





# Enhancing Multi-Object Tracking with Segmentation Masks: A Solution for Lost Object Recovery

Manuel Bendaña<sup>1</sup>, Lorenzo Vaquero<sup>2</sup>, Victor M. Brea<sup>1</sup>, and Manuel Mucientes<sup>1</sup>

<sup>1</sup> CiTIUS, Universidade de Santiago de Compostela, Spain  
{manuel.bendana.gomez,victor.brea,manuel.mucientes}@usc.es

<sup>2</sup> Fondazione Bruno Kessler, Trento, Italy  
lvaquero@fbk.eu

**Abstract.** Tracking by detection is an effective approach to addressing the multiple object tracking problem. Detections are extracted and matched across the different frames of a video. However, detection errors persist, leading to false negatives that degrade tracker performance. In this work, we propose an architecture to overcome detection failures. Instead of using bounding boxes, which lack precision in crowded situations, we propose obtaining and tracking segmentation masks for each object. Results on the MOT20 crowded dataset demonstrate our ability to improve the performance of state-of-the-art methods.

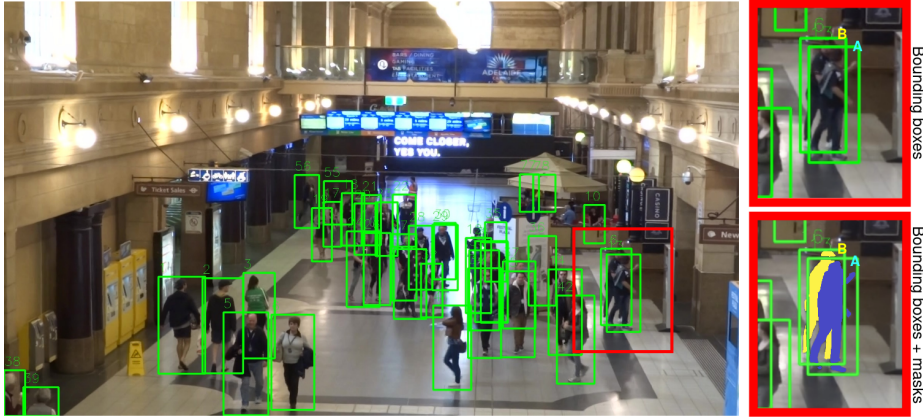
**Keywords:** multiple object tracking · segmentation

## 1 Introduction

Multiple Object Tracking (MOT) consists of assigning a unique identity to every object and preserving it over time [4]. This problem is commonly solved with Deep Learning techniques such as Convolutional Neural Networks (CNNs) [21] or transformers [23]. The most usual paradigm for solving MOT is Tracking by Detection (TbD). These approaches are based on a detector and an association mechanism that match detections with previously tracked objects.

The tracking process is stopped when no detection can be associated with the corresponding track. This can happen when the object disappears from the scene or when the detector fails to identify it. The latter situation is one of the biggest challenges that MOT faces nowadays [22] as the track could have been maintained alive given an appropriate detection. In this work, we want to solve this problem and preserve those tracks.

Relying on bounding boxes to recover missing tracks is infeasible in crowded datasets, where there are overlapping situations. Figure 1 shows an example of a crowded scene, detailing a situation in which two people overlap. We can see how object B is partially occluded by object A. Bounding boxes contain information from both objects; therefore, an identity switch may be produced, resulting in the



**Fig. 1.** Exemplar frame from a crowded dataset in MOT20 [4] and detailed occlusion situation. Bounding boxes contain mixed information. Masks are more accurate.

loss of one of the objects from the tracking process. A more precise alternative is to work with segmentation masks that better determine the boundaries of each object, having more specific visual properties.

