

論文 / 著書情報
Article / Book Information

題目(和文)	複数視点の時系列奥行き情報の統合による隠蔽に頑健な人物追跡と領域分割
Title(English)	
著者(和文)	福司謙一郎
Author(English)	Kenichiro Fukushi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9695号, 授与年月日:2014年12月31日, 学位の種別:課程博士, 審査員:熊澤 逸夫,長橋 宏,佐藤 誠,山口 雅浩,小尾 高史,長尾 智晴
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第9695号, Conferred date:2014/12/31, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis



博士学位論文

複数視点の時系列奥行き情報の統合による 隠蔽に頑健な人物追跡と領域分割

指導教官 熊澤 逸夫 教授

東京工業大学
大学院総合理工学研究科
物理情報システム専攻
福司 謙一郎

平成26年12月 提出

目次

要旨	1
第 1 章 序論	5
1.1 研究背景	5
1.2 本研究の目的	6
1.3 本研究の成果	6
1.4 本論文の構成	7
第 2 章 関連研究	9
2.1 人物追跡	9
2.1.1 複数視点を利用する追跡手法	9
2.1.2 奥行き情報を利用する追跡手法	10
2.1.3 マルチフレーム最適化による追跡手法	13
2.1.4 Data Association	16
2.2 領域分割	16
2.2.1 ビデオセグメンテーション	17
2.2.2 ジェスチャ認識への応用	18
2.3 まとめ	19
第 3 章 システム構成と提案手法の概要	21
3.1 システム構成	21
3.1.1 ステレオカメラネットワーク	21
3.1.2 追跡・領域分割プログラム	23
3.2 提案手法の概要	23
3.2.1 追跡と平面図上での領域分割	25
3.2.2 平面図上の領域からカメラ視点画像への対応付け	25
第 4 章 反復法によらない軌跡の最適化と各視点画像への逆投影	27
4.1 Geometry Map, Occupancy Map および Silhouette Map の作成	27
4.2 軌跡の作成とクラスタリング	31
4.2.1 手順 1: Occupancy Map のクリップ作成	31
4.2.2 手順 2: 軌跡作成	32
4.2.3 手順 3: 形状の類似性に基づく軌跡のクラスタリング	33

4.3	反復法によらない軌跡の最適化	36
4.3.1	誤ったクラスタの除去	36
4.3.2	途切れた軌跡の修復	39
4.4	隣接クリップ間での軌跡の ID マッチング	41
4.5	カメラ視点画像上での領域分割	41
第 5 章	計算機実験	45
5.1	実験に用いるデータセット	45
5.2	実験結果	45
5.3	平面図上での人物追跡のエラー評価	62
5.3.1	追跡エラーの分類	62
5.3.2	評価結果	62
5.3.3	クリップ長 T_B に対する追跡エラーの変化	63
5.4	カメラ視点画像上での領域分割エラー評価	64
5.4.1	評価結果	65
5.5	「干渉しないクラスタから成るパターンの総数」および「途切れた軌跡の 接続方法の総数」の評価	66
5.5.1	干渉しないクラスタから成るパターン	66
5.5.2	途切れた軌跡の接続方法	66
5.6	計算時間と通信量評価	68
5.7	まとめ	69
第 6 章	ジェスチャ認識への応用	71
6.1	概要	71
6.2	手法	71
6.3	評価	71
6.4	まとめ	79
第 7 章	結論	81
7.1	総括と成果	81
7.2	今後の課題	83
7.2.1	精度向上	83
7.2.2	システム実装の改善	84
7.2.3	拡張性	84
	本研究に関する論文	85
	謝辞	87

付録A カメラキャリブレーション	89
A.1 カメラキャリブレーションに用いるパターン	89
A.2 推定された各カメラの回転行列および並進行列	90
参考文献	91

目 次

2.1	2 視点を利用する人物追跡システム (Krumm ら [28] から引用). 図中では奥行き情報と色情報を使って検出した 2 人の人物を, 平面図に赤色と青色の円で表して, プロジェクタで表示している.	10
2.2	最大 16 台のカメラを利用する人物追跡システム (Mittal ら [30] から引用). 単独視点画像上で前景の領域分割のみを行い, 全ての視点の結果を統合して追跡を行う.	10
2.3	天井に設置された 5 台のカメラを利用する人物追跡システム (Salinas ら [36] から引用). 単独視点画像上でパーティクルフィルタを使って特徴抽出した後, 全ての視点の情報を統合して追跡が行われる手法を提案し, 図のように隠蔽の頻発する混雑環境下でも追跡を可能にした.	11
2.4	8 視点を利用する人物追跡システム (Zabulis ら [48] から引用). 視体積交差法により平面図上での人物位置を推定し, Blob トラッカーで追跡する. Zabulis らは博物館に提案システムを導入して, インタラクティブな情報提示デバイスに応用している.	12
2.5	Depth Camera の製品例. Bumblebee (PointGrey 社, 左) と Carmine (PrimeSense 社, 右)	12
2.6	平面図表現 (Plan-view map) の作成過程 (上) と実画像への適用例 (下). (a) カメラ視点画像 (b) 奥行き情報 (c) 前景画像 (d) Occupancy Map (e) Plan-view Height Map (f) Plan-view Color Map (Harville ら [22] より引用)	13
2.7	奥行き情報を基にして得た Occupancy Map に, 色情報を併用して追跡を行う手法 (Salinas ら [33] より引用). 図はカメラ視点画像と対応する平面図を表しており, 各人物の ID を表す数字と軌跡が描画されている.	14
2.8	3 台のステレオカメラから得た Occupancy Map を時系列で並べて (左), 最も当てはまる軌跡 (右) を推定する手法 (Darrell ら [12] より引用).	15
2.9	原口ら [52] は提案手法に, 色情報を追加利用する手法を提案した. 人物同士が近接して歩いている (左), 混同せずに正しく追跡が可能である (右).	15
2.10	テクスチャの類似する領域を統合し, 時間軸での対応付けも行う領域分割手法 (Grundmann ら [20] より引用). 異なる時刻でも同一の領域には同一の ID が割当てられている.	17

2.11	初期フレームで与えた領域を追跡対象物体の変形に追従させる領域分割手法 (Tsai ら [44] より引用)	18
2.12	人物のシルエットを時系列で並べて得られる 3 次元形状を基にしてジェスチャ認識を行う手法 (Blank ら [5] より引用)	19
2.13	予め登録しておいた動き (左) を未知の動画像中 (右) から検出する手法 (Ke ら [26] より引用)	20
3.1	システム構成	21
3.2	実験室の概観	21
3.3	カメラユニット	22
3.4	天井に取り付けられたステレオカメラ	22
3.5	提案手法の概要. (ii)(iii)(iv) 軌跡の作成とクラスタリング (4.2 節参照) (v) 軌跡の最適化 (4.3 節参照) (vi) カメラ視点画像上での領域分割 (4.5 節参照)	24
4.1	Occupancy Map (OM) は前景奥行き情報を床平面に平行な平面 S_0 に投影して作成したヒストグラムである. Geometry Map (GM) はカメラ視点画像 I 上の位置 \mathbf{u} とグリッド上に分割された床平面上の位置の対応を表す.	28
4.2	(1 列目) カメラ視点画像 I と背景画像 \bar{I} (2 列目) 奥行き画像 DM と背景奥行き画像 \overline{DM} (3 列目) 前景奥行き画像と対応する Occupancy Map . . .	29
4.3	軌跡の作成とクラスタリング処理の概要. (i) 2 人の人物がすれ違うシーンを示している. (ii) 時間長 T_B の Occupancy Map のクリップ (iii) Occupancy のある領域を通るように多数の軌跡が作成される (詳細は図 4.4 参照). (iv) 軌跡の形状に基づいて類似した軌跡をクラスタリングする (詳細は Algorithm 2 参照).	30
4.4	(i) 図 4.3(ii) の Occupancy Map (見易さのため Y 軸を省略) (ii) フローベクトルの作成過程. 赤の×印で示すフローベクトルは Occupancy の存在しない領域を通るため除外される. (iii) 軌跡の始点は時間幅 Δt 毎に幅 Δl の格子状に配置される. (iv) 各始点を Algorithm 1 に示す手順で延長し, 軌跡を作成する.	31
4.5	異なる視点で独立に作成された OM の例. 画像の撮影タイミングは非同期であり, 各 OM に対する相対撮影時間を付記した. カメラは水色の三角形の方向から撮影している. 時刻 0 における各人物の位置を円で示す. また, 十字の線は各人物の時刻 0 における初期位置を表す. 人物 (a) と (b) は矢印の方向に歩行しており, その Occupancy が時間と共に移動する様子が確認できる.	32
4.6	2 人の人物のデータセット D_2 の軌跡作成およびクラスタリング過程 . . .	35
4.7	3 人の人物のデータセット D_3 の軌跡作成およびクラスタリング過程 . . .	37

4.8	干渉しないクラスタから成るパターン（上）と干渉するクラスタから成るパターン（下）の例. 右図は時刻 t でのクラスタの位置を, Occupancy Map 上に重ね合わせたもの.	38
4.9	3人の人物のデータセット D_3 (図 4.7) における干渉しないクラスタから成るパターン. この例ではパターン 1 に含まれる軌跡が, Occupancy の得られている領域を最も占有するため, 最適解として選ばれる.	39
4.10	5人の人物のデータセット D_5 の最適化過程. (ii) 内のクラスタは, 見やすさのためクラスタに含まれる軌跡の時刻毎の平均位置を描画した.	40
4.11	カメラ視点画像 I_i^t 上での領域分割例	41
4.12	図 4.6 の時刻 t_2 における追跡および領域分割結果	43
4.13	5人の人物のデータセット D_5 で平面図上の領域分割を行った例（左）と, 比較のために固定サイズの円形ウィンドウを当てはめて描画した例（右）. 円形ウィンドウのサイズは Occupancy を含む必要最小限の大きさに設定したが, 人物が近接している場合はウィンドウが重複する. なお, 右下の人物の Occupancy Map が得られていないのは, 当該 Occupancy Map が図に示すカメラ視点から作成されたもので, 隠蔽による為である.	44
5.1	データセット D_1 , 時刻 t_0	48
5.2	データセット D_2 , 時刻 t_0	49
5.3	データセット D_3 , 時刻 t_0	50
5.4	データセット D_3 , 時刻 t_1	51
5.5	データセット D_3 , 時刻 t_2	52
5.6	データセット D_4 , 時刻 t_0	53
5.7	データセット D_4 , 時刻 t_1	54
5.8	データセット D_4 , 時刻 t_2	55
5.9	データセット D_5 , 時刻 t_0	56
5.10	データセット D_5 , 時刻 t_1	57
5.11	データセット D_5 , 時刻 t_2	58
5.12	データセット D_6 , 時刻 t_0	59
5.13	データセット D_6 , 時刻 t_1	60
5.14	データセット D_6 , 時刻 t_2	61
5.15	追跡クリップの時間長 T_B に対する最大追跡可能人数	64
5.16	生成されたクラスタの総数	67
5.17	干渉しないクラスタから成るパターンの総数 N_p	67
5.18	途切れた軌跡の接続方法の総数 N_r (データセット D_6 のみ縦軸のスケールが異なる事に注意)	68
5.19	カメラユニットでの計算時間とネットワーク通信量	70
5.20	人物追跡処理に要した計算時間	70

6.1	データセット D_7	72
6.2	データセット D_7 での挙手検知結果. 赤い領域は p_R が閾値 0.91 を超え, 挙手と判定された時刻を表す.	72
6.3	データセット $D_7 : t_0 = 0$ sec におけるシルエット作成結果	73
6.4	データセット $D_7 : t_1 = 4$ sec におけるシルエット作成結果	74
6.5	データセット $D_7 : t_2 = 19$ sec におけるシルエット作成結果	75
6.6	データセット $D_7 : t_3 = 33$ sec におけるシルエット作成結果	76
6.7	データセット $D_7 : t_4 = 54$ sec におけるシルエット作成結果	77
6.8	データセット $D_7 : t_5 = 69$ sec におけるシルエット作成結果	78
A.1	カメラキャリブレーションに用いるパターン	89

表 目 次

5.1	各データセットのタイムライン	46
5.2	平面図上での人物追跡エラー ($T_B = 10$ sec)	63
5.3	平面図上での人物追跡エラー ($T_B = 7$ sec)	63
5.4	平面図上での人物追跡エラー ($T_B = 3$ sec)	63
5.5	カメラ視点画像上での領域分割エラー	65
5.6	干渉しないクラスタから成るパターンの総数 N_p の時間平均 $\overline{N_p}$	69
5.7	途切れた軌跡の接続方法の総数 N_r の時間平均 $\overline{N_r}$	69

要旨

スマートスペースや Natural User Interface が急速に私達の生活に身近なものになりつつある中、人物の位置やジェスチャを取得する技術の重要性がますます高まっている。ジェスチャ認識、指紋認証、顔認証のように、センサに比較的近い限られた空間を対象とした認識技術は既に製品に組み込まれるほど普及している。また、建物内や交差点のような、より広い空間内を自由に人物が動く条件での認識技術も、多くの研究者が意欲的に取り組んでおり進歩がめざましい。このような条件下で人物追跡を困難にする主要因の一つとして、隠蔽が知られている。これは画像上で、ある物体が別の物体の背後に隠れ、一部あるいは全体が見えなくなってしまう現象である。カメラ視点画像上では奥行き情報が失われるために、画像上で重なっている領域は容易には分割できない。

近年の研究では、実時間で奥行き情報を計算できる Depth Camera が普及してきており、隠蔽を解消する手段として積極的に利用されている。通常のカメラでは失われてしまう奥行き情報は隠蔽に対して本質的な手掛かりとなる上、計測値が人物の見かけや照明変化の影響を受けにくく、実用上有用だからである。人物が鉛直方向に移動しないという仮定の下に奥行き情報を平面図で簡略化した表現は、人物追跡に利用しやすい特徴量として知られており、多くの研究で採用されている。しかし、姿勢・ジェスチャを知るのに必要な人物領域の形状は平面図表現ではほとんど失われてしまう。そのため既存研究では、隠蔽を扱いやすい、複数視点の情報を統合しやすい、といった平面図表現の特徴を活かして、頑健な人物動線検出の研究が発展している一方で、ジェスチャ認識のように人物形状を必要とする応用には適用されてこなかった。

本論文ではまず、奥行き情報を平面図に投影して得られる Occupancy Map を複数の視点で作成・統合し、複数のフレームに渡って最適化する事で、各視点の画像では領域分割を行わないまま、隠蔽に頑健な人物追跡を行う手法を提案する。次にこの人物追跡の結果をカメラ視点画像に投影することで領域分割を行う手法を提案する。一般に領域分割は物体追跡の前処理として行われるが、提案手法はその逆で、人物追跡の結果をカメラ視点画像に投影して領域分割を行う。従来手法では隠蔽によって分割が出来なかった領域でも正しく分割できる利点がある。

評価実験においては、異なる混雑度のデータセットに提案手法を適用して、隠蔽への頑健性を確認するとともに、人物追跡および領域分割のエラーを定量的に評価して手法の限界を検証する。また、従来手法では複数のフレームに渡る最適化のために動的計画法や反復法等が用いられてきたが、計算時間が大きくなる問題があった。提案手法ではヒューリスティクスなアプローチでの最適化手法を考案し、効率的な計算が可能である

事を示す。さらに、本論文で構築したシステムはカメラのネットワーク化において、画像そのものは伝送せずに、処理に必要な情報のみをネットワークに送信する。通信量の評価によって、一般的なネットワーク機器でもシステムを構築出来る事も示した。

最後に、提案手法の応用としてジェスチャ認識を例にとり、混雑環境下で挙手を検知するプロトタイプシステムを構築した。提案手法は隠蔽が発生していても全ての視点で正しく領域分割が出来るため、それらの情報を統合することで各視点での隠蔽の影響を互いに補う事が出来る。本システムでは、全ての視点の領域を使って特定の人物のシルエットを作成する。実際に混雑環境下で撮影したデータセットを用いて実験を行い、安定してジェスチャ認識が可能である結果を得、これにより提案手法の実用性も確認した。

Abstract

In this thesis, we present a computer vision-based human tracking and video segmentation system with multiple stereo cameras. Our method has two main features, as follows. First, we propose an occlusion-robust human tracking method that utilizes multi-view depth imagery acquired by the stereo cameras. Next, we propose a new segmentation technique, which can use the human tracking results to accurately extract human regions from video sequences that contain highly cluttered scenes.

Many of the methods that are used widely for human tracking, such as KLT-tracker, update the trackers “frame-to-frame,” where the features extracted from one frame are utilized to update the current state. By contrast, we propose a novel optimization technique for the “multi-frame” approach that computes the resultant trajectories directly from the video sequences. Thus, our method achieves a high level of robustness against severe occlusion, which is known to be a challenging problem in computer vision. We developed a heuristic optimization technique to estimate human trajectories, instead of using dynamic programming or an iterative approach, which means that our method is sufficiently computationally efficient to operate in realtime.

In our method, video segmentation is achieved by projecting the tracking results, which are represented as planar regions that are fitted to human shapes observed in the vertical direction, onto camera-view images. In general, video segmentation is performed before tracking in order to detect human regions. However, occlusion makes segmentation considerably difficult, where it may lead to the unfavorable fragmentation of a single human region and the confusion of multiple human regions, which both affect conventional methods adversely. Our strategy overcomes these issues successfully.

In our experimental evaluation, we processed six video sequences using our system, in which one to six people walked in a narrow laboratory space. The results demonstrate that our system is capable of handling cluttered scenes that contain severe occlusion and where people are frequently located in close proximity to each other. Moreover, minimal information is required for tracking, instead of using full camera images, which is communicated over the network in our system. Hence, the proposed method also has the advantage that commonly used network devices are sufficient to construct our tracking system.

Finally, we applied our method to a prototype demo system for detecting arm raising gestures. The results confirmed that our system is suitable for stable gesture recognition even in a highly cluttered scene.

第1章 序論

1.1 研究背景

スマートスペースや Natural User Interface (NUI) が急速に私達の生活に身近なものになりつつある中、人物の位置やジェスチャを取得する技術の重要性がますます高まっている。東京国立博物館が提供する「トーハクナビ」アプリは、来館者がスマートフォンやタブレット端末にインストールしておく、来館者の位置に応じて自動的に展示品の案内をする。家庭用ゲーム機では任天堂 Wii がモーションセンシング可能なコントローラで入力デバイスに画期的な変化をもたらしたことに始まり、最近では Xbox Kinect によりユーザーはコントローラを手をせずジェスチャ入力ができるようになった。家庭用ゲーム機にとどまらず、ハンドジェスチャ入力を組み込んだ家電製品が Consumer Electronics Show (CES) 等の家電見本市で盛んに発表されている。これらの技術の普及の為に、ユーザーが特別な機器を身に付ける必要がない事が望ましく、その点でコンピュータビジョン (CV) 技術は大きな役割を果たしている。ジェスチャ認識、指紋認証、顔認証のように、センサに比較的近い限られた空間を対象とした認識技術は既に製品に組み込まれるほど普及している。また、建物内や交差点のような、より広い空間内を自由に人物が動く条件での認識技術も、多くの研究者が意欲的に取り組んでおり進歩がめざましい [42, pp.585-587]。このような条件下で人物認識を困難にする主要因の一つとして、隠蔽が知られている。これは画像上で、ある物体が別の物体の背後に隠れ、一部あるいは全体が見えなくなってしまう現象である。カメラ視点画像上では奥行き情報が失われるために、画像上で重なっている領域は容易には分割できない。

近年の研究では、ステレオカメラや Kinect などのように、実時間で奥行き情報を計算できる Depth Camera が普及してきており、隠蔽を解消する手段として積極的に利用されている。通常のカメラでは失われてしまう奥行き情報は隠蔽に対して本質的な手掛かりとなる上、計測値が人物の見かけや照明変化の影響を受けにくく、実用上有用だからである。人物が鉛直方向に移動しないという仮定の下に、奥行き情報を平面図で簡略化した表現（例えば、Occupancy Map[4]）は、人物追跡に利用しやすい特徴量として知られており、多くの研究で採用されている [4][22][25][33][12][17]。しかし、姿勢・ジェスチャを知るのに必要な人物領域の形状は平面図表現ではほとんど失われてしまう。そのため既存研究では、隠蔽を扱いやすい、複数視点の情報を統合しやすい、といった平面図表現の特徴を活かして、頑健な人物動線検出の研究が発展している一方で、ジェスチャ認識のように人物形状を必要とする応用には適用されてこなかった。

1.2 本研究の目的

以上の背景をふまえて、本研究では複数視点の奥行き情報を使って高い混雑度での追跡を実現すると同時に、領域分割も行える手法の提案を目的とする。さらに、挙手を検知するジェスチャ認識のプロトタイプシステムを構築して手法の実用性を示す。

提案手法は大きく2段階に分かれており、まず最初に平面図上で追跡処理（4.2節、4.3節および4.4節参照）を行った後、平面図上での追跡結果をカメラ視点画像に投影して領域分割を行う（4.5節参照）。

平面図上での追跡は、複数視点の Occupancy Map を時系列で最適化することで、一般に見られる2フレーム間でのマッチングによる追跡に比べて、隠蔽に頑健な追跡を実現した。最適化は反復法を用いないヒューリスティックな手法を提案し実時間での計算が可能である。カメラ視点画像上での領域分割は、平面図上での追跡結果にカメラ視点画像への投影情報を適用する事で行われる。従来の平面図上での追跡手法には、平面図上での人物形状を矩形や円形のような単純形状でモデルしているために、任意の形状変化に対応できないものが多かった。従って、手の曲げ伸ばしの形状変化に追従出来ない場合がある。また2人以上の人物が近接している場合に各人物に対応する形状モデルの一部が重なり、その結果間違った分割につながる場合もある。一方、提案する追跡手法は、人物の大きさの違い、姿勢の変化といった人物の任意の形状変化に追従して平面図上での領域分割ができる利点も併せ持ち、平面図上での領域をカメラ視点画像上に投影する事で混雑環境下でも正しく領域分割ができる。なお、従来は領域分割結果を人物追跡に用いてきたが、提案手法はその逆である。隠蔽のある領域を正しく分割することは前述のように困難である事が知られているが、提案手法によってカメラ視点画像上で隠蔽に頑健な領域分割ができるようになることを、計算機実験の結果を用いて示す。

1.3 本研究の成果

本研究の主な成果を下記に述べる。

- 奥行き情報を平面図上に投影した Occupancy Map を時系列で得て、それに最も当てはまる軌跡を推定することで、隠蔽に頑健な追跡を実現した。さらに反復法を用いない新しい最適化手法を提案し、追跡処理が実時間で計算可能な事を確認した。
- 提案手法は人物の大きさや姿勢の変形に追従して、平面図上での領域分割が可能である。この領域を平面図上からカメラ視点画像上に逆投影することで、カメラ視点画像上における隠蔽に頑健な領域分割が行える。
- 提案手法は、ステレオカメラのネットワーク化において、画像そのものは伝送せずに、処理に必要な情報のみをネットワークに送信する。これにより通信量の小さなネットワーク機器で構築でき、拡張性に富むシステムを実現した。

第1章 序論

- 混雑環境下で挙手を検知するプロトタイプシステムを構築し，提案手法の実用性を示した．

1.4 本論文の構成

本論文の構成は以下の通りである．

第2章 関連研究

人物追跡に関連する研究を「複数視点を利用する追跡手法」「奥行き情報を利用する追跡手法」「マルチフレーム最適化による追跡手法」「Data Association」の4つのカテゴリに，また領域分割に関連する研究を「ビデオセグメンテーション」と「ジェスチャ認識への応用」の2つのカテゴリにまとめる．

第3章 システム構成と提案手法の概要

開発したシステムの構成と，提案手法の概要を説明する．

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

人物追跡および領域分割に用いる特徴量の定義を述べた後，時系列の奥行き情報から平面図上での人物追跡を行う手法を提案する．次に，平面図上での追跡結果をカメラ視点画像上に投影して領域分割を行う手法を提案する．

第5章 計算機実験

混雑度の異なる映像に提案手法を適用し，追跡および領域分割の結果を示す．追跡エラーを定量的に評価して，手法の限界も明らかにする．

第6章 ジェスチャ認識への応用

提案手法の応用例として，挙手を検知するジェスチャ認識のプロトタイプシステムを構築し，提案手法の実用性を示す．

第7章 結論

提案手法の成果と限界をまとめ，手法の発展性について述べる．

第2章 関連研究

本章では、まず人物追跡について4つのカテゴリに分類して関連研究を概説する。次に領域分割の研究例について、ビデオセグメンテーションを中心に関連研究を概説し、さらにその応用として、ジェスチャ認識の研究例にも言及する。

2.1 人物追跡

人物追跡技術の研究はコンピュータビジョンの分野でも古くから研究が行われており、膨大な研究例が存在し、下記に例示するように様々な基準で分類されうる。

- 追跡対象の数（一人・任意の人数）
- カメラの台数（単独視点・複数視点）
- カメラの設置方法（水平に取り付け・斜めに取り付け・天井から真下を向けて取り付け）
- 追跡ウィンドウの形状（矩形領域、任意形状領域）

本節では、「複数視点を利用する追跡手法」「奥行き情報を利用する追跡手法」「マルチフレーム最適化による追跡手法」「Data Association」の4つのカテゴリで既存研究を概説する。

2.1.1 複数視点を利用する追跡手法

従来より、単独視点の映像中の人物を追跡する研究が盛んに行われてきた。混雑環境下でも頑健に追跡を行うために、特徴として頭部領域を用いたり [2]、歩行者挙動モデルを導入する [54] 等、色・テクスチャの情報を高度に利用する手法が考案されてきた。特に近年の顔認識技術の発展は目覚ましく、信頼度の高い特徴として期待が集まっている [51][49]。しかし、ある人物が完全に隠蔽されてしまう場合、その間の追跡はできない。完全な隠蔽あるいは部分的な隠蔽が発生していても途切れの無い追跡を行うために、複数のカメラを利用する研究例が近年多数報告されている [28][45][12][30][36][48]。

筒井ら [45] はオプティカルフローを使って単独視点画像上で追跡を行い、隠蔽が発生した場合に他視点の追跡情報を援用する手法を提案した。このような方法では、単独画

第2章 関連研究



図 2.1: 2 視点を利用する人物追跡システム (Krumm ら [28] から引用). 図中では奥行き情報と色情情報を使って検出した 2 人の人物を, 平面図に赤色と青色の円で表して, プロジェクタで表示している.



図 2.2: 最大 16 台のカメラを利用する人物追跡システム (Mittal ら [30] から引用). 単独視点画像上で前景の領域分割のみを行い, 全ての視点の結果を統合して追跡を行う.

像上での追跡や人物検出に失敗した場合には, 他視点の情報があっても修正出来ないという欠点がある. 例えば, ある視点の画像で 2 人の人物を 1 人と認識してしまったり, 逆に 1 人の人物を 2 人と認識してしまった場合, 他の視点の情報との一貫性が無くなり追跡に失敗してしまう. これに対し, Mittal ら [30] や Salinas ら [36] は単独視点画像上では追跡を行わずに特徴抽出のみを行い, 全ての視点の特徴を統合した上で追跡を行う手法を提案している. Zabulis ら [48] は視体積交差を使って平面図上に人物領域を射影し, Blob トラッカーを割り当てることにより平面図上で人物を追跡する手法を提案し, 開発したシステムを博物館に導入して, デモンストレーション空間内での来館者の位置・移動履歴に応じて動的にコンテンツを提示する応用例を示した. この追跡手法では出入口を 1 カ所に限定し, 出入口を人物が通る場合に入室・退室を判断して Blob トラッカーの新規作成・破棄を管理している. このため追跡領域内で人物同士が近接しても人物を見失う事は無いが, 出入口を 1 カ所に限定することは実用上の制約になる.

また, これまで述べたような複数の視点によって同じ領域を重複して撮影する事によって冗長性を高める研究例の他に, 複数のカメラを使ってより広い範囲を監視する手法も報告されている [15].

2.1.2 奥行き情報を利用する追跡手法

計算機の処理能力の向上に伴って, 実時間で奥行き計測を行う Depth Camera が多数製品化されている. ステレオカメラ (図 2.5 左) は 2 つのカメラから成り, ステレオマッチング [42, Chapter.11] によって視差情報から奥行きを計測する. また Carmine (図 2.5 右) 等の RGB-D カメラは赤外線ランダムドットパターンを計測対象に照射し, ドットパター

第2章 関連研究



図 2.3: 天井に設置された 5 台のカメラを利用する人物追跡システム (Salinas ら [36] から引用). 単独視点画像上でパーティクルフィルタを使って特徴抽出した後, 全ての視点の情報を統合して追跡が行われる手法を提案し, 図のように隠蔽の頻発する混雑環境下でも追跡を可能にした.

ンの歪み方を手がかりにして奥行きを計測する. Depth Camera は通常のカメラでは失われてしまう奥行き情報を補い, 部分的な隠蔽の解消に対して本質的な情報を与える. また, 計測値が人物の外観や照明変化の影響を受けにくく, 実用上有用であるという利点もあり, 近年多くの研究で Depth Camera が利用されている [21][28][4][3][12][22][25][33][34].

Haritaoglu ら [21] は奥行き情報と輝度情報から前景抽出と人物検出を行う手法を提案し, 屋外での実験で照明変化や隠蔽の影響を受けにくい追跡手法を報告している. Beymer[4] は, 人物が鉛直方向に移動しないという仮定の下に奥行き情報を平面図で簡略化した表現の一つである Occupancy Map を考案し, これに Blob トラッカーを割り当てて追跡するシステムを開発した. 平面図表現を利用する利点として, カメラ視点画像上での追跡結果から追跡領域内の人物位置 (例えば平面図上の XY 座標) を求めるためには, 座標変換といった追加の処理が必要なのに対して, 平面図上での追跡結果は人物位置も同時に表しているため, 実利用上有益である. Harville ら [22] は平面図上に高さ情報と色情報を投影して作成した Plan-view height map および Plan-view color map を考案した. 平面図表現と色情報の併用は [3][33][34] など複数の研究で見られる. 色情報は個人の識別に役立ち, 色情報を用いない場合に比べて人物の近接に対してより頑健な追跡が可能である.

