Dual L_1-Normalized Context Aware Tensor Power Iteration and Its Applications to Multi-Object Tracking and Multi-Graph Matching

Weiming Hu, Xinchu Shi, Zongwei Zhou, and Junliang Xing
(National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences; CAS Center for Excellence in Brain Science and Intelligence Technology; University of Chinese Academy of Sciences, Beijing 100190)
{wmhu, xcshi, zwzhou, jlxing}@nlpr.ia.ac.cn

Haibin Ling
(Department of Computer and Information Science, Temple University, Philadelphia, USA)
hbling@temple.edu

Stephen Maybank
(Department of Computer Science and Information Systems, Birkbeck College, Malet Street, London WC1E 7HX)
sjmaybank@dcs.bbk.ac.uk

Abstract: The multi-dimensional assignment problem is universal for data association analysis such as data association-based visual multi-object tracking and multi-graph matching. In this paper, multi-dimensional assignment is formulated as a rank-1 tensor approximation problem. A dual L_1-normalized context/hyper-context aware tensor power iteration optimization method is proposed. The method is applied to multi-object tracking and multi-graph matching. In the optimization method, tensor power iteration with the dual unit norm enables the capture of information across multiple sample sets. Interactions between sample associations are modeled as contexts or hyper-contexts which are combined with the global affinity into a unified optimization. The optimization is flexible for accommodating various types of contextual models. In multi-object tracking, the global affinity is defined according to the appearance similarity between objects detected in different frames. Interactions between objects are modeled as motion contexts which are encoded into the global association optimization. The tracking method integrates high order motion information and high order appearance variation. The multi-graph matching method carries out matching over graph vertices and structure matching over graph edges simultaneously. The matching consistency across multi-graphs is based on the high-order tensor optimization. Various types of vertex affinities and edge/hyper-edge affinities are flexibly integrated. Experiments on several public datasets, such as the MOT16 challenge benchmark, validate the effectiveness of the proposed methods.

Index terms: Multi-dimensional assignment, Context/hyper-context aware tensor power iteration, Multi-object tracking, Multi-graph matching

1. Introduction

Multi-dimensional assignment is an important problem in data association analysis. Its aim is to find a one-to-one mapping between data in multiple sets. Many tasks can be formulated as multi-dimensional assignment.

For instance, in data association-based multi-object tracking, a batch of evidence [8, 9] is collected within a time span and tracking is treated as a multi-frame multi-object association problem. Multi-graph matching involves a search for correspondences across multi-sets of feature vectors where each feature vector is represented by a vertex and each set of feature vectors is represented by a graph.

In this paper, we propose a new multi-dimensional assignment method and apply it to data association-based multi-object tracking and multi-graph matching. In order to put our work into context, multi-dimensional assignment, data association-based multi-object tracking, and multi-graph matching are reviewed.

1.1. Related work

1.1.1. Multi-dimensional assignment

The integer optimization for multi-dimensional assignment is NP-hard for three or higher dimensional association. Some methods handle the global association using hierarchical strategies [5] in which the optimum local associations are carried out first and then are used to obtain longer tracks. There exist some approximate solutions, such as semi-definite programming [41] and Lagrange relaxation [42], for the multi-dimensional assignment problem. The existing methods can be classified into network flow-based, sampling-based, and iterative approximation-based:

- **Network flow-based methods** [2, 17, 25] decompose the global association affinity as the product of pairwise affinities between consecutive sample sets and then formulate multi-dimensional assignment as a network flow problem, which can be solved using linear programming [12], shortest path algorithms [2], the max-flow/min-cut optimization [25], or greedy search [17, 45], etc. These methods yield optimal solutions with polynomial time complexity. Their limitation is that only pairwise affinities are used and high order sequential information and longtime variation in sample features are not modeled.
- **Sampling-based methods** use probabilistic sampling strategies (e.g. Markov chain Monte Carlo sampling) [43, 44] to find a global solution for data association. The limitations of these methods are that the high-dimensional state estimation in multi-dimensional assignment typically requires a large computational cost and tuning the parameters to obtain a convergence is always difficult.
- **The iterative approximation-based methods** [7] iteratively solve two-frame assignments to search for the global solution by using the global affinity. These methods model the high order affinity. The limitations of these methods are that the computational complexity is high and the contexts between samples are not modeled.

1.1.2. Data association-based multi-object tracking

Multi-object tracking methods can be roughly divided into Bayesian filtering-based and data association-based. Bayesian filtering-based methods use only observations in the current frame to estimate the current object states [4, 13]. Data association-based methods use observations in the previous and current frames to estimate the states of the objects in these frames simultaneously, using the results of object detection in these frames. The association-based methods have become popular recently [8, 9]. They are reliable, in general, for solving data association jointly across multi-frames. This paper focuses on data association-based tracking.

Association-based multi-object tracking can be formulated as a network flow problem [2, 17, 25] by decomposing the global affinity between objects in a sequence of frames as the product of local pairwise affinities between objects in consecutive frames. The decomposition of the affinity leads to an efficient solution. However, the association discriminability is limited in that multi-frame motion information, which is useful for reducing the association ambiguity, is lost. Collins [7] used the global affinity between objects to enhance the association robustness. The limitation of his method is that interactions between the moving objects are not utilized to improve association accuracy.

Because motion contexts [1, 10, 16, 21] can reduce intrinsic association ambiguities caused by appearance similarity, occlusion, fast motion, and so on, modeling interactions among objects is useful for multi-object tracking. The classic social force model [11] used in pedestrian tracking [14, 15, 18] defines a series of social forces for an object to ensure collision avoidance and a desired direction for the destination. Its limitations are that it is complicated and requires pre-training from similar scenes, as well as prior knowledge, for example about the destination which is usually unavailable. Most methods that include an interaction-based motion model [1, 14, 15, 21] are limited to a predictive tracking framework. In [1], the motion context is a collection of trajectories of objects. It was used to predict and reacquire occluded objects. In [5], the association problem was formulated as finding the maximum weighted independent set. The interaction between two trajectories was embedded as a soft constraint. The limitation of these methods is that the local temporal association is often troubled by the intrinsic motion ambiguity.

1.1.3. Multi-graph matching

While matching two graphs has been studied intensively, multi-graph matching has received relatively less attention. In the following, two graph matching and multi-graph matching are reviewed respectively.

Matching two graphs is traditionally formulated as an optimization problem which is solved by the graduated assignment algorithm [46], the integer projected fixed point method [47], the spectral matching methods [48, 49], the path-following algorithms [39, 50, 51], etc. Both the pairwise edge affinity and the hyper-edge affinity are exploited in two-graph matching. The pairwise edge affinity is generally sensitive to the scaling and rotation, while hyper-edge affinity explores high-order structure information and is more robust to certain geometric transformations [29, 30, 52]. In particular, the algorithm in [29] uses a high-order tensor for hyper-graph matching between two graphs. Lee et al. [30] proposed a hypergraph matching method by reinterpreting the random walk concept on the hyper-graph in a probabilistic manner. Leordeanu et al. [65] proposed a hypergraph matching method, in which the parameters combining structural information and appearance information were learnt in a semi-supervised way. Nguyen et al. [66] proposed two tensor block coordinate ascent methods for hypergraph matching. Zeng et al. [67] proposed a graph matching method to address non-rigid surface matching. The limitation of two-graph matching is that high-order affinity among multi-graphs, which can be used to increase the matching consistency between vertices in different graphs, is not exploited.

Multi-graph matching methods can be roughly divided into affinity-driven and consistency-driven. The affinity-driven methods [19, 20, 35, 53] formulate multi-graph matching as an optimization problem in which the objective is usually the summation of the overall pairwise matching affinities [19], sometimes supplemented by matching consistency regularization [35]. For example, Sole-Ribalta and Serratosa [19] applied the graduated assignment algorithm [46] repeatedly across graph pairs to achieve cross graph matching. Yan et al. [35] carried out multi-graph matching by iteratively approximating the global-optimal affinity, while using regularization to gradually increase the consistency. The consistency-driven methods [34, 36] put more attention on the matching consistency. Yan et al. [36] proposed an iterative optimization solution with a rigid matching consistency constraint. Pachauri et al. [34] pooled all pairwise matching solutions into a single matrix and then estimated the globally consistent array of matches. The limitation of the above work is that high-order information both across multi-graphs and across hyper-edges is not handled.

In summary, the main limitation in the current methods for multi-dimensional assignment is that high order sequential information and longtime variation in sample features as well as the contexts between samples are not simultaneously modeled with low computational complexity. Correspondingly, the main limitations in the current methods for data association-based multi-object tracking are that motion contexts are not efficiently utilized without pre-training from similar scenes to model the interactions between moving objects, and multi-frame high-order motion information is not effectively combined with high-order appearance variation. The main limitations in the current methods for multi-graph matching are that high-order information across multi-graphs and high-order information across hyper-edges are not simultaneously modeled.

1.2. Our work

Our work handles the above main limitations in the current methods for multi-dimensional assignment, as well as data association-based multi-object tracking and multi-graph matching. As tensors are the tools for effectively representing high order information, we introduce rank-1 tensor approximation which has effective solutions with solid mathematical support, such as tensor power iteration, into the multi-dimensional assignment problem. Then, a dual L_1-normalized context/hyper-context aware tensor power iteration optimization method for multi-set sample association is proposed and applied to multi-object tracking and multi-graph matching [56].

In our dual L_1-normalized context/hyper-context aware tensor power iteration optimization method, a high-order tensor is constructed from a sequence of sets of samples. The low rank approximation to this tensor has the same affinity formulation as the multi-dimensional assignment problem. A tensor power iteration method with row/column unit norm (i.e., dual L_1-normalized) is proposed to solve the context/hyper-context aware tensor approximation problem. Interactions between sample associations are modeled as contexts or hyper-contexts and combined with the global affinity into the power iteration solution. In our multi-object tracking method, objects detected in each frame are treated as samples in a set. The global affinity is defined according to the appearance similarity between objects in different frames. Motion contexts are constructed to model the interaction between associations. Then, the dual L_1-normalized context-aware tensor power iteration optimization is applied to obtain the associations of the objects. In the multi-graph matching method, each vertex in a graph is treated as a sample, and the graph is treated as a sample set [20]. The affinity of the vertices is formulated as a global association affinity and the structure affinity over a set of hyper-edges as a hyper-context affinity. The dual L_1-normalized hyper-context aware tensor power iteration optimization is applied to match the vertices in the graphs.

The contributions of our work are summarized as follows:

- We formulate the objective of multi-dimensional assignment as the objective of rank-1 tensor approximation, and incorporate context into the multi-dimensional assignment formulation. By mathematical derivation, we ensure that the context-aware multi-dimensional assignment problem is solvable and propose an effective context-aware tensor power iteration method, in which the additional runtime for modeling the contexts is very small. We incorporate hyper-contexts into the multi-dimensional assignment problem and propose an effective hyper-context aware power iteration method. In this way, our dual L_1-normalized context/hyper-context aware tensor power iteration optimization method captures information across multiple sample sets. Contexts or hyper-contexts are utilized to characterize interactions between sample associations. The optimization framework provides the flexibility to use different context information.

- Our multi-object tracking method constructs the motion contexts to model the interaction between moving objects. The tracking method effectively integrates high-order motion information and high-order appearance variation.
- In contrast with the previous multi-graph matching methods, which use only pairwise affinities and ignore the high-order information in multi-sets of vertices, our multi-graph matching method works on high-order affinity tensors and naturally improves the matching. The information on the vertex affinities and the information on the edge/hyper-edge affinities are combined in a flexible way.

We test our multi-object tracking method and multi-graph matching method on several datasets, such as the MOT16 challenge benchmark. For different datasets or different applications, different affinities between objects are defined. For example, on the MOT16 challenge benchmark dataset, the affinities are defined using the features from deep siamese neural networks. It is shown that our methods have excellent performance in comparison with the state of the art.

The remainder of the paper is organized as follows: Section 2 briefly introduces rank-1 tensor approximation. Section 3 describes the dual L_1-normalized rank-1 tensor approximation. Sections 4 and 5 propose context and hyper-context aware tensor power iterations. Sections 6 and 7 present our multi-object tracking method and our multi-graph matching method respectively. Section 8 demonstrates the experimental results. Section 9 summarizes the paper.

2. Rank-1 Tensor Approximation

A tensor is the high dimensional generalization of a matrix. Each element in a K-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \ldots I_k \ldots I_K}$ is represented as $a_{i_1 \ldots i_{k-1} i_k i_{k+1} \ldots i_K}$ where $1 \le i_k \le I_k$. Each order of a tensor is associated with a mode. The k-mode product of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \ldots I_{k-1} \times I_k \times I_{k+1} \ldots I_K}$ and a matrix $\mathbf{W} \in \mathbb{R}^{I_k \times J_k}$ is a new tensor $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \ldots I_{k-1} \times J_k \times I_{k+1} \ldots I_K}$ whose entries are

$$b_{i_1 \ldots i_{k-1} j_k i_{i+1} i_K} = \sum_{i_k=1}^{I_k} a_{i_1 \ldots i_{k-1} i_k i_{k+1} \ldots i_K} w_{i_k j_k} \ . \tag{1}$$

This k-mode product is notated as $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \otimes_k \mathbf{W}$. In particular, the k-mode product of $\boldsymbol{\mathcal{A}}$ and a vector $\mathbf{w} \in \mathbb{R}^{I_k}$ is a $K-1$ order tensor:

$$(\mathcal{A}\otimes_k \mathbf{w})_{i_1\ldots i_{k-1}i_{i+1}i_K} = \sum_{i_k=1}^{I_k} a_{i_1\ldots i_{k-1}i_k i_{k+1}\ldots i_K} w_{i_k} \,. \tag{2}$$

A rank-1 tensor $\hat{\mathcal{C}} \in \mathbb{R}^{I_1\times\ldots I_{k-1}\times I_k\times I_{k+1}\ldots I_K}$ is a specific tensor which can be represented as the outer product ($*$) of K vectors $\{\hat{\mathbf{w}}^k \in \mathbb{R}^{I_k}\}_{k=1}^{K}$: $\hat{\mathcal{C}}=\hat{\mathbf{w}}^1 * \hat{\mathbf{w}}^2 \ldots \hat{\mathbf{w}}^k \ldots * \hat{\mathbf{w}}^K$, i.e., an element in $\hat{\mathcal{C}}$ is represented as:

$$\hat{c}_{i_1\ldots i_k\ldots i_K} = (\hat{\mathbf{w}}^1 * \ldots * \hat{\mathbf{w}}^k \ldots * \hat{\mathbf{w}}^K)_{i_1\ldots i_k\ldots i_K} = \hat{w}_{i_1}^1 \hat{w}_{i_2}^2 \ldots \hat{w}_{i_k}^k \ldots \hat{w}_{i_K}^K \,, \tag{3}$$

where $\hat{w}_{i_k}^k$ is the i_k th element in $\hat{\mathbf{w}}^k$. Let $\{\mathbf{w}^k \in \mathbb{R}^{I_k}\}_{k=1}^{K}$ be K L_2 unit-normalized column vectors and let $\mathbf{W}$ be the matrix composed of $\{\mathbf{w}^k \in \mathbb{R}^{I_k}\}_{k=1}^{K}$. A rank-1 approximation to a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\ldots I_{k-1}\times I_k\times I_{k+1}\ldots I_K}$ is obtained by finding the vectors $\{\mathbf{w}^k \in \mathbb{R}^{I_k}\}_{k=1}^{K}$ and a scalar γ for minimizing the following square of the Frobenius norm:

$$\min_{\gamma,\mathbf{W}} \left\| \mathcal{A} - \gamma \mathbf{w}^1 * \mathbf{w}^k \ldots * \mathbf{w}^K \right\|_F^2 = \min_{\gamma,\mathbf{W}} \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\ldots\sum_{i_K=1}^{I_K} \left(a_{i_1 i_2\ldots i_K} - \gamma w_{i_1}^1 w_{i_2}^2 \ldots w_{i_K}^K \right)^2 . \tag{4}$$

By solving (4), the tensor $\mathcal{A}$ is approximated by the rank-1 tensor $\gamma \mathbf{w}^1 * \mathbf{w}^k \ldots * \mathbf{w}^K$. A function g is defined as:

$$g(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K) = \left| \mathcal{A} \otimes_1 \mathbf{w}^1 \otimes_2 \mathbf{w}^2 \ldots \otimes_K \mathbf{w}^K \right| = \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\ldots\sum_{i_K=1}^{I_K} a_{i_1 i_2\ldots i_K} w_{i_1}^1 w_{i_2}^2 \ldots w_{i_K}^K \,. \tag{5}$$

With some derivations as shown in [26, 27], the optimization in (4) has the following equivalent form:

$$\max_{\mathbf{W}} g(\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K) \,. \tag{6}$$

Tensor power iteration [26, 27] has been proposed to optimize (6).

3. Dual L_1-Normalized Rank-1 Tensor Approximation

Many applications, such as multi-frame data association and multi-graph matching, can be formulated as multi-dimensional assignment problems [7]. We transform the multi-dimensional assignment problem to a rank-1 tensor approximation problem. Then, mathematical techniques for rank-1 tensor approximation are introduced to solve the multi-dimensional assignment problem.

3.1. Formulation

Suppose that there is a sequence of K+1 sets of samples and each set has N samples[1]. Let i_k be a sample index in the kth set. A trajectory $i_0 i_1 i_2 \ldots i_K$ is a sequence of K+1 samples from the K+1 sets respectively (We

[1] We assume that there is the same number of samples in each set. When there are different numbers of samples in each set, virtual samples are added to the sets to make the number of samples in each set the same. After the association is carried out, the samples associated with the virtual samples are the isolated samples.

index sample sets stating from 0 for description convenience). Let $a_{i_0 i_1 i_2 \ldots i_K}$ be the affinity of trajectory $i_0 i_1 \ldots i_K$ whose label $x_{i_0 i_1 \ldots i_K}$ is 1 if the trajectory is actually existent, otherwise is 0. An actually existent trajectory has higher affinity between the samples in it. Multi-dimensional assignment is formulated as:

$$\max \sum_{i_0=1}^{N} \sum_{i_1=1}^{N} \ldots\ldots \sum_{i_K=1}^{N} a_{i_0 i_1 \ldots i_K} x_{i_0 i_1 \ldots i_K}, \tag{7}$$

$$s.t. \begin{cases} x_{i_0 i_1 \ldots i_K} \in \{0,1\}, 0 \le k \le K; \\ \forall i_k \in \{1,2,..,N\}, \sum_{i_0=1}^{N} \sum_{i_1=1}^{N} \ldots \sum_{i_{k-1}}^{N} \sum_{i_{k+1}}^{N} \ldots \sum_{i_K=1}^{N} x_{i_0 i_1 \ldots i_k \ldots i_K} = 1. \end{cases} \tag{8}$$

Actually existent trajectories are found by solving this constrained integer optimization.

