全フレームにおいて写真の顔が検出されることはなかった。

このように、奥行きに応じたスケールのみの探索を行う本手法を用いることで、顔写真など実際の顔の大きさと異なる顔を検出せずに実物の人のみを検出することができる。



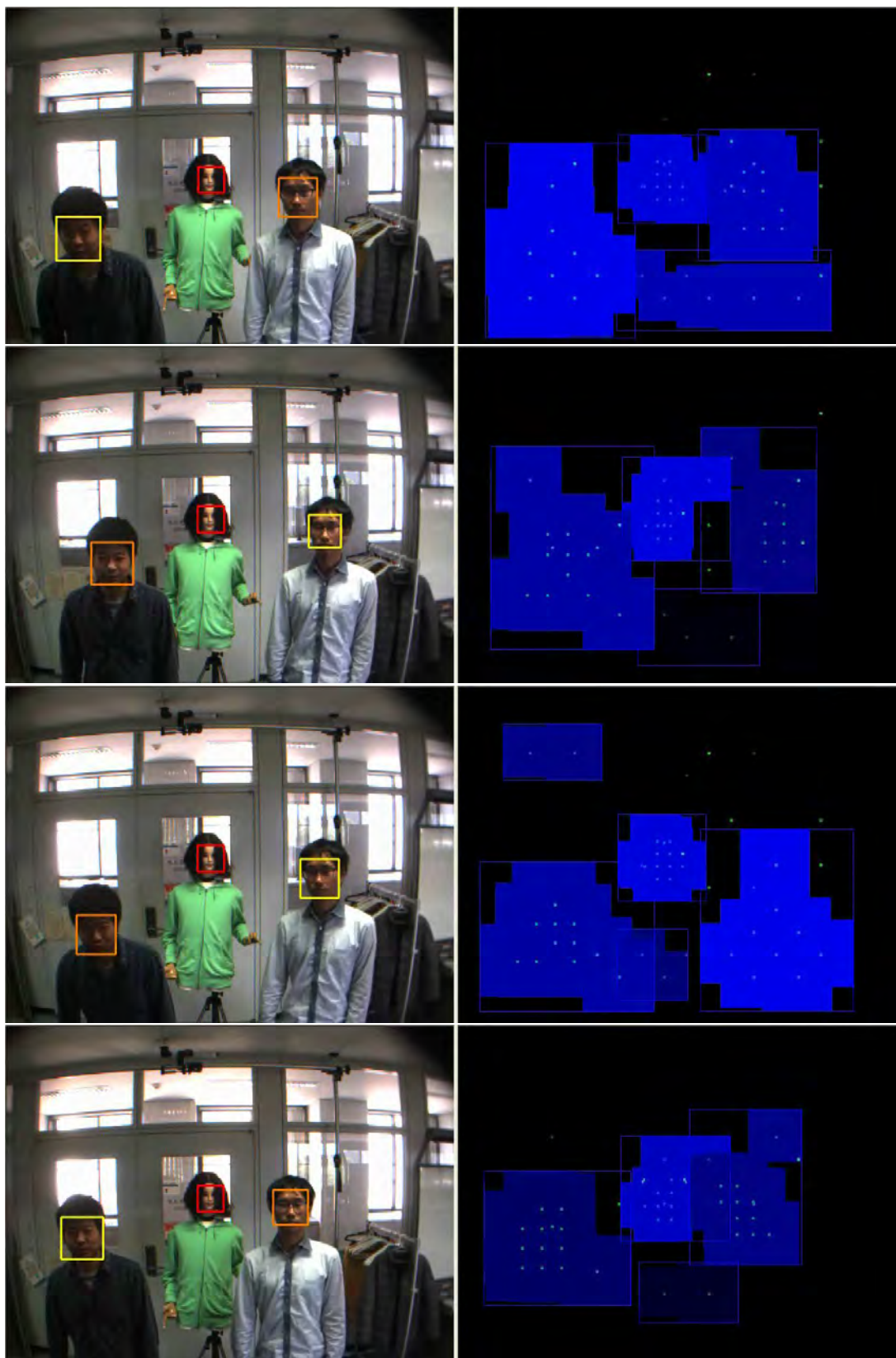


図 2.21 複数人物の検出結果例（左：入力画像と検出結果，右：距離画像と探索領域）



(a) 全探索

(b) 提案手法

図 2.22 実物と顔写真の検出例

## 2.4 まとめ

本章では、顔検出の際の膨大な識別回数を削減するため、ステレオカメラを用いることによって得られる距離情報をもとに探索範囲を限定し、ビデオレートでの顔検出を実現する方法を提案した。また、計算コストが大きいステレオ処理については、全画素ではなくスパースに配置されたサンプル画素についてのみの計算によって処理時間を抑制する方法を提案した。提案手法によるステレオカメラを用いた顔検出は、単眼カメラからの全探索による顔検出よりも検出率を向上させながらビデオレートでの処理を実現した。

実画像系列を用いた実験や様々な環境におけるデモによって、処理時間がビデオレート内に収まっていることを確認し、また、各領域の距離情報に応じた適切なスケールでの探索とステレオ画像を探索することで検出精度が向上することを確認し、提案手法の有効性を示した。

本手法で行っている奥行き情報を利用した識別回数の削減による検出の高速化手法は、既存の識別器を用いた検出を高速化させるものであり、単に顔検出だけでなく、ピラミッドスキャンを行っている一般的な物体の検出を高速化することができ、コンピュータビジョンやロボットビジョンの研究において役立つことが期待される。

## 第 3 章

# 顔検出と色弁別度追跡法を組み合わせた顔追跡システム

本章では、第 2 章にて提案した高速顔検出法と色弁別度追跡法を統合することで、自動的に追跡が開始できる高速かつ安定な顔の検出・追跡システムの構築方法について説明する。顔検出を毎フレーム行うことで顔領域を取得することは可能であるが、正面顔画像のみで構築された識別器では顔がカメラに正対していなければ検出することができず、継続的に顔領域を取得することができない。また、提案システムでは、奥行き情報から画像上での顔の大きさを推定することで顔領域の決定を安定化させ、顔検出で取得した領域からターゲット（顔）と背景（非顔）の色ヒストグラムを構築し、背景には多く含まれず追跡しやすいターゲット色を選びながらモデルを自動的にアップデートすることで、照明変動や周辺環境の変化に頑健な追跡を実現する。

本章では、まず、色弁別度追跡法についての概要を説明し、先行研究において構築された顔検出・追跡システムに残された課題を解決しながら提案システムの目標を達成する。

### 3.1 色弁別度追跡法

本節では、我々の研究グループがこれまで開発してきた色弁別度を用いた対象の検出・追跡手法 [94, 95, 96, 99, 98, 100] について説明する。また、検出処理と追跡処理を統合したシステムについても説明し、特徴や問題点を述べる。

弁別度は、画素の色（対象色と非対象色）に基づく対象の検出と追跡に共通して用いることができ、単に対象に対する類似性だけでなく、非対象に対する相違性を同時に評価する値となっている。弁別度は、 $[0,1]$  の実数値を取り、最近傍識別境界上の色に対して

ちょうど 0.5 の値となるため、最近傍識別と密接な関わりを持っている。

和田 [100] によって提案された最近傍識別を用いた色ターゲット検出では、各画素を二値的に識別している。そのため、対象の検出に失敗すると追跡ができなくなる。これに対し、飯塚ら [94, 98] は弁別度を提案し、最近傍識別による対象検出を対象追跡に拡張した。これにより、弁別度を全画素に適用して得られる「弁別性マップ」を用いることで、色に基づく対象検出だけでなく、検出できない場合でも弁別度の高い領域を追跡することができる。また、可変テンプレートマッチング法や Mean Shift 法に比べ、様々な条件下で頑健に対象を追跡できることが示されている。本研究では、飯塚らが提案している追跡手法を取り扱い、以下、その概要を説明する。

### 3.1.1 弁別性マップの構築

対象検出の問題では単に対象を検出すればよいだけであるが、対象追跡の問題では、仮に非対象と識別されても、どの程度対象と近い色をしているのかを知る必要がある。対象色らしさが計算できれば、対象として尤もらしい画像領域を求めることができ、追跡を行うことができる。

この対象色らしさを表す量を「弁別度」、全画素の弁別度を画像にしたものを「弁別性マップ」と呼ぶ。色ターゲット検出では、対象と非対象（背景）を表す色のデータをプロトタイプとして記憶しておき、入力された色がどちらのプロトタイプ集合に近いかによって最近傍識別を行っている。これに対して、弁別度は対象・非対象のプロトタイプ集合と入力との距離をそれぞれ求め、これら 2 つの距離から計算される量であり、以下の性質を持つ。

1. (0,1) の範囲の実数値である。
2. 0.5 がちょうど最近傍識別での識別境界となる。
3. 対象プロトタイプ上で最大値 1 となる。
4. 非対象上で最小値 0 になる。
5. 任意の非対象プロトタイプから最近傍の対象プロトタイプに至るパス上の色で値が単調に増加する。

このため、各画素の弁別度を調べることで対象の検出ができ、さらに弁別度の高い部分を連続する画像間で対応付けることで、追跡を行うこともできる。

弁別度  $d$  は次式のように計算できる。

$$d = \frac{D_{non-target}}{D_{target} + D_{non-target}} \quad (3.1)$$



但し,  $D_{target}$  は色空間内で対象色として与えられた最近傍ターゲットプロトタイプと入力画素値との距離,  $D_{non-target}$  は非対象色として与えられた最近傍非ターゲットプロトタイプと入力画素値との距離を表す (図 3.1). 式 (3.1) の分子に非対象色との距離を入れることで, 弁別度が類似度だけではなく相違度を統合して表すことができる. 弁別度は, 対象追跡を行う際の評価尺度として用いることができるだけでなく, 最近傍識別器による色ターゲット検出という優れた対象検出も同時に行える画像特徴となっている.

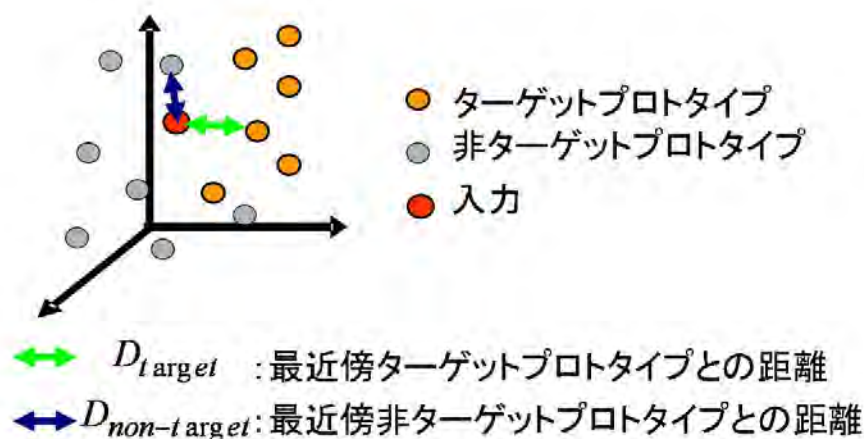
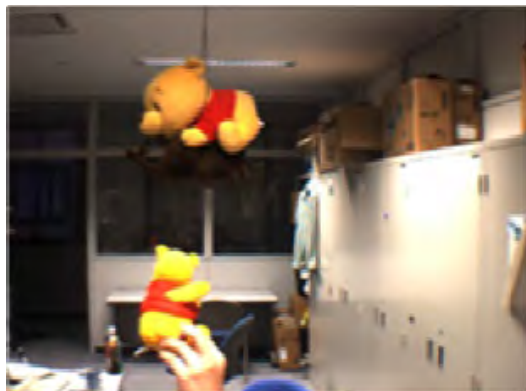


図 3.1 色空間内での距離表現

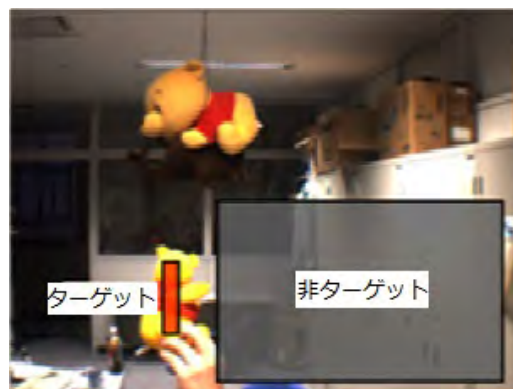
図 3.2 に弁別性マップを構築した例を示す. 図 3.2(b) は, 入力画像に対して手動で教示を行っている様子を示しており, 赤色の矩形領域をターゲット, 灰色の矩形領域を非ターゲットとしてそれぞれ指定している. 図 3.2(c)(d)(e) では, 弁別度を 0~255 の値に拡大して表示している. ターゲットとして教示したぬいぐるみの領域が顕著に現れていることがわかる. また, 図 3.2(c) では, 背景にあるぬいぐるみ部分は非ターゲットとして教示していないため弁別度が高くなっているが, 図 3.2(d) に示すような追加教示により背景のぬいぐるみ領域の弁別度が抑えられていることがわかる (図 3.2(e)).

### 3.1.2 弁別度算出の高速化

弁別性マップを構築する際, 入力画像の全ての画素において式 (3.1) を適用して弁別度を算出すると, 処理に要する時間が多くなってしまう. これでは, 追跡システムのように実時間性を求められるシステムに適用することが難しくなる. そこで弁別度 LUT (Look Up Table) を用いることで弁別度算出の高速化を行う. 図 3.3 に弁別度 LUT の構築手順を示す.



(a) 入力画像



(b) 教示範囲



(c) 教示範囲 (b) により得られた弁別性マップ



(d) 追加教示範囲



(e) 追加教示後の弁別性マップ

図 3.2 弁別性マップの算出 [98]

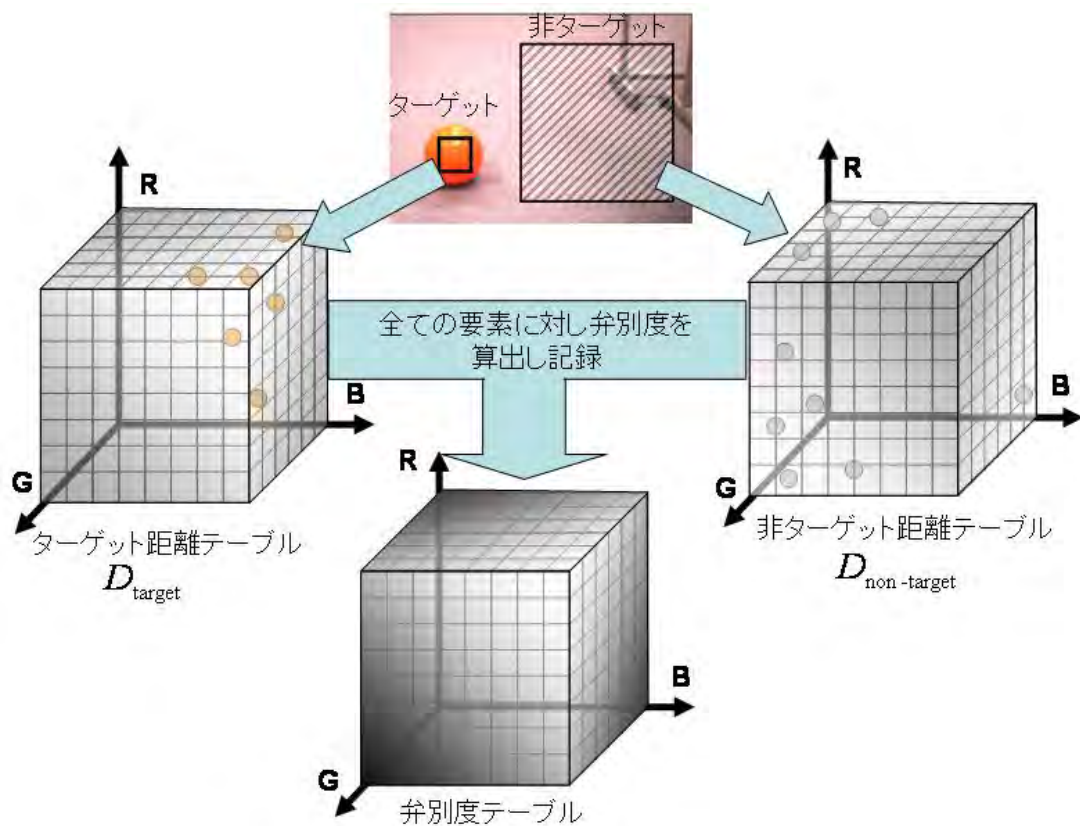


図 3.3 Look Up Table の構築手順 [98]

弁別度 LUT を構築するには、まずターゲット・非ターゲットをそれぞれのトレーニングパターンに対して色空間内で距離変換を行い、それぞれの結果を距離テーブルに書き込んでいく。この距離変換は文献 [100] で示したように横型探索で行うことにより、ほぼ距離テーブルの要素数に比例した時間で学習が終了し、色の追加・削除は局所的な距離の書き換えで済むようになる。構築されたターゲット・非ターゲット距離テーブルの各要素はプロトタイプ集合までの距離を表している。つまりターゲット距離テーブルは  $D_{target}$ 、非ターゲット距離テーブルは  $D_{non-target}$  を表していることとなり、この2つの距離テーブルをもとに量子化された色空間すべてに式 (3.1) を適用することで弁別度が格納された弁別度 LUT が構築できる。したがって、この弁別度テーブルを参照するだけで弁別度が算出できることとなり、高速に弁別性マップの構築を行うことができるようになる。



### 3.1.3 弁別性マップを用いた対象追跡

対象追跡では，画像内の追跡対象の大きさや，回転，形状などの変化に適応するため，対象を可変楕円によって表現し，この可変楕円によって画像中の対象領域の限定と位置の同定を行う．図 3.4 に示すように，推定された対象色の画素を 9 割以上囲むように可変楕円パラメータを決定する [106]．

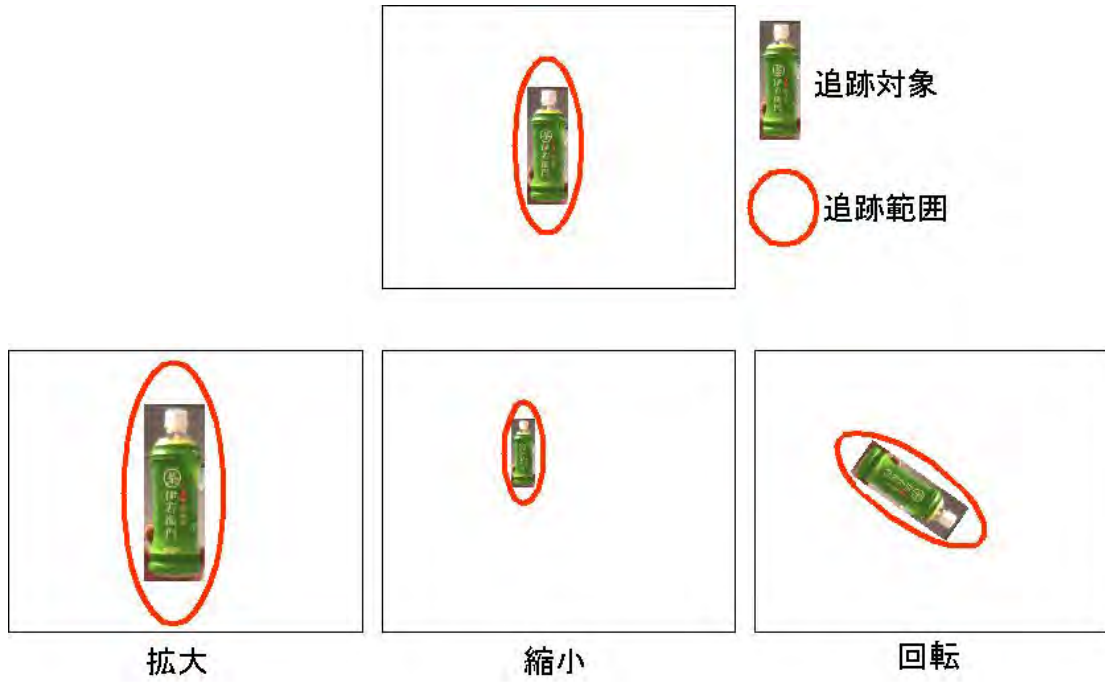


図 3.4 対象の見え方の変化 [98]

弁別性マップは各画素に弁別度を持つ 2 次元分布であり，分散共分散行列  $M$  を用いることで弁別度の高い部分にフィットする楕円を求めることができる．時刻  $t-1$  での分散共分散行列を  $M_{t-1}$  とすると時刻  $t$  での対象の存在しうる範囲  $S$  を

$$S : (\mathbf{x} - \bar{\mathbf{x}})^T M_{t-1}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) < K \quad (3.2)$$

と表現することにする．ただし， $\mathbf{x}$  は弁別性マップの画素位置  $\mathbf{x} = [x, y]^T$ ， $\bar{\mathbf{x}}$  は弁別度の重心位置， $K$  はターゲット画素の含有率で楕円サイズを決定する任意定数である．この  $\bar{\mathbf{x}}$  を更新していくことで対象の追跡が行える．この更新は次式で行う．

$$\bar{\mathbf{x}} = \frac{\sum_{\mathbf{x} \in S} \mathbf{x} d(\mathbf{x})}{N} \quad (3.3)$$

ただし,

$$N = \sum_{\mathbf{x} \in S} d(\mathbf{x}) \quad (3.4)$$

ここで,  $d(\mathbf{x})$  は  $\mathbf{x}$  における弁別度であり, 弁別度を重みとした加重平均で表される. また, 時刻  $t$  における分散共分散行列  $M$  は次式で求める.

$$M_t = \frac{1}{N} \sum_{\mathbf{x} \in S} d(\mathbf{x})(\mathbf{x} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \quad (3.5)$$

弁別度の重心と分散共分散行列の更新を繰り返すことにより, 対象追跡を行う.

更新された弁別度の重心が可変楕円の中心となり, 追跡対象の中心位置となる. 時刻  $t$  における分散共分散行列  $M_t$  から可変楕円の長軸, 短軸, 傾きを求め, 探索範囲をこの可変楕円に限定することで追跡中の背景混入を少なくすることができるため, 追跡対象に似た色が近づいても頑健な追跡が可能である.

### 3.1.4 検出・追跡システム

前述したように, 弁別度は対象追跡に用いることができる対象らしさを求めるだけでなく弁別度を 0.5 となる場所で区切り, ターゲットと非ターゲット画素を分けることで最近傍識別器によるターゲット検出を行うことができる. 検出処理と追跡処理を統合し, これらの処理を適宜切り替えることでターゲットの検出・追跡システムを構築することができる. 検出・追跡の成功と失敗判定は以下のようになっている.

#### 検出の成功・失敗判定

弁別度を用いた対象検出を行う場合, 検出されたターゲット画素の密度計算により対象領域を求める. ある矩形領域内のターゲット画素数の割合が閾値以上である場合に検出成功と見なし, そうでない場合は検出失敗とする. この時, 検出成功領域が複数見つかった場合は検出画素数が最大の領域を検出成功領域とする.

#### 追跡の成功・失敗判定

前フレームで検出された領域の重心位置, もしくは前フレームでの追跡結果の位置をもとに弁別度を用いた対象追跡を行う. 対象追跡により求められた追跡範囲である楕円内の弁別性マップの密度計算を行う. 前フレームの密度計算結果 (但し, 対象検出成功直後の場合は検出領域の密度) と現在のフレームの密度計算の結果を比較し, 急激に密度が低下した場合 (閾値を設定し大小比較で判定) は追跡に失敗とする.

### 3.1.5 従来手法のまとめと残っている問題点

本節では、色弁別度を用いた対象の検出・追跡について概説した。この手法は、色に基づく対象検出と追跡の両方の処理で用いることができる弁別性マップを構築し、Gaussianの当てはめによって対象の追跡を行っている。また、原画像ではなく弁別性マップに変換してから追跡を行っているため K-means Tracker と同様の背景混入に対する頑健性を持っている。そのため、ワイヤオブジェクトなど隙間の多い対象でも追跡が可能であり、オクルージョンに対しても頑健である。さらに、事前に得られたモデルと入力画像の色ヒストグラム同士のマッチングによる類似性評価を行っていないため、Mean Shift 法などで問題となる単峰性ヒストグラムを持っている単色物体に対する脆弱性もない。

一方、問題点として、検出・追跡システムを顔領域に適用した場合、あらかじめ人の肌色情報と周辺の背景情報の教示を手動で行なっているため、色ターゲット検出の際に顔だけでなく手など教示した色と似た色が画像中に入ってくると誤ってターゲットとして検出されてしまうことがある。追跡中の問題点として、対象と近似色の背景が追跡範囲に完全に隣接してしまうと追跡対象周辺の弁別度が高くなり、追跡範囲が広がってしまうといったことがあり、対象までの距離と楕円サイズを対応付けることで解決できることが示唆されている。また、時間の経過やパン・チルトカメラの撮影方向の変化による照明環境や背景の変化、肌色の人種・個人差などで事前に教示した色と追跡対象・背景色が異なってしまう、対象上での弁別度が低くなると、対象検出ができなかったり、検出に成功したとしても追跡が安定に行えなかったりするという問題がある。

本章では、これらの問題を解決する方法を提案し、以降、詳細を説明する。

## 3.2 適応的カラーモデルを用いた顔検出・追跡システム

色弁別度追跡法は、色に基づく弁別性マップを構築し、弁別度の高い領域に楕円フィッティングすることで追跡を行う手法である。画素毎に弁別度を算出するため、ワイヤオブジェクトの追跡やオクルージョンに対する頑健性を持っている。また、テーブル化することで高速化した弁別度算出によって弁別性マップを構築しているため、非常に高速な追跡処理が行える。また、弁別度は最近傍識別によるターゲット検出としても利用でき、検出処理と追跡処理を統合することでターゲットの検出・追跡システムを構築している。先行研究 [98] におけるターゲット検出・追跡システムの処理手順を図 3.5 に示す。

先行研究では、顔の追跡を行う場合、あらかじめ肌色情報（ターゲットプロトタイプ）を教示し、ターゲット色に基づく検出を行ってため、教示した色と似た色が画像中に入ってくると誤ってターゲットとして検出し、追跡されてしまう問題がある。提案手法では、肌色情報ではなく第2章にて提案した高速な顔検出法を用い、入力画像中から顔だけを正確に検出することでこの問題を解決する。

また先行研究では、照明環境の変化や人種、個人差による肌色の違いなどによって、あらかじめ教示したターゲット色と異なる追跡対象者を検出できなかったり、検出に成功したとしても追跡が安定に行えない問題がある。提案手法では、検出された顔領域に基いて、検出矩形領域内の画素をターゲット（肌色）プロトタイプ、矩形領域外を非ターゲット（背景色）プロトタイプとして自動的に色の学習を行うことができる。プロトタイプの学習を自動化することで、提案システムでは起動時に設置環境に合わせたカラーモデルを学習する煩わしさもなくなる。

さらに追跡中の安定性に関して、背景に対象と非常に似ている色が現れ対象と隣接してしまうと、追跡対象周辺の画素も高い弁別度を持ち、追跡範囲である楕円が広がりすぎてしまうという問題がある。これについては、対象までの距離と楕円サイズを対応付けることで解決できることが示唆されているが、追跡が安定した状態でこの対応付けを行うフレームを手動で決定する必要がある。提案手法では、人の顔検出・追跡問題に特化しているため、顔の大きさを予め仮定しておく、ステレオ視によって得られた奥行き情報から画像上での顔の大きさを推定することができる。この推定された顔サイズにより、追跡楕円がこれよりも大きくならないように制限することで、楕円が広がりすぎることなく安定な追跡が行える。また、先行研究では弁別度 LUT の更新を行っていないため、照明変動や背景の変化に適応できないのに対し、提案手法では、顔検出が成功した際に随時、LUT のアップデートを行うことでこの問題を解決する。

提案手法の要点は次のようになっている。また、提案手法の処理の流れを図 3.6 に示す。以下、提案手法について詳しく説明していく。

- 識別器を用いた顔検出と弁別度追跡を組み合わせ、顔だけを正確に追跡する
- 顔検出で得られる顔領域に基づいて構築された色ヒストグラムによって、適応的な弁別度 LUT の構築と更新を行う
- 顔までの距離に基づいて画像上での大きさを制限し、追跡を安定化する

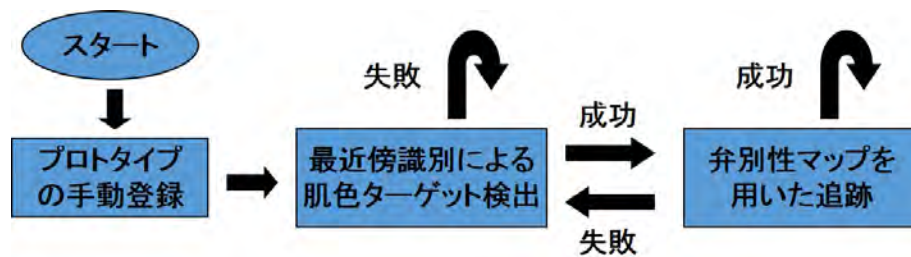


図 3.5 従来手法におけるターゲット検出・追跡システムの流れ

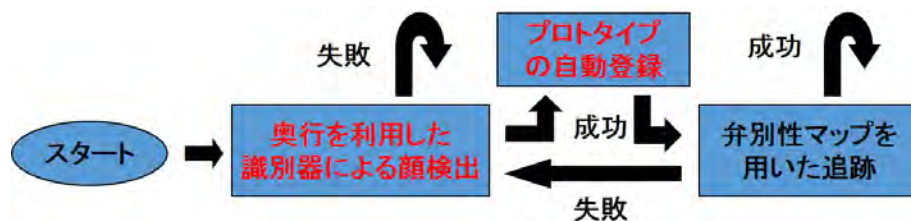


図 3.6 提案手法における顔検出・追跡の流れ

### 3.2.1 プロトタイプの自動登録による適応的カラーモデルの構築

本項では、第 2 章で提案した識別器による顔検出で得られた顔領域をもとに、顔追跡に用いる弁別度 LUT を構築するためのカラープロトタイプを自動学習する方法について述べる。

