ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747572

MODELING HUMAN MEMORY IN MULTI-OBJECT TRACKING WITH TRANSFORMERS

Yizhuo Li and Cewu Lu

Shanghai Jiao Tong University

ABSTRACT

When tracking objects, humans rely on a memory mechanism, memorize the track of an object then look for it in the current scene. In this paper, we propose Memory-based Multi-object Tracking with Transformers (MMTT) to mimic human behavior in multi-object tracking. Unlike Re-ID-based methods, MMTT solves multi-object tracking in an explicit way, with a Track Encoder to extract track memory, a Detection Encoder to extract detection interactions, and a Memory Decoder to simulate the "look" process. The design of MMTT has the ability to model both spatial and temporal information of a single track. We evaluate on commonly used MOT datasets and the experimental results demonstrate its superior effectiveness. We hope this paper can provide a novel direction for the MOT task. The code and models will be made publicly available upon acceptance.

Index Terms— Multi-object tracking, Transformer, Human memory modeling, Deep learning

1. INTRODUCTION

Multi-object tracking (MOT) in videos is one of the most fundamental computer vision tasks and has witnessed large advancement in recent years. Most state-of-the-art (SOTA) work adopts the tracking-by-detection method[1, 2, 3, 4, 5, 6] in which the algorithm first detects objects, then matches the objects across different frames. Despite its effectiveness and high performance, however, tracking-by-detection usually solves the association by Re-ID techniques, with separate models or jointly trained branches to extract Re-ID features, which is not consistent with the human visual system.

When tracking objects in video or real life, humans rely not on Re-ID-like methods but a memory mechanism. That is, humans do not compare an object in the current frame with all other objects in the last frame. Instead, humans memorize the track of an object, model its motion and then locate its further movement, then look for objects in the current scene as shown in Fig.1. For example, a Re-ID-based method gives similar results if the current frame is horizontally flipped or all objects are randomly located, which is completely different from the human visual system. Even if some methods use spatial information as constraints, they solely use it as regularization for better performance in an unnatural way. Therefore, mimicking the human brain, modeling MOT with memory and sequential information is a crucial issue.

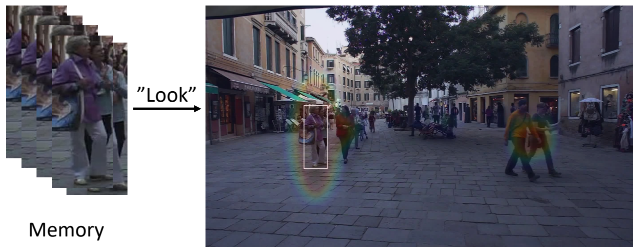

Fig. 1. Humans track object based on memory of the motion, appearance and location instead of trying to match every object in current scene.

There are a few works that build an end-to-end system for MOT, which is very close to the human vision system described above. For example, [7] builds a 3D convolution model and models tracks as tubes. However, it cannot utilize pre-trained detection models and thus requires additional training data, which is extremely expensive to collect for the MOT task. [8, 9, 10] use query-based methods rather than directly Re-ID, but they only use adjacent two frames and lack memorized temporal information. The main contribution of this paper is to build a MOT system that explicitly models human memory and can utilize pre-trained detection results at the same time.

In this paper, we propose Memory-based Multi-object Tracking with Transformers (MMTT) to explicitly model the human brain when tracking multiple objects. When tracking multiple objects, humans first keep tracks in memory as a future reference, then observe current objects, and finally look at objects to locate objects based on tracks. We model those three steps with three parts: the Track Encoder, the Detection Encoder, and the Memory Decoder. The Track Encoder models track memory, i.e., the motion, appearance, and location of previous tracks. The Detection Encoder models detection memory, i.e., the appearance, location, and interaction of current objects. The Memory Decoder plays a role as the "look" process, i.e., based on track memory, the module looks at detection memory and then predicts the status of tracks. We also propose another two techniques to further ensure the consistency with human memory mechanism. First, we

