Improving the Efficiency of Multiple Object Tracking via Adaptive Tracker Selection Suitable to Occlusion States

Bo Chen[1,a)] Muhammad Alfian Amrizal[2,b)] Satoru Izumi[1,c)] Toru Abe[1,3,d)]
Takuo Suganuma[1,3,e)]

Abstract: The objective of multiple object tracking (MOT) is to locate the position of multiple objects in a video, maintain their identities, and obtain their trajectories. A crucial challenge of MOT is on how to efficiently handle the occlusion. Based on the ability of handling occlusion, the existing trackers can be classified into occlusion-weak and occlusion-robust trackers. The former is relatively more efficient in tracking objects without occlusion; however, its tracking effectiveness considerably decreases for occluded objects. On the other hand, the latter can effectively track occluded objects but run less efficiently on non-occluded situations. In this research, we try to improve the efficiency of MOT by categorizing occlusion states into two states: non-occluded and occluded. The most suitable tracker is then deployed depending on the current occlusion state and adaptively changed when the state changes. We tested our proposal on different scenarios to evaluate its effectiveness.

Keywords: Multiple Object Tracking, Occlusion

1. Introduction

With the rapid development of vision technology, videos have become a popular media in many applications, and therefore the requirement of video processing has increased significantly. In many applications, such as visual surveillance [1], human-computer interaction [2], and virtual reality [3], it is important to understand the movement of objects in a video. For this purpose, object tracking technology has been extensively developed in the past decades. Multiple Object Tracking (MOT) is a kind of video processing technology whose objective is to locate the position of multiple objects in a video, maintaining their identities, and obtaining their trajectories [4].

The two basic strategies of existing MOT methods are Detection-Free-Tracking and Tracking-By-Detection. In Detection-Free-Tracking method, human must indicate the position of objects that will be tracked, which costs laborious work and hard to implement in reality. In Tracking-By-Detection method, a detector is used to automatically discover the objects, which makes it more practical and attracts lots of attention.

The progress of "Tracking-By-Detection" based MOT methods involves two basic components: a detector and a tracker. The task of the detector is to traverse each frame and give the position and categories (classes) of objects, meanwhile, the task of the tracker is to compare objects of different frames and find correspondence among them, by using the following information such as appearance, motion, exclusion, occlusion, etc.

The challenges of MOT include initialization and termination of tracks, similar appearance, frequent occlusion, and interactions among objects [4]. The occlusion issue might be the most critical challenge in MOT. During occlusion, parts or the whole object is covered by "front" objects, resulting in burdensome or impossibility on finding correspondence between objects from different frames. In this situation, the trajectory of object might be interrupted, and the identity of object may change (ID-switch), leading to failure of tracking.

In the past decades, many MOT methods have been proposed [5], [6], [7], [8]. Based on the ability of handling occlusion, these works could be classified into occlusion-weak and occlusion-robust types. In occlusion-weak methods, there is no occlusion handling component and the architecture is often very simple, which makes they are efficient for tracking of objects without occlusion, but unreliable for occluded ones. For occlusion-robust methods, complex architecture of occlusion handling is designed, which makes they are robust for occlusion, but inevitably requires more computational resources, even for objects without occlusion.

Recently, mobile platforms are becoming popular, many computer vision applications have been applied to them [9]. Such platforms have strictly limited computational resources, and thus computer vision applications must be developed such that they can run very efficiently on them. For this reason, the efficiency of computer vision methods attracted more attention. In this research, we try to explore the way to improve the efficiency of existing MOT methods. Due to the complexity of objects (millions

[1] Graduate School of Information Sciences, Tohoku Univercity
[2] Research Institute of Electrical Communication, Tohoku Univercity
[3] Cyberscience Center,Tohoku University
[a)] chenbo@ci.cc.tohoku.ac.jp
[b)] alfian@ci.cc.tohoku.ac.jp
[c)] izumi@ci.cc.tohoku.ac.jp
[d)] beto@tohoku.ac.jp
[e)] suganuma@tohoku.ac.jp

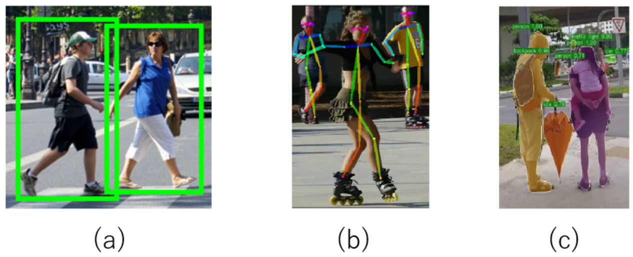


Fig. 1 Different outputs of pedestrian detectors. (a) bounding box (b) skeleton (c) mask

of categories), to simplify the research, we focus on pedestrian tracking.

