Current Challenges with Modern Multi-Object Trackers

Tomasz Stanczyk, Francois F Bremond

Current Challenges with Modern Multi-Object Trackers

Tomasz Stanczyk Francois Bremond
Inria centre at Université Côte d'Azur
2004 Rte des Lucioles, 06902 Valbonne, France
name.surname@inria.fr

Abstract

Multi-object tracking algorithms reach impressive performance on the benchmark datasets that they are trained and evaluated on, especially with their object detector parts tuned. When these algorithms are exposed to new videos though, the performance of the detection and tracking becomes poor, making them not usable. This paper tries to understand this behavior and discusses how we can move forward regarding these issues. Besides, we present the common errors made by the modern trackers, even when their trainable components are heavily tuned on the datasets as well as propose some directions and high-level ideas on how to proceed with those problems. We hope that it will lead to interesting discussions in the community towards solving such errors and further improving the performance of multi-object tracking.

1. Introduction

Multi-object tracking (MOT) is a task of associating the same objects in a video sequence across many frames. Along with the introduction of MOT challenge benchmark datasets, e.g. MOT17 [5], the development of MOT algorithms greatly progressed leading to recent development of algorithms with notable performance improvement. Current trend implies state-of-the-art algorithms following the paradigm of tracking by detection.

However, despite all the progress, there are still errors made by the MOT algorithms, which prevent MOT from being fully reliable in its application scenarios as an assistive technology and beyond. In this paper, we are identifying remaining errors made by these algorithms and provide potential direction suggestions on how to tackle them.

We select three recent state-of-the-art tracking algorithms corresponding to the main trends in modern MOT: ByteTrack [6], which uses very good object detections and performs optimized tracklet association with well-engineered tracklet management, BoT-SORT [1] which builds on the top of that and adds appearance features, and SUSHI [2], which considers more features for association, reasons over whole videos and manages to cover longer trajectories. We briefly describe the methodologies and behavior of these trackers as well as point out observed errors with visualization examples.

Further, we discuss in general what are the remaining and common errors produced by state-of-the-art MOT algorithms, where these errors possibly come from and then propose how to approach them with sketched directions of improvement. Among the others, we discuss the importance of using relevant cues for association of the detections and combining them appropriately.

Further, we highlight poor generalization of the state-of-the-art MOT algorithms, e.g. the performance of the tracker with public and private detections as well as visual performance on a custom, unannotated video. It relates to the need for training on specific datasets and the usability of an algorithm by the third-party. MOT trackers perform well on private and fine-tuned detections, worse on public detections and poorly on custom videos. We claim that this is a major issue and an important part to improve the global performance of MOT algorithms.

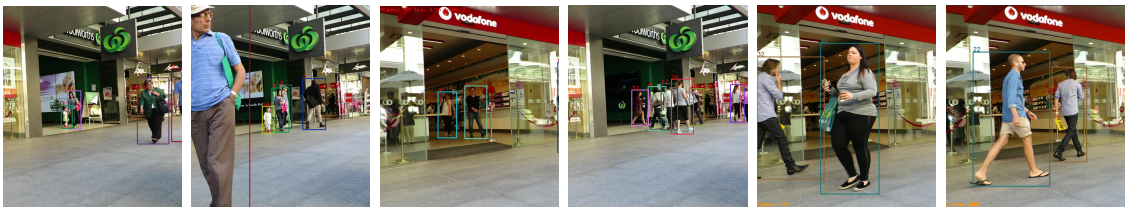
Finally, we also raise the subject of the reliability of MOT and provide remarks towards improving it, including discussion over more precise metrics and the existing yet underrated ones.

2. State-of-the-art MOT algorithms

The performance of MOT algorithms keeps improving with the recent state-of-the-art trackers obtaining impressive results on publicly available benchmark datasets. Nevertheless, these algorithms still make mistakes in certain scenarios, which limits their reliability. In this section, we focus on three recent state-of-the-art MOT trackers, where we briefly characterize their methodologies and identify the most significant remaining errors they make.