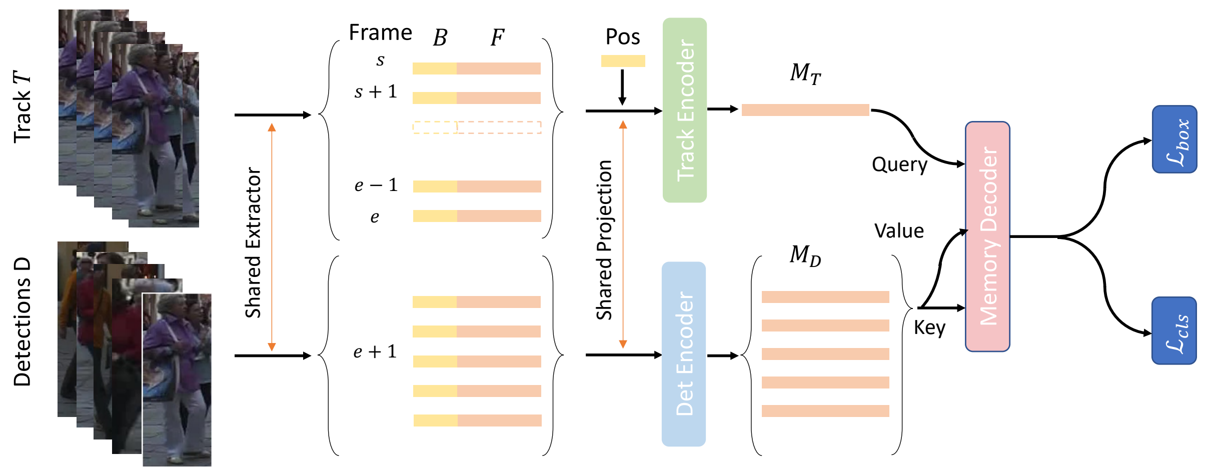

Fig. 2. The architecture of MMTT. Using the same feature extractor, the appearance feature F and bounding box B are extracted. The dashed line indicates that there can be missing part in a track. With the time embedding Pos, the Track Encoder extracts track memory M_T. The Detection Encoder extracts detection memory M_D. The Memory Decoder simulates the "look" process and makes final predictions.

design Memory Keeping, to make sure that a track will not be forgotten if not seen in the short term. Second, we propose Track Dropping, to make sure a track can be modeled even if part of the track cannot be seen.

We conduct comprehensive experiments to reveal the effectiveness of the proposed MMTT. The results reveal that MMTT is able to model the MOT process with competitive performance and is consistent with the human visual system. On popular MOT datasets MOT16 and MOT17, MMTT achieves 56.9 and 55.5 MOTA respectively. Further ablations show the characteristics of MMTT to model both spatial and temporal information and the effectiveness of designed Memory Keeping and Track Dropping. We hope this work will present a brand new perspective for the MOT community.

2. APPROACH

To formalize the MOT task, we describe a single track

$$T = \{[F_s, B_s], [F_{s+1}, B_{s+1}], \dots, [F_e, B_e]\}$$

as the composition of multiple features in previous frames, where s is the start frame of the track, which typically is bounded by model capacity, and e is the end frame of the track, which typically is the last seen frame. F_i, B_i are the visual feature and bounding box of the track in frame i respectively. It should be noted that a track is not necessarily continuous, which means some features may be lost or empty. A detection result D consists of similar features $[F, B]$. Given a set of tracks $\mathbf{T} = \{T_1, T_2, \dots, T_{N_T}\}$, and current detections $\mathbf{D} = \{D_1, D_2, \dots, D_{N_D}\}$, the objective of MOT is to predict the status of tracks in $\mathbf{T}$ and initialize new tracks based on the predictions.

As shown in Fig.2, MMTT is composed of three parts: Track Encoder, Detection Encoder, and Memory Decoder. The Track Encoder is responsible for encoding temporal information, modeling motion, appearance, and location of a single track. The Detection Encoder is responsible for encoding spatial information, modeling appearance, location, and interaction between multiple objects. The Memory Decoder is considered as a "look" module, which looks at detections based on track memory given by Track Encoder and gives the final prediction. New tracks are formed by unmatched detections. We choose the Transformer[11] architecture for all three parts for its superior ability to model sequential and variable-length data.

2.1. Track Encoder

To imitate the memory of tracks, we propose Track Encoder to extract features of tracks individually, which is a standard Transformer encoder.