In order to overcome detection failure situations, we propose an architecture based on the tracking of segmentation masks. When the detector fails and, consequently, a track is not associated, it is sent to a novel Lost Tracks Recovery (LTR) architecture. LTR includes a combination of an off-the-shelf mask generation network that provides several masks guided by a bottom approach and a transformer to ensure that a mask corresponds to a certain track. The correct mask is subsequently tracked using a segmentation-based tracker.

Our main contributions can be summarized in the following.

- We propose a new architecture, LTR, to overcome detector failures in MOT. It is based on segmentation masks for identifying and tracking each of the objects, and additional modules for deciding when to stop tracking process.
- We introduce a mask selection network based on a transformer [23] to decide if a mask corresponds to a certain track. This transformer has input features extracted from the segmentation masks and bounding boxes.
- We test the LTR model integrated with a state-of-the-art MOT system and demonstrate how results are improved in a complex dataset such as MOT20 [4], reducing the number of false negatives due to missed detections.

## 2 Related work

### 2.1 Multiple Object Tracking

During the years, different strategies have been followed to solve MOT problem. Tracking by Detection (TbD) is one of the most common. It is divided into two

subtasks: object detection and association. The detector identifies all the objects present in each frame and, with data association, identities are preserved through time.

We can find different approaches for solving MOT with the TbD paradigm. There are methods based on transformer architectures such as [20], which takes the features of the previous frame as the query for the predictions in the current frame. Center-based methods attempt to predict object center heat-maps to generate the bounding box [26] [33]. SORT-based methods boost original SORT [1] tracker performance with stronger detection and association steps [3] [6].

ByteTrack is another example of a TbD solution, taking advantage of the YOLOX detector [7], along with a two-stage association method using the Kalman filter [10] and an IoU-based matching with the Hungarian Algorithm [12]. However, a tracker like ByteTrack still fails to track objects when detection misses appear. Taking this into account, we propose a new module that overcomes these failures using a strategy based on segmentation masks.

## 2.2 Segmentation and segmentation-based tracking

The emergence of Segment Anything (SAM) [11] is an important step forward in the object segmentation paradigm. This model is able to segment any object—due to its class-agnostic nature—by providing different types of prompts: bounding boxes, points, text, or even another mask. SAM counts with an image and a prompt encoder in charge of embedding extraction, as well as a mask decoder that generates the mask. Three masks are generated per prompt, since a mask can represent the whole body or a part or subpart of it.

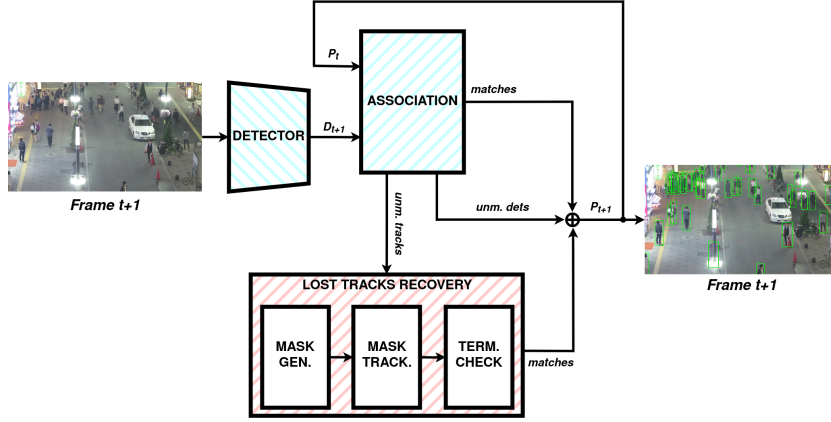
There are also architectures that perform single-object tracking using segmentation masks. AOT —Associating Objects with Transformers— is one of the top performer methods [27]. It is based on an ID assignment mechanism for joint association and decoding of multiple targets, and a Long Short-Term Transformer (LSTT) framework that performs hierarchical multi-object matching and propagation. DeAOT [28] is an evolution that further enhances AOT performance, decoupling hierarchical propagation in two parallel branches and including a more efficient module —GPM: Gated Propagation Module— for mask propagation.

