GSM: Graph Similarity Model for Multi-Object Tracking

Qiankun Liu , **Qi Chu**[*] , **Bin Liu** and **Nenghai Yu**
University of Science and Technology of China, China
liuqk3@mail.ustc.edu.cn, {qchu, flowice, ynh}@ustc.edu.cn

Abstract

The popular tracking-by-detection paradigm for multi-object tracking (MOT) focuses on solving data association problem, of which a robust similarity model lies in the heart. Most previous works make effort to improve feature representation for individual object while leaving the relations among objects less explored, which may be problematic in some complex scenarios. In this paper, we focus on leveraging the relations among objects to improve robustness of the similarity model. To this end, we propose a novel graph representation that takes both the feature of individual object and the relations among objects into consideration. Besides, a graph matching module is specially designed for the proposed graph representation to alleviate the impact of unreliable relations. With the help of the graph representation and the graph matching module, the proposed graph similarity model, named GSM, is more robust to the occlusion and the targets sharing similar appearance. We conduct extensive experiments on challenging MOT benchmarks and the experimental results demonstrate the effectiveness of the proposed method.

1 Introduction

Multi-object tracking (MOT) aims at estimating the locations of multiple objects in the video sequence and maintaining their identities consistently, which has various applications such as video surveillance and autonomous driving. Benefiting from the advances of object detection [Felzenszwalb *et al.*, 2010; Ren *et al.*, 2015; Yang *et al.*, 2016], the tracking-by-detection paradigm has become popular for MOT in the past decade. Methods following this paradigm focus on associating object detections across frames, namely data association problem.

Generally, a robust similarity model is the key to the success of data association based trackers. Most existing methods build similarity model only based on the feature of the individual object while ignoring the relations among objects. Despite the progress, using the feature of the individual object alone would be problematic in some complex scenarios. For example, since the objects usually belong to the same category (*e.g.* pedestrians or vehicles) in MOT scenarios, different objects are prone to share similar appearance. Using the individual feature alone is not sufficient to discriminate these objects well. Besides, objects in MOT scenarios usually suffer from frequent occlusions, especially in crowded scenes, which would be a great challenge for building a robust similarity model using only the feature of the individual object.

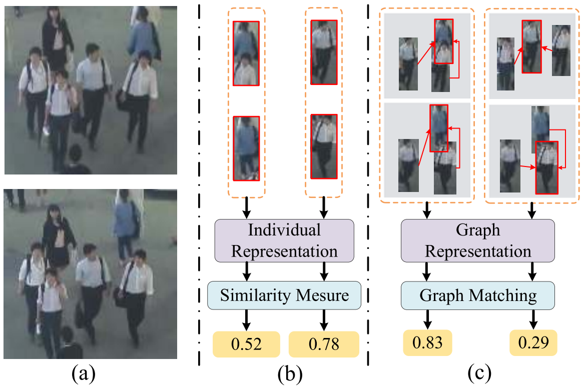


Figure 1: (a) Two adjacent frames in complex scenario. (b) The similarity scores based on individual representation are unreliable in the case of occlusion and different objects sharing similar appearance. (c) With the help of (two) neighbors, a graph can be built for each object. The similarity scores obtained by the proposed graph representations and matching are much more reliable.

In contrast to the individual feature, the relations among objects indicate the topological structure of multiple objects, which can be utilized as an important cue for facilitating the robustness of the similarity model. As shown in Figure 1, in the case of occlusion and different objects sharing similar appearance, similarity scores based on individual feature representation are unreliable. While the consistent relations among objects across frames can help to improve the reliability of the similarity scores, since the simultaneous consideration of multiple objects and their topological structure is more robust to the variants of individual objects.

In this paper, we propose a novel graph representation that

[*] Corresponding Author

utilizes the merits of both the individual feature and the relations among objects to improve the robustness of the similarity model. Specifically, for each object, regarded as a anchor, we build a directed graph of which the vertexes represent appearance feature of the anchor object and its neighbors respectively. Each vertex has a edge directed to the anchor object, which represents the relative position feature. Thus, the individual information and relations among objects are encoded by the vertexes and edges respectively.

The graph representation may introduce some unreliable neighbors and corresponding relations due to imperfect detections such as false alarms and missing detections, which would cause the problem of misalignment when calculating the similarity score of two graph representations. To handle this problem, we design a graph matching module. Specifically, we first align the two graph representations by solving the linear assignment problem among neighbors of two anchor objects and then use the aligned graph representations to calculate the similarity score. Thus, the impact of unreliable detections could be effectively suppressed.

To sum up, the contributions of this work are as follows:

First, we propose a novel graph representation that integrates the individual feature and relations among objects to improve the robustness of the similarity model.

Second, we design a graph matching module which can effectively alleviate the impact of unreliable detections.