3.1 節で述べたように、先行研究 [98] では予めターゲットプロトタイプ、非ターゲットプロトタイプを学習した上で対象の検出・追跡を行っていた。そのため、システムの設置環境にあわせた学習を毎回行う必要があり、対象人物の人種によってターゲット色（肌色）が変わる場合や照明変動が起こる度にも手動で再学習を行う必要があった。提案システム

では、顔識別器により画像中から正確な顔領域が得られるので、得られた顔領域を基に自動的に LUT (Look Up Table) の学習を行うことで、その時の照明環境、ターゲットに合わせたカラーモデルを構築することができる。

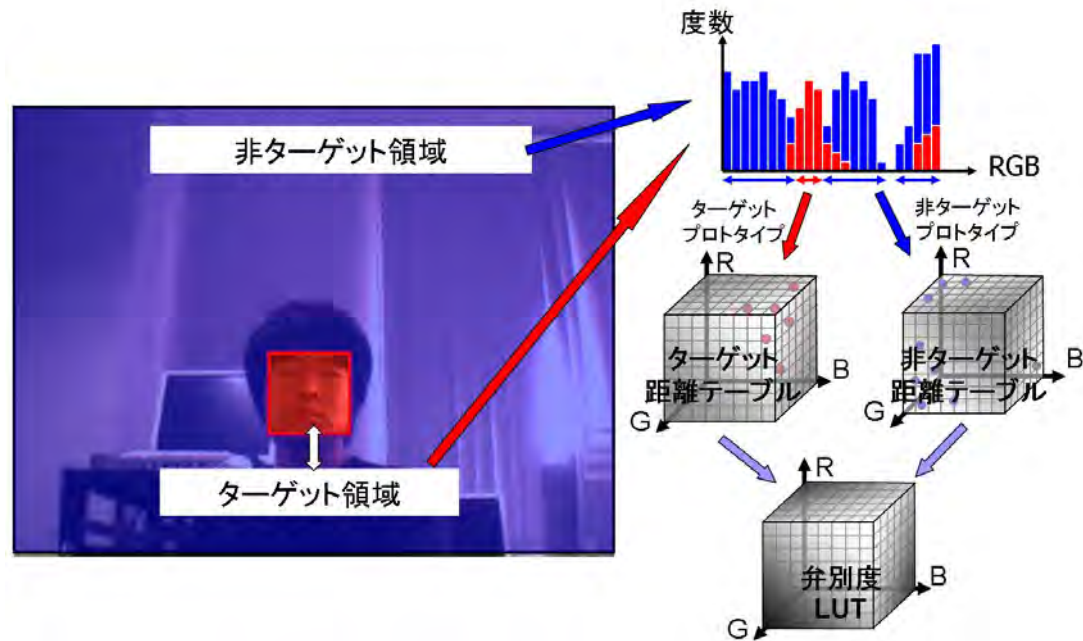


図 3.7 プロトタイプの自動学習方法

図 3.7 に示すように顔が検出（図の赤色矩形）された時、従来手動で指定する時と同様に単純に顔検出で得られた矩形の枠内をターゲットプロトタイプ、顔検出で得られた矩形の枠外を非ターゲットプロトタイプとしてしまうと、非ターゲット領域にターゲットと似た色がある場合にターゲット色まで背景と見なされてしまうことや、ターゲット領域に背景に多い色がある場合に背景の弁別度が高くなり、安定に追跡できないといったことが起こりえる。そこで提案手法では次のように学習用プロトタイプを自動教示する。

プロトタイプの登録から弁別度 LUT 構築までの処理の流れは図 3.7 のようになっている。まず、検出枠の矩形領域内（赤色領域）をターゲット（顔）領域、それ以外（青色領域）を背景領域とし、各領域ごとに、領域内の画素値を色空間に投票して、ある色がいくつあるのかを表す度数ヒストグラムを作る。図 3.7 右上のヒストグラムの横軸が色空間、縦軸が各色の度数を表している。次に、構築されたヒストグラムにおいて、ターゲットと非ターゲットの度数を比べ、度数が大きい方をそのプロトタイプとして距離テーブルに登録する。つまり、非ターゲット領域には少なくターゲット領域に多い色がターゲットプロトタイプとなり、残りの色が非ターゲットプロトタイプとなる。図 3.7 の右中、ヒストグ

ラムの下に赤色矢印がターゲットプロトタイプ、青色矢印が非ターゲットプロトタイプを表している。最後に、3.1 節で述べたように、2つの距離テーブルから弁別度 LUT を構築する。

このような学習を行うことで、できるだけ多くの背景色を教示しつつ、背景には少ないターゲット色を学習することができるため、安定した追跡が行える。

### 3.2.2 奥行き情報と適応型カラーモデルを用いた追跡の安定化

本項では、距離情報を用いた追跡範囲（可変楕円）の制約や、弁別度 LUT の自動更新（適応型カラーモデル）によってより安定な追跡を行う方法について述べる。

#### 追跡範囲の制約

先行研究 [98] でも示されているように、追跡時、背景に対象と似ている色が現れ対象と隣接してしまうと、追跡範囲である楕円が広がりすぎてしまうという問題がある。先行研究では、ステレオ追跡によって対象までの距離を測ることができる利点を活かし、対象までの距離と楕円サイズを対応付けることでこの問題を解決できることが示唆されているが、追跡が安定している状態でこの対応付けを行うフレームを手動で決定する必要がある。提案手法では、人の顔の大きさに大きな個人差が無いことを利用し、ステレオ視によって得られた距離情報から画像上での顔の大きさを推定し、楕円サイズが画像内の顔サイズと同程度となるように制限することでこの問題を解決する。

ステレオカメラで追跡を行っているため、左右の画像でそれぞれ追跡対象の中心位置（楕円の中心）が得られる。このとき得られた位置は左右でずれており、この視差を用いることで式 (2.2) と式 (2.3) により追跡中の顔の大きさを算出することができる。ここで得られた顔サイズを用いて式 (3.2) の  $K$  を設定することで、追跡楕円が顔サイズ以上に広がらないようにすることができる。

#### 弁別度 LUT の更新

3.2.1 項で述べたように、プロトタイプの自動登録により顔検出時の照明環境やターゲットと非ターゲットの色に適したカラーモデルを構築することができる。しかし、Pan-Tilt Unit を用いた能動システムであるため、追跡中に未学習の背景が入ってくることがあり、ターゲット色と似ていた場合に追跡が不安定になることがある。本研究では、追跡中に弁別度 LUT を構築しなおすことによってこの問題を解決できると考えられる。

弁別度 LUT の更新を行うには、正確なターゲットの領域を知る必要がある。例えば追



跡が失敗し、追跡楕円が対象から外れてしまったときに楕円内をターゲットプロトタイプとして登録してしまうと、誤った弁別度 LUT が構築されてしまう。そこで提案手法では、追跡時にも顔検出を行い、顔が検出された時のみ更新処理を行う。

追跡中の顔検出は前フレームの追跡対象（顔）の周辺領域のみで行い、また前述したように対象までの距離から顔サイズが求められるため、そのスケールのみでの探索を行う。対象までの距離は左右カメラの追跡楕円の中心のずれから計算でき、追跡対象の周辺領域だけを1つのスケールのみで探索を行うため、ほとんど計算コストのかからない高速な顔検出ができる。検出が成功した場合、3.2.1 項で説明した処理と同様にプロトタイプの追加登録を行い、弁別度 LUT を再構築する。このとき、再学習の計算コストを抑えるため、非ターゲット領域はターゲット領域の2倍程度としている。

### 3.2.3 提案システムの構成

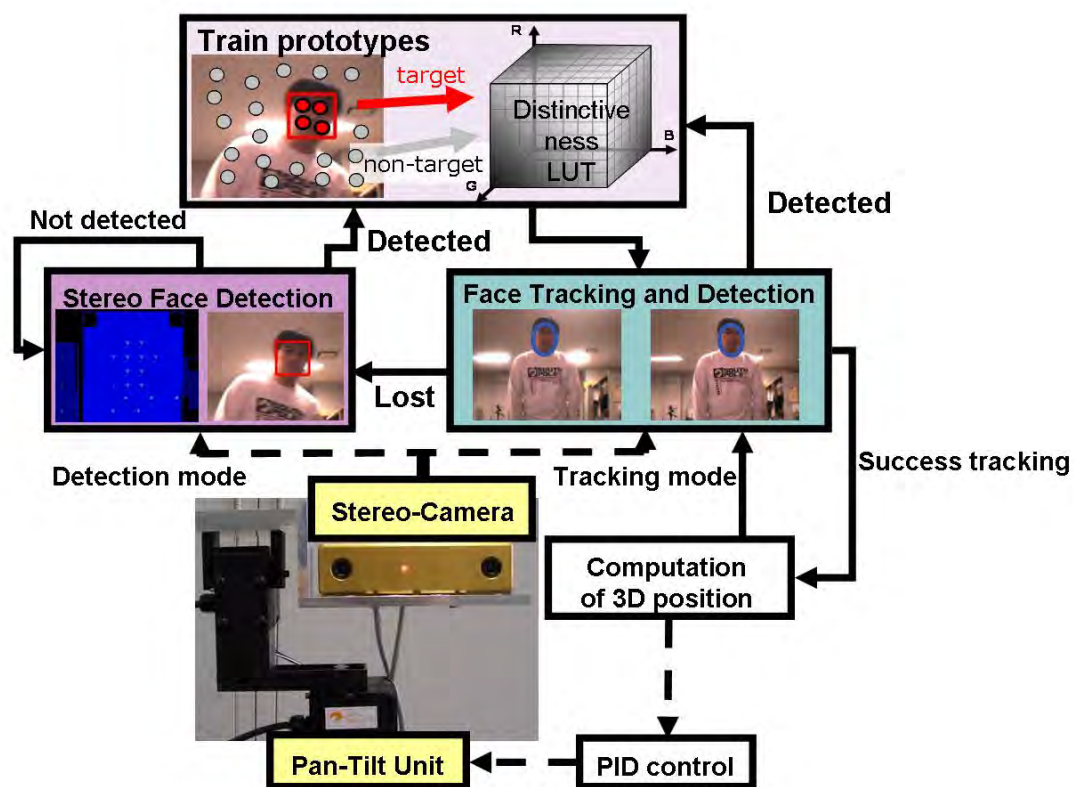


図 3.8 提案システムの構成

提案手法を用いて、能動ステレオカメラを用いた顔追跡システムを構築した。ステレオ



カメラが図 3.8 に示すように PTU 上に搭載されており，追跡対象を画像中心に捉えるよう PID 制御により自動的に追尾撮影を行う．PID 制御は「鮮明な画像撮影のための高速追従カメラシステム [103]」で用いられている方式を Pan-Tilt Unit の速度制御方式に採用している．システム構築に使用している機材は，計算機が Intel Core i5-750 の CPU を搭載した PC，ステレオカメラが PointGrey Research 社の Bumblebee2，Pan-Tilt Unit (PTU) は Directed Perception 社の model PTU-46-17.5 である．このカメラは 2 眼式平行ステレオカメラでフレームレートは 50fps である．ステレオカメラを構成しているカメラのキャリブレーションは出荷時に行われており，IEEE1394 インターフェースの搭載されている PC に接続するだけで，簡単かつ高速に，ステレオ視により 3 次元位置情報を得ることが可能である．撮影された画像はベイヤーターンで出力され，これを PC 内部で RGB カラーに変換して用いる．高速に弁別度を参照するために，LUT のサイズがキャッシュに収まるよう，色空間を  $64 \times 64 \times 64$  に量子化している．

図 3.8 に示すように提案システムでは，まず，顔検出モードからスタートして顔を検出する．顔が検出されると弁別度 LUT の構築と追跡初期位置が設定され，追跡モードへと移行する．また，追跡中にも顔検出が行われ，検出成功時に弁別度 LUT の更新が行われる．追跡が失敗すると再び顔検出モードへと切り替わる．追跡の失敗判定は，3.1 節で述べたものと同様の処理で行っている．また，識別器を用いた顔検出を行うことによって正面顔のみを検出するため，本システムに興味のある人を検出・追跡し，興味のない人，つまり，システムの前を素通りする人は検出・追跡しないというインタラクションを実現している．

### 3.3 実験

#### 3.3.1 従来システムとの比較実験

検出方法とプロトタイプ登録方法の違いによる，従来手法 [98] との比較実験を行った．以下，それぞれの結果を示し，提案手法の有効性について述べる．なお，PTU の制御は行わずに実験している．

実験に用いた画像は，ステレオカメラから撮影された  $640 \times 480$  画素の画像系列となっている．処理時間について，図 3.9 に顔検出と追跡を単独で行った場合の各フレームにおける処理時間をグラフにしたものを示す．グラフは，50FPS で撮影された約 1 分間の結果であり，赤色の線が検出，青色の線が追跡の処理時間を表している．検出の前半部分（左側）はピラミッドスキャンによる顔検出の結果で約 70msec，後半部分（右側）は第 2 章で述べた提案手法による顔検出で約 20msec，追跡処理に要する時間は 1msec 程度である．また，色ターゲット検出は約 5ms，プロトタイプの自動登録から弁別度 LUT の構築までに要する時間（図 3.9 上部の store）は約 10msec であった．

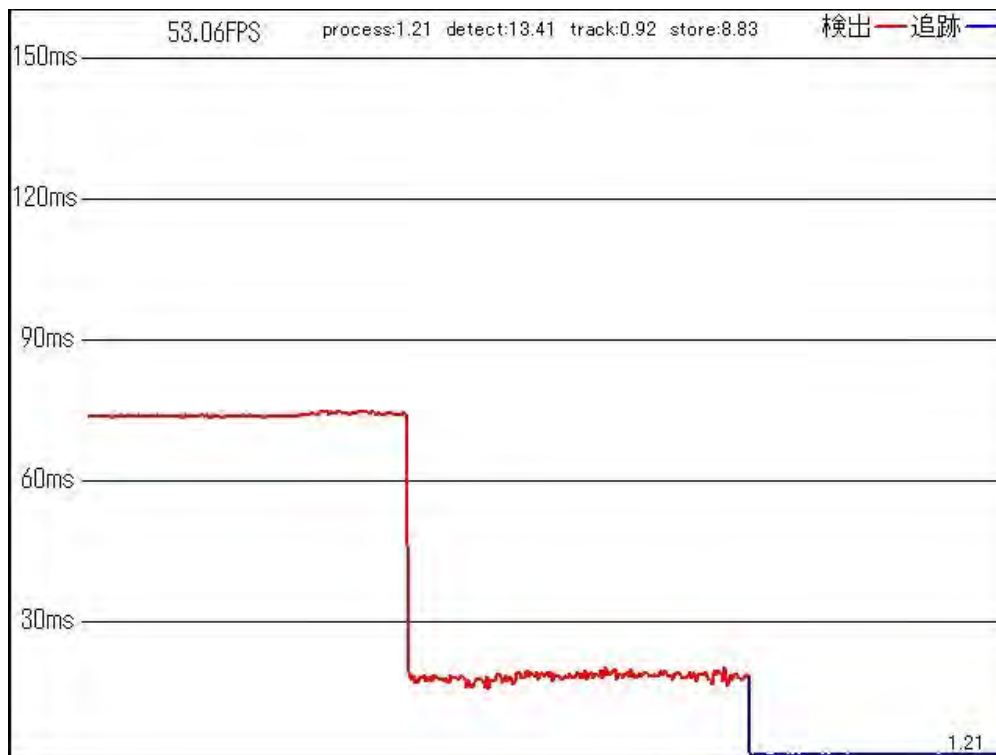


図 3.9 処理時間のグラフ

### 検出方法の比較

同じ画像系列に対して、従来手法である色ターゲット検出と提案手法である識別器を用いた検出による検出・追跡を行い、その実験結果を比較した。画像系列は、顔、手が画像中に入ってくるようなものとなっている。なお、どちらの手法も色の教示は図 3.10 の左の画像を用いて顔領域と背景領域の学習をあらかじめ手動で行っており、弁別度 LUT の更新は行わない。図の右側は、構築された弁別度 LUT を用いて得られた弁別性マップを示している。

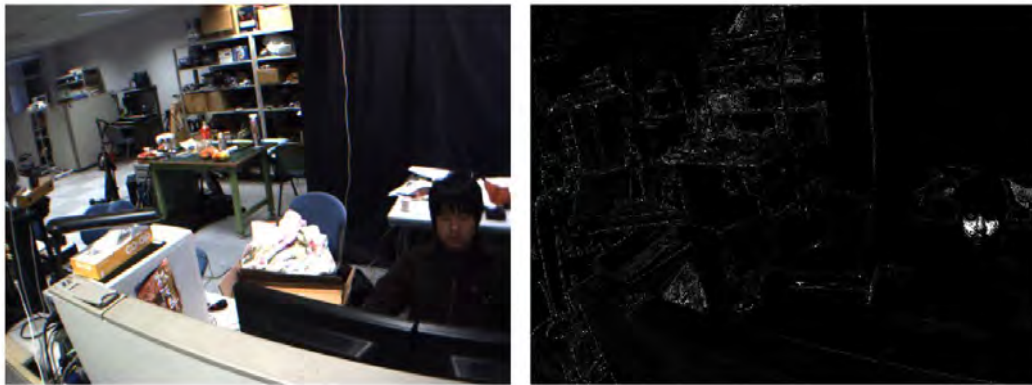


図 3.10 学習に用いた画像（左：学習画像，右：弁別性マップ）

図 3.11、図 3.12 に実験結果から部分的に抜き出してきたものを示す。図 3.11 は、画像中に顔が入ってきた時の検出・追跡結果であるが、両手法ともに顔が検出され後、追跡されていることが分かる（緑色の楕円が追跡結果）。従来手法では、肌色が画像中に入ってくるとすぐにターゲットとして検出される（① 左の画像。緑色の画素は色ターゲット検出によりターゲット画素と判断されたもの）。一方、提案手法ではステレオ処理を行っているため、画像の端の方では点対応が取れず距離計算ができない。そのため、従来手法と比べ、少し遅れて顔が検出され追跡が始まっている（③ 右の画像。赤色の矩形が顔識別器による検出結果）。図 3.12 は、手が画像中に入ってきた時の検出・追跡結果である。従来手法では、色ターゲット検出を行っているため教示色と似ている手が検出され、追跡が行われてしまっているのに対し、提案手法では、教示色と似た色の手が検出・追跡されることはない。

提案手法の検出時間は従来手法に比べると大きくなってしまっているが、約 50ms 程度であるため十分実用的である。また、一度顔が検出されれば、追跡処理は従来手法と同様

に、高速に行うことができる。さらに、応用システムを開発する上で検出精度は重要な問題であり、提案手法では顔だけを正確に検出・追跡できるようになった。

### プロトタイプ登録方法の比較

照明環境や追跡対象の異なる 4 つの画像系列を用意し、プロトタイプの教示を手動（弁別度 LUT が固定）で行った場合と各系列で検出された顔領域から自動（弁別度 LUT が系列ごとに変動）で行った場合の検出・追跡実験を行った。画像系列では、背景に肌の色と類似している段ボールがあり、一般的な色ベースの顔検出方法で誤検出が多いシーンである。系列 1～3 は照明環境が異なり、系列 4 は追跡対象が異なっている。また、手動登録では系列 1 と同様の環境で色の教示を行っており、顔検出の結果は追跡楕円の初期位置の設定にだけ用いられる。

実験結果から部分的に抜き出したものを図 3.13～3.16 に示す。各図において、最上段は顔が検出されたフレームをカラー画像で示しており、以降は追跡の様子を弁別性マップで示している。弁別性マップでは、輝度値が高いほど弁別度が大きいことを表しており、安定した追跡を行うためには、顔領域の弁別度が高く、背景領域の弁別度が低いほど理想である。また、青色の楕円が対象領域を表す追跡楕円である。

系列 1 と同様の環境で手動登録された弁別度 LUT が固定の場合では、環境の異なる系列 2～4 の時、顔領域の弁別度が小さくなっていたり背景領域の弁別度が高くなってしまっていることが分かる。一方、各画像系列において顔検出の結果から自動登録された弁別度 LUT が適応的に変化する場合では、すべての画像系列において背景領域の弁別度が低く顔領域の弁別度が高くなっていることが分かる。表 3.1 に、各系列の追跡開始後 100 フレーム間において、追跡楕円内の弁別度の平均値を比較したものを示す。表から、系列 1 では両方とも弁別度が高くなっているが、系列 2～4 では弁別度 LUT が固定されてる場合に比べ変動するほうがより高い弁別度となっていることが分かる。系列 4 では、弁別度 LUT を自動構築した場合でも他の系列に比べ弁別度が低くなっているが、顔領域と同じ色の首部分が背景領域として多く存在したためだと考えられる。このように、環境に応じて自動的に弁別度 LUT を構築することで顔領域の弁別度を高く保てるため、安定した追跡が可能となる。

表 3.2 に、各系列の追跡開始後 100 フレーム間における楕円パラメータ（長軸、短軸）の標準偏差を示す。用いた画像系列では、ターゲットに奥行きの変化がほとんどないものとなっているため、ターゲットの大きさを表す楕円の長軸と短軸が一定であるほど安定した追跡であるといえる。

弁別度 LUT が固定されている場合、系列 1, 4 では楕円パラメータの誤差が小さく安

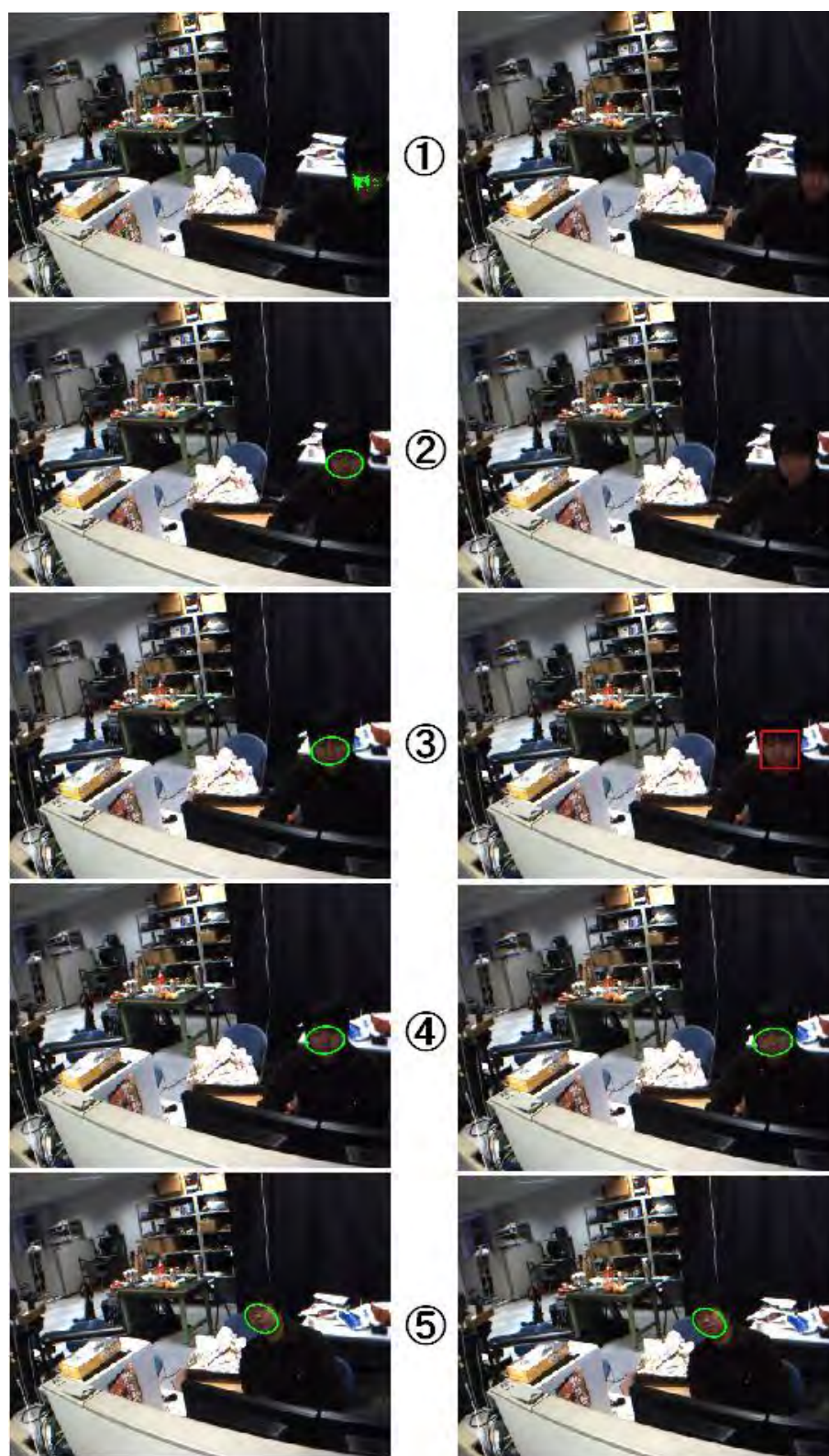


図 3.11 顔が入ってきた場合の検出・追跡結果（左：従来手法，右：提案手法）



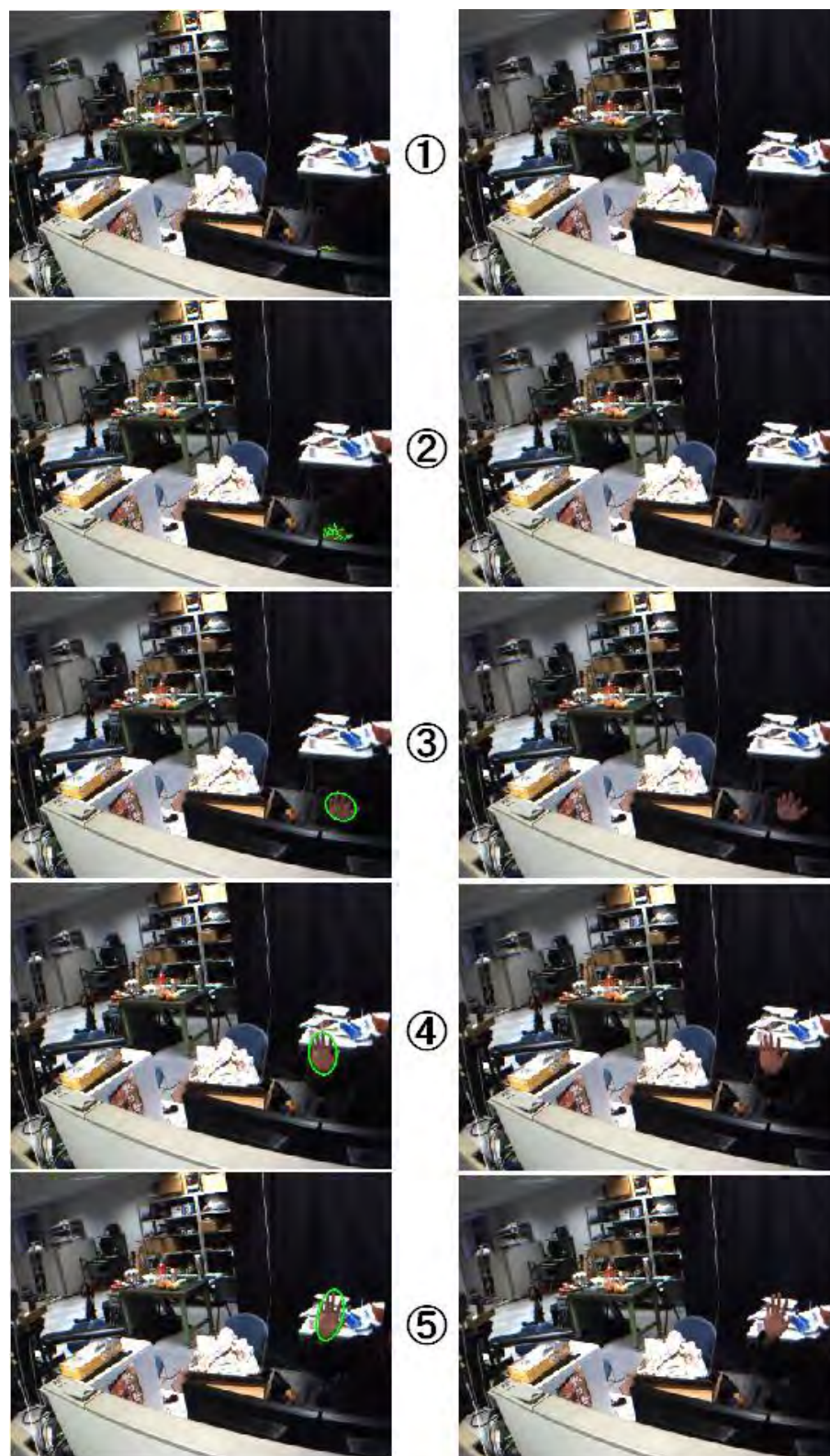


図 3.12 手が入ってきた場合の検出・追跡結果（左：従来手法，右：提案手法）

表 3.1 追跡楕円内の弁別度の平均値

	系列 1	系列 2	系列 3	系列 4
弁別度 LUT 固定	0.61	0.45	0.30	0.33
弁別度 LUT 変動	0.66	0.60	0.60	0.48