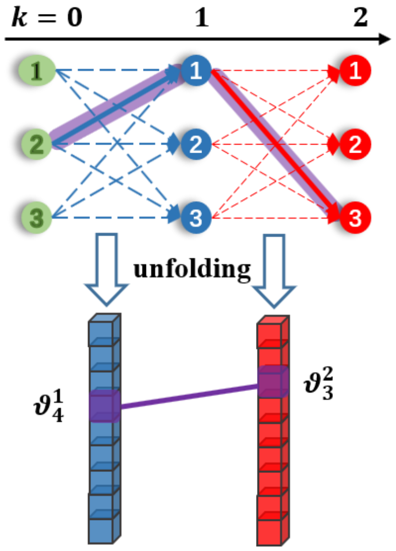


Fig. 1. The relation between the association matrix and the association vector: The sample association (i_k, i_{k+1}) between two consecutive sets k and k+1 is represented by vector $\vartheta^{k+1} \in R^{N \times N}$. The upper part shows all the possible associations between three consecutive sets; The lower part shows the corresponding vector representation. The relation between the second sample in set 0 and the first sample in frame 1 $(i_0 = 2, i_1 = 1)$ corresponds to the fourth element ϑ_4^1 in association vector ϑ^1; The association $(i_1 = 1, i_2 = 3)$ corresponds to the third element ϑ_3^2 in association vector ϑ^2.

We decompose a global association $x_{i_0 i_1 \ldots i_K}$ into a sequence of pairwise associations:

$$x_{i_0 i_1 \ldots i_K} = x_{i_0, i_1}^{1} x_{i_1, i_2}^{2} \ldots x_{i_{K-1}, i_K}^{K}, \tag{9}$$

where $x_{i_{k-1}, i_k}^{k} \in \{0,1\}$ is the association between the i_{k-1}th sample in the k-1th set and the i_k th sample in the kth set. Only if all the pairwise associations in the sequence are true (i.e., take value 1), is the global association also true. It is apparent that there are N^2 associations between two consecutive sample sets. In order to transform multi-dimensional assignment to a rank-1 tensor approximation problem, we flatten (unfold) the association matrix $\left[x_{i_{k-1},i_k}^k\right]_{N\times N}$ between the k-1th sample set and the kth sample set into a vector $\left[w_{\vartheta_k}^k\right]_{\vartheta_k=1}^{N^2}$. To more clearly distinguish the association matrix and the flattened association vector, we use bold italic font to indicate the elements in an association vector. The relation between the association matrix and the association vector is illustrated in Fig. 1. The equivalent relation between the association index ϑ_k in the vector and the indices i_{k-1} and i_k in the association matrix is:

$$\vartheta_k = (i_{k-1}-1)N + i_k . \tag{10}$$

In this way, an association indexed by (i_{k-1}, i_k) in the association matrix $\left[x_{i_{k-1},i_k}^k\right]_{N\times N}$ is also indexed by ϑ_k in the association vector $\mathbf{w}^k = \left(w_{\vartheta_k}^k\right)_{\vartheta_k=1}^{N^2}$: $x_{i_{k-1},i_k}^k = w_{\vartheta_k}^k$.

We rearrange the affinity $a_{i_0 i_1 i_2 \ldots i_K}$ using the indices in association vectors $\{\mathbf{w}^k\}$. In the k-1th set the index i_{k-1} of the sample included in the association ϑ_k is $\lceil \vartheta_k / N \rceil$, where $\lceil \ \rceil$ is the up rounding operator; and in the kth set the i_k index of the sample included in the ϑ_k th association is $\vartheta_k - \left(\lceil \vartheta_k / N \rceil - 1\right)N$:

$$\begin{cases} i_{k-1} = \lceil \vartheta_k / N \rceil, \\ i_k = \vartheta_k - \left(\lceil \vartheta_k / N \rceil - 1\right)N. \end{cases} \tag{11}$$

The consecutive associations ϑ_k and ϑ_{k+1} have affinity only if they share the same sample in the kth set. Then, we define the affinity $s_{\vartheta_1\vartheta_2\ldots\vartheta_K}$ of the global association consisting of the consecutive pairwise associations $\{\vartheta_k\}_{k=1}^K$ as follows:

$$s_{\vartheta_1\vartheta_2\ldots\vartheta_K} = \begin{cases} a_{i_0 i_1 \ldots i_K} & \mathit{if}\ \vartheta_k - \left(\lceil \vartheta_k / N \rceil - 1\right)N = \lceil \vartheta_{k+1} / N \rceil, k = 1,2,\ldots K-1 \\ 0 & \mathit{otherwise.} \end{cases} \tag{12}$$

where $\vartheta_k - \left(\lceil \vartheta_k / N \rceil - 1\right)N = \lceil \vartheta_{k+1} / N \rceil$ means that associations ϑ_k and ϑ_{k+1} share the same sample in in the kth sample set.

Using the pairwise association vectors $\{\mathbf{w}^k\}_{k=1}^K$ and the affinities defined on $\{\mathbf{w}^k\}_{k=1}^K$, we can transform multi-dimensional assignment to rank-1 tensor approximation. Let $\mathbf{W} \in \mathbb{R}^{N^2 \times K}$ be the matrix composed of $\{\mathbf{w}^k\}_{k=1}^K$. Using (10) and (12), the objective of multi-dimensional assignment formulated in (7) is transformed to the objective of rank-1 tensor approximation:

$$\max_{\mathbf{w}} g(\mathbf{w}^1_{\vartheta_1}, \mathbf{w}^2_{\vartheta_2}, ..., \mathbf{w}^K_{\vartheta_K}) = \max_{\mathbf{w}} \sum_{\vartheta_1=1}^{N^2} \sum_{\vartheta_2=1}^{N^2} .... \sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1 \vartheta_2 ... \vartheta_K} w^1_{\vartheta_1} w^2_{\vartheta_2} ... w^K_{\vartheta_K} \ . \tag{13}$$

The global constraint in (8) is decomposed into the following local pairwise constraints:

$$\begin{cases} \sum_{i_{k-1}=1}^{N} x^k_{i_{k-1}, i_k} = 1 \\ \sum_{i_k=1}^{N} x^k_{i_{k-1}, i_k} = 1 \end{cases} \quad 1 \le k \le K \ , \tag{14}$$

where $x^k_{i_{k-1}, i_k}$ is an element in the association matrix and it is equal to $w^k_{\vartheta_k}$. The dual L_1 norm in (14) is that both the rows and columns in the association matrix $\left[x^k_{i_{k-1}, i_k} \right]_{N \times N}$ are L_1-normalized. This ensures that one sample in the current set associates with only one sample in the subsequent set, and one sample in the subsequent set associates with only one sample in the current set. The optimization objective in (13) is the same as in (6). However, the original rank-1 tensor approximation in (6) is constrained by the L_2 norm, while the optimization in (13) is constrained by the dual L_1 norm. The methods for the original rank-1 tensor approximation are not suitable for solving the dual L_1 normalized rank-1 tensor approximation.

3.2. Solution

We carry out the optimization in (13) by an iterative algorithm that finds the association variables $\{w^k_{\vartheta_k}\}^{k=1,...,K}_{\vartheta_k=1,...,N^2}$. In each iteration some association variables are updated while the remaining association variables are fixed. It is required that in each iteration the value of the objective function is increased.

A power iteration method is utilized to adapt the dual L_1 unit norm constraint. The integer constraint on $w^k_{\vartheta_k}$ is relaxed to a real value constraint: $0 \le w^k_{\vartheta_k} \le 1$. Then, $w^k_{\vartheta_k}$ represents the probability of the association between the i_{k-1}th sample in the *k*-1th set and the i_k th sample in the *k*th set. A tensor power is used to iteratively update $\mathbf{w}^k = \{w^k_{\vartheta_k}\}^{N^2}_{\vartheta_k=1}$ followed by a dual L_1 unit normalization. The iteration is based on the partial differential of $g(\mathbf{w}^1, ..., \mathbf{w}^k, \mathbf{w}^{k+1}, ..., \mathbf{w}^K)$ for each association vector element $w^k_{\vartheta_k}$:

$$\frac{\partial g(\mathbf{w}^1, ..., \mathbf{w}^k, \mathbf{w}^{k+1}, ..., \mathbf{w}^K)}{\partial w^k_{\vartheta_k}} = \sum_{\vartheta_1=1}^{N^2} ... \sum_{\vartheta_{k-1}=1}^{N^2} \sum_{\vartheta_{k+1}=1}^{N^2} ... \sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1 ... \vartheta_k ... \vartheta_K} w^1_{\vartheta_1} ... w^{k-1}_{\vartheta_{k-1}} w^{k+1}_{\vartheta_{k+1}} ... w^K_{\vartheta_K} \ . \tag{15}$$

Let $\mathbf{w}^k(\tau)$ be the *k*th association vector at the τ th iteration. It has elements $\{w^k_{\vartheta_k}(\tau)\}^{N^2}_{\vartheta_k=1}$. At the $\tau+1$ th iteration, on considering the update of $\mathbf{w}^k(\tau)$, with all other association vectors

$\{\mathbf{w}^1(\tau+1),...,\mathbf{w}^{k-1}(\tau+1),\mathbf{w}^{k+1}(\tau),...,\mathbf{w}^K(\tau)\}$ fixed, the following equations are used to update $\mathbf{w}^k$:

$$w^k_{\vartheta_k}(\tau+1) \leftarrow w^k_{\vartheta_k}(\tau)\sum_{\vartheta_1=1}^{N^2}...\sum_{\vartheta_{k-1}=1}^{N^2}\sum_{\vartheta_{k+1}=1}^{N^2}...\sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1\vartheta_2...\vartheta_K} w^1_{\vartheta_1}(\tau+1)...w^{k-1}_{\vartheta_{k-1}}(\tau+1)w^{k+1}_{\vartheta_{k+1}}(\tau)...w^K_{\vartheta_K}(\tau)\,, \tag{16}$$

$$x^k_{i_{k-1},i_k}(\tau+1) \leftarrow \frac{x^k_{i_{k-1},i_k}(\tau+1)}{\sum_{l_k=1}^{N} x^k_{i_{k-1},l_k}(\tau+1)}\,, \tag{17}$$

$$x^k_{i_{k-1},i_k}(\tau+1) \leftarrow \frac{x^k_{i_{k-1},i_k}(\tau+1)}{\sum_{l_{k-1}=1}^{N} x^k_{l_{k-1},i_k}(\tau+1)}\,, \tag{18}$$

where (17) and (18) carry out the dual L_1 normalization corresponding to the constraint in (14), ensuring the one to one mapping between samples in consecutive sets. We can prove that

$$g(\mathbf{w}^1_{\vartheta_1}(\tau+1),...,\mathbf{w}^k_{\vartheta_k}(\tau+1),\mathbf{w}^{k+1}_{\vartheta_{k+1}}(\tau)...,\mathbf{w}^K_{\vartheta_K}(\tau)) \ge g(\mathbf{w}^1_{\vartheta_1}(\tau+1),...,\mathbf{w}^k_{\vartheta_k}(\tau),\mathbf{w}^{k+1}_{\vartheta_{k+1}}(\tau),...,\mathbf{w}^K_{\vartheta_K}(\tau))\,. \tag{19}$$

The convergence shown in (19) ensures that (16), (17), and (18) form an effective iteration algorithm for solving the dual L_1 normalized rank-1 tensor approximation problem. The original rank-1 tensor approximation in Section 2, constrained by the L_2 unit norm, has been proved to converge. In the following, we first prove that the iteration for the rank-1 tensor approximation constrained by the L_1 unit norm is convergent, and then give the proof of convergence of the dual L_1 normalized rank-1 tensor approximation.

Proposition A: For the L_1 normalized rank-1 tensor approximation, each element $w^1_{i_1}$ in $\mathbf{w}^1$ is updated by:

$$w^1_{i_1}(\tau+1) = \frac{1}{C^1} w^1_{i_1}(\tau)\sum_{i_2=1}^{I_2}...\sum_{i_K=1}^{I_K} a_{i_1i_2...i_K} w^2_{i_2}(\tau)...w^K_{i_K}(\tau)\,, \tag{20}$$

where C^1 is the L_1 normalization factor of $\mathbf{w}^1(\tau+1)$:

$$C^1 = \sum_{i_1=1}^{I_1} w^1_{i_1}(\tau)\sum_{i_2=1}^{I_2}...\sum_{i_K=1}^{I_K} a_{i_1i_2...i_K} w^2_{i_2}(\tau)...w^K_{i_K}(\tau)\,. \tag{21}$$

Then, we have

$$g(\mathbf{w}^1(\tau+1),\mathbf{w}^2(\tau),...,\mathbf{w}^K(\tau)) \ge g(\mathbf{w}^1(\tau),\mathbf{w}^2(\tau),...,\mathbf{w}^K(\tau))\,. \tag{22}$$

Namely, with the L_1 normalized $\mathbf{w}^1(\tau+1)$, the iteration using (20) converges.

Proof: We define two temporary vectors $\mathbf{G} = (f_1, f_2,..., f_{I_1})$ and $\mathbf{U} = (u_1,u_2,...,u_{I_1})$:

$$\begin{cases} f_{i_1} = \sum_{i_2=1}^{I_2} \cdots \sum_{x_K=1}^{I_K} a_{i_1 i_2 \ldots i_K} w_{i_2}^2(\tau) \ldots w_{i_K}^K(\tau), \\ u_{i_1} \times u_{i_1} = w_{i_1}^1(\tau). \end{cases} \tag{23}$$

Then, the following equation holds

$$\begin{aligned} & g(\mathbf{w}^1(\tau), \mathbf{w}^2(\tau), \ldots, \mathbf{w}^K(\tau)) \\ & = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} a_{i_1 i_2 \ldots i_K} w_{i_1}^1(\tau) w_{i_2}^2(\tau) \ldots w_{i_K}^K(\tau) \\ & = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} a_{i_1 i_2 \ldots i_K} u_{i_1} u_{i_1} w_{i_2}^2(\tau) \ldots w_{i_K}^K(\tau) \\ & = \sum_{i_1=1}^{I_1} u_{i_1} u_{i_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} a_{i_1 i_2 \ldots i_K} w_{i_2}^2(\tau) \ldots w_{i_K}^K(\tau) \\ & = \left\langle \mathbf{U}, \mathbf{U} \circ \mathbf{G} \right\rangle \end{aligned} \tag{24}$$

where '$\langle , \rangle$' and '$\circ$' denote the inner product and the Hadamard product (element-wise product) respectively. With the L_1 norm constraint $\|\mathbf{U}\|_2^2 = \|\mathbf{w}^1(\tau)\|_1 = 1$, the application of the Cauchy-Schwarz inequality to (24) yields

$$g(\mathbf{w}^1(\tau), \mathbf{w}^2(\tau), \ldots, \mathbf{w}^K(\tau)) = \left\langle \mathbf{U}, \mathbf{U} \circ \mathbf{G} \right\rangle \leq \|\mathbf{U}\|_2 \|\mathbf{U} \circ \mathbf{G}\|_2 = \|\mathbf{U} \circ \mathbf{G}\|_2 . \tag{25}$$

After $\mathbf{w}^1$ is iterated using (20), the objective function is represented as:

$$\begin{aligned} & g(\mathbf{w}^1(\tau+1), \mathbf{w}^2(\tau), \ldots, \mathbf{w}^N(\tau)) = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_K} a_{i_1 i_2 \ldots i_K} w_{i_1}^1(\tau+1) w_{i_2}^2(\tau) \ldots w_{i_K}^K(\tau) \\ & = \left\langle \mathbf{w}^1(\tau+1), \mathbf{G} \right\rangle = \frac{1}{C^1} \left\langle \mathbf{w}^1(\tau), \mathbf{G} \right\rangle = \frac{1}{C^1} \left\langle \mathbf{w}^1(\tau) \circ \mathbf{G}, \mathbf{G} \right\rangle \\ & = \frac{1}{C^1} \left\langle \mathbf{U} \circ \mathbf{G}, \mathbf{U} \circ \mathbf{G} \right\rangle = \frac{1}{C^1} \|\mathbf{U} \circ \mathbf{G}\|_2^2 . \end{aligned} \tag{26}$$

It is apparent that

$$C^1 = \sum_{i_1=1}^{I_1} w_{i_1}^1(\tau) \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} a_{i_1 i_2 \ldots i_K} w_{i_2}^2(\tau) \ldots w_{i_K}^K(\tau) = g(\mathbf{w}^1(\tau), \mathbf{w}^2(\tau), \ldots, \mathbf{w}^K(\tau)) . \tag{27}$$

By combining formulae (25), (26), and (27), we prove the inequality (22). The convergence for the iterations of $\mathbf{w}^2$, …, and $\mathbf{w}^K$ can be proved in the same way as for $\mathbf{w}^1$.

Proposition B: For the dual $L1$-normalized rank-1 tensor approximation, the following equations are used to update $\mathbf{w}^1$:

$$w_{\vartheta_1}^1(\tau+1) \leftarrow w_{\vartheta_1}^1(\tau) \sum_{\vartheta_1=1}^{N^2} \sum_{\vartheta_2=1}^{N^2} \cdots \sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1 \vartheta_2 \ldots \vartheta_K} w_{\vartheta_2}^2(\tau) \ldots w_{\vartheta_K}^K(\tau) \tag{28}$$

$$x^1_{i_0,i_1}(\tau+1) \leftarrow \frac{x^1_{i_0,i_1}(\tau+1)}{\sum_{l_1=1}^{N} x^1_{l_0,l_1}(\tau+1)}, \tag{29}$$

$$x^1_{i_0,i_1}(\tau+1) \leftarrow \frac{x^1_{i_0,i_1}(\tau+1)}{\sum_{l_0=1}^{N} x^1_{l_0,l_1}(\tau+1)}, \tag{30}$$

where (29) and (30) carry out the dual L_1 normalization. Then, we have

$$g(\mathbf{w}^1_{\vartheta_1}(\tau+1), \mathbf{w}^2_{\vartheta_2}(\tau), ..., \mathbf{w}^K_{\vartheta_K}(\tau)) \geq g(\mathbf{w}^1_{\vartheta_1}(\tau), \mathbf{w}^2_{\vartheta_2}(\tau), ..., \mathbf{w}^K_{\vartheta_K}(\tau)). \tag{31}$$

Proof: Similar to the definition of **G** using (23), we define a vector $\mathbf{G} = (f_1, f_2, ..., f_{N^2})$ as follows:

$$f_{\vartheta_1} = \sum_{\vartheta_2=1}^{N^2} \cdots \sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1\vartheta_2...\vartheta_K} w^2_{\vartheta_2}(\tau)...w^K_{\vartheta_K}(\tau) \quad (1 \leq \vartheta_1 \leq N^2). \tag{32}$$

Let $\mathbf{G}_{i_0} \in \mathbb{R}^{1\times N} = (f_{i_0,1}, f_{i_0,2}, ..., f_{i_0,N})$ $(1 \leq i_0 \leq N)$, where $\{i_0, j\}_{j=1}^{N}$ are defined using (11). Set $\mathbf{G} = \{\mathbf{G}_{i_0}\}_{i_0=1}^{N}$. The objective function is represented as:

$$g(\mathbf{w}^1(\tau+1), \mathbf{w}^2(\tau), ..., \mathbf{w}^K(\tau)) = \left\langle \mathbf{w}^1((\tau+1), \mathbf{G} \right\rangle = \sum_{i_0=1}^{N} \left\langle \mathbf{w}_{i_0}, \mathbf{G}_{i_0} \right\rangle. \tag{33}$$

The objective function is the sum of N components $\left\langle \mathbf{w}_{i_0}, \mathbf{G}_{i_0} \right\rangle$. Each component $\left\langle \mathbf{w}_{i_0}, \mathbf{G}_{i_0} \right\rangle$ corresponds to a L_1 normalized rank-1 tensor approximation whose iteration convergence is proved in Proposition A. Therefore, the iterations in (28), (29) and (30) for the dual L_1-normalized rank-1 tensor approximation also converge. The convergence for the iterations of $\mathbf{w}^2$, …, and $\mathbf{w}^K$ can be proved in the same way as for $\mathbf{w}^1$.