In our LTR, we have taken into account both SAM and AOT networks for fixing detection failures. SAM is used for generating masks for the tracks, and for feature extraction in the mask selection network. This network is in charge of ensuring that a mask corresponds to a track. AOT is the proposal chosen to track said masks. More details will be given in Section 3.

## 3 Methodology

### 3.1 TbD architecture

Figure 2 shows an overview of our TbD architecture. The detector —*DETECTOR* (Fig. 2)— is responsible for identifying all objects present throughout each video



**Fig. 2.** TbD architecture, adding our LTR component for overcoming detector failures.

frame, and the association mechanism —*ASSOCIATION* (Fig. 2)— matches those detections with the previous frame predictions — $D_{t+1}$  and  $P_t$ , respectively (Fig. 2). The association is performed using the well-known Hungarian Algorithm [12], with a cost matrix based on the Intersection over Union (IoU) between the current frame detection bounding boxes and previous frame predictions. The matched detection-track pairs —*matches* (Fig. 2)— are propagated to the current frame, and the unmatched detections —*unm. det.* (Fig. 2)— are initialized as new tracks. Both are concatenated and form the predictions of the current frame — $P_{t+1}$  (Fig. 2).

Unmatched tracks after association —*unm. tracks* (Fig. 2)— would normally be marked as lost. However, there are false negatives that appear due to detector failures and could be kept on track. In order to do so, we design a new architecture (LTR) that deals with those tracks —*LOST TRACKS RECOVERY* (Fig. 2). It is based on the generation and tracking of segmentation masks (Sect. 3.2).

### 3.2 Lost Tracks Recovery

**Mask generation** Every time a new object is lost, a mask needs to be generated. For that, the last available detection —from previous frame, namely  $D_t$ — is taken. The mask generation process is detailed in Figure 3 —this corresponds to block *MASK GEN.* in Figure 2. The bounding box is used to call the SAM [11] method for getting the mask —*SEGMENTATION NETWORK* (Fig. 3). SAM has a prompt encoder and an image encoder that process input prompts and images, respectively. The fusion of the prompt and the image encoding is sent to the image decoder in charge of predicting the masks. Three masks can be extracted from each prompt to solve ambiguity situations.

If the bounding box of an occluded object is used directly as a prompt, the mask can mix information from the objects inside that bounding box, or even segment only the object in front. To avoid that, we propose to generate a set

of segmentation masks using a point grid inside the original bounding box. If a point is located above the object of interest, the mask is more likely to cover only that object. In this way, with an  $N \times N$  grid,  $N \times N \times 3$  masks are generated —*N-point grid* and  $N \times N \times 3$  *candidate masks*, respectively (Fig. 3).