The basic strategy of the proposed method is to adapt the most suitable trackers depending on the situation. We classified the states of pedestrians to non-occluded and occluded states. For each state we assign the most suitable tracker to it. We check the change of states, and deploy corresponding trackers to suit the new occlusion state. We evaluated the efficiency of existing trackers for different occlusion states. To demonstrate the effectiveness of the proposed method, we tested it on different scenarios, from less-occluded to highly-occluded ones.

2. Related Works

2.1 Detector of MOT

Object detection is a fundamental task in computer vision. Benefit from the extraordinary development of research in the last decades, many detectors have been proposed. According to the generation approach, they could be classified into two categories: handcrafted detectors and deep detectors.

The handcrafted detectors utilize handcrafted features to search objects in frames. In [10], they used Deformable Part Model (DPM) to search different parts of the object, and these parts are merged as integral objects. This detector significantly surpassed other contemporary detectors, therefore it has been widely used for pedestrian detection works, including MOT challenge [11] [12].

A significant trend of research for computer vision is the implement of deep learning algorithms. In the past decade, various deep detectors have been proposed. Faster RCNN [13] , YOLO [14] , SSD [15] are quintessential examples of deep detectors. With powerful neural networks, such as ResNet [16], these detectors achieved tremendous improvement and greatly changed the computer vision region.

In this research, we focus on pedestrian detectors. The output of pedestrian detectors would be different, it depends on the way to express pedestrians. **Fig. 1** shows the examples of different types of detectors. Some detectors utilized bounding box to show the pedestrian region, which is a rectangle around the pedestrian. Other way is skeleton, which is several line segments connect joints of different body parts. A more precise way is mask of pedestrians, which only involves pixels that belong to the body.

2.2 Tracker of MOT

The second component of MOT is the tracker. By information the tracker utilized, existing trackers could be categorized into online and offline types. The online type only utilizes previous and current frame's information, and the tracking result is unchangeable for previous frames, otherwise, the offline tracker relies on past and future frames' information. In general, offline tracker is more stable than online tracker. Similar to the detector, the trackers could be categorized into handcrafted and deep types.

The handcrafted trackers try to implement tracking by handcrafted methods. For the appearance model of tracker, optical flow, color histogram and HOG [17] are widely utilized. For the motion model, linear and non-linear models are proposed. For the inference model, Kalman filter [18] and particle filter [19] are adopted.

Various types of deep trackers have been proposed. Some trackers only applied deep detectors, and then accomplish tracking based on these detections. Another way is to exchange the handcrafted tracking methods to deep ones. In Deep SORT [5], they try to find occluded objects by deep appearance features. In [20], a LSTM [21] net is trained to handle long-term correspondences. Finally, there are trackers integrated deep detection and deep tracking, utilized the end-to-end network for the complete framework.

Based on the ability of handling occlusion, these trackers could be classified into occlusion-weak and occlusion-robust types. The strong point of the former one is, they rarely considered occlusion in algorithms, result in simpler and lighter complexity, and more efficiency on the non-occluded situation, but for occluded scenarios, they easily lose targets. The latter type has opposite features: with sophisticated algorithms of occlusion handling, it is more robust for occlusion but inevitably costs more computational resources.

2.3 Occlusion in MOT

Occlusion might be the most crucial challenge in MOT. There are several different types of occlusion for pedestrians in MOT.

Pedestrians may be occluded by obstacles, includes building, vehicle, vegetation, etc. In this situation, pedestrians may be completely covered by obstacles, which are generally bigger than the size of pedestrians, makes the latter invisible for tracker. When pedestrian completely disappears, the identity and trajectory of pedestrian usually be terminated and kept for a short period until the pedestrian reappears. If it disappeared for a very long period, the general operation is to stop tracking for this particular pedestrian.

The second possibility of occlusion is pedestrian occlude with each other, or inter-occlusion of pedestrians. In this situation, if the amount of pedestrians is relatively small, pedestrians would be covered by others and reappear rapidly, the tracker could find correspondence of occluded pedestrians by visible parts. On the other hand, if the scenario is highly crowded, it is very common that only a small part of pedestrians' body is visible during occlusion, which makes it become the most challenging situation.