3.3. Discussion

In the solution, it is assumed that the sample sets have the same number of samples. When different numbers of samples exist in different sample sets, virtual samples are added to sample sets to make the number of samples in each set the same. The affinities to virtual samples are set to a fixed small value. This fixed small value corresponds to the probability that samples appear or disappear. The same small affinity of the virtual samples in a set to the samples in other sets ensures that the virtual samples do not influence the matching of the non-virtual samples in the sets. After finalization of association, the isolated samples in one set are associated with the virtual samples in other sets.

The above tensor formulation has various applications, depending on the form of the elements $s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K}$ in the tensor. In particular, two applications are 2D assignment and network flow. If $s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K}$ is decomposed as the sum of pairwise affinities: $s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K}=\sum_{k=1}^{K}s^k_{\vartheta_k}$, where $s^k_{\vartheta_k}$ denotes the affinity of the ϑ_k th association between the k-1th sample set and the kth sample set, then the objective in (13) is reformulated as: $N^{K-1}\sum_{k=1}^{K}\sum_{\vartheta_k=1}^{N^2}s^k_{\vartheta_k}w^k_{\vartheta_k}$. This optimization corresponds to the 2D assignment problem. When the affinity is computed as the product of pairwise affinities: $s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K}=\prod_{k=1}^{K}s^k_{\vartheta_k}$, the objective in (13) is rewritten as $\prod_{k=1}^{K}\sum_{\vartheta_k=1}^{N^2}s^k_{\vartheta_k}w^k_{\vartheta_k}$. This objective is appropriate for network flow [2]. As a result, tensor approximation provides a flexible framework to take advantage of global and local association affinities. However, the above tensor approximation does not encode context information between trajectories.

4. Context-Aware Tensor Power Iteration

Contexts between samples can be used to reduce the unreliability of sample associations. For example, the moving vehicles in a local spatiotemporal space usually have similar motion patterns. The determination of the association for a vehicle can be improved using the motion information about other vehicles. We combine the pairwise contextual relations between associations into the optimization objective and propose a dual L_1-normalized context-aware tensor power iteration algorithm to determine the relations between samples.

Let $c^k_{\vartheta_k\varsigma_k}$ be the contextual affinity between two associations indicated by $w^k_{\vartheta_k}$ and $w^k_{\varsigma_k}$ respectively. Embedding the contextual affinity into the temporal affinity in (13) yields a joint optimization which is a linear combination of the temporal affinity and the contextual affinity:

$$\max_{\mathbf{w}}\left(\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k=1}^{N^2}\cdots\sum_{\vartheta_K=1}^{N^2}s_{\vartheta_1\ldots\vartheta_k\ldots\vartheta_K}w^1_{\vartheta_1}\ldots w^k_{\vartheta_k}\ldots w^K_{\vartheta_K}+\alpha\sum_{k=1}^{K}\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2}c^k_{\vartheta_k\varsigma_k}w^k_{\vartheta_k}w^k_{\varsigma_k}\right), \tag{34}$$

where α is a weighting parameter which is used to balance the effects of the two affinities, and the second term models the contexts between associations. The optimization is also constrained by (14) as well as $0\le w^k_{\vartheta_k}\le 1$.

The new optimization in (34) is more difficult than the basic one in (13) due to the quadratic contextual term. We make some reformulations to (34) to make it solvable by iterations. From (14), it is apparent that

$$\frac{1}{N}\sum_{\varsigma_k=1}^{N^2}w^k_{\varsigma_k}=1,\ k=1,2,\ldots,K\ . \tag{35}$$

By using (35), the first term in (34) is rewritten as follows:

$$\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k}^{N^2}\cdots\sum_{\vartheta_{K=1}}^{N^2} s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K} w_{\vartheta_1}^1 \cdots w_{\vartheta_k}^k \cdots w_{\vartheta_K}^K$$
$$=\left(\frac{1}{N}\sum_{\varsigma_k=1}^{N^2} w_{\varsigma_k}^k\right)\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k=1}^{N^2}\cdots\sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K} w_{\vartheta_1}^1 \cdots w_{\vartheta_k}^k \cdots w_{\vartheta_K}^K \tag{36}$$
$$=\frac{1}{N}\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2}\cdots\sum_{\vartheta_{K=1}}^{N^2} s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K} w_{\vartheta_1}^1 \cdots w_{\vartheta_k}^k w_{\varsigma_k}^k \cdots w_{\vartheta_K}^K .$$

By using (35), the second term in (34) is rewritten as follows:

$$\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2} c_{\vartheta_k\xi_k}^k w_{\vartheta_k}^k w_{\varsigma_k}^k = \left(\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2} c_{\vartheta_k\varsigma_k}^k w_{\vartheta_k}^k w_{\varsigma_k}^k\right)\prod_{f\neq k}\left(\frac{1}{N}\sum_{\vartheta_f=1}^{N^2} w_{\vartheta_f}^f\right)$$
$$=\frac{1}{N^{K-1}}\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2}\cdots\sum_{\vartheta_K=1}^{N^2} c_{\vartheta_k\varsigma_k}^k w_{\vartheta_1}^1 \cdots w_{\vartheta_k}^k w_{\varsigma_k}^k \cdots w_{\vartheta_K}^K , \tag{37}$$

where $w_{\vartheta_k}^k$ is not included in the continued multiplication $\prod\limits_{f\neq k}$ because it already exists in the preceding term.

Merging (36) and (37), the optimization (34) is rewritten as:

$$\max\sum_{\vartheta_1=1}^{N^2}...\sum_{\vartheta_k=1}^{N^2}...\sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1...\vartheta_k...\vartheta_K} w_{\vartheta_1}^1 ...w_{\vartheta_k}^2 ...w_{\vartheta_K}^K + \alpha\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2} c_{\vartheta_k\varsigma_k}^k w_{\vartheta_k}^k w_{\varsigma_k}^k + \alpha\sum_{f\neq k}\sum_{\vartheta_f=1}^{N^2}\sum_{\varsigma_f=1}^{N^2} c_{\vartheta_f\varsigma_f}^f w_{\vartheta_f}^f w_{\varsigma_f}^f$$
$$=\max\frac{1}{N}\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2}\cdots\sum_{\vartheta_K=1}^{N^2}\left(s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K}+\frac{\alpha c_{\vartheta_k\varsigma_k}^k}{N^{K-2}}\right) w_{\vartheta_1}^1 \cdots w_{\vartheta_k}^k w_{\varsigma_k}^k \cdots w_{\vartheta_K}^K + \alpha\sum_{f\neq k}\sum_{\vartheta_f=1}^{N^2}\sum_{\varsigma_f=1}^{N^2} c_{\vartheta_f\varsigma_f}^f w_{\vartheta_f}^f w_{\varsigma_f}^f . \tag{38}$$

We apply the block update strategy [7] to optimize (38) iteratively. Namely, when the block variables in $\mathbf{w}^k$ are updated to yield a local optimization, other block variables $\{\mathbf{w}^f \mid f\neq k\}$ are fixed. In this way, the complicated optimization in the global space reduces to a simplified solution in a local space. The second term in the right hand of the equality sign in (38) can be omitted. Thus, the optimization (38) reduces to:

$$\max_{\mathbf{w}^k}\sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2}\cdots\sum_{\vartheta_K=1}^{N^2}\left(s_{\vartheta_1\cdots\vartheta_k\cdots\vartheta_K}+\frac{\alpha c_{\vartheta_k\varsigma_k}^k}{N^{K-2}}\right) w_{\vartheta_1}^1 \cdots w_{\vartheta_k}^k w_{\varsigma_k}^k \cdots w_{\vartheta_K}^K . \tag{39}$$

The problem in solving (39) is that $w_{\vartheta_k}^k$ and $w_{\varsigma_k}^k$ lie in the same block vector $\mathbf{w}^k$ and couple with each other. Namely, when $w_{\vartheta_k}^k$ is updated, $w_{\varsigma_k}^k$ cannot be fixed. To solve this problem, we decouple the interdependency between $w_{\vartheta_k}^k$ and $w_{\varsigma_k}^k$ to simplify the optimization. If two association hypotheses indicated by $w_{\vartheta_k}^k$ and $w_{\varsigma_k}^k$ share the same object in the *k*-1th sample set or in the *k*th sample set (i.e., $i_{k-1}=j_{k-1}$ or $i_k=j_k$, where the relation between j_{k-1} and ζ_k is defined as in (11)), then we set their contextual affinity $c_{\vartheta_k\varsigma_k}^k$ to 0. This is because one sample does not exist in two real associations between two sample sets. Then, we reformulate (39) as follows:

$$\max_{\mathbf{w}^k} \sum_{\vartheta_1=1}^{N^2} \cdots \sum_{\vartheta_k=1}^{N^2} \sum_{\{\varsigma_k : j_{k-1} \neq i_{k-1}\}} \cdots \sum_{\vartheta_K=1}^{N^2} \left(\frac{N}{N-1} s_{\vartheta_1 \cdots \vartheta_k \cdots \vartheta_K} + \frac{\alpha c^k_{\vartheta_k \varsigma_k}}{N^{K-2}} \right) w^1_{\vartheta_1} \cdots w^k_{\vartheta_k} w^k_{\varsigma_k} \cdots w^K_{\vartheta_K} \ . \tag{40}$$

Let $e_{l_1 \ldots l_k j_k \ldots l_K}$ be the element of a (*K*+1)-order augmented tensor, which is defined as:

$$e_{\vartheta_1 \ldots \vartheta_k \varsigma_k \ldots \vartheta_K} = \frac{N}{N-1} s_{\vartheta_1 \cdots \vartheta_k \cdots \vartheta_K} + \frac{\alpha c^k_{\vartheta_k \varsigma_k}}{N^{K-2}} \ . \tag{41}$$

Then, (40) is transformed to:

$$\begin{aligned} & \max_{\mathbf{w}^k} \sum_{\vartheta_1=1}^{N^2} \cdots \sum_{\vartheta_k=1}^{N^2} \sum_{\varsigma_k \neq \vartheta_k} \cdots \sum_{\vartheta_K=1}^{N^2} e_{\vartheta_1 \ldots \vartheta_k \varsigma_k \ldots \vartheta_K} w^1_{\vartheta_1} \cdots w^k_{\vartheta_k} w^k_{\varsigma_k} \cdots w^K_{\vartheta_K} \\ & = \sum_{i_k=1}^{N} \max_{\mathbf{w}^k} \sum_{\vartheta_1=1}^{N^2} \cdots \sum_{\vartheta_k=1}^{N^2} \sum_{\{\varsigma_k : j_{k-1} \neq i_{k-1}\}} \cdots \sum_{\vartheta_K=1}^{N^2} e_{\vartheta_1 \ldots \vartheta_k \varsigma_k \ldots \vartheta_K} w^1_{\vartheta_1} \cdots w^k_{\vartheta_k} w^k_{\varsigma_k} \cdots w^K_{\vartheta_K} \ . \end{aligned} \tag{42}$$

In (42), (40) is decomposed into a series of the following optimizations:

$$\max_{\mathbf{w}^k} \sum_{\vartheta_1=1}^{N^2} \cdots \sum_{\vartheta_k=1}^{N^2} \sum_{\{\varsigma_k : j_{k-1} \neq i_{k-1}\}} \cdots \sum_{\vartheta_K=1}^{N^2} e_{\vartheta_1 \ldots \vartheta_k \varsigma_k \ldots \vartheta_K} w^1_{\vartheta_1} \cdots w^k_{\vartheta_k} w^k_{\varsigma_k} \cdots w^K_{\vartheta_K} \ . \tag{43}$$

In (43), the interdependency between $w^k_{\vartheta_k}$ and $w^k_{\varsigma_k}$ is decoupled. The optimization in (43) has the same form with (13). Then, we can use the dual L_1 normalized tensor power iteration method in Section 3 to solve (43).

In actual calculation, it is not necessary to construct the (*K*+1)-order augmented tensor using (41). Instead we carry out the following iteration:

$$\begin{aligned} w^k_{\vartheta_k}(\tau+1) & \propto w^k_{\vartheta_k}(\tau) \sum_{\vartheta_1=1}^{N^2} \cdots \sum_{\{\vartheta_f | f \neq k\}} \cdots \sum_{\vartheta_K=1}^{N^2} \sum_{\{\varsigma_k | j_k \neq i_k\}} e_{\vartheta_1 \ldots \vartheta_k \varsigma_k \ldots \vartheta_K} w^k_{\varsigma_k} w^1_{\vartheta_1} \cdots w^k_{\vartheta_f} \cdots w^K_{\vartheta_K} \\ & \propto w^k_{\vartheta_k}(\tau) \sum_{\vartheta_1=1}^{N^2} \cdots \sum_{\{\vartheta_f | f \neq k\}} \cdots \sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1 \ldots \vartheta_k \ldots \vartheta_K} w^1_{\vartheta_1} \cdots w^k_{\vartheta_f} \cdots w^K_{\vartheta_K} + \sum_{\{\varsigma_k | j_{k-1} \neq i_{k-1}\}} c^k_{\vartheta_k \varsigma_k} w^k_{\varsigma_k} \ . \end{aligned} \tag{44}$$

It is seen that in the iteration the computation only involves the pairwise associations including the current set *k*. While the computational complexity of the iteration in the dual L_1 normalized tensor power iteration is $O(N^{2K})$, the computational complexity in (44) is $O(N^{2K} + N^2) = O(N^{2K})$ as $O(N^{2K}) \gg O(N^2)$. Considering the contexts only slightly increases the runtime. The dual L_1 normalized context-aware tensor power iteration is outlined as follows:

Input: temporal affinities $\{s_{\vartheta_1,\ldots,\vartheta_K}\}$, contexts $\{c^k_{\vartheta_k \varsigma_k}\}$, the maximum number *Γ* of iterations

Temporary variables: temporal affinity score $\phi^k_{i_{k-1},i_k}$ and context score $\psi^k_{i_{k-1},i_k}$

Output: associations $\{w^k_{\vartheta_k}\}$

```
$\tau$=1
While $\tau$<=$\Gamma$
    For $k$=1, 2, …, $K$ do
        For $i_{k-1}=1,2,...,N$ do
            For $i_k=1,2,...,N$ do
                $\varphi_{i_{k-1},i_k} = \sum_{\vartheta_1=1}^{N^2}\cdots\sum_{\{\vartheta_f|f\neq k\}}\cdots\sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1\ldots\vartheta_k\ldots\vartheta_K} w_{\vartheta_1}^1\cdots w_{\vartheta_f}^k\cdots w_{\vartheta_K}^K$ ;    (45)
                $\psi_{i_{k-1},i_k} = \sum_{\{\varsigma_k|j_{k-1}\neq i_{k-1}\}} c^k_{\vartheta_k\varsigma_k} w^k_{\varsigma_k}$ ;    (46)
            End For
            $\forall i_k, x^k_{i_{k-1},i_k} \leftarrow \frac{x^k_{i_{k-1},i_k}(\varphi_{i_{k-1},i_k}+\alpha\psi_{i_{k-1},i_k})}{\sum_{j_k=1}^{N} x^k_{i_{k-1},j_k}(\varphi_{i_{k-1},j_k}+\alpha\psi_{i_{k-1},j_k})}$ ;    (47)
        End For
        $\forall i_{k-1}, x^k_{i_{k-1},i_k} \leftarrow \frac{x^k_{i_{k-1},i_k}}{\sum_{j_{k-1}=1}^{N} x^k_{j_{k-1},i_k}}$ ;    (48)
    End For
    $\tau$=$\tau$+1;
End While
```

5. Hyper Context-Aware Power Iteration

We replace the pairwise contexts formulated in Section 4 with hyper-contexts among triples of associations. Suppose that $w^k_{\vartheta_k}$, $w^k_{\varsigma_k}$, $w^k_{\upsilon_k}$ represent, respectively, the associations on samples pairs $\mathbb{O}^{k-1}_{i_{k-1}} \leftrightarrow \mathbb{O}^k_{i_k}$, $\mathbb{O}^{k-1}_{j_{k-1}} \leftrightarrow \mathbb{O}^k_{j_k}$, and $\mathbb{O}^{k-1}_{l_{k-1}} \leftrightarrow \mathbb{O}^k_{l_k}$ between the k-1th sample set and the kth sample set. The affinity of the hyper-context among $w^k_{\vartheta_k}$, $w^k_{\varsigma_k}$, and $w^k_{\upsilon_k}$ is represented as $c^k_{\vartheta_k\varsigma_k\upsilon_k}$. By replacing the context affinity with the hyper-context affinity, the objective of hyper-context aware tensor approximation is extended as:

$$\max_{\mathbf{w}} \sum_{\vartheta_1=1}^{N^2}\ldots\sum_{\vartheta_k=1}^{N^2}\ldots\sum_{\vartheta_K=1}^{N^2} s_{\vartheta_1\ldots\vartheta_k\ldots\vartheta_K} w^1_{\vartheta_1}\ldots w^k_{\vartheta_k}\ldots w^K_{\vartheta_K} + \alpha\sum_{k=1}^{K}\sum_{\vartheta_k=1}^{N^2}\sum_{\varsigma_k=1}^{N^2}\sum_{\upsilon_k=1}^{N^2} c^k_{\vartheta_k\varsigma_k\upsilon_k} w^k_{\vartheta_k} w^k_{\varsigma_k} w^k_{\upsilon_k} . \tag{49}$$

Using tensor power iteration to solve the above optimization, we only need to replace (46) with

$$\psi_{i_{k-1},i_k} = \sum_{\{\varsigma_k|j_{k-1}\neq i_{k-1}\}}\sum_{\{\upsilon_k|l_{k-1}\neq i_{k-1}\}} c^k_{\vartheta_k\varsigma_k\upsilon_k} w^k_{\varsigma_k} w^k_{\upsilon_k} . \tag{50}$$

6. Multi-Object Tracking

Multi-object tracking can be carried out in a batch or online mode:

- **Batch mode:** For a batch of K+1 successive frames, N objects in each frame are detected, and N^2 association hypotheses are generated between two consecutive frames. Then, a K order tensor is constructed by computing the temporal affinity. The context affinities are computed. Based on these affinities, the dual L_1 norm context-aware tensor power iteration is applied to find the real associations between the detected objects. After that, the next batch of K+1 successive frames is processed in the same way as for the preceding batch, where the two batches share a common boundary frame. Serial expansion of the associations in all the batches in a video yields the global trajectories of the detected objects.
- **Online mode:** Given a new frame, it is combined with the preceding K frames. The dual L_1 normalized context-aware tensor power iteration is applied to find the object associations between these K+1 frames.

