Pedestrian Tracking with Occlusion State Estimation

Akihiro Enomura[1], Toru Abe[2][a] and Takuo Suganuma[2][b]
[1]*Graduate School of Information Sciences, Tohoku University, 2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan*
[2]*Cyberscience Center, Tohoku University, 2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan*
enomura@ci.cc.tohoku.ac.jp, {beto, suganuma}@tohoku.ac.jp

Keywords: Pedestrian Tracking, Tracking-by-Detection, Obstacle Area, Pedestrian Movement, Occlusion State.

Abstract: Visual tracking of multiple pedestrians in video sequences is an important procedure for many computer vision applications. The tracking-by-detection approach is widely used for visual pedestrian tracking. This approach extracts pedestrian regions from each video frame and associates the extracted regions across frames as the same pedestrian according to the similarities of region features (e.g., position, appearance, and movement). When a pedestrian is temporarily occluded by a still obstacle in the scene, he/she disappears at one side of the obstacle in a certain frame and then reappears at the other side of it a few frames later. The occlusion state of the pedestrian, that is the space-time interval where the pedestrian is missing, varies with obstacle areas and pedestrian movements. Such an unknown occlusion state complicates the region association process for the same pedestrian and makes the pedestrian tracking difficult. To solve this difficulty and improve pedestrian tracking robustness, we propose a novel method for tracking pedestrians while estimating their occlusion states. Our method acquires obstacle areas by the pedestrian regions extracted from each frame, estimates the occlusion states from the acquired obstacle areas and pedestrian movements, and reflects the estimated occlusion states in the region association process.

1 INTRODUCTION

Visual tracking of multiple pedestrians in video sequences is an important procedure for many computer vision applications. The tracking-by-detection approach is widely used for visual pedestrian tracking (Jiang and Huynh, 2018; Mekonnen and Lerasle, 2019). This approach extracts pedestrian regions from each video frame and associates the extracted regions across frames as the same pedestrian according to the similarities of extracted region features (e.g., position, appearance, and movement). When a pedestrian is temporarily and fully occluded by a still obstacle in the scene, as shown in Figure 1, he/she disappears at one side of the obstacle in a certain frame (position p in frame t) and then reappears at the other side of it a few frames later (position $p+\Delta p$ in frame $t+\Delta t$). The occlusion state of the pedestrian, that is the space-time interval where the pedestrian is missing, varies with obstacle areas and pedestrian movements, which are not able to be determined in advance. Such an unknown occlusion state complicates the region association process for the same pedestrian and makes the pedestrian tracking difficult.

To solve this kind of difficulty and improve the robustness of pedestrian tracking in video sequences, we propose a novel method for tracking pedestrians while estimating their occlusion states. The proposed method, which is based on the tracking-by-detection approach, firstly acquires still obstacle areas in the

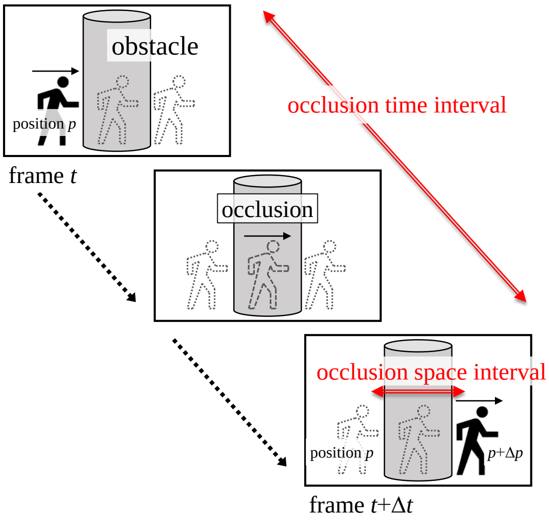


Figure 1: Occlusion space-time interval.

[a] https://orcid.org/0000-0002-3786-0122
[b] https://orcid.org/0000-0002-5798-5125

scene by the foot positions of pedestrian regions extracted from each frame, secondly estimates the occlusion states from the acquired obstacle areas and pedestrian movements, and thirdly reflects the estimated occlusion states in the region association process. In our method, the positional relations in depth direction (in front or behind) between pedestrians and obstacles are determined by focusing on the foot positions of extracted pedestrian regions, and thus the occlusion states of the pedestrians are estimated accurately.

The remainder of this paper is organized as follows: Section 2 presents the schemes to deal with occlusion problems in the existing methods for tracking multiple objects including pedestrians in video sequences, Section 3 explains the details of our proposed method for tracking pedestrians while estimating their occlusion states, Section 4 presents the results of pedestrian tracking experiments, and Section 5 concludes this paper.

2 RELATED WORK

For tracking multiple objects including pedestrians in video sequences, many methods based on the tracking-by-detection approach have been proposed. Most of them take account of occlusion problems and have some procedures to deal with these problems.

One of such procedures is to set a spatial-temporal search range for each target object region according to its position in the current frame and associate the target region with extracted regions in adjacent frames within the search range as the same object. Thus, when a target object disappears at a certain position in a certain frame due to occlusion, the methods using this procedure wait the region association for the target object until it reappears in the consecutive frames within the search range. The search range needs to be adjusted appropriately according to the occlusion state of the target object, that is the space-time interval where the target object is missing, however most existing methods use a fixed range determined in advance. The following summarizes how the existing methods determine the temporal search range for an occlusion time interval and the spatial search range for an occlusion space interval.