The third situation is "self-occlusion" or "intra-occlusion". This is a very common scenario for pedestrians, with the movement of limbs, some parts of the body would appear and disappear regularly. The detector and tracker must have the ability to ignore this interference and keep detection and tracking robustly.

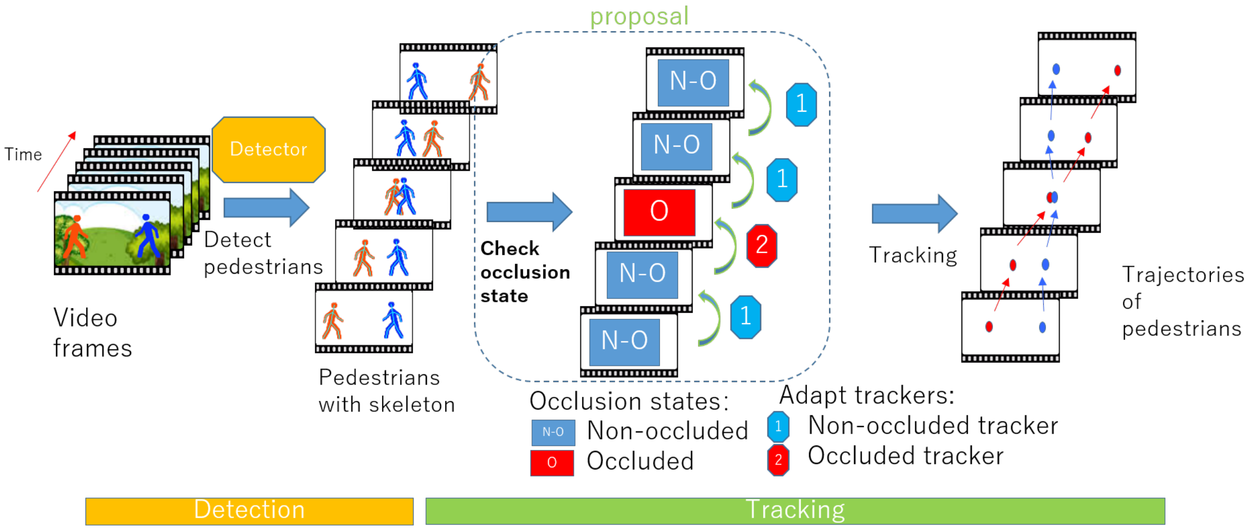


Fig. 2 Overview of the proposed method

In this research, for simplicity, we mainly focus on the second type of occlusion: inter-occlusion of pedestrians, for it's the hardest scenario of occlusion. If the detector and tracker could obtain satisfying performance under this scenario, it would keep the effectiveness under other scenarios.

2.4 Strategies of occlusion handling

The strategies of existing methods could be categorized into three types.

The first one is "Part-to-whole". This strategy based on the fact that during occlusion, generally some parts of objects still visible, the tracker could find correspondence by utilizing these parts. In [22], they divide a holistic object into several parts, and the consistency is obtained by integrating the similarity between corresponding parts. If a part is occluded, the affinity among the occluded part and non-occluded part is lower than usual, the tracker would ignore it and only integrate visible parts. In [23], which is a particular method for humans, they adopted the DPM detector, which separates the detection result into several body parts, and then appearance and motion trackers are utilized for these parts, finally, all parts are merged into final trajectories.

Another strategy is "Hypothesize-and-test". It treats the occlusion issue in MOT as a case of application of statistics. In [24], the occlusion hypothesis is generated based on occludable pair of observations, the characteristic of these pairs is close and with a similar scale. The occlusion is a distraction in the hypothesis. In the test process, the hypothesis of observation and original ones are input to a cost-flow framework and MAP is evaluated to obtain an optimal solution. In [25] and [26], occlusion patterns are used to assist detector, they generated different hypotheses based on the synthesis of two objects with different levels and patterns, and a detector is then trained on these hypotheses.

The third strategy is "Buffer-and-recover". This strategy buffers the observations and states of the object before occlusion and recovers the states after occlusion based on buffered observations and states. In [27], when occlusion happens, during up to 15 frames the trajectory is kept, and predict the possible trajectory if the object reappears, the predicted and observed trajectory is linked, and the identity is maintained. In [28], during occlusion the observation mode is activated until enough observation is obtained, the hypothesis is generated to explain the observation.

