ReMOTS: Self-Supervised Refining Multi-Object Tracking and Segmentation (1st place solution for MOTSChalelnge 2020 Track 1)

Fan Yang[*1,2], Xin Chang[1], Chenyu Dang[1], Ziqiang Zheng[3], Sakriani Sakti[1,2], Satoshi Nakamura[1,2], and Yang Wu[4]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project, Japan
[3]UISEE Technology (Beijing) Co. Ltd., China
[4]Kyoto University, Japan

Abstract

We aim to improve the performance of Multiple Object Tracking and Segmentation (MOTS) by refinement. However, it remains challenging for refining MOTS results, which could be attributed to that appearance features are not adapted to target videos and it is also difficult to find proper thresholds to discriminate them. To tackle this issue, we propose a self-supervised refining MOTS (i.e., ReMOTS) framework. ReMOTS mainly takes four steps to refine MOTS results from the data association perspective. (1) Training the appearance encoder using predicted masks. (2) Associating observations across adjacent frames to form short-term tracklets. (3) Training the appearance encoder using short-term tracklets as reliable pseudo labels. (4) Merging short-term tracklets to long-term tracklets utilizing adopted appearance features and thresholds that are automatically obtained from statistical information. Using ReMOTS, we reached the 1st place on CVPR 2020 MOTS Challenge 1 [4], with a sMOTSA score of 69.9.

1. Introduction

Multiple Object Tracking (MOT), which depends on information from the bounding box, faces a great challenge, since different objects may stay in the same bounding box and increase the ambiguity to distinguish them. Recently, some researchers in this filed have moved their eyes to Multiple Object Tracking and Segmentation (MOTS) and hope to take advantage of object-instance masks. Under such a background, the first MOTS challenge is organized to explore solutions for MOTS. We participated in this challenge

---

[*]Corresponding email: yang.fan.xv6@is.naist.jp

(May-30th-2020) and won the 1st place on Challenge 1. In this paper, we represent our solution.

2. Method Details

Overall, we apply the tracking-by-detection strategy to generate MOTS results. Since our ReMOTS is an offline approach, we refine the data association by retraining the appearance feature encoder. In each step of ReMOTS, we give a practical guidance to quantitatively select hyperparameters. Our approach is illustrated in Figure 1. After obtaining object-instance masks, we perform: (1) encoder training with intra-frame data, (2) associate masks to short-term tracklets by a short-term tracker, (3) inter-short-tracklet encoder retraining, and (4) merging short-term tracklets.

2.1. Generate Object-Instance Masks

Referring to how the public detection is generated, we obtain object-instance masks using the Mask R-CNN X152 of Detectron 2 [5] and X-101-64x4d-FPN of MMDetection [2]. We fuse their segmentation results by a modified Non-maximum Suppression (NMS). Unlike the traditional NMS, where the IoU (Intersection over Union) is applied, we propose a new metric named IoM (Intersection over Minimum) for it since heavily overlapped masks may also have low IoU values. The python code of IoM is as follows.

```python
def pixel_iom(target,prediction):
    """
    Inputs:
        target: binary mask, array([H,W])
        prediction: binary mask, array([H,W])
    Outputs:
        iom_score: float
    """
```
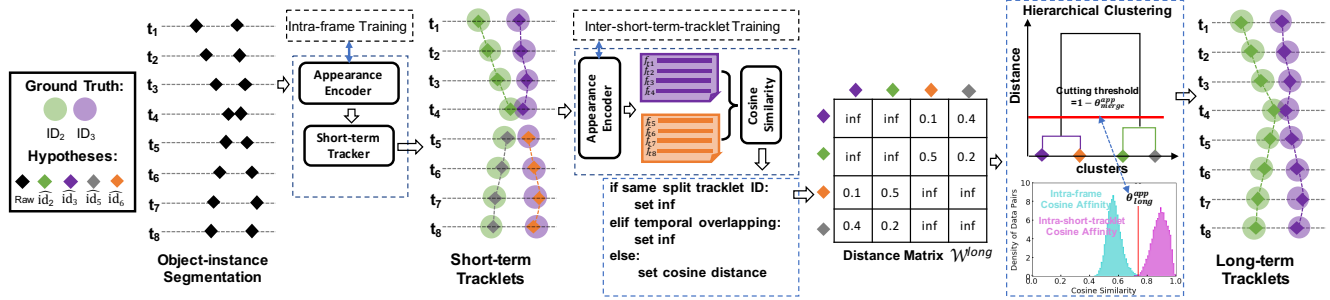
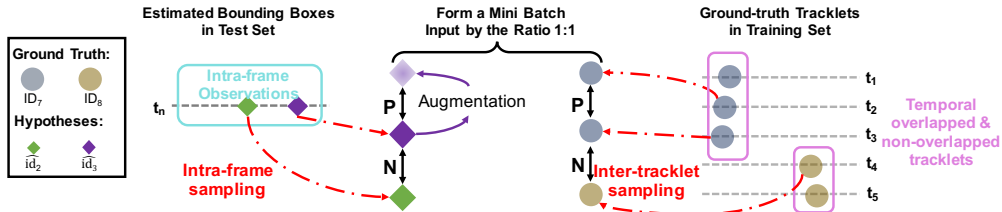Figure 1: The illustration of ReMOTS Framework.

