SDOF-TRACKER: FAST AND ACCURATE MULTIPLE HUMAN TRACKING BY SKIPPED-DETECTION AND OPTICAL-FLOW

Hitoshi Nishimura
KDDI Research, Inc.
ht-nishimura@kddi-research.jp

Satoshi Komorita
KDDI Research, Inc.
sa-komorita@kddi-research.jp

Yasutomo Kawanishi
RIKEN, Nagoya University, KDDI Research, Inc.
yasutomo.kawanishi@riken.jp

Hiroshi Murase
Nagoya University, KDDI Research, Inc.
murase@nagoya-u.jp

ABSTRACT

Multiple human tracking is a fundamental problem for scene understanding. Although both accuracy and speed are required in real-world applications, recent tracking methods based on deep learning have focused on accuracy and require substantial running time. This study aims to improve running speed by performing human detection at a certain frame interval because it accounts for most of the running time. The question is how to maintain accuracy while skipping human detection. In this paper, we propose a method that complements the detection results with optical flow, based on the fact that someone's appearance does not change much between adjacent frames. To maintain the tracking accuracy, we introduce robust interest point selection within human regions and a tracking termination metric calculated by the distribution of the interest points. On the MOT20 dataset in the MOTChallenge, the proposed SDOF-Tracker achieved the best performance in terms of the total running speed while maintaining the MOTA metric. Our code is available at https://anonymous.4open.science/r/sdof-tracker-75AE.

1 Introduction

Scene understanding from a video is one of the biggest challenges in computer vision. Humans are often the center of attention in a scene, and tracking them in a video is a fundamental problem. Multiple human tracking is the task of detecting the positions of multiple humans while maintaining their identities (IDs) over an image sequence. In real-world applications such as surveillance, autonomous vehicles, and marketing, both *accuracy* and *speed* need to be sufficiently high. In crowded scenes such as large stations, stadiums, and plazas, it often fails to detect humans, leading to ID switches. ID switch is a serious problem because it can lead to a misunderstanding of human behaviors. As well as the accuracy, the running speed is crucial in real-world applications. For example, the real-time recognition of suspicious behavior is essential in surveillance or for autonomous vehicles.

With the development of deep learning technology, the accuracy of human detection has been significantly improved, and the tracking-by-detection approach has become mainstream in recent years [1, 2, 3, 4, 5, 6]. The approach achieves human tracking by detecting humans with a human detector and associating the detection results using a similarity metric. The main advantage of this approach is that it is easy to determine the start and end of tracking even under occlusions and frame in/out. Most of methods in this approach detect humans by a deep learning-based detector and extract features from each region using another deep learning model. However, human detection and feature extraction take a lot of time; hence a rich computational resource is required for real-time tracking. Some methods tackle this problem by simultaneous human detection and feature extraction with a single deep learning model [7, 8, 9, 10, 11, 12]. However, there is a limitation on the degree to which speed can be increased without losing accuracy.

This study aims to resolve the speed-accuracy trade-off problem by bypassing every-frame human detection, which is a computationally heavy task, that is, we perform it at a certain interval. During the interval, human detection is skipped. We named this process *Skipped-Detection*. The question here is how to complement human detection in skipped frames.

We focus on the fact that someone's appearance is generally stable between adjacent frames. In such a situation, basic features are useful to associate humans between adjacent frames at a pixel level. Sparse optical flow [13], a type of optical flow, can estimate flow vectors at high speed by focusing on a small number of interest points. In this paper, we use sparse optical flow to complement between skipped human detections. The optical flow can also obtain detection results even in situations where the human detector misses someone.

Many tracking methods using optical flow have been proposed [14, 15, 16, 17, 18], and they aim to improve the tracking accuracy by human detection and optical flow at every frame. In contrast, we aim to maintain the tracking accuracy just with optical flow with the support of skipped detections. The problem is that optical flow itself cannot determine the start and termination of tracking. In this paper, we propose a novel human tracking method, which integrates Skipped-Detection and Optical-Flow, and name it *SDOF-Tracker*. In SDOF-Tracker, tracking by optical flow starts triggered by human detection and terminates based on the variance of interest points.