Krumm ら [28] は奥行き情報と色情報を使って単独視点画像上で前景抽出と人物検出

第2章 関連研究

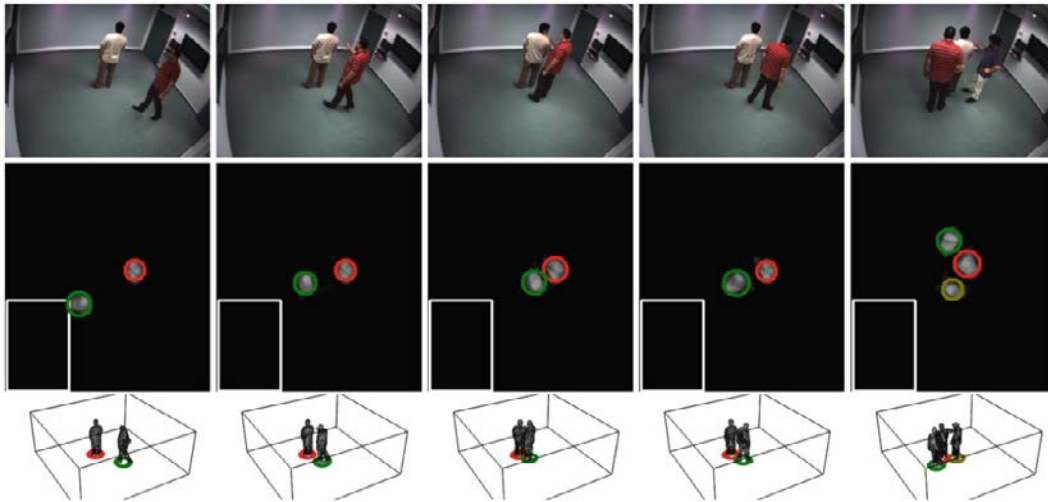


図 2.4: 8 視点を利用する人物追跡システム (Zabulis ら [48] から引用). 視体積交差法により平面図上での人物位置を推定し, Blob トラッカーで追跡する. Zabulis らは博物館に提案システムを導入して, インタラクティブな情報提示デバイスに応用している.



図 2.5: Depth Camera の製品例. Bumblebee (PointGrey 社, 左) と Carmine (PrimeSense 社, 右)

を行い, 複数の視点の検出結果を利用して追跡を行う手法を提案した. 実験では, 一定以上の混雑では単独視点画像上での人物検出に失敗するため, 全体の追跡も一貫性がとれなくなると報告されている. そこで, 単独視点画像上では追跡を行わずに, 複数の視点から得た情報を統合した後に追跡を行う手段が, 複数の研究者によって考案されている. これは 2.1.1 節で述べた, 単独視点画像上で追跡した結果を統合する手法の短所および解決のアプローチと同様である.

Beymer は [4] において単独のカメラしか利用しなかったが, Beymer の手法 [4] 以降, 多くの研究において平面図表現を利用して複数の視点の情報を統合する手法が提案されてきている. Iocchi ら [25] は Plan-view height map と人物モデルに基づいて, 4 台のステレオカメラの情報を統合して追跡を行う手法を提案した. また次節で述べるように, Darrell ら [12] は, 複数視点から得た時系列の Occupancy Map を利用することで隠蔽に頑健な追跡を行う手法を報告している.

第2章 関連研究

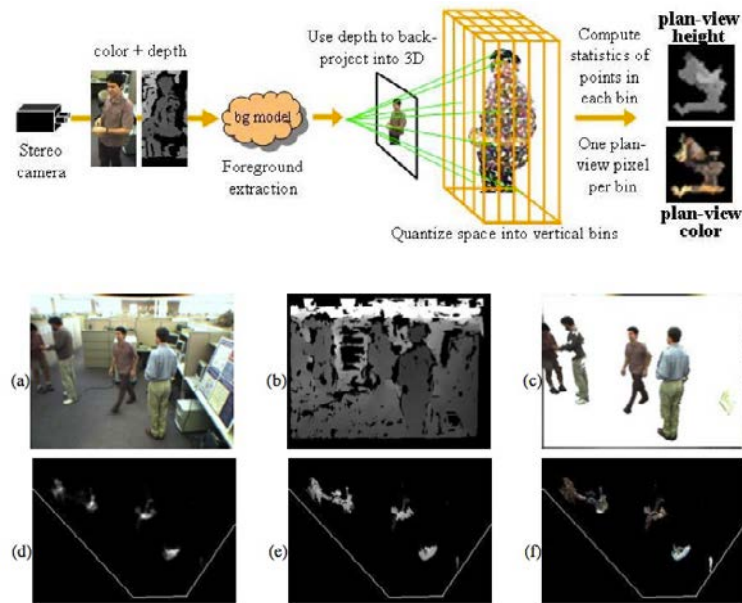


図 2.6: 平面図表現 (Plan-view map) の作成過程 (上) と実画像への適用例 (下). (a) カメラ視点画像 (b) 奥行き情報 (c) 前景画像 (d) Occupancy Map (e) Plan-view Height Map (f) Plan-view Color Map (Harville ら [22] より引用)

2.1.3 マルチフレーム最適化による追跡手法

先述までの追跡手法は、隣接する 2 フレーム間の情報を使って追跡を行っていた。典型的にはフレーム毎に人物領域を検出し、隣接フレーム間で最も類似する領域の対応付けを行う。人物の位置・速度・外観の情報を含む Blob トラッカーを利用する手法も多いが、これも同様にフレーム毎に検出された人物領域に最も当てはまるように Blob トラッカーの状態が更新される。こうした手法では、隠蔽や人物同士が近接した場合に誤追跡を引き起こすことを防ぐために、前述の [28][3][33][34] のように色情報や頭部の情報など、より豊富な特徴を使って追跡精度を改善している。このように 1 枚の画像に含まれる情報を出来る限り利用するアプローチがある一方で、複数時刻の画像列 (マルチフレーム) から最適な人物の動線を推定する手法が報告されている。

藤村ら [53] は単独視点の画像列に動的計画法を使った最適化処理を適用して、先頭フレームで指定した矩形領域を隠蔽に頑健に追跡する手法を提案した。

Darrell ら [12] の手法は、本論文で提案する手法に類似しており、複数視点のステレオカメラから画像列を取得し、マルチフレームの Occupancy Map を使って最適な軌跡の推定を行う。Occupancy Map には色やテクスチャの情報が含まれないにも関わらず、実験室内の 3 人の追跡に成功したことが示されている。しかし、それ以上混雑した環境での実験は実施されていない。また、Darrell らは反復法で最適軌跡を計算しており、実時

第2章 関連研究

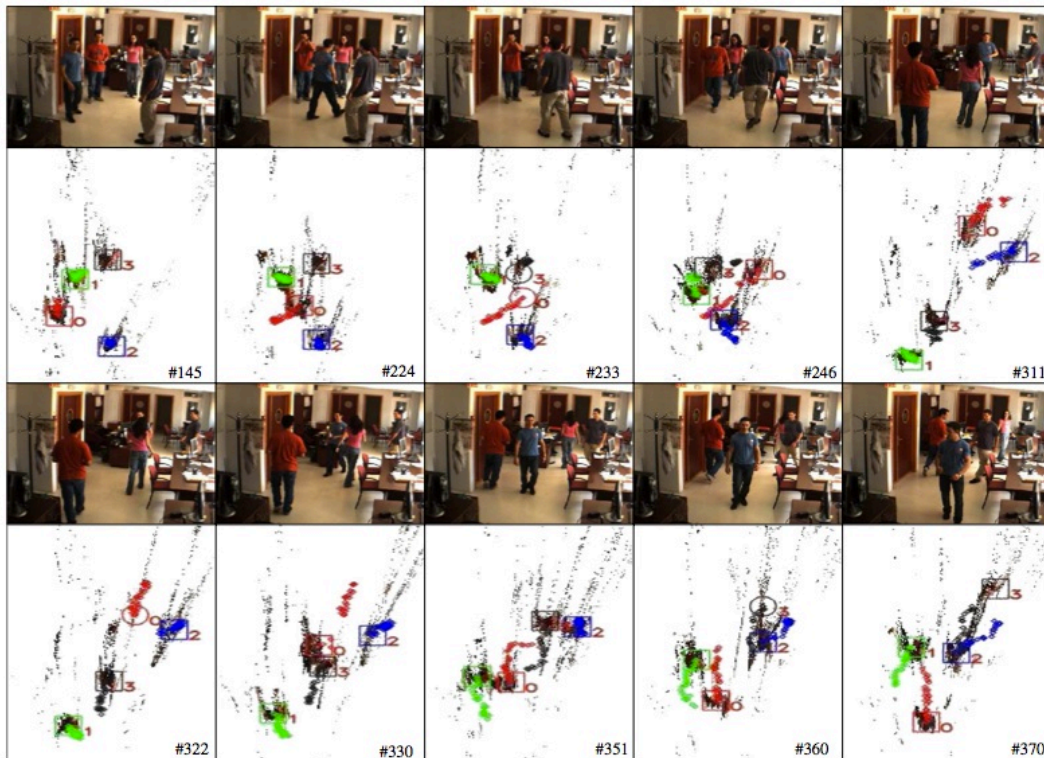


図 2.7: 奥行き情報を基にして得た Occupancy Map に、色情報を併用して追跡を行う手法 (Salinas ら [33] より引用). 図はカメラ視点画像と対応する平面図を表しており、各人物の ID を表す数字と軌跡が描画されている.

間、オンライン処理が困難であると述べている.

Fleuret ら [16] は、隠蔽と照明変動がある環境で、6 人の人物の追跡に成功している. 実験では $5.5 \text{ m} \times 5.5 \text{ m}$ の領域を、同期された 4 台のカメラで撮影している. 軌跡推定の手順は、大きく次の 2 ステップで構成される. 最初のステップは、Probabilistic Occupancy Map (POM) の計算である. ここでは、同時刻に得られた複数の視点の画像を利用し、提案する Generative Model によって明示的にオクルージョンを処理し、床平面上の任意の位置における占有確率 (Probabilities of Occupancy) を求める. Fleuret らは、この床平面と占有確率のマップを POM と呼んでいる. 次のステップでは、100 フレームの POM を用いて、動的計画法によって最適な軌跡を推定する. 計算時間は、POM 計算が 6 fps、最適化が 2 fps であると報告されている. 本手法は追跡の頑健性と計算効率の面で実用的である. 一方で、同期されたカメラの利用を前提としているため、同期用の装置やソフトウェア、そしてカメラ自体にも同期機能が必要となり、システムの拡張性や構築の柔軟性の面で制限が生ずる.

本論文で提案する手法は、色やテクスチャの情報を含まない Occupancy Map だけを

第2章 関連研究

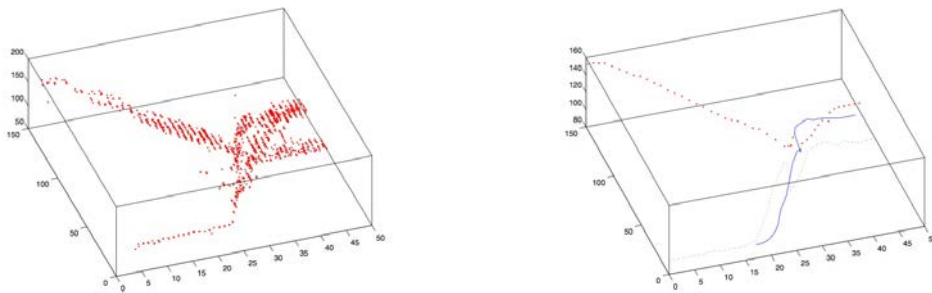


図 2.8: 3 台のステレオカメラから得た Occupancy Map を時系列で並べて (左), 最も当てはまる軌跡 (右) を推定する手法 (Darrell ら [12] より引用) .

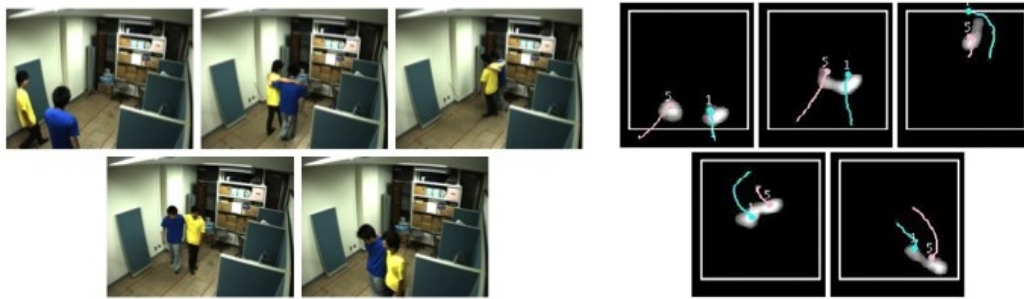


図 2.9: 原口ら [52] は提案手法に, 色情報を追加利用する手法を提案した. 人物同士が近接して歩いている (左), 混同せず正しく追跡が可能である (右) .

使いながらも, マルチフレームで最適化を行う事で混雑した環境でも追跡を可能にした. 色やテクスチャの情報は照明変化が大きい場合や, 解像度に制限がある場合には利用しにくい. しかし奥行き情報は照明変化に頑健であるので, 実際の環境で利用する上で信頼度の高い情報である. ただし, 色やテクスチャの情報は信頼度は下がるものの, 補足的に利用する事で, 奥行き情報のみで追跡する場合よりも精度が上がる事が期待される. Darrell ら [12] は色情報の追加利用は動的計画法の計算量を著しく増大させるので利用出来ないと述べているが, 本論文の提案手法は 4 章で説明するように, 動的計画法ではないヒューリスティックな最適化手法を独自に考案しており, 効率的な計算量で色情報を追加利用することも可能である [52] (図 2.9) .

また, 最初に隣接フレームの対応付けによる追跡を行った後, 次の段階で時間軸を考慮して追跡結果を最適化する手法も知られている. 最初の段階で生成される軌跡は tracklet と呼ばれており, 隠蔽やカメラの死角によって途切れが生じている軌跡を指す. Singh ら [40] は複数の人物の tracklet を適切に接続して途切れの無い軌跡を生成する手法を報告している.

2.1.4 Data Association

本節では、物体追跡問題を定式化した統計的枠組みとして知られている、Data Association について述べる。そして数例の既存研究と提案手法とを比較し、差異を明確化する。

Data Association とは、トラックと、画像から抽出された特徴量との対応付けを探索する処理の事である。Multiple hypothesis algorithm [9] は、マルチフレーム画像列から抽出した特徴量を利用して、複数の人物の追跡を行う事が出来るアルゴリズムである。このアルゴリズムの問題点は、計算量が指数関数的であることである。追跡対象物体の数が既知である場合に準最適解を求める手法として、Joint Probabilistic Data Association Filter (JPDAF) も知られている。しかし実用上、追跡対象物体の数が既知であるという条件は、現実的では無い。

Hofmann ら [24] らが提案した階層的アプローチは、複数人物の追跡問題を大域的最大事後確率推定問題で定式化し、実行可能な解法を導いた。この手法では、追跡に先立って、単独フレーム毎に人物検出を行って、位置・サイズ・検出時刻の情報を生成する。追跡処理はこの情報に基づくため、単独フレームでは人物検出が困難な程度の混雑環境下では利用できないという課題がある。

Data Association には、幾つかの拡張を試みた研究事例が存在する。例えば、Gennari ら [18] は、確率的 Data Association Estimator を改良した方法により、人物をグループ単位で追跡するアルゴリズムを提案している。この手法は、空間的位置・密度・速度に基づいて画素レベルで類似領域を統合することにより、グループ単位での人物追跡を行う。広範囲・大人数の、人物の動きの特性を分析する目的に有効である。これに対して本論文は、比較的狭い領域で個別の人物の軌跡を継続的に追跡することを目的としている。

カメラの代わりにレーザー測距計を使った事例が、Schulz ら [39]、Cui ら [11] により発表されている。レーザー測距計には、テクスチャに乏しいなど、視覚的情報が得られない対象物体でも測距可能であるという利点がある。しかしこれらの研究事例では、Hofmann ら [24] と同様に、単独フレームから人物位置を検出できる必要がある。[39] では、レーザー測距により得られた外形が局所最小値を取る位置を、人物位置として表す。[11] では、数フレームの測距データの積算分布から局所最大値を探索することで、人物の足の位置を検出する。積算分布を求める際に利用するフレーム数は、人物の動きに対して十分短い時間期間に設定される。

これらの手法に対し、提案手法は単独フレームから人物検出を行うことなく、マルチフレーム画像列から人物軌跡を推定する。

2.2 領域分割

領域分割は、画素をある基準に基づいてグループ化する処理である。基準は適用事例によって変化する。例えば単に類似色をまとめるだけの場合もあるし、物体認識の前処理として「人」「動物」といった意味のある領域を取り出したい場合もある。また画像処

第2章 関連研究

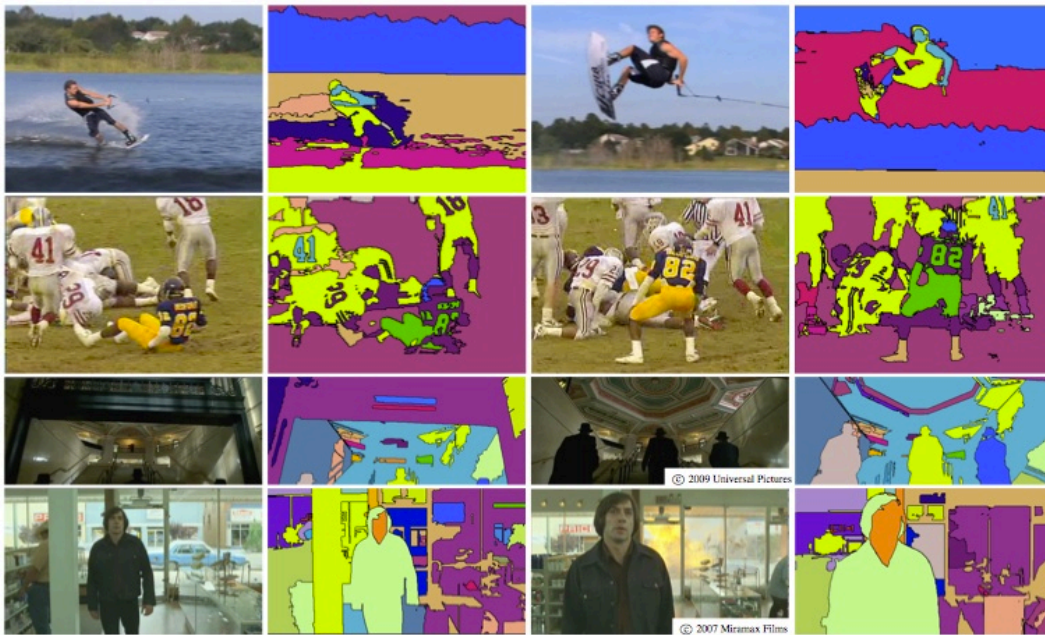


図 2.10: テクスチャの類似する領域を統合し，時間軸での対応付けも行う領域分割手法 (Grundmann ら [20] より引用)．異なる時刻でも同一の領域には同一の ID が割当てられている．

理ソフトウェアで背景と人物を切り分けたいとき等に，手動で大まかな輪郭を与えると，それに併せて領域分割が行われるようなインタラクティブな手法も数多く報告されている [32][6]．

本節では，特にビデオセグメンテーション (Video Segmentation) と呼ばれる，動画像を対象にした領域分割について概説する．

2.2.1 ビデオセグメンテーション

単独の画像で領域分割を行う場合には，Active contours, Mean shift 法, グラフカット等が知られている [42, Chapter.5]．正確な領域分割のために，複数の視点の画像を利用したり [10]，奥行き情報やオプティカルフローを使って 1 人の人物領域を 2 つ以上に過分割することを防ぐ手法 [8] など様々な方法が提案されている．ただし，これらの手法を動画像に適用しても，単独時刻で領域分割が行われるので，時間を通して同一人物に同一ラベルが割り当てられる保証は無い．そこで，一般的には隣接フレーム間で外観の類似した領域を対応付ける事によって時間を通して領域の ID が変化しないようにする措置がとられる．この処理方法は既に 2.1 で述べてきた人物追跡手法とよく似ており，追跡ウィンドウが任意に変形する方式の人物追跡手法であるととらえる事も出来る．Mittal

第2章 関連研究

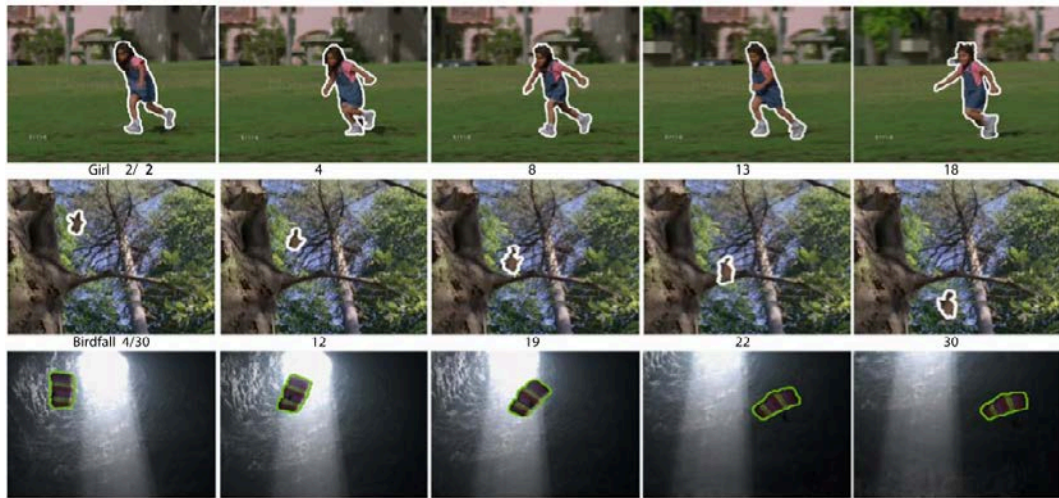


図 2.11: 初期フレームで与えた領域を追跡対象物体の変形に追従させる領域分割手法 (Tsai ら [44] より引用)

ら [30], Mohammadi ら [31] は、複数視点のビデオセグメンテーションを行う手法を提案している. このようなビデオセグメンテーションは、前段階であるフレーム毎の領域分割が正しく行えている事が前提であり、2人以上の人物を誤って同一領域と見なしてしまったり、過分割がある場合、正しく対応付けすることが困難になる. これに対し、単独の画像上で過分割した後、空間的・時間的に近傍の領域をテクスチャや動きの類似性に基づき統合する事で、時間を通して一貫したラベルを割り当てる手法が報告されている [20][23]. しかし、本来の目的が色の類似したピクセルを同じ領域に分類する事なので、頭部と胴体等で大きく色の異なる領域は統合されない. 統合の度合いを強める事でそのような領域も統合できるが、そうすると異なる人物領域も統合してしまう恐れがある. それに対して、4章で述べる我々の提案手法は奥行き情報を基にした人物追跡結果を利用して領域分割を行い、1人の人物に対して常に1つの領域を対応させることができる. これはジェスチャ認識のように、複数の人物がいる映像中から、特定の1人の領域が必要な応用例に適用しやすいという利点がある.

その他のビデオセグメンテーション手法としては、初期フレームで与えた領域を、対象物体の変形に追従して分割し続ける手法 [29][44] などが報告されている.

2.2.2 ジェスチャ認識への応用

近年、NUIの一つとしてジェスチャ認識が注目を集めていて、既にハンドジェスチャでメニュー操作が可能な家電製品が CES 等で盛んに発表されている. 家庭用ゲーム機向けのアクセサリとして 2010 年に発売された Xbox Kinect は、カメラの正面にいる複数の人物の全身の動きをとらえる事が出来、メニュー操作やダンスゲームを実現する画期

第2章 関連研究



図 2.12: 人物のシルエットを時系列で並べて得られる 3 次元形状を基にしてジェスチャ認識を行う手法 (Blank ら [5] より引用)

的なデバイスとして、多くの研究者や企業が利用し、ゲームに留まらず様々な応用を試みている。ジェスチャ認識を可能にするためには、まず始めに動画像から特定の人物の領域を取り出す事が不可欠であり、ビデオセグメンテーションが本質的な役割を果たす。Xbox Kinect SDK はビデオセグメンテーションで得られた人物領域に人物骨格モデル当てはめて、人物の姿勢を計算する機能を備えている。Blank ら [5] ら、Abdelkader ら [1] は、正面からとらえた人物領域を時系列に並べて 3 次元空間でモーションを認識する手法を考案した。これらの手法で用いられる人物領域は 2 値マスクであるが、Salinas ら [35] は奥行き情報を追加利用して精度を向上させる事に成功している。

その他のジェスチャ認識手法としては、動画像をフレーム毎に過分割し、それを統合する事無く予め登録しておいたジェスチャを検出する手法 [26] や、動きの情報から行動解析を行う手法 [37] など様々なバリエーションがある。

2.3 まとめ

本章では、人物追跡の研究を「複数視点を利用する追跡手法」「奥行き情報を利用する追跡手法」「マルチフレーム最適化による追跡手法」「Data Association」の 4 つのカテゴリに分けて概説した。次に領域分割の研究を「ビデオセグメンテーション」と「ジェスチャ認識への応用」の 2 つのカテゴリに分けて概説した。

本節では、従来手法の問題点を挙げ、提案手法の成果を明らかにする。まず人物追跡では、「複数視点の利用」「奥行き情報の利用」「マルチフレーム最適化」の 3 つの特長を取り入れる事で実環境で安定した追跡が出来ると考えられるが、マルチフレーム最適化には計算時間が大きくなるという問題点がある。そこで本研究では、従来のような動的計画法や反復法による最適化ではなく、ヒューリスティックな最適化手法を独自に提案し、計算時間の問題を解消する事に成功した。領域分割では、従来手法は色やテクスチャの類似性を利用して近傍ピクセルあるいは領域を統合するために、必ずしも 1 人の人物に 1 つの領域が割当てられる訳ではなく、通常は顔や胴体等で異なる領域となる事を述べた。本研究では領域分割の応用例としてジェスチャ認識を意識しているが、その為には人物と領域には 1 対 1 対応の関係が必要である。そこで本研究では、奥行き情報を使った人物

第2章 関連研究



図 2.13: 予め登録しておいた動き（左）を未知の動画像中（右）から検出する手法（Ke
ら [26] より引用）

追跡結果を，カメラ画像上に逆投影して領域分割する事により，人物と領域を1対1対
応させている．

第3章 システム構成と提案手法の概要

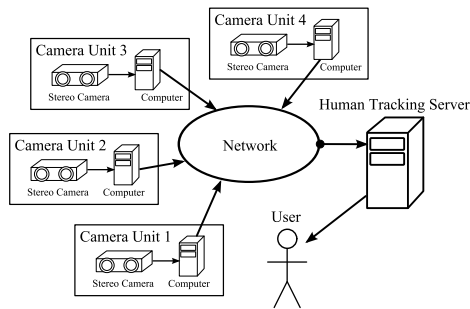


図 3.1: システム構成

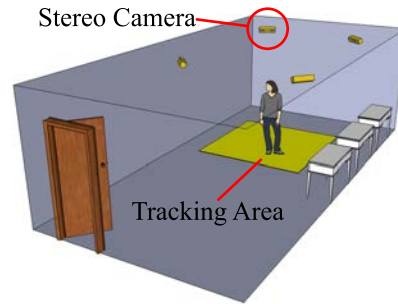


図 3.2: 実験室の概観

本章では、開発したシステムの構成と、提案手法の概要を説明する。

3.1 システム構成

本論文で提案するシステムは、“カメラユニット”と、追跡・領域分割プログラムが動作する専用の計算機とから構成され、これらの装置は1 GbpsのEthernetで互いにネットワーク接続されている。“カメラユニット”とは、ステレオカメラとそれに対応する計算機とから成る装置を指す。カメラユニットはステレオマッチングによる奥行き計測処理を行った後、4.1節で述べる中間データ作成を行い、ネットワーク経由で追跡・領域分割プログラム(3.1.2節)に転送する。追跡・領域分割プログラムは、カメラユニットからの中間データを受信・統合して人物追跡処理(4.2節, 4.3節および4.4節)と領域分割(4.5節)を行う。システム構成を図3.1に示す。

3.1.1 ステレオカメラネットワーク

奥行き情報は、実験室内に設置された4台のステレオカメラ(Bumblebee XB3, Point Grey Research社)で取得される。各カメラには一台ずつ計算機(CPU: Core 2 Quad Q6600 2.4 GHz, Memory: 2 GB, NIC: 1 Gbps)が接続されている。ステレオカメラと計算機は、IEEE1394インターフェイスで接続される。ステレオカメラは実験室の天井の、約4 m × 3 mの矩形範囲の四隅に取り付けられている(図3.2)。天井の高さは約2.5 m

第3章 システム構成と提案手法の概要



図 3.3: カメラユニット



図 3.4: 天井に取り付けられたステレオカメラ

である。隠蔽に対する頑健性を高める為に、各カメラは図 3.2 の黄色で示す追跡対象領域 (2.4 m × 2.4 m) を共通して撮影するように設置してある。5章の計算機実験では、人物は常に追跡対象領域内を歩行するよう指示される。なお、以後ステレオカメラを単に“カメラ”と呼び、“カメラ視点画像”は2視点分の画像ではなく奥行き情報が対応する方の1視点分を指す。カメラ視点画像は8bitモノクロ画像で、歪み補正済みであるとする。

提案手法はカメラが同期されていない事を前提に開発されており、同期のための設備を特別に要さないが、その代り正確な撮影時間を中間データに付して追跡・領域分割プログラムに送信する必要がある。本論文中でのセットアップでは、Network Time Protocol (NTP) によりすべてのカメラユニットの時刻を合わせた。より正確な時刻合わせが必要な場合は、例えば Precision Time Protocol (PTP) を利用すると 10 μ sec 以下の精度で同期できるが、外部装置が必要になる。

作成された中間データは、zlib ライブラリで圧縮後、ネットワークを介して追跡・領域分割プログラムに転送される。追跡・領域分割プログラムとカメラユニットの通信は Secure Shell (SSH) を利用してテキストベースで行われる。暗号化が不要な環境であれば、SSH の代わりに Remote Shell (RSH) を用いて暗号化にかかる負荷を軽減できる。

カメラキャリブレーション

提案手法は、あるカメラの座標系 Σ_C で表現される奥行き情報を、統一座標系 Σ_W で表現される Occupancy Map に変換してから、全ての視点の奥行き情報を統合する。これには、全てのカメラの位置関係を表す外部パラメータを正確に求める必要がある。内部パラメータについては Bumblebee 本体にキャリブレーションファイルが付属しており、無償で提供される Triclops SDK によって画像の歪み補正が可能である。カメラ i に対する外部パラメータは、回転行列 ${}^W\mathbf{R}_i$ と並進ベクトル ${}^W\mathbf{t}_i$ から構成される。

