低フレームレート物体検出と高フレームレート特徴点追跡の統合による高速・高精度な複数物体追跡

西村 仁志[1,a)]　小森田 賢史[1,b)]　川西 康友[1,2,3,c)]　村瀬 洋[1,3,d)]

概要

物体検出に基づく追跡手法は，物体の大きな位置変化にも頑健であり，近年主流となっている．しかし，追跡対象の物体が他の物体によって部分的に遮蔽された場合に未検出が生じ，追跡の途切れや ID スイッチが発生しやすい．この問題を解決するため，本論文では，物体検出と，物体の一部の領域のみでも追跡可能な特徴点追跡を統合した手法を提案する．実験では，MOT16 データセットを用いて，追跡の途切れと ID スイッチの削減を確認する．また，低速な物体検出は低フレームレートで行うことにより，処理の高速化も確認する．

1. はじめに

複数物体追跡は状況理解のための基礎的な技術で，マーケティング・監視・スポーツ解析等様々な分野で応用されている．複数物体追跡は画像中の各物体の位置を他の物体と間違えずに推定し続けるタスクであり，物体の位置と ID を正しく推定し続ける必要がある．しかし，実応用では，遮蔽・回転・ブレ・照明変化等の様々な要因によって，追跡の途切れや ID スイッチが発生してしまう．

近年の深層学習技術の発展により，物体検出精度は大幅に向上した．これに伴い，現在は検出に基づく追跡手法が主流となっている．これらの手法は，検出器で物体を検出し，何らかの指標を用いてその結果を対応付けることによって物体追跡を実現する．SORT [2] では，対応付け指標として，検出した矩形間の重複率を用いている．DeepSORT [9] では，対応付け指標として，矩形間の重複率に加えて深層学習に基づく特徴量を用いている．DeepSORT では，物体検出のための深層学習モデルと，対応付け指標算出のための深層学習モデルはそれぞれ別々のものになっている．FairMOT [10] では両者を 1 つに統合した深層学習モデルを用いマルチタスク学習を行うことで，より良い特徴量が抽出可能となっている．しかし，これらの検出に基づく手法では，対象物体が他の物体によって部分的に遮蔽された場合に未検出が生じ，追跡の途切れや ID スイッチが発生しやすい．

本研究では，物体の一部の領域のみでも追跡可能な特徴点追跡を利用することで，この問題の解決を目指す．特徴点追跡による物体追跡手法はこれまでにも提案されている．NOMT [5] は検出に基づく追跡手法であり，オプティカルフローによって動きのパターンを用いて対応付けをしている．しかし，あくまで検出に基づく手法であるため，未検出による追跡の途切れは解決できていない．MedianFlow [6] は検出に基づかず，オプティカルフローによって追跡するが，追跡対象が単一の場合に特化しているため，複数物体には適用が難しい．Bullinger らも同様に検出に基づかず，領域分割に基づいてオプティカルフローによって追跡しているが [4]，遮蔽時には領域分割に失敗する可能性が高い．

本論文では，物体検出と，物体の一部の領域のみでも追跡可能な特徴点追跡を統合した手法を提案する．本論文では人物に焦点を当て，他の部位と比較して遮蔽される可能性が低く，かつ動きが少ない頭部周辺に特徴点を配置する．配置した特徴点のオプティカルフローを推定し，その中央値によって物体全体の位置を推定する．特徴点追跡によって，物体が部分的に遮蔽された場合でも，未検出を防止し，追跡の途切れや ID スイッチを防ぐことができる．また，ある特定のフレームでは検出器による物体検出を行い，特徴点によって推定した位置と対応付けることによって両者を統合する．未対応の検出結果があった場合は，新規物体として追跡を開始する．なお，オプティカルフローによる特徴点追跡は物体検出や対応付けのための特徴抽出と比較して高速であるため，提案手法による高速化も期待できる．