Moreover, the proposed SDOF-Tracker has the following features to solve two problems we found in the preliminary experiments: i) If setting interest points outside the human regions (background regions), humans are tracked inaccurately. To set robust interest points, instance segmentation is performed and interest points are set inside a limited human region. Since the instance segmentation is performed simultaneously with human detection, the running time does not increase much. Moreover, these points are set around the head, which is less likely to be occluded. ii) In crowded scenes, false negatives by the human detector frequently occur. It is a crucial problem because false negatives continue to occur in subsequent frames. To prevent false negatives, even if a target human is not detected, tracking by optical flow is continued for a while.

2 Related Work

In this section, we review related work on multiple human tracking based on detection and based on detection and optical flow.

2.1 Tracking Based on Detection

Tracking-by-Detection: A tracking-by-detection approach performs human tracking by detecting humans and associating the detection results using a similarity metric. SORT [19] calculates overlap between detections and applies the Hungarian algorithm [20] for data association. SORT is widely used in real-world applications due to its speed, but it may fail in crowded scenes due to lack of appearance features for data association. DeepSORT [1] is an extended version of SORT. It utilizes not only overlaps but also appearance features for data association. MHT-MAF [2] utilizes human action features for data association. LTSiam [3] is based on a Siamese network, which has tandem inputs and the same weights in both branches. MPNTrack [4], LPC_MOT [5], and GNNMatch [6] are based on a graph neural network, which captures the dependence of graphs via message passing. However, these methods [1, 2, 3, 4, 5, 6] take a lot of time for human detection and feature extraction, so real-time tracking is unrealistic.

Joint Detection and Tracking: While the tracking-by-detection approach has a two-stage structure of detection and data association, the latest approach jointly performs detection and data association in a single neural network. Tracktor [7] can detect the position in the next frame based on the existing detector without additional training. CenterTrack [8] uses a point-by-point heatmap to predict motion, which allows for association even when someone's movement between frames is large. SimpleReID [9] learns a re-identification model in an unsupervised manner. FairMOT [10] is a simple model that consists of two homogeneous branches to predict pixel-wise objectness scores and re-ID features. TBC [11] explicitly accounts for the object counts inferred from density maps and simultaneously solves detection and tracking. TransCenter [12] is a transformer-based architecture, which handles long-term complex dependencies by using an attention mechanism. However, these methods are limited in terms the degree to which speed can be increased without losing accuracy because there is a trade-off between speed and accuracy.

2.2 Tracking Based on Detection and Optical Flow

Human tracking methods based on detection and optical flow have been proposed in the past. Everingham *et al.*proposed a method that utilizes the portion of inlier trajectories over the outliers between face detections in order to cluster them [14]. Schikora *et al.*proposed a method that can deal with false positives and ID switches by using finite set statistics [15]. Fragkiadaki *et al.*proposed a method that jointly optimizes detectlet classification and clustering of optical flow trajectories [16]. Choi [17] proposed the aggregated local flow descriptor that can accurately measure the affinity between a pair of detections. Bullinger *et al.*proposed a method that exploits instance segmentation and predicts position and shape in the next frame by optical flows [18]. However, these methods require a lot of running time because they perform human detection in every frame and combine the detection result with optical flow.