最初に、基準となるカメラを4台中から1台選び(カメラ1とする)、Zhang の手法 [50] を用いて外部パラメータ ${}^W\mathbf{R}_1, {}^W\mathbf{t}_1$ を計測した。参照パターンには、5行8列で1つのグリッドが 135 mm 四方のチェスパターン (付録 A, 図 A.1) を用いた。このとき、統

第3章 システム構成と提案手法の概要

一座標系 Σ_W で表された3次元座標 ${}^W\mathbf{x}$ と、カメラ1の座標系で表された3次元座標 ${}^1\mathbf{x}$ には

$${}^W\mathbf{x} = {}^W\mathbf{R}_1 {}^1\mathbf{x} + {}^W\mathbf{t}_1 \quad (3.1)$$

の関係がある。

次に、以下に述べる手順で ${}^1\mathbf{R}_i$ と ${}^1\mathbf{t}_i$ ($i = 2, 3, 4$) を推定する。まずチェスパターンをカメラ1とカメラ*i*の共通視野に置き、画像 I_1 と I_i を撮影すると共に、奥行き画像 DM_1, DM_i も同時に求める。続いて、OpenCVに含まれる `findChessboardCorners()` 関数で I_1 と I_i 中よりそれぞれ28点ずつコーナーを検出する。コーナーの座標から奥行き情報 DM_1, DM_i を参照すると、カメラ1とカメラ*i*のそれぞれの座標系から見た、コーナーの3次元座標のペアが28対得られる。何度かチェスパターンを移動させて同様の処理を行い、 n 対の3次元座標ペア $\{({}^1\mathbf{x}_j, {}^1\mathbf{y}_j, {}^1\mathbf{z}_j), ({}^i\mathbf{x}_j, {}^i\mathbf{y}_j, {}^i\mathbf{z}_j)\}, (1 \leq j \leq n)$ が得られたとすると、 ${}^1\mathbf{R}_i$ と ${}^1\mathbf{t}_i$ は最小二乗法を使って計算できる。最後に、先に求めておいた ${}^W\mathbf{R}_1$ と ${}^W\mathbf{t}_1$ を利用することで、カメラの*i*の外部パラメータは

$${}^W\mathbf{R}_i = {}^W\mathbf{R}_1 {}^1\mathbf{R}_i \quad (3.2)$$

$${}^W\mathbf{t}_i = {}^W\mathbf{R}_1 {}^1\mathbf{t}_i + {}^W\mathbf{t}_1 \quad (3.3)$$

のように推定される。

付録Aに、外部パラメータの推定結果を示す。

3.1.2 追跡・領域分割プログラム

追跡・領域分割プログラムは専用の計算機 (CPU: Xeon E5430 2.66 GHz \times 2, Memory: 8 GB, NIC: 1 Gbps) で実行される。カメラユニットから Occupancy Map を含む中間データを受信した後、ある時間長 T_B の分の Occupancy Map のまとまりを1単位として、次節で解説する人物追跡処理を適用する。本論文では、この Occupancy Map のまとまりの事を“クリップ”と呼ぶことにする。この計算機はシステムのユーザーインターフェイスも備えており、ユーザーはこの計算機を通じて、カメラユニットの遠隔操作やカメラ視点画像の確認などを行う事が出来る。

3.2 提案手法の概要

提案手法の概要を図3.5を使って説明する。図3.5(i)に示すように2人の人物がすれ違うシーンでの処理の例である。

第3章 システム構成と提案手法の概要

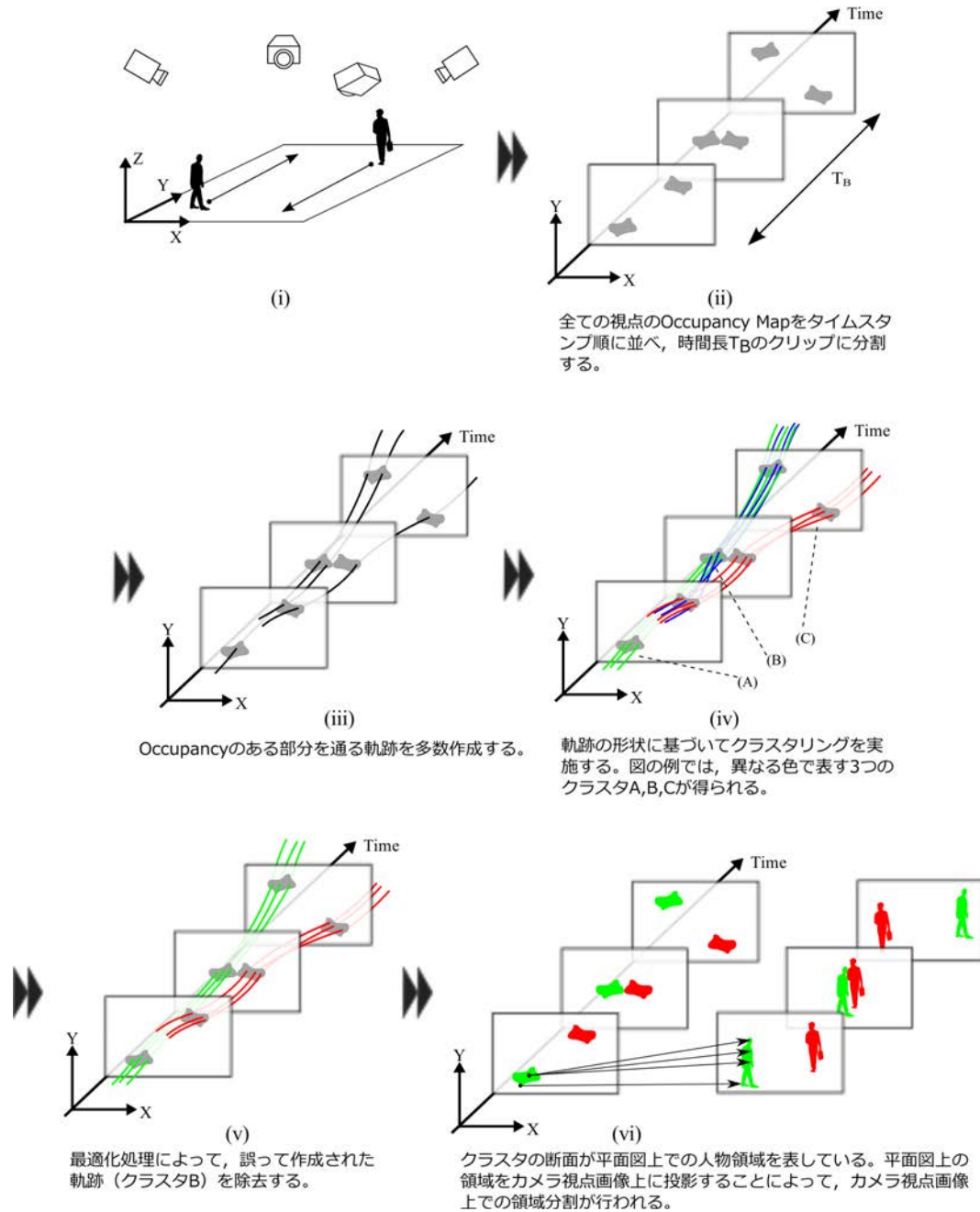


図 3.5: 提案手法の概要. (ii)(iii)(iv) 軌跡の作成とクラスタリング (4.2 節参照) (v) 軌跡の最適化 (4.3 節参照) (vi) カメラ視点画像上での領域分割 (4.5 節参照)

3.2.1 追跡と平面図上での領域分割

追跡・領域分割プログラムは、まず全てのカメラユニットから送られてきた複数視点の Occupancy Map を時系列に並べ、時間長 T_B のクリップに切り出す (図 3.5(ii))。人物追跡処理は1つのクリップに対して適用され、推定された軌跡を隣接クリップ間で対応付ける事で、時間を通して軌跡の ID を維持したまま追跡が継続される。対応付けの安定化のため、隣接するクリップには T_A の重複期間が設けられている。本論文では、 $T_B = 10$ sec, $T_A = 3$ sec としている。

まず最初に、4.2 節で説明するアルゴリズムによって、Occupancy のある部分を通るように軌跡を作成する。ここでは考えうる全ての軌跡が得られるように、十分多くの軌跡をランダムに作成する (図 3.5(iii)) ので、実際の数よりも多くの軌跡が作成される。そこで、軌跡の形状に基づいて、類似する軌跡をクラスタリングする (図 3.5(iv))。なお、この段階では誤った軌跡も作成される可能性がある。例えば、図の例では A,B,C の3つのクラスタが得られたとする (図 3.5(iv))。正しい軌跡は時間を通して1人の人物に対応してはならないが、クラスタ B は2人の人物にまたがっているため、クラスタ B に属する軌跡は誤った軌跡であるといえる。

このようなクラスタを除去し、正しい追跡結果を導くために、本論文では4.3 節で述べる新しい最適化手法を提案する。最適化処理では、全てのクラスタの組み合わせから、クラスタ同士が干渉せず、かつ軌跡が Occupancy のある部分を最も通り、さらに軌跡が滑らかなものが選び出される。図 3.5(iv) の例では「A,B,C」「A,B」「B,C」の組み合わせはクラスタが干渉する (同一時刻で2人の人物が重なる場合がある) ので不適切と判定される。「A,C」「いずれかのクラスタ1つのみ」「どのクラスタも使わない」という選択肢が残されるが、2つのクラスタを含む「A,C」が Occupancy のある部分を最も占有できるので、最適解と推定する (図 3.5(v))。

追跡結果として人物の軌跡が必要な場合は、図 3.5(v) の各クラスタに含まれる軌跡を時刻毎に平均化すればよい。一方、クラスタを平均化せず、任意の時刻でのクラスタの断面を考えた場合、これは平面図上での人物領域の形状を表している事が分かる。従来手法は平面図上での人物形状を固定サイズの矩形 (または円形等) でモデルしていた事に対し、提案手法は人物の任意の形状変化に応じて平面図上での領域分割が行われることが特徴である。

3.2.2 平面図上の領域からカメラ視点画像への対応付け

最後に、平面図上での人物領域をカメラ視点画像上に投影する事で、カメラ視点画像上での領域分割を行う。投影のための情報 (Geometry Map, 4.1 節参照) は、各カメラユニットから Occupancy Map を送信する際に併せて送信しておく。人物の大きさや姿勢の変化に適応して、平面図上での人物領域が求められるため、人物同士が近接していても領域を混同しない。一般にカメラ視点画像上での領域分割は、人物追跡等の処理に先立って行われるが、提案手法ではその逆で、最後にカメラ視点画像上での領域分割が

第3章 システム構成と提案手法の概要

行われる。隠蔽が発生している領域を正しく分割する事は困難であるが、提案手法は時系列で最適化を行う人物追跡処理と平面図上での領域分割が隠蔽を頑健に解消し、カメラ視点画像上での領域分割を正しく行う事が出来る。

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

本章では最初に、複数視点の Occupancy Map を時系列で最適化して、平面図上で隠蔽に頑健な追跡を行う手法を提案する。従来の追跡手法が固定形状の追跡ウィンドウを利用していた事に対して、提案手法では人物の大きさや姿勢変化に適応して平面図上での人物領域が変形し、追跡と同時に平面図上での領域分割が可能であるという特徴がある。次に、この結果をカメラ視点画像上に投影して、カメラ視点画像上での領域分割が行える事を示す。一般的にはカメラ視点画像上での領域分割を前処理として行ってそれを追跡に利用するが、提案手法はその逆で、追跡の後にカメラ視点画像上での領域分割が行われる。隠蔽が発生している領域を分割する事は困難だが、提案手法によって正しく領域分割する事が出来る。

4.1 Geometry Map, Occupancy Map および Silhouette Map の作成

カメラ視点画像のキャプチャ、ステレオマッチング、前景抽出および中間データ作成・送信はカメラユニットごとに分散的に行われる。カメラ視点画像のキャプチャとステレオマッチングは、それぞれ Flycapture SDK と Triclops SDK によって行われる。中間データは Occupancy Map (OM)、Geometry Map (GM)、Silhouette Map (SM) およびカメラ視点画像 I からなる。まずカメラ視点画像 I のキャプチャの後、ステレオマッチングにより奥行き画像 DM を作成する。 \bar{I} と \overline{DM} を予め人物がいない状態で撮影された背景輝度画像と背景奥行き画像とすると、輝度と奥行きの変化を I と \bar{I} 、 DM と \overline{DM} の差分により評価し、閾値を超えた部分を 1、それ以外を 0 で表したマスク F_I と F_{DM} を作成する。次にこれらの論理積 $F = F_I \cap F_{DM}$ を計算し、これを前景マスクとして利用する。閾値は輝度の差を 10、奥行き差を 20 cm とした。

次に中間データを作成する。それぞれのデータの役割と定義を下記に述べる。世界座標系 Σ_W は XY 面が実世界の床平面と一致するように設定した (図 4.1)。

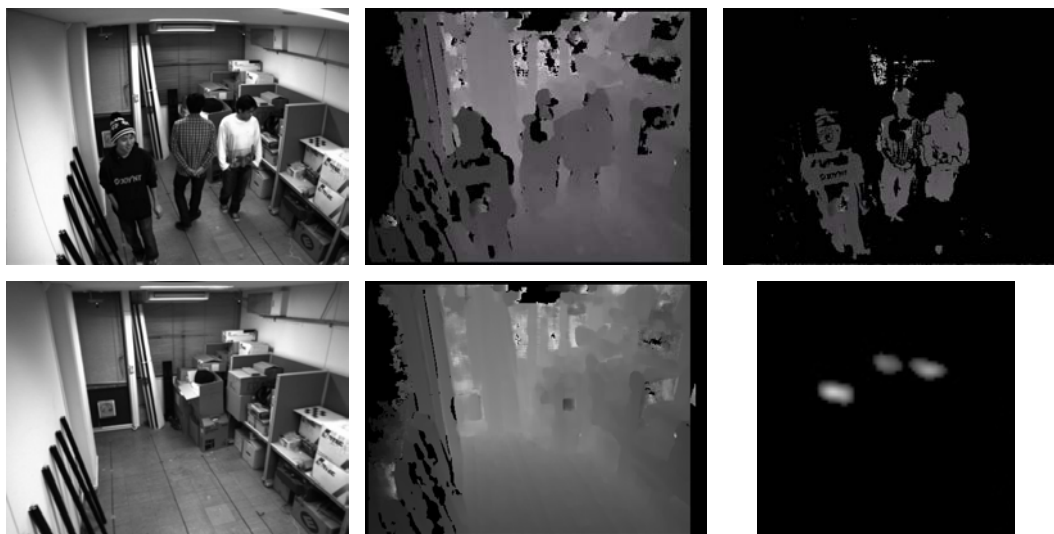


図 4.2: (1 列目) カメラ視点画像 I と背景画像 \bar{I} (2 列目) 奥行き画像 DM と背景奥行き画像 \overline{DM} (3 列目) 前景奥行き画像と対応する Occupancy Map

Silhouette Map

Silhouette Map (SM) はカメラ視点画像 I 上の位置 \mathbf{u} と YZ 平面上の位置の対応を表すマップである。6 章で述べるジェスチャ認識デモンストレーションのためのシルエット作成に用いる。GM と同様に、 YZ 平面に平行な平面 S_1 をグリッド状に分割して、あるグリッドに唯一の番号 h を付す時、 $h = SM(\mathbf{u})$ と定義する。グリッドの大きさは 1.7 cm 四方とした。シルエットは特定の人物領域のみを抽出して、人物領域を 1、それ以外を 0 で 2 値化したマスクである。ジェスチャ認識デモンストレーションでは、世界座標系の YZ 平面と平行な壁平面 S_1 に正対する人物の形状を認識する。そのため、カメラ視点ではなく特定の壁平面 S_1 にシルエットを投影している。

カメラ視点画像

カメラ視点画像 I は、ユーザーが追跡結果と共にカメラ視点画像を確認するためのものである。追跡処理には直接関わらないので、不要の場合は送信しなくても良い。本システムでは、必要なカメラ視点のみ、JPEG 圧縮を施したカメラ視点画像 I を送信している。

中間データの選び方について

最後に、中間データは圧縮後ネットワーク経由で追跡・領域分割プログラムに転送される。中間データ GM , OM , SM は前景マスク F と奥行き画像から計算されるので、

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

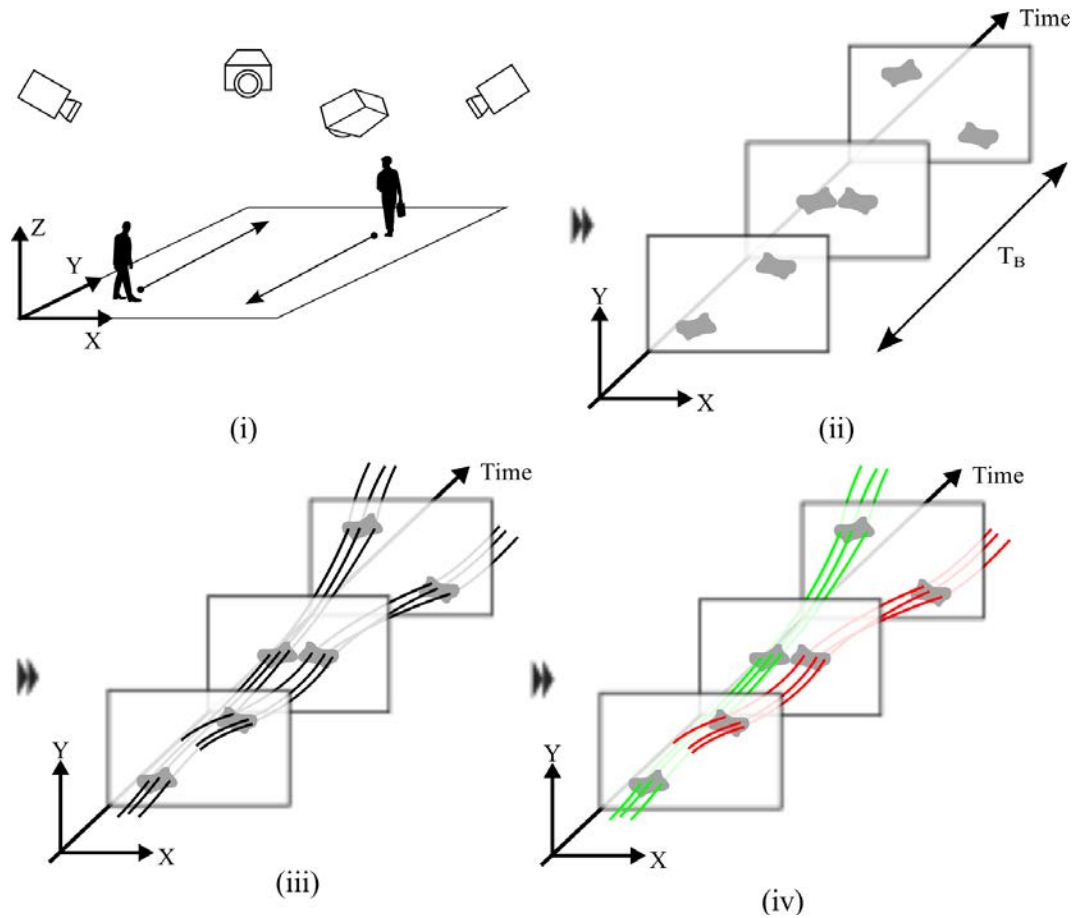


図 4.3: 軌跡の作成とクラスタリング処理の概要. (i) 2人の人物がすれ違うシーンを示している. (ii) 時間長 T_B の Occupancy Map のクリップ (iii) Occupancy のある領域を通るように多数の軌跡が作成される (詳細は図 4.4 参照). (iv) 軌跡の形状に基づいて類似した軌跡をクラスタリングする (詳細は Algorithm 2 参照).

前景マスク F を適用した奥行き画像を中間データとして代わりに送信し, 追跡・領域分割プログラム側で GM , OM , SM を計算することもできる. 本プロトタイプシステムでは, 実装段階において, マスク処理を施した奥行き画像を中間データとして送信する実験を行い, GM , OM , SM とサイズを比較したが, どちらもほぼ同等であったため, 分散処理の観点から GM , OM , SM をカメラユニット側で計算した方が良いと判断した. 中間データの選び方はアプリケーションや計算資源に強く依存するため, 分散処理・通信量を目的に応じて最適化する必要がある.

Algorithm 1 Trajectory Creation

Input: A starting point $\mathbf{x}_t = (x_t, y_t)$ at the time t .

Output: A trajectory $\mathbf{X} = \{\mathbf{x}_t, \mathbf{x}_{(t+\Delta t)}, \dots, \mathbf{x}_{(t+n\Delta t)}\}$

- 1: $\mathbf{X} \leftarrow \emptyset$
 - 2: **while** flow vectors are available around \mathbf{x}_t **do**
 - 3: Add \mathbf{x}_t to \mathbf{X}
 - 4: choose a flow-vector \mathbf{v} from the ones around \mathbf{x}_t randomly.
 - 5: $\mathbf{x}_{(t+\Delta t)} \leftarrow \mathbf{x}_t + \mathbf{v} \cdot \Delta t$
 - 6: $t \leftarrow t + \Delta t$
 - 7: **end while**
-

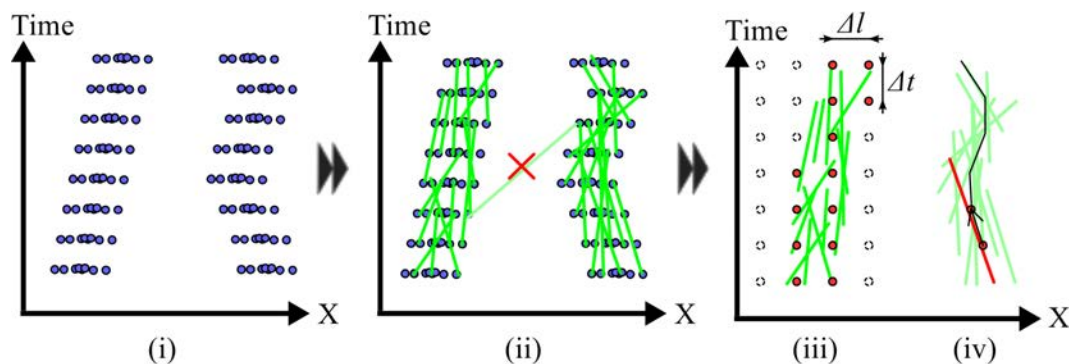


図 4.4: (i) 図 4.3(ii) の Occupancy Map (見易さのため Y 軸を省略) (ii) フローベクトルの作成過程. 赤の×印で示すフローベクトルは Occupancy の存在しない領域を通るため除外される. (iii) 軌跡の始点は時間幅 Δt 毎に幅 Δl の格子状に配置される. (iv) 各始点を Algorithm 1 に示す手順で延長し, 軌跡を作成する.

4.2 軌跡の作成とクラスタリング

追跡・領域分割プログラムは, 4.1 節で作成した中間データを全てのカメラユニットから受信し, 平面図上での追跡を実施する. 平面図上での追跡には Occupancy Map のみを用いる. 図 4.3 に, 本節の処理の概要を示す.

4.2.1 手順 1 : Occupancy Map のクリップ作成

まず全てのカメラユニットから取得した Occupancy Map を同一の XYT 空間に, 元の画像が取得された時刻順で並べたあと, 時間長 T_B のクリップに切り出す (図 4.3(ii)). 続く追跡処理はクリップ毎に独立して行われ, その後 4.4 節で述べる手順によりクリップ間で軌跡を形状に基づいて対応付ける事によって, 時間を通して軌跡の ID が維持される. 隣接クリップ間で軌跡の形状を比較するために, 隣接するクリップは 3 秒間の重複を持つ.

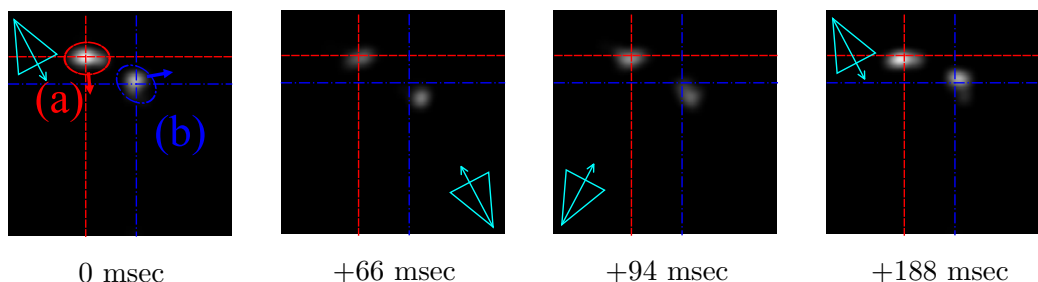


図 4.5: 異なる視点で独立に作成された OM の例. 画像の撮影タイミングは非同期であり, 各 OM に対する相対撮影時間を付記した. カメラは水色の三角形の方向から撮影している. 時刻 0 における各人物の位置を円で示す. また, 十字の線は各人物の時刻 0 における初期位置を表す. 人物 (a) と (b) は矢印の方向に歩行しており, その Occupancy が時間と共に移動する様子が確認できる.

4.2.2 手順 2 : 軌跡作成

次に, Occupancy の存在する領域を通るように多数の軌跡を作成する (図 4.3(iii)). 具体的な手順を (図 4.4) を使って説明する. 図 4.4(i) は見易さのために図 4.3(ii) のクリップの Y 軸を省略したものである.

まず異なる 2 時刻で Occupancy の存在する領域を端点に持ち, かつ線分に沿った領域にも Occupancy が存在する様な線分をランダムに多数作成する (図 4.4(ii)). 赤の×印で示すような線分は, 線分の中で Occupancy の無い領域を通るため除去される. この線分は Occupancy の存在する領域を通る軌跡を作る際に, 軌跡が移動出来る方向の情報として利用される. 本論文ではこの線分を“フローベクトル”と呼ぶ事にする. フローベクトルを除去するかどうかの判定は, フローベクトルの始点から終端点に向かって, 時刻間隔 Δt 毎にその周辺の半径 0.15 m の範囲に Occupancy があるかどうかチェックし, Occupancy がない時刻が 1 カ所でもあれば, 除去する. 提案するシステムでは, 複数のカメラの視野が重複するように設置される (図 3.2). しかし, Occupancy Map は 4.1 節で説明したように, 個々の視点の奥行き情報に基づき独立に作成される. そのため提案手法では, 下記に述べる手順によって, 各視点の Occupancy Map を統合している. この手順はまた, カメラの撮影タイミングが非同期であることの困難性に対しても, 有効に働く.

時刻 t において, あるフローベクトルの周囲に Occupancy があるかどうかをチェックするとき, $t \pm \tau$ [sec] の範囲内で得られている Occupancy Map を対象とし, 下記の式で統合した Occupancy 値を使用する.

$$\overline{OM}_t(h) = \sum_{k=-\tau}^{\tau} OM_{t+k}(h) \quad (4.2)$$

ただし, $OM_t(h)$ は時刻 t で得られた Occupancy Map の, 位置 h における値である. しか

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

し式 4.2 は、視野範囲内の物体が、時間幅 τ に対して十分遅いか静止している場合にのみ有効であるという課題がある。図 4.5 に示すように、歩行程度の速度であっても Occupancy の位置が移動する様子が確認できる。従って、移動による位置ずれを補正するために、式 4.2 を変形した式 4.3 を導入する。

$$\overline{OM}_t(h) = \sum_{k=-\tau}^{\tau} OM_{t+k}(\hat{h}_{\mathbf{v}k}) \quad (4.3)$$

ただし、 \mathbf{v} はフローベクトルの速度、 $\hat{h}_{\mathbf{v}k}$ は位置 h を始点として、速度 \mathbf{v} で k の時間幅の間、等速直線移動した場合に到達する位置を表す。

τ は大きく設定すると、隠蔽やカメラの撮影タイミングで Occupancy が得られなかった時刻があってもフローベクトルが残りやすくなり、隠蔽に頑健な軌跡作成が出来るが、一方で誤った軌跡も作られやすくなる。小さくすると隠蔽やカメラの撮影タイミングの影響を受けやすくなり、正しい軌跡作成が出来なくなる。本論文ではこれらの定数を実験的に求め、 $\Delta t = 0.1 \text{ sec}$ 、 $\tau = 0.1 \text{ sec}$ とした。

もし [30, 36, 48, 16] のように撮影タイミングが同期されているならば、ある時刻 t における複数カメラの観測情報は、たとえ視点位置が異なっても、同一位置の人物の物であるので、式 4.2 で十分である。しかし提案システムのように撮影タイミングが同期されていない場合、チェックをする時刻 t で必ずしも Occupancy Map が得られているとは限らない上、Occupancy Map を統合するには各人物の移動速度を考慮して適正に補正しなければならない。フローベクトルは、この困難性の解決に有益である。

次に、時刻間隔 Δt ごとに XY 平面上に軌跡の始点を配置する (図 4.4(iii))。始点は $\Delta l[\text{cm}]$ 間隔の格子状に均一に配置される。本論文では、 Δt はフローベクトルのチェック時間間隔と同じく $\Delta t = 0.1 \text{ sec}$ 、 $\Delta l = 0.07 \text{ m}$ を用いる。

最後に、フローベクトルを使ってそれぞれの始点を延長することによって軌跡を作成する (図 4.4(iv))。Algorithm 1 は、時刻 t で定義される始点 $\mathbf{x}_t = (x_t, y_t)$ が与えられたときに、時刻間隔 Δt ごとに位置が定義される軌跡 $\mathbf{X} = \{\mathbf{x}_t, \mathbf{x}_{(t+\Delta t)}, \dots\}$ を生成する手順を示している。まず始点 \mathbf{x}_t の近傍 (半径 0.07 m 以内) にフローベクトルが存在するか確認 (2 行目) して、存在しなければ軌跡の作成を終了する。従って、はじめから始点周辺にフローベクトルが無い場合は、軌跡 \mathbf{X} が空集合のままなので、その始点から軌跡は作成されない。フローベクトルが存在する場合には始点 \mathbf{x}_t を軌跡 \mathbf{X} に追加する (3 行目)。次に図 4.4 中赤線で示すようにフローベクトル \mathbf{v} をランダムに 1 つ選び出し (4 行目)、次の時刻 $t + \Delta t$ での位置を計算する (5 行目)。この手順を、周辺にフローベクトルが存在しなくなるか、クリップの終端時刻に到達するまで繰り返す事で 1 つの軌跡が作成される。

4.2.3 手順 3 : 形状の類似性に基づく軌跡のクラスタリング

以上の手順は XYT 空間に均一に配置される始点すべてに適用されるので、実際の人数よりも多数の軌跡が作成される。そこで最後に、軌跡を形状に基づいてクラスタリン

Algorithm 2 Trajectory Clustering

Input: Trajectories $\mathbf{V} = \{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_{N-1}\}$, which is included in the clip defined within $[t_0, t_0 + T_B]$.