- **Temporal Search Range**
  Several methods determine the temporal search range manually (Huang et al., 2008; Mitzel and Leibe, 2011; Possegger et al., 2014; Ju et al., 2017; Zhu et al., 2018). Through preliminary experiments, they choose the number of frames which obtains good tracking performance as the appropriate temporal search range. Some methods use very small temporal search ranges. For example, the method in (Bewley et al., 2016) allows the region association only between consecutive two frames. While this method prevents incorrect region associations, it is difficult to proceed the region association for the same object after occlusion.
- **Spatial Search Range**
  In (Salvi et al., 2013; Possegger et al., 2014; Ju et al., 2017; Zhu et al., 2018), the spatial search range is determined manually. As with the temporal search range described above, the spatial search range which obtains good tracking performance is determined through preliminary experiments. In many cases, the spatial search range corresponds to the width of an obstacle in the scene. Compared to those, the method in (Ju et al., 2017) sets the spatial search range automatically according to the width of a target object region within a manually-set upper range limit. In (Huang et al., 2008), the spatial search range is extended to the entire field of the frame, when the target object is lost. While this method can deal with occlusion caused by unknown size obstacles, it is likely to associate the target region with incorrect object regions. The method in (Bochinski et al., 2018) associates a target region with spatially overlapped regions across frames. This is equivalent to confine the search range to the immediate vicinity of the target region in the current frame without any regard for a long occlusion space interval.

These methods cannot cope with unknown occlusion states effectively. If the search range is set too large, target regions are likely to be associated with incorrect object regions. If, on the other hand, the search range is set too small, it is difficult to proceed the region association for the same object after occlusion.

In order to solve such trade-off problem, the size of search range should be adjusted according to occlusion states. This requires the front-behind relations of obstacles and target objects in the scene to predict the occlusion states. Some methods directly identify the position of the obstacle by acquiring the depth of a scene. A depth sensor based method is used in (Meshgi et al., 2016) and a multi-view stereo based method is used in (Osawa et al., 2007) for acquiring the scene depth. The front-behind relations between obstacles and target objects are estimated from the acquired scene depth, and then reflected in adjusting the size of search range. Although these methods can adjust the search range appropriately, their applicable environments are limited. Only a few methods (Hof-

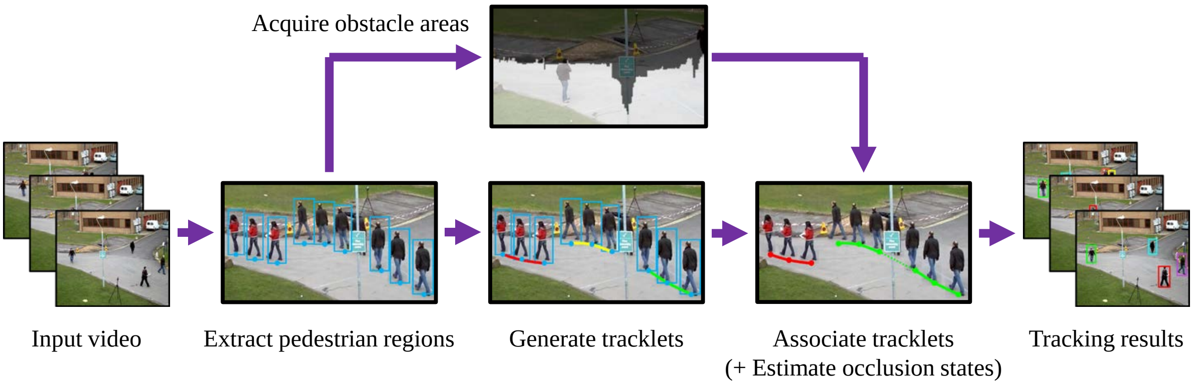


Figure 2: The overview of the proposed method.

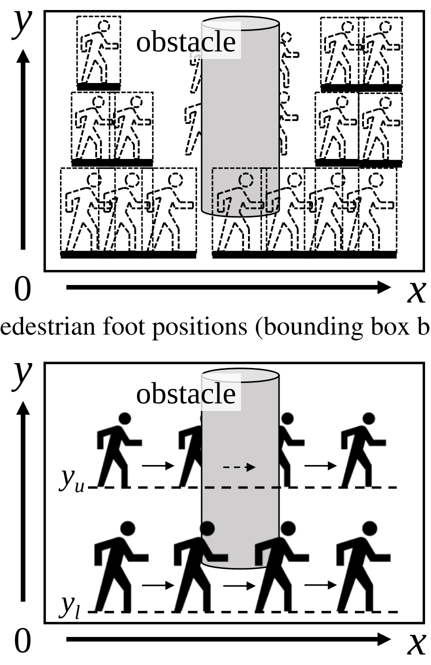
mann et al., 2013) set obstacle areas manually in the scene beforehand, however they cannot effectively estimate the front-behind relations between obstacles and target objects and cannot flexibly adapt to use in various different scenes.

3 PROPOSED METHOD

Figure 2 shows the overview of the proposed method. In our method, firstly, pedestrian regions are extracted from each video frame. From these extracted pedestrian regions, obstacle areas in the scene are acquired. At the same time, short tracks “track lets” of individual pedestrians, each of which is an intermediate result of associating regions across frames as the same pedestrian, are generated from the extracted regions. The occlusion states are estimated from the relation of the acquired obstacle areas and the generated tracklets. By the equivalent process as dynamically adjusting the size of search range according to the estimated occlusion state, our method reflects the estimated occlusion states in the association process for tracklets and achieves robustly tracking of multiple pedestrians in the scenes with obstacles.