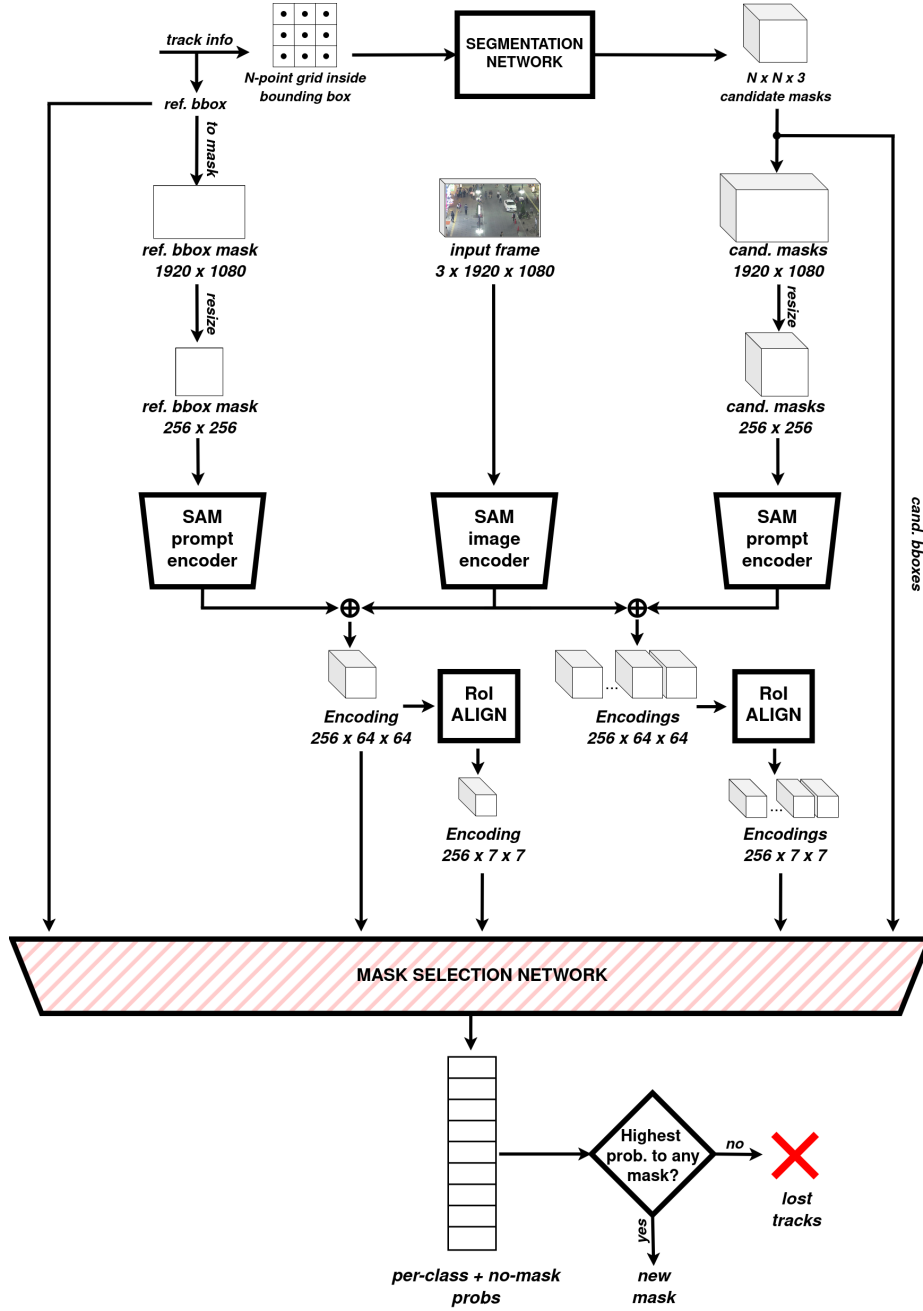
**Mask selection network details** It is necessary to check if there are one or more valid masks for a track and, in that case, choose one of them. To do so, we introduce a module named mask selection network (Fig 3). It is based on the encoder of a Vision Transformer [5] and is responsible for taking all masks and choosing the best. In Figure 3 we show how each input is processed. There are the following inputs: a set of candidate segmentation masks —*cand. masks* (Fig. 3)—, the reference bounding box of the object in the frame  $t$ , in which the mask is generated —*ref. bbox* (Fig. 3)—, and the bounding boxes surrounding each candidate mask —*cand. bboxes* (Fig. 3). Candidate boxes are provided to compute the positional encoding. The candidate masks and reference bounding box are embedded and processed as tokens.

Firstly, the reference bounding box is converted to a mask with ones inside the bounding box and zeros outside. The embeddings for each mask are extracted from SAM [11], encoding the frame information with SAM’s image encoder and each candidate mask information with SAM’s prompt encoder (Fig. 3). As said before, this embedding is normally generated in SAM as an input to the decoder which generates the prediction, so we will also consider it for our transformer. As the size of these embeddings is too large, we apply a reduction using the ROI Align operator —*RoI ALIGN* (Fig. 3)—, getting input tokens of size  $256 \times 7 \times 7$ .

