PAPER

SDOF-Tracker: Fast and Accurate Multiple Human Tracking by Skipped-Detection and Optical-Flow

Hitoshi NISHIMURA†a), Satoshi KOMORITA†b), *Members*, Yasutomo KAWANISHI†,††,†††c), *Senior Member*, *and* Hiroshi MURASE†,†††d), *Fellow*

SUMMARY Multiple human tracking is a fundamental problem in understanding the context of a visual scene. Although both accuracy and speed are required in real-world applications, recent tracking methods based on deep learning focus on accuracy and require a substantial amount of running time. We aim to improve tracking running speeds by performing human detections at certain frame intervals 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 interpolates the detection results by using an optical flow, which is 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 detection within the human regions and a tracking termination metric defined by the distribution of the interest points. On the MOT17 and MOT20 datasets in the MOTChallenge, the proposed SDOF-Tracker achieved the best performance in terms of total running time while maintaining the MOTA metric. Our code is available at https://github.com/hitottiez/sdof-tracker.

key words: *tracking, detection, optical flow*

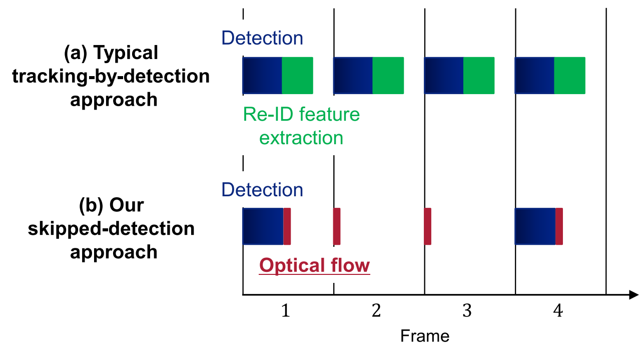


Fig. 1 Difference in the use of computational resources between the typical approach and our approach. Our approach can use computational resources effectively and achieve an average running speed sufficient for real-time tracking.

1. Introduction

Understanding the context of a scene in 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 the fundamental objective. Multiple human tracking is the task defined as detecting the positions of multiple humans while maintaining their identities (IDs) over an image sequence. In real-world applications such as surveillance, tracking needs to be performed in real-time with high accuracy. In crowded scenes such as large stations, stadiums and plazas, there is often a failure to detect humans accurately, leading to ID switches. An ID switch is a serious problem because it can lead to a misunderstanding of human behavior. In addition to the need for accurate tracking, real-time tracking is crucial for many real-world applications. For example, the real-time recognition of suspicious behavior is essential in surveillance.

With the development of deep learning technology, the accuracy of human detection has been significantly improved (e.g., Faster R-CNN [1], Mask R-CNN [2] and YOLOv4 [3]), and tracking-by-detection has become the mainstream approach in recent years [4]–[16]. The approach implements human tracking by detecting humans with a human detector and making associations with 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 a tracking event even with occlusions and frame ins/outs. Most methods using this approach detect humans by a deep learning-based detector and extract the re-identification (re-ID) features from each region using another deep learning model. However, human detection and re-ID feature extraction take a considerable amount of time; hence, an efficient computational resource is required for real-time tracking, as shown in Fig. 1 (a). Some methods tackle this problem by conducting human detection and re-ID feature extraction simultaneously with a single deep learning model [17]–[22]. However, their methods have a limitation that the speed can not be increased while keeping accuracy.

We aim to improve the tracking running speed by bypassing every-frame human detection, which is a computationally heavy task; that is, we perform the task at a certain interval. Figure 1 (b) shows our approach. During the interval, human detection is skipped and interpolated by faster processing. We name this process *Skipped-Detection*. This enables the effective use of computational resources and

Manuscript received February 10, 2022.
Manuscript revised June 7, 2022.
Manuscript publicized August 1, 2022.
†The authors are with KDDI Research, Inc., Fujimino-shi, 356–8502 Japan.
††The author is with Guardian Robot Project, RIKEN, Kyoto, 619–0288 Japan.
†††The author is with Graduate School of Informatics, Nagoya University, Nagoya-shi, 464–8601 Japan.
a) E-mail: ht-nishimura@kddi.com
b) E-mail: sa-komorita@kddi.com
c) E-mail: yasutomo.kawanishi@riken.jp
d) E-mail: murase@nagoya-u.jp
DOI: 10.1587/transele.2022EDP7022

achieves an average running speed sufficient for real-time tracking. The question to be addressed is how to interpolate human detection in skipped frames. We focus on the fact that someone's appearance is generally stable between adjacent frames. In such a situation, primitive features are useful to associate humans between adjacent frames at a pixel level. Sparse optical flow [23] can estimate flow vectors at a high speed by focusing on a small number of interest points. In this paper, we use sparse optical flow to interpolate between skipped human detections, as shown in Fig. 1 (b). Additionally, the optical flow can also estimate target locations even in situations where the human detector misses someone.