3.1 Obstacle Area Acquisition

To acquire obstacle areas in the scene, firstly, by using all pedestrian regions extracted as bounding boxes in a video sequence, the extraction frequency $F(x,y)$ of bounding box bases is counted at each pixel (x,y) for all video frames. The value of $F(x,y)$ is regarded as the total number of pedestrian foot positions which overlap at (x,y) as shown in Figure 3 (a). It would appear that pedestrian regions whose bounding box bases are at (x,y) aren’t occluded by obstacles in the video frame (image) where $F(x,y)$ is large. However, this doesn’t consider the distances from a camera to pedestrians and obstacles.

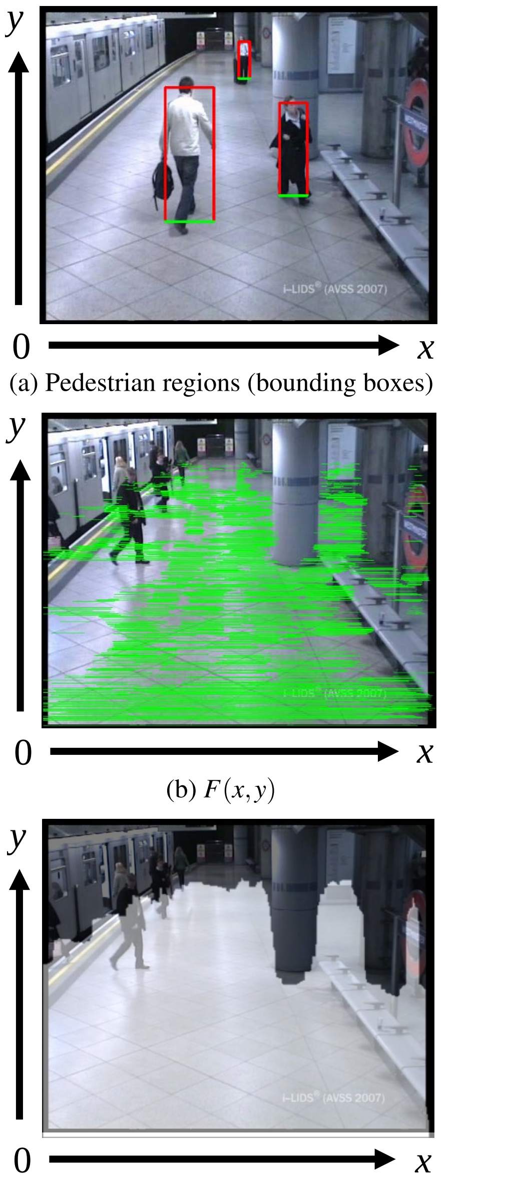


(a) Pedestrian foot positions (bounding box bases).

(b) Pedestrians and obstacles in the scene.

Figure 3: Front-behind relations in the image.

If the camera is set horizontally, and besides, pedestrians and obstacles stand perpendicularly on the flat ground, the distance from the camera to each pedestrian or each obstacle in the scene is reflected on the vertical coordinate of it in the image. Roughly speaking, as shown in Figure 3 (b), a pedestrian or an obstacle at a longer distance from the camera is appeared on the upper part in the image, whereas it at a shorter distance from the camera is appeared on the lower part in the image. Accordingly, for the same horizontal coordinate x in the image, if $F(x,y_u)$ is large at its upper part (x,y_u), then pedestrians are unlikely to be occluded by obstacles also at its lower part (x,y_l) where $y_l < y_u$. From this, the proposed method computes $F_b(x,y)$ and binarizes it to obtain



(a) Pedestrian regions (bounding boxes)

(b) $F(x,y)$

(c) $F_b^*(x,y)$

Figure 4: Examples of acquired obstacle areas.

$F_b^*(x,y) = \{0,1\}$ by

$$F_b(x,y) = \sum_{y \le h \le y_{\max}} F(x,h), \tag{1}$$

$$F_b^*(x,y) = \begin{cases} 1, & F_b(x,y) > t_b, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $y_{\max}$ is the vertical coordinate at the top of the image and t_b is a given threshold.

For a pedestrian region whose bounding box base is at (x,y), obtained $F_b^*(x,y)$ indicate whether or not there are obstacles occluding it, i.e, if $F_b^*(x,y) = 0$ then there is an obstacle area and the pedestrian region is occluded by the obstacles otherwise it isn't occluded. Thus, the front-behind relations of obstacles and pedestrians in the scene can be reflected in tracking process by referring to $F_b^*(x,y)$. Example of acquiring obstacle ares is shown in Figure 4. Figure 4 (a) shows an example of pedestrian regions extracted as bounding boxes (red lines) and their bases (green lines), (b) shows overlapped bounding box bases $F(x,y)$, and (c) shows acquired obstacle areas $F_b^*(x,y)$, where white areas indicate $F_b^*(x,y) = 1$ (none obstacle areas).