Output: Clusters $\{\mathbf{V}_0, \mathbf{V}_1, \dots, \mathbf{V}_{M-1}\}, \mathbf{V}_k \subseteq \mathbf{V}$

- 1: **for each** t in $\{t_0, t_0 + \Delta t, \dots, t_0 + T_B\}$ **do**
 - 2: Pick up points $\mathbf{P} = \{\mathbf{x}_t^p \mid p\text{-th trajectory } \mathbf{X}_p \text{ is defined at time } t\}$
 - 3: Assign labels $L(\mathbf{x}_t^p)$ to $\mathbf{x}_t^p \in \mathbf{P}$ by using CCL.
 - 4: **end for**
 - 5: Define a label sequence \mathbf{L}_k to k -th trajectory \mathbf{X}_k as $\mathbf{L}_k = \{L(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}_k\}$
 - 6: Define an adjacency list $\mathbf{E} = \{(\mathbf{X}_i, \mathbf{X}_j) \mid \mathbf{L}_i = \mathbf{L}_j (i \neq j)\}$
 - 7: Make clusters $\{\mathbf{V}_k \subseteq \mathbf{V} \mid \forall \mathbf{X}_i, \mathbf{X}_j \in \mathbf{V}_k, (\mathbf{X}_i, \mathbf{X}_j) \in \mathbf{E} \text{ and } \forall \mathbf{X}_i \in \mathbf{V}_k, \forall \mathbf{X}_j \in \overline{\mathbf{V}_k}, (\mathbf{X}_i, \mathbf{X}_j) \notin \mathbf{E}\}$
-

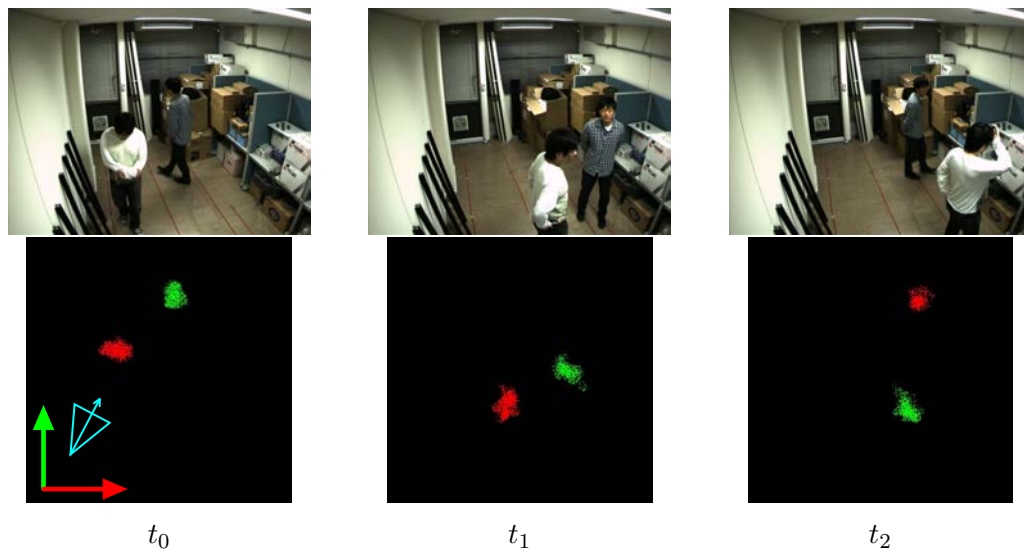
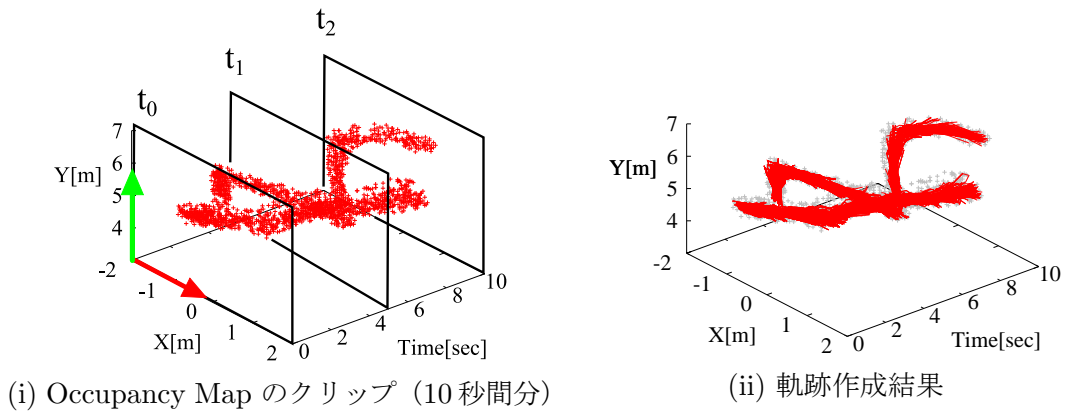
グする。同一人物に対応する軌跡の形状は類似しているので、クラスタリング後は実際の人数と同じ数のクラスタが得られると期待される。

クラスタリングの手順を Algorithm 2 に示す。手順は2段階に分かれている。まず始めに時刻間隔 Δt 毎に、その時刻に存在する軌跡の位置情報を使って Connected Component Labeling (CCL) を行う (1~4行目)。図 4.6 は2人の人物がすれ違う実際のシーン (データセット D_2) での軌跡作成過程を表した物で、図 4.6(iii) が図 4.6(i) 中の時刻 t_0, t_1, t_2 において CCL を適用した結果である。ある軌跡 i と軌跡 j の時刻 t での位置をそれぞれ $\mathbf{x}_t^i, \mathbf{x}_t^j$ とするとき、これらに同じラベルが割り当てられるかどうかはユークリッド距離 $\text{distance}(\mathbf{x}_t^i, \mathbf{x}_t^j)$ が閾値 $d_{threshold}$ 以下かどうかで判定する。すなわち、一般的な塗りつぶしと同様の結果が得られる。CCL によって \mathbf{x}_t^i に割り当てられたラベルを $L(\mathbf{x}_t^i)$ で表す。ある軌跡 \mathbf{X} に対して割り当てられたラベル列を $\mathbf{L} = \{L(\mathbf{x}_t), L(\mathbf{x}_{(t+\Delta t)}), \dots, L(\mathbf{x}_{(t+n\Delta t)})\}$ とする。なお、CCL は時刻毎に独立して適用されるので、必ずしも時間を通して同一のラベルが軌跡に割り当てられるとは限らず¹、CCL の結果は追跡結果としては利用する事は出来ない。

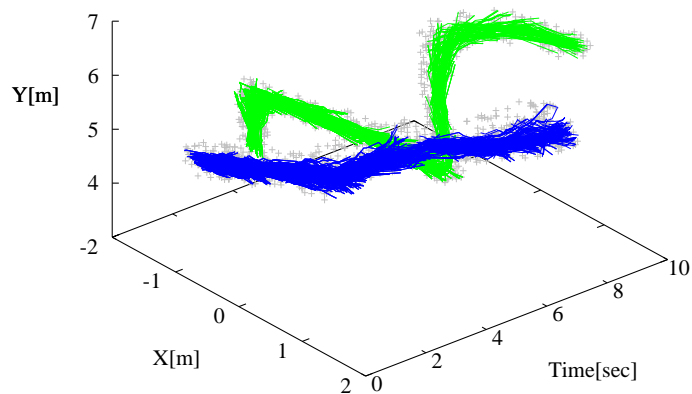
そこで2段階目の手順として、ラベル列を使って類似形状の軌跡をクラスタリングする。具体的には、任意の軌跡 \mathbf{X}_i と軌跡 \mathbf{X}_j の類似性は、ラベル列 $\mathbf{L}_i, \mathbf{L}_j$ が共通に定義される時刻範囲内で同一であると定義する。これによって、類似形状の軌跡から成るクラスタ \mathbf{V}_k を作成する (5~7行目)。最終的に、同一クラスタに属す軌跡に固有の ID を割り当てて追跡結果とする。データセット D_2 に対するクラスタリング結果 (図 4.6(iv)) に示すように、人数と同じく2つのクラスタに軌跡が分類されている事が分かる。

¹ある人物の動きを表す軌跡のラベル列 \mathbf{L} について、異なる2時刻 $\alpha, \beta (\alpha \neq \beta)$ に対して、一般に $L(\mathbf{x}_\alpha) \neq L(\mathbf{x}_\beta)$ である。

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影



(iii) CCL によるラベリング結果



(iv) クラスタリング結果

図 4.6: 2 人の人物のデータセット D_2 の軌跡作成およびクラスタリング過程

Algorithm 3 Calculating Possible Patterns

Input: Clusters $\Phi = \{\mathbf{V}_0, \mathbf{V}_1, \dots, \mathbf{V}_{N-1}\}$

Output: Patterns $\mathbf{P} = \{\Phi_0, \Phi_1, \dots, \Phi_{M-1}\}, \Phi_k \subseteq \Phi$

1: **Start**

2: Make patterns $\{\Phi_k \subseteq \Phi \mid \forall \mathbf{V}_i, \mathbf{V}_j (i \neq j) \in \Phi_k, \text{DISTANCEBETWEENCLUSTERS}(\mathbf{V}_i, \mathbf{V}_j) > 0 \text{ and } \forall \mathbf{V}_i \in \Phi_k, \exists \mathbf{V}_j \in \Phi_k, \text{s.t. DISTANCEBETWEENCLUSTERS}(\mathbf{V}_i, \mathbf{V}_j) = 0\}$

3: **End**

4: **function** DISTANCEBETWEENCLUSTERS($\mathbf{V}_1, \mathbf{V}_2$)

Suppose clusters \mathbf{V}_1 and \mathbf{V}_2 are defined within $[t_0, t_0 + T_B]$.

$\mathbf{V}(t) := \{\mathbf{x}_t \mid \mathbf{x}_t \in \mathbf{X} \subseteq \mathbf{V} \text{ is the position of the trajectory } \mathbf{X} \text{ at time } t\}$

5: **return** $\min_{t_0 \leq t \leq t_0 + T_B} \left\{ \min_{\mathbf{x}_t^1 \in \mathbf{V}_1(t), \mathbf{x}_t^2 \in \mathbf{V}_2(t)} (\text{distance}(\mathbf{x}_t^1, \mathbf{x}_t^2)) \right\}$

6: **end function**

4.3 反復法によらない軌跡の最適化

本節では、4.2節で求めた軌跡の最適化手法を導入する。図4.6に示したように、これまで述べた手順によって Occupancy Map のクリップに当てはまる軌跡を生成しクラスタリングする事が出来るが、実際には誤った軌跡やクラスタが生成される場合がある。

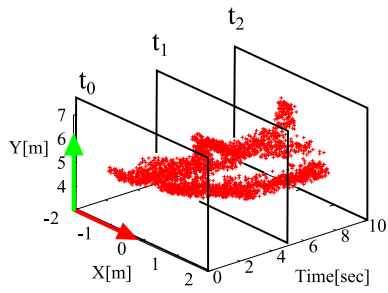
1つ目のケースは、人物同士が近接した場合に誤った軌跡・クラスタが生成され、正しい軌跡・クラスタに干渉してしまう場合がある事である。4.3.1節ではこのような誤ったクラスタを除去する手法を提案する。

2つ目のケースは、ある人物が隠蔽によってどの視点からも見えなくなった場合に、軌跡の途切れが発生する場合がある事である。軌跡の途切れは人物IDを維持する上で問題となる。4.3.2節では途切れを修復する手法について述べる。

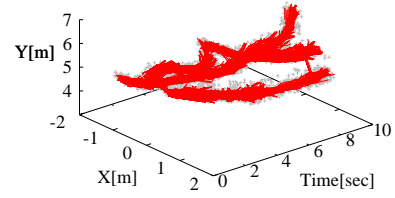
4.3.1 誤ったクラスタの除去

図4.7は、3人の人物のデータセット (D_3) に対して4.2節のアルゴリズムを適用した結果である。クラスタリング結果 (図4.7) には、図4.7(iv)に表すように5つのクラスタが含まれる結果となり、実際の数よりも多くのクラスタが生成されている。これは人物が近接した際に、誤って別の人物に対応する軌跡が生成された事が原因である。図4.7(iii)はクリップ中の時刻 t_0, t_1, t_2 でのカメラ視点画像とCCLによるラベリングの結果を示している。時刻 t_1 において、左側と中央の人物が接近している。このときのラベリング結果を見ると、赤色で示す同一のラベルが割り当てられており、このときに誤った軌跡が作成されてクラスタ5を生成するに至っている。同様の理由でクラスタ4も誤ったクラスタである。従ってクラスタ4,5は追跡結果から除去されるべきだが、実際には

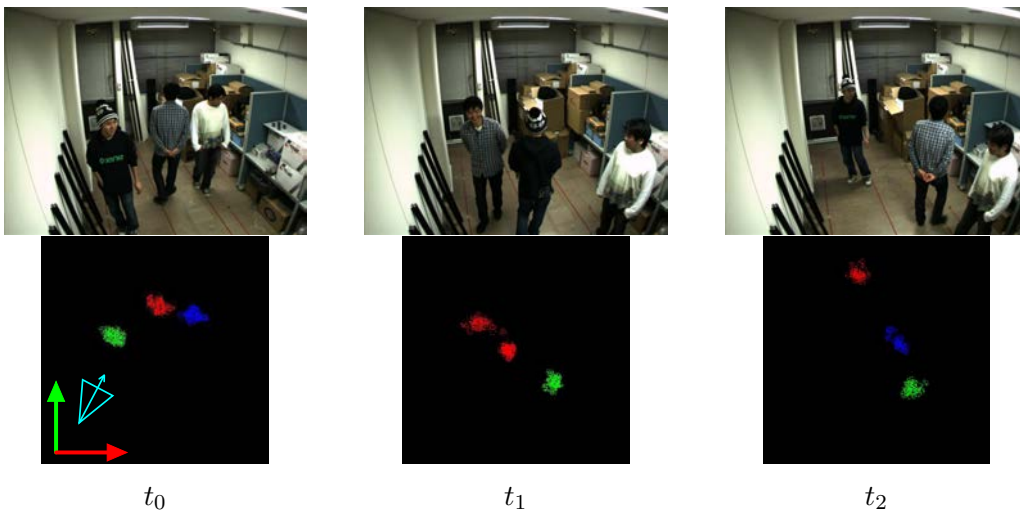
第4章 反復法によらない軌跡の最適化と各視点画像への逆投影



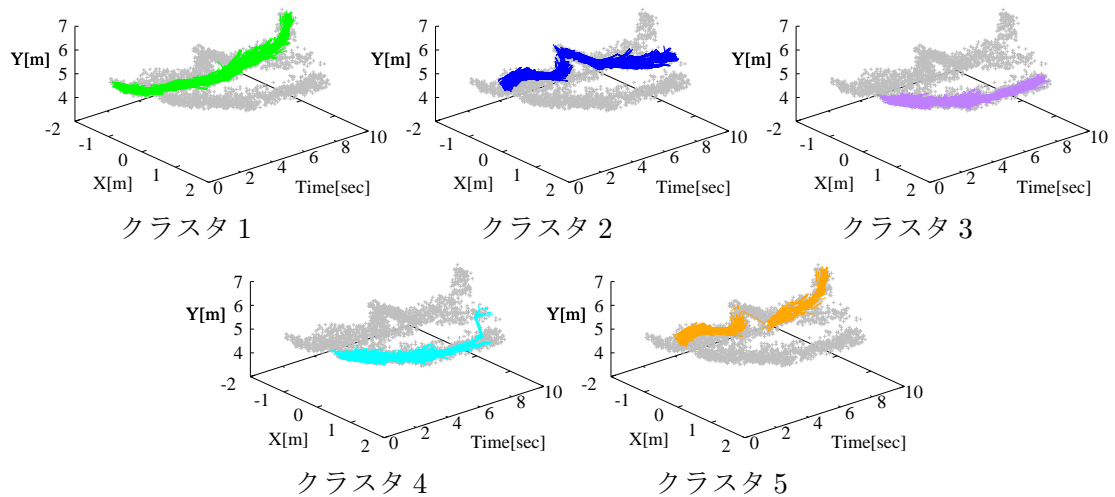
(i) Occupancy Map のクリップ (10 秒間分)



(ii) 軌跡作成結果



(iii) CCL によるラベリング結果



(iv) クラスタリング結果

図 4.7: 3 人の人物のデータセット D_3 の軌跡作成およびクラスタリング過程

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

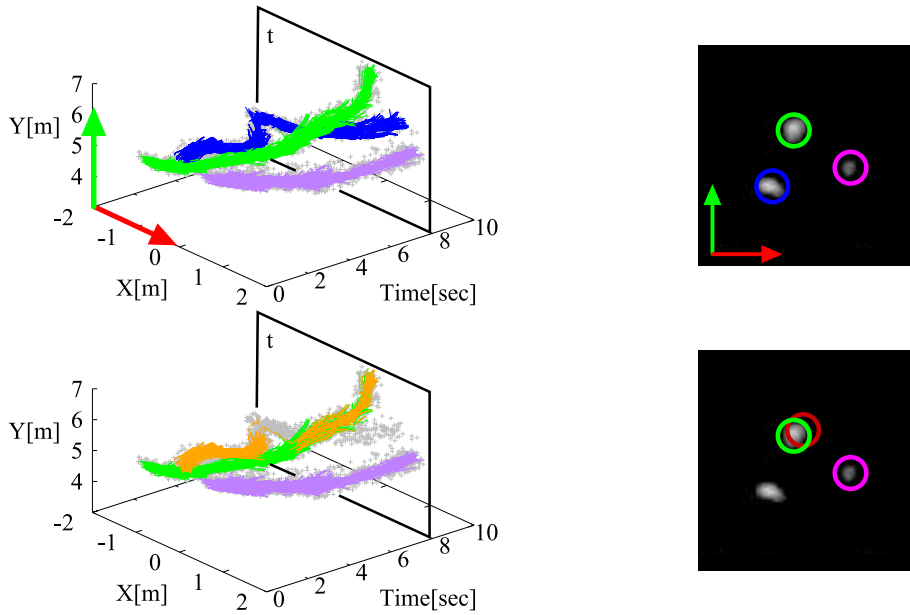


図 4.8: 干渉しないクラスタから成るパターン（上）と干渉するクラスタから成るパターン（下）の例. 右図は時刻 t でのクラスタの位置を, Occupancy Map 上に重ね合わせたもの.

Occupancy Map に正しく当てはまっているので, 単純に形状などクラスタ単独の特徴のみで判断する事は出来ない.

そこで, 本研究では「干渉しないクラスタ」のパターンを全て列挙して, その中から最適な追跡結果を選び出す手法を提案する. クラスタが干渉するとは, ある時刻で同一の領域を複数のクラスタが占める場合を指すとする. 「干渉しないクラスタ」と「干渉するクラスタ」の例を図 4.8 に示す. 図 4.8 下のパターンはクラスタ 1,3,5 から成り, 矩形で示した時刻でクラスタ 1,5 が干渉している. 右図はこの時刻での Occupancy Map とクラスタの位置を描いた物であり, クラスタ 1,5 が同一人物の領域に対応し, さらに左下の人物領域は追跡出来ておらず, 誤った追跡結果であると言える. 従って, 図 4.8 上のような「干渉しないクラスタ」から成るパターンを選び出せば, このような誤追跡を防ぐ事ができる. その手順を Algorithm 3 に示す. 一般に, 「干渉しないクラスタ」のパターンは図 4.9 に示すように複数得られる. この場合, 我々は Occupancy のある領域をクラスタが最も占めているパターンを, 最適パターン Φ_{best} であると見なす事にした.

$$\Phi_{best} = \operatorname{argmax}_{\Phi_k \in \mathbf{P}} \sum_{t=t_0}^{t_0+T_B} S_t(\Phi_k) \quad (4.4)$$

ただし, $S_t(\Phi_k)$ はパターン Φ_k が時刻 t において占有している床平面上の面積 (4.1 節で導入した床平面上のグリッドの数と等価) を表す. 図 4.9 の例では, パターン 1 が最適パターン Φ_{best} である. その他のパターンは, Occupancy があるにも拘らず, その部分を

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

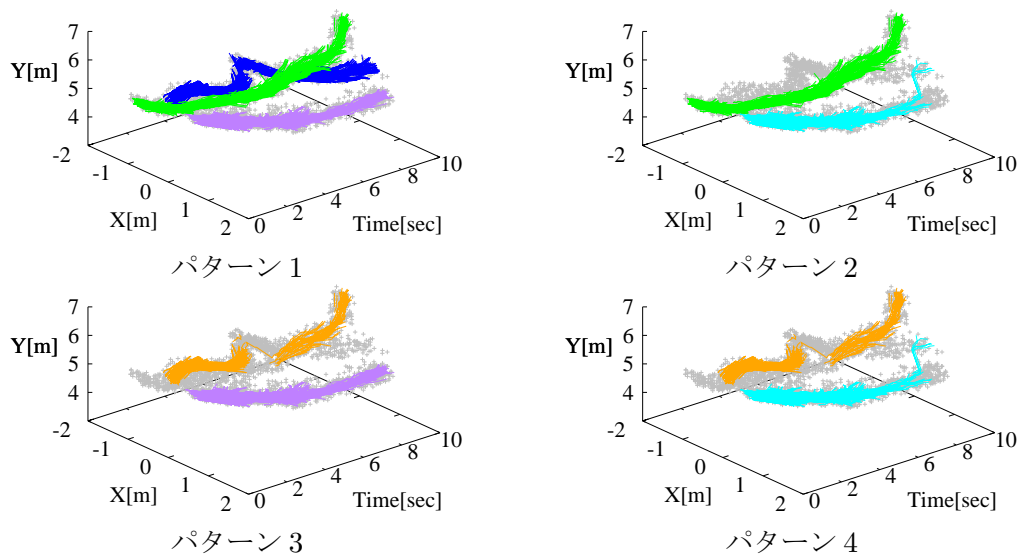


図 4.9: 3人の人物のデータセット D_3 (図 4.7) における干渉しないクラスタから成るパターン. この例ではパターン1に含まれる軌跡が, Occupancy の得られている領域を最も占有するため, 最適解として選ばれる.

クラスタが占めていない領域があり, これは人物が観測されているがトラッカーが割り当てられていないことに対応するので, 最適解ではないと考える. パターン1では, クラスタ 4,5 が除去され, Occupancy のある領域がクラスタ 1,2,3 によって占められおり, 正しい追跡結果であると言える.

4.3.2 途切れた軌跡の修復

図 4.10(ii) は 5人の人物のデータセット (D_5) のクリップのクラスタリング結果である. 見やすさのために各クラスタを平均して1本の軌跡で表している. これに 4.3.1 節で述べた「誤ったクラスタの除去」を行った最適解が図 4.10(iii) であるが, 赤丸で示すように2カ所で軌跡の途切れが発生していることがわかる. これは, ある人物がどの視点からも観測出来ない部分で, Occupancy が十分に得られないために, 軌跡の延長が出来ないことが原因である. 軌跡の途切れが起こると, 人物 ID が途中で変化してしまう.

軌跡の途切れを防ぐために, ある2つのクラスタ V_i, V_j の間に間隙があり, その時間長が Δt_c 未満である場合, その2つのクラスタの接続を試みる. 複数の接続方法がある場合は, 全ての接続方法を列挙し, 軌跡が最も滑らくなる接続方法で接続する. Δt_c は大きい程, 少ない Occupancy に対しても途切れの少ない軌跡を生成出来るが, 誤った接続²を行う恐れがあるとともに, 多くの接続方法が生じて計算時間が大きくなる可能性もある. 本研究では, 実験的に $\Delta t_c = 3 \text{ sec}$ と定めた.

²例えば, 一度退室した人物と, その後新たに入室した人物に対して同一の ID を割り当ててしまう場合が考えられる.

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

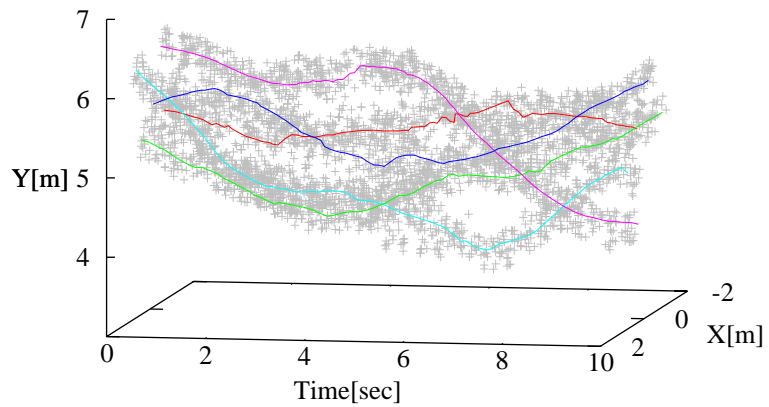
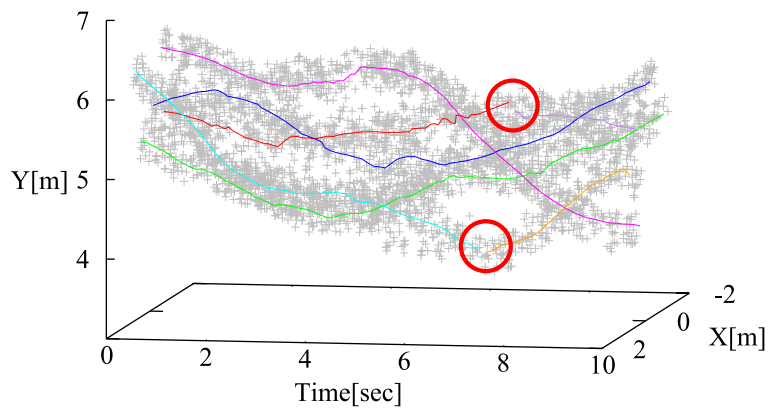
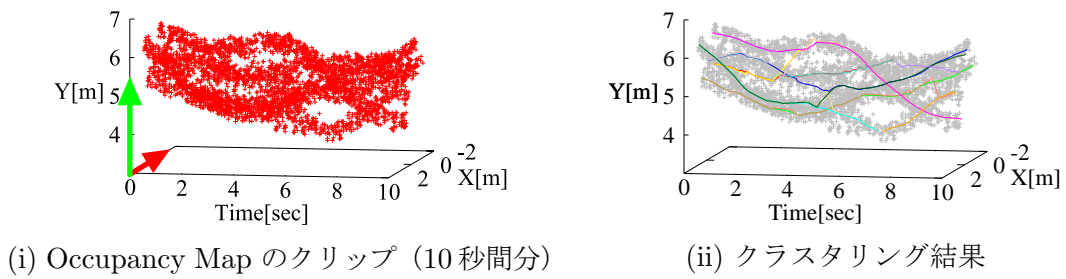


図 4.10: 5 人の人物のデータセット D_5 の最適化過程. (ii) 内のクラスタは, 見やすさのためクラスタに含まれる軌跡の時刻毎の平均位置を描画した.

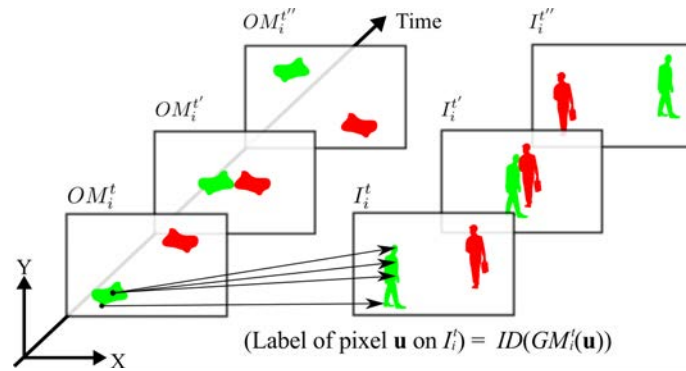


図 4.11: カメラ視点画像 I_i^t 上での領域分割例

接続の結果を図 4.10(iv) に示す. 2カ所の間隙が適切に接続され, 正しく 5 人分の軌跡が得られている事が分かる.

4.4 隣接クリップ間での軌跡の ID マッチング

人物追跡処理の最後のステップとして, 時間を通して軌跡の ID を維持するために, 隣接クリップ間, 即ち 4.3 節での処理の結果得られたクリップと, 直前に計算したクリップの間で, 軌跡の ID のマッチングを行う. マッチングは, 隣接クリップに含まれる軌跡の形状の類似性を評価することによって行う. 任意の 2 軌跡の形状の類似性は, 隣接クリップの重複期間 T_A における, 2 軌跡の平均距離によって定義した. 本提案システムでは, 軌跡の平均距離が閾値 d_{match} よりも小さいときに, 隣接クリップ間の軌跡を対応付ける. 本論文では $d_{match} = 5 \text{ cm}$ とした. もし現在のクリップのある軌跡が, 直前のクリップ中のどの軌跡とも対応付けられないとき, 提案システムはその軌跡に新規の ID を割り当てる.

4.5 カメラ視点画像上での領域分割

本節では, 4.1 節で導入した Geometry Map と, 4.2 節, 4.3 節および 4.4 節の追跡結果を使って, カメラ画像上での領域分割を行う手法を提案する.