Many tracking methods using optical flow have been proposed [10]–[14], and they attempt to improve the tracking accuracy by human detection and optical flow at every frame. In contrast, we aim to maintain the tracking accuracy only with optical flow with the support of skipped detections. The problem is that the optical flow itself cannot determine the start and termination of the tracking. In this paper, we propose a novel human tracking method that integrates Skipped-Detection and Optical-Flow, and we name it *SDOF-Tracker*. In the SDOF-Tracker, tracking by optical flow is triggered by human detection and terminated based on the variance of the interest points.

Moreover, the proposed SDOF-Tracker can prevent false negatives even if the human detector misses someone. To prevent false negatives, even if a human target is not detected, tracking by optical flow is continued for a while. Additionally, to set robust interest points for optical flow, they are set inside a limited human region obtained by an instance segmentation.

2. Related Work

In this section, we review the related work on multiple human tracking in terms of the tracking-by-detection approach and the faster approaches.

2.1 Tracking-by-Detection Approach

A tracking-by-detection approach performs human tracking by detecting humans and associating the detection results using a similarity metric. DeepSORT [4] utilizes the overlap between bounding boxes and the re-ID features extracted from the appearance and applies the Hungarian algorithm [24] for data association. The Kalman filter is applied for robust tracking. MHT-MAF [5] utilizes human action features for data associations. LTSiam [6] is based on a Siamese network, which has tandem inputs and the same weights in both branches. MPNTrack [7], LPC_MOT [8], and GNNMatch [9] are based on a graph neural network, which captures the dependence of graphs via message passing. However, these methods require considerable time for human detection and re-ID feature extractions for data associations, so a substantial computational resource is required for real-time tracking.

Many tracking methods based on optical flow have been proposed to improve tracking accuracies. Everingham *et al.* [10] proposed a method that utilizes the portion of the inlier trajectories over the outliers that are between the face detections to cluster them. Schikora *et al.* [11] proposed a method that could deal with false positives and ID switches by using finite set statistics. Fragkiadaki *et al.* [12] proposed a method that jointly optimizes detectlet classification and the clustering of optical flow trajectories. Choi [13] proposed an aggregated local flow descriptor that could accurately measure the affinity between a pair of detections. Bullinger *et al.* [14] proposed a method that exploits instance segmentation and predicts the positions and shapes in the next frame by optical flows. However, these methods require a considerable amount of running time because they perform human detection in every frame and combine the detection results with the optical flow.

2.2 Faster Approach

While the tracking-by-detection approach has a two-stage structure for detection and data association, the latest approaches jointly perform them in a single neural network for fast and accurate tracking. Tracktor [17] can detect the position in the next frame based on the existing detector without additional training. SimpleReID [19] learns a re-identification model in an unsupervised manner. TBC [21] explicitly accounts for the object counts inferred from density maps and simultaneously performs detection and tracking. TransCenter [22] is a transformer-based architecture that handles long-term complex dependencies by using an attention mechanism. Although most of these methods are faster than the previous tracking-by-detection approach, the speedup is limited because they perform human detection in every frame.

Other approaches do not utilize appearance features for data association. SORT [15] and IOU Tracker [16] utilize only the overlap between the bounding boxes and are widely used in real-world applications due to their speed. However, these methods may fail in crowded scenes due to a lack of appearance features.

Unlike human tracking, AdaVP [25] was proposed to make human detection faster by optical flow in real-time detection. Since the method does not care about consistent human IDs, the method cannot be directly applied to multiple human tracking applications discussed in this paper.

3. Proposed Method

In contrast to conventional methods, the proposed SDOF-Tracker does not perform human detection in every frame and employs only optical flow to interpolate the detection results. In this section, we first define symbols in the human tracking. Second, we introduce the overall design of the SDOF-Tracker, and then we explain each step.