3.2 Pedestrian Tracking

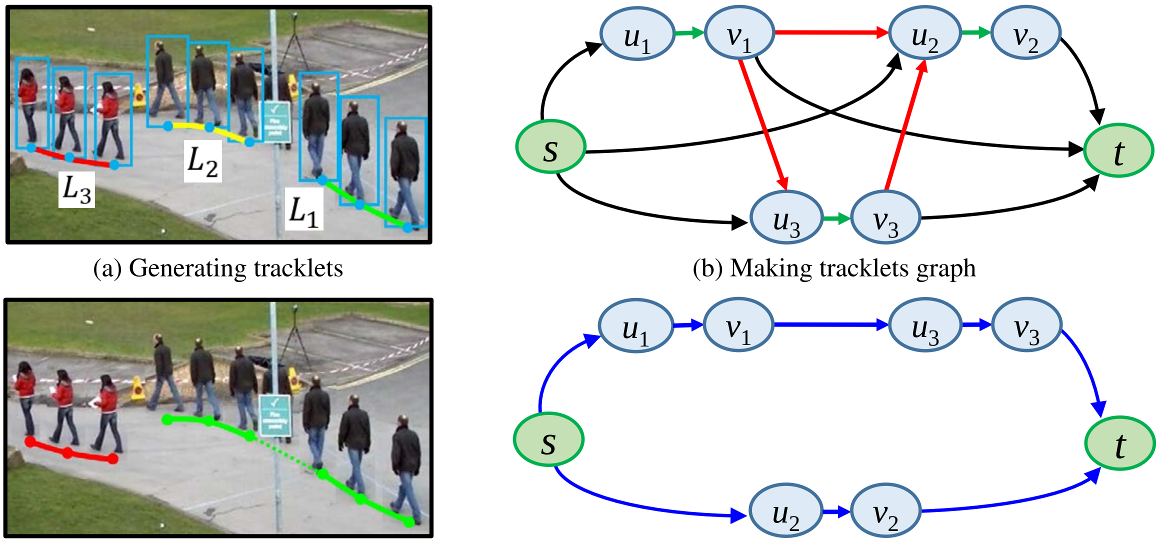
Tracking pedestrians is carried out by pursuing regions corresponded to the same pedestrian. Pursuing process consists of two stages. First, matching regions between adjacent frames to generate trajectory fragments (called "tracklet"). Multiple tracklets are generated for the same pedestrian before and after occlusion. Second, we represents the relation of each tracklets extracted in first step frames as a graph, and apply the approach, which utilizes the minimum cost of a flow network to handle multiple object tracking, for pursuing tracklets of the same object.

3.2.1 Generating Tracklets

Following (Shu et al., 2012; Ju et al., 2017), generate tracklets by one-to-one correspondence of pedestrian regions detected between adjacent frames. Assuming that the gap of same object's spatial position between adjacent frames is tiny, regions are matched by the minimum binary matching that sets matching cost as the gap of pedestrian's position (Euclidean distance). We introduce the threshold t_g to avoid switching targets, never regions whose cost exceeds the threshold. Each tracklet is represented as L_i where i is the tracklet number as shown in Figure 5 (a).

3.2.2 Pursuing Tracklets

We pursue tracklets based on (Zhang et al., 2008) algorithm. The concept of flow network model of pursuing tracklets is shown in Figure 5 (b). In this graph, the start and goal of tracking are denoted as nodes s and t. Each tracklets L_i is represented by two nodes u_i, v_i and a green edge. u_i is head, v_i is tail of L_i. Relationship of tracklets is represented by red edges, each of which has a corresponding cost and unit capacity. Most existing methods set threshold value as a fixed spatial search range SR_S, and temporal search range SR_T to cut the edges where the distance between two regions is greater than SR_S or the frame interval is greater than SR_T. Proposed method calculates brute



(a) Generating tracklets

(b) Making tracklets graph

(c) Pursuing tracklets

(d) Searching minimum cost path

Figure 5: The concept of flow network based model of pursuing tracklets.

force edges for all tracklets, unlike existing methods. For a flow from s to t, the path that has the minimum sum of corresponding costs is determined as a tracking pedestrian trajectory (Figure 5 (d)).

Generally, the corresponding cost $C_{i,j}$ is computed from the similarity $S_{i,j}$ between tracklets L_i and L_j by:

$$C_{i,j} = -\log(S_{i,j}) \tag{3}$$

where $S_{i,j}$ is determined from the similarities in such features as position, appearance, and movement between L_i and L_j. The larger the similarity, the smaller the corresponding cost, and pursuing tracklets is also easier.

In the proposed method, to deal with the difficulty caused by the space-time occlusion interval, new cost terms reflecting a pedestrian occlusion state is added to $C_{i,j}$. The new corresponding cost $C'_{i,j}$ between L_i and L_j is determined by

$$C'_{i,j} = -\log(S_{i,j}) + \alpha P_{i,j} + \beta T_{i,j} \tag{4}$$

where α,β are positive constants, and $P_{i,j}$,$T_{i,j}$ are the cost terms reflecting pedestrian occlusion state.

The cost terms $P_{i,j}$, $T_{i,j}$ are determined as the error between predicted appearance of L_i after occlusion and actual measurement of L_j. The position where L_i appears again after occlusion is denoted p_i (Plane coordinates), and the time is denoted t_i (frame number). These are predicted values considering the occlusion state in Eq. (2). The actual position of L_j is expressed as p'_j, and the time as t'_j. Cost terms $P_{i,j}$ and $T_{i,j}$ are calculated as space and time errors as shown in Eqs. (5) and (6).

$$P_{i,j} = |p_i - p'_j| \tag{5}$$

$$T_{i,j} = |t_i - t'_j| \tag{6}$$

According to Eq. (4), small $P_{i,j}$,$T_{i,j}$ decreases the corresponding cost $C'_{i,j}$, therefore pursuing L_i between L_j will be easier. This is similar to provide a *SR* around the space-time position where L_i is predicted to appear. By this way, the proposed method introduces the pedestrian occlusion state into the corresponding cost, and accomplishes the equivalent process as adjusting *SR* according to the pedestrian occlusion state.

3.2.3 Appearance Prediction of *Li* after Occlusion

To predict position p_i and time t_i where L_i appears again after occlusion, we assume that pedestrians move at a constant speed and on a straight line in the real world while he is occluded. Pedestrians appear at the boundary between $F^*(x,y) = 0$ (with occlusion) and $F^*(x,y) = 1$ (with no occlusion). The pedestrian's trajectory is determined by applying the exponential moving average method to the position of the pedestrian regions constituting the tracklet L_i.