Figure 2: Constructing training samples for intra-frame training. P and N represent positive and negative samples, respectively.

```
9    intersection = np.logical_and(target,
     prediction)
10   min_area = min(np.sum(target),np.sum(
     prediction))
11   iom_score = np.sum(intersection) / min_area
12   return iom_score
```

After performing our modified NMS, the remaining masks may still have overlapped areas. Therefore, we only keep the mask with the top confident score for each overlapping area.

2.2. Encoder Training with Intra-frame Data

We take an off-the-shelf appearance encoder and its training scheme from an object re-identification work [3]. SeResNeXt50 is used as the backbone and its global-average-pooling output, which is a 2048-dimension vector, is used as the appearance features. The triplet loss [1] is applied to train the appearance encoder. To adapt the appearance feature learning to the target videos, we incorporate intra-frame observations of target videos into a novel offline training process.

As Figure 2 shows, we can sample triplets from the training set only referring to the ground-truth tracklets. In test set, since Non-maximum Suppression (NMS) is performed, we assume that predicted object masks are exclusive within the same frame, and therefore it is easy to form negative pairs with intra-frame observations. Before tracking, we create a positive sample by augmenting an anchor sample. The augmentation process can dramatically change the pixel content of the anchor sample without altering identity. Finally, we take triplets from the training set and target set to form a mini-batch input by the ratio of $1:1$. Using such

new training samples, we retrain the appearance encoder to obtain more discriminative appearance features.

2.3. Short-term Tracker

After intra-frame training, we apply the appearance encoder to generate appearance features for data association. Since the tracker part is not our main focus, we build a simple tracker that only associates two-frame observations at once. Using the dense optical flow function of OpenCV, we generate optical flow between two adjacent frames, and then warp the mask from previous frame to current frame to calculate IoU of cross-frame masks. The distance matrix is formulated as follows:

$$\mathcal{W}_{prev,curr}^{short} =$$
$$\begin{cases} inf, & if\ IoU(mask_{prev}, mask_{curr}) = 0 \\ 1 - \frac{f_{prev}f_{curr}}{\|f_{prev}\|\|f_{curr}\|}, & otherwise \end{cases}$$
(1)

where $mask_{prev}$ and $mask_{curr}$ respectively denote the mask of the previous frame and the mask of the current frame; $\mathcal{W}_{prev,curr}^{short}$ is their edge weight (i.e., distance); f_{prev} and f_{curr} are their appearance features.