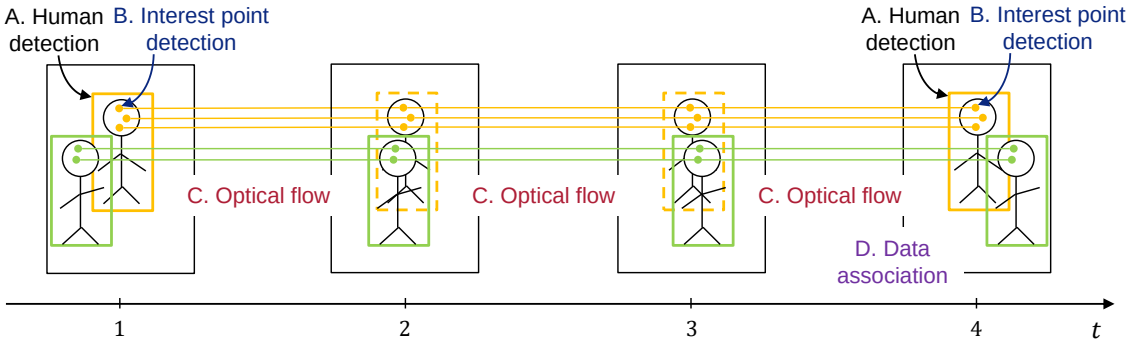


Figure 1: Human tracking by proposed SDOF-Tracker.

3 Proposed Method

First, we formulate the problem of human tracking. Second, we introduce the overall design of the proposed method, and next, we introduce each module.

3.1 Problem Formulation

We formulate the problem of human tracking. Let $B_t = (\mathbf{b}_t^1, \mathbf{b}_t^2, \cdots)$ be the bounding boxes in frame $\mathbf{o}_t$ at time t. Here, $\mathbf{b}_t^i$ denotes the i-th bounding box in frame $\mathbf{o}_t$. The bounding box is represented in the image coordinate system by $\mathbf{b} = (x, y, w, h)$, where x and y are the top-left x and y coordinates of the bounding box, respectively, and w and h are the width and height of the bounding box, respectively. For the i-th bounding box $\mathbf{b}_t^i$ in frame $\mathbf{o}_t$, let $\mathbf{a}_t^i = (\mathbf{b}_t^i, z_t^i)$ be the human ID z_t^i, and let $A_t = (\mathbf{a}_t^1, \mathbf{a}_t^2, \cdots)$ be the collection of all of these in frame $\mathbf{o}_t$. Human tracking can be formulated as the problem of finding $\{A_t \mid t \geq 1\}$ given a time series image $\{\mathbf{o}_t \mid t \geq 1\}$.

3.2 Overall Design

We aim to improve the running speed of tracking by using optical flow, which can estimate flow vectors at high speed. While the high speed tracking is performed using optical flow in every frame, detections are performed at a certain frame interval. To improve the robustness, interest points are set inside segmented regions and around the head, which is less likely to be occluded. Moreover, tracking by optical flow is continued for several frames even if a target human is not detected. This continuation can prevent false negatives, thus also preventing ID switches.

SDOF-Tracker has four modules: A. human detection, B. interest point detection, C. human tracking by optical flow, and D. data association. Figure 1 shows human tracking by SDOF-Tracker. It works in an online manner in that the tracking result is immediately available with each incoming frame. In the first frame, A. human detection and B. interest point detection are performed. After that, C. human tracking by optical flow is performed in every frame. Then, for each L frame (frame 4 in the figure), A. human detection, D. data association, and B. interest point detection (initialization) are performed. The details of each module are described from the next section.

3.3 A. Human Detection

This module estimates bounding boxes $D_t = (\mathbf{d}_t^1, \mathbf{d}_t^2, \cdots)$ using the trained human detector. In this work, we use Mask R-CNN [21] not only for detecting humans but also for performing instance segmentation to set robust interest points. In the first frame, human ID z_t^i is determined to be unique for each i.