The main components of the multi-object tracking algorithm include association hypothesis generation, tensor construction, definition of the contextual affinity, and initialization and termination. An association hypothesis between two objects from two consecutive frames respectively is generated only when they are spatially close to each other. Removing unnecessary association hypotheses in this way greatly reduces the computational complexity and storage space. In a scene, moving objects may enter, exit, reappear, be occluded, or be missed, etc. There may be detections that have no associated detection in consecutive frames. To handle this issue, virtual detections are introduced in each frame to drop out the isolated detections and thus avoid disturbing the real detection associations. Appearance models and motion models are used to construct the temporal affinity. The contextual affinity is defined using motion contexts. The context-aware tensor power iteration is applied to obtain the real valued associations between detections. The real valued solutions must be discretized to meet the integer and one-to-one mapping constraints in the assignment. We regard the real valued solutions as the costs for the corresponding associations, and apply the Hungarian algorithm to obtain the binary association outputs. In the following, we detail tensor construction, definition of the contextual affinity, and the computational complexity analysis.

6.1. Tensor construction

The form of the elements $s_{\vartheta_1 \cdots \vartheta_k \cdots \vartheta_K}$ in (34) depends on the application and the particular temporal affinities.

In some applications the temporal affinities are based on motion models while in other applications it is necessary to combine appearance models and motion models to define the temporal affinity. Two methods for defining temporal affinities are described.

The first method defines the temporal affinity of a trajectory based on a global motion affinity $m_{\vartheta_1\vartheta_2\ldots\vartheta_K}$ and appearance affinities given by associations between consecutive frames. Let $\mathbf{z}^k_{\vartheta_k} = \overrightarrow{\mathbb{O}^{k-1}_{i_{k-1}}\mathbb{O}^k_{i_k}}$ be the spatial displacement of the association between objects $\mathbb{O}^{k-1}_{i_{k-1}}$ at frame k-1 and $\mathbb{O}^k_{i_k}$ at frame k. If objects $\mathbb{O}^{k-1}_{i_{k-1}}$ and $\mathbb{O}^k_{i_k}$ are the same object, $\mathbf{z}^k_{\vartheta_k}$ is the velocity vector of the object. The global motion affinity is defined as:

$$m_{\vartheta_1\vartheta_2\ldots\vartheta_K} = \prod_{k=1}^{K-1}\exp\left(\frac{(\mathbf{z}^k_{\vartheta_k})^T\mathbf{z}^{k+1}_{\vartheta_{k+1}}}{\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2\left\|\mathbf{z}^{k+1}_{\vartheta_{k+1}}\right\|_2} + \frac{2\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2\left\|\mathbf{z}^{k+1}_{\vartheta_{k+1}}\right\|_2}{\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2^2 + \left\|\mathbf{z}^{k+1}_{\vartheta_{k+1}}\right\|_2^2}\right), \tag{51}$$

where the first part in the exponential term is the cosine of two velocity vectors measuring their direction consistency and the second part measures their amplitude consistency. This motion affinity describes object motion inertia: an object has similar velocities in consecutive frames. For an object $\mathbb{O}^k_{i_k}$, we use a gray scale histogram $\{h^1_{i_k}, h^2_{i_k}, \ldots\}$ and the area $b^k_{i_k}$ of the box bounding the object to represent its appearance. We define the appearance affinity $a^k_{\vartheta_k}$ of an association $(\mathbb{O}^{k-1}_{i_{k-1}}, \mathbb{O}^k_{i_k})$ as follows:

$$a^k_{\vartheta_k} = \frac{1}{2}\sum_{bin}\min\left(h^{bin}_{i_k}, h^{bin}_{i_{k-1}}\right) + \frac{1}{2}\min\left(\frac{b^{k-1}_{i_{k-1}}}{b^k_{i_k}}, \frac{b^k_{i_k}}{b^{k-1}_{i_{k-1}}}\right). \tag{52}$$

The two terms in the right hand of the equal sign represent the similarities of appearance and area respectively. Combining (51) and (52), the temporal affinity of a trajectory is defined as follows:

$$s_{\vartheta_1\vartheta_2\ldots\vartheta_K} = a^1_{\vartheta_1}a^2_{\vartheta_2}\cdots a^K_{\vartheta_K}m_{\vartheta_1\vartheta_2\ldots\vartheta_K}\,. \tag{53}$$

The second method that we use for defining the tensors is taken from [7]. The temporal affinity is:

$$s_{\vartheta_1\vartheta_2\ldots\vartheta_K} = E_0 - \eta\sum_{k=1}^{K}\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2 - \sum_{k=1}^{K-1}\left\|\mathbf{z}^{k+1}_{\vartheta_{k+1}} - \mathbf{z}^k_{\vartheta_k}\right\|_2\,, \tag{54}$$

where E_0 is a constant used to make the affinity positive and η is a weighting parameter. The second term on the right hand side of (54) penalizes large position translation for any association and the third term penalizes changes in velocity between consecutive associations. The intuition is that the changes in velocity and the spatial translation of the same object between consecutive frames are not large.

6.2. Motion contexts

Motion contexts are used to define the contextual affinity $c^k_{\vartheta_k \varsigma_k}$ between two associations ϑ_k and ζ_k. Low-level contexts and high-level contexts are proposed to represent interactions between associations on detected objects and interactions between trajectory segments, respectively. When between-frame motions are large, low-level context is valuable, such as in the low-frame rate or fast motion applications. In pedestrian tracking, inaccurately located object detections along with low-speed motion make raw detection-based low-level context unreliable. High level interactions are more reliable.

6.2.1. Low level context

We formulate the interaction between two associated detected objects using the motion consistency of the objects. Let $\mathbf{z}^k_{\vartheta_k}$ and $\mathbf{z}^k_{\varsigma_k}$ be the spatial displacement vectors for the association hypotheses $(\mathbb{O}^{k-1}_{i_{k-1}}, \mathbb{O}^k_{i_k})$ and $(\mathbb{O}^{k-1}_{j_{k-1}}, \mathbb{O}^k_{j_k})$ respectively. The motion consistency $m^k_{\vartheta_k \varsigma_k}$ between these two association hypotheses is defined as a linear combination of the orientation similarity and the speed similarity between the motion vectors $\mathbf{z}^k_{\vartheta_k}$ and $\mathbf{z}^k_{\varsigma_k}$:

$$m^k_{\vartheta_k \varsigma_k} = \frac{\left|(\mathbf{z}^k_{\vartheta_k})^T \mathbf{z}^k_{\varsigma_k}\right|}{\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2 \left\|\mathbf{z}^k_{\varsigma_k}\right\|_2} + \lambda \frac{\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2 \left\|\mathbf{z}^k_{\varsigma_k}\right\|_2}{\left\|\mathbf{z}^k_{\vartheta_k}\right\|_2^2 + \left\|\mathbf{z}^k_{\varsigma_k}\right\|_2^2}, \tag{55}$$

where λ is a weighting parameter balancing the orientation similarity (the first part to the right of the equality sign) and the speed similarity (the second part to the right of the equality sign).

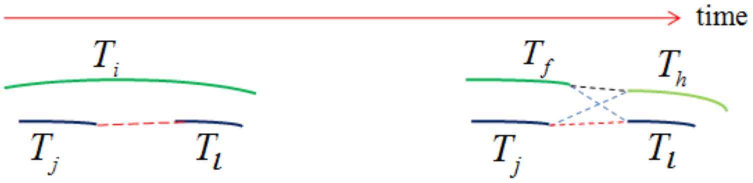
It is only necessary to model interactions between associations of objects with similar motions in local spatial neighborhoods. Furthermore, associations with contexts cannot share the same object in the same frame, because one object only belongs to one real association between two frames. We define the low-level motion context as a selective representation. The context $c^k_{\vartheta_k \varsigma_k}$ between associations $(\mathbb{O}^{k-1}_{i_{k-1}}, \mathbb{O}^k_{i_k})$ and $(\mathbb{O}^{k-1}_{j_{k-1}}, \mathbb{O}^k_{j_k})$ is set to their motion consistency $m^k_{\vartheta_k \varsigma_k}$ only if they satisfy the following three conditions (otherwise $c^k_{\vartheta_k \varsigma_k}$ is set to 0):

- $i_{k-1} \neq j_{k-1}$ and $i_k \neq j_k$,
- $\left\|\overrightarrow{\mathbb{O}^{k-1}_{i_{k-1}} \mathbb{O}^{k-1}_{j_{k-1}}}\right\|_2 < L$ and $\left\|\overrightarrow{\mathbb{O}^k_{i_k} \mathbb{O}^k_{j_k}}\right\|_2 < L$ where L is a distance threshold,
- $j_k = \max_{\tilde{j}_k} m^k_{(i_{k\text{-}1}, i_k), (j_{k-1}, \tilde{j}_k)}$.

The first condition is a one-to-one mapping constraint. The second is a spatial distance mask. The third selects the association (j_{k-1}, j_k) most similar to the association (i_{k-1}, i_k) from the associations including object j_{k-1}. This sparse representation of non-maximum removal is used to bind the most similar associations as contexts and suppress the influences from noisy and conflicting association pairs.

6.2.2. High level context

We devise two types of high level contexts to model the motion interaction on associations between tracklets (trajectory segments). Fig. 2 shows the two types of high level contexts. The first context, as shown in Fig. 2(a), includes the interactions between two associated tracklets and a tracklet. The second one includes the interactions between two associations of tracklets.



(a) (b)

Fig. 2. High-level motion contexts: (a) Interaction between tracklet association (T_j, T_l) and tracklet T_i; (b) Interaction between two tracklet associations (T_f, T_h) and (T_j, T_l).

When two tracklets T_j and T_l are associated with the motion T_i of an object as shown in Fig. 2(a), it is more likely that there is a true association between T_j and T_l. We use this prior knowledge to measure contextual affinity between T_j and T_l. Suppose that the *i*-th tracklet T_i is represented by $\{\mathbb{O}_i^{t_s^i}, \mathbb{O}_i^{t_s^i+1}, ..., \mathbb{O}_i^{t_e^i}\}$, where t_s^i and t_e^i denote, respectively, the start time and the end time of T_i. Let $\mathbf{z}_i^t$ be the spatial displacement from the target $\mathbb{O}_i^{t-1}$ to $\mathbb{O}_i^t$. For two tracklets $T_j : \{\mathbb{O}_j^{t_s^j}, ..., \mathbb{O}_j^{t_e^j}\}$ and $T_l : \{\mathbb{O}_l^{t_s^l}, ..., \mathbb{O}_l^{t_e^l}\}$, there exists association hypothesis $T_{jl} : \{\mathbb{O}_j^{t_s^j}, ..., \mathbb{O}_j^{t_e^j}, \mathbb{O}_{jl}^{t_e^j+1}, ..., \mathbb{O}_{jl}^{t_s^l-1}, \mathbb{O}_l^{t_s^l}, ..., \mathbb{O}_l^{t_e^l}\}$, where $\mathbb{O}_{jl}^t (t_e^j < t < t_s^l)$ is the virtual object interpolated using T_j and T_l. Then, the motion consistency $m_{jl,i}$ between association hypothesis T_{jl} and tracklet T_i is defined as their motion orientation similarity:

$$m_{jl,i} = \frac{1}{t_s^l - t_e^j} \sum_{t=t_e^j}^{t_s^l} \frac{\left| (\mathbf{z}_{jl}^t)^T \mathbf{z}_i^t \right|}{\left\| \mathbf{z}_{jl}^t \right\|_2 \left\| \mathbf{z}_i^t \right\|_2}, \tag{56}$$

where $\mathbf{z}_{jl}^t$ is the spatial displacement from target $\mathbb{O}_{jl}^{t-1}$ to $\mathbb{O}_{jl}^t$.

We select the spatial neighboring tracklets around T_{jl} and overlapped with T_j and T_l in the time window and use them to measure the contexts for motion consistency between T_j and T_l. These spatial neighboring tracklets $\{T_c\}_c$ satisfy the following conditions:

- $t_s^c \leq t_e^j$ and $t_s^l \leq t_e^c$,
- $\left\|\overrightarrow{\mathcal{O}_c^{t_e^j}\mathcal{O}_j^{t_e^j}}\right\|_2 < L$ and $\left\|\overrightarrow{\mathcal{O}_c^{t_s^l}\mathcal{O}_l^{t_s^l}}\right\|_2 < L$.

Let *C* be the number of tracklets in $\{T_c\}_c$. The motion context for the tracklet pair T_{jl} is estimated by:

$$c_{jl} = \frac{1}{C}\sum_{c=1}^{C} m_{jl,c} \,. \tag{57}$$

The term c_{jl} is the contextual affinity between T_j and T_l.

The second type of context, shown in Fig. 2(b), measures the motion interactions between tracklet association hypotheses. We define these motion contexts based on motion consistency between the association hypotheses. Suppose that association hypothesis T_{jl} connects tracklets T_j and T_l, and association hypothesis T_{fh} connects tracklets T_f and T_h, as shown in Fig. 2(b). Let $t_e^{jf} = \max\{t_e^j, t_e^f\}$ and $t_s^{lh} = \min\{t_s^l, t_s^h\}$. The motion consistency $m_{jl,fh}$ between T_{jl} and T_{fh} is estimated by:

$$m_{jl,fh} = \frac{1}{t_s^{lh} - t_e^{jf}} \sum_{t=t_e^{jf}+1}^{t_s^{lh}} \frac{\left|(\mathbf{z}_{jl}^t)^T \mathbf{z}_{fh}^t\right|}{\left\|\mathbf{z}_{jl}^t\right\|_2 \left\|\mathbf{z}_{fh}^t\right\|_2}, \tag{58}$$

where $\mathbf{z}_{jl}^t$ ($\mathbf{z}_{fh}^t$) is the spatial displacement from object $\mathcal{O}_{jl}^{t-1}(\mathcal{O}_{fh}^{t-1})$ to $\mathcal{O}_{jl}^t(\mathcal{O}_{fh}^t)$.

The context $c_{jl,fh}$ between T_{jl} and T_{fh} is set to their motion consistency $m_{jl,fh}$ only if the following conditions are satisfied (otherwise $c_{jl,fh}$ is set to 0):

- $t_e^{jf} < t_s^{lh}$,
- $j \neq f$ and $l \neq h$,
- $\left\|\overrightarrow{\mathcal{O}_j^{t_e^{jf}}\mathcal{O}_f^{t_e^{jf}}}\right\|_2 < L$ and $\left\|\overrightarrow{\mathcal{O}_h^{t_s^{lh}}\mathcal{O}_l^{t_s^{lh}}}\right\|_2 < L$.

6.3. Computational complexity

The computational complexity of our multi-object tracking method depends on the number of high-order trajectory hypotheses. An association hypothesis between two objects from consecutive frames respectively is made only when they are spatially close to each other. For a set of *K*+1 consecutive frames, each frame has *N* objects and every object has I association candidates in the next frame. Then, there are NM^K trajectory hypotheses. For the maximum number Γ of iterations, the computational complexity of one set association is $O(\Gamma K^2 NI^K)$.

7. Multi-Graph Matching

We apply the proposed dual L_1 normalized context/hyper-context aware tensor power iteration algorithm to multi-graph matching. A graph $\mathbb{G}$ is represented as $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{A})$, where $\mathbb{V}$ is the vertex set, $\mathbb{E}$ is the edge set, and $\mathbb{A}$ is the attribute set. The attribute set $\mathbb{A}$ includes the vertex features such as the position, appearance information, as well as the edge properties such as the distance and orientation. Given K+1 graphs $\{\mathbb{G}^k = (\mathbb{V}^k, \mathbb{E}^k, \mathbb{A}^k)\}_{k=0}^K$ with the same number N of vertices, the task of multi-graph matching is to find an optimal one-to-one correspondence between the vertices in the K+1 graphs respectively. The solution is denoted by a K+1th order assignment tensor. An element $x_{i_0 i_1 \ldots i_K}$ in the tensor denotes the group-wise matching between the vertices $(v_{i_k}^k \in \mathbb{V}^k)_{k=0}^K$ in turn. If the matching is true, then $x_{i_0 i_1 \ldots i_K} = 1$, otherwise $x_{i_0 i_1 \ldots i_K} = 0$. The vertex affinity of the group-wise matching is denoted by $a_{i_0 i_1 \ldots i_K}$. The structural affinity over the edge set $\{e_{i_0 j_0}^0, e_{i_1 j_1}^1, \cdots e_{i_K j_K}^K\}$ between vertex correspondences $x_{i_0 i_1 \ldots i_K}$ and $x_{j_0 j_1 \ldots j_K}$ is denoted as $s_{\mathbb{E}}(e_{i_0 j_0}^0, e_{i_1 j_1}^1, \cdots, e_{i_K j_K}^K)$. The objective function for multi-graph matching is formulated as:

$$\max \sum_{i_0=1}^{N} \ldots \sum_{i_k=1}^{N} \ldots \sum_{i_K=1}^{N} a_{i_0 \ldots i_k \ldots i_K} x_{i_0}^0 \ldots x_{i_k}^k \ldots x_{i_K}^K + \alpha \sum_{i_0=1}^{N} \cdots \sum_{i_K=1}^{N} \sum_{j_0=1}^{N} \cdots \sum_{j_K=1}^{N} x_{i_0}^0 \ldots x_{i_K}^K s_E(e_{i_0 j_0}^0, \cdots, e_{i_K j_K}^K) x_{j_0}^0 \ldots x_{j_K}^K . \tag{59}$$

We transform the optimization (59) into the optimization in (34) or the optimization in (49). The high-order matching $x_{i_0 i_1 \ldots i_K}$ is decomposed into pairwise ones: $x_{i_0 i_1 \ldots i_K} = x_{i_0, i_1}^1 x_{i_1, i_2}^2 \ldots x_{i_{K-1}, i_K}^K$, where x_{i_{k-1}, i_k}^k is an element in the assignment vector $\mathbf{x}^k$ for matching graphs $\mathbb{G}^{k-1}$ and $\mathbb{G}^k$. The group-wise edge affinity $s_{\mathbb{E}}(e_{i_0 j_0}^0, e_{i_1 j_1}^1, \cdots, e_{i_K j_K}^K)$ is decomposed into pairwise edge affinities as follows:

$$s_E(e_{i_0 j_0}^0, e_{i_1 j_1}^1, \cdots, e_{i_K j_K}^K) \approx \sum_{k=1}^{K} c_{i_{k-1} i_k, j_{k-1} j_k}^k , \tag{60}$$

where $c_{i_{k-1} i_k, j_{k-1} j_k}^k$ is the pairwise similarity between edge $e_{i_{k-1} j_{k-1}}^{k-1}$ in $\mathbb{G}^{k-1}$ and edge $e_{i_k j_k}^k$ in $\mathbb{G}^k$. As shown in (10), matrix index elements (i_{k-1}, i_k) and (j_{k-1}, j_k) are transformed to vector index elements ϑ_k and ζ_k. Substitution of (60) into the term after α in (59) yields:

$$\sum_{k=1}^{K} \sum_{\vartheta_k=1}^{N^2} \sum_{\varsigma_k=1}^{N^2} c_{\vartheta_k \varsigma_k}^k w_{\vartheta_k}^k w_{\varsigma_k}^k . \tag{61}$$

In this way, multi-graph matching problem is transformed into the dual L_1 normalized context-aware tensor power iteration problem in Section 4.