3. Proposal

3.1 Overview

The basic strategy of the proposal is the combination of the merits of two types trackers: the occlusion-weak tracker solves non-occluded tracking, and the occlusion-robust tracker handles occluded tracking. In this way, we could reduce the unnecessary computational cost for non-occluded pedestrians, thus improve the efficiency of whole tracking progress.

Fig. 2 shows the overview of the proposed method. For input video, firstly a pedestrian detector is applied on each frame, to give positions of pedestrians. Then we check the relationship between pedestrians, label the occlusion states (Non-occluded/Occluded). Then we assign the most suitable tracker for pedestrians. We also trace the variation of occlusion states, if it changed, we exchange the tracker promptly. Finally, the tracklets (i.e. short trajectories of tracked objects) of both trackers are merged to obtain the global trajectory of pedestrians.

3.2 Assumption of proposal

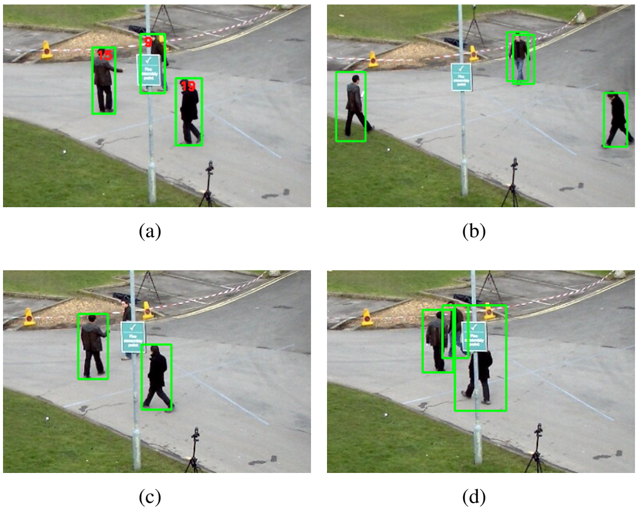
As the previous mention, the research objective of our work is pedestrian tracking. For this purpose, the assumption of this research as follows: We only track pedestrians in the video. Pedestrians without occlusion with obstacles, only inter-occlusion of pedestrians and self-occlusion will be considered. Each frame of video will be detected by a detector, and multiple pedestrians will be found out, then a tracker will be applied to these candidates of pedestrians, and output the trajectories of them.

3.3 Pedestrian detection

For each frame of video, pedestrian detection is implemented. The output involves positions of different pedestrians. The format of the result could be different types, involves skeleton, bounding box, contour, centroid, etc. The result of detection may involve many mistakes, as previously mentioned.

The ideal result of detection should be accurate: each bounding box covers only one pedestrian exactly, and even the pedestrian is occluded, the detector still could search it by visible part, and estimate the occluded part, provides proper coordinates of all four vertices of bounding box, as ground truth **Fig. 3**(a) shows.

The other possible issues of detection involves: multiple detection result for one pedestrian:Fig. 3(b), missed pedestrians by occlusion:Fig. 3(c), unsuitable size of bounding boxes:Fig. 3(d). These issues should be properly handled in the processing of detections.



(a) (b)

(c) (d)

Fig. 3 Ground truth and different issues of detection: (a) ground truth: exact bounding boxes for all pedestrians (b) detection issue: multiple bounding boxes for middle pedestrian (c) detection issue: can not detect middle occluded pedestrian (d) detection issue: larger bounding box for right pedestrian

3.4 Check occlusion states

Fig. 4 shows different occlusion situations of objects and our definition of occlusion states. In Fig. 4(a)(b), two balls move close to each other, without overlap, we classify this case to "Non-occluded" state. In Fig. 4(c)(d)(e), the "front" ball overlapped the "back" ball partially or completely, regardless how much they are overlapped, and what position (front/back) the object locates, we classify these cases to "Occluded" state.

With this definition, we could check the occlusion states of pedestrians by the relationship of positions: give bounding boxes of two pedestrians, if they are crossed, they are occluded with each other, otherwise, they are non-occluded. To express the relationship between bounding boxes quantitatively, here we utilized Intersection over Union (IoU) for calculation. Give two bounding boxes A and B, the IoU of them could be expressed as the following equation:

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \qquad (1)$$

The range of IoU is [0,1]. If two bounding boxes have no contact, the IoU is 0. If two bounding boxes perfectly overlapped, the IoU is 1.