追跡結果に含まれるクラスタは, Occupancy のある領域を通る軌跡の集合であることから, 各クラスタのある時刻での断面は平面図上での人物形状になっている. この事を利用して, まず平面図上において領域分割を行う. 具体的には, カメラ i の時刻 t における Occupancy Map (OM_i^t) 上で, グリッド h が Occupancy を持つとき ($OM_i^t(h) > 0$ のとき), 近傍の軌跡の ID をグリッド h に割り当てる (図 4.11 左). グリッド h に割り当てられた ID を $ID(h)$ と表す事にする. 次に, カメラ i の時刻 t における Geometry Map

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影

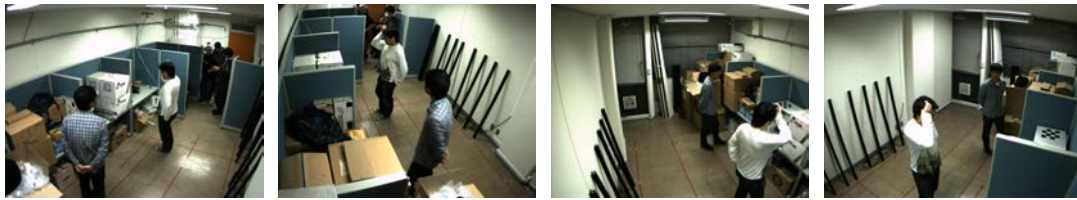
(GM_i^t) を使って, OM_i^t 上の領域をカメラ i の時刻 t におけるカメラ視点画像 I_i^t 上に対応付ける. すなわち, $GM_i^t(\mathbf{u}) = h$ であるとき, カメラ視点画像 I_i^t 上のピクセル \mathbf{u} に対して $ID(h)$ を割り当てる. 一般に, あるグリッド h に対して複数のピクセルが対応するが, その全てのピクセルに同一の $ID(h)$ を割り当てる (図 4.11 右). この処理によって, カメラ視点画像上での領域分割が完了する.

図 4.12 に図 4.6 で用いたデータセット D_2 のクリップの時刻 t_2 における追跡結果 (図 4.12(ii)) と, 平面図上での領域分割結果 (図 4.12(iii)) を示す. 平面図上での人物領域の大きさは人物の大きさに依存し, また円型ではなく楕円型なので体の向きによっても形状が変化する. 図から分かるように, 人物の形状に対して追跡ウィンドウが追従しており, 平面図上での領域分割が正しく行えている. 平面図上の領域分割結果を全てのカメラ視点画像上に投影した結果を図 4.12(iv) に示す. 2 人の人物の領域が全ての視点で得られている. しかしながら, 左上の視点では青色の人物の左腕と背中領域が欠けてしまっている事が分かる. 主な原因としてカメラキャリブレーションの誤差が考えられ, これについては 5 章にて考察する.

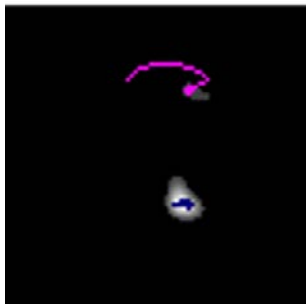
この手法を従来の平面図上での追跡手法に適用して領域分割する場合の問題点を述べる. Darrell ら [12] などの従来手法は, 固定形状の追跡ウィンドウを利用しているため, 人物領域の形状変化に追従出来なかった. 従って, 追跡ウィンドウのサイズが追跡対象の人物より小さい場合, ウィンドウ外の領域が領域分割に反映されず, 一方, 追跡ウィンドウを大きくとると, 人物同士が近接した場合に追跡ウィンドウが重複して正しく領域分割されないという問題が起こり得る. 図 4.13 は 5 人の人物のデータセット (D_5) の平面図上のセグメンテーション結果 (左) と, 従来手法との比較のために固定サイズの円形ウィンドウを当てはめた場合 (右) を表している. 固定サイズの円形ウィンドウでは重複が発生してしまうが, 提案手法では人物領域が適切に変形して近接した人物を正しく領域分割出来ている.

ジェスチャ認識等の応用例では, 大きな人物形状変化を捉える必要がある. 本節で述べた領域分割手法は, 平面図上での人物領域の任意の形状変化に追従する事が出来るため, そのような応用に非常に適している. 6 章では, 実際に挙手動作を検出するプロトタイプシステムを開発して混雑環境下でも正しく挙手を検知出来ることを確認し, ジェスチャ認識への応用の可能性を示す.

第4章 反復法によらない軌跡の最適化と各視点画像への逆投影



(i) カメラ視点画像



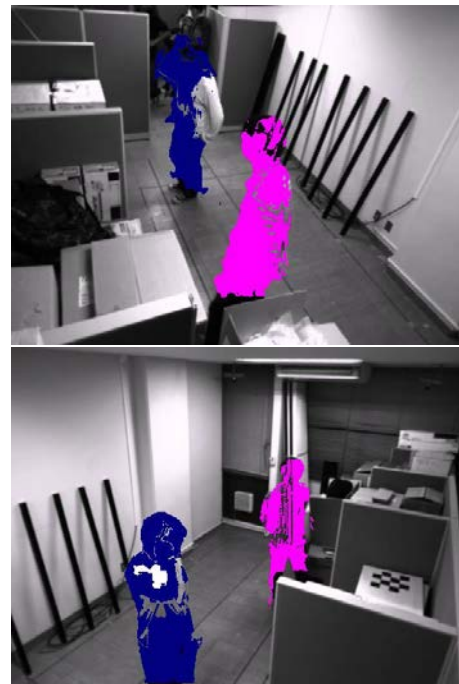
(ii) 追跡結果

曲線は過去2秒分の人物動線を表す



(ii) 平面図上での領域分割結果

Occupancy Map に重ねて描画している



(iv) カメラ視点画像上での領域分割結果

図 4.12: 図 4.6 の時刻 t_2 における追跡および領域分割結果



図 4.13: 5人の人物のデータセット D_5 で平面図上の領域分割を行った例（左）と，比較のために固定サイズの円形ウィンドウを当てはめて描画した例（右）．円形ウィンドウのサイズは Occupancy を含む必要最小限の大きさに設定したが，人物が近接している場合はウィンドウが重複する．なお，右下の人物の Occupancy Map が得られていないのは，当該 Occupancy Map が図に示すカメラ視点から作成されたもので，隠蔽による為である．

第5章 計算機実験

本章では、実験室内の一定の範囲内を1~6人の人物がランダムウォークするデータセットを使って、追跡手法がどの程度の混雑度まで正しく追跡出来るのかを調べる。また、追跡処理を行うために利用するOccupancy Mapのクリップの時間長 T_B を変化させて、追跡エラーがどのように変化するかも考察する。さらに、計算時間および通信量を計測して実用性の評価も行う。

5.1 実験に用いるデータセット

本実験では、3章で示した実験室(図3.2)で、1~6人の人物が指定された追跡領域内をランダムウォークするデータセット $D_1 \sim D_6$ を利用する。全てのデータセットは4台のステレオカメラを用いて撮影された。追跡領域の大きさは $2.4 \text{ m} \times 2.4 \text{ m}$ で、被験者は入退室時を除いて、撮影中は追跡領域外に出ないように指示されている。

まず1人ずつ追跡領域に入り、4分程度ランダムウォークを行った後、1人ずつ追跡領域外に出る。システムは自動的に人物の入退室を検出して、IDの割り当てと解除を行う。データセットの正確な長さとおよび入退室のタイミングを表5.1に記す。

5.2 実験結果

データセット D_1, D_2 については1つの時刻、データセット $D_3 \sim D_6$ については3つの異なる時刻での人物追跡と領域分割の結果を図5.1~図5.14に示す。対応する時刻は表5.1に記載した。

データセット D_1 (図5.1)では隠蔽や人物の近接の影響が無く、データセット D_2 (図5.2)でも同様にそれらの影響はほぼ無いため、問題無く追跡と領域分割が出来ている。

データセット D_3 では隠蔽や人物の近接が起こり始めるが、例えば図5.4(ii)および(iii)に示すように平面図上での人物追跡と領域分割に成功しており、カメラ視点画像上での領域分割(図5.4(iv))も正しく行えている事が分かる。

データセット D_4 では、図5.6(i)に示すOccupancy Mapに示すように、どの視点からも3人分のOccupancyしか観測できておらず、隠蔽の影響が顕著に表れ始める。しかしながら、提案手法は複数の視点のOccupancy Mapを利用することで誤り無く人物追跡が可能で、領域分割も正しく行えている。特に、図5.6(iv)の右上の視点の領域分割結果において、ピンク色の人物領域が手前の人物による隠蔽で2つに分断されているにも拘

第5章 計算機実験

表 5.1: 各データセットのタイムライン

データセット D_1		データセット D_2	
時刻 [sec]	イベント	時刻 [sec]	イベント
0	1 人目の入室	0	1 人目の入室
107(= t_0)	1 人の人物の追跡中 (図 5.1)	5	2 人全員の入室完了
		145(= t_0)	2 人の人物の追跡中 (図 5.2)
		285	1 人目の退室
214	全員の退室完了	288	全員の退室完了

データセット D_3		データセット D_4	
時刻 [sec]	イベント	時刻 [sec]	イベント
0	1 人目の入室	0	1 人目の入室
5(= t_0)	3 人全員の入室完了 (図 5.3)	8(= t_0)	4 人全員の入室完了 (図 5.6)
91(= t_1)	3 人の人物の追跡中 (図 5.4)	94(= t_1)	4 人の人物の追跡中 (図 5.7)
188(= t_2)	3 人の人物の追跡中 (図 5.5)	175(= t_2)	4 人の人物の追跡中 (図 5.8)
271	1 人目の退室	255	1 人目の退室
275	全員の退室完了	265	全員の退室完了

データセット D_5		データセット D_6	
時刻 [sec]	イベント	時刻 [sec]	イベント
0	1 人目の入室	0	1 人目の入室
10(= t_0)	5 人全員の入室完了 (図 5.9)	13(= t_0)	6 人全員の入室完了 (図 5.12)
98(= t_1)	5 人の人物の追跡中 (図 5.10)	27(= t_1)	6 人の人物の追跡中 (図 5.13)
183(= t_2)	5 人の人物の追跡中 (図 5.11)	183(= t_2)	6 人の人物の追跡中 (図 5.14)
266	1 人目の退室	265	1 人目の退室
277	全員の退室完了	276	全員の退室完了

第5章 計算機実験

らず、同一のIDの割り当てが出来ている。一般に領域分割は、近接するピクセルや領域を、色やテクスチャ等の類似性に基づいて統合していく為に、ある人物の領域が別の人物や障害物による隠蔽で分断されてしまった場合、それらを統合する事は困難である。しかし、提案手法では最初に人物追跡を行って、平面図上での領域をカメラ画像上に対応させる為、カメラ視点画像上での領域の分断に極めて頑健である。

データセット D_5 では、隠蔽と人物の近接が頻繁に発生する。例えば、図 5.11(i) に示すカメラ視点画像と Occupancy Map では、隠蔽と人物の近接が同時に発生している。Occupancy Map は色や外観の情報を持たないが、提案手法は時系列で Occupancy Map を分析する事で、このような困難な状況でも追跡に成功している。その一方で、図 5.11(iv) の領域分割結果の左上の視点のように、マゼンタ色の領域の一部に黄土色の領域が被さっており、領域分割に失敗してしまっている。領域分割エラーの定量的な評価については、5.4 節で論ずる。

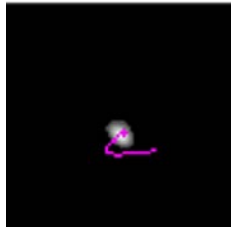
データセット D_6 では、隠蔽と人物の近接の影響が最も大きくなり、どの視点からも観測出来ない人物が現れ始める。提案手法は Occupancy の観測された部分についてのみ軌跡を作成するので、このような状況では追跡に失敗してしまう。そのため 4.3.2 節でこのような場合に発生する軌跡の途切れを修復する処理を提案したが、図 5.13 に示す時刻 t_1 においては集団の中心にいる人物の追跡に失敗してしまっている。また、図 5.13(iv) の赤色の領域とシアン色の領域の間のように、領域分割エラーもデータセット D_5 に比べて多く見られる。人物追跡エラーと、最適化処理の効果については、それぞれ 5.3 節と 5.5 節で論ずる。

また、領域分割に関しては、完全に密な領域にはならない。これは、輝度と奥行きを単純に組み合わせた背景差分を行っているためで、背景画像の輝度に近い部分が前景として得られないこと、またテクスチャの乏しい部分で奥行きが計測できないことに因る。領域膨張・縮小で軽減されるが、より根本的には、色情報を使って背景差分の精度を高めたり [19]、グラフカット [6] などの手法で前景抽出の質を改善したり、赤外線パターンを照射する RGB-D カメラ (2.1.2 節) を利用することなどが考えられる。

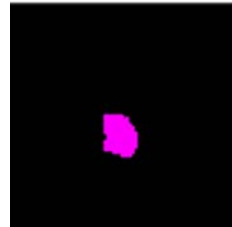
第5章 計算機実験



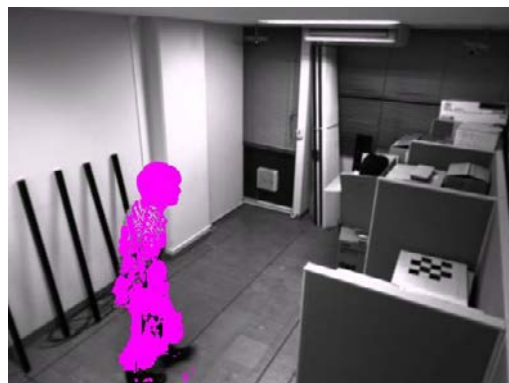
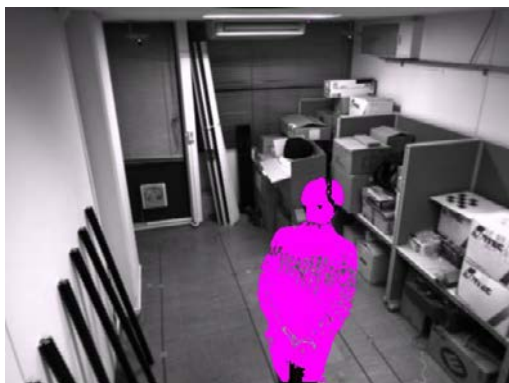
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



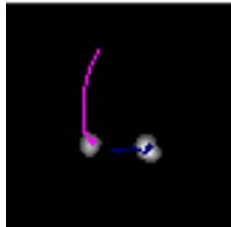
(iv) カメラ視点画像上での領域分割結果

図 5.1: データセット D_1 , 時刻 t_0

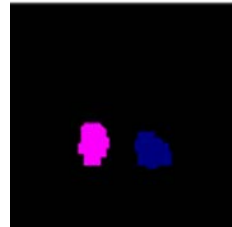
第5章 計算機実験



(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



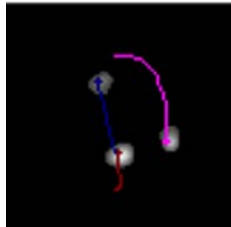
(iv) カメラ視点画像上での領域分割結果

図 5.2: データセット D_2 , 時刻 t_0

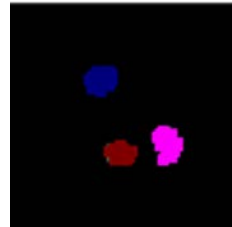
第5章 計算機実験



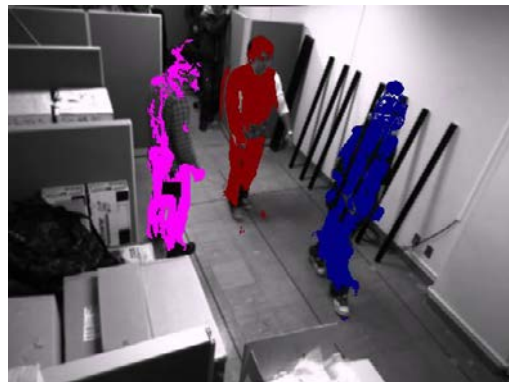
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



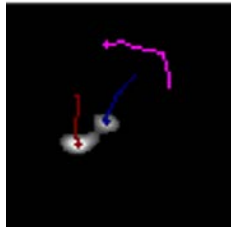
(iv) カメラ視点画像上での領域分割結果

図 5.3: データセット D_3 , 時刻 t_0

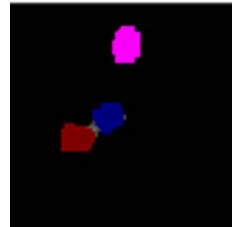
第5章 計算機実験



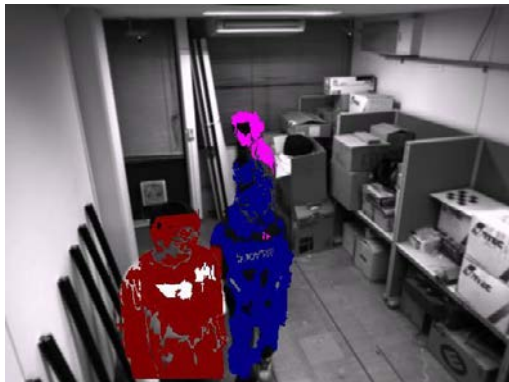
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



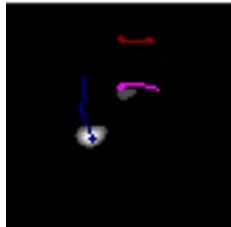
(iv) カメラ視点画像上での領域分割結果

図 5.4: データセット D_3 , 時刻 t_1

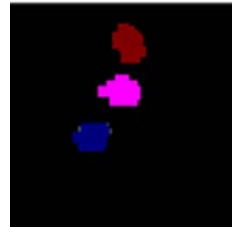
第5章 計算機実験



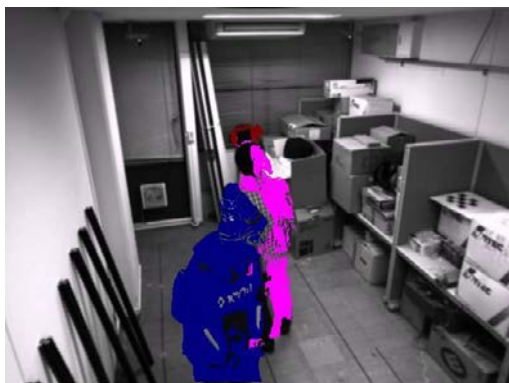
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



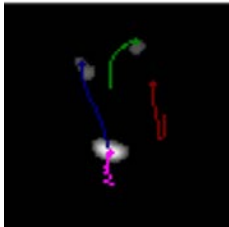
(iv) カメラ視点画像上での領域分割結果

図 5.5: データセット D_3 , 時刻 t_2

第5章 計算機実験



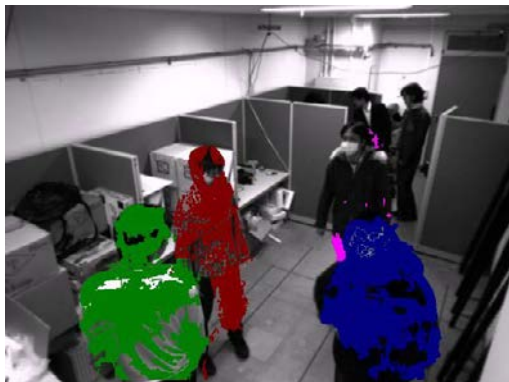
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



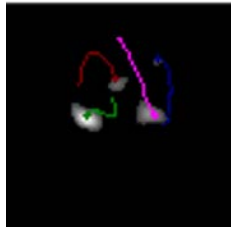
(iv) カメラ視点画像上での領域分割結果

図 5.6: データセット D_4 , 時刻 t_0

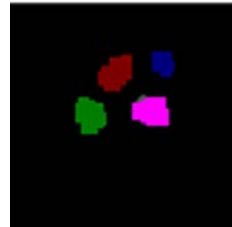
第5章 計算機実験



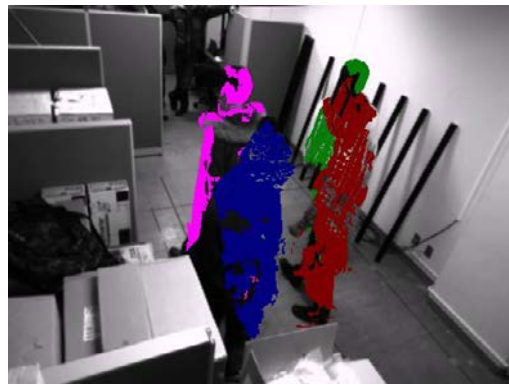
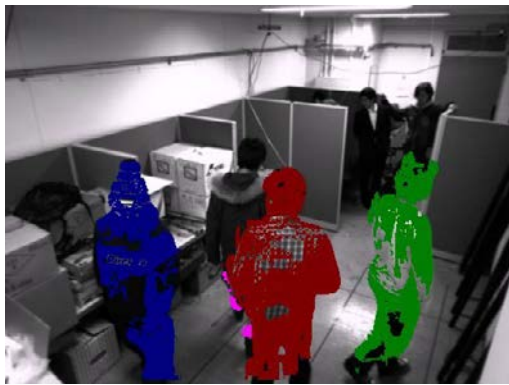
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



(iv) カメラ視点画像上での領域分割結果

図 5.7: データセット D_4 , 時刻 t_1

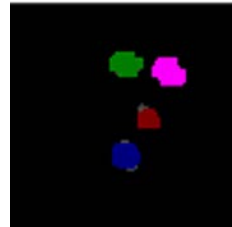
第5章 計算機実験



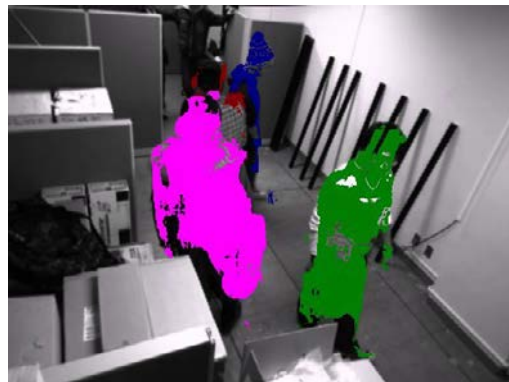
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



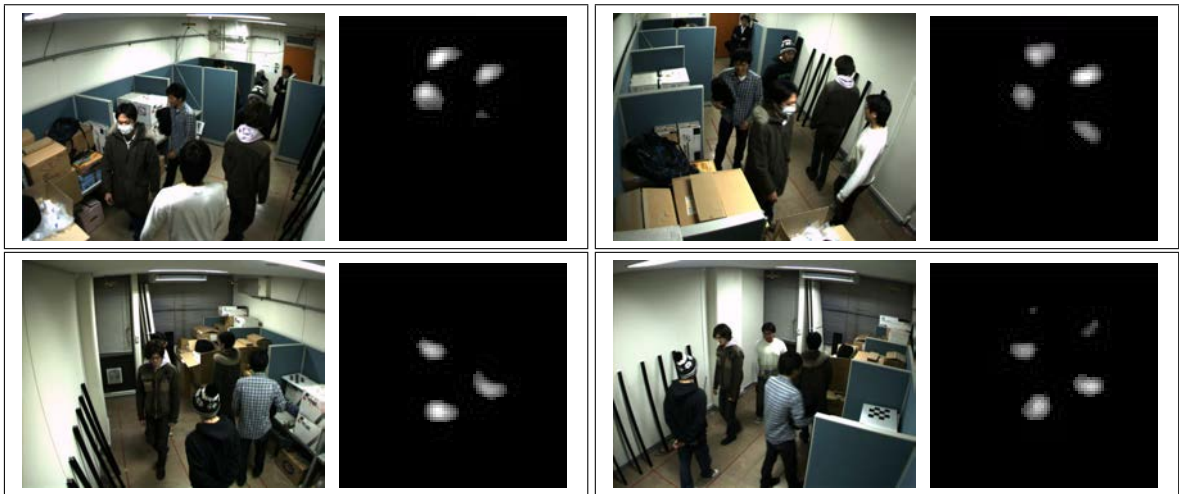
(iii) 平面図上での領域分割結果



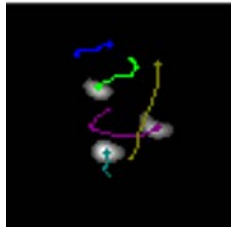
(iv) カメラ視点画像上での領域分割結果

図 5.8: データセット D_4 , 時刻 t_2

第5章 計算機実験



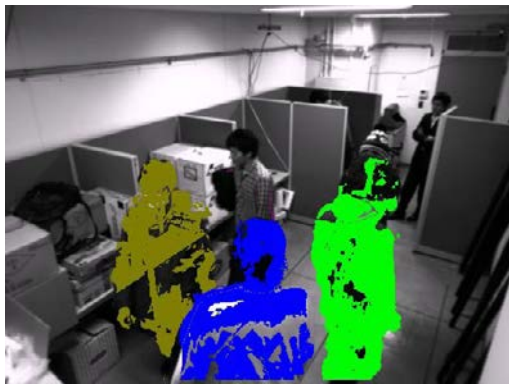
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



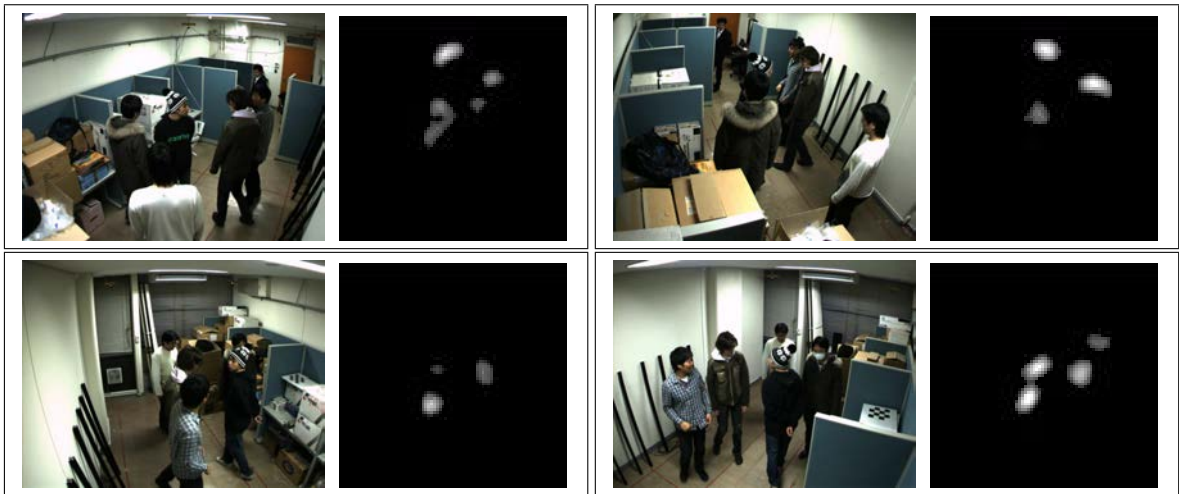
(iii) 平面図上での領域分割結果



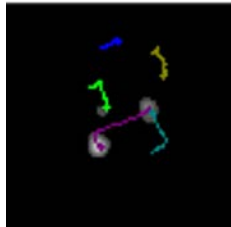
(iv) カメラ視点画像上での領域分割結果

図 5.9: データセット D_5 , 時刻 t_0

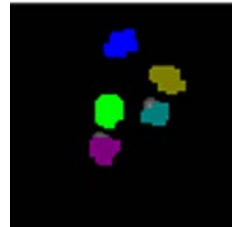
第5章 計算機実験



(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



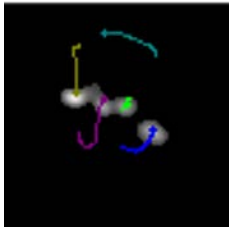
(iv) カメラ視点画像上での領域分割結果

図 5.10: データセット D_5 , 時刻 t_1

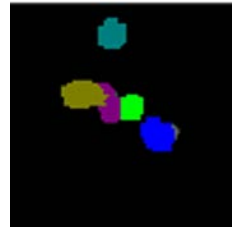
第5章 計算機実験



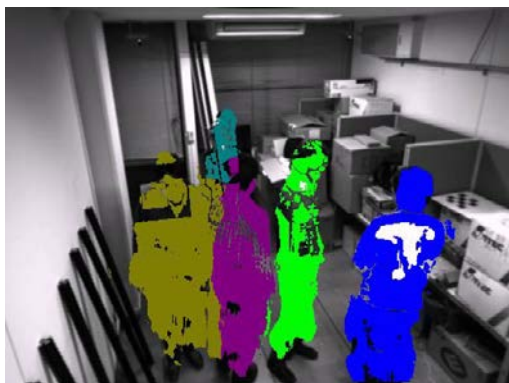
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



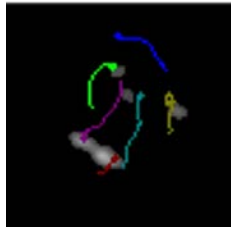
(iv) カメラ視点画像上での領域分割結果

図 5.11: データセット D_5 , 時刻 t_2

第5章 計算機実験



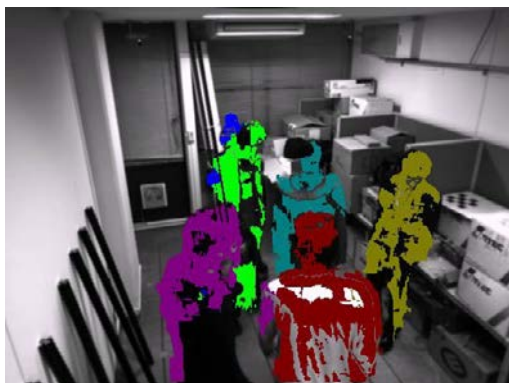
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



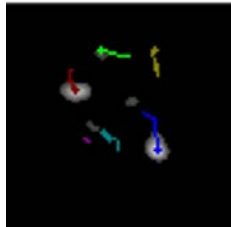
(iv) カメラ視点画像上での領域分割結果

図 5.12: データセット D_6 , 時刻 t_0

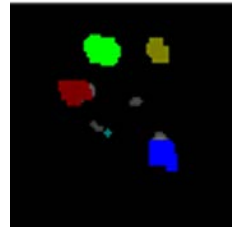
第5章 計算機実験



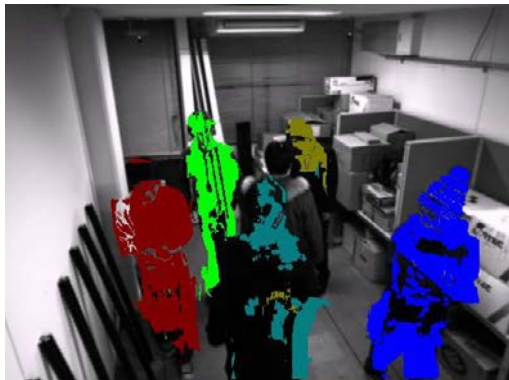
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