3.4 B. Interest Point Detection

This module sets interest points inside bounding boxes for optical flow calculation. In the first frame, target bounding boxes are $D_t = (\mathbf{d}_t^1, \mathbf{d}_t^2, \cdots)$. On the other hand, in the frame $t(t \geq 2,\ t \mid L)$, target bounding boxes are $B_t = (\mathbf{b}_t^1, \mathbf{b}_t^2, \cdots)$. To improve the robustness, the interest points are set around the head because the head is less likely to be occluded. We assume that the y-axis of the head is located from y to $y + 0.3h$, where y denotes the top-left y coordinates of the bounding box. Furthermore, we use the instance segmentation result to limit the region of interest points to improve the robustness. The segmentation region is eroded using the morphological operator [22] to avoid setting interest points in the background regions. Interest points $P_t^i = (\mathbf{p}_t^{i1}, \mathbf{p}_t^{i2}, \cdots, \mathbf{p}_t^{iQ})$ are randomly sampled inside

the eroded segmentation region, where Q is a predetermined parameter. We do not use interest point detection methods to reduce the running time.

3.5 C. Human Tracking by Optical Flow

In this module, bounding boxes $\widehat{B}_t = (\widehat{\mathbf{b}}_t^1, \widehat{\mathbf{b}}_t^2, \cdots)$ are estimated from $B_{t-1} = (\mathbf{b}_{t-1}^1, \mathbf{b}_{t-1}^2, \cdots)$ by optical flow. In the following, we explain how to predict i-th bounding box $\widehat{\mathbf{b}}_t^i$ from the $\mathbf{b}_{t-1}^i$. First, the optical flow $\Delta_t^i = (\delta_t^{i1}, \delta_t^{i2}, \cdots, \delta_t^{iQ})$, which indicates where the interest point set $P_{t-1}^i = (\mathbf{p}_{t-1}^{i1}, \mathbf{p}_{t-1}^{i2}, \cdots, \mathbf{p}_{t-1}^{iQ})$ has moved, is estimated. Second, the location of the bounding box $(\widehat{x}_t^i, \widehat{y}_t^i)$ is obtained by adding the median of the optical flow.

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

Third, w_t^i and h_t^i are determined as the same value as time $t-1$ because they change very little between adjacent frames. Finally, human ID z_t^i inherits the same ID as time $t-1$.

However, the tracking may fail when the interest points track other humans or objects. In such cases, interest points often spread out rapidly. In this work, the determination of the termination of tracking is performed using the ratio of the variance of interest points between adjacent frames. The ratio is calculated by the variance of the interest point $P_{t-1}^i = (\mathbf{p}_{t-1}^{i1}, \mathbf{p}_{t-1}^{i2}, \cdots, \mathbf{p}_{t-1}^{iQ})$ in frame o_{t-1} and the interest point $P_t^i = (\mathbf{p}_{t-1}^{i1} + \delta_t^{i1}, \mathbf{p}_{t-1}^{i2} + \delta_t^{i2}, \cdots, \mathbf{p}_{t-1}^{iQ} + \delta_t^{iQ})$ in frame o_t as follows:

$$\alpha_t^i = \frac{\mathrm{var}(P_t^i)}{\mathrm{var}(P_{t-1}^i)} \tag{2}$$

Note that the interest points estimated by optical flow may have noise, so we remove such interest points before calculating the variances (*e.g.*Hotelling theory). Also, the tracking is terminated when the number of interest points becomes less than a predetermined threshold R.

3.6 D. Data Association

In this module, each bounding box $\{\widehat{\mathbf{b}}_t^i \in \widehat{B}_t\}$ estimated by optical flow is associated with each detection $\{\mathbf{d}_t^i \in D_t\}$ estimated by the human detector in each L frame. The data association has three important roles: the estimation of human ID, determination of the start of tracking, and determination of termination of tracking. The Hungarian algorithm [20] is used for the association. The cost matrix for the Hungarian algorithm is calculated using the IoU (Intersection over Union) between the detections and bounding boxes. When performing association, if the cost is larger than a predefined threshold ε, the bounding box is not associated with the detection to prevent false association. For each matching pair, human ID z_t^i and bounding box $\mathbf{b}_t^i$ are determined to have the same value as the matched detection $\mathbf{d}_t^i$. For each unmatched detection, tracking starts as a new human ID. For each unmatched bounding box, the tracking is terminated. However, in crowded scenes, bounding boxes tend to be unmatched due to false negatives. In this work, even if a bounding box is unmatched within M frames, the tracking is continued.