Third, we apply the proposed graph similarity model (GSM) to Tracktor [Bergmann *et al.*, 2019], the current state-of-the-art online MOT tracker, and achieve the best performance on MOT benchmarks [Milan *et al.*, 2016] in most metrics including MOTA and IDF1.

2 Related Work

2.1 Tracking-by-Detection Paradigm

Thanks to the advances of object detectors [Felzenszwalb *et al.*, 2010; Ren *et al.*, 2015; Yang *et al.*, 2016], numerous methods based on tracking-by-detection paradigm have been developed for MOT. These methods focus on associating object detections provided by a pre-defined detector across frames, namely the data association problem. Generally, the existing works can be categorized into online methods [Sadeghian *et al.*, 2017; Zhu *et al.*, 2018; Xu *et al.*, 2019; Bergmann *et al.*, 2019] and offline methods [Li *et al.*, 2009; Maksai and Fua, 2019]. Online methods process video sequences frame-by-frame and generate trajectories only using information up to the current frame, which are suitable for causal applications. While offline methods process video sequences in a batch way and can utilize the whole video information including the future frames to better handle the data association problem.

The proposed method is also based on the tracking-by-detection paradigm and focuses on improving the robustness of the similarity model for more accurate association. Although the proposed method is applicable to both online and offline trackers, we only use the simpler online setting for better comparison in this paper.

2.2 Similarity Model

A robust similarity model is crucial for data association based MOT trackers. Most existing works utilize the feature of individual object including appearance and motion to measure similarity.

Appearance is an important cue to discriminate different objects, which is widely used in MOT. Many early works extracted hand-craft appearance feature such as raw pixel template [Yamaguchi *et al.*, 2011; Pellegrini *et al.*, 2009], color histogram [Izadinia *et al.*, 2012; Xiang *et al.*, 2015], HOG [Izadinia *et al.*, 2012; Kuo *et al.*, 2010] and so on. Recently, deep networks have been adopted to MOT for modeling appearance of objects [Zhu *et al.*, 2018]. In this work, we also utilize deep networks to extract appearance feature. Different from previous methods that directly use appearance feature to compute the similarity score, appearance feature is used together with the feature of relations among objects to build a graph representation for each object in our work.

Motion is also a commonly used cue in MOT. Most methods assumed that the objects move smoothly in the image space and designed different motion models to capture the dynamic behaviour of individual objects such as linear motion model [Milan *et al.*, 2013; Breitenstein *et al.*, 2009] and nonlinear motion model [Yang and Nevatia, 2012]. However, the movement of object is not always smooth and thus may not be predictable, especially when the camera moves. In this paper, instead of modeling the movements of individual objects, we utilize the relative positions among objects which are more robust to the camera motion.

Besides these individual features, relations among objects are also helpful for measuring similarity. However, due to its complexity, only a few works successfully encoded the relations among objects. The pioneering works [Helbing and Molnar, 1995; Pellegrini *et al.*, 2009; Yamaguchi *et al.*, 2011] modeled a few interactions among objects such as collision avoidance or group attraction with the hand-designed patterns. Recently, [Sadeghian *et al.*, 2017] modeled the relations among objects using the occupancy map where the neighborhood of object are treated as a fixed size occupancy grid. However, the representation of the occupancy grids only models the rough distribution of objects' location, without differentiating individual objects. [Xu *et al.*, 2019] utilized relation network to encode the relations information, which makes the individual object differentiable. Specifically, the relations information was encoded as the attention weights for aggregating appearance information from other objects to strengthen the input appearance feature. However, representing the relations information in such a implicit way can not take full use of the topological structure among objects. Besides, both the representation of occupancy grids and the attention weights are sensitive to the unreliable relations due to imperfect detections such as false alarms and missing detections. Different from these methods, we integrate the appearance of individual objects and the relations among objects into a unified graph representation which can make them both differentiable. What's more, we also design a graph matching module to alleviate the impact of unreliable relations.

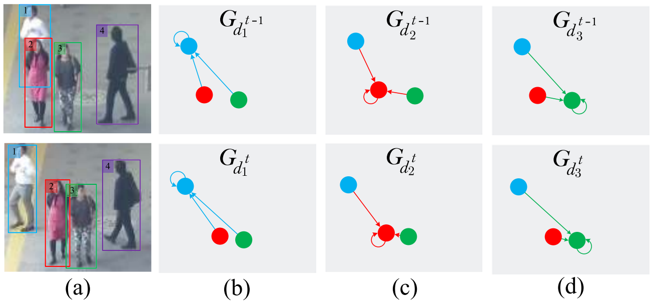


Figure 2: Top row: frame $t-1$. Bottom row: frame t. The numbers located at the top-left corner of bounding-boxes denote the indices of detections. (a) Detected objects and the bounding-boxes. (b) Graphes for d_1^{t-1} and d_1^t. (c) Graphes for d_2^{t-1} and d_2^t. (d) Graphes for d_3^{t-1} and d_3^t. Each graph consists of $K+1$ vertexes and $K+1$ directed edges ($K=2$ in this figure). The circle nodes (vertexes) denote the appearance features extracted from image patches, and the directed edges denote the relative position features. Note that although the vertexes in $G_{d_1^t}, G_{d_2^t}, G_{d_3^t}$ are the same, the edges are different.