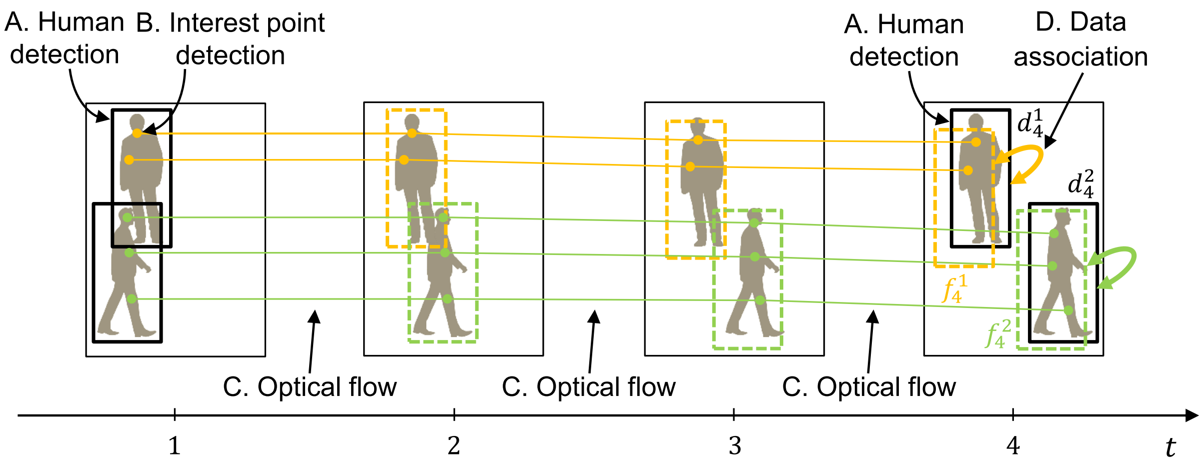


Fig. 2 Human tracking by the proposed SDOF-Tracker.

3.1 Symbol Definition

We define symbols in the human tracking. Let $B_t = (\mathbf{b}_t^1, \mathbf{b}_t^2, \ldots)$ 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 pair of the bounding box $\mathbf{b}_t^i$ and its tracklet ID z_t^i. Let $A_t = (\mathbf{a}_t^1, \mathbf{a}_t^2, \ldots)$ 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 speeds. While high-speed tracking is performed using optical flow in every frame, detections are just performed at a certain frame interval. To improve the robustness, interest points are set inside segmented regions. Moreover, tracking by optical flow is continued after several frames even if the human detector misses a human target. This continuation can prevent false negatives, thus also preventing ID switches.

3.3 Details of Each Step

SDOF-Tracker has four steps: A. human detection, B. interest point detection, C. human tracking by optical flow, and D. data association. Figure 2 shows human tracking by the 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 executed. From the frame, C. human tracking by optical flow is executed for L frames. After that (frame 4 in the figure), A. human detection, D. data association, and B. interest point detection (initialization) are executed. The details of each step are described below.

A. Human Detection

This step estimates bounding boxes $D_t = (\mathbf{d}_t^1, \mathbf{d}_t^2, \ldots)$ using the trained human detector, where $\mathbf{d} = (x, y, w, h)$. In this work, we use a multitask network that not only detects humans but also performs instance segmentations to set robust interest points. In the first frame, bounding box b_t^i is determined to have the same value as d_t^i and ID z_t^i is determined to be unique for each i.

B. Interest Point Detection

This step sets interest points inside the bounding boxes for optical flow calculations. In the first frame, the target bounding boxes are $D_t = (\mathbf{d}_t^1, \mathbf{d}_t^2, \ldots)$. On the other hand, in frame $\mathbf{o}_t$ ($t \geq 2$), the target bounding boxes are $B_t = (\mathbf{b}_t^1, \mathbf{b}_t^2, \ldots)$. To improve the robustness of C. human tracking by optical flow, we use the instance segmentation result to limit the region of the interest points. The segmentation mask is represented as a binary image that indicates whether the human region. The segmentation region is eroded using the morphological operator [26] to avoid setting interest points in the background regions. Each pixel is set to 1 if all the pixels in the kernel region have a pixel value of 1; otherwise, they are set to 0. The points of interest $P_t^i = (\mathbf{p}_t^{i1}, \mathbf{p}_t^{i2}, \ldots, \mathbf{p}_t^{iQ})$ are randomly sampled inside the eroded segmentation region, where Q is a predetermined parameter.

C. Human Tracking by Optical Flow

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

is obtained by adding the median of the optical flow.

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

Third, the width and height of the bounding box, 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, when $t \neq 1 + nL$, $\mathbf{b}_t^i$ is determined to have the same value as $\mathbf{f}_t^i$ and tracklet ID z_t^i inherits the same ID as time $t-1$. Otherwise, $\mathbf{b}_t^i$ and z_t^i are determined by data association using $\mathbf{f}_t^i$ as described in the next Sect. D.

However, tracking may fail when the points of interest track other humans or objects. In such cases, interest points often spread out rapidly. In this work, the termination of tracking is determined using the ratio of the variance of the interest points between the adjacent frames. The ratio is calculated by the variance of the interest point $P_{t-1}^i = (\mathbf{p}_{t-1}^{i1}, \mathbf{p}_{t-1}^{i2}, \dots, \mathbf{p}_{t-1}^{iQ})$ in frame $\mathbf{o}_{t-1}$ and the interest point $P_t^i = (\mathbf{p}_{t-1}^{i1} + \delta_t^{i1}, \mathbf{p}_{t-1}^{i2} + \delta_t^{i2}, \dots, \mathbf{p}_{t-1}^{iQ} + \delta_t^{iQ})$ in frame $\mathbf{o}_t$ 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. For removing noise, we use Hotelling theory, which Hotelling theory is a fundamental method of outlier determination that assumes that data is generated with a normal distribution. Additionally, the tracking is terminated when the number of interest points becomes less than a predetermined threshold R.