For multiple hyper-graph matching, hyper-edges are considered based on vertex triples. Let $c^k_{i_k j_k l_k}$ be the affinity between hyper-edges $e^{k-1}_{i_{k-1}j_{k-1}l_{k-1}}$ and $e^k_{i_k j_k l_k}$ corresponding to vertex triples $\{v^{k-1}_{i_{k-1}}, v^{k-1}_{j_{k-1}}, v^{k-1}_{l_{k-1}}\}$ and $\{v^k_{i_k}, v^k_{j_k}, v^k_{l_k}\}$ respectively. We decompose the group-wise hyper-edge affinity $s_{\mathbb{E}}(e^0_{i_0 j_0 l_0}, e^1_{i_1 j_1 l_1}, \cdots, e^K_{i_K j_K l_K})$ into pairwise hyper-edge affinities as follows:

$$s_E(e^0_{i_0 j_0 l_0}, e^1_{i_1 j_1 l_1}, \cdots, e^K_{i_K j_K l_K}) \approx \sum_{k=1}^{K} c^k_{i_{k-1}i_k, j_{k-1}j_k, l_{k-1}l_k} \; . \tag{62}$$

Substitution of (62) into the term after α in (59) yields the hyper context-aware tensor power iteration problem in Section 5.

The vertex affinity used in the optimization is defined by considering the local appearance similarity between vertices that correspond to image feature points. Each vertex is associated with a shape context feature vector [28] of a feature point in an image. Let $\mathbf{y}^k_{i_k}$ be the column feature vector of vertex $v^k_{i_k}$. All the feature vectors from the vertex set $\{v^0_{i_0}, v^1_{i_1}, \cdots, v^K_{i_K}\}$ are stacked into a matrix $\mathbf{Y}_{i_0 i_1 \ldots i_K} = \{\mathbf{y}^0_{i_0}, \mathbf{y}^1_{i_1}, \cdots, \mathbf{y}^K_{i_K}\}$. Let $eigen(\mathbf{Y}_{i_0 i_1 \ldots i_K}, d)$ be the d-th eigenvalue of the matrix $\mathbf{Y}_{i_0 i_1 \ldots i_K}$ when the eigenvalues are ranked in descending order. The high-order vertex affinity $a_{i_0 i_1 \ldots i_K}$ is computed as:

$$a_{i_0 i_1 \ldots i_K} = \frac{eigen(\mathbf{Y}_{i_0 i_1 \ldots i_K}, 1)}{\sum_d eigen(\mathbf{Y}_{i_0 i_1 \ldots i_K}, d)} \; . \tag{63}$$

This vertex affinity measures the compactness of the feature vector set.

We define the contextual affinity $c^k_{\vartheta_k \varsigma_k \upsilon_k}$ by considering the difference in angles formed by hyperedges as hyper-contexts. Let $w^k_{\vartheta_k}$ represent the matching between vertices $v^{k-1}_{i_{k-1}}$ and $v^k_{i_k}$. Let $w^k_{\varsigma_k}$ be the matching between vertices $v^{k-1}_{j_{k-1}}$ and $v^k_{j_k}$. Let $w^k_{\upsilon_k}$ be the matching between vertices $v^{k-1}_{l_{k-1}}$ and $v^k_{l_k}$. The hyper-edge on the vertex triple $\{v^{k-1}_{i_{k-1}}, v^{k-1}_{j_{k-1}}, v^{k-1}_{l_{k-1}}\}$ in $\mathbb{G}^{k-1}$ forms a triangle whose three angles $\{\theta^{k-1}_1, \theta^{k-1}_2, \theta^{k-1}_3\}$ are used as the features for this hyper-edge. The hyper-edge on the vertex triple $\{v^k_{i_k}, v^k_{j_k}, v^k_{l_k}\}$ in $\mathbb{G}^k$ has its triangle features $\{\theta^k_1, \theta^k_2, \theta^k_3\}$. The triangle structure of the hyper-edge is invariant to rotation and scaling. The relation between hyper-edges and triangle features is illustrated in Fig. 3. The hyper-edge affinity $c^k_{\vartheta_k \varsigma_k \upsilon_k}$ [29, 30] is defined as:

$$c^k_{\vartheta_k \varsigma_k \upsilon_k} = \exp\left(-\frac{\sum_{h=1}^{3} (\sin\theta_h^k - \sin\theta_h^{k-1})^2}{2\sigma^2} \right), \tag{64}$$

where σ^2 is a regularized factor.

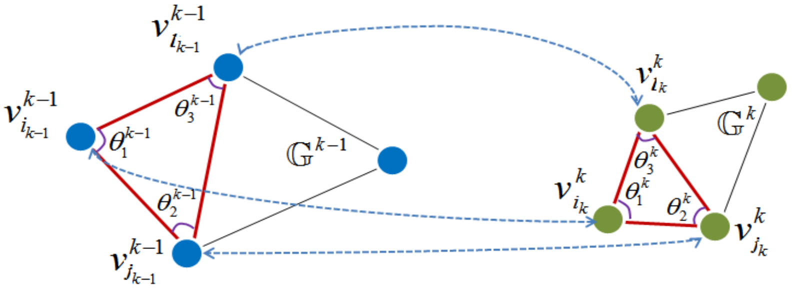


Fig. 3. Hyper-edges and their triangle features.

Using the hyper-context tensor power iteration solution, a sequence of real-valued pairwise matching vectors $\{\mathbf{w}^k\}_{k=1}^K$ is obtained. The real-valued matching matrix is further discretized using the Hungarian algorithm. Given the pairwise matching vectors $\{\mathbf{w}^k\}_{k=1}^K$, the group-wise matching $\{x_{i_0 i_1 \cdots i_K}\}$ is derived naturally.

The computational complexity for our multi-graph matching method depends on the number of hyper-edge triples. The number of group-wise matches grows with the number of graphs. For K+1 graphs, each graph has N vertices and each vertex has I matching candidates. There are NI^K matches and $K(NI)^3$ hyper-edge triples. In this way, the computational complexity is $O(\Gamma NI^K K^2 + \Gamma K(NI)^3)$ for Γ iterations. A divide-and-conquer strategy can be used for acceleration.

8. Experimental Results

The experimental results for multi-object tracking are shown first (*Please see the supplemental videos*), followed by the experimental results for multi-hyper-graph matching.

8.1. Multi-object tracking

We evaluated the proposed multi-object tracking methods on the following five public datasets: Columbus Large Image Format (CLIF) [22], PSU-data [10], PETS 2009, TUD-Stadtmitte, and MOT16 challenge.

8.1.1. CLIF and PSU

The CLIF and PSU datasets contain low frame-rate sequences, on which the proposed low-level motion context was used. Three CLIF sequences, CLIF1, CLIF2, and CLIF3, were used. There are 80~100 objects in each frame in CLIF1 and CLIF2, and 150~200 objects in CLIF3. These sequences are very challenging, because of fast motions, a large number of objects, small apparent sizes of objects, and similar object appearances, etc. The PSU dataset consists of three sparse sequences, “Sparse-1”, “Sparse-2”, and “Sparse-3” with 3∼5 people per frame and three dense sequences, “Dense-1”, “Dense-2”, and “Dense-3” with 25~20 people per frame. These sequences are challenging because of the large spatial displacement per frame and the unavailability of object appearance information.

We compared our methods with the following four multi-object tracking methods:

- **Hungarian assignment algorithm:** This is an optimal solution for each association of two successive frames (i.e., each single pair of successive frames). The association among K+1 successive frames is carried out by K pairwise associations, each of which is determined by the Hungarian assignment algorithm. This comparison is used to show the effect of multi-frame associations in contrast with two frame associations.
- **Network flow algorithm [17]:** As stated in Section 3.3, when the temporal affinity is computed as the product of pairwise affinities (i.e., affinities between two successive frames), the tensor rank-1 approximation-based algorithm reduces to the network flow-based algorithm which is a pairwise association-based method. This comparison is used to show the effect of high-order affinity representation in contrast with two frame association-based affinity representation.
- **Min-cost flow algorithm [6]:** This framework incorporates the high-order constraints in three consecutive frames for multi-object tracking by using candidate matching pairs. The framework is solved efficiently through Lagrangian relaxation to min-cost network flow. This comparison is used to show the effect of utilizing higher order information in our tensor power iteration.
- **Iterated Conditional Modes (ICM)-like method [7]:** This ICM-like method is similar to ours in that it is multi-assignment-based and uses a global affinity representation and a block update strategy. The difference is that in each iteration step the ICM-like method uses the Hungarian algorithm to yield binary object association relations.

We used the correct matching percentage and the false matching percentage to evaluate the association performance. Let $cm(t)$, $wm(t)$, and $g(t)$ be, respectively, the numbers of correct associations, false associations, and ground truth associations between frame t-1 and frame t. The correct match percentage P_c and false matching percentage P_f are defined as:

$$\begin{cases} P_c = 100 \times \dfrac{\sum_t cm(t)}{\sum_t g(t)}, \\ P_f = 100 \times \dfrac{\sum_t wm(t)}{\sum_t g(t)}. \end{cases} \tag{65}$$

Because the sum of the numbers of the correct and false associations is not necessarily equal to the number of ground truth associations, the sum of P_c and P_f is not necessarily equal to 1.

As there are no training samples on the CLIF dataset and the PSU dataset, we randomly selected 12 consecutive frames from each of these two datasets to tune the parameters. The number of frames in a batch for association determination was taken to 5 and 6 for the CLIF and PSU datasets respectively. On the CLIF dataset, the temporal affinity defined in (53) was used. On the PSU dataset, the temporal affinity defined in (54) was used, and η in (54) was set to 0.5. The parameter α in (34) was set to 10 and 5 for the CLIF and PSU datasets respectively. A larger α was used on the CLIF dataset, since the object motions are better-regulated in the CLIF scenarios and modeling motion interaction contexts is more important. The parameter λ in (55) was 0.6 and 2.0 for the CLIF and PSU datasets respectively. A smaller λ was used on the CLIF dataset, since the orientation consistency is more remarkable in the CLIF scenarios due to path constraints. The number of iterations was set to 100 for all the sequences, because in all the experiments, when 100 iterations were reached, the dual L_1-normalized rank-1 tensor approximation method and the dual L_1-normalized context-aware tensor power iteration method both showed convergence.

Table 1. Comparison results on the CLIF dataset

Performance / Method	Correct matching percentage			False matching percentage		
	CLIF-1	CLIF-2	CLIF-3	CLIF-1	CLIF-2	CLIF-3
Hungarian	77.8	88.9	85.9	22.0	11.6	14.2
Network flow	65.4	71.6	74.6	34.1	28.1	25.7
ICM	83.1	89.6	87.3	16.9	10.3	12.9
Tensor approximation	91.1	92.1	91.4	11.9	9.4	9.4
Context power iteration	94.7	96.0	95.8	6.0	4.8	4.1

Table 2. Comparison results on the sparse scene of the PSU dataset

Performance / Method	Correct matching percentage			False matching percentage		
	Sparse-1	Sparse-2	Sparse-3	Sparse-1	Sparse-2	Sparse-3
Network flow [17]	94.57	99.72	99.96	5.43	0.28	0.04
Hungarian	98.84	99.97	99.97	1.03	0.01	0.00
ICM	98.87	99.97	99.95	0.97	0.01	0.00
Min-cost flow [6]	-	-	-	0.41	0.00	0.00
Tensor approximation	99.45	99.98	99.99	0.50	0.00	0.00
Context power iteration	99.74	99.98	99.99	0.24	0.00	0.00

Table 1 shows the quantitative comparison results on the CLIF dataset. Tables 2 and 3 show the quantitative comparison results on the sparse and dense scenes of the PSU dataset, respectively, where the results for the network flow method [17] were taken from [7]. From these tables, the following points are revealed:

- Our rank-1 tensor approximation method and our dual L_1 normalized context-aware tensor power iteration method perform better than the ICM-like method. One reason is that the association probability is retained in the iteration process in our methods till the final decision.
- Our dual L_1 normalized context-aware tensor power iteration method performs better than our rank-1 tensor approximation method. In particular, on the CLIF dataset both P_c and P_f are improved, and P_f has a remarkable relative decrease. This demonstrates that the proposed motion contexts and their solution are effective. The motion context is very useful for reducing the association ambiguity, as the decision of the local association is influenced by not only its temporal coherence on the whole trajectory, but also its spatial interaction with other associations. Though the performances of the rank-1 tensor approximation method on the PSU dataset are close to saturation, yet the embedding of the proposed motion context improves the results remarkably.
- Our dual L_1 normalized context-aware method improves the false matching rate more prominently than the correct matching rate. This is because motion consistency contexts reduce the uncertainty of associations, which directly reduces the false matches and thereby indirectly increases the correct matching rate.
- The min-cost flow [6] has excellent performance on the PSU dataset, but our dual L_1 normalized context-aware tensor power iteration method yields better results.

Table 3. Comparison results on the dense scene of the PSU dataset

Performance / Method	Correct matching percentage			False matching percentage		
	Dense-1	Dense-2	Dense-3	Dense-1	Dense-2	Dense-3
Network flow [17]	78.65	98.64	99.77	21.35	1.36	0.23
Hungarian	92.40	99.62	99.86	7.37	0.35	0.11
ICM	93.63	99.74	99.91	6.26	0.24	0.08
Min-cost flow [6]	-	-	-	1.46	0.17	0.10
Tensor approximation	96.98	99.78	99.94	3.01	0.20	0.05
Context power iteration	98.41	99.88	99.94	1.58	0.11	0.05

- On more challenging sequences, our methods yield a larger increase of performance than on less challenging sequences.

- The two frame associations-based multi-object tracking methods, the Hungarian assignment algorithm and the network flow algorithm, perform much worse than the rank-1 tensor approximation method and the dual L_1 normalized context-aware tensor power iteration method. On the CLIF dataset and the dense scene of the PSU dataset, our methods yield much higher correct matching percentages and much lower false matching percentages than the Hungarian assignment algorithm and the network flow algorithm. This indicates that it is very effective to utilize the global temporal affinity and capture high-order motion, instead of utilizing only the pairwise association affinity.

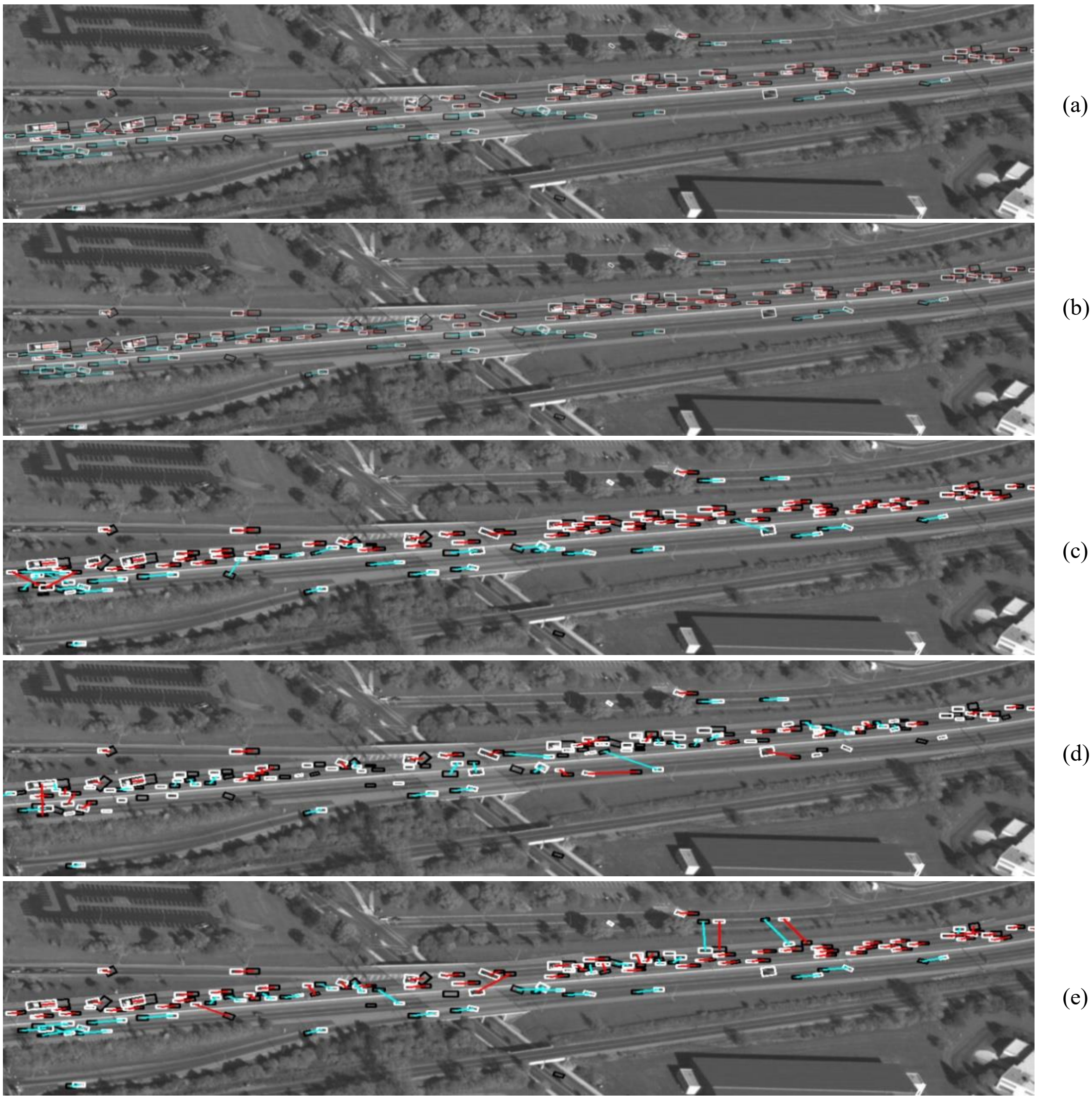


Fig. 4. Results of multi-object association for different methods on the CLIF-3 sequence: (a) Dual L_1 normalized context-aware tensor power iteration (with 2 mismatches); (b) Rank-1 tensor approximation (with 8 mismatches); (c) ICM-like association (with 13 mismatches); (d) Hungarian association (with 30 mismatches); (e) Network flow (with 26 mismatches). White/black rectangles: vehicle detections in the current/last frame; Red/green lines: associations on two orientations.

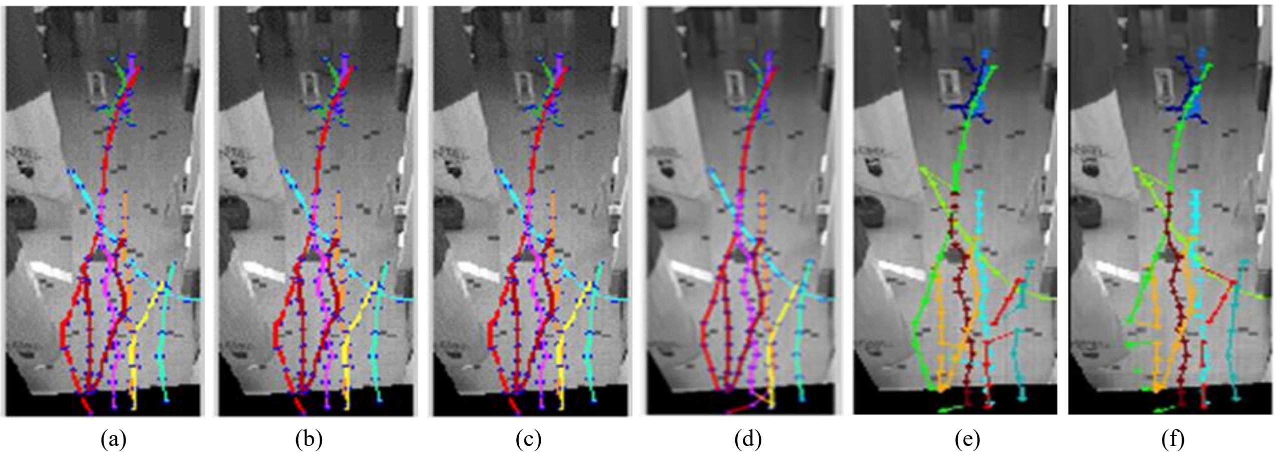


Fig. 5. Multi-object association results of different methods on 360 example frames in the Sparse-1 sequence: (a) Dual L_1 normalized context-aware tensor power iteration (with 0 mismatches); (b) Rank-1 tensor approximation (with 0 mismatches); (c) Min-cost flow (with 0 mismatches) (d) ICM-like (with 1 mismatches); (e) Hungarian association (with 7 mismatches); (f) Network flow (with 6 mismatches).