Besides constraining data association with low IoU values, we also hope to constrain data association with low appearance similarity. However, it is tricky to heuristically determine a threshold for constraining. We tackle this issue by analyzing the intra-frame distribution. Specifically, the histogram of appearance cosine similarity between intra-frame masks can be approximated by a normal distribution, and within three standard deviations is 99.7% of the observation pairs (see Figure 4). We set an appearance affinity threshold
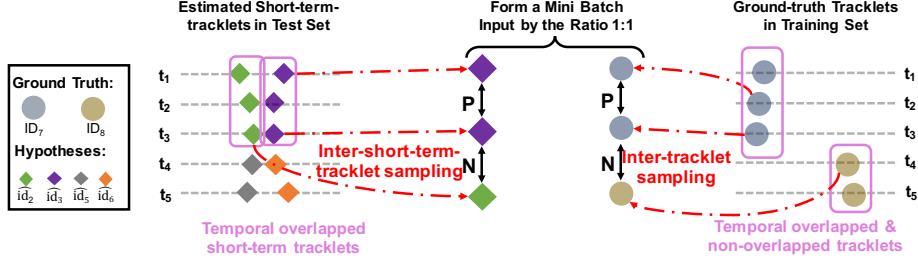
Figure 3: Constructing training samples for inter-short-tracklet training. P and N represent positive and negative samples, respectively.

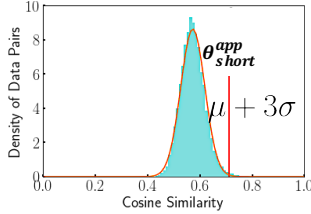at three standard deviations, as value θ_{short}^{app}.

Figure 4: The appearance threshold value for short-term tracking.

Consequently, using automatically obtained θ_{short}^{app}, we further process $\mathcal{W}_{prev,curr}^{short}$ by

$$\mathcal{W}_{prev,curr}^{short} = \begin{cases} inf, & if\ \mathcal{W}_{prev,curr}^{short} > 1 - \theta_{short}^{app} \\ \mathcal{W}_{prev,curr}^{short}, & otherwise \end{cases}. \quad (2)$$

We apply linear assignment on $\mathcal{W}_{prev,curr}^{short}$ to determine the association of masks between the previous frame and the current frame. Due to misdetection and occlusion, such a process can only generate short-term tracklets. However, short-term tracklets reduce the risk of mixing different identities, which is an important condition in next-step process.

2.4. Inter-short-tracklet Encoder Retraining

As we assume each short-term tracklet may only contain a unique identity, they can be used as dependable pseudo labels to train the feature encoder. However, different short-term tracklets, which have no overlap in the temporal domain, may still hold the same identity. Therefore, we do inter-short-tracklet retraining under the constraint that sampled short-term tracklets must be temporally overlapping within the same video.

We illustrate the process of training data sampling for inter-short-tracklet training as Figure 3 shows. Within a video, we first sample two identities that appear in a randomly chosen frame, and then randomly choose another frame for one of the selected identities, thus constructing a triplet. Other settings of inter-short-tracklet training are the same as intra-frame retraining. We update the appearance features after inter-short-tracklet retraining and use them in the next step.

2.5. Merging Short-term Tracklets

With better appearance features and more robust spatio-temporal information of short-term tracklets, we are able to merge short-term tracklets into long-term ones. The merging process is summarized in Figure 1. Short-term tracklets association is formulated as a hierarchical clustering problem on a weighted graph, in which each node represents a tracklet and the graph edges are represented in a distance matrix $\mathcal{W}^{long}$, defined as

$$\mathcal{W}_{k1,k2}^{long} =$$
$$\begin{cases} inf, & if\ k1 = k2 \\ inf, & if\ Distance(\Pi_{k1}, \Pi_{k2}) > \theta^t \\ inf, & if\ \Pi_{k1} \cap \Pi_{k2} \neq \varnothing \\ \frac{1}{N_{k1}N_{k2}} \sum_{i \in \Pi_{k1}} \sum_{j \in \Pi_{k2}} \left(1 - \frac{f_i^{k1} f_j^{k2}}{\|f_i^{k1}\|\|f_j^{k2}\|}\right), & otherwise \end{cases},$$
$$(3)$$

where for tracklets T_{k1} and T_{k2}, $\mathcal{W}_{k1,k2}^{long}$ is their edge weight (i.e., distance); Π_{k1} and Π_{k2} are their temporal ranges; f_i^{k1} and f_j^{k2} are their appearance features at frame i and j, and N_{k1} and N_{k2} are the number of observations within the tracklets, respectively.

Whenever the matching condition between two short-term tracklets violates any of the following three principles: (1) different short-term track ID, (2) the temporal gap between two short-term tracklets are within θ^t frames (we use $\theta^t = 15$), and (3) no temporal overlap between two short-term tracklets, we set their distance value to be infinite. To hold these constraints in the whole process of hierarchical clustering, we apply the centroid linkage criteria to determine the distance between clusters.