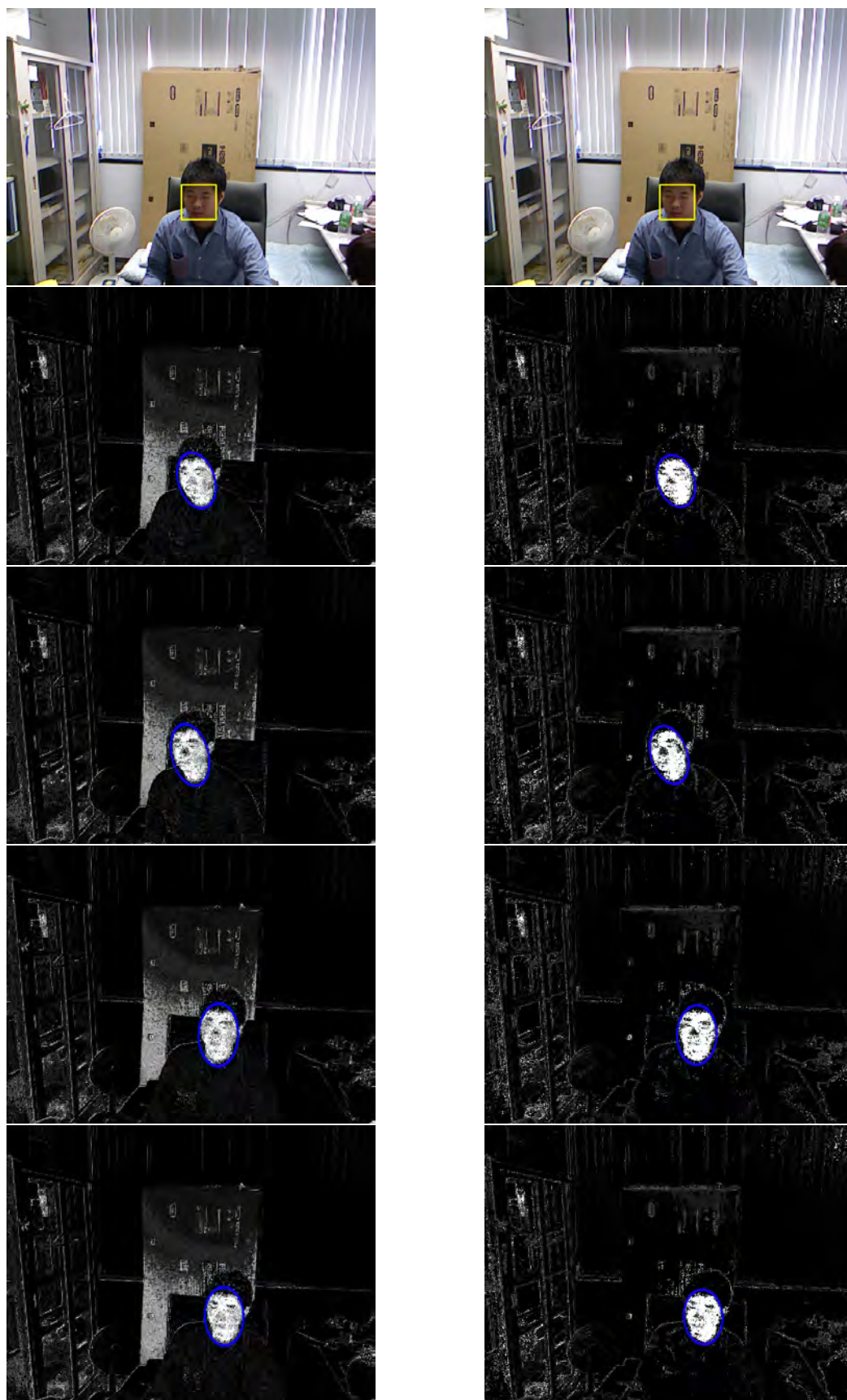
定した追跡が行えているが、系列 2, 3 では楕円パラメータの誤差が大きくなっており安定した追跡が行えていないことが分かる．系列 2 については、図 3.14(a) のように顔領域の弁別度はある程度高いものの背景の弁別度も高くなってしまっているため追跡が不安定になっている．系列 3 については、図 3.15(a) のように顔領域の弁別度が小さくなっており追跡が失敗してしまっている．系列 4 については、図 3.16(a) のように顔領域と背景の弁別度が同じような値になってしまっているが、弁別度が小さい髪領域によって分断されてるため追跡楕円が広がらずに追跡が行えていた．

一方、弁別度 LUT が変動する場合では、すべての系列において楕円パラメータの誤差が数ピクセルとなっており、安定した追跡が行えていることが分かる．

以上のように、従来手法では手動によって事前に一度だけターゲット・背景色のプロトタイプを教示していたため、照明環境や追跡対象が変わると追跡が不安定になってしまうことがあり、安定した追跡を行うためには環境ごとに再教示を行う必要があった．提案手法では、顔検出で得られた顔領域に基づいて自動的にプロトタイプを取得して弁別度 LUT の構築を行うため、検出時の環境に適応したカラーモデルによる安定した追跡を自動的に開始できるようになった．

表 3.2 LUT 固定・変動時の楕円パラメータの標準偏差（単位：画素）

		系列 1	系列 2	系列 3	系列 4
弁別度 LUT 固定	長軸	5.21	43.86	34.01	6.19
	短軸	1.07	29.43	31.87	1.97
弁別度 LUT 変動	長軸	4.42	4.12	5.82	6.22
	短軸	1.61	1.29	1.49	4.41

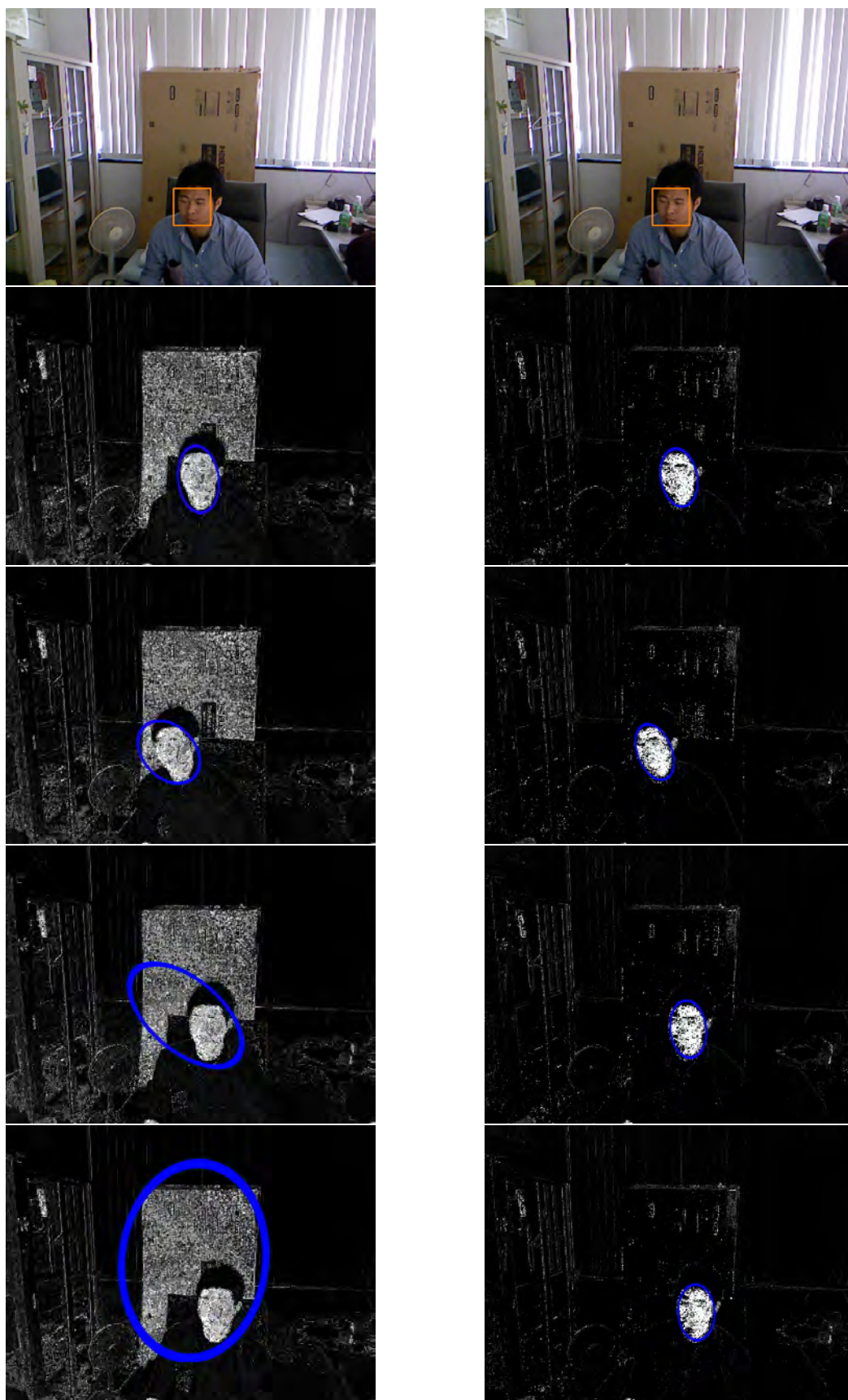


(a) 弁別度 LUT 固定

(b) 弁別度 LUT 変動

図 3.13 手動登録と自動登録の比較 (系列 1)

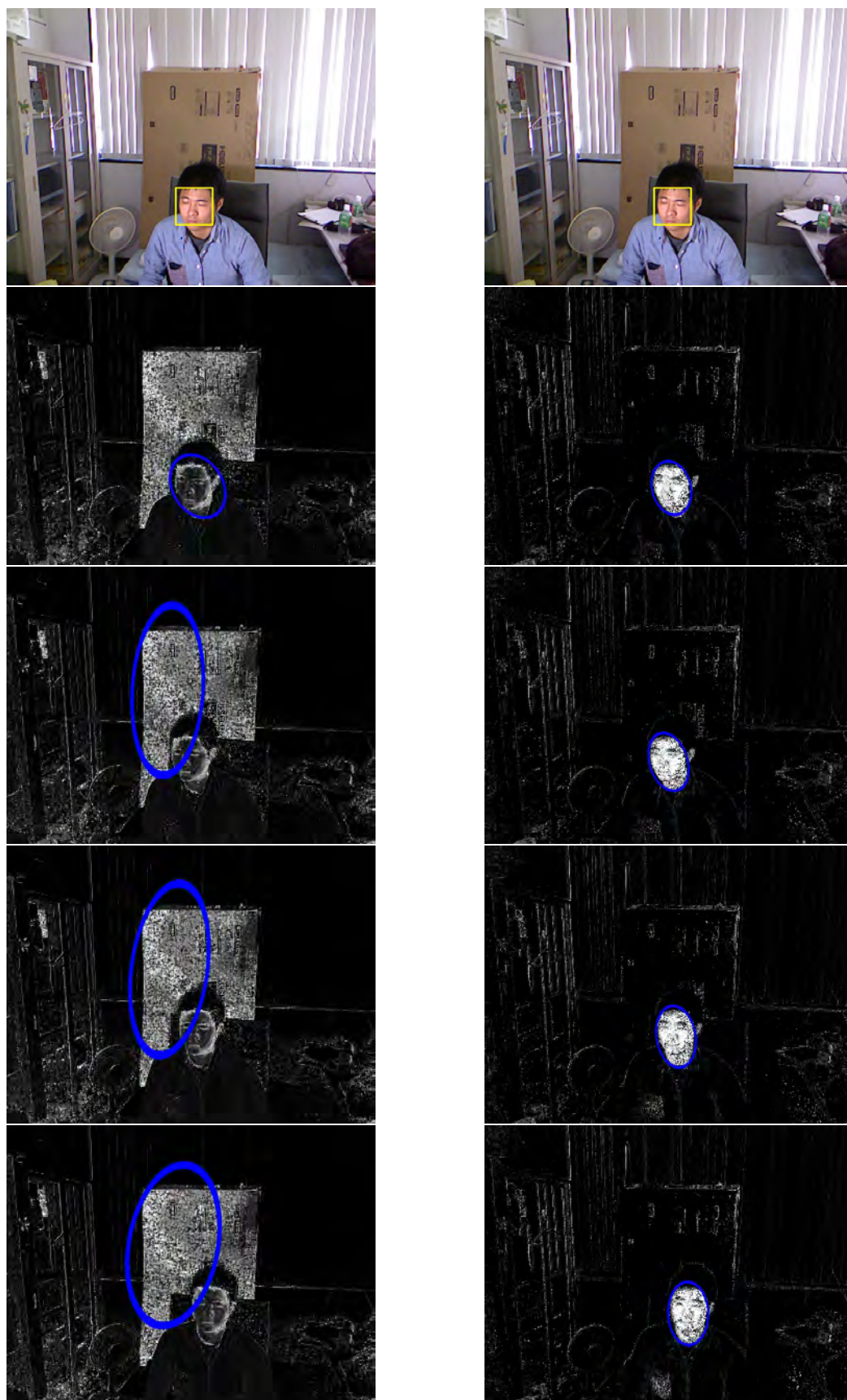




(a) 弁別度 LUT 固定

(b) 弁別度 LUT 変動

図 3.14 手動登録と自動登録の比較 (系列 2)

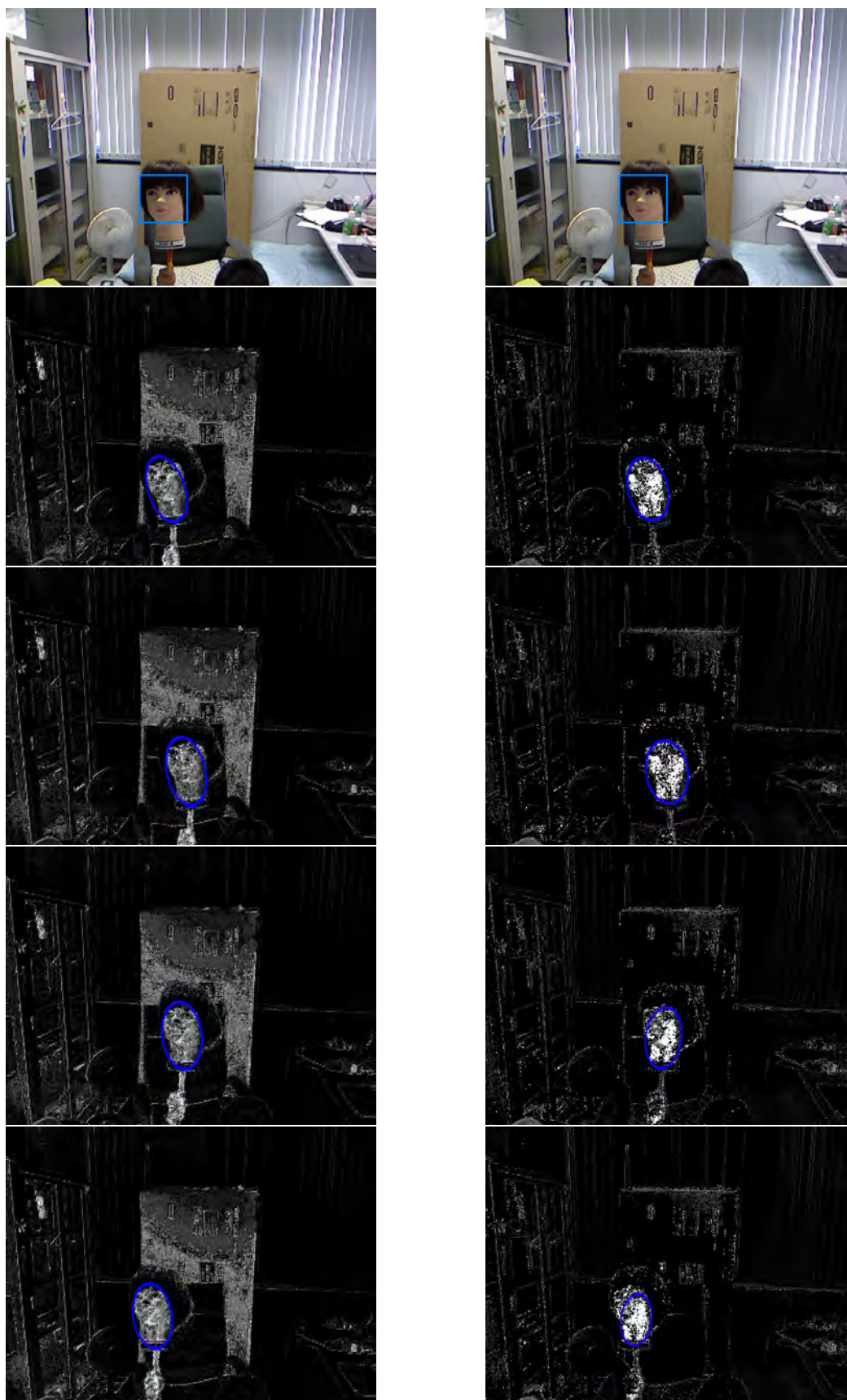


(a) 弁別度 LUT 固定

(b) 弁別度 LUT 変動

図 3.15 手動登録と自動登録の比較 (系列 3)





(a) 弁別度 LUT 固定

(b) 弁別度 LUT 変動

図 3.16 手動登録と自動登録の比較 (系列 4)

### 3.3.2 追跡中の安定化処理の有無による比較

弁別度追跡法は，対象追跡において，可変テンプレートマッチング法や Mean Shift 法に比べ有効であることが示されている [98]．本項では，奥行き情報の利用や追跡モデルの更新によって，より安定に顔追跡が行えることを示す．

#### 未知の背景が混入する場合

ここでは，追跡の安定化に関して行った比較実験について述べる．追跡対象が固定されており，背景領域が変化するような 300 フレームの画像系列を用い，追跡楕円のサイズ制約，弁別度 LUT の更新について，処理の有無による比較を行った．この画像系列は，追跡開始時には存在しない顔と似た色の背景領域が混入してくるようなものとなっている．楕円制約は，追跡中の顔までの距離から式 (2.3) により求められた画像上での顔サイズによって行われる．弁別度 LUT の更新は，追跡中に行われる顔検出が成功した際に随時更新されるが，顔検出を 30 フレーム間隔で行うようにしている．

図 3.17 は，(a) 安定化処理を行わなかった場合，(b) 楕円制約を行った場合，(c) 弁別度 LUT の更新を行った場合の追跡結果の一部をそれぞれ示している．最上段は顔が検出されたフレームをカラー画像で示しており，このフレームにおいて顔領域から弁別度 LUT が自動構築される．2 段目以降は追跡の様子を弁別性マップで示しており，安定な追跡を行うために，顔領域の明度値が高く背景領域の明度値が低いほど理想である．また，青色の楕円が追跡中の対象領域を表している．

安定化処理を行わなかった場合（図 3.17(a)）では，顔と似た色の背景が顔領域に近づき，追跡楕円と弁別度が高い背景が重なると追跡楕円が広がってしまい，安定した追跡ができなくなっている．一方，楕円制約を行った場合（図 3.17(b)）では，追跡楕円と弁別度が高い背景領域が重なっても追跡楕円が広がらず，安定した追跡が行えている．また，弁別度 LUT の更新を行った場合（図 3.17(c)）では，新たに混入してきた背景下で弁別度 LUT の再構築が行われるため，背景領域の弁別度を低く保つことができ，安定した追跡が可能となっている．図 3.18 に各フレームにおける追跡楕円の長軸の長さをグラフにしたものを示す．安定化処理を行っていない場合は楕円の大きさが不安定になっているのに対し，安定化処理を行った場合は楕円の大きさがほぼ一定となっており，追跡が安定していることがわかる．

また，追跡中の楕円パラメータ（x 座標，y 座標，長軸，短軸）について，その標準偏差を表 3.3 に示す．対象が固定されているため，楕円パラメータの標準偏差が 0 に近いほど

表 3.3 安定化処理時の追跡楕円パラメータの標準偏差（単位：画素）

	x 座標	y 座標	長軸	短軸
安定化処理なし	93.61	20.10	49.38	28.98
楕円制約	2.13	1.13	3.70	0.49
弁別度 LUT 更新	1.38	1.24	1.34	0.60

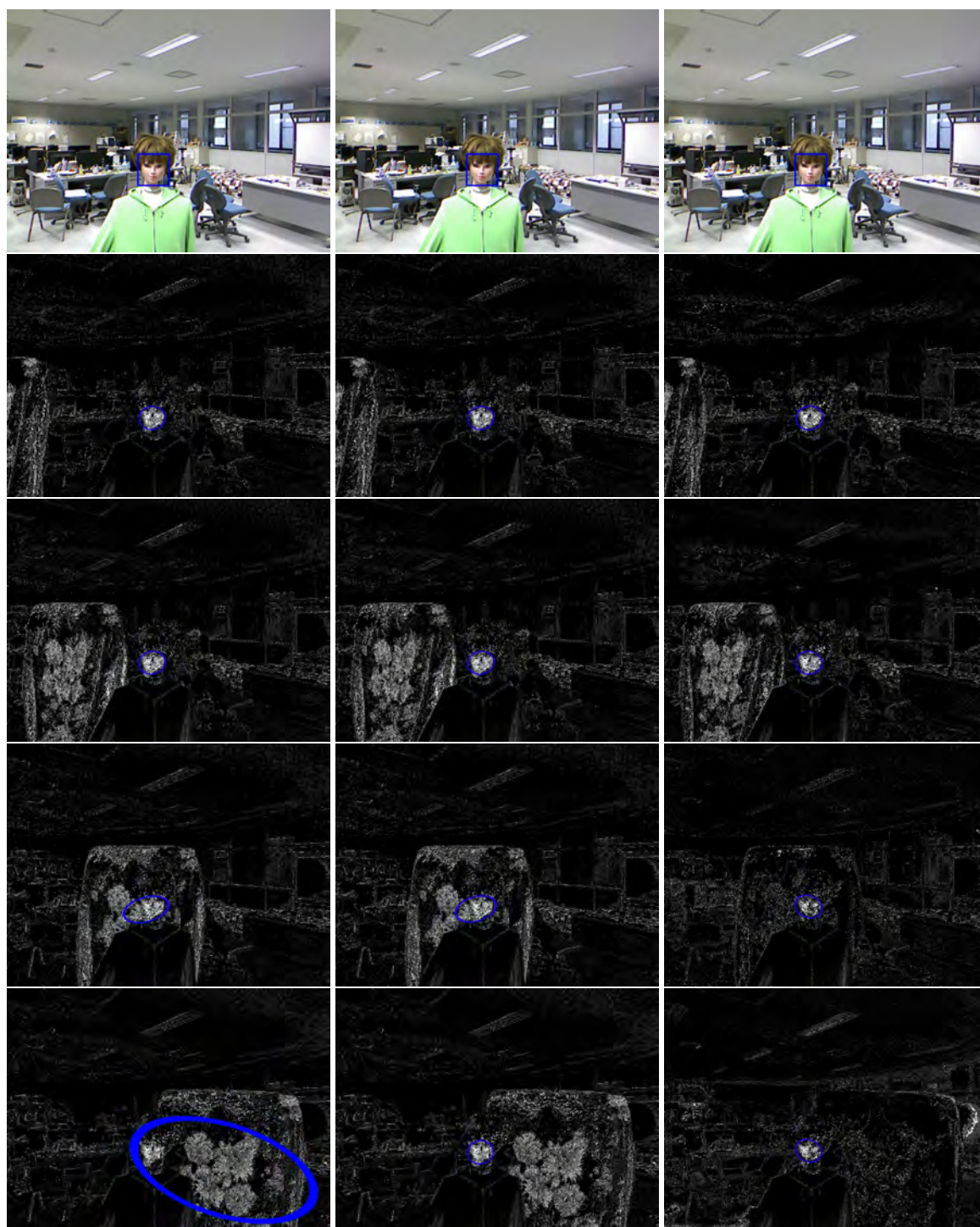
追跡が安定していることを意味するが、安定化処理を行った場合では、数ピクセルの誤差で安定した追跡が行えていることが分かる。このように、未知の背景の混入により背景領域の弁別度が高くなるような場合でも、楕円制約により追跡楕円の拡大を抑えることができ、未知の背景を非ターゲットとして登録した上で弁別度 LUT の再構築を行うことで、より安定した追跡を行えることが確認できた。

#### 追跡中に照明変動が起こる場合

肌の色と類似している段ボールがある背景下で、顔追跡中に照明変動が起こるような画像系列に対し、弁別度 LUT の更新を行った場合と行わなかった場合の追跡結果を図 3.19、図 3.20 に示す。両図とも、左の列がカラー画像、右の列が弁別性マップでの処理結果を表しており、③の画像で照明変動が起こっている。また、画像内の矩形が顔検出結果、楕円が追跡時の対象領域を表している。図 3.20 で背景領域の弁別度が高くなっているのは、更新処理時間を抑えるためにプロトタイプ自動登録時の背景領域を追跡対象の周辺だけにしているためである。

弁別度 LUT の更新を行わなかった場合、照明変動が起こったフレーム（図 3.19③）で急激に顔領域の弁別度が低くなり追跡に失敗している。また、追跡失敗により顔検出モードへ戻った後、再び顔が検出され（図 3.19④）、追跡が再開されている（図 3.19⑤）。一方、LUT の更新を行っている場合は、顔が検出される度に随時 LUT が再構築されるため、追跡中に照明変動があった場合にも追跡が途切れることなく対象を追跡し続けることができる。

このように、追跡中に照明変動が起こるような場合にも、プロトタイプの追加登録により弁別度 LUT の再構築を行うことで追跡が失敗することを少なくできる。しかし問題点として、LUT の再構築直後はキャッシュの関係で LUT の参照速度が遅くなってしまうため、LUT の更新が頻繁に起こるような場合、LUT の更新自体に要する時間も含め追跡速度が低下してしまうことと、顔検出は正面顔のみであるため、横を向いていたりなど照明変動中に顔が検出されない場合は LUT が更新されず、追跡に失敗してしまうことがあ



(a) 安定化処理なし

(b) 楕円制約

(c) 弁別度 LUT 更新

図 3.17 安定化処理による追跡結果の比較

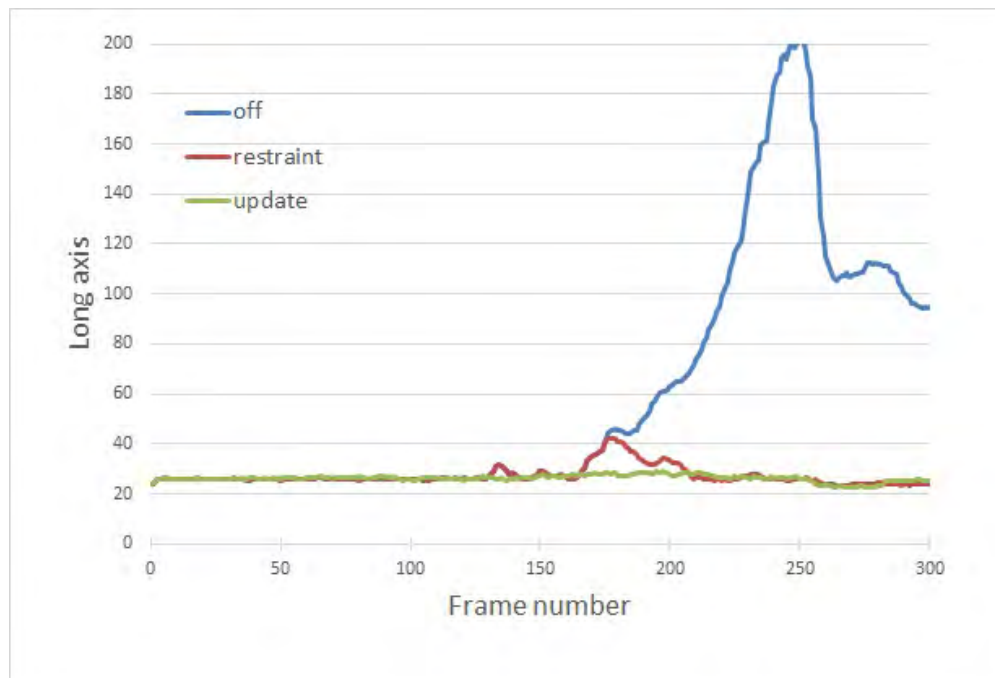


図 3.18 追跡楕円の長軸の変化

る。前者については、マルチスレッドにより追跡処理と更新処理を並列に行うことで処理時間を高速化することが可能であると考えられる。しかし後者に関しては、弁別度 LUT の再構築を行うタイミングが問題であり、今後検討していく必要がある。

### 3.3.3 他の追跡手法との比較実験

前述した追跡の安定化処理を両方行った提案手法と CAM Shift 法 [71, 93] による顔の追跡を行った。CAM Shift 法は、色ヒストグラムを用いて追跡を行う Mean Shift 法を拡張し、対象の大きさや姿勢の変化に対応した追跡手法であり、対象領域を楕円として得ることができる。実験では、顔検出で得られた矩形領域内で色ヒストグラムを構築することで、提案手法と同様に自動的に追跡が開始できるようにした。また、追跡対象を固定し、背景が変化する 300 フレームの画像系列（前項 3.3.2 の「未知の背景が混入する場合」と同じ）を用い、対象領域を表す楕円のパラメータで追跡性能の比較を行う。

図 3.21 に追跡結果の一部を示す。CAM Shift の結果画像では、得られた対象を表す楕円と中心位置を示している。図 3.21(a) の CAM Shift では、1 段目に示している顔検出の結果から対象の情報だけの色ヒストグラムを構築しているため、髪の色が混入して頭部全体が対象領域として追跡されている。また、対象と似た色の背景が重なると追跡楕円が