2.1. ByteTrack

ByteTrack [6] is an efficient MOT algorithm processing the consecutive frames of a video sequence on the fly, i.e.



(a) (b) (c) (d) (e) (f)

Figure 1: Examples of issues faced by the discussed algorithms. (a) and (b) visualize output of ByteTrack [6], (c) and (d) of BoT-SORT [1], and (e) and (f) of SUSHI [2]. The presented sequence comes from the MOT17 dataset [5]. Best viewed digitally.

it does not access the future frames while processing the current one. It uses a strong and robust object detector, YOLOX [4] which is pre-trained on a MOT dataset of interest. During the association process between the existing tracklets and new detections, ByteTrack considers bounding boxes with both high and low confidence scores so as to recover occluded objects. It predicts next locations of the tracks with Kalman Filter and associates them with the new bounding boxes based the intersection over union (IoU) metrics. It is a well-engineered algorithm with robust tracklet management system. It servers as a very good baseline without any association learning scheme or visual cues.

Even though ByteTrack keeps the tracks of the major part of the targets, the association simply based on IoU leads to many identity switches, when id number of one tracked subject is assigned to another. It is especially visible in the crowded scenarios and over people positioned close to each other. Further, ByteTrack tends to lose its targets after relatively short occlusions. More than one id number gets assigned to a single person, which leads to split tracklets. An example of the aforementioned issues is presented in Fig. 1a and 1b. Over the same sequence and following the occlusions caused by the people with id 1 and 5, the person with id 3 gets a new id, 14, whereas the person staying next to them gets exactly their id, 3 after losing their own, 2. Such situations make the tracking output unreliable for further processing or applications.

2.2. BoT-SORT

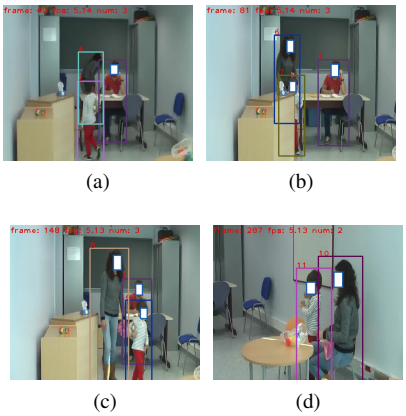
BoT-SORT [1] is a MOT algorithm built on the top of ByteTrack, thus it also processes the frames on the fly and uses the pre-trained YOLOX object detector. Its extensions include incorporated camera motion compensation and adjusted Kalman filter as well as fusing IoU and re-identification (re-ID) features. The re-ID features are extracted via person re-ID model trained on the particular dataset (e.g. MOT17).

Compared to ByteTrack which uses only the IoU for its associations, the number of identity switches among the Bot-SORT output tracklets is significantly reduced when applying person re-ID features and other extensions. However, the presence of identity switches still prevails, together with split tracklets despite using these re-ID features, even when trained specifically on the MOT datasets. Some of the associations performed between tracklets and detections are incorrect and not all the targets are kept tracked under an occlusion. An example is presented in Fig 1c and 1d, where people with completely different appearances and temporal differences in their presences are assigned to the same id number 4 without any constraints.

2.3. SUSHI

SUSHI [2] is a MOT algorithm accessing all frames at the time as a global optimization approach. It is a graph based association method, which for its detections uses YOLOX trained on the considered dataset in the same manner as ByteTrack and BoT-SORT. SUSHI hierarchically joins tracklets starting from length 1 (single detections) into 512-frame long tracklets. It extracts features (association cues) through re-ID architecture analogous to that from BoT-SORT (appearance) and via mathematical derivations (motions consistency, time and position information). Such initial association features are passed to a multi-layer perceptron. The extracted embeddings are then propagated across the graph (edges) to perform joining of the tracklets (nodes). The weights of the graph are shared over the architectural building blocks, each for a different hierarchy level of a sub-clip processing.

