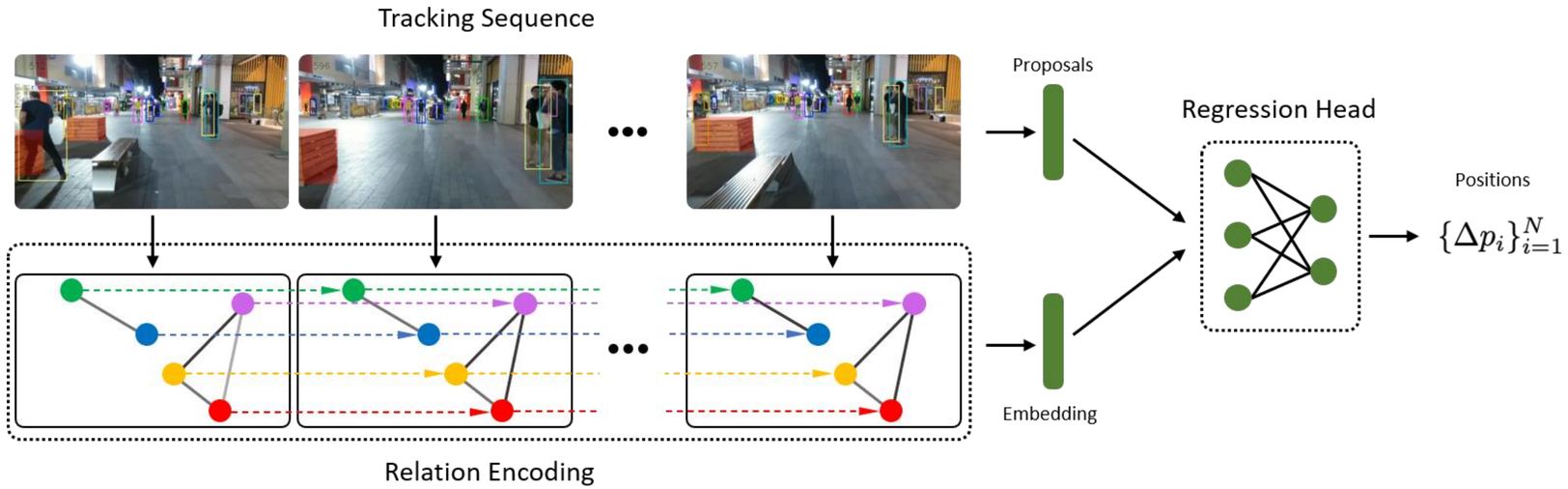
Relational Prior for Multi-Object Tracking

Artem Moskalev, Ivan Sosnovik and Arnold Smeulders
UvA-Bosch Delta Lab, University of Amsterdam, Netherlands



(Figure 1) The spatio-temporal relational graph is built on top of the tracked instances. The constructed graph is used by **Relation Encoding Module** (REM) to perform message-passing and compute relational prior online. Computed **relation prior** used to guide the regression head of a backbone tracker.

Summary

When objects are part of a group, the mutual occlusions make individual tracking harder. Rather than rejecting that information, analyzing group membership is useful as group has more uniquely identifying characteristics than just a set of individual objects. In this paper, we set out to exploit group relations for robust multi-object tracking.

Multi-object online tracking has recently made progress with tracking-by-regression methods. These methods track each object independently from one another, which makes tracking challenging in a case of dense interactions between objects in a scene (Figure 2).

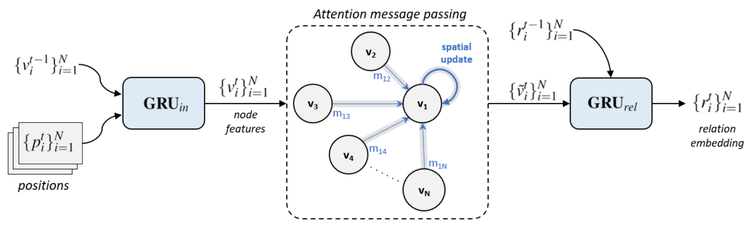
In this work, we extend current tracking-by-regression methods with online group relations. To do so, we develop relation encoding module, which produces relational prior, encoding the group structure for each object. The relation encoding module is implemented as a plug-in extension for tracking-by-regression methods.

Contributions

- We develop a method to encode inter-object relations online in dense scenes by running spatial-temporal message passing
- We demonstrate the virtue of relational prior to improve the tracking of multiple objects

Relation prior for online tracking

To encode inter-object relations, the relation encoding module takes a set of tracked instances as input and runs a message passing algorithm over the spatio-temporal relational graph.



The procedure consists of computing input features, computing graph messages, aggregating messages and updating node features.

Graph-attention:

$$\alpha_{ij}^t = \frac{\exp(\text{LeakyReLU}([\mathbf{W}_{a_1} v_i^t]^T [\mathbf{W}_{a_2} v_j^t]))}{\sum_{j \in \mathcal{N}_i} \exp(\text{LeakyReLU}([\mathbf{W}_{a_1} v_i^t]^T [\mathbf{W}_{a_2} v_j^t]))}$$

Aggregation & Update:

$$\textit{(spatial)} \quad \tilde{v}_i^t = \sigma(\mathbf{W}_u [v_i^t \| \sum_{j \in \mathcal{N}_i} \alpha_{ij}^t m_{ij}^t] + \mathbf{b}_u)$$

$$\textit{(temporal)} \quad r_i^t = \mathbf{GRU}_{\text{rel}}(\tilde{v}_i^t, r_i^{t-1})$$

Experiments

We extend tracking-by-regression model to reason about the object's position based both on appearance and relation cues (Figure 1). To do so, we condition the predicted positions of the objects on their relation priors. To that end, we concatenate the appearance features extracted from proposal regions with the relation embeddings of the corresponding objects.

	Method	HOTA ↑	IDF1 ↑	MOTA ↑	MOTP ↑	MT ↑	ML ↓
MOT17	**RelTracktor** *(Ours)*	**45.8**	**56.5**	**57.2**	**79.0**	21.9	**34.3**
	Tracktor [1]	44.8	55.1	56.3	78.8	21.1	35.3
	deepMOT [21]	42.4	53.8	53.7	77.2	19.4	36.6
MOT20	**RelTracktor** *(Ours)*	**43.4**	**53.0**	**54.1**	79.2	**36.7**	**22.6**
	Tracktor [1]	42.1	52.7	52.6	**79.9**	29.4	26.7
	SORT20 [3]	36.1	45.1	42.7	78.5	16.7	26.2

Performance comparison on MOT17 and MOT20. The relation-aware RelTracktor model outperforms the baseline model with no relations on both benchmarks.

A higher IDF1 score indicates that our model **robustly preserves the identities** of the objects throughout the sequence, while also providing more **accurate localization** as indicated by the MOTP score

(Figure 2) Top: tracking without relations, so independent trajectories are assumed. Dense bodily interaction causes tracking failure. Bottom: extending the tracker with relational prior makes it more robust to occlusions.

https://openreview.net/forum?id=1MZnMuu8mg4