D. Data Association

In this step, each bounding box $\mathbf{f}_t^i \in F_t$ estimated by optical flow is associated with each detection $\mathbf{d}_t^j \in D_t$ estimated by the human detector in each L frame. The data association has three important roles: the estimation of a tracklet ID, determination of the start of tracking, and determination of the termination of the tracking. The Hungarian algorithm [24] is used for the association. The cost matrix for the Hungarian algorithm is calculated using the intersection over union (IoU) between the detections and bounding boxes. When performing an association, if the cost is larger than a predefined threshold ε, the bounding box is not associated with the detection to prevent a false association.

For each matching pair, bounding box $\mathbf{b}_t^i$ is determined to have the same value as $\mathbf{d}_t^j$. Tracklet ID z_t^i is determined to have the same value as z_{t-1}^i corresponding to $\mathbf{f}_t^i$. For each unmatched detection, tracking starts with a new tracklet 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

To verify the effectiveness and efficiency of the proposed SDOF-Tracker, we conducted human tracking experiments using two major datasets, MOT17 and MOT20.

4.1 Experimental Conditions

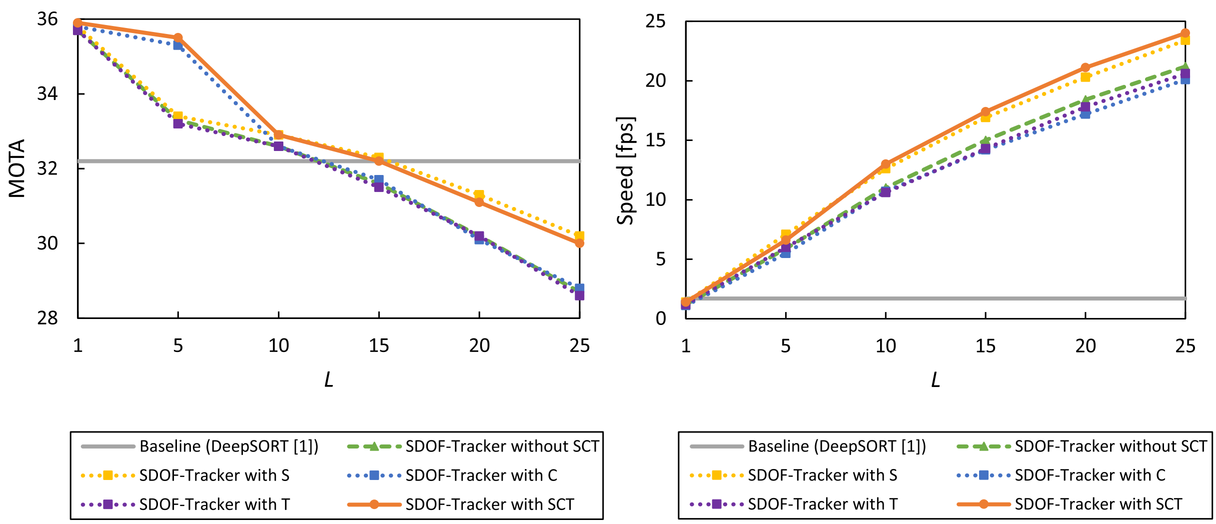
Dataset: For the experiments, we used two major datasets, MOT17 [27] and MOT20 [28]. They were captured with a fixed or moving camera in a square, street and shopping mall. MOT17 includes less dense crowds but more diverse scenarios than MOT20. For MOT17, the frame rate is from 14 to 25fps, the resolution is from (640×480) to $(1,920 \times 1,080)$, the time is from 15 to 85 seconds, and the total number of objects is from 24 to 222. We used 21 sequences in the test set. On the other hand, for MOT20, the frame rate is 25fps, the resolution is from $(1,173 \times 880)$ to $(1,920 \times 1,080)$, the time is from 17 to 133 seconds, and the total number of objects is from 90 to 1,121. We used 4 sequences in the training set. The training set was used for evaluation (Sect. 4.2, 4.3, and 4.4) because the test set does not have ground truth. Note that any model was not trained.
Evaluation Metric: The evaluation metrics include the number of objects tracked for more than 80% of the flow line (mostly tracked; MT), the number of objects tracked for less than 20% (mostly lost; ML), recall (Rcll), precision (Prcn), ID switches (IDsw), fragmentation (Frag), and multiple object tracking accuracy (MOTA) [29]. 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 4.20 GHz CPU, 32 GB RAM, and an NVIDIA GeForce Titan X Pascal GPU.
Implementation Details: As the baseline method, human detection and re-ID feature extraction are performed in every frame. We used Mask R-CNN [2] for the human detector, and it was trained using MS COCO [30]. 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 values of which were set by preliminary experiments. The frame interval for human detection was set to $L = 5$. The frame length for tracking continuation was set to $M = 10$. For the segmentation region eroding, a 2×2 kernel was applied two times. For the optical flow calculation, the Lucas-Kanade method [23] was used. The window size was set to 15×15 and the height of image pyramid was set to 2. The maximum and minimum numbers of interest points were set to $Q = 10$ and $R = 3$, respectively. The parameter for human association was 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