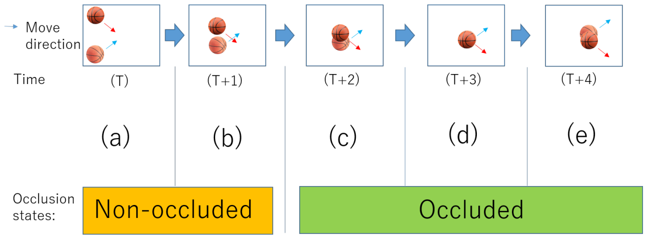


Fig. 4 Different situations of objects interaction and our definition of occlusion states: (a) Non-occluded (b) Non-occluded (c) Partial-occluded (d) Full-occluded (e) Partial-occluded. (c) (d) (e) are "Occluded" in our definition.

Table 1 Expected types of trackers

Types	FPS	MOTA
Occlusion-weak	higher	lower
Occlusion-robust	lower	higher

The detection issues should be handled in the whole tracking method, we try to solve the "multiple detection" issue by setting the threshold of IoU between two bounding boxes in the same frame. For ground truth, the threshold is 0, for unstable detection, the threshold should be larger than 0, but can not be too large to decrease the ability of occlusion detection.

3.5 Assign suitable trackers for corresponding occlusion states

This step is the start of tracking progress. We inspect the occlusion states of pedestrians and assign suitable trackers to it.

We check the speed and robustness to occlusion of different trackers, and classify them to different types: Speed: the speed of tracker indicates how fast it could implement for inputs, particularly, frame per second (FPS) is the metric to measure the performance of a tracker. For a video, higher FPS means faster speed. Robustness to occlusion: there are different metrics to express the performance of tracker for occlusion handling. MOTA is a widely used metric in the evaluation of MOT methods, the definition of MOTA is:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \qquad (2)$$

Where t is frame index, FN is false negative, FP is false positive, IDSW is ID switch, GT is ground truth objects. Generally, the MOTA is expressed as a percentage, the range is (-∞,100].

Our target is to improve the efficiency of existing tracking methods, so the suitable trackers we expected are:

Occlusion-weak tracker: this tracker should effectively track the objects without occlusion, which means the structure of the tracker should be simplified and the computational cost should be low, and the ability for occlusion handling is not necessary, i.e. higher FPS, lower MOTA.

Occlusion-robust tracker: this tracker should have robust occlusion handling ability, the computational cost of it must be higher than the former tracker, i.e. lower FPS, higher MOTA.

The expected types of trackers are shown in **Table 1**.

For objects that labeled "Non-occluded", the Occlusion-weak tracker will be utilized. Each pedestrian in detection will be assigned with an individual identity (ID) if it does not exist in any existing tracklets. If a new pedestrian occurs, a new tracklet will be created, and the position of the pedestrian will be added to

Table 2 Detail of data

Training sequences					
Name	Resolution	Length	Tracks	Boxes	Or
TUD-Stadtmitte	640x480	179	10	1156	0.50
TUD-Campus	640x480	71	8	359	0.72
PETS09-S2L1	768x576	795	19	4476	0.19
ETH-Bahnhof	640x480	1000	171	5415	0.54
ETH-Sunnyday	640x480	354	30	1858	0.63
ETH-Pedcross2	640x480	840	133	6263	0.87
ADL-Rundle-6	1920x1080	525	24	5009	0.82
ADL-Rundle-8	1920x1080	654	28	6783	0.54
KITTI-13	1242x375	340	42	762	0.37
KITTI-17	1242x370	145	9	683	0.56
Venice-2	1920x1080	600	26	7141	0.82

the tracklet. Then, the tracker will search the adjacent position in the next frame, and match all candidates of the new frame with the previous frame, and find best-matched pedestrians, add them to existing tracklets. If a tracklet can not be matched with any pedestrians in the new frame, then it will be suspended, until a new pedestrian could be matched with it.

For objects that labeled “Occluded”, the Occlusion-robust tracker will be utilized. The occlusion scenarios may be variable during occlusion, involves partial occlusion or complete occlusion. For partially occluded objects, the tracker could utilize visible parts to calculate correspondence with the previous frame’s objects, for completely occluded objects, the tracker usually infers the potential position and states by its well-designed occlusion handling algorithm. These trackers could verify the predicted position of the last frame’s pedestrians with the new detection, and correct the deviated predictions.