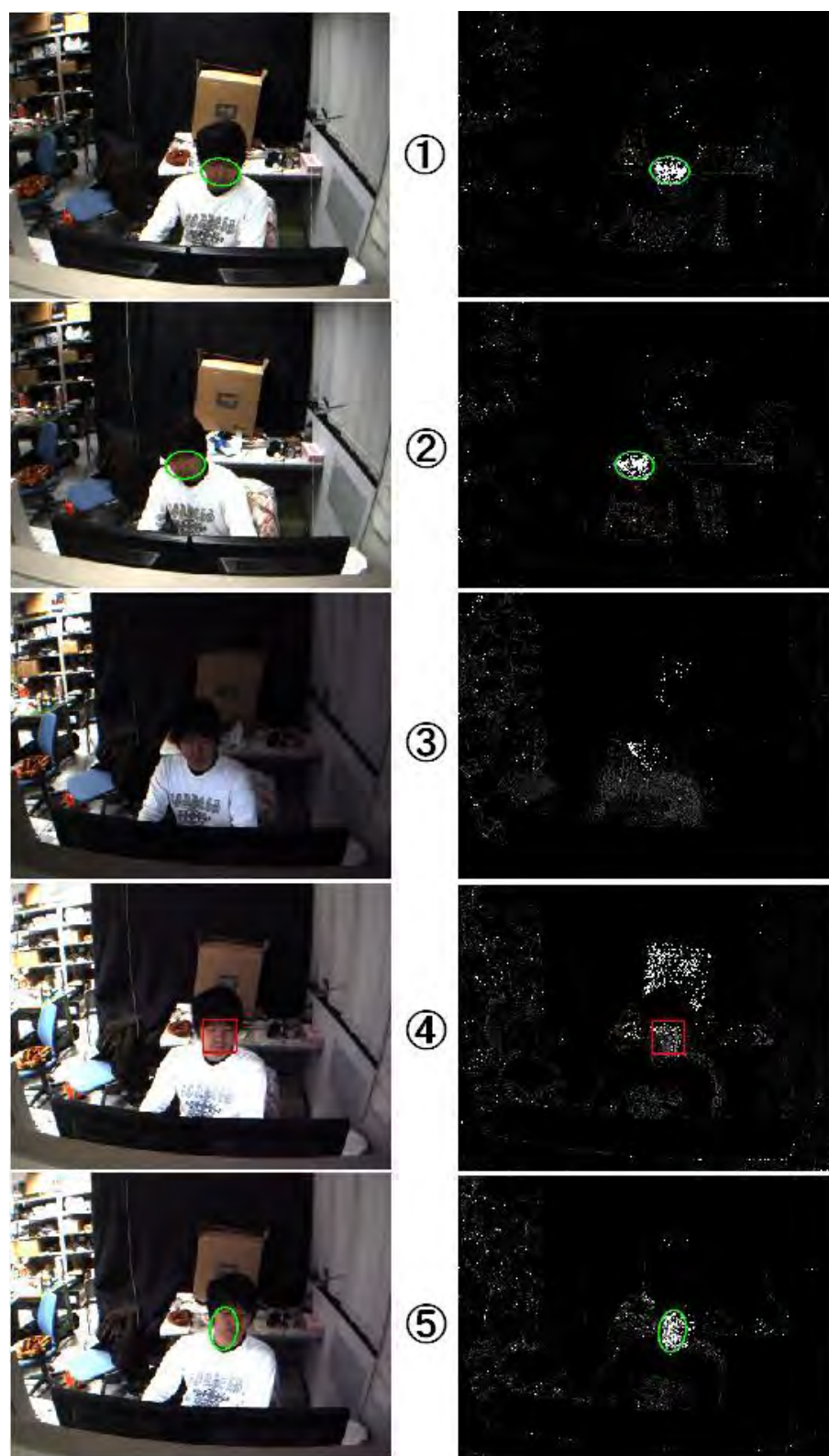


図 3.19 LUT の更新をしない場合の追跡結果（左：結果画像，右：弁別性マップ）



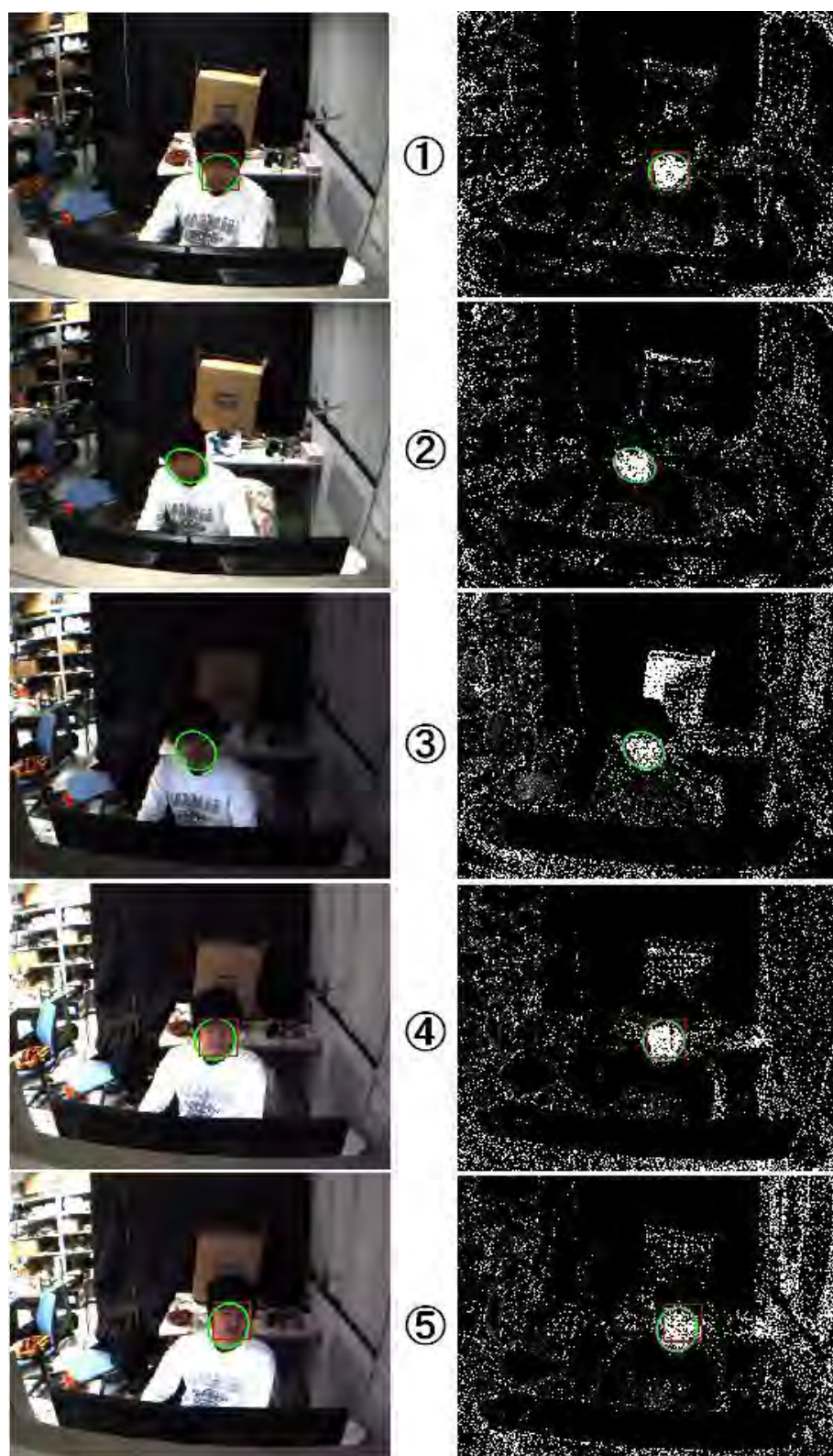


図 3.20 LUT の更新を行った場合の追跡結果（左：結果画像，右：弁別性マップ）

表 3.4 CAM Shift と提案手法の追跡楕円パラメータの標準偏差（単位：画素）

	x 座標	y 座標	長軸	短軸
CAM Shift	96.53	27.30	58.36	89.56
提案手法	1.78	1.38	1.36	0.35

広がってしまい、対象領域を正しく追跡できていないことがわかる。一方、図 3.21(b) の提案手法では、追跡楕円に変化がなく、安定した追跡を行えていることがわかる。追跡開始から 300 フレーム目までの楕円パラメータ（x 座標，y 座標，長軸，短軸）の標準偏差を表 3.4 に示す。CAM Shift では追跡楕円の位置や大きさが不安定となり誤差が大きくなっているのに対し、提案手法ではほとんど誤差がなく、追跡楕円のパラメータが一定で終始安定した追跡が行えていたことがわかる。



図 3.21 CAM Shift 法と提案手法の比較

### 3.4 まとめ

本章では，ステレオ視による顔検出と色弁別度追跡法を統合し，追跡のための色教示を顔検出時に自動で行うことにより，正確かつ高速な顔検出・追跡システムを構築する方法を提案した．

奥行き情報を利用して高速化された識別器による顔検出処理を行い，追跡の初期化処理として必要なターゲットプロトタイプと非ターゲットプロトタイプの自動取得によって，検出時の照明環境や追跡対象に合わせた適応的なカラーモデルを構築することができ，安定な追跡が行えることを示した．

実験では，顔のみが正確に検出されること，ターゲット・背景色の自動学習により照明変動や追跡対象の変化に対して安定に追跡が行えることを確認し，従来手法との比較により提案手法の有効性を示した．また，追跡楕円の制約や弁別度 LUT の更新を行うことで，より安定した顔追跡が行えることを確認した．さらに，様々な環境下で行ったデモ（研究業績 6, 11）において，人種の異なる追跡対象や照明環境に適応して安定な顔検出・追跡が行えており，提案手法の有効性を確認している．

課題としては，追跡中の弁別度 LUT の更新頻度があり，頻繁に更新を行うと追跡処理の速度が低下してしまう．そのため，LUT の再構築処理のマルチスレッド化などによる高速化や，更新タイミング，更新方法についての検討が必要であると考えている．

## 第4章

# 顔検出と 6D K-means Tracker を組み合わせた髪追跡システム

本章では、追跡手法として奥行き情報を用いた 6D K-means Tracker を提案し、第2章で提案した顔検出法と組み合わせた髪追跡システムの構築方法について述べる。このシステムでは、背景に応じた非ターゲットクラスタ中心の配置法 [113] に使っている特徴ベクトルにも奥行き情報を導入することによって、次元拡張に伴う処理コストの増加を抑制しながら追跡の安定化を行う。

髪の色やスタイルには重要な個人特徴が含まれているため、画像から髪領域を抽出できれば、顔画像処理の特徴として利用することができる。ビデオレートで髪領域を把握することで、髪型を利用した識別や認証などの応用事例において、前処理として利用したり、処理の簡素化、高速化が期待できる。また、顔領域だけの追跡を行った場合、顔領域が見えなくなるような頭部の姿勢変化があると追跡を継続することができなくなるが、髪領域を追跡することで、頭部の姿勢変化に関わらず頑健な追跡が可能となる。

髪領域は、スタイルや姿勢変化による見えの変化も大きく、形状モデルを利用しにくい。そのため画素単位の検出・追跡手法が望ましい。また、髪領域のテクスチャ情報は乏しく、東洋人では髪の色が基本的に黒・灰・白であり色彩情報も乏しく、背景に同じ色が存在する可能性も高いため、色情報だけでなく距離情報を用いることで安定に髪領域と背景を分離できると考えられる。

前章で述べた色弁別度追跡法は、色情報のみでターゲットモデルや背景モデルを構築し、LUT を用いた高速化を行っているため、常に変化する奥行き情報をモデルに導入することは難しい。一方 K-means Tracker は、フレームごとにクラスタ中心（モデルを表現する特徴ベクトル）の更新を行っているため、奥行き情報の導入が容易であり、モデル



を次元拡張することで追跡の安定化が期待できる。

本章では、まず、K-means Tracker と FB-Clustering Tracker (K-means Tracker を改良したもの) についての概要を説明し、奥行き情報の導入により髪追跡を安定化させる方法を説明する。また、第2章で提案した高速顔検出法と組み合わせ、環境に応じた追跡モデルを構築しながら自動的に髪・顔・頭部(髪 + 顔)をそれぞれ追跡可能なシステムを構築する。

## 4.1 K-means tracker

K-means tracker[104, 105, 106, 107, 114, 108, 109, 113, 110, 112, 111] は、我々の研究グループが提案している対象追跡手法である。この追跡手法は、本章で述べる髪や顔領域の追跡システムにおいて、ベースとして用いられる。また、この K-means tracker の拡張について述べる。したがって、まず、従来の K-means tracker について説明する。

### 4.1.1 K-means tracker の概要

K-means tracker は、画像上の追跡対象と背景の両方に対して複数のクラスタ中心を割り当て、3次元の色空間と2次元の座標空間を合わせた5次元特徴空間における K-means クラスタリング [115] によって、サーチエリア内の各画素をターゲットと非ターゲットに分けることにより追跡を行う手法である。追跡対象に割り当てられたクラスタ中心をターゲットクラスタ中心、背景に割り当てられたクラスタ中心を非ターゲットクラスタ中心と呼ぶ。特徴空間内で、これらクラスタ中心とサーチエリア内の各画素との距離を計算し、その距離に基づいてターゲットか非ターゲットのラベルが入力画素に付けられる。このように、K-means tracker はピクセルワイズな追跡手法となっているため、追跡対象の領域に混入した背景画素を排除しながら非剛体の対象を追跡できる頑健性を持っている。

K-means tracker では、ターゲットクラスタ中心は画像中の追跡対象上に、非ターゲットクラスタ中心は、画像上で追跡対象を取り囲むよう背景上に複数配置される。追跡対象が動けば、K-means クラスタリングの結果、ターゲットクラスタ中心の重心位置が更新される。このとき、対象を取り囲むように非ターゲットクラスタ中心が再配置される。これを毎フレーム繰り返すことで、動画像上での対象追跡を行っている。

和田ら [104] によって提案された K-means tracker では、初期フレームにおいて配置された非ターゲットクラスタ中心同士の幾何学的関係を変化させずに再配置を行っていたため、回転や大きさの変化など追跡対象の形状変化に追従することが難しかった。この問題を解決するために、華ら [105, 106, 107, 114] は長軸、短軸、傾きの自由度を持った可変

楕円をターゲットラベルの付いた画素群に当てはめる方法を提案した．これにより，入力画像中の追跡対象の大きさや形状の変化にも追従できるようになった．華らの手法をベースとし，大池ら [108, 109, 113] によって適応的な非ターゲットクラスタ中心の配置法が提案され，これまでサーチエリアを表す楕円上に等間隔で機械的に配置されていた非ターゲットクラスタ中心を背景の状況を考慮した適切な数，位置に配置されるようにした．これによって，背景情報のサンプリングが効果的に行われ，冗長な距離計算を抑制し効率的な処理となることで，追跡処理の安定化と高速化が実現した．一方，戚ら [110, 112] は特徴空間に 1 次元の奥行き情報を追加し，2 次元座標空間における距離をマハラノビス汎距離で評価する手法（FB-Clustering Tracker と名付けられている）を提案し，追跡の安定化を行った．

このように，K-means tracker はこれまで多くの改良がなされ，さまざまな条件下における対象追跡を他の追跡手法に比べて頑健に行えることが示されている．本研究では，大池らが提案した K-means tracker をベースとし，戚らのように奥行き情報を追加した 6D K-means Tracker を提案する．本節にて，大池らの K-means tracker について説明し，次節で FB-Clustering Tracker について概説する．

#### 4.1.2 K-means tracker の構成要素

図 4.1 は，K-means tracker における主な構成要素を示している．この手法では，初期フレームにて追跡対象上と背景に対して複数のクラスタ中心を割り当て，サーチエリア楕円内の各画素を K-means クラスタリングによってターゲットと非ターゲットにラベリングしながら追跡を行う．以下，各要素について説明する．

#### 5 次元特徴空間

追跡対象を構成する画素の色の類似性と空間的近接性を同時に表現するために，各画素の特徴を 3 次元色空間  $\mathbf{c} = [R, G, B]^T$  と 2 次元座標空間  $\mathbf{p} = [x, y]^T$  からなる 5 次元ベクトル  $\mathbf{f} = [\mathbf{c}, \mathbf{p}]^T$  として扱う．2 つの画素  $a, b$  の特徴ベクトル  $\mathbf{f}_a, \mathbf{f}_b$  間の距離は，色空間内と座標空間内のユークリッド距離に重み  $\alpha$  を用いて次式のように定義される．

$$d(\mathbf{f}_a, \mathbf{f}_b) = \|\mathbf{c}_{\mathbf{f}_a} - \mathbf{c}_{\mathbf{f}_b}\|^2 + \alpha \|\mathbf{p}_{\mathbf{f}_a} - \mathbf{p}_{\mathbf{f}_b}\|^2 \quad (4.1)$$

ただし， $0 < \alpha < 0.5$  である．

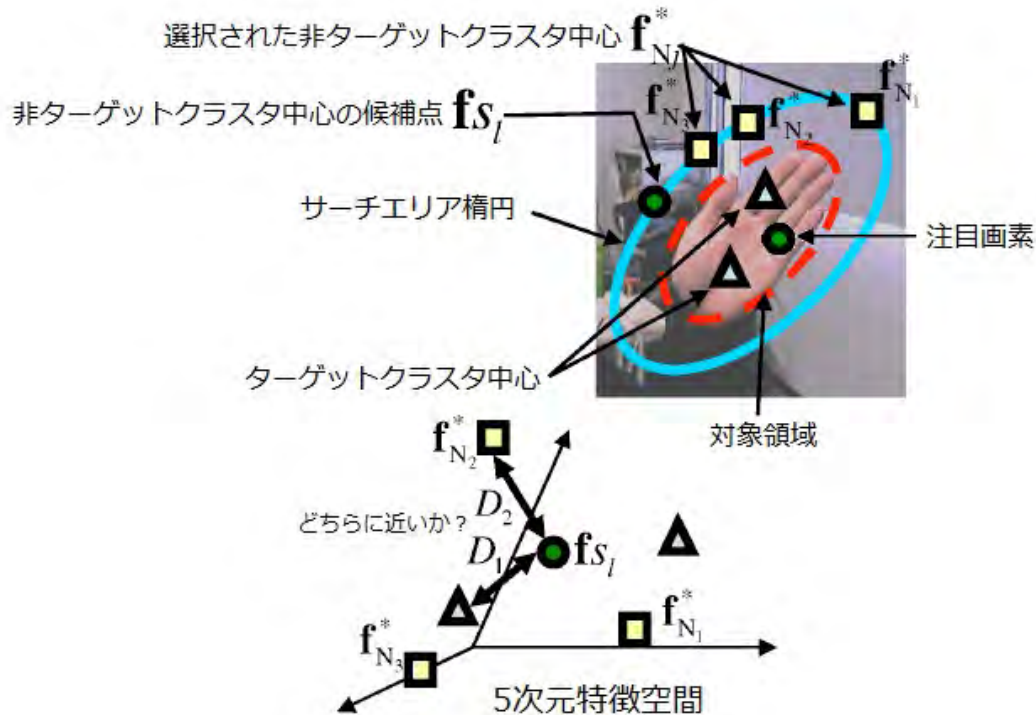


図 4.1 K-means tracker の構成要素 [113]

### ターゲットと非ターゲット

従来の追跡アルゴリズムでは、ターゲットのみモデル化し、ある領域をターゲットに属するか否かを判断するためには閾値が必要であったため、様々な環境で安定に動作させるための閾値の設定が困難であった。一方、K-means tracker では、図 4.1 の□と△で示すように、ターゲットと非ターゲットの両方を使用して追跡している。5次元特徴空間内で、入力画素がターゲットクラスタ中心、非ターゲットクラスタ中心のどちらに近いかに判断することによって属するクラスタを決定するため、閾値を設ける必要がない。

### 可変楕円モデル

図 4.1 中の赤破線で示すような可変楕円モデルが華らによって導入され、ターゲットの回転や大きさなど形状の変化に対して安定な追跡を行うことができる。可変楕円モデルとは、長軸、短軸、傾き、重心などの自由度を持った楕円パラメータを変化させることができる、対象領域を表現するモデルである。この楕円は、画像上での対象領域を取り囲むように配置され、ターゲットとしてクラスタリングされた画素集合の大きさや形状に合わせ



るように可変楕円のパラメータが毎フレーム更新される。

現フレームにおける可変楕円内のターゲット点の集合を  $S$ ，可変楕円内のターゲット点の数を  $N$ ，ターゲット点を  $\mathbf{y} = [y_x, y_y]^T$ ，とすると，次フレームにおける可変楕円モデルの中心座標  $\mathbf{c}$  は以下の式で求められる。

$$\mathbf{c} = \frac{1}{N} \sum_{\mathbf{y} \in S} \mathbf{y} \quad (4.2)$$

この値は，ターゲットクラスタ中心の重心と等しくなる。また，検出されたターゲット点の分布を正規分布であると仮定することで，ターゲット点の分布のマハラノビス距離より可変楕円を以下の式で求めることができる。

$$[\mathbf{y} - \mathbf{c}]^T \Sigma^{-1} [\mathbf{y} - \mathbf{c}] = J \quad (4.3)$$

ここで，

$$J = -2 \ln(1 - P) \quad (4.4)$$

$$\Sigma = \frac{1}{N} \sum_{\mathbf{y} \in S} [\mathbf{y} - \mathbf{c}][\mathbf{y} - \mathbf{c}]^T \quad (4.5)$$

ただし， $P = 0.95$  である。これは，ターゲットに属する画素がターゲットの重心（楕円の中心）から標準偏差の 2 倍の範囲内に存在する確率である。この等確率線が，対象領域を表す可変楕円となる。また，正規分布当てはめを行うことにより，楕円の外側付近のターゲットとしてクラスタリングされた画素を，アウトライアと見なすことができる。これにより，誤クラスタリングの影響を小さくすることができる。

サーチエリア楕円は，求められた対象領域を表す楕円の長軸，短軸をそれぞれ  $\gamma$  倍したものとする（実験的に  $\gamma = 1.3 \sim 1.5$  とされており，追跡対象の見かけの運動速度が速いほど大きい値を選ぶ必要がある）。非ターゲットクラスタ中心は，このサーチエリア楕円上に配置される。

#### 適応的な非ターゲットクラスタ中心の配置

華らの K-means Tracker では，非ターゲットクラスタ中心はサーチエリア楕円上に等間隔で配置されていた。そのため，非ターゲットクラスタ中心同士の間背景にしたい色が存在しても，そこに非ターゲットクラスタ中心が配置されず，追跡が不安定になる場合があった。これを避けるために非ターゲットクラスタ中心の数を増やすと処理コストが高くなってしまう。また，似た特徴を持つ非ターゲットクラスタ中心が複数あった場合，こ

れらとの距離計算が、追跡にほとんど寄与しない無駄な処理となってしまうことがある。このような問題を解決するため、背景の状況に応じた適応的な非ターゲットクラスタ中心の配置法が大池らに提案され [113]、追跡の安定化と高速化が実現している。

この手法では、サーチエリア楕円の輪郭上の候補点において  $D_1$  が  $D_2$  より短い場合、その候補点を非ターゲットクラスタ中心と決定する。輪郭上を走査してターゲットにクラスタリングされそうな領域の 1 点を非ターゲットクラスタ中心とすることで、各フレーム毎の背景に応じた非ターゲットクラスタ中心の個数や位置が決定される。以下、決定手順を説明する。

非ターゲットクラスタ中心候補点  $\mathbf{f}_s$  の個数を  $q$ 、 $l$  番目の候補点を  $\mathbf{f}_{s_l}$  ( $l = 1 \cdots q$ ) とする。まず、サーチエリア楕円の輪郭上に非ターゲットクラスタ中心の 1 点目を定める必要があるため、楕円の長軸と、その楕円の輪郭との交点の一つを、一つ目の非ターゲットクラスタ中心とする。

$$\mathbf{f}_{N_1}^* = \mathbf{f}_{s_1} \quad (4.6)$$

ここで、 $\mathbf{f}_{N_j}^*$  は  $j$  番目の非ターゲットクラスタ中心を表す特徴ベクトルである。これは、全ての  $\mathbf{f}_{s_l}$  に対して以下に述べる処理を行うことで順次決定されていく。

ある  $\mathbf{f}_{s_l}$  から最も近いターゲットクラスタ中心までの距離と、既に決まっている非ターゲットクラスタ中心までの距離を求める。これらの距離は、5 次元特徴空間内における距離計算と同じ式で計算される。

$$D_T(\mathbf{f}_{s_l}) = \min_{i=1 \sim n} \{d(\mathbf{f}_{s_l}, \mathbf{f}_{T_i}^*)\} \quad (4.7)$$

$$D_N(\mathbf{f}_{s_l}) = \min_{j=1 \sim m'} \{d(\mathbf{f}_{s_l}, \mathbf{f}_{N_j}^*)\} \quad (4.8)$$

ここで、 $\mathbf{f}_{T_i}^*$  は  $i$  番目のターゲットクラスタ中心を表す特徴ベクトル、 $n$  はターゲットクラスタ中心の数、 $m'$  はこれまでに決定している非ターゲットクラスタ中心の数、 $d(\mathbf{a}, \mathbf{b})$  は 5 次元空間内における  $\mathbf{a}$ ,  $\mathbf{b}$  間の距離である。 $D_T(\mathbf{f}_{s_l}) < D_N(\mathbf{f}_{s_l})$  なら、画素  $\mathbf{f}_{s_l}$  は非ターゲットクラスタ中心と判定する。

$$\mathbf{f}_{N_{(m'+1)}}^* = \mathbf{f}_{s_l} \quad (if \ D_T(\mathbf{f}_{s_l}) < D_N(\mathbf{f}_{s_l}) ) \quad (4.9)$$

この計算を全ての  $\mathbf{f}_s$  に対して行う。これにより、背景状況に応じた追跡に適切な非ターゲットクラスタ中心の配置と個数を、毎フレーム、適応的に決定することができる。

#### 4.1.3 K-means tracker の処理手順

K-means tracker の追跡処理は、図 4.2 に示すように「追跡対象の初期化」、「クラスタリング」、「楕円フィッティング」、「非ターゲットクラスタ中心の配置」の 4 つの処理から

なる．それぞれの処理段階に分けて，K-means tracker における処理手順を説明する．



図 4.2 K-means tracker の処理手順

#### Step 1 : 追跡対象の初期化

初期フレーム ( $t = 0$ ) において，追跡対象の色構成に合わせた数 ( $n$ ) のターゲットクラスタ中心点を手動で指定し，背景上の一点  $\mathbf{b}$  を指定する．指定されたターゲットクラスタ中心の重心  $\mathbf{c}$  を中心とし， $\|\mathbf{c} - \mathbf{b}\|$  を半径とした円を初期フレームにおけるサーチエリア楕円とする．非ターゲットクラスタ中心は，サーチエリア楕円上に適応的に配置され，ターゲットクラスタ中心点と非ターゲットクラスタ中心点の画素における位置  $(x, y)$  と色  $(R, G, B)$  情報から各クラスタの初期特徴ベクトルが構成される．

#### Step 2 : クラスタリング

$t$  フレームにおけるサーチエリア楕円内のある画素  $\mathbf{f}_u$  に対して，5次元特徴空間内にてターゲットクラス中心  $\mathbf{f}_{T_i}^*$ ; ( $i = 1 \sim n$ ) との最短距離，

$$D_T(\mathbf{f}_u) = \min_{i=1 \sim n} \{d(\mathbf{f}_u, \mathbf{f}_{T_i}^*)\} \quad (4.10)$$

と非ターゲットクラスタ中心  $\mathbf{f}_{N_j}^*$ ; ( $j = 1 \sim m$ ) との最短距離

$$D_N(\mathbf{f}_u) = \min_{j=1 \sim m} \{d(\mathbf{f}_u, \mathbf{f}_{N_j}^*)\} \quad (4.11)$$

をそれぞれ計算する． $D_T(\mathbf{f}_u) < D_N(\mathbf{f}_u)$  であれば，その画素はターゲット点とし，そうでなければ非ターゲット点とすることでクラスタリングを行う．サーチエリア楕円内の全

ての画素に対してこの処理を行い、ターゲットとしてクラスタリングされた画素を用いてターゲットクラスタ中心を更新する．ターゲットクラスタ  $i$  にクラスタリングされた画素を  $\mathbf{f}_{t_i}$ ，その画素の集合を  $\mathbf{F}_{T_i}$ ， $\mathbf{f}_{t_i}$  の総数を  $N_i$  とし，以下の式で  $\mathbf{f}_{T_i}^* = [\mathbf{c}_{T_i}^*, \mathbf{p}_{T_i}^*]^T$  を更新する．

$$\mathbf{p}_{T_i}^* = \frac{1}{N_i} \sum_{\mathbf{f}_{T_i} \in \mathbf{F}_{T_i}} \mathbf{p}_{T_i} \quad (4.12)$$

$$\mathbf{c}_{T_i}^* = \beta \frac{1}{N_i} \sum_{\mathbf{f}_{T_i} \in \mathbf{F}_{T_i}} \mathbf{c}_{T_i} + (1 - \beta) \mathbf{c}_{T_i}^{*t-1} \quad (4.13)$$

ここで， $\mathbf{c}_{T_i}^{*t-1}$  は 1 フレーム前のターゲットクラスタ中心の色， $\beta$  は 1 次遅れ系でターゲットクラスタ中心の色を更新するための係数である ( $0 \leq \beta \leq 1$ )．

#### Step 3 : 楕円フィッティング

ターゲットとしてクラスタリングされた画素集合の分布に合わせて式 (4.2) と (4.3) を用いて対象領域を表す可変楕円のパラメータを決定する．また，この可変楕円を定数倍したものを次フレームにおけるサーチエリア楕円とする．

#### Step 4 : 非ターゲットクラスタ中心の更新

更新されたサーチエリア楕円の輪郭上の画素から適応的に新しい非ターゲットクラスタ中心が選ばれる．更新後の非ターゲットクラスタ中心点の画素における位置  $(x, y)$  と色  $(R, G, B)$  より各非ターゲットクラスタ中心の特徴ベクトルを更新し， $t = t + 1$  として **Step 2** に進む．

### 4.1.4 まとめと 5 次元の K-means Tracker における問題点

本節では，K-means tracker の基本原理について説明した．K-means tracker は改良が重ねられ，可変テンプレートマッチング法や Mean Shift 法など他の追跡手法よりも安定な追跡を行えることが示されている [114, 111]．また，画素単位のクラスタリングによる追跡は，任意の非剛体物体を追跡することが可能である．しかし，5 次元特徴ベクトルの K-means tracker には，以下に挙げるような問題がある．

### 類似背景色問題

適応的な非ターゲットクラスタ中心の配置法により，背景状況への頑健性は向上したものの，サーチエリア楕円内にターゲットと類似色の背景が混入した場合，誤ってターゲット画素としてクラスタリングされてしまう（図 4.3）．従来の K-means tracker における 5 次元特徴空間では，このような類似色背景を正確にクラスタリングすることはできず，対象領域の拡大によって追跡が不安定となり，失敗してしまう可能性がある．

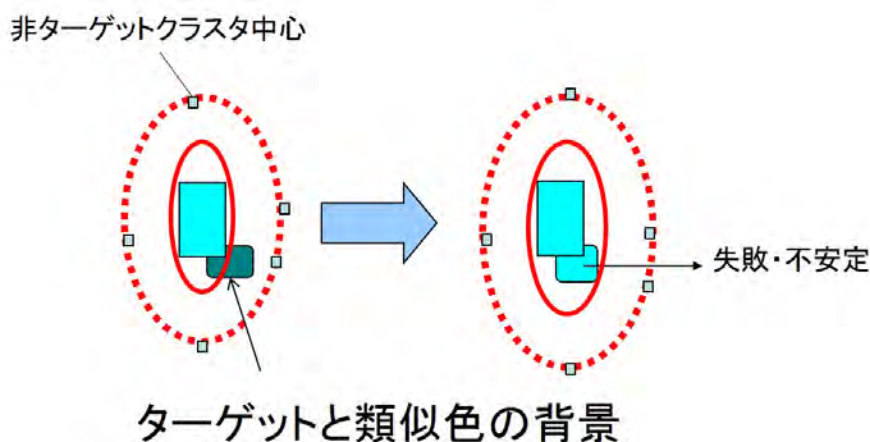


図 4.3 類似背景色問題

### 画像内のターゲットサイズ問題

空間的距離を画像上の画素単位で計算するため，ターゲットが小さいく写っている時は 2 次元座標空間における距離は小さくなり，ターゲットが大きく写っているときは 2 次元座標空間における距離は大きくなる．このように，画像内のターゲットの大きさによって，5 次元特徴空間内における 2 次元位置特徴ベクトルの寄与率が変わることで，大きく写っているターゲットの画素がすべてターゲット画素としてクラスタリングされない場合や，近接する背景が誤ってターゲットとしてクラスタリングされる場合がある．図 4.4 のように，背景上の注目画素（▲）について，ターゲットが小さい場合は  $D_T > D_N$  となり背景としてクラスタリングされるが，ターゲットが大きくなると  $D_T < D_N$  となりターゲットとしてクラスタリングされてしまう．

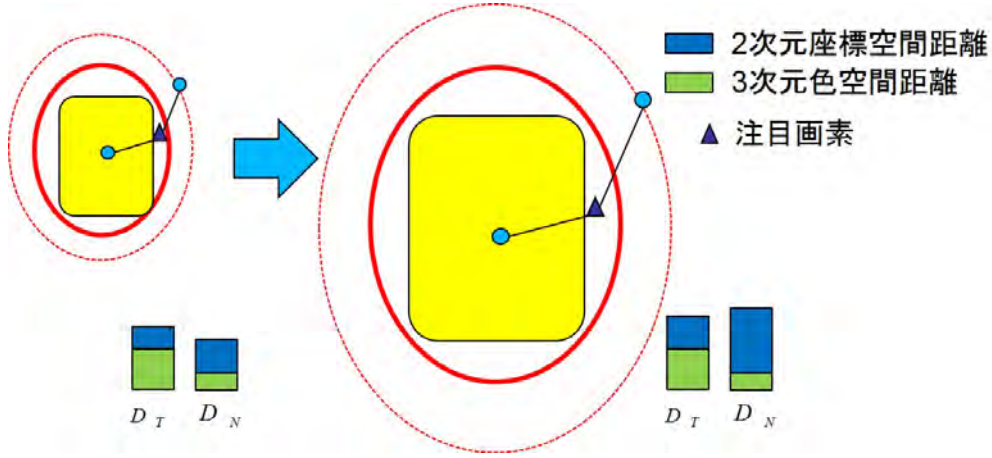


図 4.4 画像内のターゲットサイズ問題

## 4.2 FB-Clustering Tracker

威ら [112] は, 5 次元の K-means Tracker の問題を解決するために, 奥行き情報の導入とマハラノビス汎距離によって 2 次元座標空間内の距離評価を行う手法を提案し, FB-Clustering Tracker (Foreground and Background Clustering Tracker) と名付けた.