2. 複数物体追跡の定式化

本章では物体追跡の問題を定式化する．$B_t = (\mathbf{b}_t^1, \mathbf{b}_t^2, \cdots)$ を，時刻 t におけるフレーム $\mathbf{o}_t$ 中の物体矩形とする．ここで，$\mathbf{b}_t^i$ はフレーム $\mathbf{o}_t$ 中の i 番目の物体矩形を示す．物

[1] 株式会社 KDDI 総合研究所
[2] 理化学研究所 情報統合本部 GRP
[3] 名古屋大学
[a)] ht-nishimura@kddi-research.jp
[b)] sa-komorita@kddi-research.jp
[c)] yasutomo.kawanishi@riken.jp
[d)] murase@nagoya-u.jp

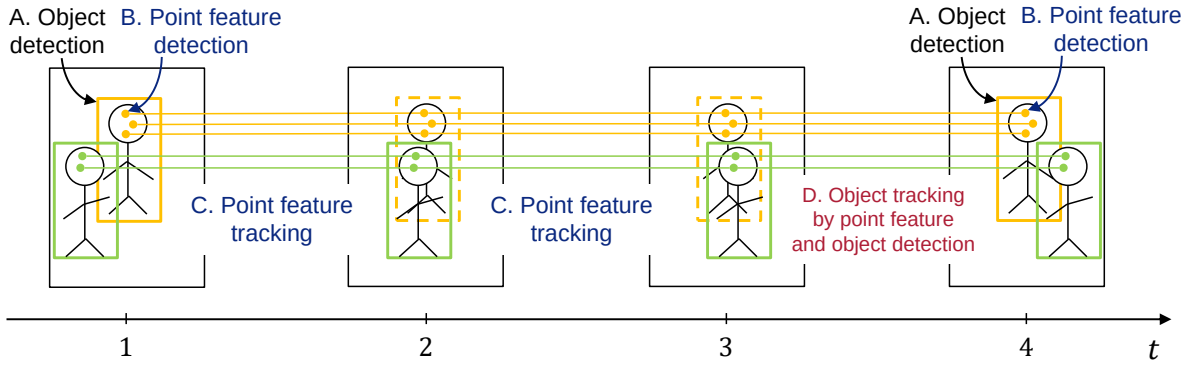


図 1: 提案手法による物体追跡.

体矩形は画像座標系において $\mathbf{b} = (x, y, w, h)$ で表し，x と y はそれぞれ矩形の左上の x 座標と y 座標を，w と h はそれぞれ矩形の幅と高さを示す．また，フレーム $\mathbf{o}_t$ 中の i 番目の物体矩形 $\mathbf{b}_t^i$ に対して，物体 ID z_t^i を付与したものを $\mathbf{a}_t^i = (\mathbf{b}_t^i, z_t^i)$ とし，これらをフレーム $\mathbf{o}_t$ 中で全て集めたものを $A_t = (\mathbf{a}_t^1, \mathbf{a}_t^2, \cdots)$ とする．物体 ID は，フレーム間で物体を対応付けすることによって求まる．つまり，物体追跡は，時系列画像（フレーム列）$O = \{\mathbf{o}_t \mid t \geq 1\}$ が与えられたとき，$\Omega = \{A_t \mid t \geq 1\}$ を求める問題と定式化できる．

3. 提案手法

提案手法では，基本的には特徴点によって追跡を行うが，L フレームごとに物体検出を行い，両者を統合する．特徴点は遮蔽する可能性の低い頭部周辺に配置し，追跡する．図 1 に提案手法による追跡の様子を示す．提案手法は，A. 物体検出，B. 特徴点検出，C. 特徴点追跡による物体追跡，D. 特徴点追跡と物体検出による物体追跡の 4 つのモジュールで構成される．初期フレームでは A. 物体検出と B. 特徴点検出を行う．それ以降は，毎フレーム C. 特徴点追跡による物体追跡を行う．そして，L フレームごとに（図ではフレーム 4），A. 物体検出，D. 特徴点追跡と物体検出による物体追跡，B. 特徴点検出（初期化）を行う．以下では，各モジュールについて説明する．