4 Experiments

In order to verify the effectiveness and efficiency of the proposed SDOF-Tracker, human tracking experiments were conducted.

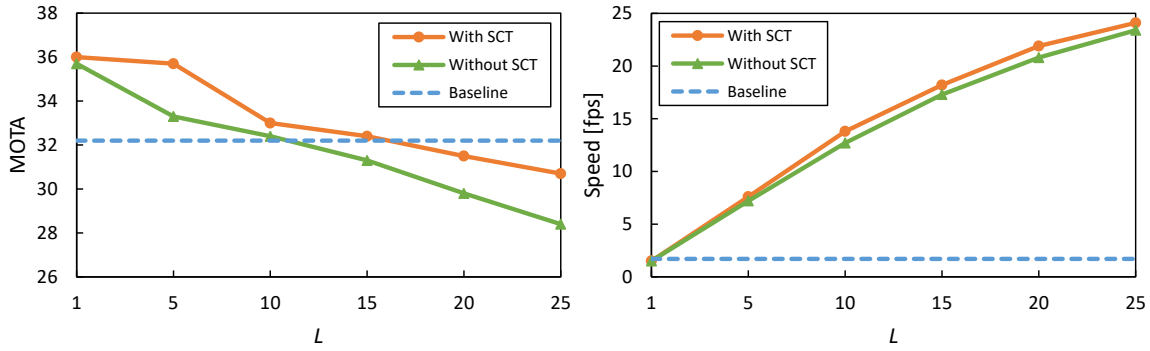
4.1 Experimental Conditions

Dataset: For the experiments, we used the latest challenging MOT20 dataset [23]. MOT20 was captured with a fixed or moving camera in a square, street, and shopping mall. The frame rate is $25\mathrm{fps}$, the resolution from (1173×880)–$(1,920 \times 1,080)$, the time from 17–133 seconds, and the total number of objects from 90–1121. We used 4 sequences in the validation dataset.

Evaluation Metric: The evaluation metrics include the number of objects tracked more than 80% of the flow line (Mostly Tracked; MT), the number of objects tracked less than 20% (Mostly Lost; ML), Recall (Rcll), Precision (Prcn), ID switches (IDsw), Fragmentation (Frag), and Multiple Object Tracking Accuracy (MOTA). MOTA is a widely used and comprehensive metric that combines three error sources (false negative, ID switch, and false positive). We also measured the average speed per 1 frame. We used an Intel Core i7-7700K 4.20GHz CPU, 32GB RAM, and an NVIDIA GeForce Titan X Pascal GPU.

S	C	T	MT ↑	ML ↓	Rcll [%] ↑	Prcn [%] ↑	IDsw ↓	Frag ↓	MOTA ↑
			228	727	40.7	86.4	13,893	18,319	33.3
✓			226	726	40.7	**86.6**	13,738	17,932	33.4
	✓		**291**	632	44.8	83.6	9,837	15,422	35.3
		✓	220	731	40.5	86.5	14,650	18,968	33.1
✓	✓	✓	**291**	**623**	**44.9**	84.1	**9,578**	**14,856**	**35.7**

Table 1: Ablation study. S: Segmentation, C: Continuation, T: Termination.



(a) MOTA. (b) Speed.

Figure 2: Change in tracking accuracy and speed with increasing frame interval (L) for human detection.

Implementation Details: As the baseline method, human detection and feature extraction are performed in every frame. We used Mask R-CNN [21] for the human detector, and it was trained using MS COCO [24]. The same human detection result was used for the baseline and the proposed SDOF-Tracker. The threshold of human detection was set to 0.2 The following are the parameters for SDOF-Tracker. The frame interval for human detection was set to $L = 5$. The frame length for tracking continuation was set to $M = 10$. For the optical flow calculation, the Lucas-Kanade method [13] was used. The max and minimum number of interest points were set to $Q = 10$ and $R = 3$, respectively. The parameters for human association were set to $\varepsilon = 0.7$.