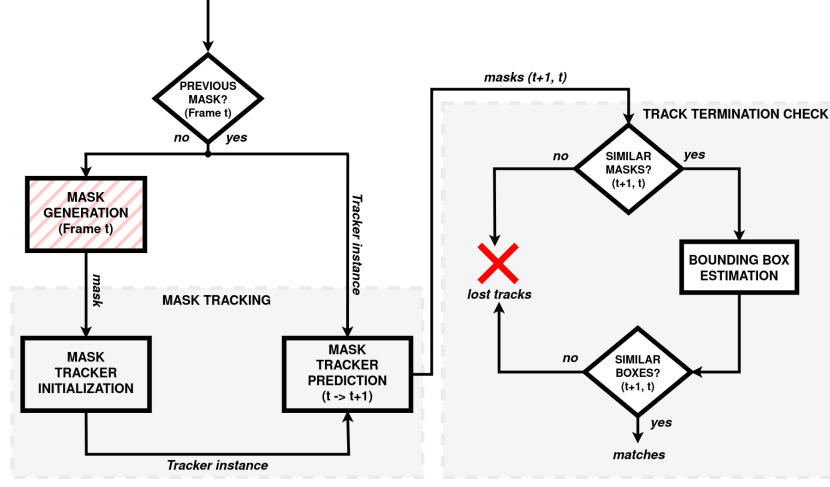
Those tokens are then resized to the internal dimension of the transformer via a linear projection, and the positional encoding is computed using the bounding boxes and added afterwards to each token. Then, encoder layers are applied, ending with an MLP layer with the output size of the number of candidate masks + 1. Softmax activation function is calculated at the end. Probabilities per each mask and an extra *No – Mask* token representing the situation when no mask is suitable are given —*per-class + no-mask probs* (Fig. 3). The track will be marked as lost if the highest probability is assigned to this extra token.

**Mask tracking** The complete LTR pipeline can be seen in Figure 4. The mask generation process mentioned above is the first step —*MASK GENERATION* (Fig. 4). After that, the tracking process begins —*MASK TRACKING* (Fig. 4). Firstly, the mask is used to initialize a new instance of the segmentation tracker —*MASK TRACKER INITIALIZATION* (Fig. 4). We have chosen AOT [27] [28] as the tracker, which takes  $t$  as the reference frame and sets the mask there. AOT will make the prediction by propagating the mask from frame  $t$  to  $t + 1$  —*MASK TRACKER PREDICTION* ( $t \rightarrow t+1$ ) (Fig. 4)— with its LSTT [27].

An object could already have been tracked in previous frames thanks to LTR. In that case, it would have a mask already generated for the previous frame  $t$ , and an instance of the tracker already initialized —*Tracker instance* (Fig. 4). Therefore, only a new propagation will be done.



**Fig. 3.** Details of the mask generation process using the segmentation and mask selection networks.



**Fig. 4.** Details of the process for managing detector failures. Tracks that have not been matched after association will follow this pipeline.

**Termination check** After the prediction is completed, it is checked that the mask is correct —*TRACK TERMINATION CHECK* (Fig. 4). To do that, the masks of frames  $t$  and  $t + 1$  are compared to check if they are similar enough, that is, if their overlap —measured using the IoU— is over a certain threshold  $T$  —*SIMILAR MASKS?* ( $t+1, t$ ) (Fig. 4).

A similar comparison is made between the bounding box of  $t$  and the new estimate for frame  $t + 1$  using the same criteria —*SIMILAR BOXES?* ( $t+1, t$ ) (Fig. 4). For that, a bounding box is estimated for the track in  $t + 1$  —*BOUNDING BOX ESTIMATION* (Fig. 4)— as there is no detection to match in this case. The new bounding box is computed taking into account the position in the frame  $t$  and both segmentation masks. The center of the bounding box is displaced from the one in  $t$  following the displacement of the mask centers between  $t$  and  $t + 1$ .