The SUSHI tracking algorithm behaves exceptionally well and reaches remarkable performance by handling many challenging cases such as occlusion and crowded groups, especially those with which ByteTrack and BoT-SORT face issues. Nevertheless, there is room for improvement as some identity switches are still present, especially when people appear in the scene, as if initiating a track was not constrained enough. When one person appears on the scene, SUSHI has a tendency to link them to the person who has already been on the scene before. An example is pre-



(a) (b) (c) (d)

Figure 2: Visual performance of BoT-SORT with its components trained on the MOT17 dataset applied to our custom video. Best viewed digitally.

Algorithm	MOTA	IDF1	HOTA	MT	ML	ID Sw.
BT priv.	80.3	77.3	63.1	1,254	342	2,196
BT pub.	67.4	70.0	56.1	730	735	1,331
SUSHI priv.	81.1	83.1	66.5	1,374	312	1,149
SUSHI pub.	62.0	71.5	54.6	801	741	1,041

Table 1: Numerical performance comparison of ByteTrack (BT) and SUSHI on MOT17 dataset when using private and public detections.

sented in Fig. 1e and 1f, where person with id number 22 leaves the scene and then another person enters the scene from the same side and receives the same id. The two persons do not appear together on the scene and the association is made despite very different appearances. Further, there are tracklets which are split and one person might have two identities despite the total length of the tracklet being less than aforementioned 512 frames.

Lastly, SUSHI is an offline process, as it accesses all the video sequence frames at once. Remaining challenge is to make it more generalized and online as it is in the case of ByteTrack and BoT-SORT.

3. Discussion

3.1. MOT tracker's errors

Despite their excellence, MOT state-of-the-art trackers still face issues, even with datasets on which their components such as object detectors and re-ID extractor were trained. The most apparent errors include incorrect associations resulting in identity switches and initiating a new tracklet with the person, who has already been on the scene, as well as losing targets due to occlusions, resulting in split tracklets.

Using additional cues, e.g. appearance features seems to improve the tracking process as it visible in the performance of ByteTrack [6] and BoT-SORT [1] which is built on top of the former with re-ID features involved. However, using even a re-ID extractor trained on the dataset of interest is not sufficient. The associations made are unreliable and the errors still prevail as demonstrated in Fig 1c and 1d. The association process requires more information in the form of additional cues fused properly and appropriate constraints to reduce the errors.

3.2. Proposed directions

As each type of an association cue brings additional information, all available types should be taken into account. It includes e.g. appearance, motion, time and location cues. Starting from ByteTrack utilizing a powerful object detector and location information, further performance improvement is obtained in BoT-SORT, which also applies re-ID features. SUSHI [2], which also extracts time and motion features and passes them through a multi-layer perceptron to a multi-layered graph structure reaches even more improvement gain.

One type of cues cannot just be simply added next to another one (e.g. IoU and re-ID) as it could result in chaotic and thus unreliable connection of bounding boxes of different entities in challenging scenarios such as occluded and crowded spots. More complex manner for combining cues is required and it remains as an important problem which we believe once addressed, can significantly boost the performance of MOT trackers.

Further, as it is shown in Fig. 1e and 1f, even more complex approach of combining the cues from SUSHI struggles with an incorrect association of two different subjects. It tends to link newly appeared subjects to those which have already been present on the scene. Potential direction with respect to this issue could be to impose more constraints for the association process, e.g. based on the context information, to build more confidence before linking.

Current state-of-the-art MOT trackers might produce corrupted tracklets with more than one identity (identity switch). One of the potential solutions could be to perform tracklet split, e.g. based on the appearance features. Only bounding boxes with relatively similar appearances would be stored in one tracklet. Further, one identity might be spread over more than one tracklet, which might happen both with and without the aforementioned extra split. The next step could involve joining the tracklets based on the available cues and their proper combinations together with relevant constraint imposed. E.g., two tracklets of possibly different lengths could be joined into one if the subjects look similar, but they also keep similar motion and their location information is within the considered displacement bounds.