4 EXPERIMENTS

To demonstrate the efficacy of the proposed method, we carry out pedestrian tracking experiments. We present our results using two type datasets and compare our method with conventional methods.

4.1 Experiment Overview

For evaluation using actual video AVSS2007 (720 × 576pixels, 25fps) (AVSS2007, 2007), 2400 frames

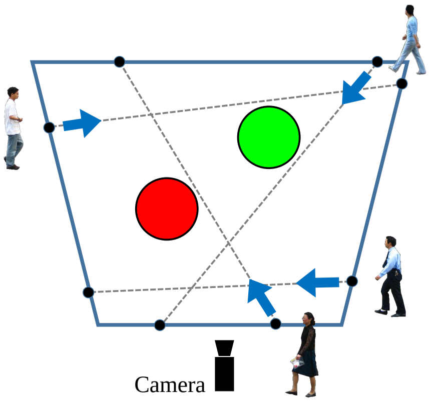


Figure 6: Example of pedestrian trajectory in the simulator (Pedestrians are initially placed on blue line, red and green circle represent obstacles (cylinder), gray dashed line is trajectory of each pedestrian).

Table 1: Search range (SR) setting in conventional method (”width” is width of frame and ”fps” is frame rate of video).

	Spatial SR	Temporal SR
Method (a)	width*0.03	fps*0.4
Method (b)	width*0.20 (set properly)	fps*2.0 (set properly)
Method (c)	width*0.30	fps*4.0

(frame t = 2502650) are extracted, and manually associating detected regions is used as the ground truth. Additionally, to obtain precise ground truth (pedestrian loci) and make scenes simple except occlusion, simulation videos (CG animations) are used as input videos. Twenty videos are created by using POV-Ray (POV-Ray, 2019), each of which is 800 × 600pixels, 5fps, and 50frames. There are two obstacles (cylinder) in the scene, and four pedestrians walk. As shown in Figure 6, pedestrian's start line (blue trapezoid) is located around the obstacle, and four start points are placed on each edge. Pedestrian moves in straight line toward the opposite side of the trapezoid. Start points and end points are randomly determined. In the videos, various types of occlusion occur to pedestrians by obstacles or other pedestrians.

The skeletons of pedestrians are extracted from each frame by OpenPose (Cao et al., 2017), and a bounding rectangle of every extracted skeleton is used as a pedestrian region.

In Eq. (4), the similarity $S_{i,j}$ between pedestrian regions L_i and L_j is set as

$$S_{i,j} = S_p \times S_t \times S_a \times S_s \tag{7}$$

where S_p, S_t, $S_a$and S_s are the similarities between L_i and L_j in position, time (frame), appearance, region size, respectively. The position similarity is determined as $S_p = \exp(-$Euclidean distance between L_i and $L_j)$, the time similarity S_s is determined as $S_p = \exp(-$frame interval between L_i and $L_j)$, the appearance similarity S_a is normalized correlation between color histograms in L_i and L_j. Consequently, $S_{i,j}$ ranges from 0 to 1.

Table 2: Evaluation metrics.

Metric	Definition
FM	The total number of times a trajectory is fragmented (interrupted during tracking).
SW	The total number of switches its matched ground truth identity.
MS	The total number of missed targets.
FP	The total number of false positives.
MOTP	Multiple Object Tracking Precision. The misalignment between the annotated and the predicted bounding boxes.
MOTA	Multiple Object Tracking Accuracy. This measure combines three error sources: FP, MS and SW.
RS	Ratio of tracks which are correctly recovered from Short occlusion. We define the occlusion is caused by other target.
RL	Ratio of tracks which are correctly recovered from Long occlusion. We define the occlusion is caused by static obstacle.

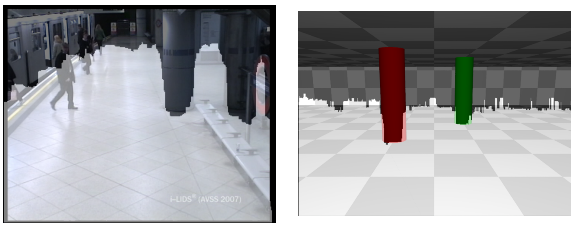

(a) AVSS2007 (b) POV-Ray animation

Figure 7: Acquired obstacle areas.

We evaluate the tracking performance when a new cost term $P_{i,j}$, $T_{i,j}$ in Eq. (4) are introduced into the matching cost (proposed method $\alpha, \beta > 0$) and is not introduced (conventional method $\alpha, \beta = 0$). As shown in Table 1, the space-time search range (SR) of the conventional method is set to three variations, which are named Method (a), (b), and (c), respectively. Spatial SR is based on the width of the video, and temporal SR is based on the frame rate of the video. As shown in Table 1, (a) sets the search range extremely small, and (c) sets it extremely large compared to the occlusion interval. (b) manually sets the search range which the method shows better performance (that means fewer identity switches) in each dataset.

A summary and short description of the used mea-

Table 3: Quantitative results (↑:the higher is the better, ↓:the lower is the better).