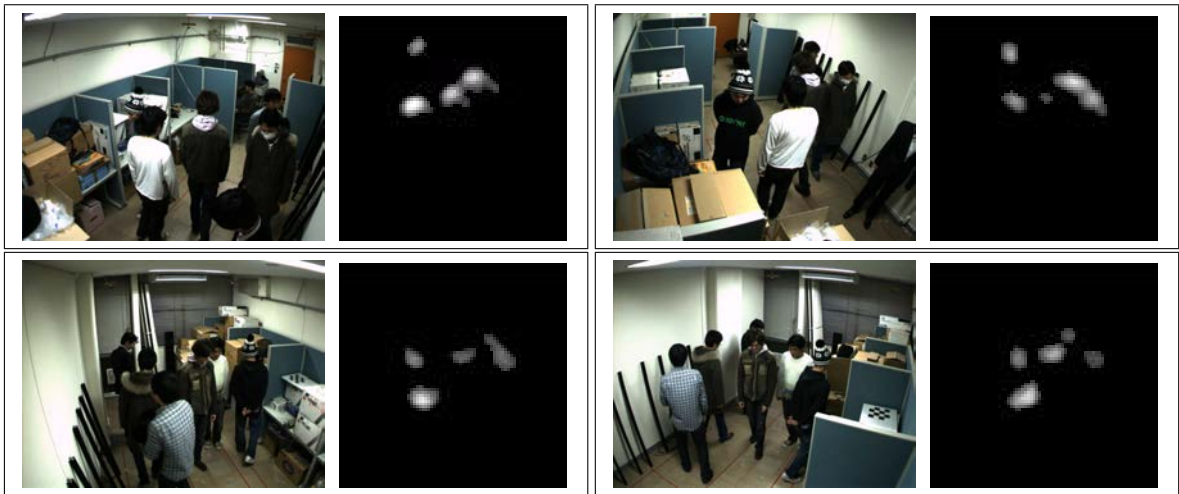
(iii) 平面図上での領域分割結果



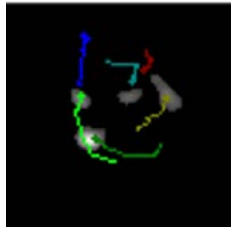
(iv) カメラ視点画像上での領域分割結果

図 5.13: データセット D_6 , 時刻 t_1

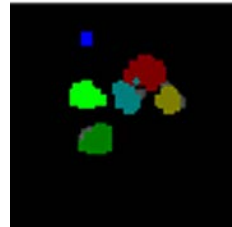
第5章 計算機実験



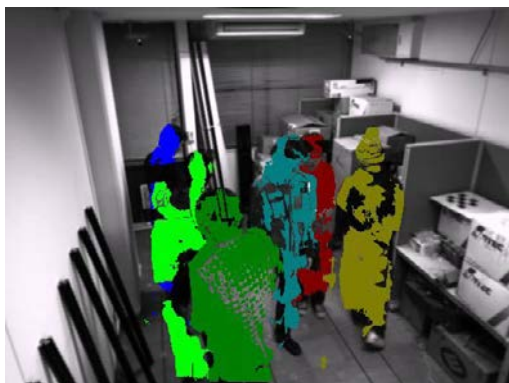
(i) カメラ視点画像と Occupancy Map



(ii) 平面図上での追跡結果



(iii) 平面図上での領域分割結果



(iv) カメラ視点画像上での領域分割結果

図 5.14: データセット D_6 , 時刻 t_2

5.3 平面図上での人物追跡のエラー評価

本節では、4.2節、4.3節および4.4節で提案した、平面図上での人物追跡手法がどの程度の混雑度まで間違えずに追跡出来るのかを明らかにする事を目的とし、データセット $D_1 \sim D_6$ で発生した追跡エラーの定量評価を行う。

本論文では、平面図上での人物追跡に成功している状態を以下のように考える

- クラスタに含まれる軌跡のある時刻での平均位置が、実際の人物領域内にあること。評価における人物領域は、目視によって人為的に定めた。
- 追跡領域内にいる全ての人物に ID が割当てられ、かつその ID が変化しないこと。

5.3.1 追跡エラーの分類

実験の結果認められた追跡エラーを以下の2つに分類し、データセット毎に発生頻度を調査した。

検出失敗

人物が追跡領域内にいるにも拘らず、ID が割当てられていない状態。本論文では検出失敗度数（単位は [秒 × 人]）を評価する¹。

ID 変化

ある人物の ID が途中で変化するエラー。ID 変化の発生には、次に述べる2つの要因がある。1つ目は今まで追跡していた人物に短期間の検出失敗が起こり、その前後で異なる ID が割当てられる場合である。混雑環境下でどの視点からも人物が見えなくなった場合に起こりやすい。2つ目は複数の人物の間で ID が入れ替わる (ID スワップ) 場合で、1度のスワップで複数の人物の ID 変化を伴う。近接した人物が並行して移動する場合に起こりやすい。本論文では ID 変化の回数（単位は [回]）を評価する。

5.3.2 評価結果

データセット $D_1 \sim D_6$ における、各追跡エラーの値を表 5.2 に示す。データセット $D_1 \sim D_5$ では追跡エラーは見られず、映像全編に渡って追跡が正しく行えた。一方、データセット D_6 では、追跡エラーが発生する事が分かった。例えば、図 5.13 は検出失敗が起こった時刻のデータで、追跡領域内中心にいる人物の検出に失敗している。カメラ視点画像と Occupancy Map から分かるように、この人物は全ての視点で隠蔽が発生してい

¹例えば、映像中のある1秒間で n 人の人物の検出に同時に失敗している場合、検出失敗度数 = n である。

第5章 計算機実験

表 5.2: 平面図上での人物追跡エラー ($T_B = 10$ sec)

	D_1	D_2	D_3	D_4	D_5	D_6
ID 変化 [回] (ID スワップ [回])	0(0)	0(0)	0(0)	0(0)	0(0)	12(3)
検出失敗度数 [秒 × 人]	0	0	0	0	0	26.3

表 5.3: 平面図上での人物追跡エラー ($T_B = 7$ sec)

	D_1	D_2	D_3	D_4	D_5	D_6
ID 変化 [回] (ID スワップ [回])	0(0)	0(0)	0(0)	0(0)	1(0)	14(1)
検出失敗度数 [秒 × 人]	0	0	0	0	0	30.3

表 5.4: 平面図上での人物追跡エラー ($T_B = 3$ sec)

	D_1	D_2	D_3	D_4	D_5	D_6
ID 変化 [回] (ID スワップ [回])	0(0)	0(0)	0(0)	2(0)	5(1)	18(1)
検出失敗度数 [秒 × 人]	0	0	0	1.0	1.1	11.1

て、Occupancy が得られていない事が検出失敗の原因である。短時間であれば 4.3.2 節の軌跡の接続で追跡を継続出来る可能性があるが、この場合は隠蔽の期間が長く軌跡の接続が出来なかった。そのため、後にこの人物を再検出した際には新しい ID が割当てられるので、ID 変化も伴う。

また、このような混雑環境下では複数の人物が並行して移動する場面が多く見られ、そのため誤った軌跡が多数作成され、ID スワップを引き起こす原因となっていることが分かった。

5.3.3 クリップ長 T_B に対する追跡エラーの変化

5.3.2 節では、一度の追跡処理に利用する Occupancy Map のクリップ長 $T_B = 10$ sec を利用した結果を示した。本節では、 $T_B = 4, 7$ sec を用いた場合に、追跡結果にどのような影響を与えるかを、追跡エラーを計測する事で評価する。

表 5.3 および表 5.4 に追跡エラーの計測結果を示す。

$T_B = 7$ sec では、データセット D_6 において ID 変化回数、検出失敗度数ともに増加した。また、データセット D_5 で ID 変化が発生した。データセット $D_1 \sim D_4$ では追跡エラーは見られなかった。

次に $T_B = 4$ sec では $T_B = 7$ sec に比べて、データセット D_6 と D_5 において ID 変化回

第5章 計算機実験

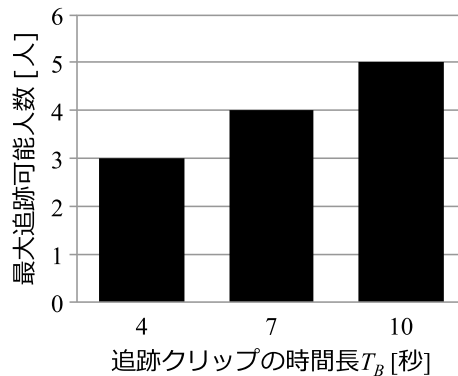


図 5.15: 追跡クリップの時間長 T_B に対する最大追跡可能人数

数, 検出失敗度数が共に増加した. また, データセット D_4 で ID 変化が発生した. データセット $D_1 \sim D_3$ では追跡エラーは見られなかった.

以上をまとめると, T_B を減少させるにつれて, 混雑度の高いデータセットでは追跡エラーが多くなる事が分かった. 図 5.15 に, $T_B = 4, 7, 10$ sec に対して, 追跡エラーが起こらなかった最大の人数を示す. 提案手法は, 色やテクスチャの情報を持たない Occupancy Map であっても, 時系列で分析する事で高い混雑度での追跡が出来るという考えに基づいているが, 確かに長い時間の分析を行う事でより高い混雑度の追跡が可能になっている事が分かった. 一方で, T_B を大きくした場合, 追跡結果が出力されるまでの遅延も大きくなる. 従って, 追跡可能な混雑度と追跡結果の遅延にはトレードオフの関係があり, T_B を調整する事で精度と利便性のどちらかを優先させる事ができる.

5.4 カメラ視点画像上での領域分割エラー評価

本節では, 4.5 節で提案した, カメラ視点画像上での領域分割手法がどの程度の混雑度まで間違えずに領域分割出来るのかを明らかにする事を目的とし, データセット $D_1 \sim D_6$ で発生した領域分割エラーの定量評価を行う.

本論文では, カメラ視点画像上での領域分割に成功している状態を以下のように考える.

- (1) 1 人の人物に対して 1 つの領域が割当てられていること. 同一 ID の領域が 2 人の人物にまたがっていたり, 1 人の人物に異なる ID の領域が複数割当てられている状態は失敗と考える.
- (2) 追跡領域内にいる全ての人物に ID が割当てられ, かつその ID が変化しないこと.

(2) は本研究の目的がビデオセグメンテーションである事からの要求だが, 既に 5.3 にて評価しているので, 本節では (1) の領域分割エラーのみを評価の対象とする. 正しく領域分割されているかどうかは目視によって判断し, エラー発生箇所と期間を乗じた「領域分割エラー度数」(単位は [秒 \times 箇所]) で数値化する.

表 5.5: カメラ視点画像上での領域分割エラー

	D_1	D_2	D_3	D_4	D_5	D_6
領域分割エラー度数 [秒 × 箇所]	0	2.3	5.0	16.6	38.3	58.7

5.4.1 評価結果

表 5.5 に、各データセットに対する領域分割エラー度数 [秒 × 箇所] を示す。混雑度が大きい程、領域分割エラーも増えている事が分かる。

領域分割エラーを発生させる要因として下記の2つが考えられる。

- 平面図上の領域分割失敗に伴う領域分割エラー
- カメラキャリブレーションの誤差による領域分割エラー

平面図上の領域分割失敗に伴う領域分割エラー

平面図上の領域分割が失敗している場合、それに伴ってカメラ視点画像上での領域分割も失敗する。提案手法は、色やテクスチャの情報を使わずに Occupancy のある領域を通る軌跡を作成する。そのため、例えば2人の人物が並行して歩いている場合に間違っただ対応の軌跡が生成されてしまう。追跡処理では、軌跡を形状の類似性を使ってクラスタリングするため、並行移動が短期間であれば2人の人物に対して別々のクラスタが生成される。しかしながら、得られたクラスタのある時刻での断面を利用して平面図上での領域分割を行うため、並行移動している期間で生成された誤った軌跡が2人の領域の重複を生み、領域分割エラーを引き起こしてしまう。このエラーは、人物同士がすれ違う等、近接していてもその期間が一時的な物であれば起こりにくいですが、同じ方向への移動を続ける並行移動の場合には発生し易い。

このエラーを防ぐ為には、カメラ視点画像上での色やテクスチャ情報を使って事後的に領域分割結果を改善したり、人物追跡処理の段階で色やテクスチャ情報を追加利用する等の方法が必要である。

カメラキャリブレーションの誤差による領域分割エラー

本論文ではカメラキャリブレーションによって、3次元空間中のある1点は、どの視点のカメラからも同一の平面図上の位置に投影されると仮定している。しかしながら、実際にはキャリブレーションには誤差があり、ある人物の領域はいくらかのずれを持ったまま平面図上に投影される。そのため、平面図上の領域分割は正しく行っていたとしても、人物同士が近接している場合、カメラ視点画像上では誤って別の人物の ID が割当てられてしまう場合がある。

第5章 計算機実験

また領域分割がうまく行かない別の例として、4.5節で述べたように、ある特定の視点で領域が得られない場合がある。図4.4に示したように、軌跡を作成する際に利用するフローベクトルは、そのベクトルに沿って Occupancy が存在するもの以外は除去される。そのため、キャリブレーションの誤差によって平面図上での人物領域にずれが生じていると、ずれてはみ出ている部分（特に周辺部）において軌跡が作成されにくい傾向がある事が分かった。軌跡が作成されなかった部分では平面図上で領域が得られず、それに伴いカメラ視点画像上でも領域が得られなくなってしまう。

以上のように、キャリブレーション誤差が大きな視点においては、近接している人物の間での間違ったID割り当てや、領域が得られない問題が発生する事が分かった。これを防ぐ為には、より精度の高いキャリブレーションを行ったり、カメラ視点画像上で事後的に領域分割結果を改善する等の方法が必要である。

5.5 「干渉しないクラスタから成るパターンの総数」および「途切れた軌跡の接続方法の総数」の評価

本節では、4.3節で述べた「干渉しないクラスタから成るパターン」と「途切れた軌跡の接続方法」の総数をそれぞれ評価し、最適化処理がどの程度追跡結果に影響を与えているかを考察する。

5.5.1 干渉しないクラスタから成るパターン

干渉しないクラスタから成るパターンの総数（以下 N_p と表記する）は、人物同士の近接によって誤った軌跡が生成されると増加する。あるクリップに対して、 $N_p = 1$ である場合は最適化処理は追跡に貢献していない事になるが、 N_p が大きい傾向が見られれば、誤った軌跡が多数存在することになるので、相対的に最適化処理の重要性が高くなる。

データセット $D_1 \sim D_6$ に対する、生成されたクラスタの総数と N_p をそれぞれ図5.16と図5.17に示す。データセット D_1 での生成クラスタ数は、データセットに含まれる人数と同じく1個、同様にデータセット D_2 についてもほとんどの時刻で2個である。ところがデータセット D_3 以降、混雑度の高い状況では実際の人数よりも多くのクラスタが生成されるようになり、それに伴って N_p も増加傾向にある。

N_p の時間平均 $\overline{N_p}$ を表5.6に示す。データセット D_1 では $\overline{N_p} = 1$ となり、最適化処理は追跡処理に影響を与えていない。一方、データセット D_2 から D_6 では $\overline{N_p}$ に単調増加傾向が認められ、混雑度が高い程最適化処理が重要な役割を果たしている事が分かる。

5.5.2 途切れた軌跡の接続方法

途切れた軌跡の接続方法の総数（以下 N_r と表記する）は、隠蔽によって Occupancy Map が得られない箇所が存在した場合に増加する。あるクリップに対して、 $N_r = 0$ であ

第5章 計算機実験

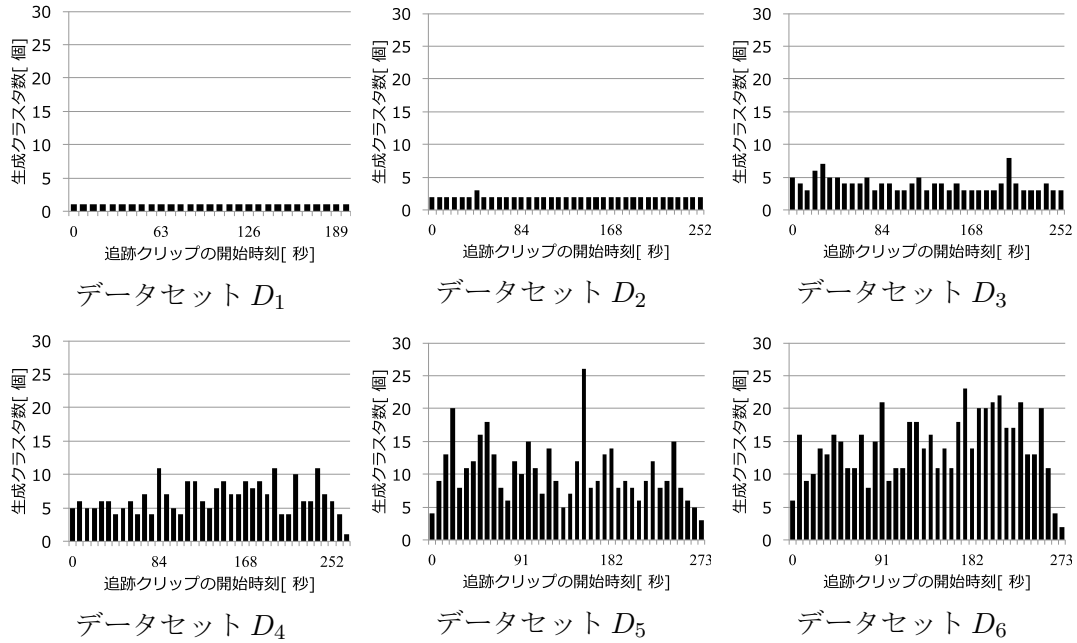


図 5.16: 生成されたクラスタの総数

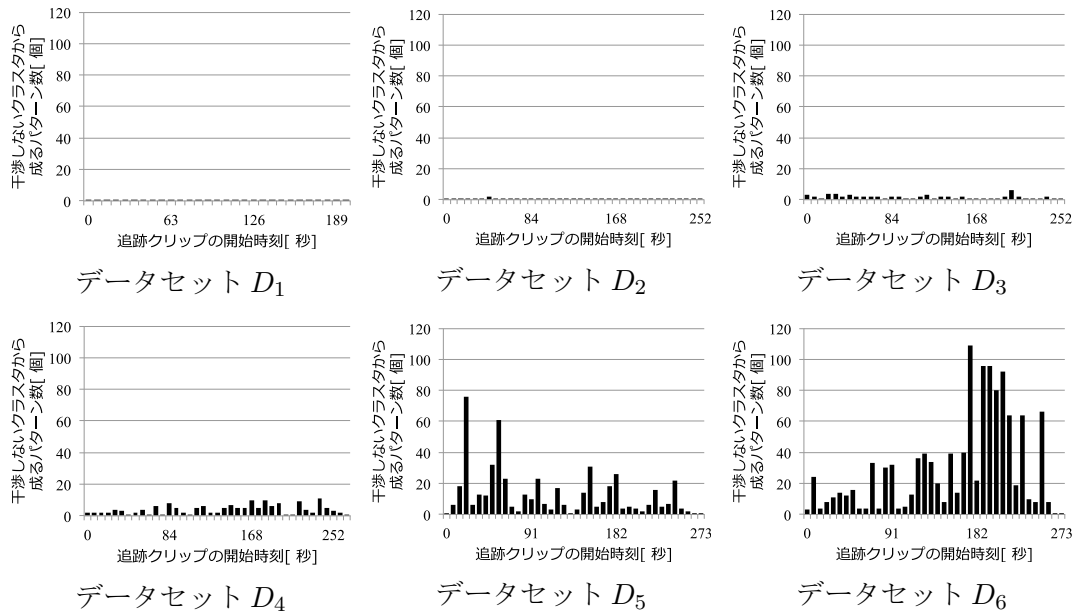


図 5.17: 干渉しないクラスタから成るパターンの総数 N_p

第5章 計算機実験

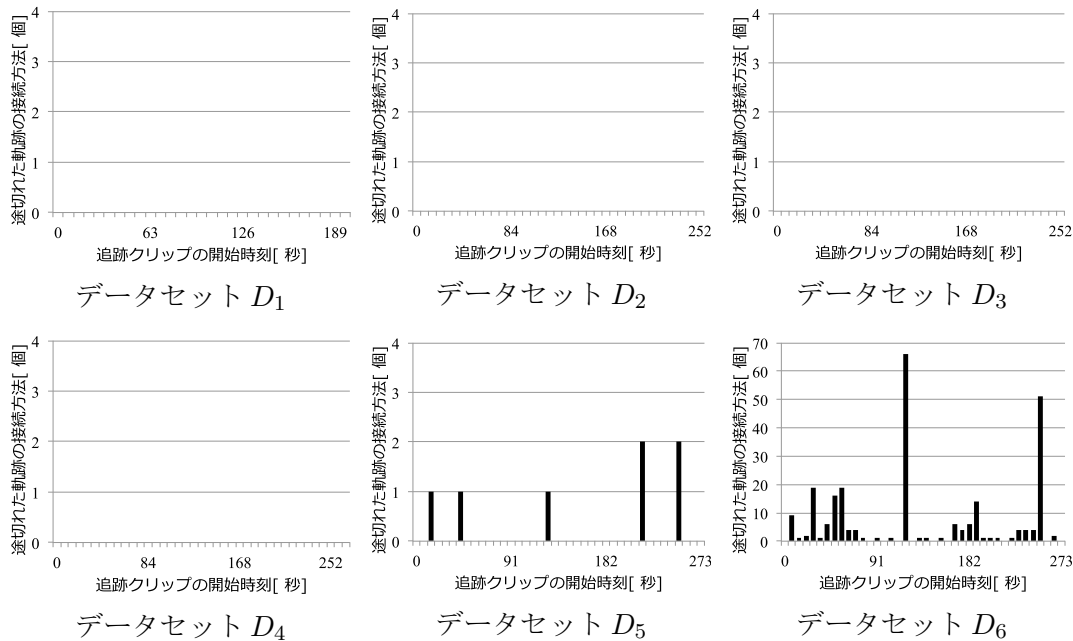


図 5.18: 途切れた軌跡の接続方法の総数 N_r (データセット D_6 のみ縦軸のスケールが異なる事に注意)

る場合は最適化処理は追跡に貢献していない事になるが、 N_r が大きい傾向が見られれば、軌跡の途切れが多数存在することになるので、相対的に最適化処理の重要性が高くなる。

データセット $D_1 \sim D_6$ に対する、 N_r を図 5.18 に示す。 $D_1 \sim D_4$ では途切れが発生していないため $N_r = 0$ だが、 D_5 および D_6 においては途切れた軌跡の接続が行われている事が分かる。特に D_6 では隠蔽による Occupancy の欠如が原因となり、多くの途切れが発生している様子が分かる。

N_r の時間平均 $\overline{N_r}$ を表 5.7 に示す。データセット $D_1 \sim D_4$ では $\overline{N_r} = 0$ となり、最適化処理は追跡処理に影響を与えていない。データセット D_5 から D_6 では $\overline{N_r}$ に単調増加傾向が認められ、混雑度が高い程最適化処理が重要な役割を果たしている事が分かる。一方で、5.3.2 節で述べたように、データセット D_6 では ID 変化や検出失敗の追跡エラーが発生しておりエラーの修復が出来ていないことから、途切れた軌跡の接続で扱える混雑度は、データセット D_5 が限度である事も分かった。

5.6 計算時間と通信量評価

本節では、カメラユニットでの処理時間（ステレオマッチング・中間データ作成・圧縮および送信）と、追跡処理の計算時間、およびネットワークに送信するデータのサイズを評価する。

カメラ視点画像取得から中間データ送信完了までにかかる時間、および中間データのサイ

第5章 計算機実験

表 5.6: 干渉しないクラスタから成るパターンの総数 N_p の時間平均 $\overline{N_p}$

	D_1	D_2	D_3	D_4	D_5	D_6
$\overline{N_p}$ [個]	1.0	1.0	1.9	4.1	13.0	29.7

表 5.7: 途切れた軌跡の接続方法の総数 N_r の時間平均 $\overline{N_r}$

	D_1	D_2	D_3	D_4	D_5	D_6
$\overline{N_r}$ [個]	0.0	0.0	0.0	0.0	0.2	6.3

ズ (通信量) を図 5.19 に示す. 図 5.19 は 4.1 節で述べた全ての中間データ (OM, GM, SM, I_{jpeg}) を作成・送信した場合を示している. 最もデータサイズが大きくなる D_6 でも約 1.6 MB/s であり, 一般的な 1 Gbps (125 MB/s) のネットワーク機器で十分に扱える通信量である. 平均計算時間は D_4 では平均 6.2 fps, D_5 では 5.8 fps, D_6 では 5.5 fps となった. Bumblebee XB3 のフレームレートが約 10 fps なので, コマ落ちが発生してしまう. コマ落ちを防ぐためには, 中間データの圧縮・送信処理をマルチスレッドで行うように実装するか, 高速化が必要であると言える.

追跡・領域分割プログラムにおける $D_4 \sim D_6$ の計算時間を図 5.20 に示す. 提案手法は 1 度の追跡処理で時間長 $T_B = 10$ sec の Occupancy Map のクリップを分析し, 隣接クリップ間には $T_A = 3$ sec の重複期間がある. 従って, 実時間のスループットを確保するには 7 sec 以下で計算を終える必要がある. 図 5.20 のように, $D_4 \sim D_6$ のいずれのデータセットにおいても, 全てのクリップが 5 sec 以下で計算できる結果が得られており, 実時間追跡・領域分割処理が可能であることが確認できた.

5.7 まとめ

本章では, 同じ広さの領域内を 1~6 人の人物がランダムウォークするデータセット $D_1 \sim D_6$ に対して提案手法を使って, 提案手法の隠蔽への頑健性と計算時間および通信量を評価した.

まず各データセットに対する人物追跡と領域分割結果の定性的な評価を述べ, 続いて平面図上での人物追跡エラーとカメラ視点画像上での領域分割エラーの定量評価を行った. 平面図上での人物追跡エラー評価では, 追跡失敗を「ID 変化」と「検出失敗」の 2 つに分類してデータセット毎に頻度を数値化した. その結果, Occupancy Map のクリップ時間長 $T_B = 10$ sec の場合, データセット $D_1 \sim D_5$ では追跡失敗は起こらず, D_6 ではエラーが認められる結果となった. 異なる T_B に対する評価結果からは, T_B を大きくする程, 高い混雑度でも追跡出来る事が確認された. また, カメラ視点画像上での領域分割エラー評価では, 混雑度に対して領域分割エラー度数が単調増加することが分かった.