3 Method

In this paper, we focus on improving the robustness of the similarity model. To this end, we propose a Graph Similarity Model (GSM) that can be applied to any data association based MOT method. To better understand the proposed GSM, we first introduce the general data association procedure.

3.1 Data Association

Let $D^t = \{d_i^t\}_{i=1}^{I_t}$ denotes the set of detections in the t-th frame, where I_t is the number of detections. Each detection d_i^t is denoted as $d_i^t = (b_i^t, p_i^t)$, where p_i^t is the image patch cropped from frame t, and $b_i^t = (x_i^t, y_i^t, w_i^t, h_i^t)$ is the bounding-box represented by the center coordinate, width and height.

The data association procedure between frame $t-1$ and t can be solved by some linear assignment algorithms, such as Hungarian algorithm, while providing a cost matrix $M \in \mathbb{R}^{I_{t-1} \times I_t}$. The element $m_{i,j}$ in M:

$$m_{i,j} = \mathcal{C}(d_i^{t-1}, d_j^t), \tag{1}$$

is the cost between d_i^{t-1} and d_j^t, where $\mathcal{C}(\cdot,\cdot)$ is the function to compute the cost based on individual representations of d_i^{t-1} and d_j^t.

Most existing methods focus on the learning of representation of object detections or the design of $\mathcal{C}(\cdot,\cdot)$ without taking the relations among objects into consideration, which results in an un-robust cost $m_{i,j}$, as shown in Figure 2 (a), detections d_1^{t-1} (heavily occluded) and d_1^t are the same object, but it's hard to get a proper cost between them based on Eq. (1).

To take advantage of the relations among objects, we propose:

$$m_{i,j} = \mathcal{C}_G(G_{d_i^{t-1}}, G_{d_j^t}), \tag{2}$$

where $G_{d_i^t}$ is the directed graph constructed for d_i^t, $\mathcal{C}_G(\cdot,\cdot)$ is the graph matching function to get the cost based on two graphs. The relations among objects (topological structure of objects) are embedded into the directed edges of graph.

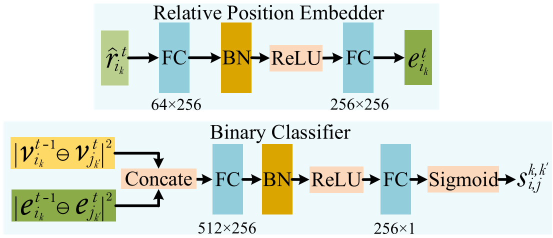


Figure 3: Relative Position Embedder and Binary Classifier.

3.2 Graph Similarity Model

In order to build a graph for detection d_i^t, we first get its top K neighbors intra the same frame t. In this paper, we use the Euclidean distance between the center coordinates of bounding-boxes as the default measurement to get the neighbors. Let $N_{d_i^t} = \{d_i^t\} \cup \{d_{i_k}^t\}_{k=1}^K$ be the ordered set of anchor d_i^t and its K neighbors, where $d_{i_k}^t$ is the k-th neighbor of d_i^t. To simplify the notation, we define d_i^t as its 0-th neighbor, i.e. $N_{d_i^t} = \{d_{i_k}^t\}_{k=0}^K$. Taking the detections in Figure 2 (a) for example, when $K=2$, $N_{d_1^t} = \{d_{1_0}^t, d_{1_1}^t, d_{1_2}^t\} = \{d_1^t, d_2^t, d_3^t\}$ and $N_{d_2^t} = \{d_{2_0}^t, d_{2_1}^t, d_{2_2}^t\} = \{d_2^t, d_3^t, d_1^t\}$.

Graph Representation

The directed graph $G_{d_i^t} = (V_{d_i^t}, E_{d_i^t})$, which is constructed upon $N_{d_i^t}$, consists of $K+1$ vertexes and $K+1$ directed edges (see Figure 2 (b), (c) and (d)).

The set of vertexes $V_{d_i^t}$ are defined as:

$$V_{d_i^t} = \{v_{i_k}^t\}_{k=0}^{k=K}, \quad v_{i_k}^t = f_{CNN}(p_{i_k}^t), \tag{3}$$

where $f_{CNN}(\cdot)$ denotes the forward function of a Convolutional Neural Network (CNN) and $v_{i_k}^t$ is the appearance feature vector extracted from the image patch $p_{i_k}^t$ of detection $d_{i_k}^t$.