4.2 Ablation Study

In this section, we verify the effectiveness of each of the three factors in the SDOF-Tracker, the segmentation for point extraction (S), tracking continuation (C), and tracking termination using interest points (T). We set the frame interval for human detection to $L = 5$.

Table 1 shows the performances with the three factors combined. First, let us explain the segmentation for point extraction. As expected, the precision and the number of ID switches improved, and as a result, MOTA improved. Second, let us explain the tracking continuation. As expected, recall significantly improved. As a result, MT, ML, the number of ID switches, the number of fragmentations, and MOTA also improved. Finally, let us explain the tracking termination using interest points. Although the precision improved, the number of ID switches and MOTA degraded. This implies that interest points are not accurately set inside the human region by segmentation. It is considered that the ratio of the variance of interest points is appropriately calculated because their points are not accurately set inside the human region. The combination of all of the above achieved the highest performance for almost all metrics (MT, ML, Recall, IDsw, Frag, and MOTA). In the combination, tracking termination also contributed. This implies that interest points are accurately set inside the human region by segmentation.

4.3 Analysis of Accuracy and Speed

We evaluated whether the running speed can be improved while maintaining the tracking accuracy when the frame interval (L) for human detection is increased. Speed includes the time required for human detection. For the baseline method, human detection is performed in every frame, and it is equivalent to DeepSORT [1]. On the other hand, the SDOF-Tracker performs human detection in every L frame. In SDOF-Tracker, we evaluated whether segmentation, tracking continuation, and tracking termination are performed or not. In order to compare the accuracy fairly, we use

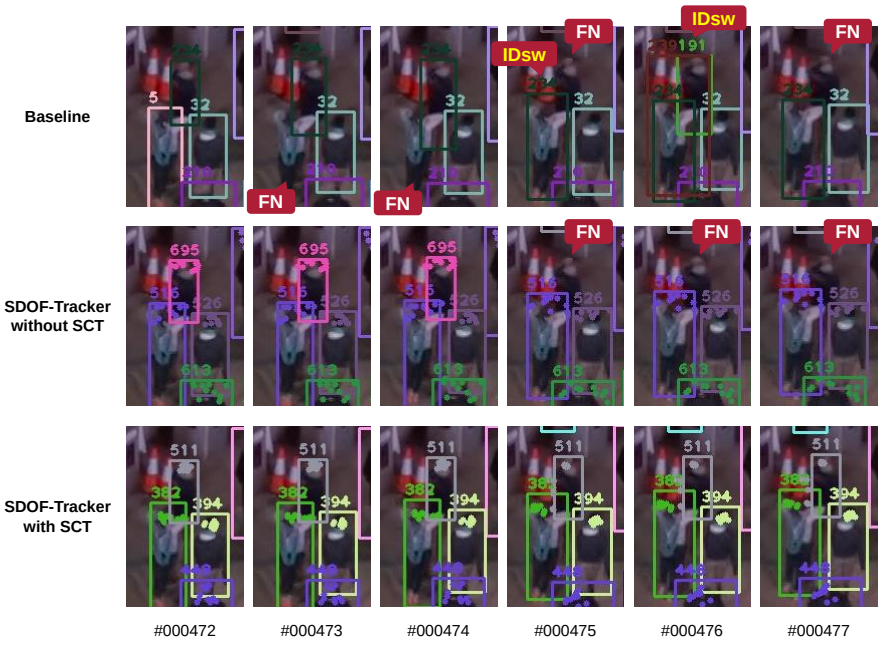


Figure 3: Cropped example of the tracking result using the baseline and SDOF-Tracker.

Figure 4: Example of tracking result using SDOF-Tracker

the same detection result using Mask-RCNN in both with/without SCT. Therefore, the segmentation time is included when evaluating "without SCT", but the actual speed without segmentation is even faster.