Further, a good re-ID mechanism should be provided, which among the others would be able to identify a person in the crowd and track them merely by their appearance parts. It should be also robust to poor lighting conditions.

3.3. Generalizability and reliability

Modern MOT tracker performance looks very good on the presented datasets, but it is dependent on trainable components such as object detector and appearance feature re-ID extractor, which in turn reduces the generalizability. Table 1 presents selected available numerical results comparing performance of ByteTrack and SUSHI with private detections, where the algorithms generate their own detections, and with public detections, which are provided by the dataset authors. The comparison[1] is made on the test set of the MOT17 dataset [5]. As visible in the table, the performance of the algorithms is dependent on the training phase. For the private detections, both ByteTrack and SUSHI use a powerful object detector YOLOX [4] trained on the dataset of interest, e.g. MOT17. High ranking results on the considered datasets are obtained due to the very accurate private detections provided and the performance mostly drops when using the public detections, which in fact are already good, but not heavily tuned on the dataset of interest as it is in the case of the private detections.

Further, the trackers cannot be smoothly generalized to new, not annotated videos provided by the user as their performance is heavily dependent on a trainable object detector and a re-ID feature extractor. Based on the available code, we run one of the trackers, BoT-SORT with its components trained on MOT17 data on our custom videos and present the visual output in Fig. 2a- 2d. Our video present relatively not difficult scenario with three subjects involved, yet even within its short duration, many errors are made by the tracker. Not only are the tracklets of the two main subjects split into many shorter tracklets, but they also suffer from several identity switches. It shows how sensitive re-ID features are to the video scene and that modern MOT trackers can perform well on the MOT datasets when tuned, but poorly on the new video scene.

We claim that important direction to consider in the MOT community is going towards a tracker which once set, can be used on other videos easily, without requiring too much (or ideally none) tuning on the data. An interesting direction could be new mechanisms for learning on the fly with no annotation provided, e.g. utilizing unsupervised re-ID as in [3]. Currently, all the approaches are tuned and thus performance-specific to the existing MOT datasets. We suggest that for the new MOT challenges, the test videos are not similar to the training ones.

Further, to make MOT more reliable, the produced tracklets should cover the whole trajectories of human subjects. We argue that mostly tracked (MT) and mostly lost (ML) metrics related to MOT, are underrated and should be further considered in the evaluations of MOT algorithms. Mostly tracked measures the amount of the objects being tracked for at least 80% and mostly lost for at most 20% of their lifespan, which demonstrates the algorithm coverage of the actual track. Further, we propose to introduce a more-fine grained set of metrics, such as mostly tracked, yet measuring the number of objects being tracked for at least 90 or even 95% of their lifespan. In this manner, more precise track coverage can be measured and thus more reliable algorithms can be developed. Besides, we claim that more significant tracklets should be taken into account to receive meaningful evaluation. In other words, counting short tracklets should be limited as it adds noise to the metrics and thus it prevents clear observations on the algorithm performance.

[1] Available at: https://motchallenge.net/results/MOT17/?det=All.

4. Conclusion

As discussed in the paper, MOT algorithms still make errors leading to identity switches, split or even missed tracklets. Further, their performance degrades when using public detections in place of the private ones and becomes poor on custom videos outside the benchmark datasets. With the errors we mentioned and the directions we proposed, we hope to stimulate interesting discussions in the community and developments in the field towards improving the global performance of MOT.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

[2] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22877–22887, June 2023.

[3] Hao Chen, Benoit Lagadec, and François Bremond. Ice: Inter-instance contrastive encoding for unsupervised person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14960–14969, 2021.

[4] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

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

[6] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. 2022.