FB-Clustering Tracker では, ターゲットと背景をより分離しやすくして類似背景色問題を解決するために, 各画素を 3 次元色空間  $\mathbf{c} = [R, G, B]^T$  と 2 次元座標空間  $\mathbf{p} = [x, y]^T$  に加え, 1 次元の距離  $d$  からなる 6 次元ベクトル  $\mathbf{f} = [\mathbf{c}, \mathbf{p}, d]^T$  とした. 2 つの画素  $a, b$  の特徴ベクトル  $\mathbf{f}_a, \mathbf{f}_b$  間の距離は, 色特徴ベクトルの距離  $d_c$ , 座標特徴ベクトルの距離  $d_p$ , 奥行き要素の距離  $d_z$  に重み  $\alpha_1, \alpha_2$  を用いて次式のように定義される.

$$d(\mathbf{f}_a, \mathbf{f}_b) = d_c(\mathbf{c}_a, \mathbf{c}_b) + \alpha_1 d_p(\mathbf{p}_a, \mathbf{p}_b) + \alpha_2 d_z(d_a, d_b) \quad (4.14)$$

色特徴ベクトルの距離  $d_c$  は, 照明変化などの影響を少なくするため相関値によって評価され, 色空間内の 2 つの特徴ベクトルのなす角で与えられる.

$$d_c(\mathbf{c}_a, \mathbf{c}_b) = \cos \theta = \frac{\mathbf{c}_a \cdot \mathbf{c}_b}{\|\mathbf{c}_a\| \|\mathbf{c}_b\|} \quad (4.15)$$

座標特徴ベクトルの距離  $d_p$  は, 画像内のターゲットサイズに影響されにくいマハラノビス汎距離によって評価され, ターゲット画素集合の分散共分散行列  $\Sigma$  を用いて, 2 つの特徴ベクトルを確率変数としたベクトル間の非類似性として与えられる.

$$d_p(\mathbf{p}_a, \mathbf{p}_b) = \sqrt{(\mathbf{p}_a - \mathbf{p}_b)^T \Sigma^{-1} (\mathbf{p}_a - \mathbf{p}_b)} \quad (4.16)$$

奥行き要素の距離  $d_z$  は、ユークリッド距離で評価される。

$$d_z(d_a, d_b) = \|d_a - d_b\|^2 \quad (4.17)$$

これにより、ターゲットと似た色の背景を奥行きの違いにより正しくクラスタリングすることが可能であるが、画素間の空間的近接性を画像上での距離と奥行きの違いに分けて評価しているため、これらのバランスを考慮する必要がある。また、FB-Clustering Tracker では、計算上の関係で非ターゲットクラスタ中心の配置が等分割法であり、背景の環境が考慮されていない。そのため、背景となってほしい領域に非ターゲットクラスタ中心が配置されない場合、奥行き情報を生かせず追跡が不安定になる可能性がある。

### 4.3 6D K-means Tracker

K-means Tracker は、画像上の追跡対象と背景の両方に対して複数のクラスタ中心を割り当て、5次元特徴空間（色、画素位置）における K-means クラスタリングによって、サーチエリア内の各画素をターゲットと非ターゲットにラベリングすることにより追跡を行う手法である。特徴空間内でクラスタ中心と入力画素の距離を計算し、その距離に基づいてターゲットか非ターゲットのラベルが入力画素に付けられる。追跡対象が動けば、クラスタリングの結果、画像上でのターゲットクラスタ中心の重心位置が更新され、ターゲットの追跡が行われる。また、K-means Tracker は画素単位でのクラスタリングを行い、可変楕円をターゲット画素にフィッティングさせることで、追跡対象の大きさや形状の変化にも追従することができるため、様々な形状の髪領域を追跡するのに有効である。

しかし、従来の K-means Tracker には次のような問題点がある。追跡対象と類似する色を持つ背景画素がサーチエリアに混入した場合、5次元特徴空間ではその背景画素が誤ってターゲット画素としてクラスタリングされる可能性が高く、その影響によりターゲット領域やサーチエリアが不安定となり、追跡が失敗してしまう場合がある。また、空間的距離を画像上の画素単位で計算するため、ターゲットが小さく写っている時は2次元座標における距離は小さくなり、ターゲットが大きく写っている時は2次元座標における距離は大きくなる。このように、画像内のターゲットの大きさによって、5次元特徴空間内における2次元位置特徴ベクトルの割合が変わることで、大きく写っているターゲットの画素が、すべてターゲット画素としてクラスタリングされない可能性が高くなる。さらに、クラスタ中心の初期位置を手動で指定しなければならないという問題がある。

そこで本論文では、大池らの手法 [113] をベースとし、各画素をワールド座標空間に変換した上で画素間の空間的近接性を評価する 6D K-means Tracker を提案する。本章では、各画素の位置情報を3次元化することで、K-means Tracker の2つの問題を同時に

解決しつつ提案システムを構築する方法について述べる．提案システムの特徴は，以下のようになっている．

- 3次元の色情報と3次元の空間位置情報からなる6次元特徴空間における6D K-means Trackerにより，安定な追跡を行う．
- 顔検出と6D K-means Trackerを組み合わせ，検出された顔領域に基づいて，ターゲット（髪・顔）と背景のクラスタ中心の初期特徴ベクトルを自動的に決定する．

### 4.3.1 RGB-D カメラ

RGB-Dカメラとは，カラー画像(RGB)に加えて奥行き画像(DepthあるいはDistance)を取得できるカメラデバイスであり，距離計測センサの性能向上により手軽に3次元計測が行えるようになってきた．本髪追跡システムで用いる6D K-means Trackerは，画素単位のクラスタリングを行っており，奥行き情報による拡張を行うため，画素毎の奥行き情報を高速に取得できるKinect[116]と呼ばれるRGB-Dカメラを用いる．Kinectは，Microsoft社製のゲーム機のユーザインタフェース用デバイスとして開発され，被写体のカラー画像と距離画像をリアルタイムに撮影できる安価なデバイスである（図4.5）．Kinectを用いることで，解像度が $640 \times 480$ 画素のカラー画像と距離画像を高速（30fps）に取得することができる．また，計測可能な距離範囲は $500mm \sim 10000mm$ となっている．図4.6にKinectで取得された画像例を示す．距離画像については，距離値を $0 \sim 255$ に圧縮して表示しており，明るいほどKinectまでの距離が近いことを意味している．



図 4.5 Microsoft 社の Kinect



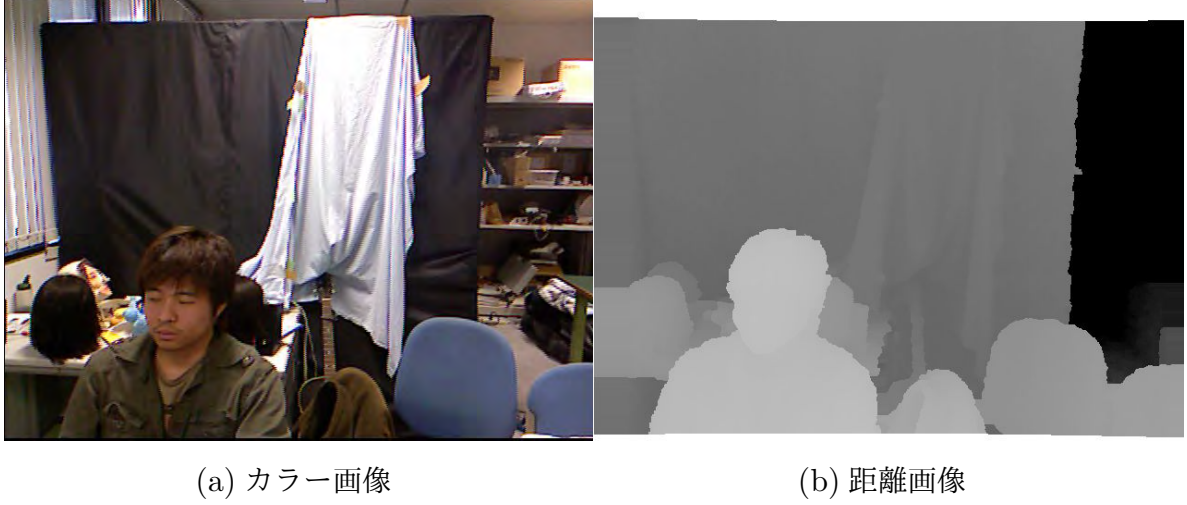


図 4.6 Kinect で得られる画像

### 4.3.2 6次元特徴空間でのクラスタリング

本手法では、Kinect で獲得されたターゲット、背景までの奥行き情報とそれぞれの色情報を活用し、奥行き特徴を K-means Tracker の 5 次元特徴空間に追加して 6 次元に拡張する（図 4.7 参照）。これによって、K-means Tracker の類似背景色問題とターゲットサイズ問題が同時に解決される。

画素の位置  $(x, y)$  を表現する 2 次元ベクトルを 3 次元カメラ座標空間  $(X, Y, Z)$  に射影する。

$$\mathbf{p}_2 = [x \ y]^T \rightarrow \mathbf{p}_3 = [X \ Y \ Z]^T \quad (4.18)$$

また、追跡対象を構成する画素の色の類似性と空間的近接性を同時に表現するために、各画素の特徴を 3 次元色空間  $\mathbf{c}_3 = [R \ G \ B]^T$  と 3 次元カメラ座標空間  $\mathbf{p}_3 = [X \ Y \ Z]^T$  からなる 6 次元ベクトル  $\mathbf{f}_6 = [\mathbf{c}_3 \ \mathbf{p}_3]^T$  として扱う。2 つの画素  $a, b$  間の距離  $d(\mathbf{f}_6^a, \mathbf{f}_6^b)$  は、色空間内と 3 次元カメラ座標空間内のユークリッド距離に重み  $\alpha$  を用いて、次式のように定義される。

$$d(\mathbf{f}_6^a, \mathbf{f}_6^b) = \|\mathbf{c}_3^a - \mathbf{c}_3^b\|^2 + \alpha \|\mathbf{p}_3^a - \mathbf{p}_3^b\|^2 \quad (4.19)$$

ターゲットと背景が空間的に離れていれば、サーチエリア内に混入したターゲットと類似色の背景画素が、式 (4.19) より異なる  $d(\mathbf{f}_6^a, \mathbf{f}_6^b)$  を求めることができるため、正しくクラスタリングされる。また、各画素を 3 次元カメラ座標系に変換した上で空間的近接性を求めているため、ターゲットサイズが一定であれば、カメラまでの距離変化によって起こる画像内ターゲットサイズの変化による問題も解決できる。

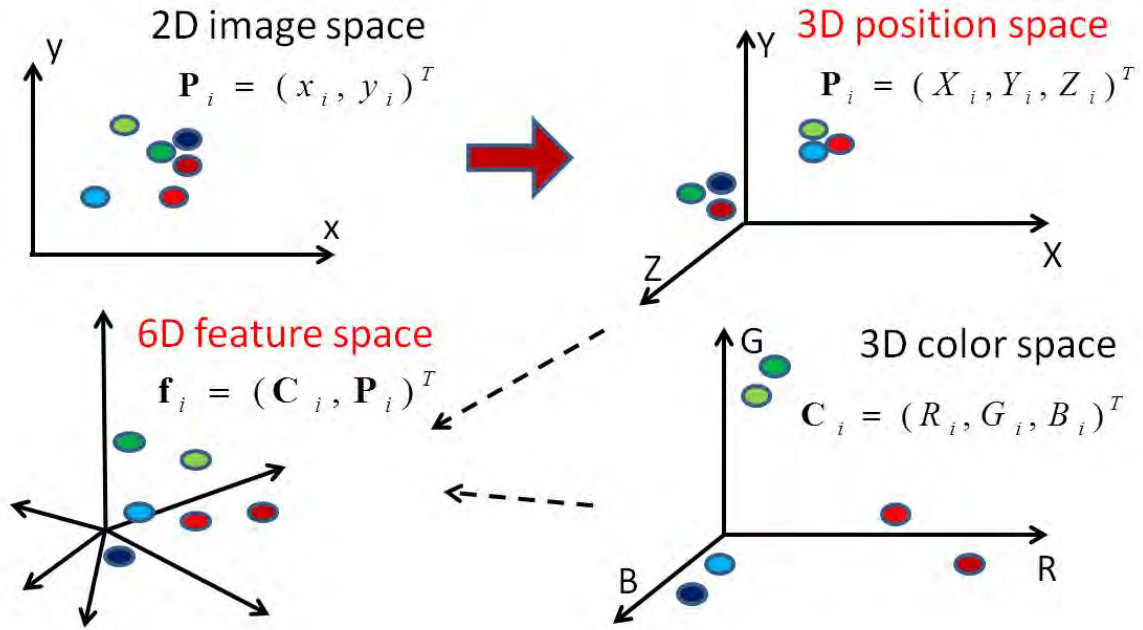


図 4.7 6次元特徴空間

本手法では、クラスタが未知である注目画素  $\mathbf{f}_6^u$  からターゲットクラスタ中心  $\mathbf{f}_6^T$  への最短距離  $D_T$ 、非ターゲットクラスタ中心  $\mathbf{f}_6^{NT}$  への最短距離  $D_{NT}$  は、以下のように定義される (図 4.8)。

$$D_T = \min_{i=1 \sim n} \{d(\mathbf{f}_6^{Ti}, \mathbf{f}_6^u)\} \quad (4.20)$$

$$D_{NT} = \min_{j=1 \sim m} \{d(\mathbf{f}_6^{NTj}, \mathbf{f}_6^u)\} \quad (4.21)$$

ここで、 $n, m$  は、それぞれターゲットクラスタ中心と非ターゲットクラスタ中心の個数であり、 $n$  は追跡初期フレームで設定され (4.4.2 参照)、 $m$  はフレーム毎に動的に設定される (4.3.3 参照)。  $D_T$  と  $D_{NT}$  を比較することによって、注目画素  $\mathbf{f}_6^u$  がターゲットクラスタか非ターゲットクラスタにクラスタリングされる。

$$\mathbf{f}_6^u \rightarrow \text{Traget} \quad \text{if} \quad \{D_T(t) < D_{NT}(t)\}. \quad (4.22)$$

$$\mathbf{f}_6^u \rightarrow \text{Non-Traget} \quad \text{if} \quad \{D_T(t) > D_{NT}(t)\}. \quad (4.23)$$

### 4.3.3 クラスタ中心の更新

本手法では、可変楕円によるターゲット領域の記述は従来と同様の方法で行われ、ターゲットモデルと背景モデルを表すクラスタ中心の更新は、拡張された 6 次元特徴空間にお

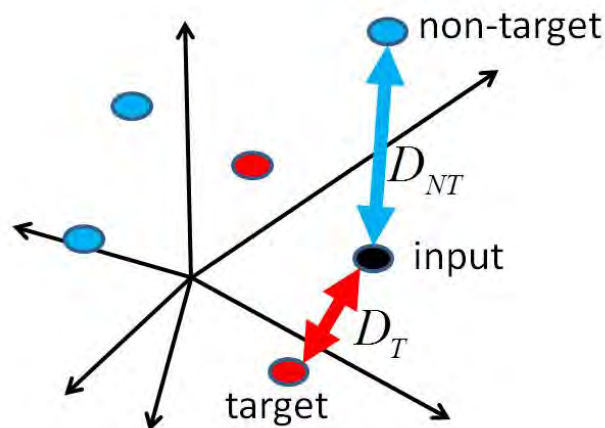


図 4.8 クラスタリングにおける注目画素とクラスタ中心の距離

いて行われる。

従来手法と同様に，ターゲットとしてクラスタリングされた画素集合はガウス確率密度関数で近似することができると仮定する．ターゲットとしてクラスタリングされた画素の分布のマハラノビス距離より可変楕円を求め，ターゲット画素を 95% 含む時のパラメータ（長軸，短軸，傾き，重心）を用いてターゲットの形状が記述される．求めた楕円を一定倍数拡大した楕円が次フレームのサーチエリアとなる．

ターゲットクラスタ中心は，ターゲットとしてクラスタリングされた画素における特徴ベクトルの平均により更新される．非ターゲットクラスタ中心は，サーチエリアの輪郭上を背景と仮定し，この輪郭上の画素を非ターゲットクラスタ中心の候補点として次のようにして配置される（4.1 節を参照）．

まず，輪郭上の画素の 1 点が非ターゲットクラスタ中心として登録される．次に，輪郭上の画素を順にクラスタリングし，ターゲットとしてクラスタリングされたものが，非ターゲットクラスタ中心として追加登録されていく．このように，輪郭上を走査してターゲットにクラスタリングされそうな領域の 1 点を非ターゲットクラスタ中心とすることで，各フレーム毎の背景に応じた非ターゲットクラスタ中心の個数や配置が決定される．

このように，サーチエリアとクラスタ中心を毎フレーム更新することで，画像内の追跡対象の大きさや形状の変化，背景の変化に対して安定な追跡が行える．

また，適応的な非ターゲットクラスタ中心の配置により，特徴空間を 6 次元に拡張した提案手法では，背景クラスタの数を従来手法よりも減少させることができる．従来手法では，背景にターゲットとの類似色があった場合，非ターゲットクラスタ中心の数が増えてしまう傾向にある．提案手法では，色が似ているクラスタ中心間でも奥行きの違いによっ

て特徴空間内での距離が大きくなるため、ターゲットと背景の分類が容易になる。そのため、少ない非ターゲットクラスタ中心の数で背景をモデル化することが可能となり、次元拡張による距離計算のコスト増加を抑制することができる。

## 4.4 髪・顔・頭部領域の検出・追跡システム

### 4.4.1 システム構成

奥行き情報の追加により拡張された 6D K-means Tracker による追跡と奥行き情報を利用した高速な顔検出（第2章）を組み合わせることで、自動的に顔や髪領域の追跡を開始できるシステムを構築することができる。

図 4.9 に提案システムが髪を追跡する場合の構成を示す。追跡開始時に必要なターゲット領域と背景領域を顔検出によって得られた位置情報から取得する。そのため、本システムでは、まず顔検出処理を開始し、顔が検出されれば追跡処理へ移行する。追跡が失敗した場合は顔検出処理へ移行する。提案システムでは、追跡中にターゲットとしてクラスタリングされた画素が極端に少なくなった場合、対象の追跡に失敗したと判断している。

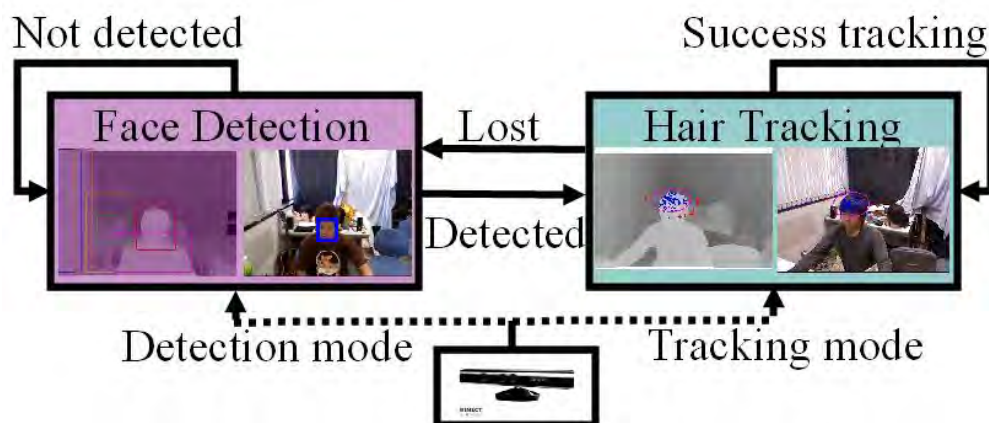


図 4.9 提案システムの構成図（髪を追跡する場合）

### 4.4.2 ターゲットと非ターゲットクラスタ中心の自動初期化

顔検出に成功した場合、顔検出によって得られた矩形を用いて、追跡初期フレームのターゲット、非ターゲットクラスタ中心の 6 次元特徴ベクトルが対象領域の画素から自動的に設定される。

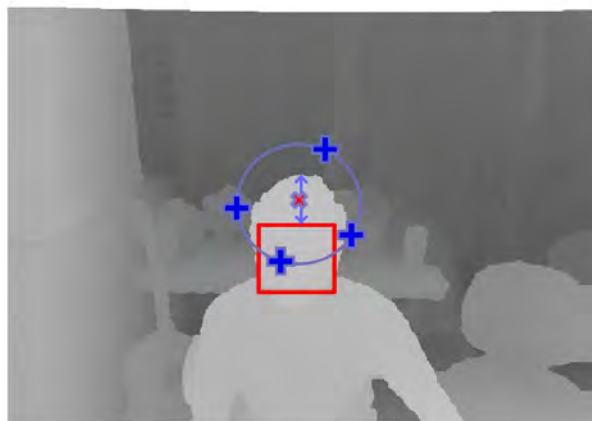
髪を追跡する場合，図 4.10(a) に示すように，ターゲットクラスタ中心（赤点）は，顔として検出された赤矩形の上辺の中点から上方向のデプスのエッジ（デプスの値が極端に変わる）までの中点とする．非ターゲットクラスタ中心（青プラス）は，ターゲットクラスタ中心を円心，半径を顔サイズ（赤矩形の幅）とする青円上に自動的に配置（4.3.3 を参照）される．このように，ターゲットクラスタ中心は追跡対象から自動的に獲得されるため，髪の色に関係なく追跡を行うことができる．

提案手法は髪追跡システムを構築できるだけではなく，初期クラスタの配置によって，顔追跡システムや，頭部（顔と髪の同時）追跡システムも簡単に構築することができる．

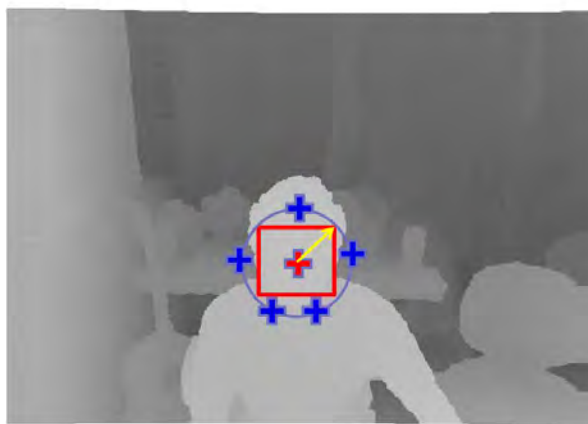
顔を追跡する場合，図 4.10(b) に示すようにターゲットクラスタ中心は顔検出によって得られた赤矩形の中心，非ターゲットクラスタ中心は矩形の外接円上に配置され，その円を初期フレームにおける可変楕円として追跡が開始される．

頭部を追跡する場合は，図 4.10(c) に示すように 2 つのターゲットクラスタ中心を顔と髪の場合と同様に配置し，2 つのターゲットクラスタ中心の中点から赤矩形の右下までを半径とする円周上に非ターゲットクラスタ中心を配置する．

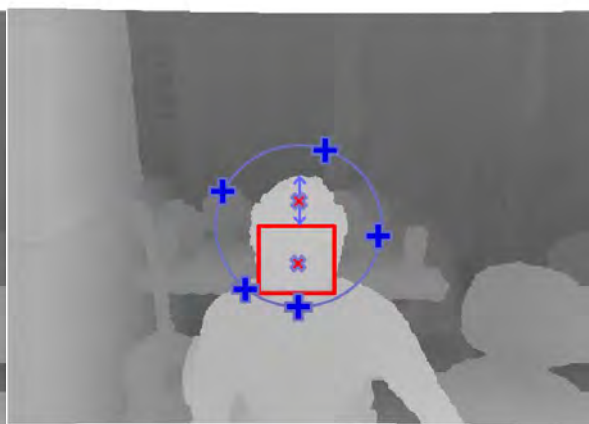
頭部追跡を行った場合，頭部の回転運動などで顔領域をロストしても髪領域だけで追跡を継続することができ，顔領域が出現すると再び顔領域の追跡を再開することができる．



(a) 髪追跡の初期化



(b) 顔追跡の初期化



(c) 頭部追跡の初期化

図 4.10 追跡の初期化

## 4.5 実験

提案手法の有効性と安定性を確認するため、様々な環境での実験を行った。全ての処理はビデオレートで行えることが確認できている。

実験では、CPU が Intel Core i5-750、メモリが 2GB の汎用 PC を用いている。1 フレームの処理時間について、顔検出処理は約 14ms、追跡処理は、対象が一人の場合、約 10msec であった。また、式 (4.19) における  $\alpha$  を予備実験により決定した 0.4 に固定して全ての実験を行っており、追跡対象の色が異なってもこの固定値で安定な追跡が可能であった。4.3.3 節で述べたサーチエリアの倍率は 1.5 としている。

なお、以下の実験結果の画像では、矩形は顔検出の結果、楕円はターゲットを記述する追跡結果の可変楕円であり、青色の画素はターゲットとしてクラスタリングされた画素、赤十字はサーチエリアの輪郭上に自動配置された非ターゲットクラスタ中心である。

### 4.5.1 従来手法との比較実験

K-means Tracker は、対象追跡において、可変テンプレートマッチング法や Mean Shift 法に比べ有効であることが示されている [114, 111]。本項では、奥行き情報の追加によって、高速性を損なわずにより安定に追跡が行えることを示す。

#### 非ターゲットクラスタ中心数と速度

5 次元特徴空間を用いた従来手法と奥行き情報により特徴空間を 6 次元に拡張した提案手法による追跡中の非ターゲットクラスタ中心の数に関する比較を行った。

図 4.11 は、300 フレームの画像系列に対し追跡を行った際のフレームごとの非ターゲットクラスタ中心数の変化を比較したものである。横軸がフレーム数、縦軸が非ターゲットクラスタ中心の数となっており、両手法ともに追跡は 58 フレーム目から開始され、最終フレームまで継続された。従来手法（図の青色●）では、非ターゲットクラスタ中心の数が 10 前後であるのに対し、提案手法（図の赤色▲）では、4 つ前後となっており、特徴次元を拡張したことによってより少ないクラスタ数で背景モデルを表現し、追跡が行われていることがわかる。