Dataset	Method	FM↓	SW↓	MS↓	FP↓	MOTP↑	MOTA↑	RS↑	RL↑
AVSS2007	Generating tracklets	121	55	61	912	0.8606	0.8297	0.00 (0/89)	0.00 (0/43)
	Method (a)	57	62	63	912	0.8600	0.8272	0.28 (25/89)	0.00 (0/43)
	Method (b)	12	83	23	928	0.8632	0.8231	0.52 (47/89)	0.51 (22/43)
	Method (c)	8	92	23	928	0.8625	0.8227	0.44 (40/89)	0.37 (16/43)
	Proposed method	17	64	23	928	0.8639	0.8268	0.52 (47/89)	0.74 (32/43)
POV-Ray animation	Generating tracklets	48	25	9	77	0.9516	0.9268	0.00 (0/28)	0.00 (0/24)
	Method (a)	47	26	9	77	0.9514	0.9275	0.82 (6/28)	0.00 (0/24)
	Method (b)	17	30	7	77	0.9525	0.9262	0.64 (18/28)	0.58 (14/24)
	Method (c)	5	34	7	79	0.9507	0.9211	0.64 (18/28)	0.42 (10/24)
	Proposed method	2	26	7	76	0.9535	0.9278	0.64 (18/28)	0.83 (20/24)

sures is given in Table 2. We use the widespread measures in (Bernardin and Stiefelhagen, 2008) called Switch (SW), Miss (MS), False Positive (FP), Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). Additionally, we apply further metrics that are presented in (Li et al., 2009), that is Fragment (FM). To evaluate focusing on robustness to occlusion, we use Recover from Short-term occlusion (RS) and Recover from Long-term occlusion (RL) are introduced in (Song et al., 2010). These represent the Ratio of tracks which are correctly recovered from short/long occlusion. In the experiment, RS is used as an evaluation metric for occlusion between targets (pedestrian) , and RL is used for occlusion due to static obstacles.

4.2 Experimental Results

First, show results of obstacle areas detection. To detect obstacle areas, the threshold in Eq. (2) is set as t_b = 0. Figure 7 shows detected obstacle areas $F_b^*(x,y)$, where white areas indicate $F_b^*(x,y) = 1$ (none obstacle areas). Figure 7 (a) is the result obtained from 6879 pedestrian regions. Figure 7 (b) is the result obtained from 3212 pedestrian regions of all 20 videos. These detections include false positives. The detection result of the area without the pedestrian's trajectory (i.e. background excluding floor and obstacles) can be ignored because it does not affect the subsequent tracking process.

Examples of tracking for each method are shown in Figures 8 and 9. Each track is assigned a unique id and a color rectangle. We refer to the performance of each method by comparing the transition of tracking in the same frame. Figure 9 also shows the top views of pedestrian trajectories and tracking results in the simulator. Each symbol represents the pedestrian trajectory, and the colored lines represent the tracking results (the colors match those of the rectangle in Figure 9). Focus on tracking example of Method (a) and (c), these are examples of failed tracking. In Method (a), the space-time SR is set small compared to the occlusion interval, so the same pedestrian does not appear in the SR after occlusion. It causes interruption of tracking, and increasing FM. In Method (b), the space-time SR is set large, so many pedestrians appear simultaneously in the search range after occlusion. It causes an incorrect association, and increasing SW. On the other hand, proposed method succeeds in matching the same person after occlusion and achieve lower FM/SW.

Table 3 presents quantitative results of each approach on two datasets. "Generating tracklets" indicates the tracking evaluation at the tracklet stage. All the following methods pursue these common tracklets. When SR of the conventional method expands from (a) to (c), FM decreases and SW increases. In such a trade-off relationship, Method (b) has an appropriate SR and keep both metrics relatively low. Among the conventional methods, method (b) shows better results in MOTP. The proposed method shows the best performance of all methods. Our stable tracking result is due to the adjustment of the matching cost $C'_{i,j}$ based on the obstacle areas detection.

FP and MS maintain almost constant values between the proposed method and the conventional method. FP greatly depends on the performance of the human detector. Also, since MS occurs in the phase of generating tracklets, it is not affected by subsequent processing (pursuing tracklets). Due to the constancy of FP and MS, MOTA is greatly affected by SW. Among the all methods, method (a) shows the best results for MOTA, but leaves a very large FM problem.

Proposed method shows relatively high RL. The advantage of being able to adjust the SR according to the occlusion state proves effective tracking for static obstacles. Incidentally, Method (b) got high performance in RS. Since occlusion between targets occurs in a short or medium term, a method with appropriate SR is advantageous.