第5章 計算機実験

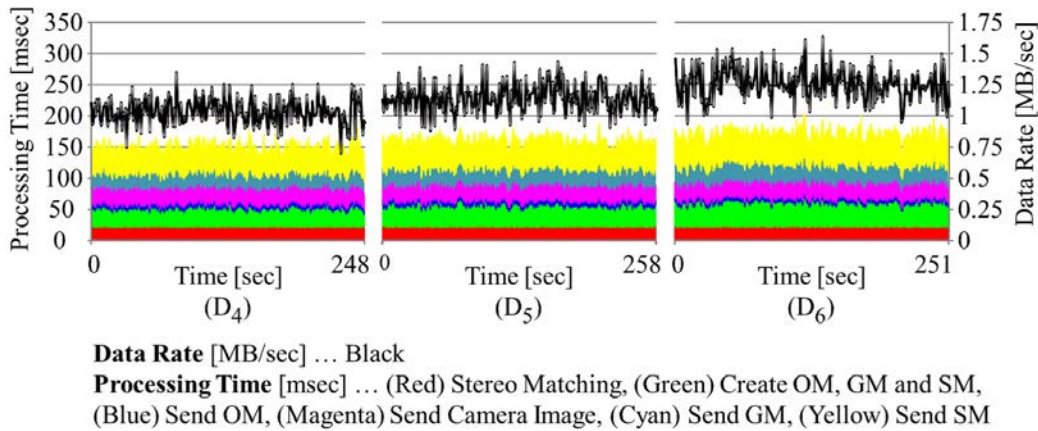


図 5.19: カメラユニットでの計算時間とネットワーク通信量

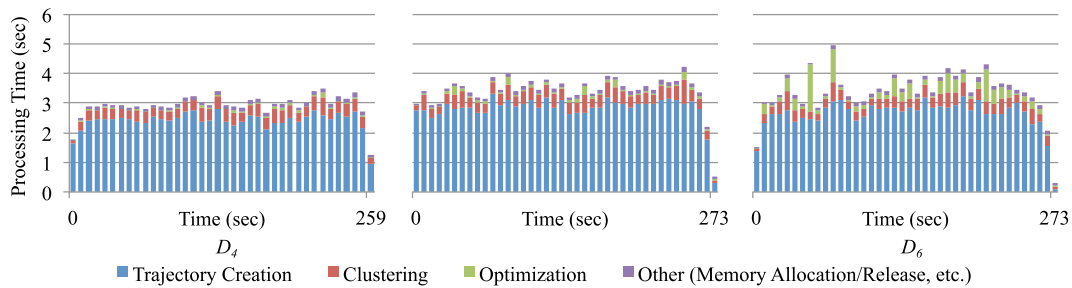


図 5.20: 人物追跡処理に要した計算時間

続いて「干渉しないクラスタから成るパターン」と「途切れた軌跡の接続方法」の総数を評価する実験を行った。その結果、データセット D_6 では極めて多くの途切れが発生している事が分かり、これにより Occupancy Map を利用する提案手法の限界も明らかになった。

最後に計算時間と通信量の評価を行った。人物追跡処理の計算時間はデータセット $D_1 \sim D_6$ の全てにおいて、実時間で計算可能なことが確認できた。このため、提案する人物追跡手法の限界は、精度面からデータセット D_5 と同程度の混雑度 ($0.87[\text{人}/\text{m}^2]$) であると言える。

また、提案手法は画像そのものを伝送せず、中間データに変換してネットワークに送信する。本研究で用いている Bumblebee から得られる画像のサイズは左右カメラでそれぞれ 640×480 画素であることから、グレースケール画像としてネットワークに送信した場合、データサイズは 600 KB である。それに対し、床平面上の $4 \text{ m} \times 4 \text{ m}$ の領域に対応する Occupancy Map は 13 KB に過ぎない。本システムではこれら中間データを圧縮して送信しており、通信量は一般的に入手可能な 1 Gbps のネットワーク機器で十分に扱えるサイズになっている。そのため容易にネットワーク化出来る利便性がある。

第6章 ジェスチャ認識への応用

6.1 概要

ジェスチャ認識は領域分割の応用として最も期待される技術の一つである。隠蔽下でもジェスチャを認識する為には、正しく領域分割を行うだけでなく、複数の視点の結果を統合して欠けのない形状情報を得る必要がある。隠蔽に頑健で、かつ時刻が変化しても同一の人物を示す領域には複数の視点にまたがって同一のラベルを割り当てられる本手法は、ジェスチャ認識の領域分割に最適と言える。

6.2 手法

本研究では、形状情報としてシルエットを利用し、挙手検知を行うプロトタイプシステムを実装した。本デモンストレーションでは、混雑した部屋で、特定の壁平面(図 4.1 中の平面 S_1) に正対した人物が右手を上げているかどうかを提案手法を用いて認識する。シルエットは、領域分割の結果から、各人物領域に対して、対応する奥行き情報を SM を用いて壁平面 S_1 に投影する事で計算される。直感的には、奥行き情報を影絵のように壁平面に投影していると言える。単独視点のシルエットではカメラと人物の位置関係、あるいは隠蔽によって人物領域が欠けてしまう場合があるので、全てのカメラから得たシルエットを重ね合わせて、欠けのないシルエットにして認識に用いる。

6.3 評価

実験に用いる映像(データセット D_7)を図 6.1 に示す。 D_7 では、天井の約 $6\text{ m} \times 3\text{ m}$ の矩形範囲の四隅に取り付けられた合計 4 台のステレオカメラを用いて、先に入室した人物が手を上げ下げする様子を撮影した。図 6.1 ではそのうちの 1 台のカメラ視点画像を示している。途中からほかの入室者があり、図の視点からは隠蔽で挙手の様子が見えにくくなっている。

各時刻 $t_0 \sim t_5$ におけるシルエット作成結果を図 6.3～図 6.8 に示す。図 6.3～図 6.8(i) は全ての視点のシルエットを重ね合わせたもので、(iv) はそれぞれのカメラ視点画像の奥行き情報で作ったシルエットを表している。カメラの外部パラメータに誤差があったため、統合したシルエットにずれが見られるが、手を上げ下げしている様子が見て取れる。

第6章 ジェスチャ認識への応用

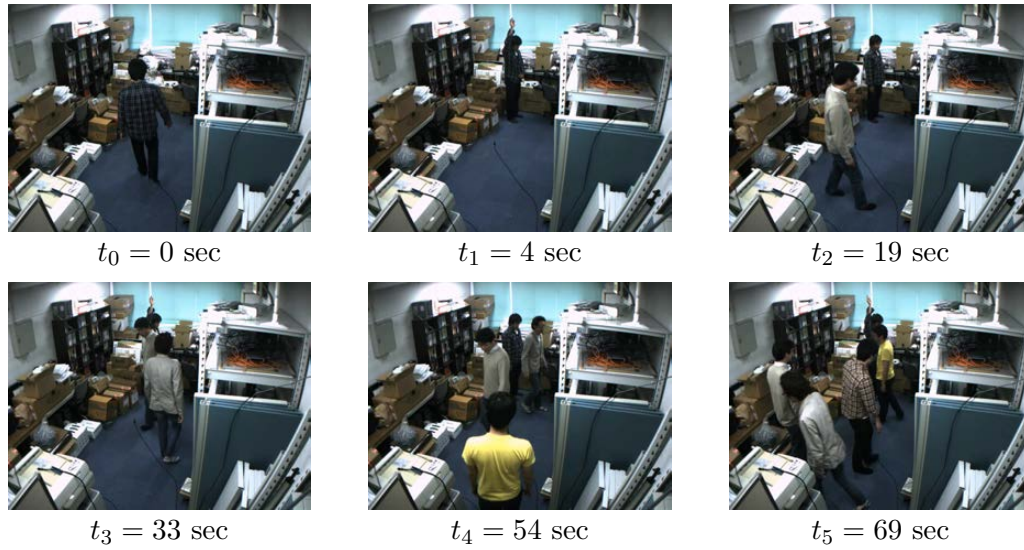


図 6.1: データセット D_7

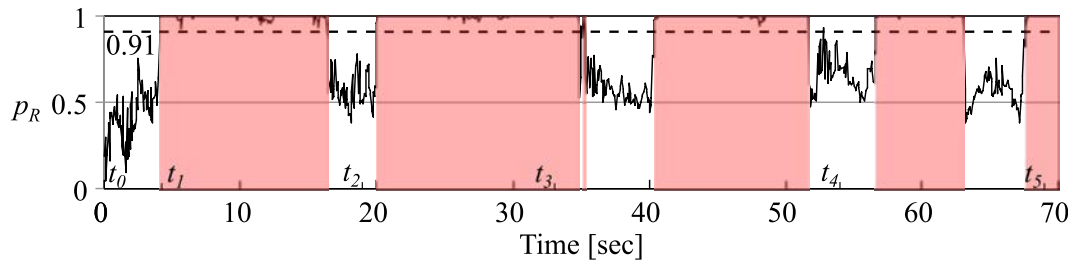


図 6.2: データセット D_7 での挙手検知結果. 赤い領域は p_R が閾値 0.91 を超え, 挙手と判定された時刻を表す.

挙手の判定はシルエットの最上部から 20 cm の矩形領域 (図 6.3～図 6.8(i) 中赤枠で示した) に含まれるシルエットのうち右側にあるシルエットの割合 p_R に閾値を設けて行った. 本実験では $p_R = 0.91$ とした. 図 6.2 に挙手検知結果を示す. 赤色で示した範囲が挙手と判定された時間である. 時間を経るごとに混雑度が増し, 隠蔽により挙手の様子が欠けたり見えなくなるが, 統合シルエットは安定して形状情報を与え続けるので正しく挙手の判定ができる.

第6章 ジェスチャ認識への応用



(i) 4 視点を統合したシルエット

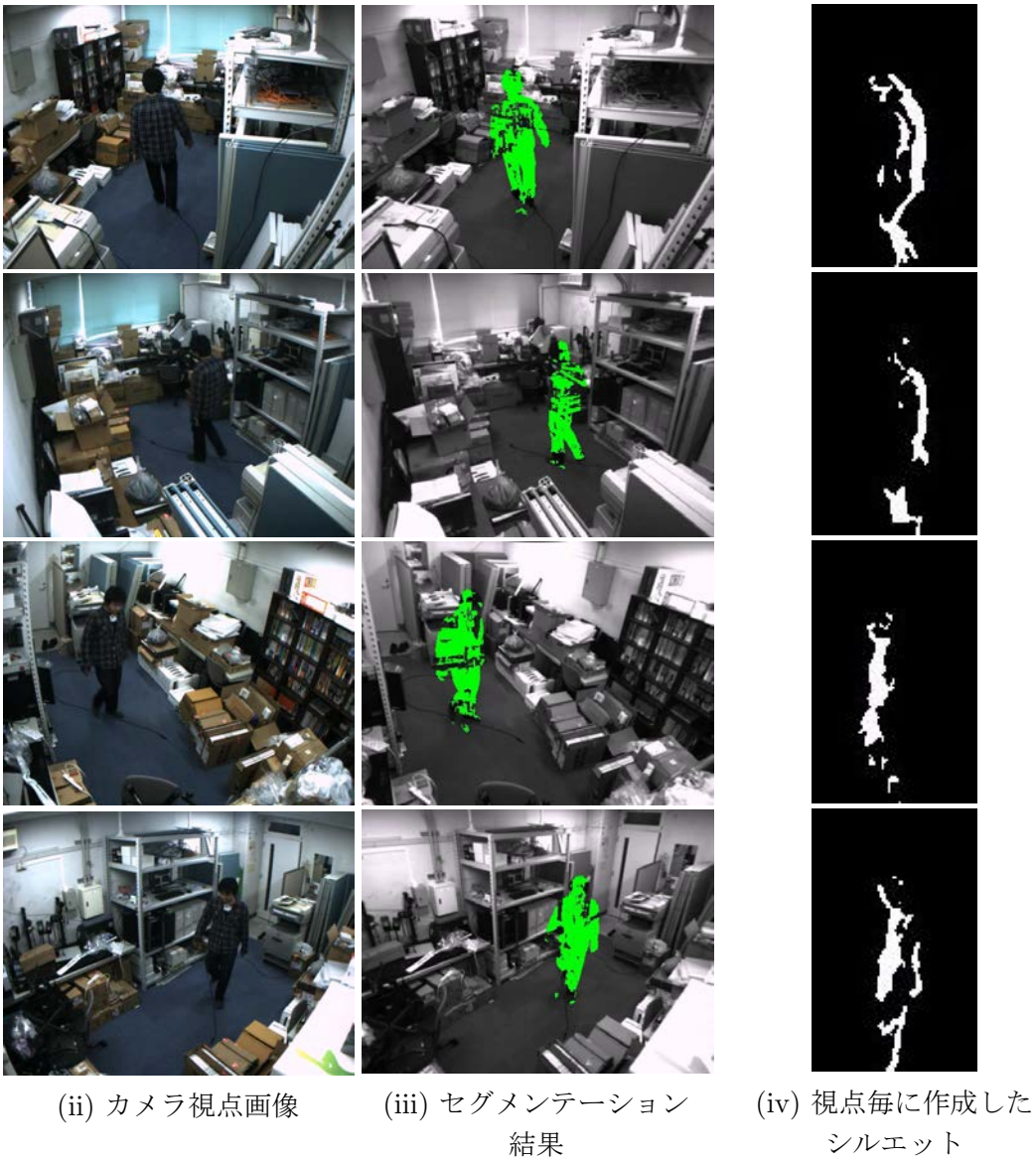


図 6.3: データセット $D_7 : t_0 = 0 \text{ sec}$ におけるシルエット作成結果

第6章 ジェスチャ認識への応用



(i) 4 視点を統合したシルエット

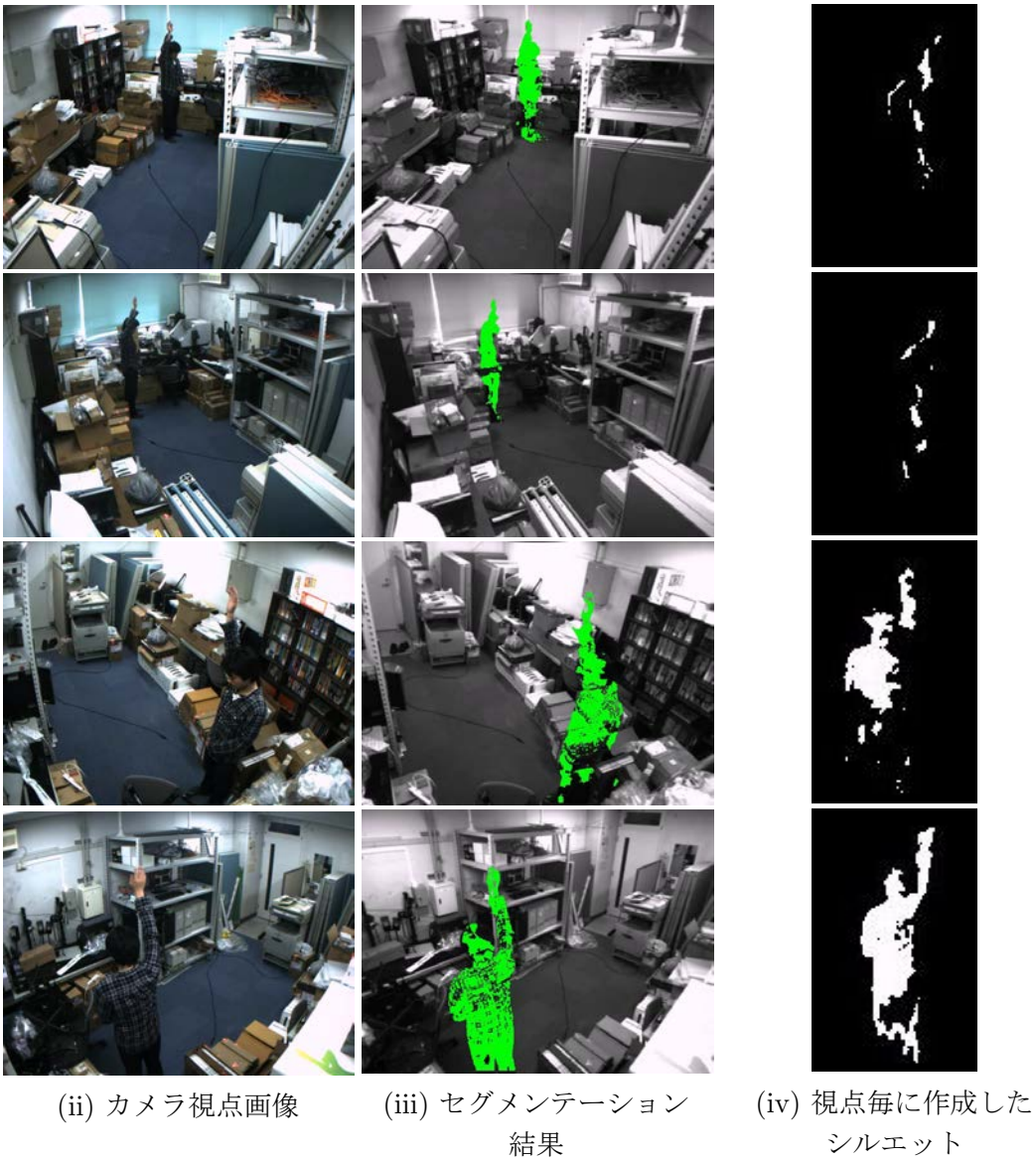


図 6.4: データセット $D_7 : t_1 = 4 \text{ sec}$ におけるシルエット作成結果

第6章 ジェスチャ認識への応用



(i) 4 視点を統合したシルエット

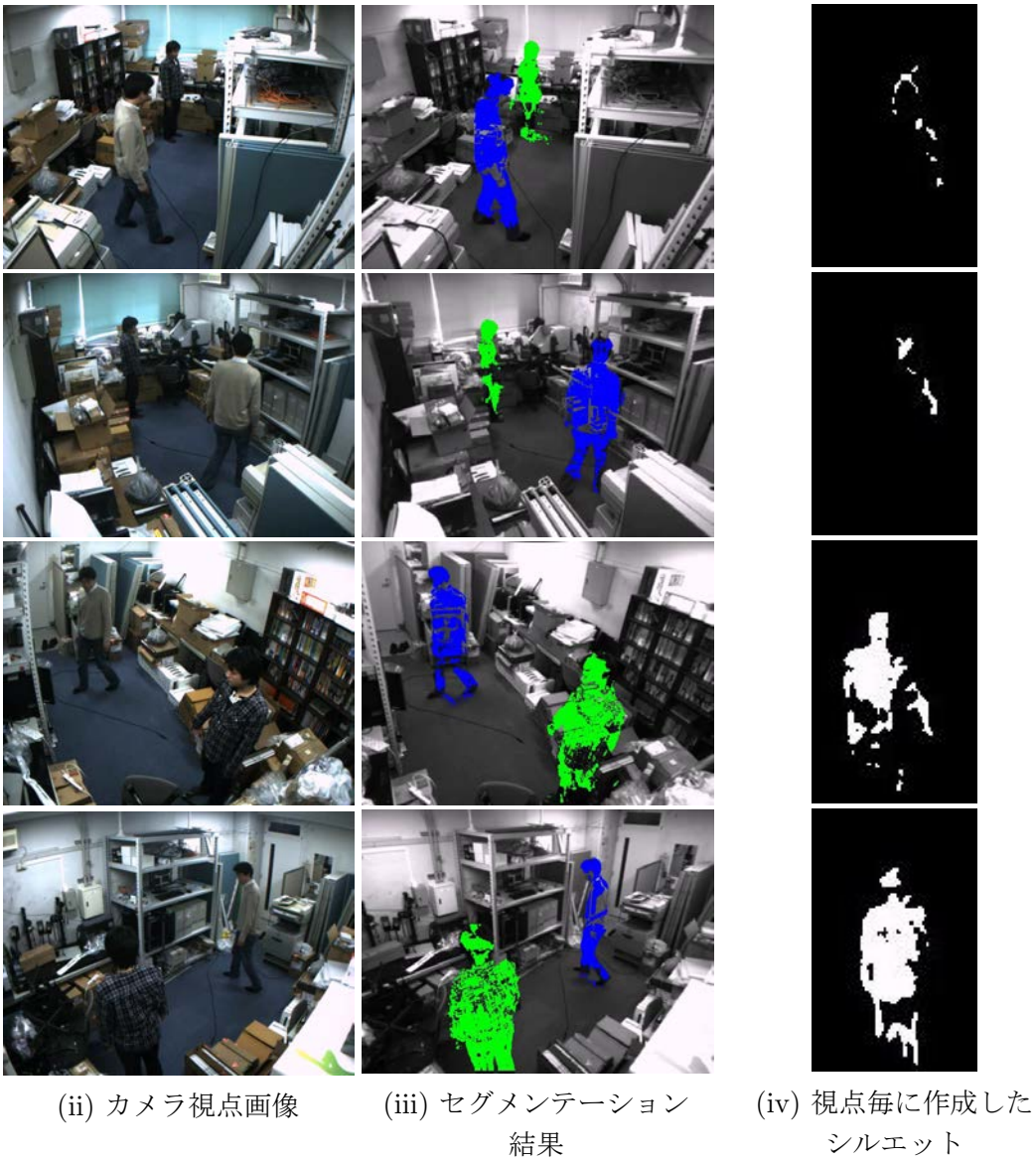


図 6.5: データセット $D_7 : t_2 = 19 \text{ sec}$ におけるシルエット作成結果

第6章 ジェスチャ認識への応用



(i) 4 視点を統合したシルエット

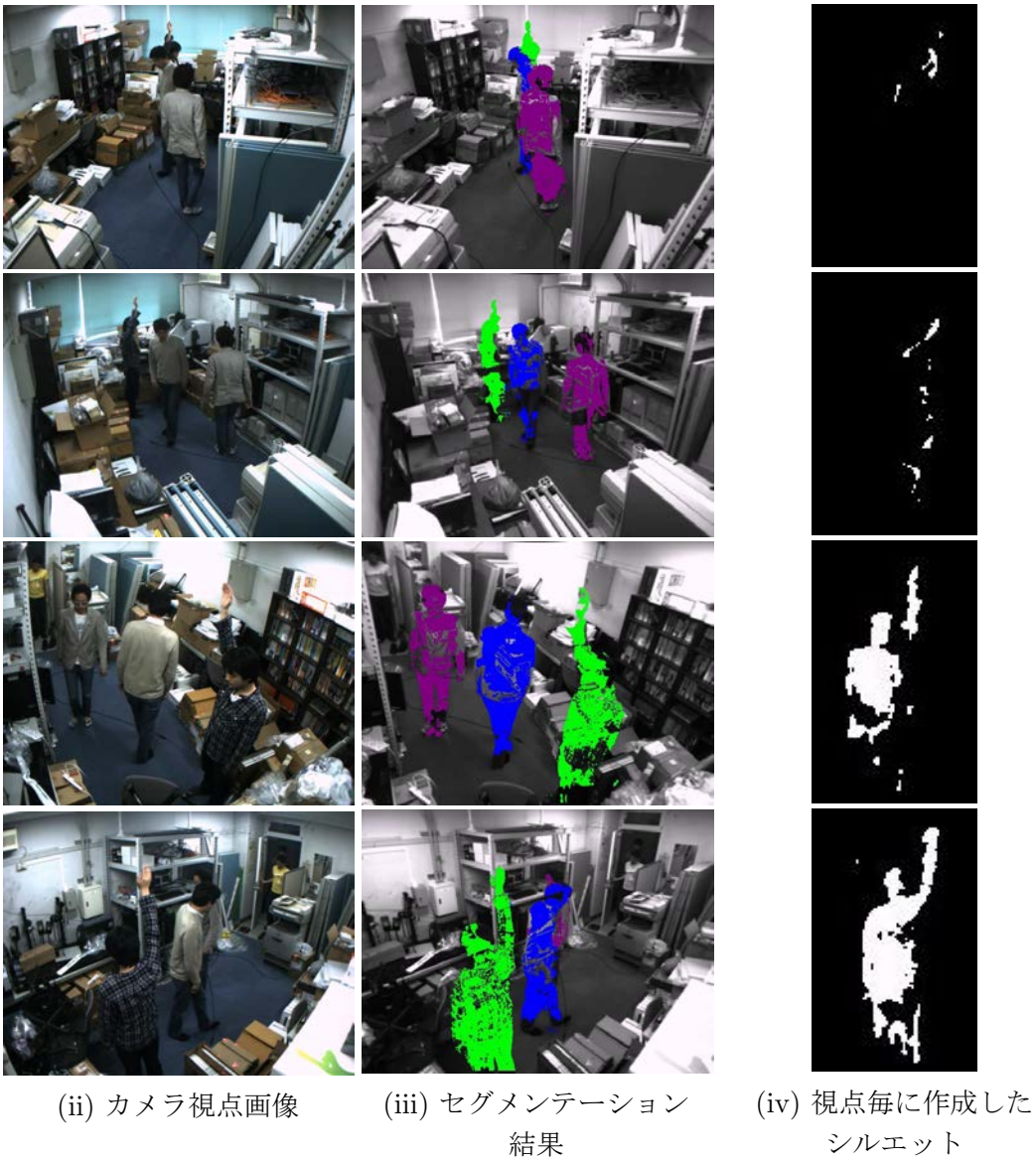


図 6.6: データセット D_7 : $t_3 = 33$ sec におけるシルエット作成結果

第6章 ジェスチャ認識への応用



(i) 4 視点を統合したシルエット

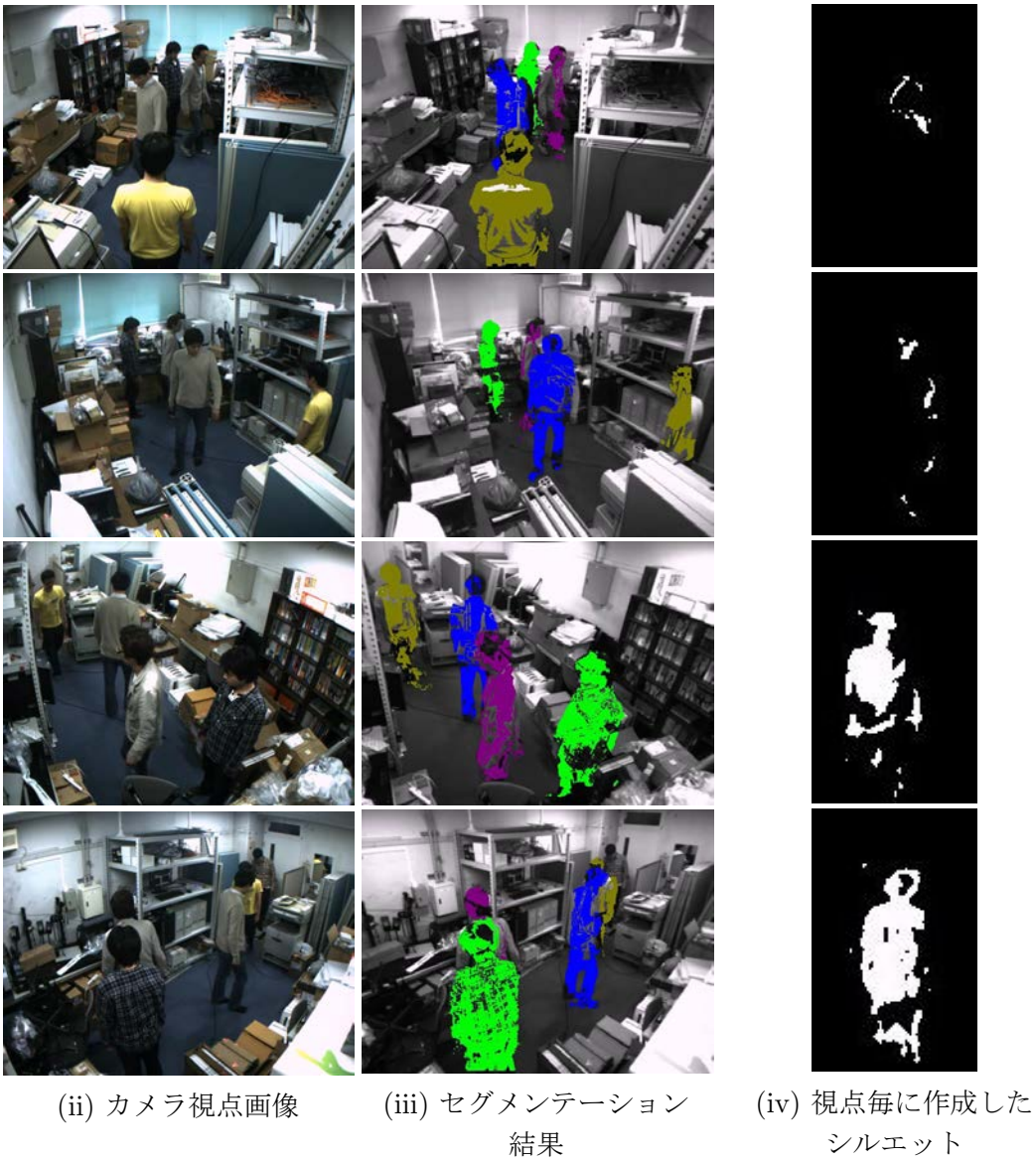


図 6.7: データセット D_7 : $t_4 = 54$ sec におけるシルエット作成結果

第6章 ジェスチャ認識への応用



(i) 4 視点を統合したシルエット

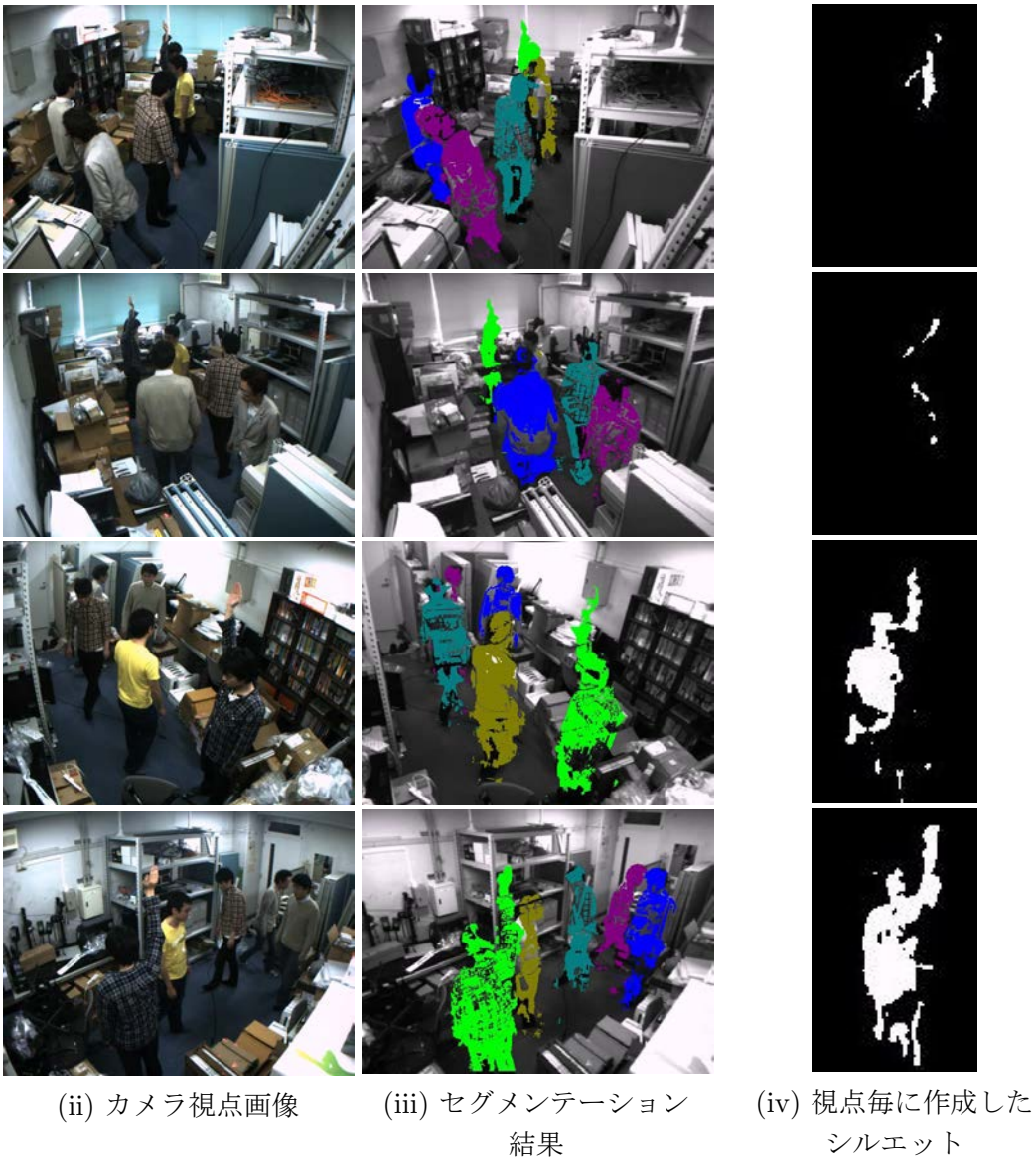


図 6.8: データセット D_7 : $t_5 = 69$ sec におけるシルエット作成結果

6.4 まとめ

本デモで用いたシルエットは、複数の視点から得た 100 msec のシルエットを単純な重ね合わせにより統合しているため、歩行中や動的なジェスチャの認識には適さないが、カメラを同期させたり、複数の非同期カメラから奥行きを推定する手法 [27] によって解決され得る。このプロトタイプシステムを使うと、隠蔽が発生するような混雑環境下でも、特定の人物領域を個別に抽出できるので、一人のジェスチャ認識を対象として開発された手法と簡単に組み合わせられ、複数人物で動作するように拡張できる。例えば、シルエットを用いた、より高度なジェスチャ認識には [5] 等が報告されている。

第7章 結論

近年実用化が盛んな Natural User Interface やスマートスペースには、人物追跡や領域分割技術が不可欠である。しかし実際の環境で安定した人物追跡や領域分割を行うには様々な困難性が伴う。本論文では、困難性の一つである隠蔽に焦点を当て、複数の視点から奥行き情報を得て、それを時系列で分析することで隠蔽に頑健な人物追跡と領域分割を行う手法を提案した。

本章では本論文の内容を総括し、今後の発展性について述べる。

7.1 総括と成果

本論文ではまず、画像処理における人物認識において隠蔽が主要な困難性として知られている事を挙げ、奥行き情報の利用によって隠蔽への頑健性を上げる研究が近年多数報告されていることを述べた。次に、特徴量の利用のし易さから奥行き情報を簡略化した平面図表現が利用される事が多いが、平面図上では人物の形状が失われるためジェスチャ認識への適用がなされてこなかった事を指摘した。