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

	Pattern	S	C	T	MT ↑	ML ↓	Rcll [%] ↑	Prcn [%] ↑	IDsw ↓	Frag ↓	MOTA ↑
Baseline (DeepSORT [1])	-	-	-	-	236	730	40.7	86.4	13,731	18,198	33.3
SDOF-Tracker	1	✓			229	724	40.8	**86.5**	13,622	17,796	33.4
	2		✓		300	621	44.8	83.6	9,709	15,157	35.3
	3			✓	229	731	40.7	86.4	13,967	18,318	33.2
	4	✓	✓		**303**	619	**44.9**	83.8	9,574	**14,756**	**35.5**
	5	✓		✓	224	728	40.7	**86.5**	14,013	18,098	33.3
	6		✓	✓	299	616	44.8	83.7	9,685	15,084	35.4
	7	✓	✓	✓	**303**	**615**	**44.9**	83.9	**9,537**	14,770	**35.5**



(a) MOTA. (b) Speed.

Fig. 3 Change in tracking accuracy and speed with increasing frame interval (L) for human detection.

point extraction (S), tracking continuation (C) and the tracking termination using interest points (T). They are explained in B. interest point detection, D. data association, and C. human tracking by the optical flow in Sect. 3.3, respectively. We used the MOT20 dataset for the experiment.

Table 1 shows the performances with the three factors combined. First, let us explain the segmentation for the point extraction. As expected, the precision improved, and as a result, MOTA improved. Second, let us explain the tracking continuation. As expected, the 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. It is speculated that the ratio of the variance of interest points is not appropriately calculated because their points are not accurately set inside the human region. The combinations of all of the processes above achieved the highest performance for almost all metrics (MT, ML, Recall, IDsw, Frag and MOTA).

4.3 Analysis of Accuracy and Speed

We evaluated whether the running speed could be improved while maintaining the tracking accuracy when the frame interval (L) for human detection is increased. The 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 [4]. On the other hand, the SDOF-Tracker performs human detection in every L frame. In the SDOF-Tracker, we evaluated how segmentation, tracking continuation, and tracking termination affect the accuracy and speed. To compare the accuracy fairly, we use the same detection result using Mask-RCNN, both with and without SCT. Therefore, the segmentation time is included when evaluating "without SCT", but the actual speed without segmentation is even faster. We used the MOT20 dataset for the experiment.

Figure 3 (a) 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. "with S" is not so effective when $L = 5$, but is the most effective of S, C, and T when $L \geq 10$. This suggests that the importance of setting good interest points is increasing as L increases.

On the other hand, Fig. 3 (b) shows the change in running speed. As L increases, the running speed increases in both "with/without SCT". The speed improvement rate ac-