図 4.12 は、追跡中のフレームごとの処理時間をグラフにしたものである。両手法ともに処理時間は約 10msec となっている。提案手法では、クラスタリングを行う際に各画素を 3 次元ワールド座標空間に変換する処理が必要となるため、特徴同士の距離計算にかかる処理コストが増加してしまう。しかし、非ターゲットクラスタ中心の数を少なくするこ



とができているため、追跡処理全体としては、従来手法と同程度の処理時間で行うことができる。

図 4.13 は、同実験の追跡処理時の様子を示しており、赤色の十字が配置された非ターゲットクラスタ中心である。従来手法では、背景部分に多くの非ターゲットクラスタ中心が配置されているのに対し、提案手法では、追跡対象と奥行きに差がある背景部分において非ターゲットクラスタ中心の数が少なくなっていることがわかる。このように、提案手法では、より効率的に非ターゲットクラスタ中心を配置することができるようになったため、次元拡張による計算コストの増加を抑え、従来手法と同様に高速な処理が可能となる。

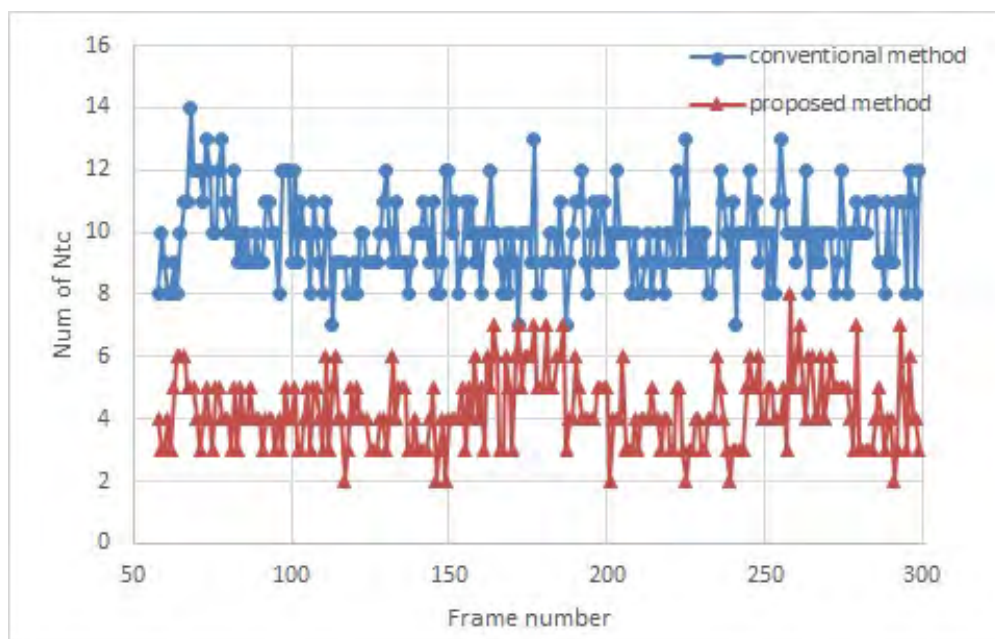


図 4.11 追跡時の非ターゲットクラスタ中心の数

#### 類似色背景の場合

図 4.14 は、300 フレームの画像系列に対して従来手法と提案手法による髪の追跡を行い、安定性に関する比較を行ったものである。この画像系列では、追跡の安定性を比較するために、ターゲット（黒髪）を固定し、背景のみを変化（白と黒のチェッカーボードが横切る）させている。

図 4.14(a) は従来手法による追跡を行った結果であるが、188 フレームや 246 フレームのように背景にターゲットと類似色のチェッカーボードが近づいた時に、追跡楕円が広がってしまっていることがわかる。一方、図 4.14(b) のように提案手法では、全フレーム

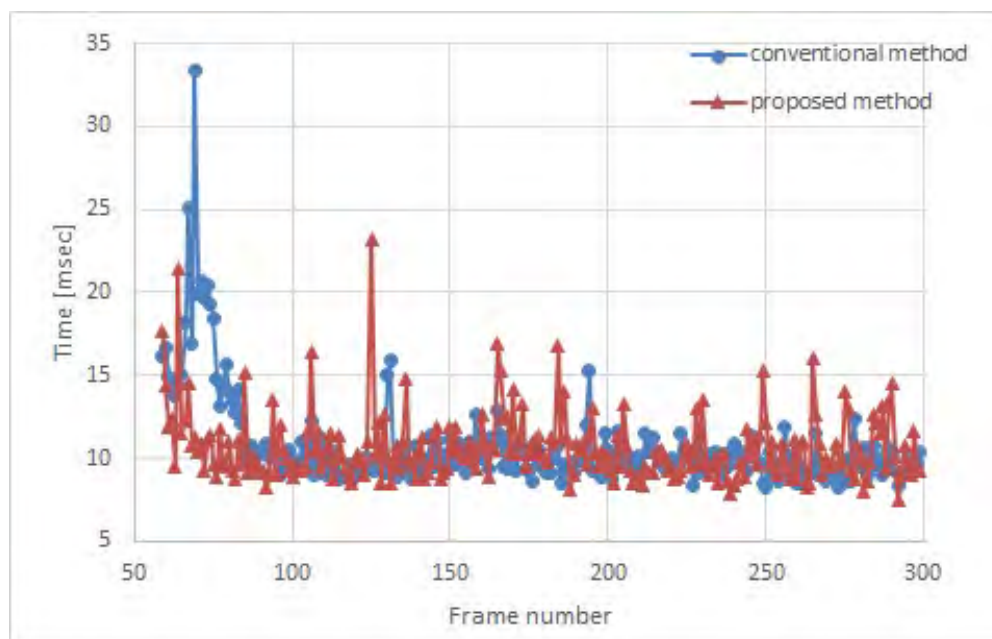


図 4.12 追跡処理にかかる時間

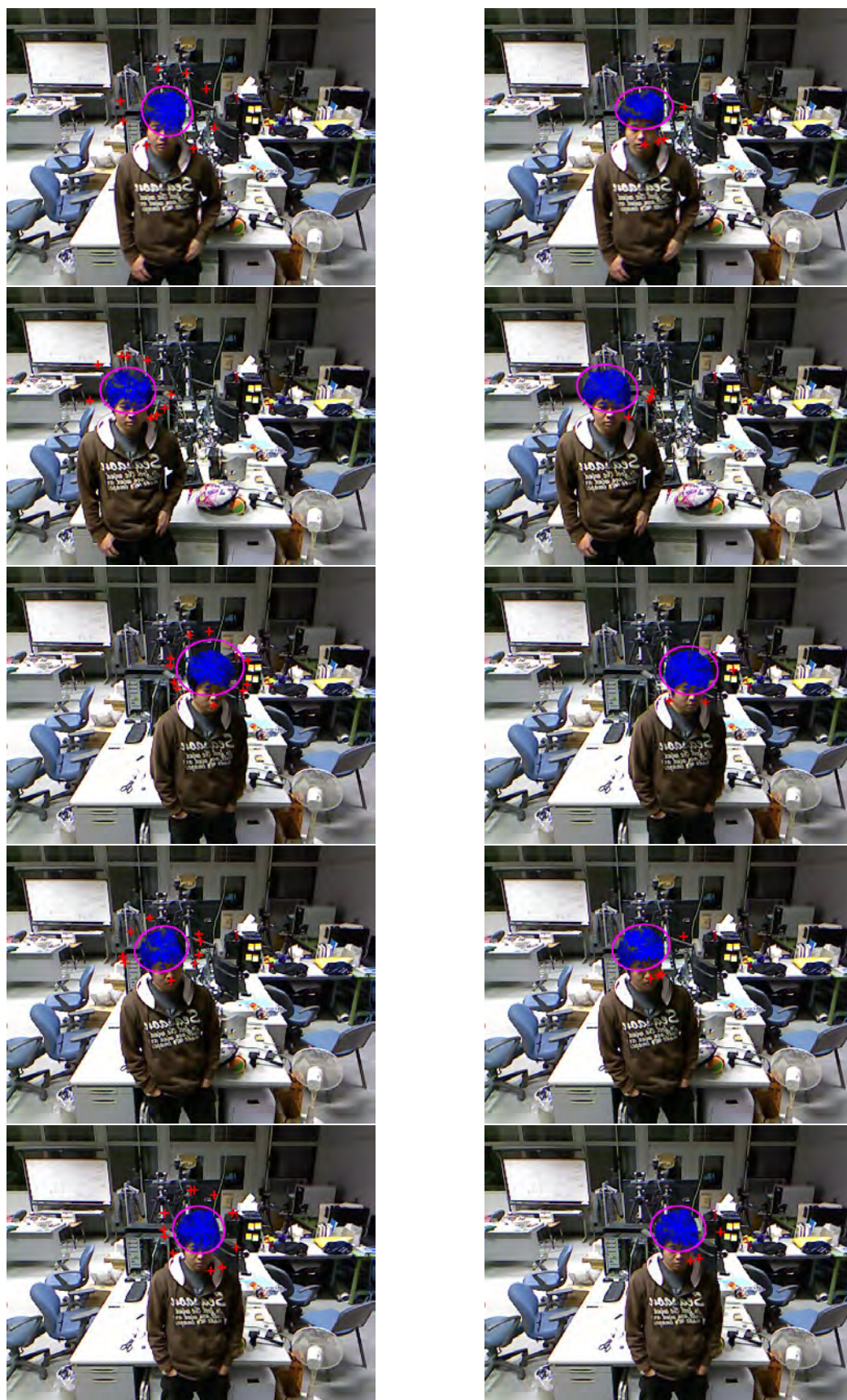
において安定した追跡を行えていることが確認できた．図 4.15 に各フレームにおける追跡楕円の長軸の長さをグラフにしたものを示す．提案手法では，楕円の大きさが一定となっていることがわかる．

また，表 4.1 は，図 4.14 に示す動画の追跡中の楕円パラメータ（楕円中心の  $x$  座標， $y$  座標，楕円の長軸，短軸）の標準偏差を比較したものである．ターゲットが固定されているため，楕円パラメータの標準偏差が 0 に近いほど正しく追跡できていることを意味するが，提案手法は数ピクセルの誤差となっており，追跡結果が安定していることがわかる．

表 4.1 追跡楕円パラメータの標準偏差（単位：画素）

	$x$ 座標	$y$ 座標	長軸	短軸
従来手法	4.33	4.6	13.19	12.06
提案手法	0.63	0.67	1.07	1.19

このように，背景がターゲットと類似色の場合，従来の 5 次元特徴空間における K-means Tracker では，背景がターゲット画素としてクラスタリングされてしまい，対象領域が広がって追跡が不安定になっているのに対し，奥行き情報を利用して特徴空間を 6 次元に拡張した提案手法では，安定した追跡を行うことができる．



(a) 従来手法

(b) 提案手法

図 4.13 従来手法と提案手法の追跡結果例



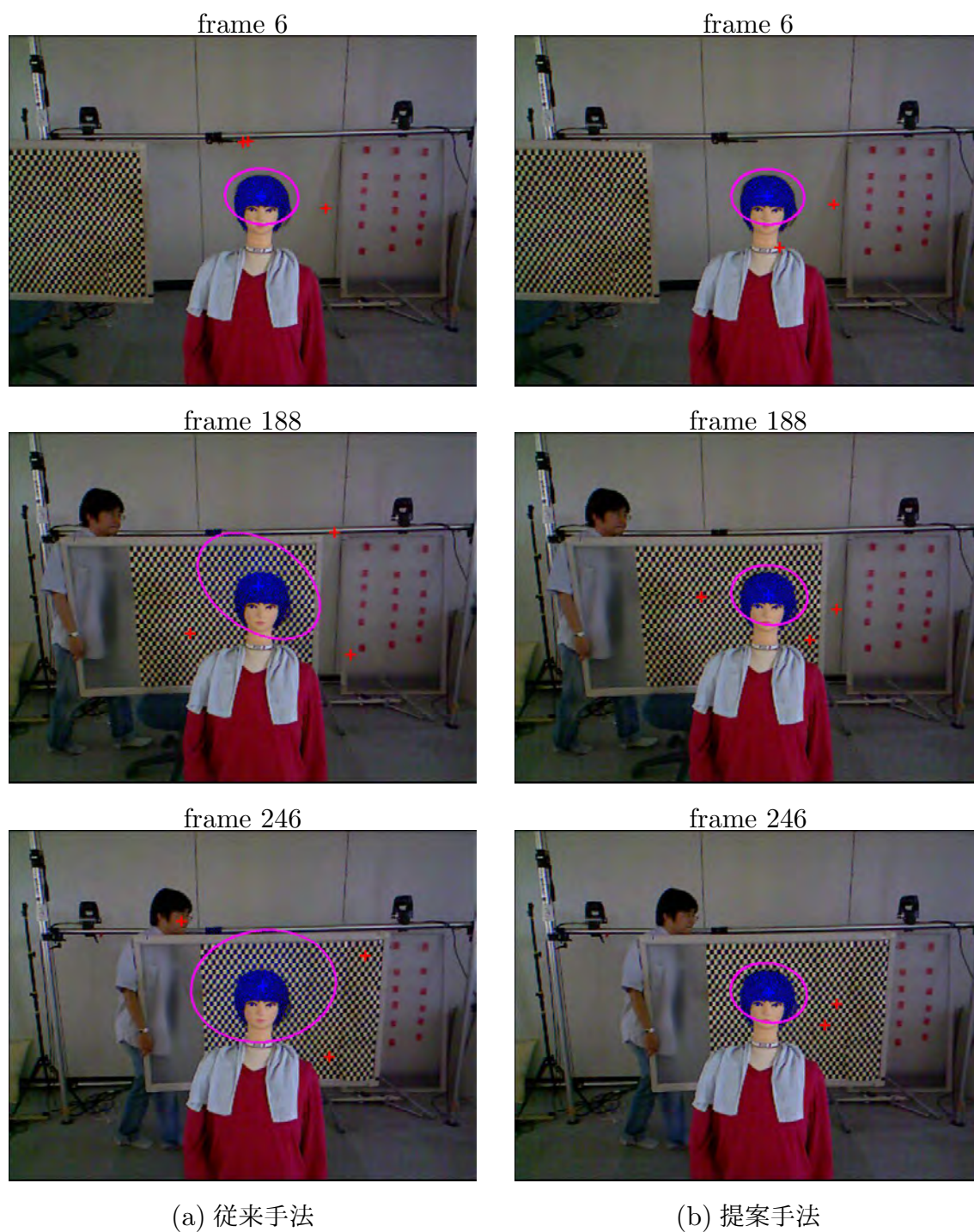


図 4.14 類似背景色の髪追跡結果例

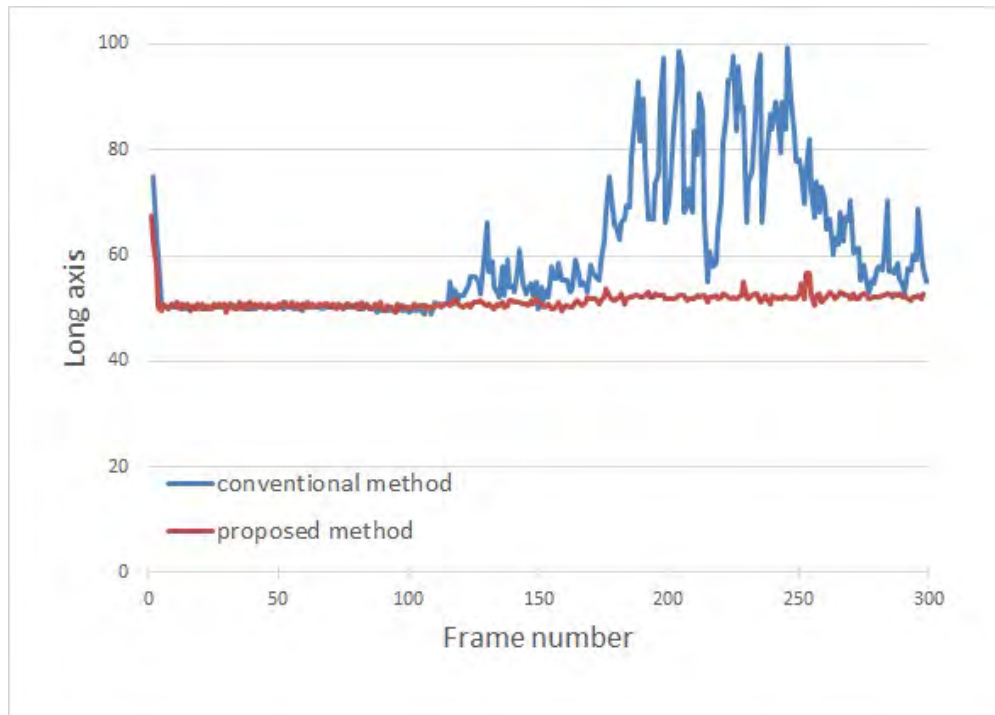


図 4.15 追跡楕円の長軸の変化

#### ターゲットサイズが変わる場合

カメラから追跡対象までの距離の変化で、画像上のターゲットサイズが変わる場合の従来手法と提案手法による比較実験を行った。ターゲットサイズが徐々に大きくなるような 300 フレームの画像系列を用い、顔と髪からなる頭部領域の追跡を行った。この実験では、背景にターゲットと類似色の段ボールがある。

図 4.16 に追跡結果の一部を示す。従来の 5 次元特徴空間における追跡では、ターゲットサイズが小さい時には近くにターゲットと類似色の背景があっても安定したクラスタリングが行える。しかし、ターゲットサイズが大きくなるにつれて 2 次元座標空間における距離が占める割合が大きくなるため、誤ったクラスタリングが起こりやすくなる。図 4.16(a) のように、216 フレーム目では、一部段ボールの背景領域がターゲットの方が近いと判断され、誤ってターゲットとしてクラスタリングされてしまったことで追跡楕円が広がっている。このように、誤ってクラスタリングされた結果が以降のフレームに影響を与え、追跡が不安定になったり失敗してしまうことがある。

一方、提案する 6 次元空間における追跡では、空間的近接性を 3 次元のワールド座標空間において評価しているため、ターゲットの大きさは常に一定の大きさとなる。図

4.16(b) に示すように，提案手法では終始安定した追跡を行うことができた．

また，追跡の安定性を評価するため，図 4.17 のように頭部領域を表す正解画像を用意し，各フレームにおいてターゲットとしてクラスタリングされた画素が頭部領域であった割合で評価した．

$$\text{ターゲット正解率} = \frac{\text{頭部領域にあるターゲット画素数}}{\text{ターゲット画素数}}$$

従来手法と提案手法の各フレームにおけるターゲット正解率をグラフにしたものを図 4.18 に示す．従来手法では，200 フレーム目付近から誤ったクラスタリングが増えているが，提案手法では，全フレームにおいて安定していることが分かる．

以上の結果より，6 次元特徴空間を用いた本手法が，特徴空間の拡張に伴う処理コストの増加を抑えつつ追跡の安定化に貢献できることを確認できた．





図 4.16 ターゲットサイズが変化する場合の追跡結果



(a) 入力画像



(b) 頭部領域を表す画像

図 4.17 頭部領域を表す正解画像の例

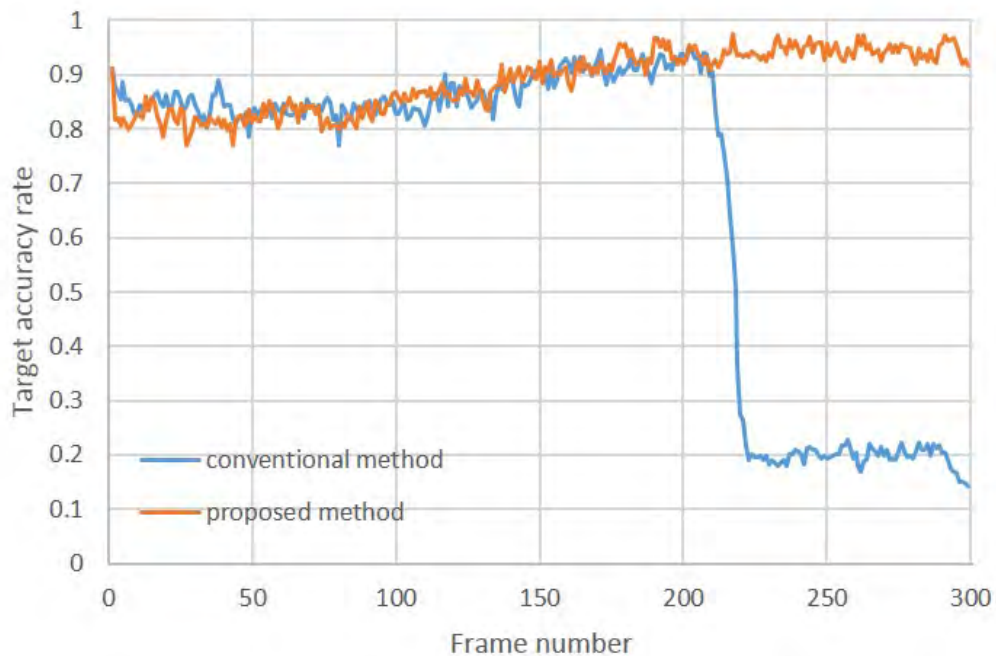


図 4.18 ターゲット正解率

#### 4.5.2 他の追跡手法との比較実験

提案手法と Particle Filter[86, 93] による髪の追跡を行った。Particle Filter は、確率密度分布を多数のサンプル（パーティクル）で近似し、確率の高い領域を対象として追跡する手法である。実験では、各パーティクルに尤度を与える尤度関数を以下のように 2 種類定義した。

尤度関数  $L(d)$  は、ターゲットとパーティクルの距離  $d$  に対して、平均を 0, 分散を  $\sigma$  として持つような正規分布として定義する.

$$L(d) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d^2}{2\sigma^2}\right) \quad (4.24)$$

ターゲットとパーティクルの距離について、色情報  $\mathbf{c} = [R, G, B]^T$  のみを用いた  $d_1$  と、それに奥行き情報  $z$  を加えた  $d_2$  として 2 種類定義した.  $d_1$  は、ターゲットの色  $\mathbf{c}_t$  とパーティクルの色  $\mathbf{c}_p$  のユークリッド距離とする.

$$d_1 = \|\mathbf{c}_t - \mathbf{c}_p\|^2 \quad (4.25)$$

$d_2$  は、ターゲットの奥行き  $z_t$  とパーティクルの奥行き  $z_p$  のユークリッド距離を加え、重み  $\alpha$  を用いて次のように定義する.

$$d_2 = \|\mathbf{c}_t - \mathbf{c}_p\|^2 + \alpha \|z_t - z_p\|^2 \quad (4.26)$$

実験では、 $\sigma$  を 30,  $\alpha$  を 0.4, パーティクルの数を 3000 としている. また、ターゲットの情報は 4.4.2 で述べた髪領域のターゲットクラスタ中心の設定方法と同じように取得することで、提案手法と同様に自動で追跡が開始できるようにした.

実験に用いた画像系列は、対象が自由に動くようなものとなっている. 追跡結果の一部を図 4.20 に示す. Particle Filter の結果画像では、青色の画素がパーティクルの位置を表し、黄色の十字がパーティクルの重心を表している. 図 4.20(a) のように Particle Filter で色情報のみを用いた場合では背景の暗い部分と髪領域を区別することができず、追跡が失敗してしまっている. 一方、図 4.20(b) のように奥行き情報を追加した場合は、髪領域の位置を正しく追跡できていることがわかる.

このように、奥行き情報を用いることで追跡手法の性能を大きく向上させることができる. 図 4.21 は、60 フレームから 180 フレームまでの、Particle Filter におけるパーティクルの重心と提案手法におけるターゲット画素の重心の軌跡をグラフにしたものである. 重心の推定精度を評価するため、図 4.19 のように各フレームにおいて頭部の位置を表す基点（白色の十字）を両眉の中心に手動で設定し、各手法で得られた重心との距離（画像上でのユークリッド距離）を計測した. この距離が一定の値であるほど追跡が安定しているといえる. 図 4.22 に、追跡開始から最終フレームまでの各手法で得られた重心と基点との距離をグラフにしたものを示す. また、各手法の標準偏差は表 4.2 のようになっている. 奥行き情報を追加した Particle Filter 2 と提案手法は誤差が数ピクセルで基点から重心までの距離が一定の値となっており、安定して髪領域の位置を取得できていることがわかる.



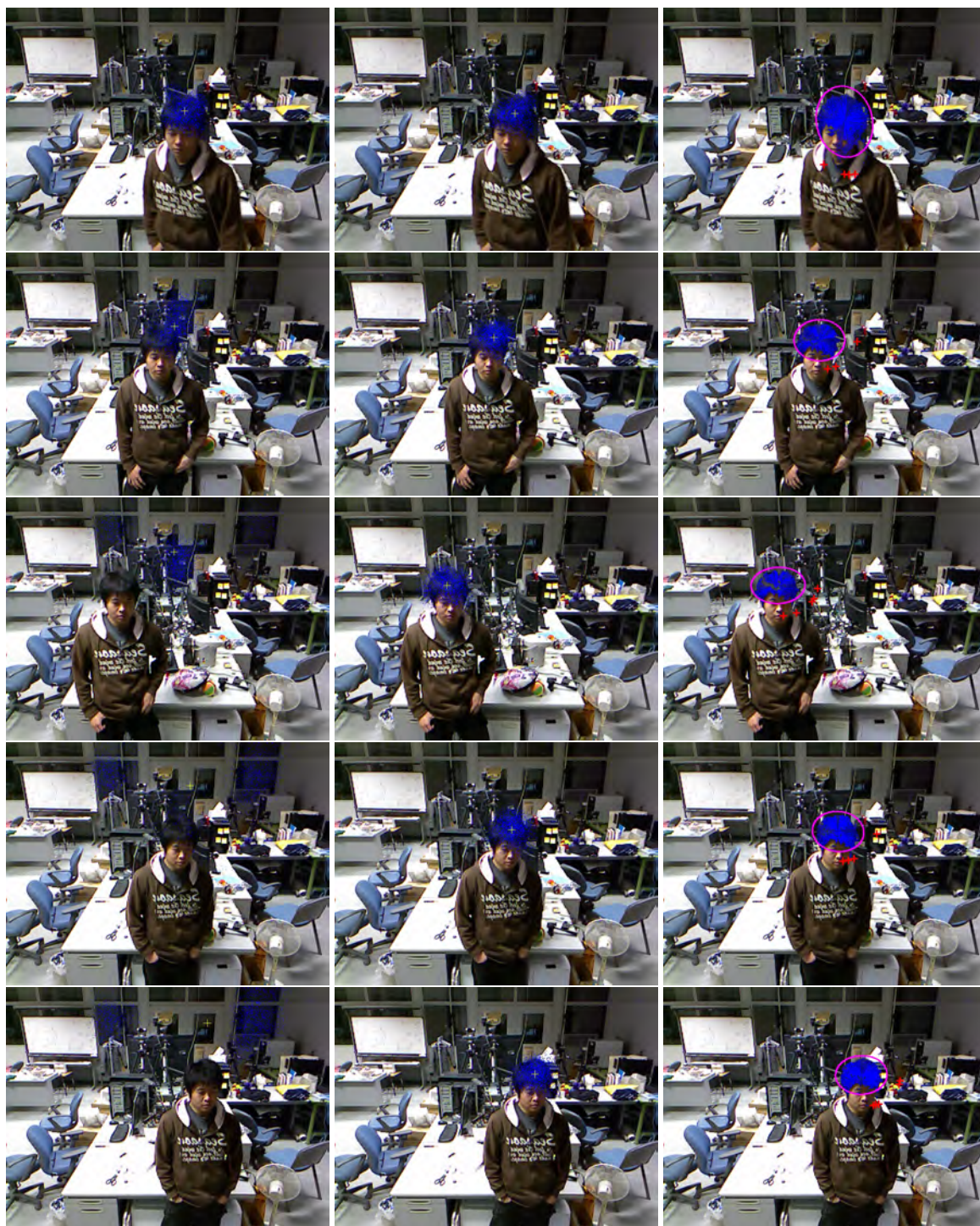
図 4.19 頭部の基点（白色の十字）

表 4.2 基点と重心の距離の標準偏差（単位：画素）

Particle Filter 1	Particle Filter 2	提案手法
46.60	2.37	1.84

Particle Filter では、色情報だけを用いた場合、髪領域を追跡することが難しかったが、奥行き情報を追加することで、提案手法と同程度の精度で髪領域の位置を得ることができた。しかし、Particle Filter は、ランダムに配置されたパーティクルの位置でのターゲットらしさを確率として得ているため、髪の大まかな領域はわかるが、画素単位で髪領域を得ることはできない。一方、図 4.20(c) のように提案手法では、画素単位でターゲットか背景かの判別を行っているため、髪領域だけをより密に取得できていることがわかる。また、Particle Filter は、尤度関数やパーティクルの数など多くのパラメータによって追跡性能が決まるため、安定した追跡を行うには処理コストも考慮した上でこれらを吟味する必要があるが、提案手法は、シンプルな処理で高速かつ安定した追跡を行うことができる。



(a) Particle Filter 1 ( $d_1$ )(b) Particle Filter 2 ( $d_2$ )