従来より、人物追跡と領域分割に関して多くの研究がなされてきた。そこで関連研究の章では、人物追跡の研究を「複数視点を利用する追跡手法」「奥行き情報を利用する追跡手法」「マルチフレーム最適化による追跡手法」「Data Association」の4つのカテゴリに分けて概説した。次に領域分割の研究を「ビデオセグメンテーション」と「ジェスチャ認識への応用」の2つのカテゴリに分けて概説した。最後に、従来手法の問題点を挙げ、提案手法が従来手法と異なる点を明らかにした。まず人物追跡では、「複数視点の利用」「奥行き情報の利用」「マルチフレーム最適化」の3つの特長を取り入れる事で実環境で安定した追跡が出来ると考えられるが、マルチフレーム最適化には計算時間が大きくなるという問題点がある。そこで本研究では、従来のような動的計画法や反復法による最適化ではなく、ヒューリスティックな最適化手法を独自に提案し、計算時間の問題を解消する事に成功した。領域分割では、従来手法は色やテクスチャの類似性を利用して近傍ピクセルあるいは領域を統合するために、必ずしも1人の人物に1つの領域が割当てられる訳ではなく、通常は顔や胴体等で異なる領域となる事を述べた。本研究では領域分割の応用例としてジェスチャ認識を意識しているが、その為には人物と領域には1対1対応の関係が必要である。そこで本研究では、奥行き情報を使った人物追跡結果を、カメラ画像上に逆投影して領域分割する事により、人物と領域を1対1対応させている。

3章では、開発したシステムの構成と、提案手法の概要を説明した。実験にはステレオ

第7章 結論

カメラを4台利用する。ステレオカメラには1台ずつ計算機が接続されてカメラユニットを構成し、ステレオマッチングと中間データ作成がカメラユニット毎に分散的に処理される。生成された中間データは一般的な機器で構成されたネットワークを通じて、別の計算機で実行されている追跡・領域分割プログラムに送信される。また、ステレオカメラの位置関係は、事前に調べておく必要があることから、本章ではカメラキャリブレーションの手順に着いても解説した。

4章では、人物追跡と領域分割に利用する中間データを定義したあと、時間長 T_B の Occupancy Map のクリップを使って軌跡を作成する方法を述べた。Occupancy のある領域を通る軌跡がいくつも作成されるが、形状に基づいてクラスタリングする事で実際の人数と同じ数のクラスタにまとめられる。しかし実際には、混雑度が高くなると近接する人物との間で間違った軌跡が作成されたり、隠蔽によって軌跡が途切れる場合があることを、実際のデータセットを使って例示した。それらのエラーを修復する為に最適化が必要である事を述べ、まず間違った軌跡を除去する手順を示した。ここでは、間違った軌跡がある場合に「干渉するクラスタ」が生成される事から、「干渉しないクラスタ」のパターンを選び出す事で誤ったクラスタを除去する。次に、途切れた軌跡を接続する手順を述べ、実際のデータセットによってその効果を確認した。最後に、平面図上での追跡結果をカメラ視点画像上に逆投影して領域分割する手法を提案した。ここまでで得られたクラスタは、Occupancy のある部分を通る軌跡の集合なので、ある時刻での断面が平面図上での人物領域に相当する。実際のデータセットを使って人物の形状に適応して領域が得られる事を確認し、人物が近接していても領域分割が正しく行える事を確認した。

5章では、同じ広さの領域内を1~6人の人物がランダムウォークするデータセット $D_1 \sim D_6$ に対して提案手法を適用した実験結果を示した。まず各データセットに対する人物追跡と領域分割結果の定性的な評価を述べ、続いて平面図上での人物追跡エラーとカメラ視点画像上での領域分割エラーの定量評価を行った。平面図上での人物追跡エラー評価では、追跡失敗を「ID変化」と「検出失敗」の2つに分類してデータセット毎に頻度を数値化した。その結果、Occupancy Map のクリップ時間長 $T_B = 10 \text{ sec}$ の場合、データセット $D_1 \sim D_5$ では追跡失敗は起こらず、 D_6 ではエラーが認められる結果となった。異なる T_B に対する評価結果からは、 T_B を大きくする程、高い混雑度でも追跡出来る事が確認された。また、カメラ視点画像上での領域分割エラー評価では、混雑度に対して領域分割エラー度数が単調増加することが分かった。続いて「干渉しないクラスタから成るパターン」と「途切れた軌跡の接続方法」の総数を評価する実験を行った。その結果、データセット D_6 では極めて多くの途切れが発生している事が分かり、これにより Occupancy Map を利用する提案手法の限界も明らかになった。最後に計算時間と通信量の評価を行った。人物追跡処理の計算時間はデータセット $D_1 \sim D_6$ の全てにおいて、実時間で計算可能なことが確認できた。このため、提案する人物追跡手法の限界は、精度面からデータセット D_5 と同程度の混雑度 ($0.87[\text{人}/\text{m}^2]$) であると言える。また、提案手法は画像そのものを伝送せず、中間データに変換してネットワークに送信する事から、通信量は一般的に入手可能な 1Gbps のネットワーク機器で十分に扱えるサイズであり、容

第7章 結論

易にネットワーク化出来る利便性がある事が確認出来た。

6章では、応用の面から提案手法の実用性を調べる為に、挙手動作を検知するジェスチャ認識のプロトタイプシステムを構築した。評価実験では実験室内に5人の人物がおり、うち1人が特定の壁平面に正対して右手を上げ下ろしする。提案手法は全ての視点で領域分割結果が得られるので、それらを重ね合わせる事で、隠蔽が発生していても全身のシルエットを取得出来る事が特徴である。これにより、混雑した環境下でも挙手動作を安定して検出出来る実験結果を得、提案手法の応用性を確認出来た。

7.2 今後の課題

7.2.1 精度向上

データセット D_6 で追跡に失敗する要因としては、隠蔽により Occupancy が得られず軌跡が推定出来なかったことが原因である。これに対しては、より多くの視点を追加する事で改善出来ると考えられる。また、現在はステレオマッチングのエラーの影響を除去する為に、Occupancy Map に閾値処理を施している。そのため、ある視点において隠蔽が部分的な物であったとしても、Occupancy が得られない場合がある。より大きな奥行き画像を使ってエラーの影響を相対的に小さくしたり、エラーの少ない Depth Camera を利用すると言ったアプローチが考えられる。

ステレオカメラの代替になり得る Depth Camera としては、赤外線ランダムパターンを計測対象物に照射する RGB-D カメラ (2.1.2 節) が知られている。RGB-D カメラには、テクスチャに乏しい物体でも計測できる利点がある。しかし、複数の RGB-D カメラが重複領域を持つ場合、赤外線パターンが互いに干渉して計測精度が悪化する。そのため、複数の RGB-D カメラを同時利用する場合は、重複領域をできる限り減らさなくてはならない。この課題に対する解決方法の一例として、Shake'n'Sense [7] が報告されている。Shake'n'Sense は、RGB-D カメラに振動モータを取り付ける事により、カメラ間の相互干渉を緩和する手法である。提案システムでは、装置を単純化する為にステレオカメラを利用したが、このような方法で相互干渉を防ぐことが出来れば、RGB-D カメラは精度向上に役立つと考える。

提案手法で用いる Occupancy Map は色やテクスチャ情報を持っていないが、それらの情報も追加利用する事で追跡精度や領域分割の精度を向上出来る。原口ら [52] は、提案手法を改良して補助的に色情報を付加利用し、人物の近接に極めて頑健な追跡が出来る事を確認している。

また、例えば、顔や頭部領域の検出技術 [2] を利用して、頭部領域のみから Occupancy Map を作成するという方法も考えられる。頭部領域は身体全体の領域よりも小さいので、人物が近接したときに混同する問題を軽減できると思われる。この方法の限界は、混雑環境下においては頭部が常にカメラから見える訳では無い事である。しかし、頭部領域に含まれる顔のテクスチャなどの情報を、上で述べた色情報のように補足的に使用することによって、追跡精度を向上させることが可能であると考えられる。

7.2.2 システム実装の改善

追跡処理の計算時間について、図 5.20 に見られるように現在は $T_B = 10$ sec の追跡クリップを処理するのに 3~5 sec 程度かかっており、やや大きな遅延が発生する。このため、Kinect で行われるような即座にレスポンスを返すジェスチャ入力には追跡・領域分割プログラムの改善が望まれる。追跡・領域分割プログラムは、図 5.20 から分かるように、計算の多くが軌跡作成に費やされている。現在は実装を単純化する目的で、当該処理は 1 コアのみで計算しているが、軌跡は独立して作成されるため、並列計算によって高速化が可能である。

また、カメラユニットは現在ステレオカメラ 1 台と計算機 1 台から成る。実験に用いた計算機は 4 コアを備えるが、前述のように単純化のため 1 コアしか用いていない。しかし複数のコアを使って計算させれば、(1) 高速化によりコマ落ちを防ぐ効果と、(2) 1 台の計算機に複数のステレオカメラを接続できるようになり、カメラユニットの装置構成がより簡易になる効果、の 2 つの効果が得られる。

7.2.3 拡張性

最後に、提案システムの拡張性について論ずる。本章で提案した人物追跡手法は、従来は複数のカメラの撮影タイミングが同期されていなければならなかった事に対して、撮影タイミングが非同期である複数のカメラから作成された Occupancy Map を使って追跡を行うことが出来る。従って、特別な外部装置を必要とせず、容易にカメラユニットの台数を増やすことが出来る。

Occupancy Map, Geometry Map, および Silhouette Map (4.1 節) の作成に要する計算時間に関して、これらのデータの計算はカメラユニット毎に独立して分散処理されるため、カメラの追加によってシステム全体の処理に遅延が発生する事は無い。一方、カメラ台数の増加によって追跡の対象となる人数が増加すると、軌跡の作成 (4.2 節) および最適化 (4.3 節) の計算時間は、図 5.20 に示したように増加する。この課題に対して、まず軌跡の作成に関しては、前節で述べたように並列計算で高速化が可能である。最適化に関しては、クラスタを空間的に互いに素なクラスタ群¹に分割して、それぞれの群に対して個別に最適化を行うことで、パターン数を大幅に減らすことが可能である [9]。

以上のような対策を行う事によって、計算時間の課題は解決する事が可能であり、提案システムの高い拡張性を保つことが出来る。

¹任意の 2 つのクラスタ群が互いに素であるとは、それぞれのクラスタ群に含まれるどのクラスタ間の距離も、ある一定の閾値より大きいことを指す。

本研究に関する論文

【査読付論文】

- Kenichiro Fukushi and Itsuo Kumazawa. Occlusion-Robust Human Tracking with Integrated Multi-View Depth Imagery (複数視点の時系列奥行き情報の統合による隠蔽に頑健な人物追跡) . IEICE Trans. on Information and Systems, Vol.E97-D, No.12, pp.3181-3191, Dec. 2014.
- 福司 謙一郎, 熊澤 逸夫. 複数視点の時系列奥行き情報の統合による隠ぺいに頑健な領域分割. 信学論 D, Vol.J96-D, No.4, pp.1039-1047, Apr. 2013.

【国際会議 (査読付)】

- Kenichiro Fukushi and Itsuo Kumazawa. Real-time human tracking using multiple-view stereo images and continuity constraints of long duration (複数視点のステレオ画像と軌跡の連続性の制約を利用した実時間人物追跡) . Proc. IEEEJ Image Electronics and Visual Computing Workshop (IEVC) 2010, pp.1C-2-1-7, Nice, France, Mar. 2010.

【国内学会・研究会】

- 原口 真悟, 福司 謙一郎, 熊澤 逸夫. 多数のステレオカメラと色情報を用いた人物追跡手法. 信学技報:PRMU, Vol.109, No.470, pp.229-234, Mar. 2010.
- 福司 謙一郎, 熊澤 逸夫. 複数の非同期ステレオカメラを用いた軌跡モデルに基づく人物追跡手法. Proc. 画像の認識・理解シンポジウム (MIRU) 2009, pp.1293-1300, 島根, Jul. 2009.

【その他】

- 福司 謙一郎, 熊澤 逸夫. 複数カメラを用いた人物追跡手法の研究例, 映像情報メディア学会誌:映像情報メディア, Vol.62, No.7, pp.993-996, 2008.

謝辞

本論文は、東京工業大学大学院総合理工学研究科物理情報システム専攻に在籍中の研究成果をまとめたものです。

本研究を行うにあたり、指導教官として研究実施の機会を与えて頂き、さらに、研究の進め方、考え方など、常日頃よりわかりやすく御指導頂いた同専攻教授熊澤逸夫先生に深く感謝致します。また、副査として御助言を頂くとともに本論文の細部にわたり御指導頂いた同専攻併任教授佐藤誠先生、同専攻併任教授長橋宏先生、同専攻教授山口雅浩先生、同専攻准教授小尾高史先生、横浜国立大学大学院環境情報研究院教授長尾智晴先生に深く感謝致します。

そして、ゼミを通じて活発な議論にお付き合い頂いた熊澤研究室の皆様に深く感謝致します。

筆者が2010年10月から2012年1月にかけて米国MITメディアラボ、カメラカルチャーグループに訪問学生として在籍していた期間、研究のみならず慣れない留学生活についても親身にサポートして下さったMITメディアラボ准教授、カメラカルチャーグループディレクター Ramesh Raskar 先生に、深く感謝申し上げます。また留学に際し御尽力賜った慶應義塾大学教授稲見昌彦先生と、資金援助をして下さった東京工業大学博士一貫教育グループおよびマサチューセッツ工科大学に厚くお礼申し上げます。

留学中に研究を共にしたMITメディアラボ、カメラカルチャーグループの皆様、今後の一層の活躍をお祈りします。特に、研究プロジェクトを一緒に行った、Christian Schönauer さん、Jan Zizka さん、Dr. Alex Olwal さんに多大なる親愛と感謝の気持ちを送りたいと思います。

同志社女子大学教授上田信行先生、慶應義塾大学准教授田中浩也先生、神戸大学助教(当時)竹川佳成先生、およびMITメディアラボバイオメカトロニクスグループ(当時)遠藤謙さん、同研究所アフェクティブコンピューティンググループ佐野あかねさん、MITコンピュータ科学・人工知能研究所(当時)新山龍馬さんからは、留学中公私に渡り数えきれない程の御助言御助力を賜り、厚くお礼申し上げます。

最後に、研究生活を送る上で様々な面で支えとなってくれた家族と友人に心より感謝致します。

付録A カメラキャリブレーション

A.1 カメラキャリブレーションに用いるパターン

カメラキャリブレーションには、図 A.1 に示す 5×8 マスのチェスボードパターン（1 辺の長さが 135mm）を用いた。

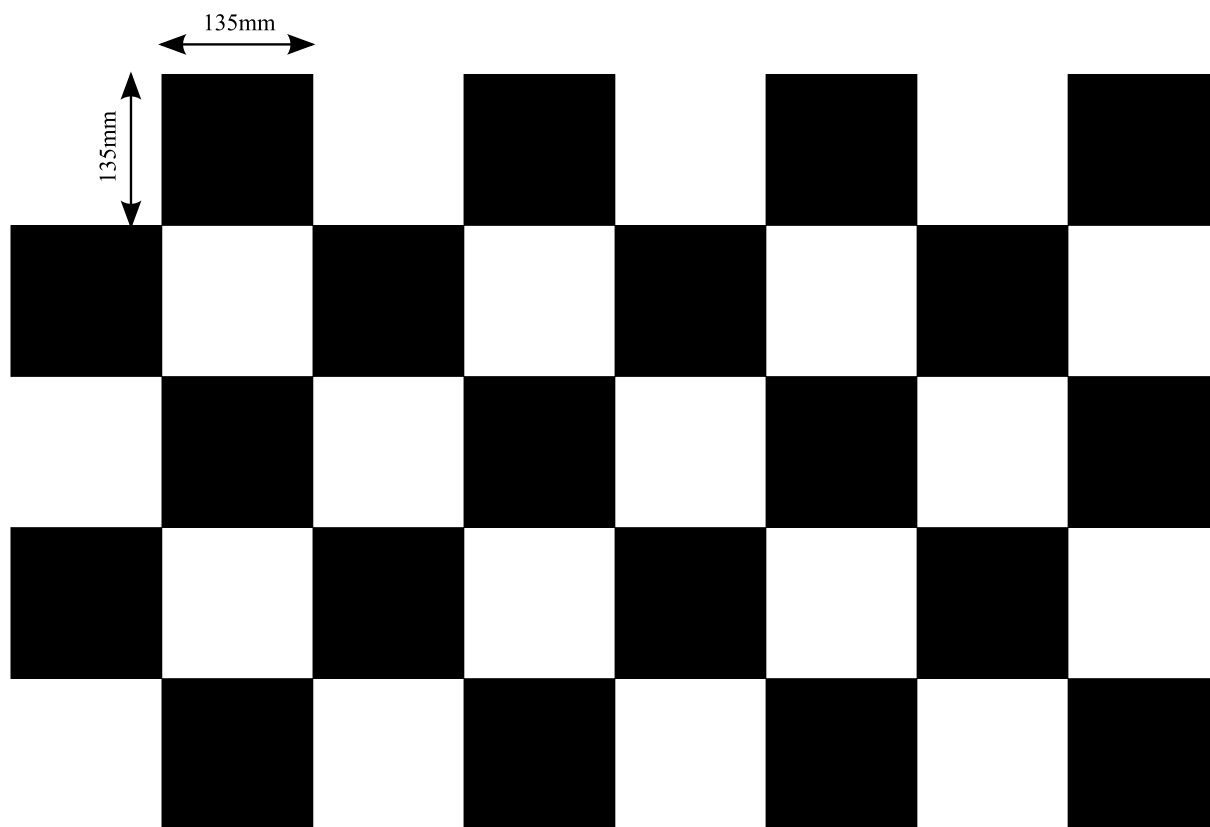


図 A.1: カメラキャリブレーションに用いるパターン

A.2 推定された各カメラの回転行列および並進行列

$${}^W\mathbf{R}_1 = \begin{bmatrix} -0.266637 & 0.463392 & -0.845088 \\ 0.963665 & 0.142675 & -0.225816 \\ 0.015931 & -0.874593 & -0.484597 \end{bmatrix}, {}^W\mathbf{t}_1 = \begin{bmatrix} 3092.296842 \\ 902.731857 \\ 2329.730878 \end{bmatrix} \quad (\text{A.1})$$

$${}^W\mathbf{R}_2 = \begin{bmatrix} 0.427458 & 0.381825 & -0.777368 \\ 0.881606 & -0.182259 & 0.409734 \\ 0.00365627 & -0.883594 & -0.425495 \end{bmatrix}, {}^W\mathbf{t}_2 = \begin{bmatrix} 3091.74 \\ -1774.32 \\ 2249.77 \end{bmatrix} \quad (\text{A.2})$$

$${}^W\mathbf{R}_3 = \begin{bmatrix} -0.57667 & -0.504453 & 0.717895 \\ -0.861354 & 0.212963 & -0.489561 \\ 0.0568883 & -0.837922 & -0.502144 \end{bmatrix}, {}^W\mathbf{t}_3 = \begin{bmatrix} -2482.62 \\ 785.784 \\ 2248.62 \end{bmatrix} \quad (\text{A.3})$$

$${}^W\mathbf{R}_4 = \begin{bmatrix} 0.69584 & -0.345072 & 0.653738 \\ -0.749439 & -0.297338 & 0.587958 \\ -0.00728282 & -0.918726 & -0.45147 \end{bmatrix}, {}^W\mathbf{t}_4 = \begin{bmatrix} -2164.07 \\ -1835.4 \\ 2268.23 \end{bmatrix} \quad (\text{A.4})$$

参考文献

- [1] Mohamed F. Abdelkader, Wael Abd-Almageed, Anuj Srivastava, and Rama Chellappa. Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds. *Computer Vision and Image Understanding*, Vol. 115, No. 3, pp. 439–455, 2011.
- [2] Irshad Ali and Matthew N. Dailey. Multiple human tracking in high-density crowds. *Image and Vision Computing*, Vol. 30, No. 12, pp. 966–977, 2012.
- [3] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi, and L. Scozzafava. Real-time people localization and tracking through fixed stereo vision. In *Proc. Int. Conf. on Innovations in Applied Artificial Intelligence*, pp. 44–54, 6 2010.
- [4] David Beymer. Person counting using stereo. In *Proc. Workshop on Human Motion*, pp. 127–133, Austin, Texas, USA, 12 2000. IEEE.
- [5] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proc. 10th IEEE Int. Conf. on Computer Vision*, Vol. 2, pp. 1395–1402, Beijing, China, 10 2005.
- [6] Yuri Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. 8th IEEE Int. Conf. on Computer Vision*, Vol. 1, pp. 105–112, Vancouver, British Columbia, Canada, 8 2001.
- [7] D. Alex Butler, Shahram Izadi, Otmar Hilliges, David Molyneaux, Steve Hodges, and David Kim. Shake'n'sense: Reducing interference for overlapping structured light depth cameras. In *Proc. Conf. on Human Factors in Computing Systems*, pp. 1933–1936, Austin, TX, USA, 5 2012.
- [8] Cevahir Çiğla and A.Aydin Alatan. Depth assisted object segmentation in multi-view video. In *Proc. 3DTV Conf.: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 185–188, Istanbul, 5 2008.
- [9] Ingemar J. Cox. A review of statistical data association techniques for motion correspondence. *Int. J. Computer Vision*, Vol. 10, No. 1, pp. 53–66, 1993.

参考文献

- [10] ChunHui Cui, Qian Zhang, and KingNgi Ngan. Multi-view video based object segmentation - a tutorial. *ECTI Trans. on Electrical Eng.*, Vol. 7, No. 2, pp. 90–105, 8 2009.
- [11] Jinshi Cui, Hongbin Zha, Huijing Zhao, and Ryosuke Shibasaki. Laser-based detection and tracking of multiple people in crowds. *Computer Vision and Image Understanding*, Vol. 106, No. 2-3, pp. 300–312, 2007.
- [12] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenszwalb. Plan-view trajectory estimation with dense stereo background models. In *Proc. 8th IEEE Int. Conf. on Computer Vision*, Vol. 2, pp. 628–635, Vancouver, British Columbia, Canada, 7 2001.
- [13] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proc. Int. Conf. on Computer Vision*, Vol. 2, pp. 1071–1078, Nice, France, 10 2003.
- [14] Jan Dupač, Jiří Matas, and Filip Naiser. Ultra-fast tracking based on zero-shift points. *Image and Vision Computing*, pp. 1–46, 8 2012.
- [15] Gwenn Englebienne, Tim van Oosterhout, and Ben Krose. Tracking in sparse multi-camera setups using stereo vision. In *Proc. ACM/IEEE Intl. Conf. on Distributed Smart Cameras*, pp. 1–6, 8 2009.
- [16] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 267–282, 2 2008.
- [17] Kenichiro Fukushi and Itsuo Kumazawa. Real-time human tracking using multiple-view stereo images and continuity constraints of long duration. In *Proc. IEEEJ Image Electronics and Visual Computing Workshop*, pp. 1C-2-1-7, Nice, France, 3 2010.
- [18] G. Gennari and G. D. Hager. Probabilistic data association methods in visual tracking of groups. In *Proc. Conf. on Computer Vision and Pattern Recognit.*, Vol. 2, pp. 876–881, Washington, DC, USA, 6 2004.
- [19] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proc. Conf. on Computer Vision and Pattern Recognit.*, Vol. 2, pp. 459–464, Fort Collins, CO, USA, 6 1999.
- [20] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 2141–2148, Atlanta, GA, USA, 6 2010.

参考文献

- [21] I. Haritaoglu, D. Harwood, and L.S. Davis. *W⁴S*: A real time system for detecting and tracking people in $2\frac{1}{2}D$. In *Proc. European Conference on Computer Vision*, Vol. 1, pp. 877–892, Freiburg, Germany, 6 1998.
- [22] Michael Harville. Stereo person tracking with short and long term plan-view appearance models of shape and color. In *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp. 522–527, Como, Italy, 9 2005.
- [23] M. El Hassani, S. Jehan-Besson, L. Brun, M. Revenu, M. Duranton, D. Tschumperl, and D. Rivasseau. A time-consistent video segmentation algorithm designed for real-time implementation. *VLSI Design*, Vol. 2008, No. 2, pp. 7:1–7:12, 1 2008.
- [24] Martin Hofmann, Michael Haag, and Gerhard Rigoll. Unified hierarchical multi-object tracking using global data association. In *Proc. 2013 IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 22–28, Clearwater, FL, USA, 1 2013.
- [25] Luca Iocchi and Robert C. Bolles. Integrating plan-view tracking and color-based person models for multiple people tracking. In *Proc. IEEE Int. Conf. on Image Process.*, Vol. 3, pp. III – 872–5, Genoa, Italy, 9 2005.
- [26] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event detection in crowded videos. In *Proc. Int. Conf. on Computer Vision*, pp. 1–8, 10 2007.
- [27] Felix Klose, Christian Lipski, and Marcus Magnor. Reconstructing shape and motion from asynchronous cameras. In *Proc. 15th Int. Workshop on Vision, Modeling and Visualization*, pp. 171–177, Siegen, Germany, 11 2010.
- [28] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easyliving. In *Proc. IEEE Int. Workshop on Visual Surveillance*, pp. 3–10, 7 2000.
- [29] Abdol-Reza Mansouri, Amar Mitiche, and Michael Aron. Pde-based region tracking without motion computation by joint space-time segmentation. In *Proc. Int. Conf. on Image Processing*, Vol. 3, pp. III – 113–16, Barcelona, Catalonia, Spain, 9 2003.
- [30] Anurag Mittal and Larry S. Davis. M₂Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int. J. Computer Vision*, Vol. 51, No. 3, pp. 189–203, 2003.

参考文献

- [31] Gelareh Mohammadi, Frederic Dufaux, Thien Ha Minh, and Touradj Ebrahimi. Multi-view video segmentation and tracking for video surveillance. In *SPIE Proc. Mobile Multimedia/Image Processing, Security and Applications*, Orlando, 4 2009.
- [32] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *Proc. SIGGRAPH*, pp. 191–198, 6 1995.
- [33] Rafael Muñoz-Salinas. A bayesian plan-view map based approach for multiple-person detection and tracking. *Pattern Recognit.*, Vol. 41, No. 12, pp. 3665–3676, 12 2008.
- [34] Rafael Muñoz-Salinas, Miguel García-Silvente, and Rafael Medina Carnicer. Adaptive multi-modal stereo people tracking without background modelling. *Journal of Visual Communication and Image Representation*, Vol. 19, No. 2, pp. 75–91, 2008.
- [35] Rafael Muñoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas, and A. Carmona-Poyato. Depth silhouettes for gesture recognition. *Pattern Recognition Letters*, Vol. 29, No. 3, pp. 319–329, 2008.
- [36] Rafael Muñoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas, and A. Carmona-Poyato. Multi-camera people tracking using evidential filters. *Int. Journal of Approximate Reasoning*, Vol. 50, No. 5, pp. 732–749, 2009.
- [37] Nandita M. Nayak, Yingying Zhu, and Amit K. Roy-Chowdhury. Vector field analysis for multi-object behavior modeling. *Image and Vision Computing*, pp. 1–34, 8 2012.
- [38] C. Reinbacher, M. Ruther, and H. Bischof. Fast variational multi-view segmentation through backprojection of spatial constraints. *Image and Vision Computing*, Vol. 30, No. 11, pp. 797–807, 2012.
- [39] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. *Int. J. Robotics Research*, Vol. 22, No. 2, pp. 99–116, 2003.
- [40] Vivek Kumar Singh, Bo Wu, and Ramakant Nevatia. Pedestrian tracking by associating tracklets using detection residuals. In *Proc. IEEE Workshop on Motion and Video Computing*, pp. 1–8, Copper Mountain, CO, USA, 6 2010.
- [41] Junghae Suh, Michelle Dawson, and Justin Hanes. Real-time multiple-particle tracking: applications to drug and gene delivery. *Advanced Drug Delivery Reviews*, Vol. 57, No. 2, pp. 63–78, 2005.
- [42] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

参考文献

- [43] Radu Timofte, Karel Zimmermann, and Luc Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. *Machine Vision and Applications*, pp. 1–15, 12 2011.
- [44] David Tsai, Matthew Flagg, Atsushi Nakazawa, and James M. Rehg. Motion coherent tracking using multi-label mrf optimization. *Int. Journal of Computer Vision*, Vol. 100, No. 2, pp. 190–202, 2012.
- [45] Hideki Tsutsui, Jun Miura, and Yoshiaki Shirai. Optical flow-based person tracking by multiple cameras. In *Proc. Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 91–96, Baden-Baden, Germany, 8 2001.
- [46] L.P.J. Vosters, Caifeng Shan, and Tommaso Gritti. Real-time robust background subtraction under rapidly changing illumination conditions. *Image and Vision Computing*, Vol. 30, No. 12, pp. 1004–1015, 2012.
- [47] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, Vol. 38, No. 4, p. 13, 2006.
- [48] Xenophon Zabulis, Dimitris Grammenos, Thomas Sarmis, Konstantinos Tzevanidis, Pashalis Paderis, Panagiotis Koutlemanis, and Antonis A. Argyros. Multi-camera human detection and tracking supporting natural interaction with large-scale displays. *Machine Vision and Applications*, Vol. 24, No. 2, pp. 319–336, 2013.
- [49] Cha Zhang and Zhengyou Zhang. A survey of recent advances in face detection. *Microsoft Research TechReport*, No. MSR-TR-2010-66, pp. 1–17, 2010.
- [50] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, pp. 1330–1334, 11 2000.
- [51] Mingcai Zhou, Lin Liang, Jian Sun, and Yangsheng Wang. Aam based face tracking with temporal matching and face segmentation. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 701–708, San Francisco, CA, USA, 6 2010.
- [52] 原口真悟, 福司謙一郎, 熊澤逸夫. 多数のステレオカメラと色情報を用いた人物追跡手法. 信学技報:PRMU, Vol. 109, No. 470, pp. 229–234, 3 2010.
- [53] 藤村一行, 内田誠一. 大局的最適化に基づくトラッキング : DP トラッキング. 信学論 D, Vol. J92-D, No. 8, pp. 1279–1288, 2009.

参考文献

- [54] 布施孝志, 中西航. 歩行者挙動モデルを統合した人物自動追跡手法の構築. 土木学会
論文集 D3, Vol. 68, No. 2, pp. 92-104, 2012.