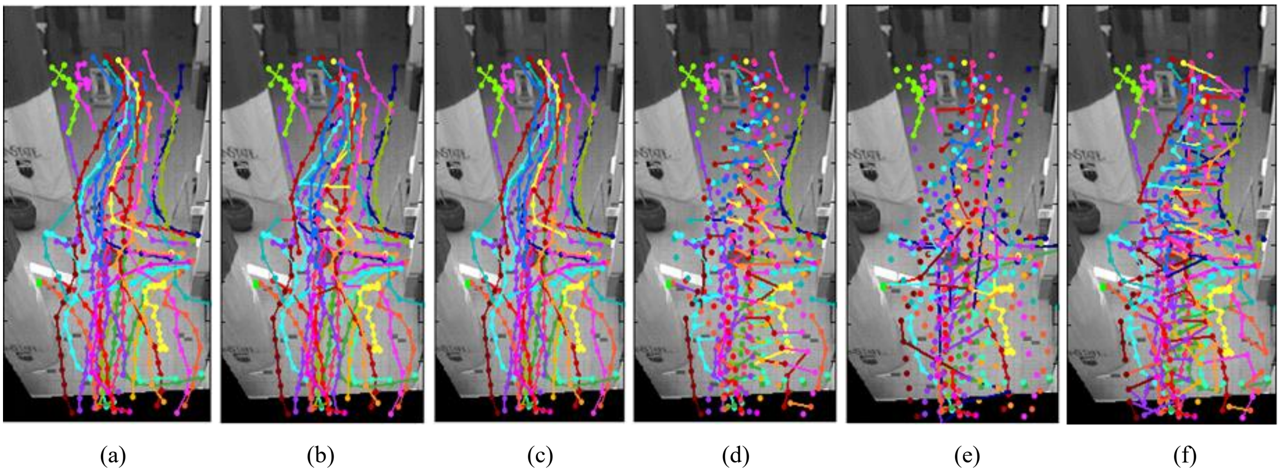
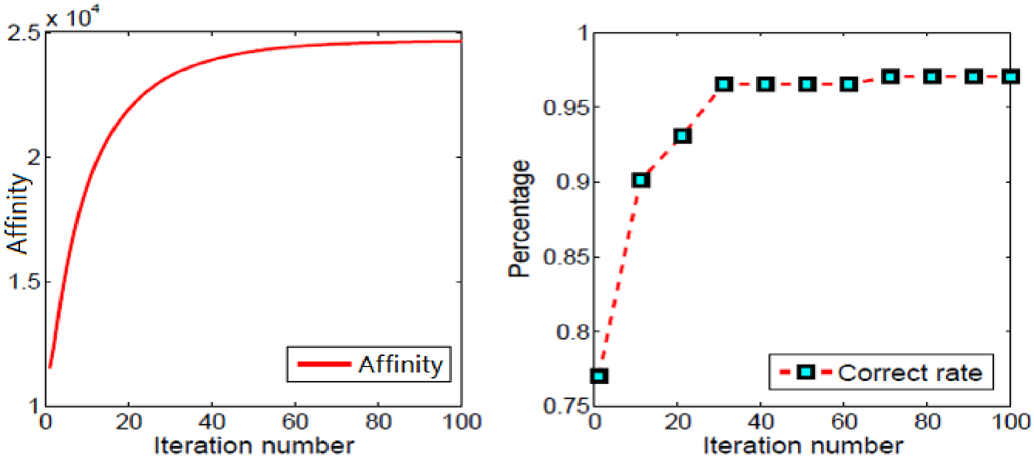


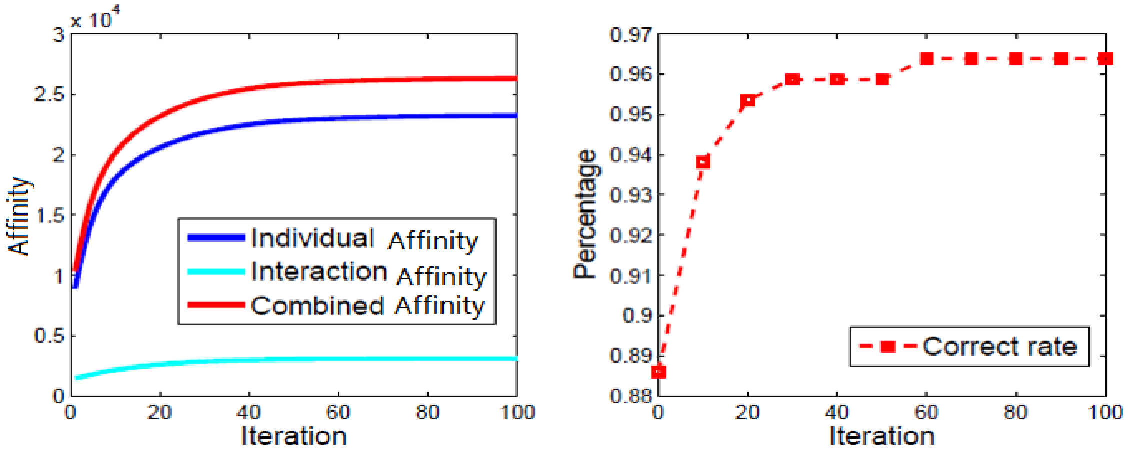
Fig. 6. Multi-object association results of different methods on 220 example frames in the Dense-1 sequence: (a) Dual L_1 normalized context-aware tensor power iteration (with 1 mismatch); (b) Rank-1 tensor approximation (with 6 mismatches); (c) Min-cost flow (1 mismatches) (d) ICM-like (with 22 mismatches); (e) Hungarian association (with 13 mismatches); (f) Network flow (with 30 mismatches).

Figs. 4, 5, and 6 show some examples of the association results of the rank-1 tensor approximation method, the dual L_1 normalized context-aware tensor power iteration method, and the competing methods on the CLIF-3 sequence, the Sparse-1 sequence, and the Dense-1 sequence, respectively. It is seen that the rank-1 tensor approximation method and the dual L_1 normalized context-aware tensor power iteration method yield fewer mismatches (association errors) than the two frame associations-based methods, the Hungarian association method and the network flow method. Our methods perform better than the Hungarian association method and the network flow method on both sparse and dense scenarios. These comparisons indicate the effectiveness of the high order temporal affinities.



(a) (b)

Fig. 7. The affinity and association performance variations in the iteration process for the rank-1 tensor approximation for one frame set in the PSU dataset: (a) The curve of the affinity as a function of the number of iterations; (b) Correct match rate curve.



(a) (b)

Fig. 8. The affinity and association performance variations in the iteration process for the dual L_1 normalized context-aware tensor power iteration for one frame set in the PSU dataset: (a) The curves of different affinities as functions of the number of iterations; (b) The correct matching rate curve.

The Dense-1 sequence was used as an example to show the process of convergence of our rank-1 tensor approximation and our dual L_1 normalized context-aware tensor power when the number of iterations increases. Fig. 7 (a) shows the curve of the temporal affinity as a function of the number of variations for the rank-1 tensor approximation. Fig. 7(b) shows the curve of the correct associations, where a binary decision is made for every 10 iterations. It is seen that the rank-1 tensor approximation tends to converge and the association results are improved when the number of iterations increases. Fig. 8(a) shows the curves of the temporal affinity, the motion context affinity, and the temporal and context combined affinity as functions of the number of iterations for the dual L_1 normalized context-aware tensor power iteration. Fig. 8(b) shows the curve of the correct associations. It is seen that the temporal affinity, the context affinity, and the combined affinity all increase together during the iteration process, and the association performance improves gradually. Therefore, the designed association affinities are reasonable.

Table 4. Runtimes of different multi-object tracking methods on the CLIF dataset and the PSU dataset (seconds)

Sequences Method	CLIF-1	CLIF-2	CLIF-3	Dense-1	Dense-2	Sparse-1
Network flow [17]	10	10	24	9	17	2
Hungarian	34	31	383	2	2	1
ICM	93	100	1113	23452	842	15
Our tensor approximation	86	116	455	936	320	13
Our context power iteration	88	119	459	942	323	14

All tests run on a laptop (2.1GHz Intel Core i7 with 8G memory) without code optimization. Table 4 compares the runtimes of our rank-1 tensor approximation method, our dual L_1 normalized context-aware tensor power iteration method, and the competing methods on the CLIF dataset and the PSU dataset, where the runtime for the object detection is excluded but the runtime required to build the affinity tensors is included. The following points are noted:

- The rank-1 tensor approximation method and the dual L_1 normalized context-aware tensor power iteration method overall require less runtime than the ICM-like method, in particular on the larger and more complex sequences CLIF-3 and Dense-1 which include more objects, have lower frame rates, and require the production of more global trajectory hypotheses.
- The runtimes of the rank-1 tensor approximation method and the dual L_1 normalized context-aware tensor power iteration method are higher than the runtimes of the Hungarian association method and the network flow method. This indicates that modeling high-order temporal affinities require more runtimes.
- The runtime difference between our rank-1 tensor approximation method and our context-aware power iteration method is very marginal. This indicates that modeling context in our context-aware power iteration method does not increase the runtime by very much. The motion contexts are efficiently modeled in our context-aware power iteration method.

8.1.2. PETS 2009 and TUD-Stadtmitte

Two pedestrian datasets, PETS 2009 and TUD-Stadtmitte, were used to test the performance of high-level motion context-based power iteration for pedestrian association. As in [31], the hierarchical association strategy was utilized: first, based on the results of pedestrian detection, low-level detection associations were carried out to produce basic tracklets; then, high-level tracklet associations were found to produce object trajectories.

We used the rank-1 tensor approximation to obtain the basic tracklets using (53) as the temporal affinity. For the high-level tracklet associations, a longer time interval ensures that high-level motion contexts introduced in Section 6.2.2 are useful for tolerating the inaccuracy of object detection. Therefore, we used the proposed dual L_1 normalized context-aware tensor power iteration to find the high-level association on the tracklet sets.

Given two tracklets $T_j:\{\mathcal{O}_j^{t_s^j},...,\mathcal{O}_j^{t_e^j}\}$ and $T_l:\{\mathcal{O}_l^{t_s^l},...,\mathcal{O}_l^{t_e^l}\}$, their contextual affinity cc_{jl} was computed using (57). Let $h_b^j\,(h_b^l)$ be the value in the *b*-th bin of the average color histogram of the tracklet T_j (T_l). The appearance affinity cp_{jl} between T_j and T_l is defined as:

$$cp_{jl}=\sum_b \min(h_b^j,h_b^l)\,, \tag{66}$$

where *b* indexes the bin number of the histogram. Let $\Delta t=t_s^l-t_e^j$ be the time gap between T_j and T_l, and let Θ be the temporal threshold for possible tracklet associations. It is not useful to consider the affinity between T_j and T_l when Δt is large. So, the temporal distance affinity ct_{jl} between T_j and T_l is defined as:

$$ct_{jl}=\begin{cases}\exp\left(-\dfrac{\Delta t}{\Theta}\right), & if\ 0<\Delta t<\Theta\\ 0, & otherwise.\end{cases} \tag{67}$$

Let $\Delta d=\mathbf{p}_l^{t_s^l}-\mathbf{p}_j^{t_e^j}$ be the spatial displacement from the object $\mathcal{O}_j^{t_e^j}$ to $\mathcal{O}_l^{t_s^l}$, and let $\mathbf{z}_l^{t_s^l}\,(\mathbf{z}_j^{t_e^j})$ be the velocity of $T_l\,(T_j)$ at time $t_s^l(t_e^j)$. The spatial distance affinity cd_{jl} is defined according to the differences between Δd and the predicted distances that the objects corresponding to T_j and T_l may move during Δt:

$$cd_{jl}=\frac{1}{2}\exp\left(-\frac{\left\|\Delta d-\Delta t\,\mathbf{z}_j^{t_e^j}\right\|^2}{2\left\|\mathbf{z}_j^{t_e^j}\right\|^2}\right)+\frac{1}{2}\exp\left(-\frac{\left\|\Delta d-\Delta t\,\mathbf{z}_l^{t_s^l}\right\|^2}{2\left\|\mathbf{z}_l^{t_s^l}\right\|^2}\right). \tag{68}$$

The association affinity used in (34) is defined by $c_{jl}=(cp_{jl}+cd_{jl}+cc_{jl})ct_{jl}$.

The pedestrian detection results in [23, 24] were used as the association inputs. As there are no training samples for the PETS 2009 dataset and the TUD-Stadtmitte dataset, we randomly selected 10 consecutive frames from each of these two datasets to tune the parameters. The parameter α in (34) was set to 0.4 and 0.2 for the PETS 2009 and TUD-Stadtmitte datasets respectively, since the object motions are more regulated in the PETS 2009 scenarios than in the TUD-Stadtmitte scenarios. The parameter Θ in (67) is independent of scenarios. It was set to 25 empirically for both datasets. Finally, two kinds of metrics were applied to evaluate the tracking performance. The first is the CLEAR MOT metric [3] including multi-object tracking accuracy (MOTA) and multi-object tracking precision (MOTP). The second metric [23, 24] evaluates the numbers of mostly/partially tracked (MT/PT), numbers of fragments, and ID switches.

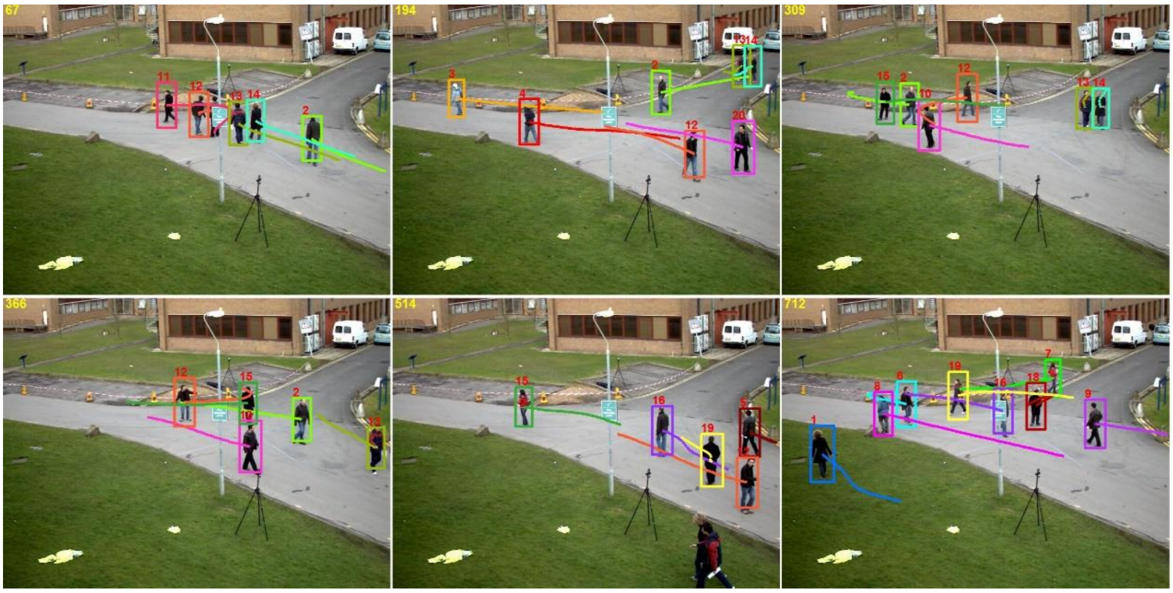


Fig. 9. Tracking results of our dual L_1 normalized context-aware tensor power iteration method with both contexts shown in Fig. 2 on the PETS 2009 dataset: the trajectory ID is shown in the top left corner of the bounding box of each object. The current state of each object and its historical trajectory in the most recent 50 frames are also shown.

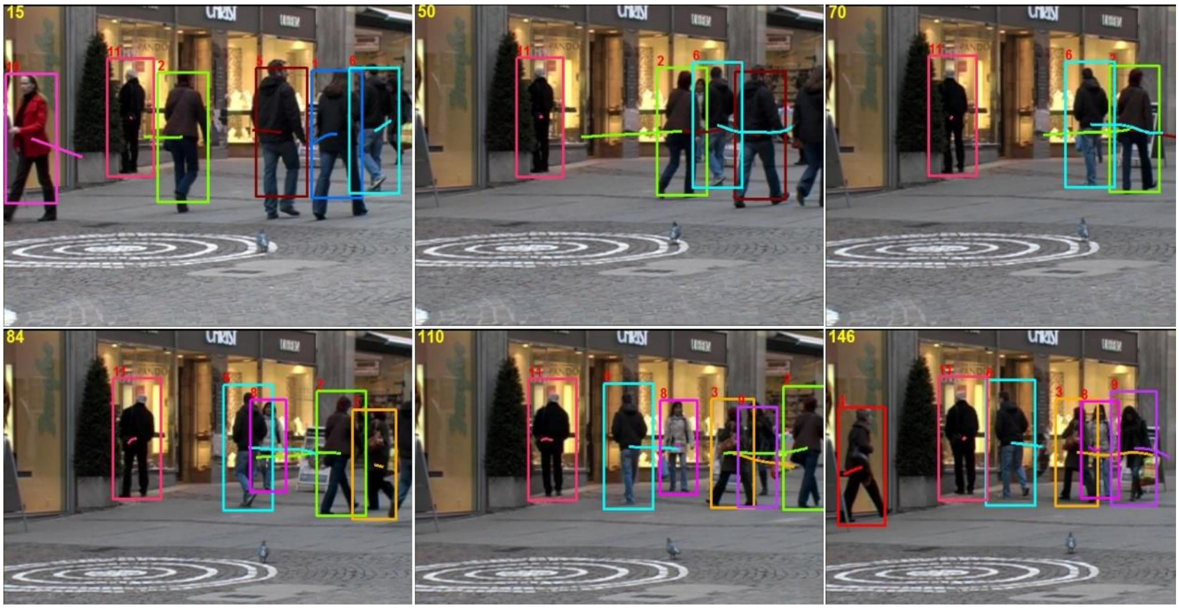


Fig. 10. Tracking results of our dual L_1 normalized context-aware tensor power iteration method with both contexts shown in Fig. 2 on the Stadtmitte dataset.