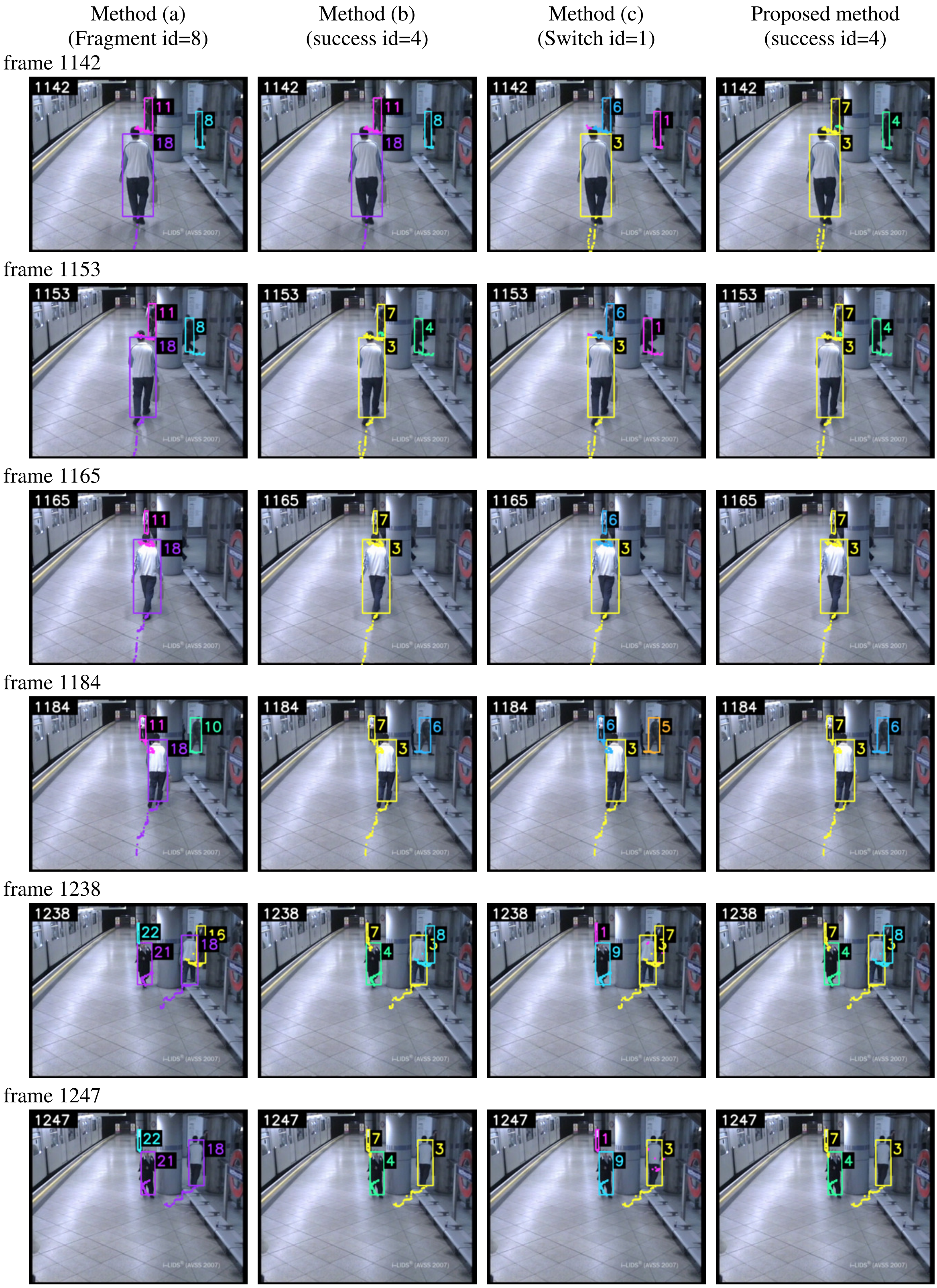


Figure 8: Examples of tracking results (AVSS2007).