A standard Transformer decoder layer consists of a multi-head attention layer and a feed-forward network. For feature $[F_i, B_i]$ of track T at timestamp i, a linear projection layer Proj first map it to input feature $\mathrm{Proj}([F_i, B_i]) + \mathrm{Pos}_i$, where Pos_i is the time embedding of timestamp i. Thus a single track T is converted to n tokens such that $T \in \mathbb{R}^{n \times c}$, where $n = e - s + 1$ is the number of frames and c is the feature dimension. In a single-head attention layer,

$$Q = TW^Q,\ K = TW^K,\ V = TW^V \tag{1}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{c \times c}$ are weight matrices. And the result of attention operation $\mathrm{attn}()$ is givn by

$$A = \mathrm{Softmax}\left(QK^\top\right) \tag{2}$$

$$\mathrm{attn}\left(T,T,T\right)=AV \tag{3}$$

In multi-head attention $\mathrm{MHA}()$, T is split to M parts, where M is the number of heads and every part is split to dimension $c'=c/M$,

$$\tilde{T}=\left[\mathrm{attn}_1(T^1,T^1,T^1);\ldots;\mathrm{attn}_M(T^M,T^M,T^M)\right] \tag{4}$$

$$\mathrm{MHA}\left(T,T,T\right)=\mathrm{LayerNorm}\left(T+\mathrm{Dropout}\left(\tilde{T}L\right)\right) \tag{5}$$

where $\mathrm{LayerNorm}$ is layer normalization, $\mathrm{Dropout}$ is dropout operation and $L\in\mathbb{R}^{c\times c}$ is a linear projection. Passing the result to a feed-forward network which is effectively a two layer linear projection with ReLU activation followed also by residual connection, $\mathrm{Dropout}$ and $\mathrm{LayerNorm}$ completes the Transformer encoder layer. Stacking multiple layers gives us the track memory M_T.

It should be noted that tracks are individually encoded, which means, unlike detections, there is no interaction when encoding tracks. This is consistent with the intuition of human behavior, where humans only focus on one object at a time even if there are multiple objects in the scene. The tracks are not necessarily complete from s to e, leading to possible masks in attn operation. This property also results in the Track Dropping technique described in Sec.2.3. However, there should be at least one valid timestamp in the track to ensure the track memory is valid. The Track Encoder encodes both temporal, spatial, and appearance information of tracks into track memory.

2.2. Detection Encoder

The Detection Encoder serves a similar purpose as the Track Encoder. Unlike Track Encoder, which only encodes one single track at a time, the Detection Encoder also aims at simulating the interactions between detections, such as occlusions and relative positions. Given current detections $\mathbf{D}=\{D_1,D_2,\ldots,D_{N_D}\}$, the i-th detection is converted to input space using Proj shared with Track Encoder. By using the same projection layer, the consistency of the input features is preserved between tracks and detection.

The detection memory M_D is also extracted by a standard Transformer encoder, which consists of multiple encoder layers, and each layer is composed of a multi-head attention $\mathrm{MHA}\left(\mathbf{D},\mathbf{D},\mathbf{D}\right)$ and a feed-forward network.

2.3. Memory Decoder

The Memory Decoder is designed to mimic the "look" process of the human brain. That is, with the memory of each track, humans look at the detections and track each object. The Memory Decoder is designed as a standard Transformer decoder. To simulate the "look" process, we use track memory M_T as query input and detection memory M_D as key and value inputs. Compared to an encoder layer, a decoder layer has an extra multi-head attention layer $\mathrm{MHA}\left(M_T,M_D',M_D'\right)$, where M_D' is the result of first multi-head self-attention. The decoder is formed by stacking multiple decoder layers.

The Transformer decoder gives a result feature R for each qurey track T. To get the final prediction, a linear predictor P_{box} predicts the bounding box $\hat{b}$ and another linear predictor P_{end} predicts the end flag $\hat{y}$. Based on [12], the bounding box loss is given by GIoU loss [13] and L1 loss

$$\mathcal{L}_{\text{box}}=\lambda_{\text{iou}}\mathcal{L}_{\text{iou}}\left(b,\hat{b}\right)+\lambda_{L1}\left\|b-\hat{b}\right\| \tag{6}$$

where $\mathcal{L}_{\text{iou}}$ is GIoU loss, λ_{iou} and λ_{L1} are coefficients for GIoU loss and L1 loss respectively, b is the ground truth bounding box for track T at current frame if the track is not ended. The classification loss is given by Binary Cross Entropy $\mathcal{L}_{\text{cls}}=y\log\hat{y}+(1-y)\log\left(1-\hat{y}\right)$, in which $y\in\{0,1\}$ is the end flag of track T in current frame. The final loss is given by

$$\mathcal{L}=\mathcal{L}_{\text{box}}+\lambda_{\text{cls}}\mathcal{L}_{\text{cls}} \tag{7}$$

where λ_{cls} is the coefficients for classification loss.