If a track does not pass any of the similarity filters, it will be marked as lost and not maintained in  $t + 1$  —*lost tracks* (Fig. 4). In other cases, the estimated bounding box for  $t + 1$  will be assigned to the track —*matches* (Fig. 4).

## 4 Implementation details

The detector of the TbD approach has been implemented with YOLOX-X and we take the association method and related parameters from ByteTrack [30]. However, our approach could be integrated into any TbD solution. As the segmentation method, we have used SAM with the largest backbone size. For tracking masks, we have used AOT. In particular, we have chosen the DeAOT-L model with the SwinB transformer backbone.

We have selected a grid size for mask generation of  $N = 3$ , getting a total of 27 masks per object as candidates for the selection network. The Mask Selection Network has the sizes of a ViT-Base model: 12 layers, 12 heads, 3072 as the output size of feed forward layers inside each encoder block, and an internal dimension of 768. We train the network for 100 epochs with a batch size of 256, cross-entropy loss function, and AdamW optimizer. We use a learning rate scheduler with the step policy, with a maximum learning rate of  $10^{-5}$  and a minimum learning rate of  $10^{-6}$ . The threshold  $T$  for controlling the similarity of the masks and the bounding boxes between frames—in track termination check—is set to 0.9. All the experimentation was performed on a single NVIDIA A100 GPU.

## 5 Results

We report the results of our proposal on MOT20 [4], a dataset of the MOTChallenge benchmark. We focus on this dataset as it contains very crowded scenes, with objects constantly overlapping.

We report the results on commonly used MOT metrics like MOTA—Multiple Object Tracking Accuracy—, IDF1 score—Identity F1 score, which is more related to how long each object is detected correctly through the video—, HOTA [15]—Higher Order Tracking Accuracy, a metric that measures how well the trackers detect, associate, and locate each of the objects— and the number of identity switches (IDSW).

Table 1 compares the state-of-the-art with our approach. We can see how ByteTrack improves the previous state-of-the-art in MOTA, staying behind in other metrics such as HOTA or IDF1 score. We are reporting their results without including their offline interpolation and per-sequence thresholds. The addition of our LTR solution in a TbD architecture like ByteTrack has proven to be helpful: we improve by 0.3 in MOTA and by 0.9 in HOTA. Moreover, the greatest improvement is achieved in the IDF1 score, which improves by more than 1 point.

In Table 2 we provide a more precise analysis in terms of the number of false positives and false negatives. Since we aim at tracks that would normally be marked as lost, this has an impact on the number of false negatives, which is reduced by more than 3,000. At the same time, some tracks are incorrectly recovered, increasing the number of false positives. Nevertheless, the reduction in false negatives is more significant, resulting in the observed improvement in the previously mentioned metrics.

Finally, we can see in Figure 5 an example of how LTR recovers an object missed by the detector. Taking ByteTrack as the baseline tracker, we show how it is unable to track object ID 194, as no detection is matched—its bounding box is marked in red. Conversely, we generate a mask in frame  $t$  and make the prediction for  $t + 1$  thanks to LTR. Overcoming these kind of situations helped us to boost the results of the tracker.

**Table 1.** Results of our proposal and comparison with another state of the art trackers in MOT20 test set. ByteTrack and OC-SORT results are reported avoiding their interpolation and per-video tuning tricks. We report MOTA, HOTA, IDF1 and Identity Switches.