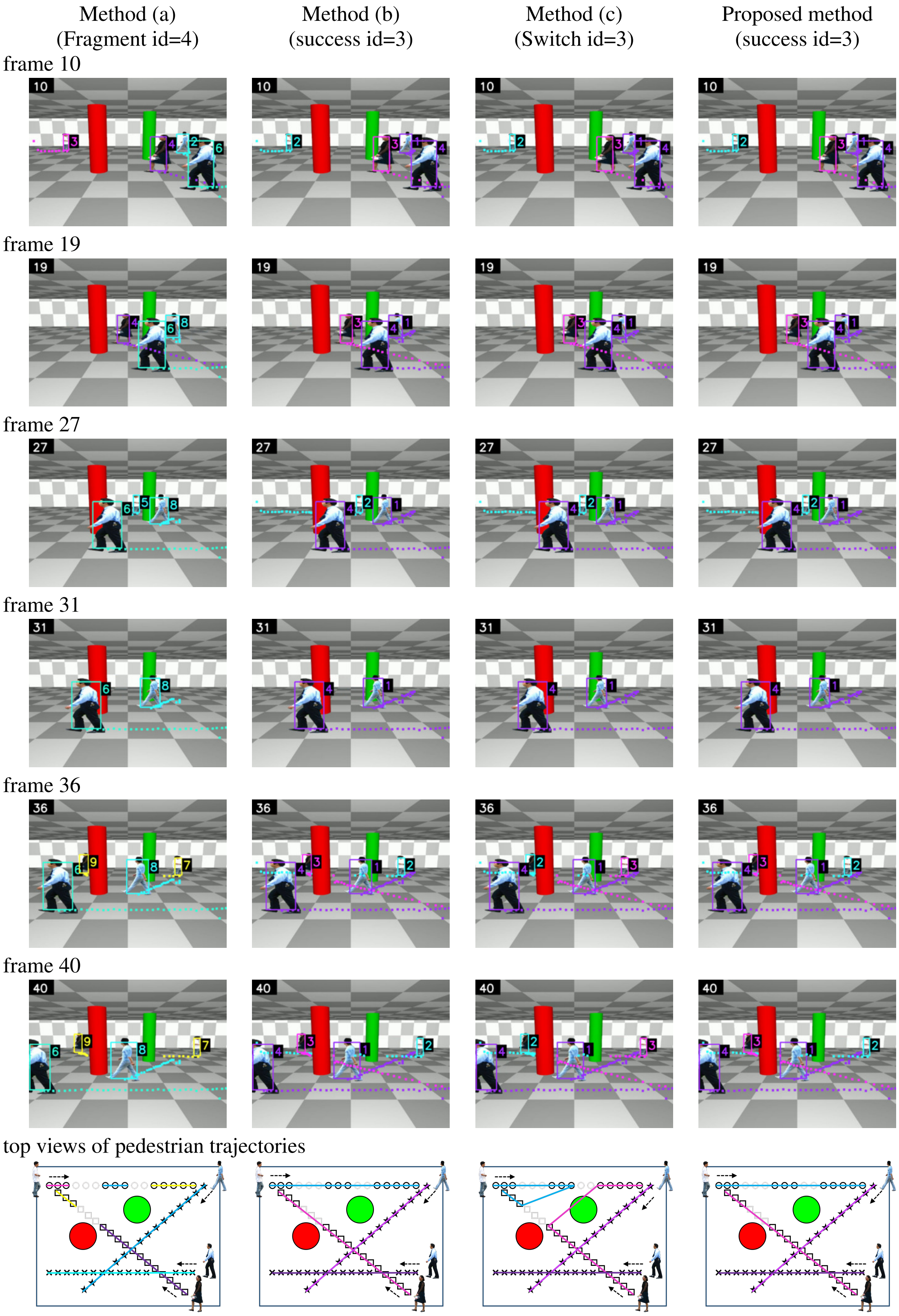


Figure 9: Examples of tracking results (POV-Ray animation).

5 CONCLUSIONS

In this paper, we proposed a method for tracking multiple pedestrians in video sequences. The proposed method extracts pedestrian regions in each video frame, detects obstacle areas in the scene from the extracted pedestrian regions, and tracks pedestrians while estimating their occlusion states from the detected obstacle areas. The efficacy of our proposal was demonstrated through experiments on simulation video sequences. The experimental results showed that the proposed method, which estimates the occlusion states of pedestrians and reflects them on region association process, improves the robustness in visual tracking multiple pedestrians under situations where pedestrians are temporary occluded by still objects.

In future work, we plan to investigate a method for updating detected obstacle areas by new input video frames, and extend the proposed method in order to deal with situations where pedestrians are temporary occluded by occasionally moving obstacles, e.g, temporary parked cars and stacked objects.

REFERENCES

AVSS2007 (2007). i-Lids dataset for AVSS2007. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html.

Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process.*, 2008(1):Article ID 246309.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In *IEEE Int. Conf. Image Process.*, pages 3464–3468.

Bochinski, E., Senst, T., and Sikora, T. (2018). Extending IOU based multi-object tracking by visual information. In *IEEE Int. Conf. Adv. Video Signal Based Surv.*, pages 1–6.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conf. Comput. Vision Pattern Recognit.*, pages 1302–1310.

Hofmann, M., Haag, M., and Rigoll, G. (2013). Unified hierarchical multi-object tracking using global data association. In *IEEE Int. Workshop Perform. Eval. Tracking Surv.*, pages 22–28.

Huang, C., Wu, B., and Nevatia, R. (2008). Robust object tracking by hierarchical association of detection responses. In *Eur. Conf. Comput. Vision*, pages 788–801.

Jiang, Z. and Huynh, D. Q. (2018). Multiple pedestrian tracking from monocular videos in an interacting multiple model framework. *IEEE Trans. Image Process.*, 27(3):1361–1375.

Ju, J., Kim, D., Ku, B., Han, D. K., and Ko, H. (2017). Online multi-person tracking with two-stage data association and online appearance model learning. *IET Comput. Vision*, 11(1):87–95.

Li, Y., Huang, C., and Nevatia, R. (2009). Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *IEEE Conf. Comput. Vision Pattern Recognit.*, pages 2953–2960.

Mekonnen, A. A. and Lerasle, F. (2019). Comparative evaluations of selected tracking-by-detection approaches. *IEEE Trans. Circuits Syst. Video Technol.*, 29(4):996–1010.

Meshgi, K., ichi Maeda, S., Oba, S., Skibbe, H., zhe Li, Y., and Ishii, S. (2016). An occlusion-aware particle filter tracker to handle complex and persistent occlusions. *Comput. Vision Image Unders.*, 150:81–94.

Mitzel, D. and Leibe, B. (2011). Real-time multi-person tracking with detector assisted structure propagation. In *IEEE Int. Conf. Comput. Vision Workshops*, pages 974–981.

Osawa, T., Wu, X., Sudo, K., Wakabayashi, K., Arai, H., and Yasuno, T. (2007). MCMC based multi-body tracking using full 3D model of both target and environment. In *IEEE Int. Conf. Adv. Video Signal Based Surv.*, pages 224–229.

Possegger, H., Mauthner, T., Roth, P. M., and Bischof, H. (2014). Occlusion geodesics for online multi-object tracking. In *IEEE Conf. Comput. Vision Pattern Recognit.*, pages 1306–1313.

POV-Ray (2019). POV-Ray The persistence of vision raytracer, persistence of vision raytracer pty. ltd. http://www.povray.org/.

Salvi, D., Waggoner, J., Temlyakov, A., and Wang, S. (2013). A graph-based algorithm for multi-target tracking with occlusion. In *IEEE Workshop Appl. Comput. Vision*, pages 489–496.

Shu, G., Dehghan, A., Oreifej, O., Hand, E., and Shah, M. (2012). Part-based multiple-person tracking with partial occlusion handling. In *IEEE Conf. Comput. Vision Pattern Recognit.*, pages 1815–1821.

Song, B., Jeng, T.-Y., Staudt, E., and Roy-Chowdhury, A. K. (2010). A stochastic graph evolution framework for robust multi-target tracking. In *Eur. Conf. Comput. Vision*, pages 605–619.

Zhang, L., Li, Y., and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *IEEE Conf. Comput. Vision Pattern Recognit.*

Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., and Yang, M.-H. (2018). Online multi-object tracking with dual matching attention networks. In *Eur. Conf. Comput. Vision*, pages 379–396.