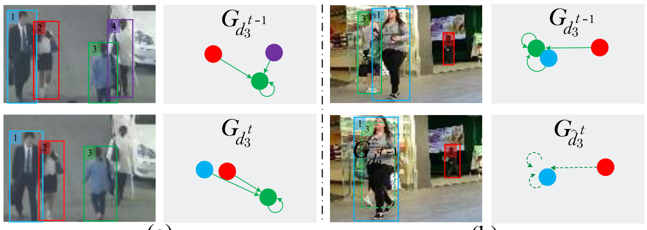
In order to embed the relation between d_i^t and one of its neighbors $d_{i_k}^t$, we define the relative position between them as:

$$\begin{aligned} r_{i_k}^t &= f_{RP}(b_i^t, b_{i_k}^t) \\ &= (\frac{x_{i_k}^t - x_i^t}{w_i^t}, \frac{y_{i_k}^t - y_i^t}{h_i^t}, \log(\frac{h_{i_k}^t}{h_i^t}), \log(\frac{w_{i_k}^t}{w_i^t}), \\ &\qquad \frac{x_{i_k}^t - x_i^t}{w^t}, \frac{y_{i_k}^t - y_i^t}{h^t}, \frac{w_{i_k}^t - w_i^t}{w^t}, \frac{h_{i_k}^t - h_i^t}{h^t}), \end{aligned} \tag{4}$$

where w^t and h^t are the width and height of frame t respectively, and $f_{RP}(\cdot,\cdot)$ denotes the function to get relation position $r_{i_k}^t$ when given two bounding-boxes. This 8-d relative position is encoded to a high-dimensional (64-d in default) representation by the method in [Vaswani *et al.*, 2017], which is denoted as $\hat{r}_{i_k}^t$. The set of directed edges $E_{d_i^t}$ are defined as:

$$E_{d_i^t} = \{e_{i_k}^t\}_{k=0}^{k=K}, \quad e_{i_k}^t = f_{RPE}(\hat{r}_{i_k}^t), \tag{5}$$

where $f_{RPE}(\cdot)$ denotes the forward function of Relative Position Embedder (RPE), as shown in Figure 3.



(a) (b)

Figure 4: Top row: frame $t-1$. Bottom row frame t. The numbers located at the top-left corner of bounding-boxes denote the indices of detections. (a) Graph $G_{d_3^{t-1}}$ and $G_{d_3^t}$ are different due to the missed detection in frame t. (b) It's hard for a detector to detect the object within the dashed green box, which leads to the missing of this object in frame t while tracking online. However, the position of this object can be estimated with the help of its neighbors and graph matching module.

Graph Matching Module

Given two graphs $G_{d_i^{t-1}}$ and $G_{d_j^t}$, we first get a similarity matrix $S_{i,j} \in \mathbb{R}^{(K+1)\times(K+1)}$, the element in k-th row and $k^{'}$-th column is:

$$s_{i,j}^{k,k^{'}} = BC([|v_{i_k}^{t-1} \ominus v_{j_{k^{'}}}^{t}|^2, |e_{i_k}^{t-1} \ominus e_{j_{k^{'}}}^{t}|^2]), \tag{6}$$

where $\ominus$ is element-wise substraction between two feature vectors, $|\cdot|^2$ denotes element-wise square operation, $[\cdot,\cdot]$ denotes the concatenation of two feature vectors, and $BC(\cdot)$ denotes the Binary Classifier, as shown in Figure 3.

Intuitively, the similarity between graph $G_{d_i^{t-1}}$ and $G_{d_j^t}$ can be defined as:

$$s_{i,j} = \frac{1}{K+1}\sum\nolimits_{k=0}^{K} s_{i,j}^{k,k}. \tag{7}$$

We call this matching method $hard$ graph matching. Due to the false positive and false negative in the provided detections, the similarity $s_{i,j}$ in Eq.(7) is unreliable when d_i^{t-1} and d_j^t are the same object. As shown in Figure 4 (a), both the first and second neighbors of d_3^{t-1} and d_3^t are different objects. However, the first neighbor of d_3^t and the second neighbor of d_3^{t-1} are the same object. To alleviate such misalignment between the neighbors, we propose another $soft$ graph matching method:

$$\hat{s}_{i,j} = \frac{1}{K+1}(s_{i,j}^{0,0} + f_{LA}(\hat{S}_{i,j})), \tag{8}$$

where $\hat{S}_{i,j} \in \mathbb{R}^{K\times K}$ is the matrix by removing the first row and column of $S_{i,j}$, and $f_{LA}(\cdot)$ is a modified linear assignment function, which solves the linear assignment problem and returns the maximum total similarity. Compared with hard graph matching, soft graph matching is positive to the case that two graphs are built for the same object, but be negative to the case that two graphs are built for different objects since it gets a higher similarity score for all different pair of graphs. Nevertheless, the negative effect to the later case is negligible thanks to the effective representation of graphs.