Figure 2a shows the change in the tracking accuracy. In "with SCT", MOTA is almost the same when $L = 1$ as when $L = 5$. Then, MOTA decreases when $L \geq 5$, and is almost the same when $L = 15$ as the baseline. By contrast, in "without SCT", MOTA decreases when $L \geq 1$, and is almost the same when $L = 10$ as the baseline. On the other hand, Figure 2b shows the change in running speed. As L increases, the running speed increases in both "with/without STC". The speed improvement rate according to L is higher with STC than without STC. This is because the frequency of the termination is increased and the number of tracked humans is decreased as L increases. Thus, SDOF-Tracker with SCT can improve the running speed while maintaining the tracking accuracy.

	Rcll [%] ↑	Prcn [%] ↑	IDsw ↓	MOTA ↑	Speed [fps] ↑ (Tracking)	Speed [fps] ↑ (Detection)	Speed [fps] ↑ (Total)
SDOF-Tracker	58.0	84.6	3,532	46.7	19.2	38.0	12.8
SORT [19]	48.8	90.2	4,470	42.7	57.3	7.6	6.7
LTSiam [3]	58.5	84.0	4,509	46.5	30.3	7.6	6.1
MPNTrack [4]	61.1	94.9	1,210	57.6	6.5	7.6	3.5
TBC [11]	62.3	89.5	2,449	54.5	5.6	7.6	3.2
SimpleReID [9]	55.3	97.8	2,178	53.6	1.3	7.6	1.1
Tracktor [7]	54.3	97.6	1,648	52.6	1.2	7.6	1.0
TransCenter [12]	71.4	88.3	4,493	61.0	1.0	7.6	0.9
LPC_MOT [5]	58.8	96.3	1,562	56.3	0.7	7.6	0.6
mfi_tst [25]	66.6	90.5	1,919	59.3	0.5	7.6	0.5
GNNMatch [6]	56.8	96.9	2,038	54.5	0.1	7.6	0.1

Table 2: MOTChallenge result on MOT20 dataset. The result is cited from MOTChallenge web page[1] (Our entry name on the web page is "FlowTracker").

4.4 Tracking Examples

Figure 3 shows a cropped example of the tracking result using the baseline and SDOF-Tracker. This is a scene where three people are walking toward the back. In the baseline, ID switches (IDsw) occur due to false negatives (FN). In SDOF-Tracker, human detection is performed in frame 475 because we set $L = 5$. In SDOF-Tracker without SCT, the false negatives are prevented in frame 473 and 474 due to tracking by optical flow. However, the other false negatives remain. This is because the optical flow cannot start tracking when the false negative occurs in frame 475, which is a chance for human detection. On the other hand, in SDOF-Tracker with SCT, all false negatives and ID switches are prevented due to tracking continuation in frame 475. Moreover, interest points are accurately set on the regions of human heads. Figure 4 shows an example of the tracking result using SDOF-Tracker. Even though this is a very crowded scene, most humans are accurately tracked because interest points are set on the heads.

4.5 MOTChallenge Result

We compared the SDOF-Tracker to the state-of-the-art methods in MOTChallenge[1] on the MOT20 dataset. We compared the performance with methods which have been published in the literature. We use the public detection results of MOTChallenge to compare both accuracy and speed fairly. The speed of detection (7.6 fps) was cited from the literature [26]. Using this speed, the speed of SDOF-Tracker was estimated as 38.0 fps because the frame interval for human detection was set to $L = 5$. Note that SDOF-Tracker did not use a GPU for human tracking. Since the public detection results do not include segmentation results, we do not limit regions for setting interest points. Table 2 shows the MOTChallenge result. SDOF-Tracker achieved the best performance in terms of the total speed. Nevertheless, MOTA was better than SORT and LTSiam.