Figs. 9 and 10 show the tracking results of our dual L_1 normalized context-aware tensor power iteration method, with both contexts shown in Fig. 2, on the PETS 2009 and TUD-Stadtmitte datasets, respectively. It is shown that there is no ID switch. Tables 5 and 6 compared our multi-object tracking methods with state of the art methods in [17, 23, 24] on the PETS 2009 and TUD-Stadtmitte datasets respectively, where our methods include the rank-1 tensor approximation method and the dual L_1 normalized tensor power iteration methods with high-level context shown in Fig. 2 (a) alone, context shown in Fig. 2 (b) alone, or both contexts shown in Figs. 2(a) and (b). The following points are noted:

Table 5. Comparison results on the PETS 2009 dataset

Performance / Method	Rec.↑	Prec.↑	MOTA↑	MOTP↑	MT↑	PT↓	Frag↓	IDs↓
[23]	91.8	99.0	-	-	89.5	10.5	9	0
[17]	94.0	97.4	88.9	80.9	89.5	10.5	13	10
Rank-1 tensor approximation	96.0	98.2	92.7	81.8	94.7	5.3	11	7
Ours with contexts shown in Fig. 2(a)	97.4	98.5	94.7	81.4	94.7	5.3	8	6
Ours with contexts shown in Fig. 2(b)	96.6	98.8	94.9	81.6	94.7	5.3	8	5
Ours with contexts in both Figs. 2(a) and 1(b)	97.7	98.9	96.1	81.8	94.7	5.3	6	4

Table 6. Comparison results on the TUD-Stadtmitte dataset

Performance / Method	Rec.↑	Prec.↑	MOTA↑	MOTP↑	MT↑	PT↓	Frag↓	IDs↓
[24]	87.0	96.7	-	-	70.0	30.0	1	0
[17]	83.8	96.5	75.9	82.6	80.0	20.0	10	8
Rank-1 tensor approximation	83.9	98.8	80.4	87.7	70.0	30.0	5	3
Ours with contexts shown in Fig. 2(a)	85.4	98.6	81.3	87.8	80.0	20.0	2	2
Ours with contexts shown in Fig. 2(b)	83.7	99.7	81.8	88.8	80.0	20.0	2	1
Ours with contexts in both Figs. 2(a) and (b)	84.0	99.9	82.5	89.3	80.0	20.0	1	0

- Our dual L_1 normalized context-aware tensor power iteration methods, overall, yield more accurate results than the rank-1 tensor approximation method. There are fewer fragments and much fewer ID switches, as well as higher TA, TP, Prec., and Rec. The ID switch even reduces to 0. Both types of motion contexts shown in Fig. 2 improve the tracking results. This illustrates the effectiveness of the motion contexts on reducing association errors and on merging short tracklets into long tracks. A combination of these two types of high-level contexts improves the performance more significantly. This shows the mutual complementarity between the two types of contexts.
- Although our methods utilize the simple histogram appearance model, while the methods in [23, 24], which are pairwise association-based, utilize much more powerful learnt appearance model, our methods in general perform better than the methods in [23, 24]. On the PETS 2009 dataset, our methods have fewer fragments, and higher values for MT and Rec. On the TUD-Stadtmitte dataset, our methods have higher values for MT and Rec. The network flow-based algorithm [17] which is two-frame

association-based performs worse than our methods as well as the methods in [23, 24]. These partly indicate the effectiveness of high-order temporal affinities.

8.1.3. MOT16 challenge benchmark

On the MOT16 challenge benchmark dataset, we tested the performance of the dual L_1-normalized hyper-context aware tensor power iteration method for online multi-object tracking. The association affinity used in (49) was defined using the appearance affinity and motion consistency. The hyper-context in (49) was defined using the spatial structural potential.

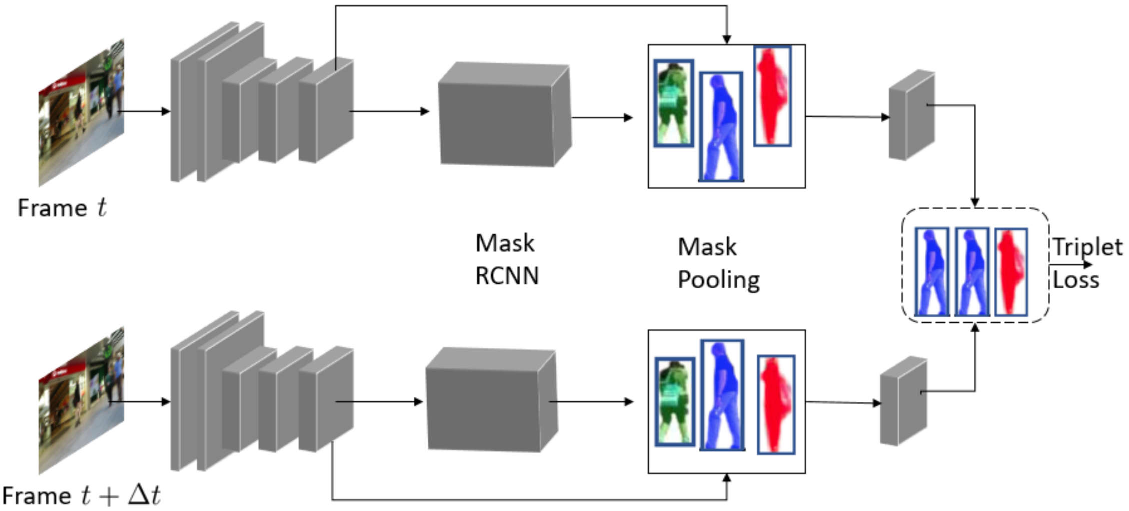


Fig. 11. The architecture of the siamese neural network for extracting appearance affinity based on identity masks.

The appearance affinity was obtained using a siamese neural network with object patch tuples as the inputs. It is hard to distinguish objects that close to each other, as the bounding boxes overlap and share many common features. Thus, we extracted the appearance features just from the mask area using a siamese network framework based on the mask RCNN (regions with convolutional neural network features) with a ResNet-FPN (residual network-feature pyramid network) backbone [54] (shown in Fig. 11). Objects' boxes and masks were produced by the mask RCNN, and the features were selected by masks from the final convolutional layer of the third stage of ResNet50. Then, triple samples were input to a shallow siamese neural network to extract the 128-dimensional appearance features. Let $pred_r$, pos_r, and neg_r be the feature vectors of the prediction, positive and negative samples for the r-th input tuple. The triplet loss is defined as:

$$\mathcal{L} = \sum_r \max(\cos(pred_r, neg_r) - \cos(pred_r, pos_r) + \Omega, 0), \tag{69}$$

where Ω denotes the threshold for the margin of separation between correct and incorrect pairs and cos(.,.)

denotes the cosine distance between two vectors. Let f_i^k be the deep appearance feature vector of the observation of association *i* at frame *k*, extracted by the network. The energy produced by appearance affinity for the tuple of associations *i*, *j*, and *l* is calculated by $\cos(f_i^k, f_i^{k+1})\cos(f_j^k, f_j^{k+1})\cos(f_l^k, f_l^{k+1})$.

To estimate the motion consistency, the velocity of one object was assumed to be a constant in a short period. A simple linear model was used to predict the objects state. Let $\mathbb{O}_c'$ be the predicted observation of association *c* and $\mathbb{O}_c$ be of the observation of association *c*. The motion consistency between associations *i*, *j*, and *l* is characterized as:

$$\exp\left(-\frac{\omega_1}{3}\sum_{c\in\{i,j,l\}}\left\|\mathbb{O}_c'-\mathbb{O}_c\right\|_2\right), \tag{70}$$

where ω_1 is a weight parameter.

The spatial structural potential is defined using the relative structural information of observations, as it is an affine-invariant potential. To model this affine-invariant property, we define the spatial potential of association tuple $\{i,j,l\}$ as:

$$\exp\left(-\omega_2\sum_{c_1,c_2\in\{i,j,l\}}\left\|distance(\mathbb{O}_{c_1}',\mathbb{O}_{c_2}')-distance(\mathbb{O}_{c_1},\mathbb{O}_{c_2})\right\|\right), \tag{71}$$

where ω_2 is a weight parameter. We used the absolute difference of distances between observations rather than relative ratio, because the larger the distance the smaller the possibility of changing the relative position.

The MOT16 challenge benchmark includes a training sample set and a test sample set. Each of these sets is composed of seven sequences, with frontal-view scenes taken by moving cameras or top-down surveillance setups. Evaluation was carried out according to the following metrics: multi-object tracking accuracy (MOTA), multi-object tracking precision (MOTP), ID F1 score (IDF1) [55], mostly tracked targets (MT), mostly lost targets (ML), false positives (FP), false negatives (FN), identity switches (IDs), and fragmentation (Frag). The samples in the training set are used for researchers to tune the parameters. The ground-truth of the samples in the test set is not supplied to researchers. The results on the test set must be sent to the MOT16 challenge benchmark dataset organizers who report the accuracy of the results. The parameters were tuned to yield an optimal multi-object tracking accuracy (MOTA). We found that when ω_1 and ω_2 are larger than 1.5, the values of MOTA are too low. Then, we sampled ω_1 and ω_2 from 15 values 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, and 1.5. The optimal combination of ω_1 and ω_2 is determined with the maximum of MOTA. For the

MOT16-14 sequence, parameter ω_1 was determined as 0.4, and ω_2 was determined as 1.1. For the all the other sequences, ω_1 was determined as 0.7 and ω_2 was determined as 0.3. Thus, two sets of parameter values were used. In order to set the correct benchmarking protocol, we also tested our method using one set of parameter values, i.e., ω_1 was set to 0.7 and ω_2 was set to 0.3 for all the sequences.

Table 7. Comparison results on the MOT16 challenge dataset: We compared with other published methods with private detections

Method	Mode	MOTA(%)↑	MOTP (%)↑	IDF1(%)↑	MT (%)↑	ML(%)↓	IDs↓	Frag↓	FP↓	FN↓	Hz(fps)↑
LMP_p [57]	Batch	71.0	80.2	70.1	46.9	21.9	434	587	7880	44564	0.5
KDNT [58]	Batch	68.2	79.4	60.0	41.0	19.0	933	1093	11479	45605	0.7
MCMOT HDM [59]	Batch	62.4	78.3	51.6	31.5	22.2	1394	1318	9855	57257	34.9
NOMTwSDP16 [60]	Batch	62.2	79.6	62.6	32.5	31.1	406	642	5119	63352	3.1
POI [58]	Online	66.1	79.5	65.1	34.0	20.8	805	3093	5061	55914	9.9
DeepSORT 2 [61]	Online	61.4	79.1	62.2	32.8	18.2	781	2008	12852	56668	17.4
SORTwHPD16 [62]	Online	59.8	79.6	53.8	25.4	22.7	1423	1835	8698	63245	59.5
IOU [63]	Online	57.1	77.1	46.9	23.6	32.9	2167	3028	5702	70278	3004.6
RAR16wVGG [64]	Online	63.0	-	63.8	39.9	22.1	482	1251	13663	53248	1.6
Ours-1	Online	64.8	78.6	73.5	40.6	22.0	794	1050	13470	49927	18.2
Ours-2	Online	63.2	78.7	73.2	38.5	21.6	821	1103	12980	50635	17.7

Table 7 compares our dual L_1 normalized context-aware tensor power iteration method with the state-of the art methods on the MOT16 challenge benchmark dataset where "Ours-1" refers to the results of our method using two sets of parameter values and "Ours-2" refers to the results of our method using one set of parameter values for all the sequences. It is seen that our tracker is a strong competitor to the competing online trackers [58, 61, 62, 63, 64] which are pairwise association-based. In particular, our method returns the highest identified detection score and MT, and fewer fragments among all the online pairwise association-based methods while maintaining competitive MOTA scores, ML, and identity switches. This shows the effectiveness of the high-order temporal affinities in our tracker. Furthermore, our method returns a higher number of false positives which impair the tracking accuracy. In general, applying a larger confidence threshold to the detections potentially increases the tracking performance. Most of the false positives in our model were generated from the sporadic detector responses at static scene geometry. Due to high-order spatial structural information and larger temporal distance (i.e., the number of the frames before the current frame, which are used to find the object associations for online tracking), these false positives usually are propagated to subsequent association results. As shown by the score of IDF1, which is more appropriate than MOTA to evaluate the robustness of the tracker, these mismatches do not lead to continual identity switches. It is noted that when one set of parameter values was used the accuracy of our method is only slight reduced. It is still comparable with the state of the art. In addition, our model even yields more accurate results than some state of the art methods in batch mode, such as NOMT which is significantly more complex and uses frames in the near future. Some qualitative results are shown in Fig. 12. It is seen that objects are tracked correctly when even occlusions are encountered or the scene changes greatly due to camera movement. The runtime for object associations in our method is approximately 18 frames per second and it can be much faster with parallel operation.

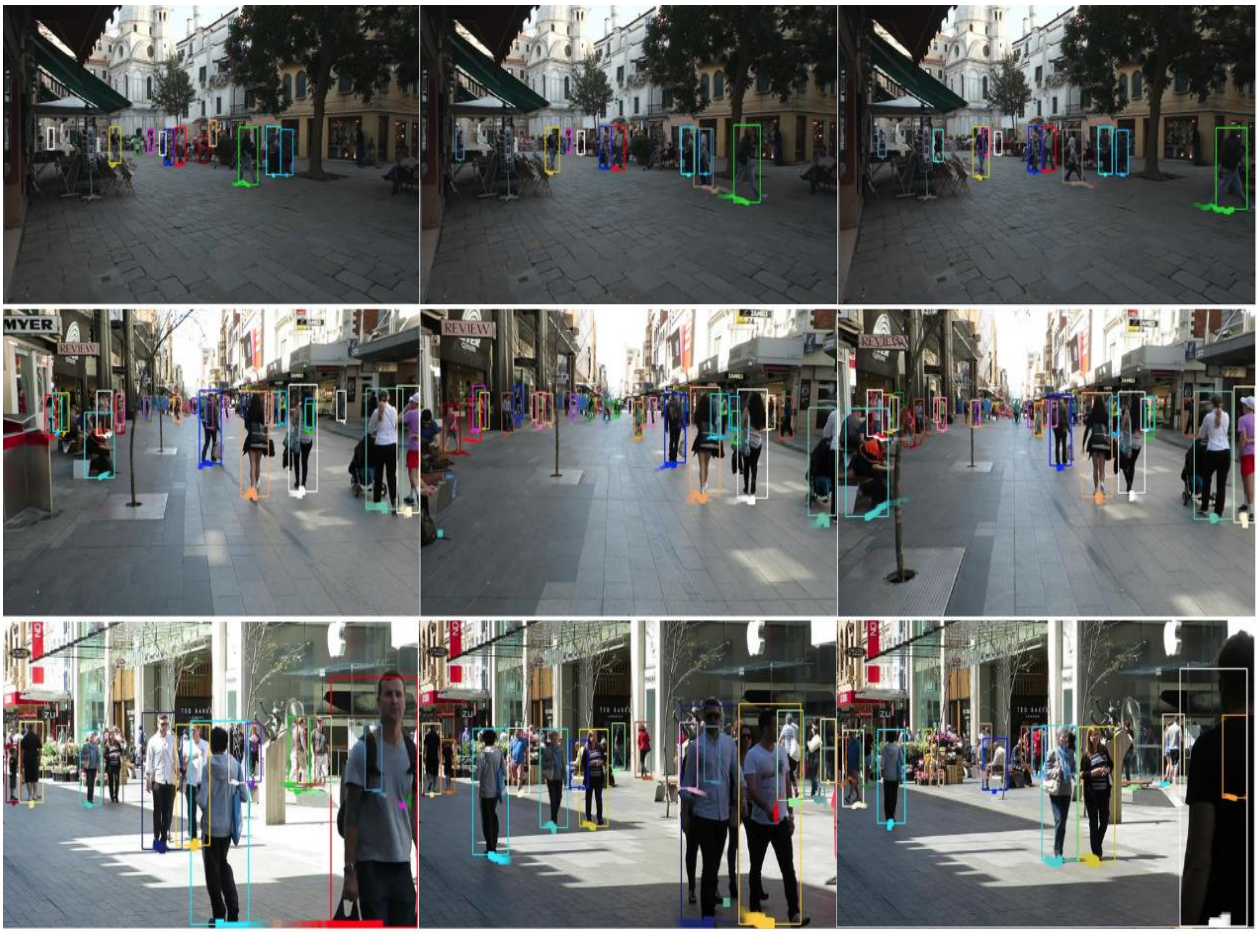


Fig. 12. Some qualitative results on the MOT16 benchmark: the same identity labeled by box with the same color.

To evaluate the effectiveness of proposed high-order affinity containing motion consistency and spatial structural information, we checked the changes in MOTA under different tensor orders and different temporal distances on the MOT16-10 sequence which was recorded using a moving camera. As shown in Fig. 13, both the MOTA and IDF1 are improved along with the increase of the tensor order, as richer spatial information is extracted. This clearly shows the effectiveness of high-order temporal affinities. We observed that more samples from previous times benefit the performance of our tracker within a certain period. However, the performance is not improved and may even decrease when the temporal distance is too long. The reason is that the linear motion assumption is no longer valid and the appearance changes greatly after a long time interval.

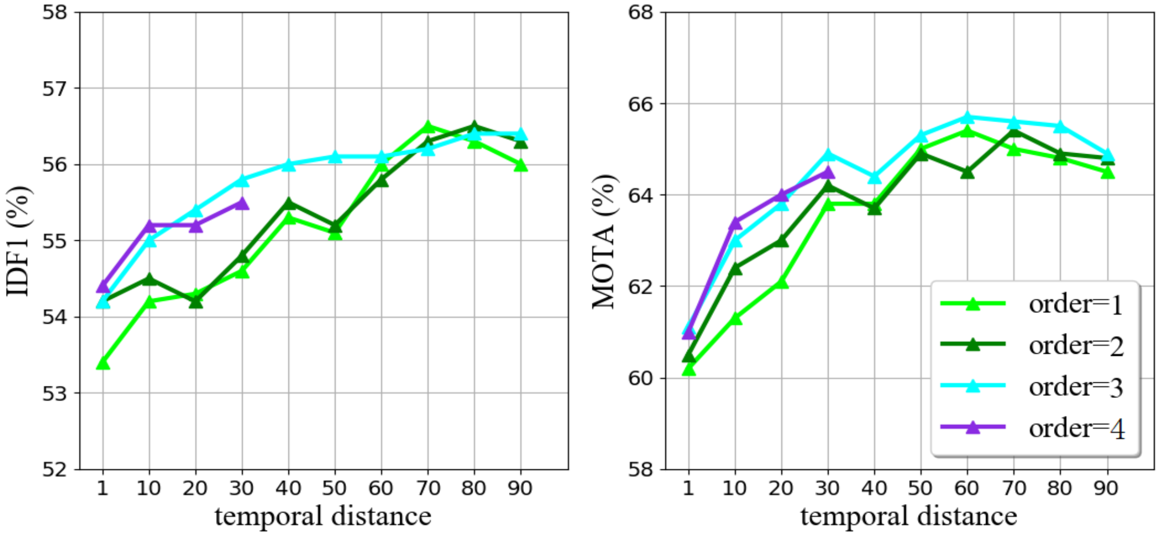


Fig. 13. Performance of our method on the MOT16-10 sequence with different orders of the affinity tensor: The maximum temporal distance was set to 30 and the maximum order was set to 4.