3.6 Switch of trackers

If the occlusion states of one pedestrian change from the previous frame, we will change the current tracker to a different one. For states from Occluded to Non-occluded, we just export the current position as well as the corresponding identity to an occlusion-weak tracker for further tracking. For state variation from Non-occluded to Occluded, the situation is more complex, we need to search the previous frames, recover the trajectory for the object, and then change the tracker to occlusion-robust one.

3.7 Combination of trajectories

Finally, the tracklets of two trackers are merged into the final result, to obtain global trajectories of all objects.

4. Experiment

4.1 Evaluation of occlusion detection method

To evaluate the effectiveness of our occlusion detection method in 3.4, we tested it on the public benchmark dataset.

We choose MOT Challenge 2015 [11] training set as our test dataset, which is a widely used benchmark for MOT. It contains images of different scenarios, with different occlusion intensities. It provides official detections of pedestrians, which is obtained by object detector from [29], based on Aggregated Channel Features (ACF). It also provides the ground truth, which indicates the precise position of pedestrians (by bounding box), and identities of them. The detail of data is shown in the **Table 2**.

To evaluate the occlusion intensities of these sequences, we define the occlusion ratio Or of each sequence s as:

$$Or(s) = \frac{amount\ of\ occluded\ bbox}{total\ amount\ of\ bbox} \qquad (3)$$

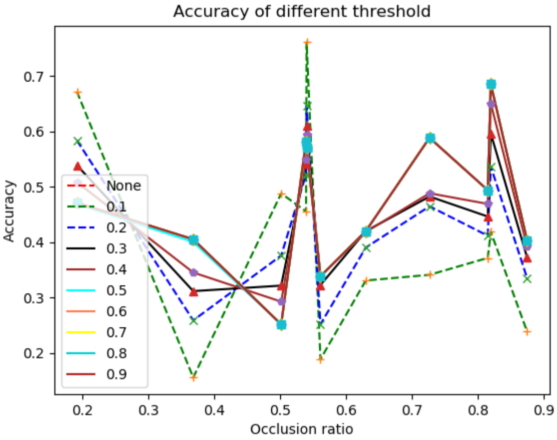


Fig. 5 Result of different thresholds of occlusion detection

Table 3 Setting of experiment

CPU	Intel Core i7-8700
Memory	16 GB
Occlusion-weak tracker	iou-tracker [7]
Occlusion-robust tracker	SORT-tracker [5]

The Or of each sequence is shown in the last column of Table 2.

We applied our occlusion detection method on these sequences, with different thresholds of IoU, which is used to control whether the two bounding boxes will be treated as the same person or not. **Fig. 5** shows the result. It shows comparison of results with threshold (from 0.1 to 0.9) and without threshold (None).

4.2 Evaluation of tracker switch strategy

To evaluate the efficiency of our proposal of tracker switch, we tested the strategy on existing trackers.

The setting of our experiment is shown in the **Table 3**. The detail of data is the same as the previous experiment.

In our assumption, the occlusion-weak tracker should be as fast as possible, and the occlusion-robust should be extremely stable for occlusion. Unfortunately, the state-of-the-art trackers are not satisfied, and the implementation of them also requires vast work. Therefore we checked existing open-sourced trackers, and selected suitable ones for our experiment.

The two trackers we used in the experiment are both handcrafted trackers. The occlusion-weak tracker is iou-tracker [7]. The assumption of iou-tracker is high fps frames, that the object’s movement between adjacent frames is very small, thus the tracker just searches the best-matched object from the current frame’s detection for the previous detection, which is defined by IoU of bounding boxes. Then the best match is updated to tracklet. If a tracklet cannot find a match with current detection, it is terminated, if a detection cannot be matched with the previous tracklet, the tracker will assign a new tracklet for it.

The occlusion-robust tracker is a more complex tracker: SORT-tracker [8]. We should point out that compare to the state of the art trackers, it is not the ideal choice since it does not have a

Table 4 Validation of two trackers

	MOTA	fps
iou-tracker	58.3%	12083
SORT-tracker	67.0%	380.3

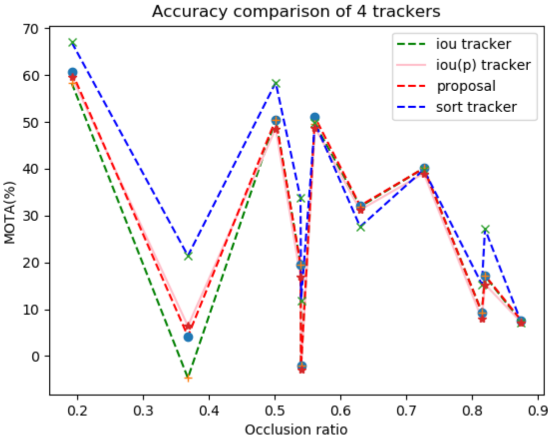


Fig. 6 Result of accuracy comparison of 4 methods.