To further improve the consistency with the human visual system, we propose two techniques named Memory Keeping and Track Dropping. Despite that our design should endow the model with the ability to distinguish between "missing" (occluded or not detected) and "gone" with spatial and temporal information, we can still keep those ended tracks and see if they will appear in the future and we call this Memory Keeping. Note that methods that use Re-ID can also use Memory Keeping by maintaining a feature pool, while our model conducts it in a much more natural way due to the ability to handle variable-length input and the design of time embedding Pos. MMTT does not simply keep the features of missing tracks but also is aware of the temporal information of the missing part. And thanks to this characteristic, we propose another technique called Track Dropping, which randomly drops part of tracks for data augmentation. Track Dropping is beyond the reach of methods that do not learn temporal information or can only read fixed-length input.

3. EXPERIMENTS

3.1. Settings

Dataset We evaluate our MMTT on three popular MOT benchmarks, MOT16, MOT17 and MOT20[16, 17]. MOT16 and MOT17 consist of the same videos but different public detections including 7 training videos and 7 test videos. For the test set, MOT16 has 5919 frames, 759 tracks, and 182326 boxes in total while MOT17 has 564228 boxes in total. MOT20 is a dense dataset, including 4 training videos and 4 test videos, 4479 frames, and 765465 boxes, with a density of 170.9 per frame. MMTT requires processing all detections in a frame at a time and this requirement increases linearly

Table 1. Results of the online state-of-the-art models on MOT-16 and MOT-17 datasets using public detection. Our model does not adopt any other additional datasets such as dense detection or matching techniques such as Re-ID.

	Model	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
MOT17	MMTT(Ours)	55.5	56.6	18.2	**28.8**	**6,537**	237,498	6,825
	MTP[14]	51.5	54.9	20.5	35.5	29,623	241,618	2,563
	Tracktor [2]	53.5	52.3	19.5	36.3	12,201	248,047	2,072
	GSM Tracktor[15]	**56.4**	**57.8**	**22.2**	34.5	14,379	**230,174**	**1,485**
MOT16	MMTT(Ours)	56.9	57.5	19.0	**26.6**	**2,541**	73,743	2,260
	MTP[14]	48.3	53.5	17.0	38.7	9,799	83,712	733
	Tracktor [2]	54.5	52.5	19.0	36.9	3,280	79,149	682
	GMS Tracktor[15]	**57.0**	**58.2**	**22.0**	34.5	4,332	**73,573**	**475**

with the length of tracks. Therefore, we are not able to train MMTT on MOT20 due to the limitation of GPU memory. As a result, we use the training set 01 and 02 in MOT20 for ablations. For fair comparison, we use refined public detections and standard MOT metrics[18] for evaluation.

Hyber-Paramters We use a pre-trained Faster RCNN[19] to extract appearance feature F. Note this detector is a universal detector and not trained for Re-ID purpose. For input, we use a maximum length of 8 for each track and the probability for Track Dropping is set to 0.1. For model details, we adopt 1-layer Transformer with 4 heads for Track Encoder, Detection Encoder, and Memory Decoder. The dimension c is set to 64 and the hidden dimension in the feed-forward layer is set to 256. For optimization, we use an AdamW[20] optimizer with a batch size of 8 and an initial learning rate of 1×10^{-4}. The model is trained for 150 epochs and the learning rate is decayed to 1×10^{-3} at the 120th epoch. The loss coefficients are set to $\lambda_{\text{box}} = 2.0, \lambda_{\text{iou}} = 5.0, \lambda_{\text{cls}} = 1.0$ and the length for Memory Keeping is set to 3.