3.1 A. 物体検出

学習済みの物体検出器を用いて $B_t = (\mathbf{b}_t^1, \mathbf{b}_t^2, \cdots)$ を推定する．また，3.4 節の特徴点追跡と物体検出による物体追跡で用いるために，対応付けのための特徴量を抽出しておく．

3.2 B. 特徴点検出

推定した各矩形の中から特徴点を検出する．本論文では，遮蔽が生じにくい人物の頭部を中心として，その同心円上に特徴点を配置する．図 2 に頭部からの特徴点検出の様子を示す．頭部の位置は骨格推定手法によって求めることも可能であるが，本論文では処理の高速化のため，$(x + w/2, y + 0.1h)$ の位置にあると仮定する．そして，これを中心として半径 mr $(m = 1, 2, \cdots, M)$，角度 $2\pi n/K$ $(n = 0, 1, \cdots, K-1)$ で定義される点を特徴点とする．ここで，M, K はあらかじめ定められた自然数とする．また，r は $r = \alpha w/M$ として，矩形の幅 w に基づいて変化させる．ここで，α はあらかじめ定められた実数とする．こうして，i 番目の矩形 $\mathbf{b}_1^i$ から，特徴点集合 $P_1^i = (\mathbf{p}_1^{i1}, \mathbf{p}_1^{i2}, \cdots)$ を検出する．

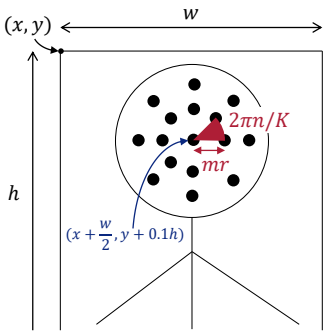


図 2: 頭部からの特徴点検出.

3.3 C. 特徴点追跡による物体追跡

特徴点追跡によって，矩形 $B_{t-1} = (\mathbf{b}_{t-1}^1, \mathbf{b}_{t-1}^2, \cdots)$ から $\widehat{B}_t = (\widehat{\mathbf{b}}_t^1, \widehat{\mathbf{b}}_t^2, \cdots)$ を推定する．以下では，i 番目の矩形 $\mathbf{b}_{t-1}^i$ から $\widehat{\mathbf{b}}_t^i$ を予測する方法を説明する．時刻 $t-1$ における特徴点集合 $P_{t-1}^i = (\mathbf{p}_{t-1}^{i1}, \mathbf{p}_{t-1}^{i2}, \cdots)$ が時刻 t でどこに移動したかを示すオプティカルフロー $\Delta_t^i = (\delta_t^{i1}, \delta_t^{i2}, \cdots)$ を推定する．矩形の位置 $(\widehat{x}_t^i, \widehat{y}_t^i)$ は，オプティカルフロー Δ_t^i の中央値 $\widetilde{\Delta}_t^i$ をもとの位置に加算することによって求める．

$$(\widehat{x}_t^i, \widehat{y}_t^i) = (x_{t-1}^i, y_{t-1}^i) + \widetilde{\Delta}_t^i \tag{1}$$

以上の処理を，全ての矩形に対して実行する．

3.4 D. 特徴点追跡と物体検出による物体追跡

L フレームごとに，特徴点追跡によって推定した矩形 $\widehat{B}_t$ と物体検出によって推定した矩形 B_t とを対応付ける．対応付けにはハンガリー法を用いる [2], [9]．ハンガリー法で用いるコスト行列は，矩形間の IoU（重複率）及び特徴量

表 1: 性能評価結果.