The main challenge of applying hierarchical clustering is on how to set a proper cutting threshold. We do not give a heuristic value, and we let the data speak for themselves instead. We suppose that intra-frame and inter-short-tracklet cosine similarity histograms can be separated at θ_{merge}^{app} (see Figure 5) after inter-short-tracket retraining, though small overlapping might exist. Without accessing to the ground-truth, this could be a reasonable boundary to distinguish objects based on appearance features. Therefore, we set $1 - \theta_{merge}^{app}$ as the cutting threshold in hierarchical clustering.

Rank	Method	sMOTSA↑	IDF1↑	MOTSA↑	MOTSP↑	MODSA↑	MT↑	ML↓
1st place	ReMOTS (ours)	69.9	75.0	83.9	84.0	85.1	248	12
2nd place	PTPM	68.8	68.5	82.6	84.1	83.7	244	19
3rd place	PT	68.4	64.9	82.6	83.9	84.4	248	10

Table 1: The performance on CVPR 2020 MOTS Challenge test set (up to submission deadline at May-30th-2020).

Sequence	Method	sMOTSA↑	IDF1↑	MOTSA↑	MOTSP↑	MODSA↑	MT↑	ML↓
MOTS20-01	ReMOTS	68.5	81.9	83.9	82.5	84.8	8	0
MOTS20-06	ReMOTS	74.9	76.8	88.8	85.0	90.6	156	4
MOTS20-07	ReMOTS	65.4	68.3	80.6	81.9	81.6	36	4
MOTS20-12	ReMOTS	71.8	82.0	82.9	87.2	83.8	48	4

Table 2: The performance of ReMOTS on each sequence of CVPR 2020 MOTS Challenge test set (up to submission deadline at May-30th-2020).
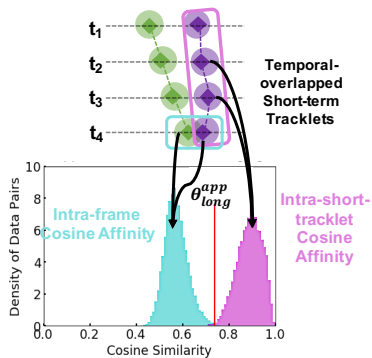

Figure 5: The appearance threshold value for merging short-term tracklets.

2.6. Experimental Setup

The only neural network model - appearance encoder [3], used in this work, is not our contribution and our ReMOTS can do the same refining when other appearance models are used. Therefore, we do little change to the default setting of [3], except for forming novel training samples in our intra-frame training and inter-short-tracklet training. Here, we omit the other details described in [3].

3. Results

We report the performance of our ReMOTS on the MOTChallenge evaluation system, with metrics introduced in [4]. In Table 1, we list the performance of top-3 methods up to the submission deadline. Our method mainly outperform the other two methods in terms of IDF1 score, and therefore leads to state-of-the-art performance in this challenge. The detailed performance on each test sequence is listed in Table 2. Though the same method is applied, it can be observed that the performance of each sequence varies a lot. This may be attributed to the diversity between videos, which calls for more exploration in automatically adapting MOTS models to target videos. Our ReMOTS analyzes the statistical information at the entire video level, but the temporal local statistical information, which might be useful for

fine-grained adaption, has not been considered yet.

4. Conclusion

We present our solution which wins the CVPR 2020 MOTS Challenge 1. In our proposed ReMOTS framework, intra-frame training and inter-short-tracklet training are introduced for learning better appearance features for more effective data association, which are our main contributions. Besides, we quantitatively demonstrate how to select proper thresholds by analyzing the statistical information of tracklets, which could be useful for other multiple object tracking works. The main limitation of ReMOTS is that it cannot be used in real-time scenarios, but it may bring insights to design better online MOTS method with feature adaptation.

ACKNOWLEDGEMENTS

References

[1] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.

[2] et al. Chen, Kai. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[3] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[4] Paul Voigtlaender, Michael Krause, Aljoša Ošep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *CVPR*, 2019.

[5] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2.