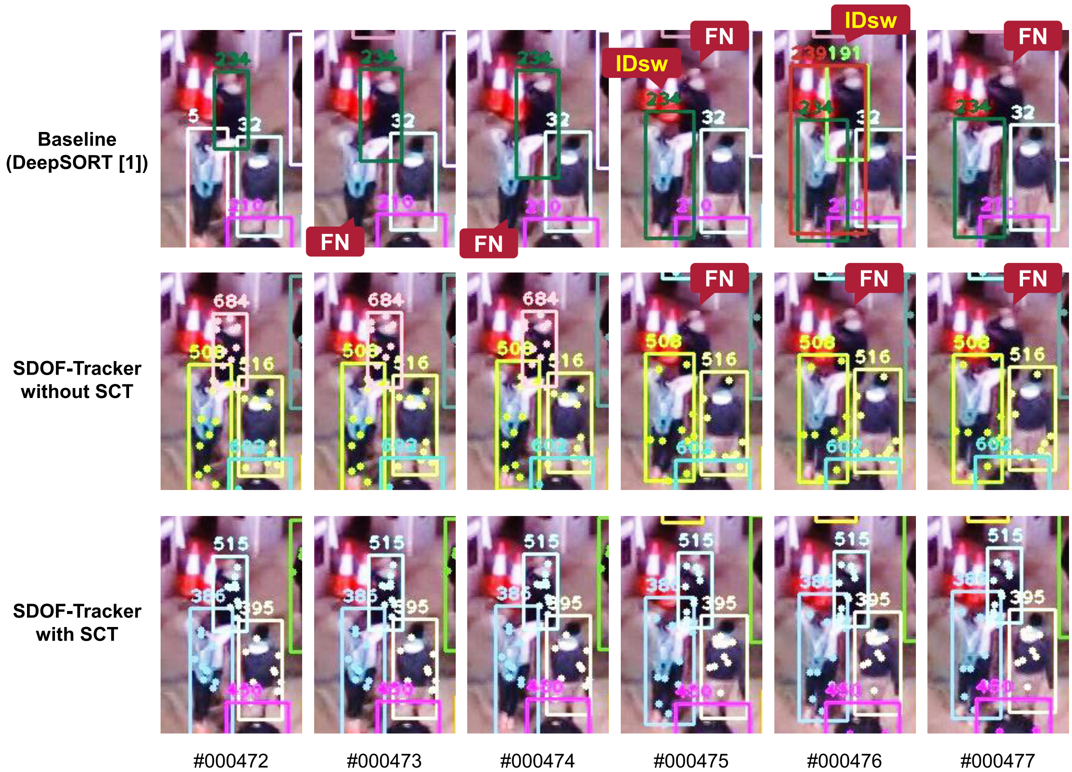


Fig. 4 Cropped example of the tracking result using the baseline and SDOF-Tracker. The brightness and contrast of the images were increased for visibility.

cording to L is higher with SCT than without SCT. This is because the frequency of termination is increased and the number of tracked humans is decreased as L increases. Thus, the SDOF-Tracker with SCT can improve the running speed while maintaining the tracking accuracy.

4.4 Tracking Examples

Figure 4 shows a cropped example of the tracking result using the baseline and SDOF-Tracker in the MOT20 dataset. This is a scene where three people are walking towards the back. In the baseline, ID switches (IDsw) occur due to false negatives (FN). In the SDOF-Tracker, human detection is performed in frame 475 because we set $L = 5$. In the SDOF-Tracker without SCT, false negatives are prevented in frames 473 and 474 due to tracking by optical flow. However, the other false negatives remain. This is because the optical flow cannot start when the false negative occurs in frame 475, which is a chance for human detection. On the other hand, in an 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 with human bodies. Figure 5 shows an example of the tracking result using the SDOF-Tracker with SCT. Even though this is a very crowded scene, most humans are accurately tracked.

4.5 MOTChallenge Result

We compared the SDOF-Tracker to the state-of-the-art methods in the MOTChallenge† on the MOT17 and MOT20 datasets. We compared the performance with methods that have been published in the research literature. We use the public detection results of the MOTChallenge to fairly compare both accuracy and speed. Note that the SDOF-Tracker did not use a GPU for human tracking. Since the public detection results do not include segmentation results, regions were not limited for setting interest points, i.e., it is equivalent to pattern 6 in Table 1. The runtimes of detections are not published on the MOTChallenge, so we cited the runtimes from the related literature [32], [33].

MOT17: The runtime of detection was assumed to be 60.0 ms [32]. Using this runtime, the average runtime of the SDOF-Tracker was estimated as 25.1 ms when the frame interval for human detection was set to $L = 5$. Table 2 shows the MOTChallenge result on the MOT17 dataset. The SDOF-Tracker achieved the best performance in terms of the total runtime. Nevertheless, MOTA was better than GM_PHD.

MOT20: The runtime of detection was assumed to be 131.6 ms [33]. Using this runtime, the average runtime of the SDOF-Tracker was estimated as 68.0 ms when the frame interval for human detection was set to $L = 5$. Table 3

†https://motchallenge.net

Fig. 5 Example of tracking result using the SDOF-Tracker with SCT.

Table 2 MOTChallenge result on MOT17 dataset. The result is cited from the MOTChallenge web page† (Our entry name on the web page is "FlowTracker").

	Rcll [%] ↑	Prcn [%] ↑	IDsw ↓	MOTA ↑	Avg. runtime [ms] ↓ (Excluding detection)	Avg. runtime [ms] ↓ (Total)
SDOF-Tracker	52.3	82.9	5,927	40.4	16.3	25.1
IOU Tracker [16]	50.1	93.4	5,988	45.5	0.7	60.7
SORT [15]	49.0	90.7	4,852	43.1	7.0	67.0
GM_PHD	41.4	90.8	4,607	36.4	26.0	86.0
GMPHD_Rd17	54.3	88.9	3,865	46.8	32.5	92.5
GMPHDOGM17	54.8	92.8	3,125	49.9	32.6	92.6
CoCT_pub	56.5	92.2	1,657	51.4	33.8	93.8
PHD_LMP	51.8	91.3	4,977	45.9	34.0	94.0
BLSTM_MTP_O	57.2	91.6	2,566	51.5	49.8	109.8
MOTDT17	55.6	92.9	2,474	50.9	54.6	114.6
NOTA	55.2	93.9	2,285	51.3	56.2	116.2

Table 3 MOTChallenge result on MOT20 dataset. The result is cited from the MOTChallenge web page† (Our entry name on the web page is "FlowTracker").