	Rcll [%] ↑	Prcn [%] ↑	MT ↑	PT	ML ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	Speed [fps] ↑
SORT [2]	**55.34**	**78.93**	**146**	263	**108**	3,884	3,099	37.05	22.02	15.44
SORT [2] + Flow (Proposed)	55.11	78.62	138	270	109	**1,927**	**2,329**	**38.38**	**22.31**	**18.43**
DeepSORT [9]	**55.34**	**78.93**	**145**	265	**107**	3,008	3,089	37.84	22.02	6.75
DeepSORT [9] + Flow (Proposed)	55.11	78.61	134	273	110	**1,666**	**2,318**	**38.61**	**22.31**	**10.92**

のコサイン距離の 2 種類を用いて個別に算出し，それらを用いて 2 段階で対応付けする．まず，特徴量による対応付けを行う．次に，対応付けられなかった矩形 $\{\widehat{\mathbf{b}}_t^i \in \widehat{B}_t\}$ と矩形 $\{\mathbf{b}_t^i \in B_t\}$ を対象に IoU による対応付けを行う．最後に，対応付けられなかった矩形 $\{\widehat{\mathbf{b}}_t^i \in \widehat{B}_t\}$ は追跡を終了し，対応付けられなかった矩形 $\{\mathbf{b}_t^i \in B_t\}$ は新たに追跡を開始する．なお，対応付けの際は，コストがしきい値以上の場合は対応付けないようにして誤対応を防ぐ．しきい値は矩形と特徴量で個別に設定し，それぞれ $\varepsilon_{\text{rect}}$，$\varepsilon_{\text{feat}}$ とする．

4. 実験

提案手法の有効性と効率を評価するため，物体追跡実験を行った．

4.1 実験条件

実験には，最も標準的な MOT16 データセット [8] を用いた．データセットは市街や店舗内において，固定あるいは移動カメラで撮影されている．フレームレートは 14–30fps，解像度は (640×480) – $(1,920 \times 1,080)$，時間は 18–60 秒，物体数は 25–125 と様々である．データセットのうち学習データセット 7 シーケンスを用いた（テストデータセットの正解データは公開されていないため）．なお，この学習データセットを用いて検出器や追跡器等の学習は行っていない．

ベースライン手法として，代表的な SORT [2]，DeepSORT [9] を用い，これらに提案手法を適用した．物体検出器として YOLOv4 [3] を用い，検出スコアが 0.2 以上の検出結果を用いた．提案手法において，オプティカルフローの算出には Lucas-Kanade 法 [7] を用いた．物体検出を行うフレーム間隔 L は 2 とした．特徴点検出のパラメータとして，$M = 3, K = 8, \alpha = 0.35$ とした．物体の対応付けのパラメータとして，$\varepsilon_{\text{rect}} = 0.7, \varepsilon_{\text{feat}} = 0.3$ とした．

評価指標として，再現率（Rcll），適合率（Prcn），動線のうち 80% 以上追跡できた物体の数（Mostly Tracked；MT），20–80% 追跡した物体の数（Partially Tracked；PT），20% 以下しか追跡できなかった物体の数（Mostly Lost；ML），ID スイッチの数（ID switch；IDs），追跡の途切れた数（Fragmentation；FM），複数物体追跡精度（Multiple Object Tracking Accuracy；MOTA），複数物体追跡精度（Multiple Object Tracking Precision；MOTP）を用いた [1]．また，1 フレームにかかる平均速度も計測した．CPU には Intel Core i7-7700K 4.20GHz，メモリには 32GB RAM，そして GPU には NVIDIA GeForce Titan X Pascal を用いた．

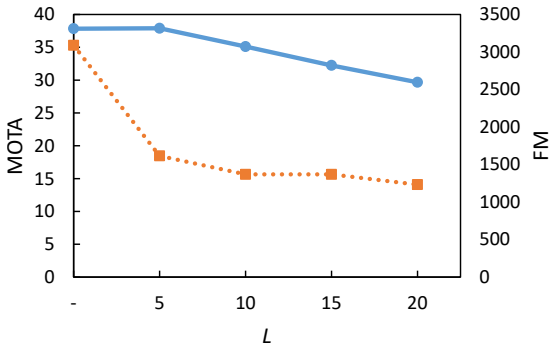


図 3: 物体検出を行うフレーム間隔（L）を増やした際の追跡精度の変化．横軸の”-” は，毎フレーム物体検出を行った場合，つまり DeepSORT [9] を示す．実線は MOTA，点線は FM を示す．