Tracker	MOTA	HOTA	IDF1	IDSW
MTrack [29]	63.5	—	69.2	6,031
MeMOT [2]	63.7	54.1	66.1	1,938
GSDT [25]	67.1	53.6	67.5	3,230
Decode-MOT [13]	67.2	54.5	69.0	2,805
OUTrack [14]	68.6	54.7	65.7	1,532
FairMOT [31]	61.8	54.6	67.3	5,243
TrackFormer [16]	68.6	54.7	65.7	1,532
TransTrack [20]	64.5	—	59.2	3,565
AOH [9]	67.9	55.1	70.0	2,698
CrowdTrack [19]	70.7	55.0	68.2	3,198
OC-SORT [3]	73.1	60.5	74.4	1,307
SGT [8]	72.8	56.9	70.5	2,649
CorrTracker [24]	65.2	—	69.1	5,183
MTracker [32]	66.3	—	66.7	2,715
MO3TR-YOLOX [34]	72.3	57.3	69.0	2,200
CountingMOT [17]	70.2	57.0	72.4	2,795
CenterTrack [33]	45.8	31.8	36.6	6,296
TransCenter [26]	72.9	50.2	57.7	2,625
GHOST [18]	73.7	61.2	75.2	1,264
StrongSORT [6]	72.2	61.5	75.9	1,066
ByteTrack [30]	74.0	59.2	72.6	1,069
ByteTrack + LTR	74.3	60.1	73.9	1,069
	<b>+0.3</b>	<b>+0.9</b>	<b>+1.3</b>	—

**Table 2.** Results of our proposal and comparison with ByteTrack in MOT20 test set. ByteTrack results are reported avoiding their interpolation and per-video tuning tricks. We report the number of false positives and false negatives.

Tracker	FP	FN
ByteTrack [30]	16,749	116,927
ByteTrack + LTR	18,890	113,617
	<b>+2,141</b>	<b>-3,310</b>

## 6 Conclusions and future work

In this work, we have proposed a new TbD architecture that maintains active tracks that would normally be marked as lost due to detection failures. Our solution is based on segmentation masks, which are more precise than methods based on bounding boxes. We used SAM to obtain multiple candidate segmentation masks for each of the objects, a new transformer to select one of the candidate masks, and AOT to track masks through time.



**Fig. 5.** Example of our architecture overcoming a miss from ByteTrack.

Our module, in integration with a state-of-the-art tracker, is able to increase its performance in a crowded dataset such as MOT20. Metrics like MOTA, HOTA and IDF1 are improved, as well as the false negatives. In future work, we plan to extend the experimentation taking into account more TbD solutions — our architecture can be integrated into any TbD approach — and more datasets.

## Acknowledgements

This research was partially funded by the Spanish Ministerio de Ciencia e Innovación (grant numbers PID2020-112623GB-I00, PID2021-128009OB-C32 and PID2023-149549NB-I00), and the Galician Consellería de Cultura, Educación e Universidade (grant numbers ED431C2018/29 and ED431G2019/04). These grants are co-funded by the European Regional Development Fund (ERDF). Manuel Bendaña is supported by the Spanish Ministerio de Universidades under the FPU national plan (grant number FPU22/01828).

## References

1. Bewley, A., Ge, Z., Ott, L., Ramos, F.T., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016. pp. 3464–3468 (2016)
2. Cai, J., Xu, M., Li, W., Xiong, Y., Xia, W., Tu, Z., Soatto, S.: Memot: Multi-object tracking with memory. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. pp. 8080–8090 (2022)
3. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric SORT: rethinking SORT for robust multi-object tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023. pp. 9686–9696 (2023)
4. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I.D., Roth, S., Schindler, K., Leal-Taixé, L.: MOT20: A benchmark for multi object tracking in crowded scenes. CoRR **abs/2003.09003** (2020)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby,