Finally, Eq.(2) can be rewritten as:

$$m_{i,j} = 1 - \hat{s}_{i,j}, \tag{9}$$

Finding Lost Objects based on GSM

In the field of MOT, an object may be occluded severely by others which is undetectable for detectors. As shown in Figure 4 (b), the object within green dashed box is lost in frame t while tracking online. However, its position can be estimated by the proposed GSM since its neighbors are detected and tracked.

Suppose d_i^{t-1} is lost in frame t. For each tracked neighbor $d_{i_k}^{t-1}$, let $d_{i_k}^t$ be the corresponding detection in frame t. A bounding-box $\bar{b}_i^{t,k}$ in frame t can be estimated for d_i^{t-1} based on $b_{i_k}^t$:

$$\bar{b}_i^{t,k} = f_{RP}^{-1}(r_{i_k}^{t-1}, b_{i_k}^t), \tag{10}$$

where $f_{RP}^{-1}(\cdot,\cdot)$ denotes the inverse function of $f_{RP}(\cdot,\cdot)$ in Eq.(4). By averaging all estimated $\bar{b}_i^{t,k}$, we can get the final estimated bounding-box $\bar{b}_i^t$ for d_i^{t-1}.

We further sample several candidate bounding-boxes for d_i^t based on $\bar{b}_i^t$. Let $\hat{b}_i^t$ be one of the sampled bounding-boxes, and $\hat{d}_i^t = (\hat{b}_i^t, p_i^{t-1})$ be the candicate detection for d_i^{t-1} in frame t. A graph $G_{\hat{d}_i^t}$ can be built based on $N_{\hat{d}_i^t} = \{\hat{d}_i^t\} \cup \{d_{i_k}^t\}_{k=1}^K$. Then a similarity score is computed between $G_{\hat{d}_i^t}$ and $G_{d_i^{t-1}}$. Among all these candidate detections, the one with the highest similarity score is chosen as the tracked state for d_i^{t-1} in frame t if the similarity score is high enough.

4 Experiments

The proposed Graph Similarity Model is implemented based on PyTorch library. Evaluation is on a workstation with 2.6 GHz CPU and Nvidia TITAN Xp GPU.

4.1 Datasets

Experiments are conducted on MOT Benchmarks, including MOT16 and MOT17 [Milan *et al.*, 2016]. The MOT17 dataset contains 14 sequences each of which is provided with three sets of public detections. The detections produced by different object detectors each with increasing performance, namely DPM [Felzenszwalb *et al.*, 2010], FRCNN [Ren *et al.*, 2015] and SDP [Yang *et al.*, 2016]. Seven of these 14 sequences are used for training and the remains are for testing. The MOT16 dataset also contain the same sequences as MOT17 but only provided with one set of public detections produced by DPM.

4.2 Evaluation Metrics

We adopt the standard metrics of MOT Benchmarks: Multi-Object Tracking Accuracy (MOTA) [Bernardin and Stiefelhagen, 2008], Multi-object Tracking Precision (MOTP), [Bernardin and Stiefelhagen, 2008], how often an object is identified by the same ID (IDF1), Mostly Tracked objects (MT), Mostly Lost objects (ML), number of False Positives (FP), number of False Negatives (FN), number of Identity Switches (IDS) [Li *et al.*, 2009] and number of Fragments (Frag).

models	MOTA↑	IDF1↑	IDS↓	MT↑	ML↓	FP↓	FN↓	Frag↓
Naive^0	40.0%	38.7%	114	**16.5%**	38.0%	835	9595	277
GSM^0	39.9%	38.7%	123	**16.5%**	39.2%	967	9478	297
Naive$_\mathrm{h}^5$	34.8%	36.7%	470	10.1%	38.0%	1371	9741	569
Naive$_\mathrm{s}^5$	32.8%	17.8%	707	8.9%	**36.7%**	1244	9745	538
GSM$_\mathrm{h}^5$	37.2%	36.4%	190	12.7%	43.0%	1150	9702	283
GSM$_\mathrm{s}^5$	40.4%	44.6%	91	15.2%	39.2%	**799**	9583	**259**
GSM$_\mathrm{s+f}^5$	**41.4%**	**46.6%**	**84**	**16.5%**	38.0%	860	**9357**	287

Table 1: Tracking performance on validation set with different settings. Values in bold highlight the best results. The superscripts of model names denote the values of K.