(c) 提案手法

図 4.20 Particle Fileter と提案手法の比較

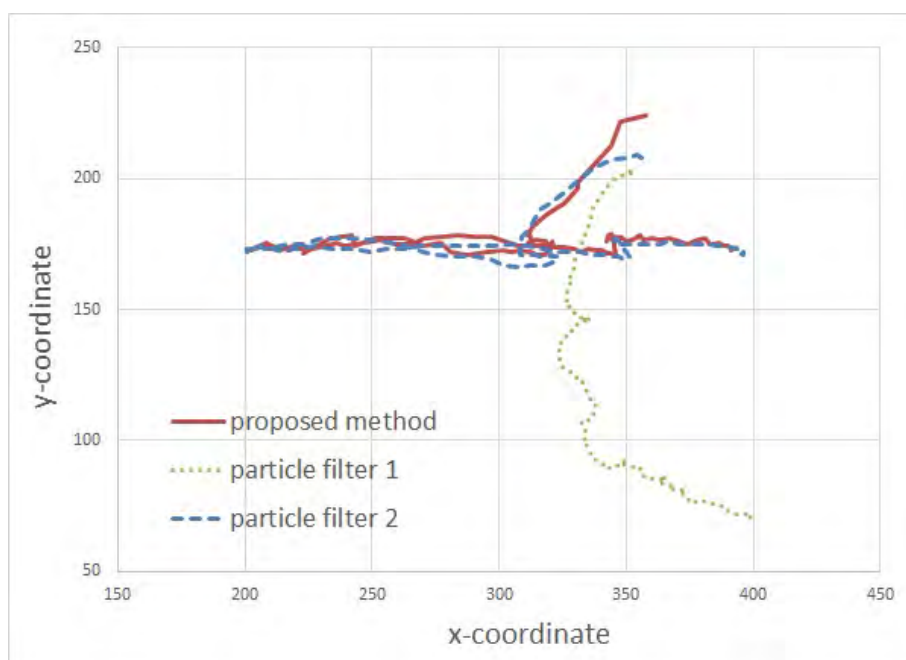


図 4.21 重心の軌跡（画像の x, y 座標）

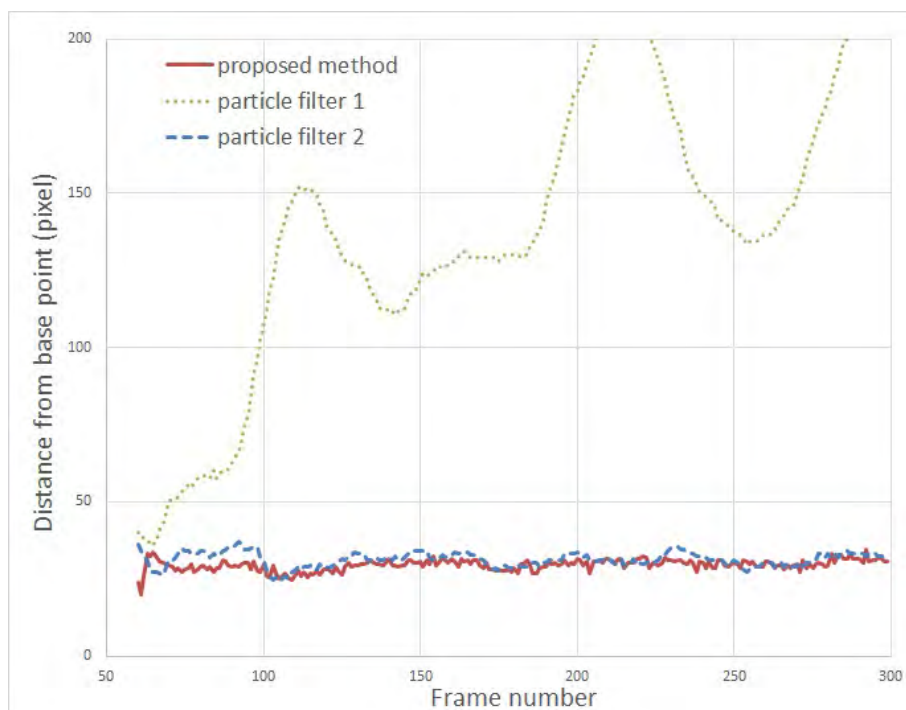


図 4.22 各フレームの基点と重心の距離（画素）



### 4.5.3 様々な髪の追跡結果

図 4.23 は、髪の色やスタイルの異なる複数の対象を追跡した結果例である。この実験で用いた画像系列は約 30 秒で、頭部の回転運動を伴った移動を行う 3 体の対象（茶髪セミロング、金髪ショートのマネキンと黒髪ショートの人）に対し、検出・追跡を行なっている。94, 213, 421 フレーム目でそれぞれ 1 人目, 2 人目, 3 人目が検出された後、追跡が開始され、最終フレームまで 3 人とも追跡は継続された。

このように、提案手法では髪の色やスタイルに関わらず追跡を行うことが可能であり、ターゲットとしてクラスタリングされた画素により髪領域を得ることができる。また、追跡中にも顔検出を行い新たな顔の出現を検出することで、複数対象の追跡を行うことができる。しかし、追跡対象を 1 人に限定した場合、図 1.11 のように検出と追跡が交互に行われるため高速な処理が可能であるのに対し、複数対象を追跡する場合、顔検出処理を常に行う必要があり、また、追跡中の対象が増えると追跡処理のコストも増えるため、対象が増えるに連れてビデオレートでの処理は困難となる。

提案手法で構築された顔検出と顔・髪追跡システムを用いた国内・国際会議上の実機デモ（研究業績 8, 12）において、100 人程度の人種や年齢の異なる人を対象に実験を行った。デモでは、肌の色、髪の色やスタイルに関わらず、追跡が自動的に開始でき、安定な追跡を行えることが確認できている。

しかし、長髪で髪と似た色の服を着ているとターゲット領域が服にまで拡大してしまい追跡が不安定になるケースや、髪領域の初期特徴をハイライト部分から取得してしまい安定な追跡が行えないケースもあった。

### 4.5.4 頭部追跡

提案システムでは、顔領域と髪領域を同時に追跡することで頭部追跡が行える。図 4.24 は、提案手法を用いて顔だけを追跡した場合と、頭部（顔と髪）を追跡した場合の比較結果である。ここで用意した画像系列には 2 人の人物が写っており、同じ画像系列に対し追跡領域の異なる 2 つの方法での比較実験を行った。

図 4.24(a) は顔領域だけの追跡を行っているが、95 フレーム目には一人目の追跡が失敗し、125 フレーム目には二人目の追跡も継続できなくなっている。このように、顔だけの追跡では、顔（肌色）領域が追跡中に小さくなると追跡が失敗してしまうなど、顔の向きによって追跡が不安定になる場合がある。

一方、図 4.24(b) のように顔だけではなく、髪も同時に追跡した場合は、最終フレーム



図 4.23 複数対象の追跡

まで失敗することなく追跡を継続することができた．頭部の姿勢が変化してもどちらかの領域は必ずカメラに映るため，顔領域に加え髪領域を追跡することで，顔がカメラに対して後ろを向いた場合でも追跡を継続することができる．

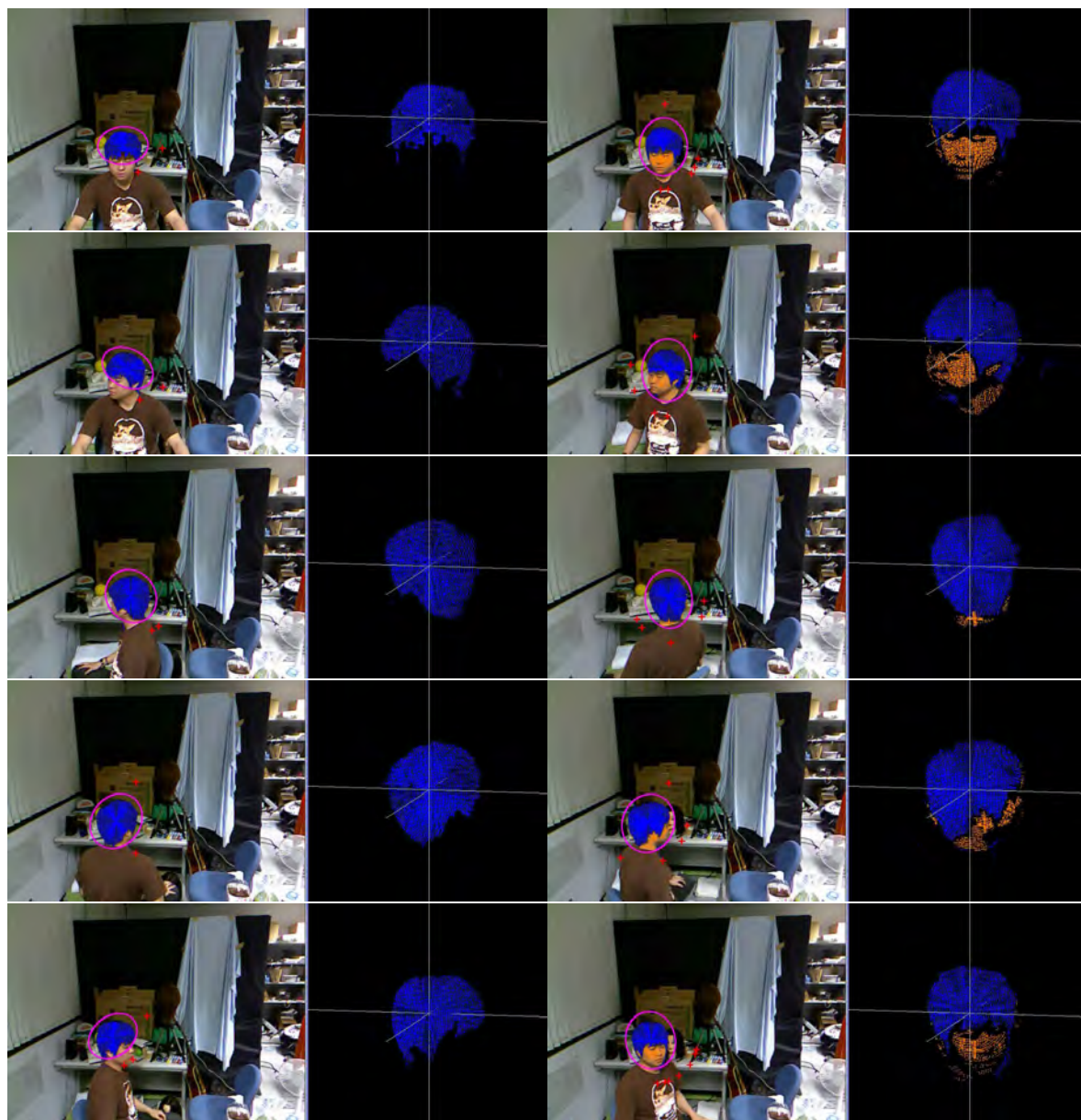
このように，動画像から顔や髪領域を抽出するために頭部領域の追跡を行う場合，本手法によって，頭部の姿勢変化に関わらず頑健な追跡を行うことが可能となる．

図 4.25 は，髪追跡，頭部追跡の結果を 3D 表示したものである．このように，髪領域を追跡することで頭部の姿勢に関わらず継続的な追跡が可能となり，背景から正しく分離された髪や顔領域の 3 次元形状をリアルタイムに得ることができる．





図 4.24 顔追跡と頭部 (顔と髪) 追跡の比較



(a) 髪追跡

(b) 頭部（顔と髪）追跡

図 4.25 髪，頭部（顔と髪）追跡結果の 3D 表示

### 4.5.5 提案システムを用いた応用実験

提案手法によって高速かつ安定に画像中から頭部領域を抽出することができるため、これらの取得できる情報を用いた様々な応用例が考えられる。以下、頭部の姿勢推定を行った例と運動視差を利用した立体的な表示方法について述べる。

#### 頭部の姿勢推定

ここでは、簡易的な処理によって頭部の姿勢推定を行った例を示す。本実験では、提案手法を用いて頭部追跡を行った際に得られる髪領域と顔領域を利用し、カメラから見えるこれら領域の変化によって頭部の姿勢推定を行っている。また、応用の一例を示す実験であり、以下の様な簡易的な閾値処理によってカメラに対する上下左右の4方向のみの推定を行っている。

それぞれの方向の判断基準は、次のようになっている。図 4.26 のように、髪と顔領域と判断された画素の重心を原点とする2次元座標軸をおき、それぞれの象限の髪領域の割合を  $hRate1 \sim hRate4$  とし、方向判定を以下のように行う。また、推定結果の例を図 4.27 に示す。図中の赤い矢印が推定結果を表している。

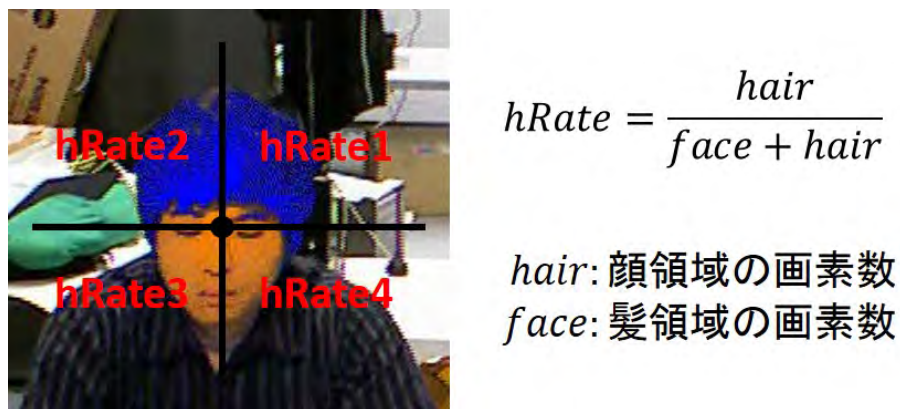


図 4.26 頭部姿勢推定の処理

右方向  $if(hRate3 > 0.2 \text{ and } hRate4 < 0.1)$   
左方向  $else if(hRate4 > 0.2 \text{ and } hRate3 < 0.1)$   
下方向  $if(hRate4 > 0.3 \text{ and } hRate3 > 0.3)$   
上方向  $else if(hRate1 < 0.7 \text{ and } hRate2 < 0.7)$



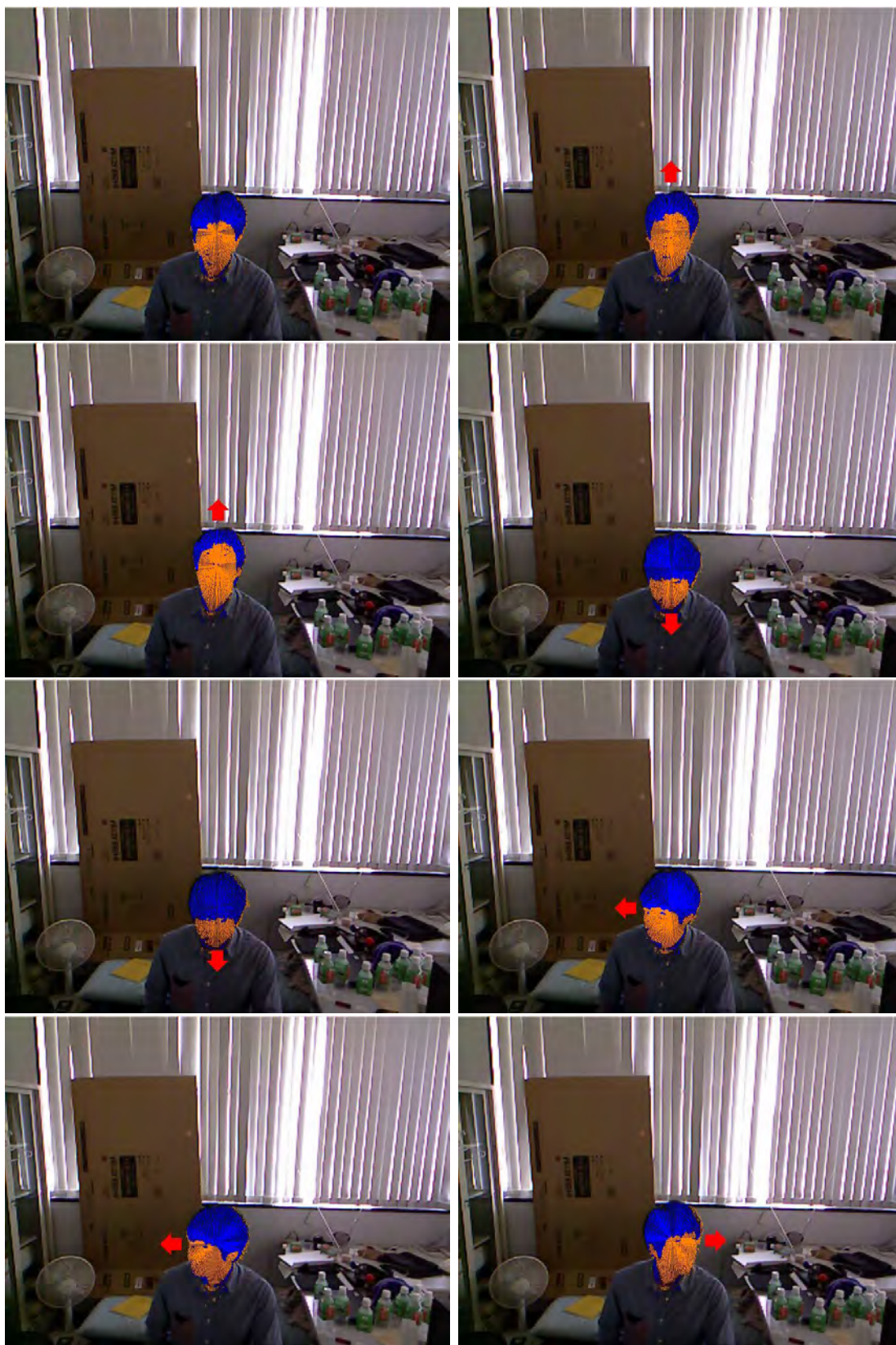
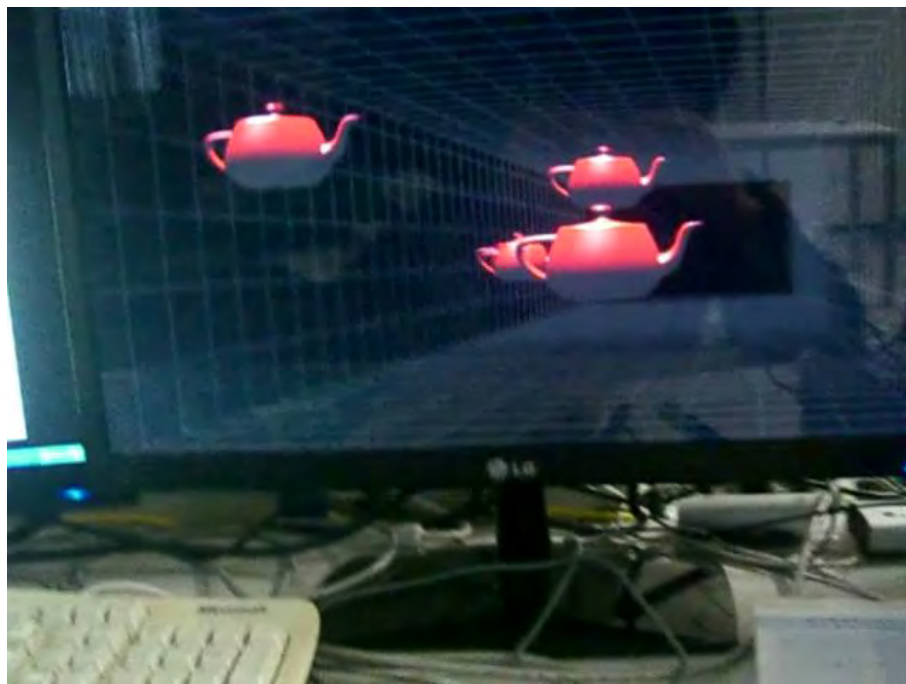


図 4.27 頭部の姿勢推定

### 運動視差による立体視

人が立体的な視覚を得る方法としては様々なものがあるが、特に重要なものとして視差立体視がある。また、視差立体視として、左右の目における両眼視差と視点移動による運動視差がある。平面なディスプレイにおいて立体視を提供する際、両眼視差を利用した場合では、左右の目に別々の画像を見せるために特殊なディスプレイを用いるか偏光メガネなどを装着する必要がある。一方、運動視差を利用した場合は、ユーザの視点移動に応じて適切な CG を生成することによって、通常のディスプレイにおいて立体視を実現することができる。

本実験では、提案手法によって取得できる頭部位置を利用した応用例として、特殊な装置を身に付ける必要のない運動視差を利用した立体視を行った。追跡で得られる頭部領域の重心を視点位置とし、視点に応じた透視投影変換によって適切な画像を生成する。追跡中に視点からディスプレイを撮影した様子を図 4.28 に示す。



右側から見た場合



左側から見た場合

図 4.28 運動視差による立体視

## 4.6 まとめ

本章では，Kinect を用いることで高速かつ高精度な距離情報を容易に得られることを利用し，K-means クラスタリングの特徴空間を 5 次元から 6 次元に拡張することによって，従来の K-means Tracker の類似色背景問題と画像内のターゲットサイズ問題を解決できる手法を提案した．また，距離情報を用いて高速に顔を検出して追跡のためのクラスタ中心を自動初期化する手法を提案し，自動的に開始できる髪・顔・頭部領域の安定な追跡システムを構築した．

本章で提案した追跡手法は，従来の K-means Tracker を奥行き情報を用いた拡張により安定化させることで，追跡が難しい髪領域を追跡できることを示した．実験において，様々な色・スタイルの髪や頭部姿勢の変化による顔や髪領域の変形に対し，安定した追跡が行えることを確認した．

また，この追跡手法は適切な初期クラスタ中心を設定することで，髪領域だけでなく任意のマルチ非剛体物体をより安定に追跡することが可能であるため，コンピュータビジョンにおいて基礎的な処理である一般的な対象追跡に役立つことが期待される．

## 第 5 章

# おわりに

### 5.1 本論文のまとめ

本論文では，奥行き情報を利用した顔検出の高速化と追跡の安定化について述べ，検出処理と追跡処理を統合することによって顔や髪領域の検出・追跡システムを構築した。

検出処理については，第 2 章にて，奥行き情報を用いることで顔の検出処理を高速化・高精度化する方法について述べた．本手法ではステレオカメラを用い，ステレオ画像を探索することで検出精度を向上させながら，ステレオ処理で得られる奥行き情報を利用することで単眼カメラでの探索よりも高速に顔検出を行うことができる．算出された画像内の各領域の奥行き情報に基づき，その領域に顔がある場合の画像上での顔の大きさを算出することによって，その領域で探索する顔のサイズを決定する．こうして探索する際の領域とスケールを限定することで識別回数を削減し，検出の高速化を行った．また，ステレオ処理のコストを削減するため，スパースサンプリングによって効率的にステレオ処理を行う方法を提案した．比較実験により，同じ Viola-Jones 法をベースとした全探索に比べて提案手法の方が，検出速度を 3～5 倍ほど向上させることができることを確認し，識別回数の削減にともなって誤検出を減らすことができるため，検出精度を向上させることができることも確認できた．RGB-D カメラなど，ステレオ処理を必要とせず高速に奥行き情報を取得できるカメラデバイスを用いれば，さらなる高速化が期待できる．また，本手法は，奥行きに応じて限定したスケールのみ探索を行っているため，顔写真など実際の大きさとは異なる顔を検出せず実物の顔のみを検出できる顔検出法である．さらに，本論文で提案した顔検出を高速化する手法は，顔だけでなく大きさが既知の対象であれば，その検出を高速化することが可能となる．

第 2 章で提案した顔検出処理と追跡手法を組み合わせることで，追跡を自動的に開始す



ることができる顔や髪を検出・追跡システムについて、第3章と第4章にて説明した。

第3章では、追跡手法として色弁別度追跡法を用い、顔の検出・追跡システムを構築する方法について述べた。この追跡手法と第2章で提案した顔検出を組み合わせたシステムにより、従来システムの色に基づく検出で問題となっていた誤検出を解決し、顔だけを正確に追跡できることを示した。提案システムでは、顔検出で得られた顔位置から自動的に追跡の初期化処理を行っている。追跡処理に必要なターゲットプロトタイプと背景プロトタイプを自動的に選択し、弁別度 LUT を構築しているため、人種や肌の色に関わらず追跡を開始することができるシステムとなっている。また、追跡中にも顔検出を行い、弁別度 LUT のアップデートを行うことで、照明や周辺環境の変化に対する頑健性を向上させた。さらに、顔検出の高速化手法で提案しているアイデアと同様に、追跡中の顔とカメラの距離から画像上での顔の大きさを推定することができるため、これを用いてターゲット領域を表す楕円に制約を与えることで、顔と類似色の背景が画像内に混入していても安定した追跡が行えることを示した。

第4章では、5次元の K-means Tracer を奥行き情報により拡張した6次元特徴空間における 6D K-means Tracker を用いて、テクスチャや色情報が乏しく追跡することが難しい髪領域の追跡が行えることを示した。K-means Tracker は、フレームごとに非ターゲットクラスタ中心を再配置しているため、ターゲットと類似色の背景混入にある程度頑健であったが、ターゲット領域に近接しすぎると追跡が不安定になってしまう問題があった。提案手法では、拡張された奥行き情報により、近似色であってもターゲットと背景を正しく分離することができ、安定した追跡が行えることを示した。また、特徴空間の拡張によって特徴同士の距離計算にかかるコストは増加するものの、特徴同士の分離性能が高まった提案手法では、背景モデルを表現する非ターゲットクラスタ中心の数を少なくすることができるため、処理速度を落とさずに安定性を高めることができる。提案した 6D K-means Tracker と顔検出を組み合わせ、顔や髪領域の検出・追跡システムを構築した。顔検出で得られた顔位置と距離画像を利用して自動的に追跡初期値のターゲットクラスタ中心と非ターゲットクラスタ中心を設定しているため、髪の色や形に関係なく追跡を行うことができる。また、髪領域の追跡を行うことで、頭部の姿勢変化に関わらず安定して追跡を継続することが可能となった。実験を通して、提案手法の有効性を確認し、本手法によって得られる情報を用いた利用例についても示した。本論文で提案した 6D K-means Tracker は、従来手法の処理速度を維持しつつ安定性を向上させているため、髪領域や顔領域だけでなく任意の非剛体物体をより安定して追跡することが可能となる。



## 5.2 今後の課題

### 複数人物の追跡

本論文で述べた検出・追跡システムでは、追跡処理がターゲットごとに独立しており、複数人物の追跡を容易に行うことができる。第4章で述べたシステムでは、追跡対象を常に画像中央に捉える Pan-Tilt Unit の制御を行わなければ、複数人物の追跡を行うことができる。追跡対象を1人に限定した場合は、検出処理と追跡処理が自動的に切り替わるため同時に2つの処理が行われることはなく、システムが追跡モードの時は非常に高速な処理ができる。しかし、複数の人物を追跡しようとした場合は、新たな人物が画像中に入ってくるのを検出するため、追跡処理時にも検出処理を行わなければならない、追跡処理よりも比較的処理コストが高い検出処理を常に行う必要がある。そのため、追跡対象が増えるにつれ、システム全体としての処理をビデオレートで行うことは難しくなっていく。

この問題を解決するためには、検出処理の領域を画像の端だけにする、追跡領域の近くは検出を行わないといったようにすることで検出処理を行う領域を削減するなど、検出処理のコストを抑えることが考えられる。また、K-means Tracker における各画素のクラスタリングを並列処理にするなど追跡処理を高速化することや、検出処理と追跡処理を並列化することでシステム全体を高速化することも考えられる。

### 追跡初期化時のターゲットモデル

第5章で述べた髪領域の検出・追跡システムでは、追跡の初期化時にターゲットモデルとして、髪領域であると思われる部分に自動的にターゲットクラスタ中心を配置している。この配置が適切な箇所に行われれば安定した追跡を開始することができるが、照明環境などにより生じるハイライト部分にターゲットクラスタ中心が配置されてしまうと、色情報がターゲットと大きく異なってしまうため、安定した追跡が行えないといった問題がある。

この問題の解決策としては、ターゲットクラスタ中心の数を増やすことが考えられる。複数のターゲットクラスタ中心で追跡を行うことで、適切に配置されなかったクラスタ中心は、そのクラスタに分類される画素が少なくなるため追跡への影響は少なくなり、ターゲットを適切に表現するクラスタ中心によって追跡を行うことができる。しかし、ターゲットクラスタ中心の数を増やせば、それだけ追跡処理のコストが増えることとなり、それらのバランスが重要となる。また、ターゲットクラスタ中心を1つの点から取得するのではなく、一定の領域から平均値を用いて取得するなど統計的な方法も考えられ、ター

ゲットクラスタ中心の配置法には議論の余地が残っている.

## 参考文献

- [1] Isao Hara, Alexander Zelinsky, Toshihiro Matsui, Hideki Asoh, Takio Kurita, Masaru Tanaka, and Kazuhiro Hotta. Communicative functions to support human robot cooperation. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, Vol. 2, pp. 683–688. IEEE, 1999.
- [2] Shinichi Satoh and Takeo Kanade. Name-it: Association of face and name in video. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 368–373. IEEE, 1997.
- [3] Osamu Hasegawa, Katsunobu Itou, Takio Kurita, Satoru Hayamizu, Kazuyo Tanaka, Kazuhiko Yamamoto, and Nobuyuki Otsu. Active agent oriented multimodal interface system. In *IJCAI*, pp. 82–87. Citeseer, 1995.
- [4] K Sato, K Chihara, et al. A robust face identification against lighting fluctuation for lock control. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 42–47. IEEE, 1998.
- [5] Syed A Rizvi, P Jonathon Phillips, and Hyeonjoon Moon. The feret verification testing protocol for face recognition algorithms. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 48–53. IEEE, 1998.
- [6] 赤松茂. コンピュータによる顔の認識の研究動向. 電子情報通信学会誌, Vol. 80, No. 3, pp. 257–266, 1997.
- [7] 岩井儀雄, 勞世, 山口修, 平山高嗣. 画像処理による顔検出と顔認識 (サーベイ (2)). 情報処理学会研究報告. CVIM,[コンピュータビジョンとイメージメディア], Vol. 2005, No. 38, pp. 343–368, 2005.
- [8] Rama Chellappa, Charles L Wilson, and Saad Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, Vol. 83, No. 5, pp.

705–741, 1995.