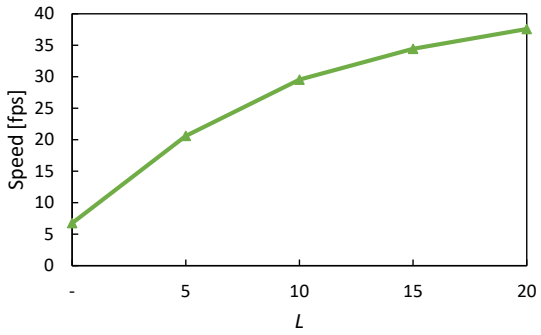
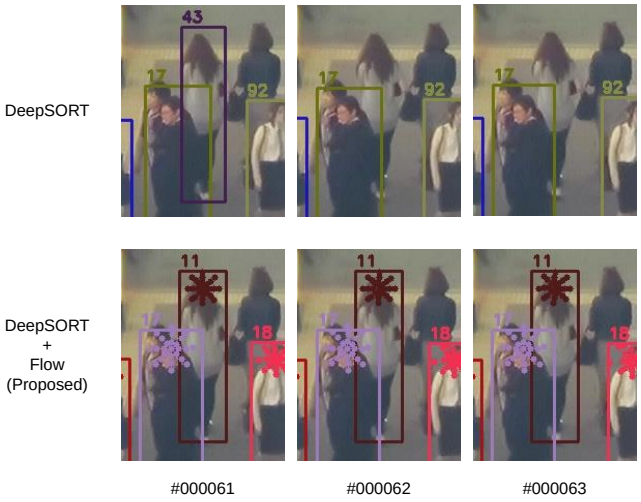


図 4: 物体検出を行うフレーム間隔（L）を増やした際の処理速度の変化．

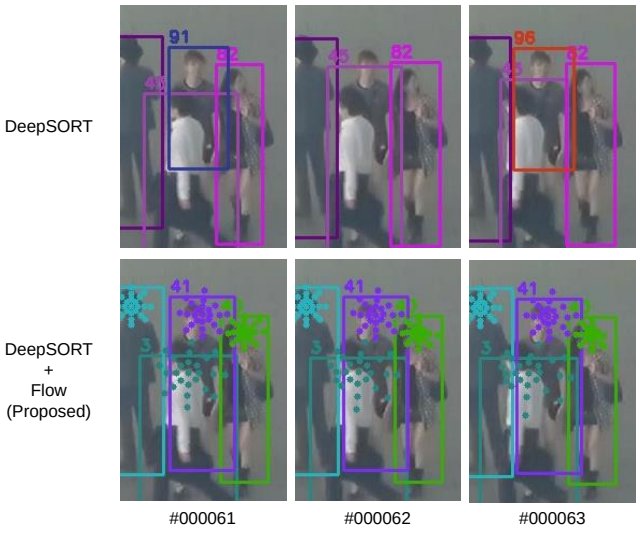
4.2 実験結果

表 1 に性能評価結果を示す．提案手法を SORT・DeepSORT に適用したいずれの場合においても，再現率・適合率を保持したまま，フラグメント数・ID スイッチ数が半分程度に削減されている．その結果，総合指標である MOTA も向上している．また，処理速度も向上している．

一方，フラグメント数が削減されたにもかかわらず，再現率は向上しなかった．これは，提案手法によって逆に検出ができなかった状況が増えたことを示唆している．これは，オプティカルフローによる追跡時は，新たに画像中に



(a) フラグメントを防止できた例．DeepSORT では ID:43 の追跡が途切れているのに対し，提案手法では ID:11 として追跡し続けている．



(b) ID スイッチを防止できた例．DeepSORT では ID が 91 から 96 にスイッチしているのに対し，提案手法では ID:41 として一意に追跡し続けている．