4.3 Implement Details

All image patches are resized to 64 × 128. The CNN used to extract the appearance features from image patches is modified from ResNet-34 [He *et al.*, 2016]. Particularly, the last fc-layer is removed, producing a 256 channel feature map with spatial size 2 × 4. The feature map is reshaped to a 2048-d feature vector followed by another fc-layer, which produces a 256-d appearance feature vector. The relative position is also embedded into a 256-d feature vector by RPE.

The appearance CNN, RPE and binary classifer are trained end-to-end with binary cross entropy loss for 30 epochs. The input $[|v_{i_k}^{t-1} \ominus v_{j_{k'}}^t|^2, |e_{i_k}^{t-1} \ominus e_{j_{k'}}^t|^2]$ (see Eq. (6)) of the classifier is a positive sample only when the following two conditions are met: (1) d_i^{t-1} and d_j^t are the same object, (2) $d_{i_k}^{t-1}$ and $d_{j_{k'}}^t$ are also the same object. The learning rate is initialized as 0.002 and decades every 10 epochs with exponential decay rate 0.5. Online hard example mining (OHEM) was adopted to address the imbalance of positive/negative issue.

4.4 Ablation Study

The 7 sequences in MOT16 train split are divided into train set and validation set to conduct ablation study. Validation set: MOT16-09 and MOT16-10. Training set: the rest sequences in MOT16 train split.

In order to show the effectiveness of our proposed GSM model, a Naive model, which consists of an appearance CNN and a binary classifier, was also designed. The classifier in naive model predicts the similarity score of two detections based on the individual appearance feature vectors. Then two simple trackers (denoted as GSM and Naive) are designed. Note that the only difference between them is the similarity model they used. Both trackers track objects by performing data association procedure between two frame.

The study was carried out on the validation set with public DPM [Felzenszwalb *et al.*, 2010] detections. The detections were directly used without any processing.

Quantitative Results

Quantitative results are shown in Table 1. The first two rows compare the Naive and GSM models without neighbors. Two models achieve almost the same performance. Recall the 8-d relative position in Eq.(4), the relative position of one detection to itself is all zeros, which means that only appearance features are used in GSM when $K = 0$.

The following four rows compare the hard and soft graph matching (indicated by the subscripts of model names) when 5 neighbors are used for Naive and GSM models. For Naive models, the graph are constructed without edges. Both hard and soft graph matching methods have a negative impact on the performance and there are obvious reasons for that: (1) Two graphs that are built for two different objects are much similar to each other if the same objects are used as the neighbors. (2) The negative effect to the case that two graphs are built for two different objects can not be negligible since only appearance features are used. For GSM models, hard graph matching decades the tracking performance when comparing GSM$_\mathrm{h}^5$ with GSM^0. The reason is that two graphs built for the same object possess a much lower similarity score if the neighbors are not all the same, as shown in Figure 4 (a). Compared with Naive^0 and GSM^0, GSM$_\mathrm{s}^5$ achieves a 5.9% higher IDF1 and a much lower IDS, demonstrating that GSM$_\mathrm{s}^5$ tracks one object with the same ID more often.

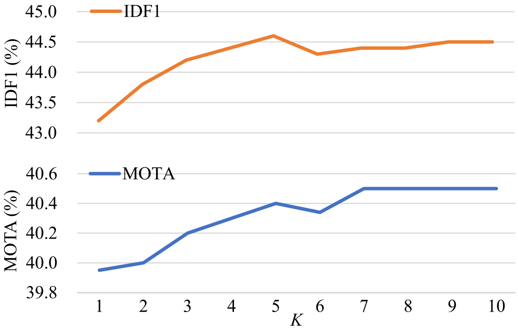


Figure 5: Tracking performance with respect to the value of K. Soft graph matching is used in the tracker.

The last row shows the effectiveness of finding lost objects with the help of graph matching, denoted as GSM$_\mathrm{s+f}^5$. 64 candidate bounding-boxes are sampled for each lost object. The one that best matches the graph is used as the tracked position if the similarity score is high enough. As shown in Table 1, GSM$_\mathrm{s+f}^5$ possesses a 1.0% higher MOTA and a 2.0% higher IDF1 compared with GSM$_\mathrm{s}^5$. In addition, GSM$_\mathrm{s+f}^5$ achieves a better FN, which means some lost objects are re-find successfully. However, re-findinging some lost objects also leads to a higher FP.