- [9] Erik Hjelmås and Boon Kee Low. Face detection: A survey. *Computer vision and image understanding*, Vol. 83, No. 3, pp. 236–274, 2001.
- [10] Kevin W Bowyer, Kyong Chang, and Patrick Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition. *Computer vision and image understanding*, Vol. 101, No. 1, pp. 1–15, 2006.
- [11] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 24, No. 1, pp. 34–58, 2002.
- [12] Ashok Samal and Prasana A Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern recognition*, Vol. 25, No. 1, pp. 65–77, 1992.
- [13] William A Barrett. A survey of face recognition algorithms and testing results. In *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, Vol. 1, pp. 301–305. IEEE, 1997.
- [14] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, Vol. 35, No. 4, pp. 399–458, 2003.
- [15] Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, and Fuyan Zhang. Face recognition from a single image per person: A survey. *Pattern recognition*, Vol. 39, No. 9, pp. 1725–1745, 2006.
- [16] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 24, No. 1, pp. 34–58, 2002.
- [17] Toshiyuki Sakai, Makoto Nagao, and Shinya Fujibayashi. Line extraction and pattern detection in a photograph. *Pattern recognition*, Vol. 1, No. 3, pp. 233–248, 1969.
- [18] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974.
- [19] Dario Maio and Davide Maltoni. Real-time face location on gray-scale static images. *Pattern Recognition*, Vol. 33, No. 9, pp. 1525–1539, 2000.
- [20] Venu Govindaraju, Sargur N Srihari, and David B Sher. A computational model for face location. In *Computer Vision, 1990. Proceedings, Third International*

- Conference on*, pp. 718–721. IEEE, 1990.
- [21] Guangzheng Yang and Thomas S Huang. Human face detection in a complex background. *Pattern recognition*, Vol. 27, No. 1, pp. 53–63, 1994.
- [22] Constantine Kotropoulos and Ioannis Pitas. Rule-based face detection in frontal views. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, Vol. 4, pp. 2537–2540. IEEE, 1997.
- [23] Jie Yang and Alex Waibel. A real-time face tracker. In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*, pp. 142–147. IEEE, 1996.
- [24] 呉海元, 陳謙, 谷内田正彦. ファジイパターン照合を用いた色彩画像からの顔検出システム. 電子情報通信学会論文誌 D, Vol. 80, No. 7, pp. 1774–1785, 1997.
- [25] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K Jain. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 24, No. 5, pp. 696–706, 2002.
- [26] Ying Dai and Yasuaki Nakano. Face-texture model based on sgld and its application in face detection in a color scene. *Pattern recognition*, Vol. 29, No. 6, pp. 1007–1017, 1996.
- [27] Thomas K Leung, Michael C Burl, and Pietro Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pp. 637–644. IEEE, 1995.
- [28] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 20, No. 1, pp. 23–38, 1998.
- [29] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, Vol. 1, pp. 746–751. IEEE, 2000.
- [30] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, pp. I–511. IEEE, 2001.
- [31] David E Rumelhart, James L McClelland, PDP Research Group, et al. *Parallel distributed processing*, Vol. 1. IEEE, 1988.
- [32] Kah-Kay Sung and Tomaso Poggio. Example-based learning for view-based hu-

- 
- man face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 20, No. 1, pp. 39–51, 1998.
- [33] Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: an application to face detection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 130–136. IEEE, 1997.
  - [34] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, Vol. 1, pp. 746–751. IEEE, 2000.
  - [35] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, Vol. 55, No. 1, pp. 119–139, 1997.
  - [36] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, Vol. 57, No. 2, pp. 137–154, 2004.
  - [37] 勞世竑, 山下隆義, 岡本卓也, 川出雅人. 高速全方向顔検出. MIRU2004, vol. II, pp. 271–276, 2004.
  - [38] B-Z Shaick and Leonid Yaroslavsky. Accelerating face detection by means of image segmentation. In *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on*, Vol. 1, pp. 411–416. IEEE, 2003.
  - [39] Junguk Cho, Shahnam Mirzaei, Jason Oberg, and Ryan Kastner. Fpga-based face detection system using haar classifiers. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pp. 103–112. ACM, 2009.
  - [40] Daniel Hefenbrock, Jason Oberg, Nhat Thanh, Ryan Kastner, and Scott B Baden. Accelerating viola-jones face detection to fpga-level using gpus. In *FCCM*, pp. 11–18, 2010.
  - [41] RI Hg, Petr Jasek, Clement Rofidal, Kamal Nasrollahi, Thomas B Moeslund, and Gabrielle Tranchet. An rgb-d database using microsoft’s kinect for windows for face detection. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pp. 42–46. IEEE, 2012.
  - [42] Tri Huynh, Rui Min, and Jean-Luc Dugelay. An efficient lbp-based descriptor for facial depth images applied to gender recognition using rgb-d face data. In



- Computer Vision-ACCV 2012 Workshops*, pp. 133–145. Springer, 2013.
- [43] Microsoft. Face tracking sdk. <http://msdn.microsoft.com/en-us/library/jj130970.aspx>.
- [44] Abhijit Jana. *Kinect for Windows SDK Programming Guide*. Packt Publishing Ltd, 2012.
- [45] Mingcai Zhou, Lin Liang, Jian Sun, and Yangsheng Wang. Aam based face tracking with temporal matching and face segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 701–708. IEEE, 2010.
- [46] Kazuya Ueki, Hiromitsu Komatsu, Satoshi Imaizumi, Kenichi Kaneko, Nobuhiro Sekine, Jiro Katto, and Tetsunori Kobayashi. A method of gender classification by integrating facial, hairstyle, and clothing images. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 4, pp. 446–449. IEEE, 2004.
- [47] Yaser Yacoob and Larry S Davis. Detection and analysis of hair. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 28, No. 7, pp. 1164–1169, 2006.
- [48] Jia-Kai Chou and Chuan-Kai Yang. Simulation of face/hairstyle swapping in photographs with skin texture synthesis. *Multimedia tools and applications*, Vol. 63, No. 3, pp. 729–756, 2013.
- [49] Kuang-chih Lee, Dragomir Anguelov, Baris Sumengen, and Salih Burak Gokturk. Markov random field models for hair and face segmentation. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pp. 1–6. IEEE, 2008.
- [50] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 23, No. 11, pp. 1222–1239, 2001.
- [51] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 26, No. 2, pp. 147–159, 2004.
- [52] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 467–475. Morgan Kauf-

mann Publishers Inc., 1999.

- [53] Pauline Julian, Christophe Dehais, Francois Lauze, Vincent Charvillat, Adrien Bartoli, and Ariel Choukroun. Automatic hair detection in the wild. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 4617–4620. IEEE, 2010.
- [54] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, Vol. 1, No. 4, pp. 321–331, 1988.
- [55] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, Vol. 61, No. 1, pp. 38–59, 1995.
- [56] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *Computer Vision–ECCV 2012*, pp. 746–760. Springer, 2012.
- [57] Xiaofeng Ren, Liefeng Bo, and Dieter Fox. Rgb-(d) scene labeling: Features and algorithms. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2759–2766. IEEE, 2012.
- [58] Azriel Rosenfeld and Avinash C Kak. *Digital picture processing*, Vol. 1. Elsevier, 2014.
- [59] Paul Fieguth and Demetri Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 21–27. IEEE, 1997.
- [60] JL Crowley and J Martin. Experimental comparison of correlation techniques. In *IAS-4, International Conference on Intelligent Autonomous Systems, Karlsruhe*, 1995.
- [61] AJ Fitch, Alexander Kadyrov, William J Christmas, and Josef Kittler. Orientation correlation. In *BMVC*, pp. 1–10, 2002.
- [62] Ulf M Cahn von Seelen and Ruzena Bajcsy. Adaptive correlation tracking of targets with changing scale. 1996.
- [63] Bohyung Han and Larry Davis. On-line density-based appearance modeling for object tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Vol. 2, pp. 1492–1499. IEEE, 2005.

- [64] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 26, No. 6, pp. 810–815, 2004.
- [65] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, Vol. 21, No. 1, pp. 32–40, 1975.
- [66] Dorin Comaniciu and Peter Meer. Mean shift analysis and applications. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, Vol. 2, pp. 1197–1203. IEEE, 1999.
- [67] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, Vol. 2, pp. 142–149. IEEE, 2000.
- [68] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 25, No. 5, pp. 564–577, 2003.
- [69] Robert T Collins. Mean-shift blob tracking through scale space. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, Vol. 2, pp. II–234. IEEE, 2003.
- [70] Tao Zhao and Ramakant Nevatia. Tracking multiple humans in crowded environment. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Vol. 2, pp. II–406. IEEE, 2004.
- [71] Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.
- [72] Michael Boyle. The effects of capture conditions on the camshift face tracker. *Alberta, Canada: Department of Computer Science, University of Calgary*, 2001.
- [73] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.
- [74] Gregory D Hager and Kentaro Toyama. X vision: A portable substrate for real-time vision applications. *Computer Vision and Image Understanding*, Vol. 69, No. 1, pp. 23–37, 1998.
- [75] 深尾隆則, 金出武雄. 2 段階特徴点追従アルゴリズム. 情報処理学会研究報告. CVIM,[コンピュータビジョンとイメージメディア], Vol. 2003, No. 109, pp. 103–

- 110, 2003.
- [76] Shian-Ru Ke, Hoang Le Uyen Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. A review on video-based human activity recognition. *Computers*, Vol. 2, No. 2, pp. 88–131, 2013.
  - [77] Hans P Moravec. Visual mapping by a robot rover. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 1*, pp. 598–600. Morgan Kaufmann Publishers Inc., 1979.
  - [78] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, Vol. 15, p. 50. Manchester, UK, 1988.
  - [79] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593–600. IEEE, 1994.
  - [80] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Vol. 60, No. 2, pp. 91–110, 2004.
  - [81] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, Vol. 81, pp. 674–679, 1981.
  - [82] Walter R Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.
  - [83] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, Vol. 29, No. 1, pp. 5–28, 1998.
  - [84] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. *Computer Vision—ECCV'96*, pp. 343–356. Springer, 1996.
  - [85] Michael Isard and Andrew Blake. A smoothing filter for condensation. *Computer Vision—ECCV'98*, pp. 767–781. Springer, 1998.
  - [86] Michael Isard and Andrew Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *Computer Vision—ECCV'98*, pp. 893–908. Springer, 1998.
  - [87] Prahlad Vadakkepat, Peter Lim, Liyanage C De Silva, Liu Jing, and Li Li Ling. Multimodal approach to human-face detection and tracking. *Industrial Electronics, IEEE Transactions on*, Vol. 55, No. 3, pp. 1385–1393, 2008.
  - [88] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *Computer Vision—ECCV' 98*, pp. 484–498. Springer, 1998.
  - [89] Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J

- Taylor. View-based active appearance models. *Image and vision computing*, Vol. 20, No. 9, pp. 657–664, 2002.
- [90] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pp. 232–237. IEEE, 1998.
- [91] Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 22, No. 4, pp. 322–336, 2000.
- [92] Sumit Basu, Irfan Essa, and Alex Pentland. Motion regularization for model-based head tracking. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, Vol. 3, pp. 611–616. IEEE, 1996.
- [93] Opencv. <http://opencv.org/>.
- [94] 飯塚健男, 和田俊和. 色弁別度を用いた実時間ステレオ対象検出・追跡. 画像の認識・理解シンポジウム (MIRU2006) インタラクティブセッション, pp. 1072–1077, 2006.
- [95] Hiroshi Oike, Toshikazu Wada, Takeo Iizuka, Haiyuan Wu, Takahiro Miyashita, and Norihiro Hagita. Detection and tracking using multi-color target models. In *Optics East 2007*, pp. 67640C–67640C. International Society for Optics and Photonics, 2007.
- [96] Toshikazu Wada. Visual object tracking using positive and negative examples. In *Robotics Research*, pp. 189–199. Springer, 2011.
- [97] 飯塚健男, 中村恭之, 和田俊和. ステレオカメラによる色ターゲットの 3 次元位置計測 (卒論セッション). 情報処理学会研究報告. CVIM,[コンピュータビジョンとイメージメディア], Vol. 2004, No. 40, pp. 65–70, 2004.
- [98] 飯塚健男. 弁別度を用いた実時間ステレオ対象検出・追跡. 和歌山大学大学院システム工学研究科修士論文, 2006.
- [99] VRL. ”らしさ”に基づく対象検出・追跡. [http://vrl.sys.wakayama-u.ac.jp/VRL/studyresult/study\\_result\\_1.html](http://vrl.sys.wakayama-u.ac.jp/VRL/studyresult/study_result_1.html).
- [100] 和田俊和. 最近傍識別器を用いた色ターゲット検出. 情報処理学会論文誌. コンピュータビジョンとイメージメディア, Vol. 44, No. 17, pp. 126–135, 2003.
- [101] 大池洋史, 呉海元, 華春生, 和田俊和. 高速追従型能動カメラシステム. システム制御情報学会論文誌, Vol. 20, No. 3, pp. 114–121, 2007.

- 
- [102] Hiroshi Oike, Haiyuan Wu, Chunsheng Hua, and Toshikazu Wada. Clear image capture-active cameras system for tracking a high-speed moving object. In *ICINCO-SPSMC*, pp. 94–102, 2007.
- [103] 大池洋史, 呉海元, 加藤丈和. 鮮明な画像撮影のための高速追従カメラシステム. 知能メカトロニクスワークショップ講演論文集, Vol. 9, pp. 79–84, 2004.
- [104] 和田俊和, 濱塚俊明, 加藤丈和. K-means トラッキング: 背景混入に対して頑健な対象追跡法. 画像の認識理解シンポジウム (MIRU2004), pp. 7–12, 2004.
- [105] HUA Chunsheng, WU Haiyuan, CHEN Qian, and WADA Toshikazu. Object tracking with target and background samples. *IEICE transactions on information and systems*, Vol. 90, No. 4, pp. 766–774, 2007.
- [106] Chunsheng Hua, Haiyuan Wu, Toshikazu Wada, and Qian Chen. K-means tracking with variable ellipse model. *IPSJ Digital Courier*, Vol. 1, pp. 508–517, 2005.
- [107] Chunsheng Hua, Haiyuan Wu, Qian Chen, and Toshikazu Wada. K-means tracker: A general algorithm for tracking people. *Journal of Multimedia*, Vol. 1, No. 4, pp. 46–53, 2006.
- [108] Hiroshi Oike, Haiyuan Wu, and Toshikazu Wada. Adaptive selection of non-target cluster centers for k-means tracker. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4. IEEE, 2008.
- [109] 大池洋史, 呉海元, 和田俊和. K-means tracker における適応的な非ターゲットクラスタ中心の配置法 (一般セッション 5). 情報処理学会研究報告. CVIM,[コンピュータビジョンとイメージメディア], Vol. 2008, No. 3, pp. 205–210, 2008.
- [110] Yiqiang Qi and Haiyuan Wu. A pixel-wise tracking algorithm using stereo camera. In *The 7th International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 198–201, 2010.
- [111] VRL. K-means tracker. [http://vrl.sys.wakayama-u.ac.jp/VRL/studyresult/study\\_result\\_3.html](http://vrl.sys.wakayama-u.ac.jp/VRL/studyresult/study_result_3.html).
- [112] 威意強. 画素のクラスタリング法と物体追跡への応用. 和歌山大学大学院システム工学研究科修士論文, 2012.
- [113] 大池洋史. 能動カメラの高速追従制御による移動物体の鮮明な画像撮影方法に関する研究. 和歌山大学大学院システム工学研究科博士論文, 2009.
- [114] 華春生. K-means clustering based pixel-wise object tracking. 和歌山大学大学院システム工学研究科博士論文, 2007.



- 
- [115] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pp. 100–108, 1979.
- [116] Microsoft. Kinect. <http://www.xbox.com/ja-JP/kinect>.

# 研究業績

本論文の一部は、下記論文の内容を含む。

## 学術論文

- 1     鈴木一正, 呉海元, “Kinect を用いた髪 of 自動検出と追跡システム, システム制御情報学会論文誌”, Vol.26, No.9, pp.323-329, 2013 .
- 2     Kazumasa Suzuki, Haiyuan Wu and Qian Chen, “Fast Hair and Head Tracking System Using 3D Image Sensor”, Journal of Communication and Computer, pp.212-223, 2013.

## 国際会議（査読あり）

- 3     Kazumasa Suzuki, Hiroshi Oike, Haiyuan Wu, Toshikazu Wada, “Video-rate face detection and tracking using active stereo-camera”, The 3rd Pacific-Rim Symposium on Image and Video Technology (PSIVT), pp. 275-279, 2009.
- 4     Haiyuan Wu, Kazumasa Suzuki, Toshikazu Wada, and Qian Chen, “Accelerating Face Detection by Using Depth Information”, The 3rd Pacific-Rim Symposium on Image and Video Technology (PSIVT), pp. 657-667, 2009.（被引用回数：15 回）
- 5     Kazumasa Suzuki, Haiyuan Wu, Toshikazu Wada and Qian Chen, “Fast Face Detection and Tracking using Stereo-camera and Adaptive Color Model”, The 6th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp.275-279, 2009.
- 6     Kazumasa Suzuki, Qian Chen, Haiyuan Wu and Toshikazu Wada, “Detecting and Tracking Faces with Stereo-camera and Adaptive Color Model”, The

- 12th International Conference on Computer Vision (ICCV), Demo, 2009.
- 7 Yiqiang Qi, **Kazumasa Suzuki**, Haiyuan Wu, Qian Chen, “EK-means Tracker: A Pixel-wise Tracking Algorithm using Kinect”, The third Chinese Conference on Intelligent Visual Surveillance (IVS), pp.77-80, 2011.
- 8 **Kazumasa Suzuki**, Haiyuan Wu, Qian Chen, “Tracking Hair in Video-rate with Kinect”, The 11th Asian Conference on Computer Vision (ACCV), Demo, pp.1-2, 2012.
- 9 **Kazumasa Suzuki**, Haiyuan Wu and Qian Chen, “Video-rate Hair Tracking System Using Kinect”, IAPR The Fourth Computational Color Imaging Workshop (CCIW), pp.212-224, 2013.

## 国内会議（査読ありポスター，デモ発表）

- 10 鈴木一正, 呉海元, 和田俊和, “ステレオカメラを用いた顔検出の高速化”, 画像の認識・理解シンポジウム (MIRU2008) インタラクティブセッション, pp.1462-1467, 2008-7.
- 11 大池洋史, 鈴木一正, 呉海元, 和田俊和, “正面顔をキーとした能動ステレオ顔追跡システム”, 画像の認識・理解シンポジウム (MIRU2008) デモセッション, pp.1692-1693, 2008-7.
- 12 鈴木一正, 戚意強, 呉海元, “拡張 K-means Tracker を用いた顔・髪の毛の追跡”, 画像の認識・理解シンポジウム (MIRU2011) デモセッション, pp.1699-1700, 2011-7.

## 国内発表（査読なし）

- 13 鈴木一正, 呉海元, 和田俊和, “ステレオカメラを用いた顔検出の高速化”, 情報処理学会研究報告, 情処研報, CVIM, pp.107-112, 2008-7.
- 14 鈴木一正, 戚意強, 呉海元, “Kinect を用いた顔・髪の毛の追跡”, 情報処理学会研究報告, 情処研報, Vol.2012-CVIM-180, No.58, pp.317-320, 大阪, 2012-1.
- 15 鈴木一正, 粉川豊, 津田嶺雪, 呉海元, “循環器 OCT 画像の解析と 3 次元再構築”, 情報処理学会, 第 77 回全国大会, 1S-02, 京都, 2015-3.
- 16 李鵬, 鈴木一正, 呉海元, 陳謙, “色弁別度を特徴量とした個人識別”, 情報処理学会, 第 77 回全国大会, 1S-07, 京都, 2015-3.

## 関連ページ

Video-rate Hair Tracking System Using Kinect :

[http://www.wakayama-u.ac.jp/~wuhhy/wu3\\_move.html](http://www.wakayama-u.ac.jp/~wuhhy/wu3_move.html)

## 付録 A

# Viola-Jones 法による顔検出

顔検出の手法は数多く提案されているが，Viola と Jones が提案している AdaBoost 学習法とカスケード構造を組み合わせた手法 [30, 36] は，実行速度の速さと検出率の高さから世間で大変注目を浴びており，顔検出の研究と応用において広く利用されている．ここでは，Viola-Jones 法による顔検出について概説する．

Viola らの手法には次のような 3 つの特徴がある．

1. 積分画像を使い領域の大きさによらず一定の計算量でその領域の平均輝度値を高速に計算できる Haar-like 特徴量を用いている
2. 1 つの特徴量で 1 つの弱識別器を構成し，AdaBoost を使って，有効な識別器を選び出し，その線形結合で全体として高性能な強識別器を構成する
3. 複数の強識別器を直列につなぎ，カスケード型の識別器を構成する

## A.1 Haar-like 特徴量

Haar-like 特徴は，局所的な明暗差を表す特徴量である．図 A.1 は Haar-like 特徴のパターンを表しており，黒色領域の輝度値の和から白色領域の輝度値の和を引いた非常にシンプルな値で表現される．この 1 つのパターンで得られる値で 1 つの弱識別器を構築し，これらパターンの位置や大きさを変えた数十万種類の特徴を用意し，Adaboost によって識別に重要な特徴を選択する．

図 A.2 は，AdaBoost で選択された 1 番目と 2 番目の特徴を示している．1 番目の特徴では両目と頬との明暗差，2 番目の特徴では両目と鼻筋との明暗差を見ており，これら特徴が顔か否かの判定において重要な特徴となっていることがわかる．このように，

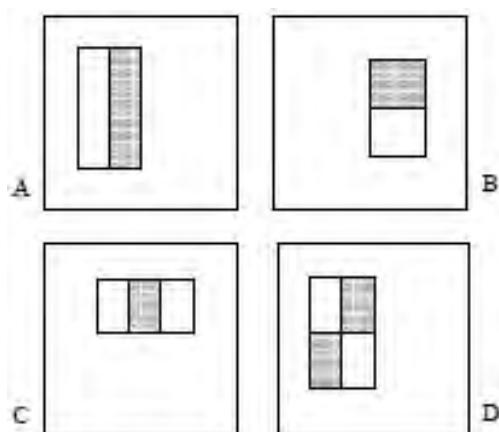


図 A.1 Haar-like 特徴のパターン [36]

Adaboost によって選択された複数の特徴を線形結合することで強識別器が構築される。

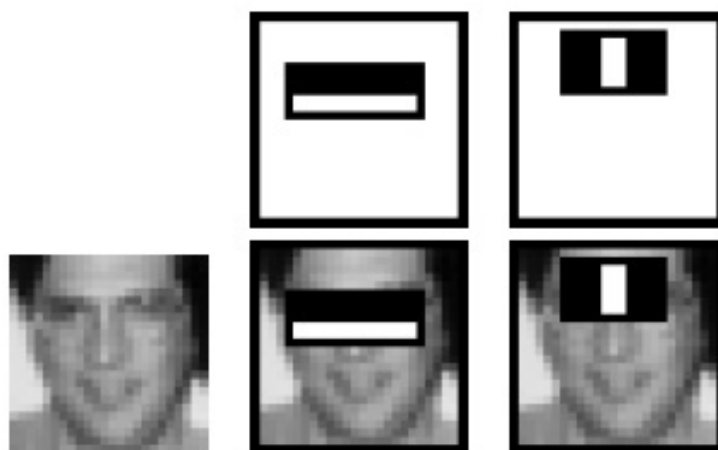


図 A.2 AdaBoost で選択された特徴 [36]

## A.2 積分画像

矩形領域の平均輝度値の差で表される Haar-like 特徴は，図 A.3 に示すような積分画像 (Integral Image) を用いることで高速に計算することができる．式 (A.1) のように，積分画像を  $ii(x, y)$ ，原画像を  $i(x, y)$  とすると，積分画像の各画素値には，画像左上端から



画素位置までの矩形領域における画素値の総和が入っている。

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (\text{A.1})$$

また、次の 2 つの式を用いることで、一回の走査で積分画像を計算することができる。

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (\text{A.2})$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (\text{A.3})$$

$s(x, y)$  は、各画素位置から画像上端までの列の総和を表しており、 $s(x, -1) = 0$ ,  $ii(-1, y) = 0$  である。

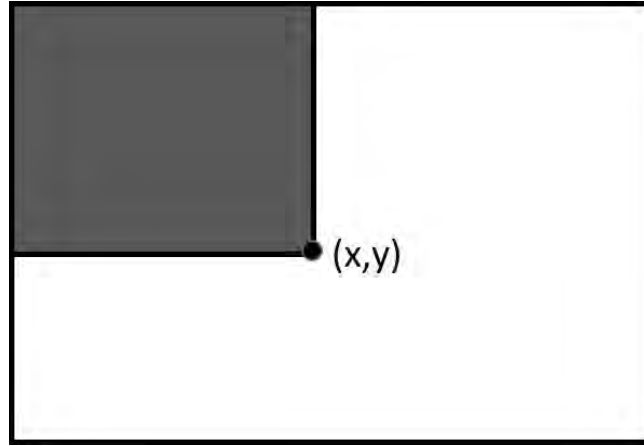


図 A.3 積分画像における画素値

このような積分画像を用いることで、画像中の任意の矩形領域における画素値の総和、平均、標準偏差などを定数時間で算出することができる。例えば、図 A.4 の矩形領域  $D$  の総和は 4 つの値の加減算で求めることができる。積分画像における 1 の値は領域  $A$  の総和を表しており、2 は  $A + B$ 、3 は  $A + C$ 、4 は  $A + B + C + D$  であり、領域  $D$  の総和は  $4 + 1 - (2 + 3)$  として計算できる。

### A.3 カスケード構造

この手法では、高速に計算できる特徴量に加えて、学習データのバリエーションを変えて構築した強識別器を一行に並べたカスケード構造によって検出の高速化を行っている。図 A.5 に Viola らのカスケード型識別器を示す。カスケード型の識別器は、後ろの層にいくほど計算量が多いが識別能力が高くなるように強識別器が連結された構造になっ

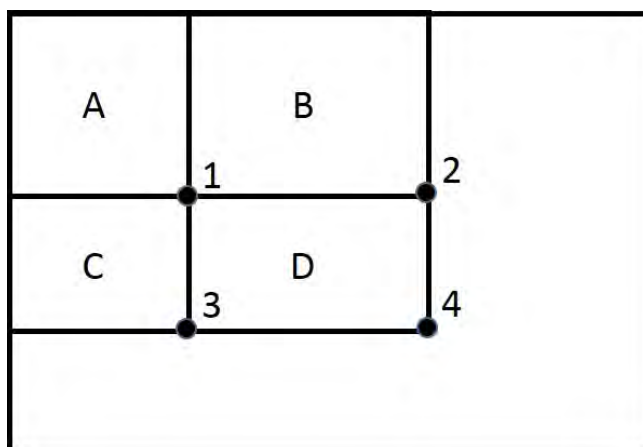


図 A.4 積分画像を用いた矩形領域の総和計算

ており，まず識別能力は低いが計算量の小さい前の層で明らかに顔ではないものを Early Reject する．前層で顔と判断されたものは以降のより高い識別能力を持った複雑な識別器を使って顔かどうかの判別を行う．全ての強識別器において顔と判断されたものが，最終的に顔領域として識別結果が出力される．

一般的な画像には顔ではない背景部分が多いため，このように計算量の小さいカスケード構造の前半で画像の大半を占める背景部分の判別を終えることで，全体の処理を高速化することができる．

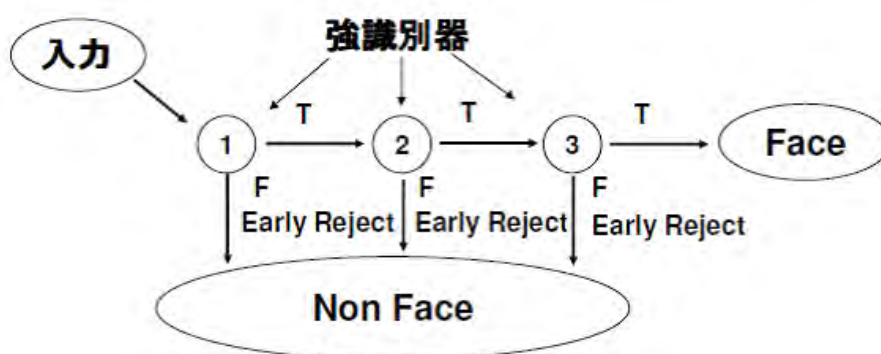


図 A.5 カスケード型識別器

## A.4 まとめ

ここでは、顔検出において主流なアルゴリズムとなっている Viola-Jones 法について説明した。積分画像によって高速に計算できるシンプルな Haar-like 特徴量と、大半の識別を構造が単純な計算量の少ない前半の層で行えるカスケード構造により、この手法は他の顔検出手法に比べ高速に顔を検出することができる。そのため、このアルゴリズムをベースとした顔検出器が現在主流となっており、公開されている顔検出ライブラリや身の回りの家電製品、セキュリティ分野に至るまで広く利用されている。

## 謝辞

本研究の過程において，指導教官として長きに渡る懇切なる御指導，御鞭撻を賜りました和歌山大学システム工学部呉海元教授，和田俊和教授に心より深く感謝致します．また，国内，国際会議での学会発表などの貴重な経験をさせて頂いたこと，ならびに私生活の面での様々のご配慮を頂いたことを重ねてお礼申し上げます．

本研究を進めるにあたり，研究に関わる内容のみならず，プログラミング等その他技術的な面でも多くの御指導，御助言を賜りました陳謙准教授に心から深く感謝致します．

本論文を作成するにあたり，学位論文審査委員として懇切なる御指導を賜りました和歌山大学システム工学部入野俊夫教授に心から深く感謝いたします．

平素の研究活動にあたり，種々のご協力を頂きました Vision and Robotics Laboratory (VRL) の修了生・卒業生・在学生・事務補佐員・その他関係者の方々に心よりお礼申し上げます．

最後に，学部から，博士前期課程を経て，博士後期課程修了に至るまで，学生生活を精神的，経済的に支えてくれた著者の家族に感謝の意を表します．