3.2. Main Results

We report main results in Tab.1 and compare with SOTA results. We only show online algorithms that are peer-reviewed and adopt the public detections. As we can see, on the MOT17 dataset, MMTT achieves results comparable to SOTA with a MOTA of 55.5. Similarly, on MOT16, our MMTT model also achieves competitive results, with a MOTA of 56.9. It is worth noting that our model has high ID switches, which is the main reason affecting the performance. This is acceptable considering we are not using any Re-ID technique, any additional data, or a pre-trained feature extractor.

3.3. Ablation Study

Due to the submitting restriction of MOT Challenge, we conduct ablation study on the 01 and 02 training sequences of MOT20, with the model trained on MOT17. We are only reporting ID switches as it is the only metric that changes significantly.

On the length of tracks We study the length of tracks in the Track Encoder and the results are shown in Fig.3. As we can see, the number of ID switches decreases along with the length of tracks increasing. Despite we are not able to afford a longer length of tracks, it can be seen that our MMTT model has the ability to encode temporal information and the ability improves with longer input. This demonstrates the effectiveness of our design using the Transformer.

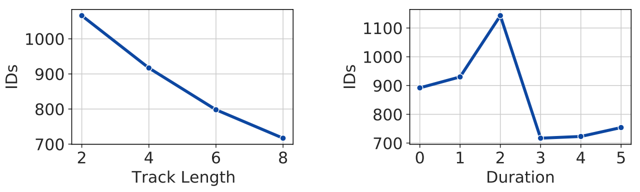

Fig. 3. Left: IDs for different length of tracks. Right: IDs for different duration of Memory Keeping.

On Memory Keeping To show the effectiveness of Memory Keeping, we test different duration shown in Fig.3. As expected, the number of ID switches decreases with a longer keeping duration. However, further increasing the duration of memory will degrade the performance. We believe this is attributed to that longer duration results in accumulation in tracks, thus making it harder for Memory Decoder. We also notice that the number of ID switches increases at first, this is because a too short duration is not enough to handle missing tracks but leads to more noise.

On Track Dropping We also check the effective design of Track Dropping. Using a Track Dropping Probability of 0.1 gives us 717 ID Switches while without Track Dropping, the number of ID Switches is increased to 916.

4. CONCLUSION

In this paper, we present Memory based Multi-object Tracking with Transformers (MMTT) to solve the challenging MOT task. MMTT is composed of Track Encoder, Detection Encoder, and Memory Decoder. By imitating human behavior, MMTT has the ability to learn both spatial and temporal features with the help of a memory-like mechanism flexibly and naturally. The experimental results on widely used MOT datasets demonstrate the effectiveness of MMTT. We hope our design can provide the community with a new perspective and motivate future research.

5. REFERENCES

[1] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan, “Poi: Multiple object tracking with high performance detection and appearance feature,” in *European Conference on Computer Vision*. Springer, 2016, pp. 36–42.

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe, “Tracking without bells and whistles,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[3] Peng Chu and Haibin Ling, “Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6172–6181.

[4] Loïc Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle, “Improving multi-frame data association with sparse representations for robust near-online multi-object tracking,” in *European Conference on Computer Vision*. Springer, 2016, pp. 774–790.

[5] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu, “Fairmot: On the fairness of detection and re-identification in multiple object tracking,” *International Journal of Computer Vision*, pp. 1–19, 2021.

[6] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang, “Towards real-time multi-object tracking,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 107–122.

[7] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu, “Tubetk: Adopting tubes to track multi-object in a one-step training model,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6308–6318.

[8] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” *arXiv preprint arXiv:2101.02702*, 2021.

[9] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo, “Transtrack: Multiple-object tracking with transformer,” *arXiv preprint arXiv:2012.15460*, 2020.

[10] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl, “Tracking objects as points,” in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, “End-to-end object detection with transformers,” in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[13] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[14] Chanho Kim, Li Fuxin, Mazen Alotaibi, and James M Rehg, “Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9553–9562.

[15] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu, “Gsm: Graph similarity model for multi-object tracking.,” in *IJCAI*, 2020, pp. 530–536.

[16] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler, “Mot16: A benchmark for multi-object tracking,” *arXiv preprint arXiv:1603.00831*, 2016.

[17] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé, “Mot20: A benchmark for multi object tracking in crowded scenes,” *arXiv preprint arXiv:2003.09003*, 2020.

[18] Keni Bernardin and Rainer Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[20] Ilya Loshchilov and Frank Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.