	Rcll [%] ↑	Prcn [%] ↑	IDsw ↓	MOTA ↑	Avg. runtime [ms] ↓ (Excluding detection)	Avg. runtime [ms] ↓ (Total)
SDOF-Tracker	58.0	84.6	3,532	46.7	52.1	68.0
SORT [15]	48.8	90.2	4,470	42.7	17.5	149.1
LTSiam [6]	58.5	84.0	4,509	46.5	33.0	164.6
MPNTrack [7]	61.1	94.9	1,210	57.6	153.8	285.4
TBC [21]	62.3	89.5	2,449	54.5	178.6	310.2
SimpleReID [19]	55.3	97.8	2,178	53.6	769.2	900.8
Tracktor [17]	54.3	97.6	1,648	52.6	833.3	964.9
TransCenter [22]	71.4	88.3	4,493	61.0	1,000.0	1,131.6
LPC_MOT [8]	58.8	96.3	1,562	56.3	1,428.6	1,560.2
mfi_tst [31]	66.6	90.5	1,919	59.3	2,000.0	2,131.6
GNNMatch [9]	56.8	96.9	2,038	54.5	10,000.0	10,131.6

shows the MOTChallenge result on the MOT20 dataset. The SDOF-Tracker achieved the best performance in terms of the total runtime. Nevertheless, MOTA was better than SORT and LTSiam.

Discussion: SORT is widely used in real-world applications such as surveillance, and is capable of tracking with practically acceptable accuracy despite its high speed. As shown in MOTA and the average runtime (total) of Table 2 and 3, SDOF-Tracker is comparable to SORT in accuracy, but much faster. Compared to SORT, the speed of SDOF-Tracker is more than twice as fast on both MOT17 and MOT20.

5. Conclusion

In this paper, we proposed the SDOF-Tracker, a fast and accurate human tracking method using skipped detection and optical flow. In the SDOF-Tracker, tracking by optical flow is triggered by human detection and ends based on the variance of the interest points. To maintain accuracy, we introduced robust interest point detection within human regions and a tracking termination metric calculated by the distribution of the interest points. In our experiments, we confirmed that the SDOF-Tracker can improve the running speed while maintaining the tracking accuracy when the frame interval for human detection is increased. Moreover, the SDOF-Tracker achieved the best performance in terms of the total running time (68.0 ms) 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 for human detections.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol.39, no.6, pp.1137–1149, 2017.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), pp.2961–2969, Oct. 2017.

[3] A. Bochkovskiy, C.Y. Wang, and H.Y.M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," Computing Research Repository arXiv Preprint arXiv:2004.10934, 2020.

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), pp.3645–3649, Sept. 2017.

[5] H. Nishimura, K. Tasaka, Y. Kawanishi, and H. Murase, "Multiple human tracking with alternately updating trajectories and multi-frame action features," ITE Transactions on Media Technology and Applications, vol.8, no.4, pp.269–279, 2020.

[6] O. Urbann, O. Bredtmann, M. Otten, J.P. Richter, T. Bauer, and D. Zibriczky, "Online and real-time tracking in a surveillance scenario," Computing Research Repository arXiv Preprint arXiv:2106.01153, 2021.

[7] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.6247–6257, June 2020.

[8] P. Dai, R. Weng, W. Choi, C. Zhang, Z. He, and W. Ding, "Learning a proposal classifier for multiple object tracking," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.2443–2452, June 2021.

[9] I. Papakis, A. Sarkar, and A. Karpatne, "A graph convolutional neural network based approach for traffic monitoring using augmented detections with optical flow," 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp.2980–2986, Sept. 2021.

[10] M. Everingham, J. Sivic, and A. Zisserman, "Taking the bite out of automated naming of characters in tv video," Image. Vision. Comput., vol.27, no.5, pp.545–559, 2009.

[11] M. Schikora, W. Koch, and D. Cremers, "Multi-object tracking via high accuracy optical flow and finite set statistics," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1409–1412, 2011.

[12] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi, "Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions," Computer Vision – ECCV 2012, Lecture Notes in Computer Science, vol.7576, pp.552–565, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[13] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," 2015 IEEE International Conference on Computer Vision (ICCV), pp.3029–3037, Dec. 2015.

[14] S. Bullinger, C. Bodensteiner, and M. Arens, "Instance flow based online multiple object tracking," 2017 IEEE International Conference on Image Processing (ICIP), pp.785–789, Sept. 2017.