very good occlusion handling algorithm. We choose this tracker due to its similar strategy of tracking with the iou-tracker. It also based on the IoU between previous frame and current frames' objects, and choose the objects with larger IoU than the pre-defined threshold. The difference between iou-tracker and SORT-tracker is the latter has a prediction part, which is achieved by Kalman filter [18], and the filter then associates previously predicted tracklets and current detections by linear segmentation. It is more complex than iou-tracker, which makes it much slower.

We validated the two trackers on the MOT dataset 2015, and **Table 4** shows the result.

From the result of the evaluation, we confirmed the iou-tracker has very high performance on bounding box based tracking (over 10000fps), but with lower MOTA. On the other hand, the SORT-tracker has a relatively slower speed (less than 1000fps) but has a higher MOTA. These attributes of the two trackers are what we expected.

Our strategy is the switch of different types of trackers. Due to the difficulty of combining the iou-tracker and SORT-tracker (they work on very different ways), we modified the iou-tracker to add the more robust feature as SORT-tracker: we always predict the potential position of next frame from the last trajectory and use this prediction for tracking of next frame. We name this tracker by "iou(p)-tracker". We combined the iou-tracker and iou(p)-tracker by our strategy, which switches different trackers according to the occlusion states of pedestrians. If it is non-occluded, we apply iou-tracker, if it is occluded, we apply iou(p)-tracker, we name this method by "proposal". We test all above trackers (SORT-tracker, iou-tracker, iou(p)-tracker, proposal) on dataset MOT 2015. As the experimental results, accuracy (MOTA) is shown in **Fig. 6**, and calculation is shown in **Fig. 7**.

4.3 Result and discussion

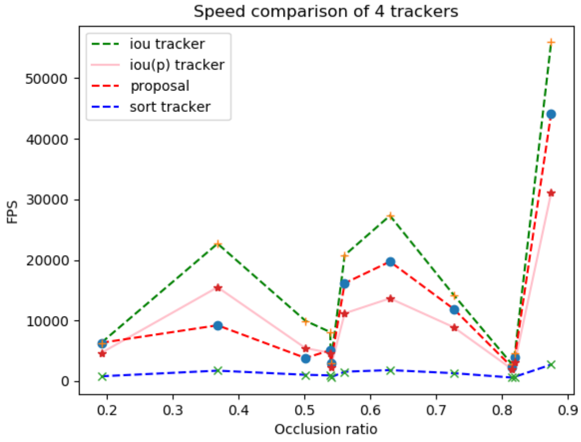


Fig. 7 Result of speed comparison of 4 methods.

From the result of occlusion detection with different thresholds Fig. 5, we noticed that threshold 0.4 could achieve the best accuracy in highly-occluded sequences (0.6-0.9), and the result is very unstable. This result shows the proposed method needs improvement to handle unstable pedestrian detections.

From accuracy result Fig. 6, the most complex SORT-tracker could achieve the highest accuracy in most scenarios, and the simplest iou-tracker is inverse, and the iou(p)-tracker could achieve slightly higher accuracy compare to our proposal for lower occlusion ratio scenarios, and in high occlusion ratio situation, all trackers obtained similar results, and the accuracy significantly decreased, shows the challenge of occlusion handling. This result shows these trackers can not deal with highly occluded situations, but for low occlusion ratio scenarios, our proposal tracker could achieve similar accuracy as iou(p)-tracker, which demonstrates the effectiveness of this strategy.

Fig. 7 shows the speed comparison. For all scenarios, the simplest iou-tracker is the fastest tracker, and the most complex SORT-tracker is the slowest. The modified iou(p)-tracker and proposal are intermediate. For lower occlusion ratio scenarios (0.2-0.5), due to the occlusion detection cost, the iou(p)-tracker is more efficient. However in higher occlusion ratio situation (0.6-0.9), our proposal reduced the computational cost significantly. This result indicates that the occlusion detection method must be efficient enough to offset the extra cost of itself, thus the proposal could achieve the expected performance and effectiveness.