図 5: 追跡結果の例．市街の様子を高視点の固定カメラで撮影した MOT16-04 シーケンスから数フレーム抽出した．矩形上に記載の数字は推定した物体 ID を示す．

写り込んだ物体を追跡し始めることができないため，その時間分だけ未検出が発生しているからだと考えられる．

次に，物体検出を行うフレーム間隔を増やしていった際，追跡精度を保ったまま処理速度を削減できるかどうかを評価した．ベースライン手法として DeepSORT [9] を用いた．図 3 に，追跡精度の変化を示す．$L = 5$ の場合は，MOTA を同等に保ち，フラグメント数は半分程度に削減されている．$L \geq 10$ の場合は，MOTA は徐々に低下し，フラグメント数の削減量もなくなっていく．一方，図 4 に，処理速度の変化を示す．$L = 5$ の場合は約 3 倍高速化されるが，その後 $L \geq 10$ の場合は高速化度合は徐々に緩やかになっている．したがって，物体検出を行うフレーム間隔を増やしていった際，追跡精度を保ったまま処理が高速化されることが確認できた．

図 5 に追跡結果の例（$L = 5$）を示す．DeepSORT では遮蔽によって未検出が生じ，追跡が途切れているが，提案手法では途切れずに追跡できている．同様に，DeepSORT では遮蔽によって未検出が生じ，ID スイッチが発生しているが，提案手法では ID を保持したまま追跡できている．

5. 結論

本論文では，物体検出と，物体の一部の領域のみでも追跡可能な特徴点追跡を統合した複数物体追跡手法を提案した．提案手法では，遮蔽する可能性が低い頭部周辺の特徴点をオプティカルフローによって追跡し，物体全体の位置を推定する．また，ある特定のフレームにおいては検出器による物体検出を行い，特徴点によって推定した位置と対応付けることによって両者を統合する．実験では，標準的な MOT16 データセットを用いて，再現率・適合率を保持したまま，フラグメント数・ID スイッチ数が半分程度に削減され，かつ処理が高速化されていることを確認した．今後は，オプティカルフローによる追跡時でも，新たに画像中に写りこんだ物体を追跡開始できるような手法を検討する．

参考文献

[1] Bernardin, K. and Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics, *EURASIP Journal on Image and Video Processing*, Vol. 2008, No. 246309, pp. 1–12 (2008).

[2] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B.: Simple online and realtime tracking, *Proc. ICIP*, pp. 3464–3468 (2016).

[3] Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M.: YOLOv4: Optimal speed and accuracy of object detection, *arXiv:2004.10934* (2020).

[4] Bullinger, S., Bodensteiner, C. and Arens, M.: Instance flow based online multiple object tracking, *Proc. ICIP*, pp. 785–789 (2017).

[5] Choi, W.: Near-online multi-target tracking with aggregated local flow descriptor, *Proc. ICCV*, pp. 3029–3037 (2015).

[6] Kalal, Z., Mikolajczyk, K. and Matas, J.: Forward-backward error: Automatic detection of tracking failures, *Proc. ICPR*, pp. 2756–2759 (2010).

[7] Lucas, B. D. and Kanade, T.: An iterative image registration technique with an application to stereo vision, *Proc. IJCAI*, pp. 121–130 (1981).

[8] Milan, A., Leal-Taixé, L., Reid, I., Roth, S. and Schindler, K.: MOT16: A benchmark for multi-object tracking, *arXiv:1603.00831* (2016).

[9] Wojke, N., Bewley, A. and Paulus, D.: Simple online and realtime tracking with a deep association metric, *Proc. ICIP*, pp. 3645–3649 (2017).

[10] Zhang, Y., Wang, C., Wang, X., Zeng, W. and Liu, W.: FairMOT: On the fairness of detection and re-identification in multiple object tracking, *arXiv:2004.01888* (2020).