[15] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), pp.3464–3468, Sept. 2016.

[16] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp.1–6, 2017.

[17] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp.941–951, 2019.

[18] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," Computer Vision – ECCV 2020, Lecture Notes in Computer Science, vol.12349, pp.474–490, Springer International Publishing, Cham, 2020.

[19] S. Karthik, A. Prabhu, and V. Gandhi, "Simple unsupervised multi-object tracking," Computing Research Repository arXiv Preprint arXiv:2006.02609, 2020.

[20] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," Int. J. Comput. Vision., vol.129, no.11, pp.3069–3087, 2021.

[21] W. Ren, X. Wang, J. Tian, Y. Tang, and A.B. Chan, "Tracking-by-counting: Using network flows on crowd density maps for tracking multiple targets," IEEE Trans. Image Process., vol.30, pp.1439–1452, 2021.

[22] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers with dense queries for multiple-object tracking," Computing Research Repository arXiv Preprint arXiv:2103.15145, 2021.

[23] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proc. 7th International Joint Conference on Artificial Intelligence, pp.121–130, 1981.

[24] H.W. Kuhn, "The Hungarian method for the assignment problem," Nav. Res. Logist. Q., vol.2, no.1-2, pp.83–97, 1955.

[25] M. Liu, X. Ding, and W. Du, "Continuous, real-time object detection on mobile devices without offloading," 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp.976–986, 2020.

[26] A.K. Jain, Fundamentals of digital image processing, Prentice-Hall, Inc., 1989.

[27] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," Computing Research Repository arXiv Preprint arXiv:1603.00831, 2016.

[28] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," Computing Research Repository arXiv Preprint arXiv:2003.09003, 2020.

[29] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," EURASIP Journal on Image and Video Processing, vol.2008, no.246309, pp.1–12, 2008.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft COCO: Common objects in context," Computer Vision – ECCV 2014, Lecture Notes in Computer Science, vol.8693, pp.740–755, Springer International Publishing, Cham, 2014.

[31] J. Yang, H. Ge, J. Yang, Y. Tong, and S. Su, "Online multi-object tracking using multi-function integration and tracking simulation training," Appl. Intell., vol.52, no.2, pp.1268–1288, 2022.

[32] L. Bommes, X. Lin, and J. Zhou, "MVmed: Fast multi-object tracking in the compressed domain," 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp.1419–1424, Nov. 2020.

[33] J. True and N. Khan, "Motion vector extrapolation for video object detection," Computing Research Repository arXiv Preprint arXiv:2104.08918, 2021.

Hitoshi Nishimura received the B.E. and M.E. degrees in engineering from Kobe University, Japan, in 2013 and 2015, respectively. He received the Ph.D. degree in informatics from Nagoya University, Japan, in 2021. He joined KDDI Research, Inc. in 2016 and has been engaged in human tracking and action recognition research. He received the ITE Niwa & Takayanagi best paper award in 2020. He is a member of ITE.

Satoshi Komorita received the B.E. and M.E. degrees from the University of Tokyo in 2004 and 2006, respectively. He joined KDDI Corporation in 2006 and engaged in mobile network research, IEEE Standardization, and smartphone development. He is currently the Senior Manager in charge of Media Recognition Laboratory in KDDI Research, Inc. His current research interests are human pose recognition and position estimation from images.

Yasutomo Kawanishi received the B.Eng. degree in engineering and the M.Inf. and Ph.D. degrees in informatics from Kyoto University, Japan, in 2006, 2008, and 2012, respectively. He became a Postdoctoral Fellow with Kyoto University, in 2012. In 2014, he moved to Nagoya University, Japan, as a Designated Assistant Professor, where he became an Assistant Professor, in 2015, and a Lecturer, in 2020. Since 2021, he has been the Team Leader of the Multimodal Data Recognition Research Team, RIKEN Guardian Robot Project. His main research interests include robot vision for environmental understanding and pattern recognition for human understanding, especially pedestrian detection, tracking, retrieval, and recognition. He received the Best Paper Award from SPC2009 and the Young Researcher Award from the IEEE ITS Society Nagoya Chapter. He is a member of IEEE and IIEEJ.

Hiroshi Murase received the B.Eng., M.Eng., and Ph.D in engineering from Nagoya University, Japan. From 1980 to 2003 he was a research scientist at the Nippon Telegraph and Telephone Corporation (NTT). He has been a professor of Nagoya University since 2003. He was awarded the IEEE CVPR Best Paper Award in 1994, the IEICE Distinguished Achievement and Contributions Award in 2018. He received Shijyu-hosho (the Medal with Purple Ribbon) in 2012. His research interests include computer vision, pattern recognition, and multimedia information processing. He is a life fellow of IEEE, and a fellow of IPSJ.