The two results demonstrate that it is possible to improve the efficiency of trackers by combining fast-weak and slow-robust trackers. The current experiments only tested simple trackers, and we believe the strategy could be more effective on more complex trackers.

5. Conclusion and future work

We proposed a simple strategy to improve the efficiency of existing tracker methods. We combine different trackers, classify the objects by different occlusion states, and apply the most suitable tracker on them. We tested our proposal on existing datasets, and evaluate it under different situations.

In the future, we will try to utilize more stable features in our tracker, such as the intensity of illumination, optical flow, etc. A more challenging objective is the implementation of deep learning methods, in the current stage we tested handcrafted methods, the future work involves deep learning algorithms. We will test the strategy on more complex trackers to evaluate the effectiveness.

References

[1] Wang, X.: Intelligent multi-camera video surveillance: A review, *Pattern recognition letters*, Vol. 34, No. 1, pp. 3–19 (2013).

[2] Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B. and Kasturi, R.: Understanding transit scenes: A survey on human behavior-recognition algorithms, *IEEE transactions on intelligent transportation systems*, Vol. 11, No. 1, pp. 206–224 (2009).

[3] Uchiyama, H. and Marchand, E.: Object detection and pose tracking for augmented reality: Recent approaches, *18th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)* (2012).

[4] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X. and Kim, T.-K.: Multiple object tracking: A literature review, *arXiv preprint arXiv:1409.7618* (2014).

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

[6] Sadeghian, A., Alahi, A. and Savarese, S.: Tracking the untrackable: Learning to track multiple cues with long-term dependencies, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 300–311 (2017).

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

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

[9] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A. and Le, Q. V.: Mnasnet: Platform-aware neural architecture search for mobile, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828 (2019).

[10] Felzenszwalb, P. F., Girshick, R. B., McAllester, D. and Ramanan, D.: Object detection with discriminatively trained part-based models, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 32, No. 9, pp. 1627–1645 (2009).

[11] Leal-Taixé, L., Milan, A., Reid, I., Roth, S. and Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking, *arXiv preprint arXiv:1504.01942* (2015).

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

[13] Ren, S., He, K., Girshick, R. and Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, pp. 91–99 (2015).

[14] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A.: You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788 (2016).

[15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C.: Ssd: Single shot multibox detector, *European conference on computer vision*, Springer, pp. 21–37 (2016).

[16] He, K., Zhang, X., Ren, S. and Sun, J.: Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (2016).

[17] Dalal, N. and Triggs, B.: Histograms of oriented gradients for human detection (2005).

[18] Reid, D.: An algorithm for tracking multiple targets, *IEEE transactions on Automatic Control*, Vol. 24, No. 6, pp. 843–854 (1979).

[19] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E. and Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter, *2009 IEEE 12th International Conference on Computer Vision*, IEEE, pp. 1515–1522 (2009).

[20] Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M. and Tran, D.: Detect-and-track: Efficient pose estimation in videos, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 350–359 (2018).

[21] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).

[22] Hu, W., Li, X., Luo, W., Zhang, X., Maybank, S. and Zhang, Z.: Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 12, pp. 2420–2440 (2012).

[23] Izadinia, H., Saleemi, I., Li, W. and Shah, M.: 2 T: multiple people multiple parts tracker, *European Conference on Computer Vision*, Springer, pp. 100–114 (2012).

[24] Zhang, L., Li, Y. and Nevatia, R.: Global data association for multi-object tracking using network flows, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1–8 (2008).

[25] Tang, S., Andriluka, M. and Schiele, B.: Detection and tracking of occluded people, *International Journal of Computer Vision*, Vol. 110, No. 1, pp. 58–69 (2014).

[26] Tang, S., Andriluka, M., Milan, A., Schindler, K., Roth, S. and Schiele, B.: Learning people detectors for tracking in crowded scenes, *Proceedings of the IEEE international conference on computer vision*, pp. 1049–1056 (2013).

[27] Mitzel, D., Horbert, E., Ess, A. and Leibe, B.: Multi-person tracking with sparse detection and continuous segmentation, *European Conference on Computer Vision*, Springer, pp. 397–410 (2010).

[28] Ryoo, M. S. and Aggarwal, J. K.: Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1–8 (2008).

[29] Dollár, P., Appel, R., Belongie, S. and Perona, P.: Fast feature pyramids for object detection, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 36, No. 8, pp. 1532–1545 (2014).