Extensive experiments are also conducted to find the best value of K, as shown in Figure 5. Over all, the tracker achieves a slightly higher MOTA but almost the same IDF1 when $K \geq 5$. Here we try to give an explanation. With the increase of K, the similarity score of two graphs obtained by Eq.(8) that built for the same object should be more reliable. On the contrary, the similarity score of two graphs that are built for different objects is more unreliable. The positive and negative impact on the similarity score are offset by each other to further improve the tracking performance. A larger value of K takes more time to align the neighbors. We set K as 5 in default to trade off the time consumption and tracking performance. The time consumption for constructing a graph and matching two graphs are about 0.15 ms and 0.03 ms, respectively. On our validation set, replacing the baseline similarity model with the proposed GSM leads to the tracking speed dropping from 93.7 fps to 61.5 fps.

benchmark	trackers	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓	Frag↓
MOT16	MTDF [Fu *et al.*, 2019]	45.7%	72.6%	40.1%	14.1%	36.4%	12018	84970	1987	3377
	STAM [Chu *et al.*, 2017]	46.0%	74.9%	50.0%	14.6%	43.6%	6895	91117	**473**	1422
	DMMOT [Zhu *et al.*, 2018]	46.1%	73.8%	54.8%	17.4%	42.7%	7909	89874	532	1616
	AMIR [Sadeghian *et al.*, 2017]	47.2%	75.8%	46.3%	14.0%	41.6%	**2681**	92856	774	1675
	STRN [Xu *et al.*, 2019]	48.5%	73.7%	53.9%	17.0%	34.9%	9038	84178	747	2919
	Tracktor [Bergmann *et al.*, 2019]	54.4%	**78.2%**	52.5%	19.0%	36.9%	3280	79149	682	1480
	$\mathrm{GSM}_{\mathrm{Tracktor}}$ (Ours)	**57.0%**	78.1%	**58.2%**	**22.0%**	**34.5%**	4332	**73573**	475	**859**
MOT17	DMAN [Zhu *et al.*, 2018]	48.2%	75.7%	55.7%	19.3%	38.3%	26218	263608	2194	5378
	HAM_SADF [Yoon *et al.*, 2018]	48.3%	77.2%	51.1%	17.1%	41.7%	20967	269038	1871	3020
	MTDF [Fu *et al.*, 2019]	49.6%	75.5%	45.2%	18.9%	**33.1%**	37124	241768	5567	9260
	MOTDT[Chen *et al.*, 2018]	50.9%	76.6%	52.7%	17.5%	35.7%	24069	250768	2474	5317
	STRN [Xu *et al.*, 2019]	50.9%	75.6%	56.0%	18.9%	33.8%	25295	249365	2397	9363
	FAMNet [Chu and Ling, 2019]	52.0%	76.5%	48.7%	19.1%	33.4%	14138	253616	3072	5318
	Tracktor [Bergmann *et al.*, 2019]	53.5%	**78.0%**	52.3%	19.5%	36.6%	**12201**	248075	2012	4611
	$\mathrm{GSM}_{\mathrm{Tracktor}}$ (Ours)	**56.4%**	77.9%	**57.8%**	**22.2%**	34.5%	14379	**230174**	**1485**	**2763**

Table 2: Results of different trackers on MOT benchmarks. Values in bold highlight the best results.

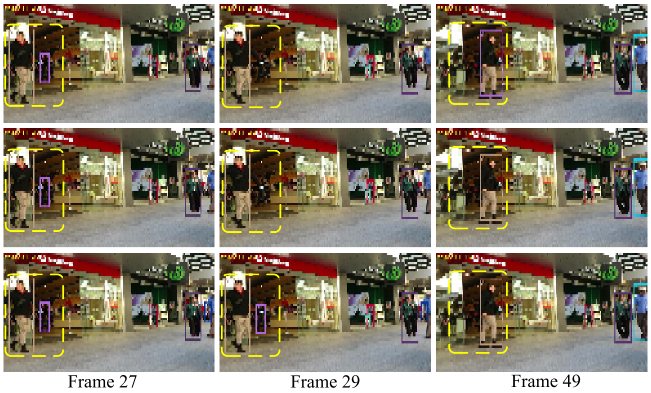


Figure 6: Top row: track results of Naive^0. Middle row: track results of $\mathrm{GSM}_{\mathrm{s}}^{5}$. Bottom row: track results of $\mathrm{GSM}_{\mathrm{s+f}}^{5}$. Please pay attention to the objects within yellow dashed boxes. The color of bounding-boxes and the numbers located at the top-left corner of bounding-boxes indicate the IDs of objects.

Qualitative Results

Qualitative results are also shown in Figure 6 to demonstrate the effectiveness of graph matching and the finding of lost object. At frame 27, there are two objects (denoted as object-2 and object-8) in the yellow dashed box. At frame 29, object-2 is occluded by object-8, leading to un-detected by the detector. However, the position of object-2 at frame 29 can be well estimated by $\mathrm{GSM}_{\mathrm{s+f}}^{5}$. At frame 49, object-2 is totally occluded by object-8. The Naive^0 model recognizes object-8 as object-2 erroneously. However, both $\mathrm{GSM}_{\mathrm{s}}^{5}$ and $\mathrm{GSM}_{\mathrm{s+f}}^{5}$ track objects successfully.