5 Conclusion

In this paper, we proposed SDOF-Tracker, a fast and accurate human tracking method using skipped-detection and optical-flow. In SDOF-Tracker, tracking by optical flow starts triggered by human detection and ends based on the variance of interest points. To maintain accuracy, we introduced robust interest point selection within human regions and a tracking termination metric calculated by the distribution of the interest points. In the experiments, we confirmed that SDOF-Tracker can improve the running speed while maintaining the tracking accuracy when the frame interval for human detection is increased. Moreover, SDOF-Tracker achieved the best performance in terms of the total speed (12.8 fps) while maintaining MOTA (46.7) on the MOT20 dataset in the MOTChallenge. In the future, we will develop a method that can dynamically change the frame interval of human detection.

References

[1] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings of the 24th IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017.

[1] https://motchallenge.net

[2] Hitoshi Nishimura, Kazuyuki Tasaka, Yasutomo Kawanishi, and Hiroshi Murase. Multiple human tracking with alternately updating trajectories and multi-frame action features. *ITE Transactions on Media Technology and Applications*, 8(4):269–279, 2020.

[3] Oliver Urbann, Oliver Bredtmann, Maximilian Otten, Jan-Philip Richter, Thilo Bauer, and David Zibriczky. Online and real-time tracking in a surveillance scenario. *Computing Research Repository arXiv Preprint arXiv:2106.01153*, 2021.

[4] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6247–6257, 2020.

[5] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *Proceedings of the 2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[6] Ioannis Papakis, Abhijit Sarkar, and Anuj Karpatne. GCNNMatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. *Computing Research Repository arXiv Preprint arXiv:2010.00067*, 2021.

[7] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the 17th IEEE International Conference on Computer Vision (ICCV)*, pages 941–951, 2019.

[8] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *Computing Research Repository arXiv Preprint arXiv:2004.01177*, 2020.

[9] Shyamgopal Karthik, Ameya Prabhu, and Vineet Gandhi. Simple unsupervised multi-object tracking. *Computing Research Repository arXiv Preprint arXiv:2006.02609*, 2020.

[10] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *Computing Research Repository arXiv Preprint arXiv:2004.01888*, 2020.

[11] Weihong Ren, Xinchao Wang, Jiandong Tian, Yandong Tang, and Antoni B. Chan. Tracking-by-counting: Using network flows on crowd density maps for tracking multiple targets. *IEEE Transactions on Image Processing*, 30:1439–1452, 2021.

[12] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. TransCenter: Transformers with dense queries for multiple-object tracking. *Computing Research Repository arXiv Preprint arXiv:2103.15145*, 2021.

[13] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 121–130, 1981.

[14] Mark Everingham, Josef Sivic, and Andrew Zisserman. Taking the bite out of automated naming of characters in tv video. *Image and Vision Computing*, 27(5):545–559, 2009.

[15] Marek Schikora, Wolfgang Koch, and Daniel Cremers. Multi-object tracking via high accuracy optical flow and finite set statistics. In *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1409–1412, 2011.

[16] Katerina Fragkiadaki, Weiyu Zhang, Geng Zhang, and Jianbo Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *Proceedings of the 12th European Conference on Computer Vision (ECCV) Part 5, Lecture Notes in Computer Science*, pages 552–565, 2012.

[17] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV)*, pages 3029–3037, 2015.

[18] Sebastian Bullinger, Christoph Bodensteiner, and Michael Arens. Instance flow based online multiple object tracking. In *Proceedings of the 24th IEEE International Conference on Image Processing (ICIP)*, pages 785–789, 2017.

[19] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Proceedings of the 23rd IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[20] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[22] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[23] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *Computing Research Repository arXiv Preprint arXiv:2003.09003*, 2020.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV) Part 5, Lecture Notes in Computer Science*, pages 740–755, 2014.

[25] Jieming Yang, Hongwei Ge, Jinlong Yang, Yubing Tong, and Shuzhi Su. Online multi-object tracking using multi-function integration and tracking simulation training. *Applied Intelligence*, pages 1–21, 2021.

[26] Julian True and Naimul Khan. Motion vector extrapolation for video object detection. *Computing Research Repository arXiv Preprint arXiv:2104.08918*, 2021.