- N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021 (2021)
6. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. *IEEE Trans. Multim.* **25**, 8725–8737 (2023)
  7. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: exceeding YOLO series in 2021. *CoRR* **abs/2107.08430** (2021)
  8. Hyun, J., Kang, M., Wee, D., Yeung, D.: Detection recovery in online multi-object tracking with sparse graph tracker. In: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*. pp. 4839–4848 (2023)
  9. Jiang, M., Zhou, C., Kong, J.: AOH: online multiple object tracking with adaptive occlusion handling. *IEEE Signal Process. Lett.* **29**, 1644–1648 (2022)
  10. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* **82**(1), 35–45 (03 1960)
  11. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B.: Segment anything. In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. pp. 3992–4003 (2023)
  12. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**, 83–97 (1955)
  13. Lee, S., Park, D., Bae, S.H.: Decode-mot: How can we hurdle frames to go beyond tracking-by-detection? *IEEE Trans. Image Process.* **32**, 4378–4392 (2023)
  14. Liu, Q., Chen, D., Chu, Q., Yuan, L., Liu, B., Zhang, L., Yu, N.: Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *Neurocomputing* **483**, 333–347 (2022)
  15. Luiten, J., Osep, A., Dendorfer, P., Torr, P.H.S., Geiger, A., Leal-Taixé, L., Leibe, B.: HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.* **129**(2), 548–578 (2021)
  16. Meinhardt, T., Kirillov, A., Leal-Taixé, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. pp. 8834–8844 (2022)
  17. Ren, W., Chen, B., Shi, Y., Jiang, W., Liu, H.: Countingmot: Joint counting, detection and re-identification for multiple object tracking. *CoRR* **abs/2212.05861** (2022)
  18. Seidenschwarz, J., Brasó, G., Serrano, V.C., Elezi, I., Leal-Taixé, L.: Simple cues lead to a strong multi-object tracker. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. pp. 13813–13823 (2023)
  19. Stadler, D., Beyerer, J.: On the performance of crowd-specific detectors in multi-pedestrian tracking. In: *17th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2021, Washington, DC, USA, November 16-19, 2021*. pp. 1–12 (2021)
  20. Sun, P., Jiang, Y., Zhang, R., Xie, E., Cao, J., Hu, X., Kong, T., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple-object tracking with transformer. *CoRR* **abs/2012.15460** (2020)
  21. Vaquero, L., Mucientes, M., Brea, V.M.: Siammt: Real-time arbitrary multi-object tracking. In: *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*. pp. 707–714. *IEEE* (2020)

22. Vaquero, L., Xu, Y., Alameda-Pineda, X., Brea, V.M., Mucientes, M.: Lost and found: Overcoming detector failures in online multi-object tracking. In: *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXIII. Lecture Notes in Computer Science*, vol. 15131, pp. 448–466. Springer (2024)
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. pp. 5998–6008 (2017)
24. Wang, Q., Zheng, Y., Pan, P., Xu, Y.: Multiple object tracking with correlation learning. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. pp. 3876–3886 (2021)
25. Wang, Y., Kitani, K., Weng, X.: Joint object detection and multi-object tracking with graph neural networks. In: *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*. pp. 13708–13715 (2021)
26. Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: Transcenter: Transformers with dense representations for multiple-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(6), 7820–7835 (2023)
27. Yang, Z., Wei, Y., Yang, Y.: Associating Objects with Transformers for Video Object Segmentation. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems* (2021)
28. Yang, Z., Yang, Y.: Decoupling features in hierarchical propagation for video object segmentation. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022* (2022)
29. Yu, E., Li, Z., Han, S.: Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. pp. 8824–8833 (2022)
30. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII. Lecture Notes in Computer Science*, vol. 13682, pp. 1–21 (2022)
31. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **129**(11), 3069–3087 (2021)
32. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Robust multi-object tracking by marginal inference. In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII. Lecture Notes in Computer Science*, vol. 13682, pp. 22–40 (2022)
33. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV. Lecture Notes in Computer Science*, vol. 12349, pp. 474–490 (2020)
34. Zhu, T., Hiller, M., Ehsanpour, M., Ma, R., Drummond, T., Reid, I.D., Rezatofighi, H.: Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(11), 12783–12797 (2023)