4.5 Results on MOT Benchmarks

We apply the proposed Graph Similarity Model to the state-of-the-art method Tracktor [Bergmann *et al.*, 2019], denoted as $\mathrm{GSM}_{\mathrm{Tracktor}}$, and compare the integrated tracker with other online methods. Results are shown in Table 2.

MOT16. $\mathrm{GSM}_{\mathrm{Tracktor}}$ achieves the best results in all metrics except MOTP, FP and IDS. In terms of IDS, $\mathrm{GSM}_{\mathrm{Tracktor}}$ takes the second place, only a little higher (475 than 473) than the first. Compared with Tracktor, MOTA and IDF1 are greatly improved by 5.7% and 3.6% respectively. What's more, IDS is also significantly reduced by 30.4%. The better results of IDF1 and IDS achieved by $\mathrm{GMS}_{\mathrm{Tracktor}}$ demonstrates the effectiveness of the proposed graph representation and matching.

MOT17. Overall, the integrated tracker $\mathrm{GSM}_{\mathrm{Tracktor}}$ achieves the state-of-the-art results. Compared with Tracktor, $\mathrm{GSM}_{\mathrm{Tracktor}}$ performs better in most metrics. Specifically, MOTA and IDF1 are improved by 2.9% and 5.5%, and IDS is reduced by more than 20%, which demonstrates that the similarity score obtained by graph representation and matching is much more reliable.

5 Conclusion

In this paper, a Graph Similarity Model (GSM) is proposed to improve the accuracy of data association for MOT. Based on the designed graph representation, the proposed GSM model takes both the individual representation and the relations among objects into consideration. Besides, we build a graph matching module that can effectively alleviate the impact of the unreliable detections. With the help of the graph representation and the graph matching module, the proposed GSM can effectively improve the robustness of the similarity model, especially on the case where occlusion happens and different objects share the similar appearance. In addition, the proposed GSM can be applied to any MOT trackers based on data association. Experimental results on challenging MOT benchmarks demonstrate the effectiveness of our GSM.

Acknowledgments

This work is partly supported by the Major Scientific Research Project of Zhejiang Lab (No. 2019DB0ZX01), the Key Laboratory Foundation of the Chinese Academy of Sciences (CXJJ-17S044) and the Fundamental Research Funds for the Central Universities (WK2100330002, WK3480000005).

References

[Bergmann *et al.*, 2019] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019.

[Bernardin and Stiefelhagen, 2008] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.

[Breitenstein *et al.*, 2009] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, pages 1515–1522. IEEE, 2009.

[Chen *et al.*, 2018] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, pages 1–6. IEEE, 2018.

[Chu and Ling, 2019] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV*, pages 6172–6181, 2019.

[Chu *et al.*, 2017] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *ICCV*, pages 4836–4845, 2017.

[Felzenszwalb *et al.*, 2010] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[Fu *et al.*, 2019] Zeyu Fu, Federico Angelini, Jonathon Chambers, and Syed Mohsen Naqvi. Multi-level cooperative fusion of gm-phd filters for online multiple human tracking. *IEEE Transactions on Multimedia*, 21(9):2277–2291, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Helbing and Molnar, 1995] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[Izadinia *et al.*, 2012] Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. 2 t: multiple people multiple parts tracker. In *ECCV*, pages 100–114. Springer, 2012.

[Kuo *et al.*, 2010] Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, pages 685–692. IEEE, 2010.

[Li *et al.*, 2009] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960. IEEE, 2009.

[Maksai and Fua, 2019] Andrii Maksai and Pascal Fua. Eliminating exposure bias and metric mismatch in multiple object tracking. In *CVPR*, pages 4639–4648, 2019.

[Milan *et al.*, 2013] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *TPAMI*, 36(1):58–72, 2013.

[Milan *et al.*, 2016] 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.

[Pellegrini *et al.*, 2009] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268. IEEE, 2009.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Sadeghian *et al.*, 2017] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*, pages 300–311, 2017.

[Vaswani *et al.*, 2017] 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*, pages 5998–6008, 2017.

[Xiang *et al.*, 2015] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *ICCV*, pages 4705–4713, 2015.

[Xu *et al.*, 2019] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *ICCV*, pages 3988–3998, 2019.

[Yamaguchi *et al.*, 2011] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *CVPR*, pages 1345–1352. IEEE, 2011.

[Yang and Nevatia, 2012] Bo Yang and Ram Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR*, pages 1918–1925. IEEE, 2012.

[Yang *et al.*, 2016] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, pages 2129–2137, 2016.

[Yoon *et al.*, 2018] Young-chul Yoon, Abhijeet Boragule, Young-min Song, Kwangjin Yoon, and Moongu Jeon. Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. In *AVSS*, pages 1–6. IEEE, 2018.

[Zhu *et al.*, 2018] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *ECCV*, pages 366–382, 2018.