To evaluate the effectiveness of our feature extraction neural network based on the identity mask, we compared the proposed tracker with the DeepSORT tracker whose code is available and whose performance is better than other online trackers. As shown in Fig. 14, for either DeepSORT or our method, using the features with identity masks yields better results than using the features without identity masks. Whether using the features with or without identity masks, the proposed tracker outperforms the DeepSORT tracker.

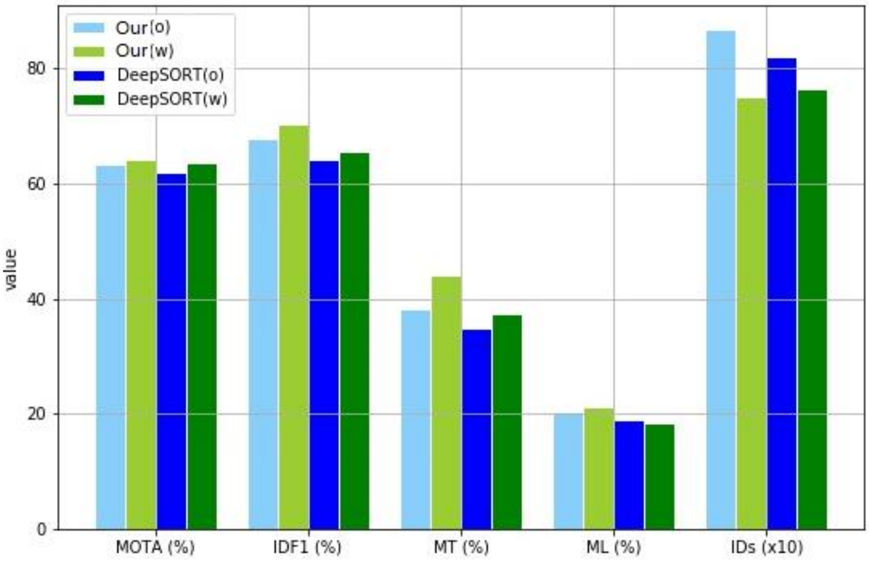


Fig. 14. Tacking results using features with/without identity masks.

8.2. Multi-graph matching

In real scenarios for multi-graph matching, the graphs often have noisy structures with outlier vertices. These vertices should not be mapped to any real vertex. By adding dummy vertices to graphs, we make the number of the vertices in each graph the same. The dummy vertices in a graph are allowed to match with non-dummy vertices in other graphs. A small affinity is set for the dummy vertices to suppress erroneous matches.

To test the performance of our multi-graph matching method, extensive experiments were conducted on the following public benchmark datasets:

- **The CMU House/Hotel dataset [32]:** The House and Hotel sequences contain 111 frames and 101 frames respectively, and each frame has thirty landmarks. Following the setting in [35], we randomly selected 10 landmarks as the inliers, and randomly selected 3 landmarks from the rest as the outliers.
- **The WILLOW-ObjectClass dataset [33]**: The ObjectClass dataset consists of five real world image sequences. Four sequences were used in the experiments including Duck (50 images), Car (40 images), Motorbike (40 images), and Winebottle (66 images). There are 10 manually annotated landmarks in each image, but the annotations are not accurate. With the same settings as in [37], the 10 landmarks were used as the inliers and supplemented by 2 outliers detected from the background using the SIFT detector.
- **The Repetitive Structure dataset [34]:** This dataset consists of two sequences describing repetitive structures, which make image matching difficult due to the ambiguous features. The Building sequence (16 images) was selected as the test sequence. For each image, we retained 10 landmarks as the inliers and randomly sampled three landmarks from the rest as the outliers.

Graph sets with various sizes were utilized to validate the performances of the multi-graph matching methods. Generally, the experiments were conducted on 4-graph, 6-graph, 8-graph, 10-graph, and 12-graph matching tasks. For the robustness evaluation, 10 random tests were performed for each matching task, and the result is the average of all the 10 tests.

The impacts of the two basic components, the vertex affinity and the hyper-edge affinity, vary in different scenarios. So, the parameter α in the optimization (59) is dependent on the scenario. The more stable the graph structure, the more confident the component of the hyper-edge affinity. For the CMU-House/Hotel and Building sequences, the Motorbike and Winebottle sequences, and the Duck and Car sequences, we used the first 10 frames in each sequence to tune the parameter α. The parameter α was empirically set to 8 for the CMU-House/Hotel and Building sequences, 4 for the Motorbike and Winebottle sequences, and 2 for the Duck and Car sequences. The regularized factor σ^2 in (64) only depends on the triangle features of hyperedges. It is independent of scenarios. It was empirically set to 2 throughout the experiments. The number of the dual L_1 normalized tensor power iterations was set to 100 throughout all the experiments.

There are two main measures of multi-graph matching: 1) accuracy: the number of correctly matched inliers divided by the total number of inliers, as popularly used in related work [38, 39, 40]; 2) consistency: the number of consistent matches divided by the number of all possible matches (a detailed definition can be found in [35]. In our work, the accuracy metric was mainly applied, since our method naturally guarantees 100% of consistency, which is a merit of our method.

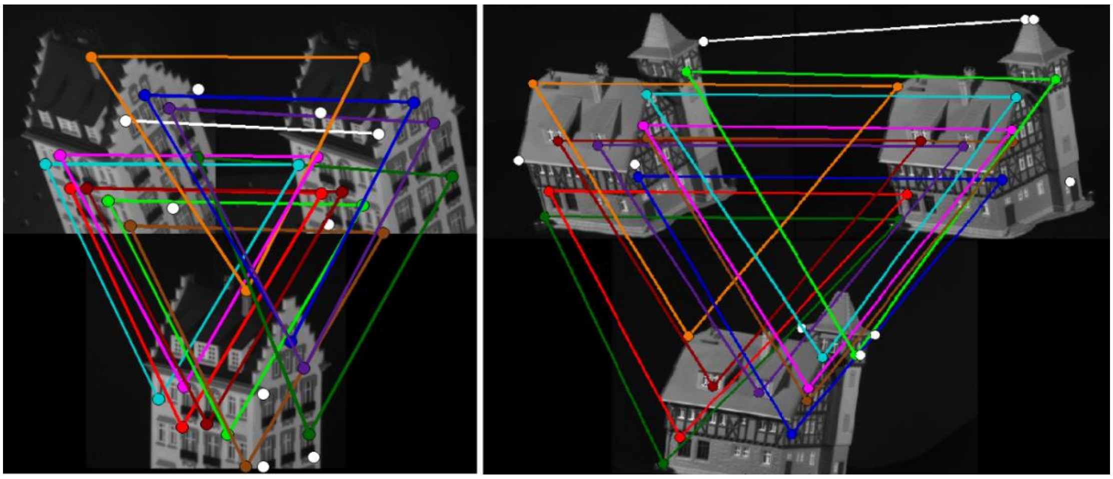
The effectiveness of the proposed method was verified by comparing with the state of the art [34, 35, 36]. They are the permutation synchronization method (Match-Sync) [36], the alternative optimization method (Match-Opt) [36], and the graduated consistency-regularized optimization algorithm (Match-Grad) [35]. The results of the three competing methods on the CMU dataset and WILLOWObjectClass dataset are taken from the work in [35, 37].

The proposed optimization objective consists of two components: the unary vertex affinity and the hyper-edge affinity. Each objective can be used alone in the optimization to solve the matching problem. The proposed method is flexible to accommodate different kinds of optimizations. When the unary vertex affinity is exploited only, the optimization degenerates into the multi-dimensional assignment, and the solution is termed as “Tensor-MDA”. With the hyper-edge affinity used only, the problem degenerates into the hyper-graph matching (HGM), and the method is termed as “Tensor-HGM”. The method for multi-graph matching using both the vertex affinity and the hyper-edge affinity is termed as “Tensor-MGM”. All the three methods, Tensor-MAD, Tensor-HGM, and Tensor-MGM, were tested in the experiments.

1) The CMU-House/Hotel dataset: The qualitative results on the Hotel and House sequences are shown in Fig. 15. It is seen that our Tensor-MGM method yields few mismatches and has an excellent performance. The results meet the full consistency, which is derived from the high-order matching naturally. The quantitative results on the CMU-House/Hotel sequences are presented in Table 8. It is seen that our multi-graph matching method, Tensor-MGM, which uses both the vertex affinity and the hyper-edge affinity performs the best in almost all the tests. Moreover, the tensor-MGM method has a remarkable improvement over the state of the art. The two variants, Tensor-MDA which uses the vertex affinity only and Tensor-HGM which uses the hyper-edge affinity only, also yield good results on this dataset. The pairwise graph matching (2-graph matching) using our tensor-MGM method yields an accuracy of more than 99% on the House and Hotel sequences. However, for multi-graph matching, there is the consistency measure besides the accuracy measure. Pairwise associations-based multi-graph matching methods cannot ensure 100% of consistency, while our tensor MGM-based multi-graph matching method naturally guarantees 100% of consistency. We compare the consistency for the following two methods:

- the 3-graph matching carried out by pairwise graph matching using our tensor-MGM (Pairwise)
- the 3-graph matching carried out directly using our tensor-MGM (D-Tensor).

The results are shown in Table 9. It is seen that the D-Tensor method keeps the accuracy of the Pairwise method and yields a consistency of 100%. However, the consistencies for the Pairwise method are 93.9% and 96.9% for the CMU-House sequence and the CMU-Hotel sequence, respectively.



(a) (b)

Fig. 15. Matching results of the proposed method across three graphs on the CMU-House/Hotel dataset: (a) The Hotel sequence; (b) The House sequence. The vertices and matches are color-coded, and correct matches appear in the same color as the vertices that they connect. White circles denote outliers. Best viewed in color.

Table 8. Matching accuracy (%) on the CMU-House/Hotel dataset

	CMU-House						CMU-Hotel					
	Match-Sync [34]	Match-Opt [36]	Match-Grad [35]	Tensor-MDA	Tensor-HGM	Tensor-MGM	Match-Sync [34]	Match-Opt [36]	Match-Grad [35]	Tensor-MDA	Tensor-HGM	Tensor-MGM
2-graph	86.4	--	--	74.9	94.1	99.3	87.4	--	--	75.4	95.5	99.8
4-graph	85.2	99.0	84.0	90.5	89.4	96.6	87.7	93.2	90.0	88.7	86.7	96.7
6-graph	83.2	81.6	85.2	90.0	87.7	96.2	81.5	72.3	87.9	87.3	85.3	97.7
8-graph	76.4	82.4	87.2	81.7	89.6	95.5	60.5	62.8	84.9	78.5	80.7	97.1
10-graph	68.8	80.2	79.2	78.1	83.6	94.3	63.8	69.1	84.5	78.7	83.7	93.7
12-graph	75.4	80.0	82.6	75.2	83.5	95.1	70.0	68.5	88.1	73.6	86.1	97.9

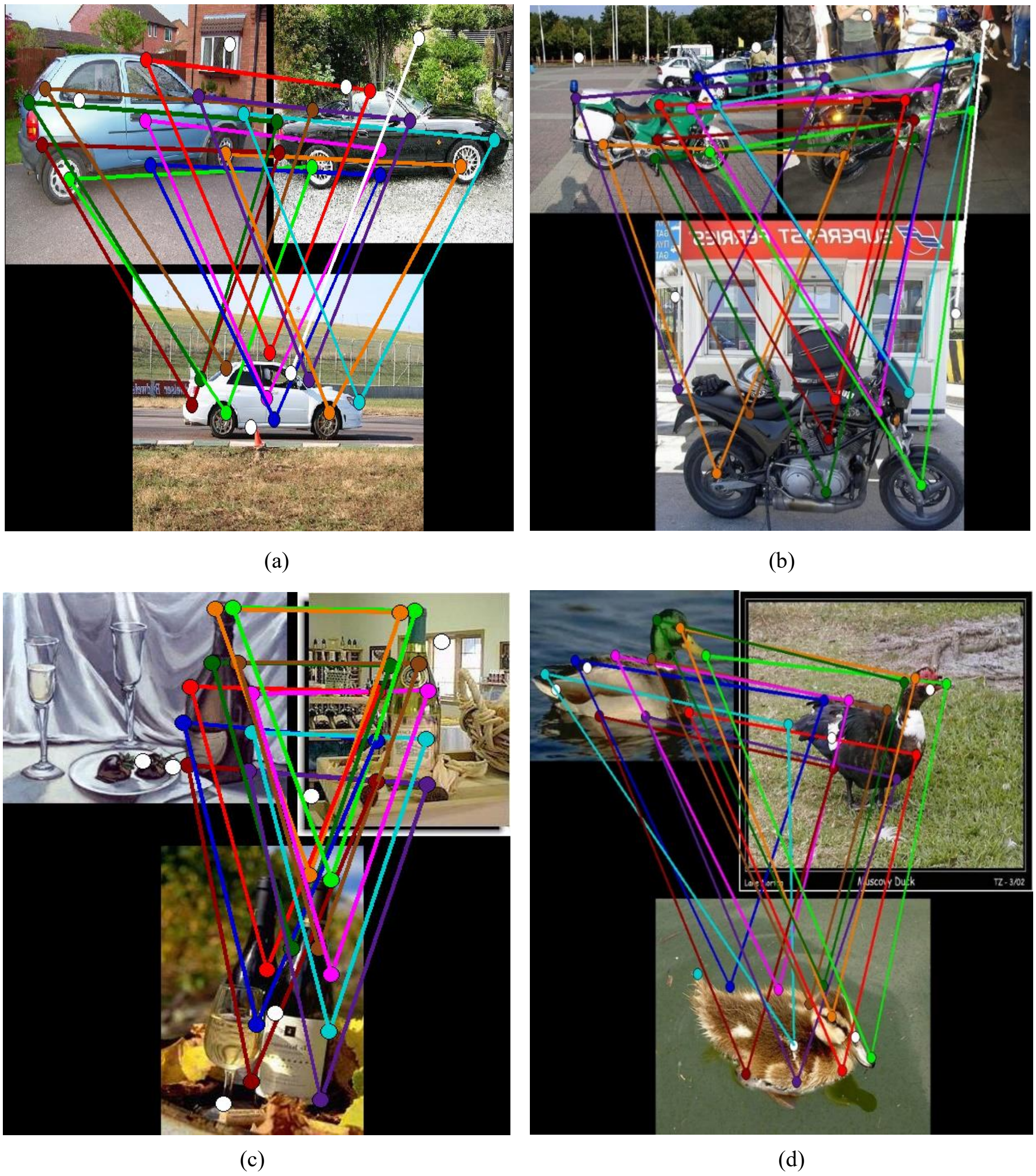
Table 9. The results of comparison of consistency (%) between the 3-graph matching carried out by pairwise graph matching (Pairwise) and the 3-graph matching directly using tensor-MGM (D-Tensor).

Method	Measure	CMU-House	CMU-Hotel
Pairwise	Accuracy	98.7	98.6
	Consistency	93.9	96.9
D-Tensor	Accuracy	98.1	98.7
	Consistency	100.0	100.0

To investigate the relevance of the affinity cost vs multi-graph matching, we compared the results of using different local appearance features, i.e., the features of color histogram, HoG (Histogram of Oriented Gradient), SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features), deep learning on AlexNet, and shape context. Table 10 shows the accuracy results when different local appearance features of vertices are used for our tensor-MGM method. It is seen that the results of the 2-graph matching and the 4-graph matching for different features are close to each other. The results of the 6-graph matching and the 8-graph matching for the HoG and shape context features are higher than those for other features.

Table 10. The accuracies of our tensor-MGM method for different local appearance features on the CMU House sequence.

Number of graphs	Color histogram	HoG	SIFT	SURF	Deep learning	Shape context
2-graph	98.7	98.9	97.9	98.3	98.5	98.8
4-graph	96.7	98.1	96.6	96.5	96.3	96.6
6-grpah	91.7	96.6	95.4	90.1	90.1	96.2
8-graph	85.7	92.3	90.6	83.3	83.0	95.5



(a)

(b)

(c)

(d)

Fig. 16. Matching results of the proposed method across three graphs on the WILLOW-ObjectClass dataset: (a) Car; (b) Motorbike; (c) Wine bottle; (d) Duck. The vertices and matches are color-coded, and correct matches appear in the same color as the vertices that they connect. White circles denote outliers. Best viewed in color.

Table 11. Matching accuracy (%) on the WILLOW dataset

	WILLOW-Car						WILLOW-Motorbike					
	Match-Sync [34]	Match-Opt [36]	Match-Grad [35]	Tensor-MDA	Tensor-HGM	Tensor-MGM	Match-Sync [34]	Match-Opt [36]	Match-Grad [35]	Tensor-MDA	Tensor-HGM	Tensor-MGM
4-graph	54.2	57.1	63.3	62.2	68.7	79.5	75.9	78.7	78.4	78.3	69.7	87.5
8-graph	61.5	69.6	74.3	62.6	49.6	75.7	84.6	82.5	86.3	76.5	65.7	85.6
12-graph	55.8	66.0	80.5	55.4	40.3	67.1	81.3	84.3	87.1	70.5	58.7	80.3

	WILLOW-Winebottle						WILLOW-Duck					
	Match-Sync [34]	Match-Opt [36]	Match-Grad [35]	Tensor-MDA	Tensor-HGM	Tensor-MGM	Match-Sync [34]	Match-Opt [36]	Match-Grad [35]	Tensor-MDA	Tensor-HGM	Tensor-MGM
4-graph	49.8	71.2	64.2	81.2	82.7	97.0	35.3	42.3	40.0	60.2	58.2	65.8
8-graph	37.5	91.2	82.9	78.9	76.4	94.3	40.8	45.8	50.6	60.6	48.2	61.3
12-graph	69.0	92.7	93.1	71.3	63.2	93.7	45.9	56.6	72.7	49.4	33.8	58.3

2) The WILLOW-ObjectClass dataset: The large pose and viewpoint variations, flexible landmark annotations, and noisy outliers make the matching on the ObjectClass sequences extremely difficult, in particular for the Duck and Car sequences. The qualitative results on the WILLOW dataset are shown in Fig. 16. The full matching consistency is clearly observed from the figure. The quantitative results on this dataset are presented in Table 11. It is seen that our dual L_1-normalized hyper-context aware tensor power iteration algorithm (Tensor-MGM) yields more accurate results than the two variants, Tensor-MDA and Tensor-HGM. Our Tensor-MGM method obtains the best results for the 4-graph and 8-graph matching tasks on all the sequences in the WILLOW dataset. For the 12-graph matching task, our method yields the most accurate result on the Winebottle sequence and the second accurate results on the Car and Duck sequences. Although the method in [35] yields higher accuracy than our Tensor-MGM method for the 12-graph matching task, as stated in [35], its consistency is about 70% [35]. As stated above, our method naturally yields 100% consistency. The results show the effectiveness of our multi-graph matching method as well as the effectiveness of the combination of the vertex affinity and the hyper-edge affinity.

3) The Repetitive Structure dataset: The qualitative matching is presented in Fig. 17 which shows the full matching consistency. The quantitative results on this dataset are presented in Table 12. The building sequence has many repetitive patterns and viewpoint changes, but the annotations are stable. In this case, our method achieves the favorable performance.

We discuss the results on these benchmark datasets from the aspects of convergence, vertex affinity, affinity combination, and complexity:

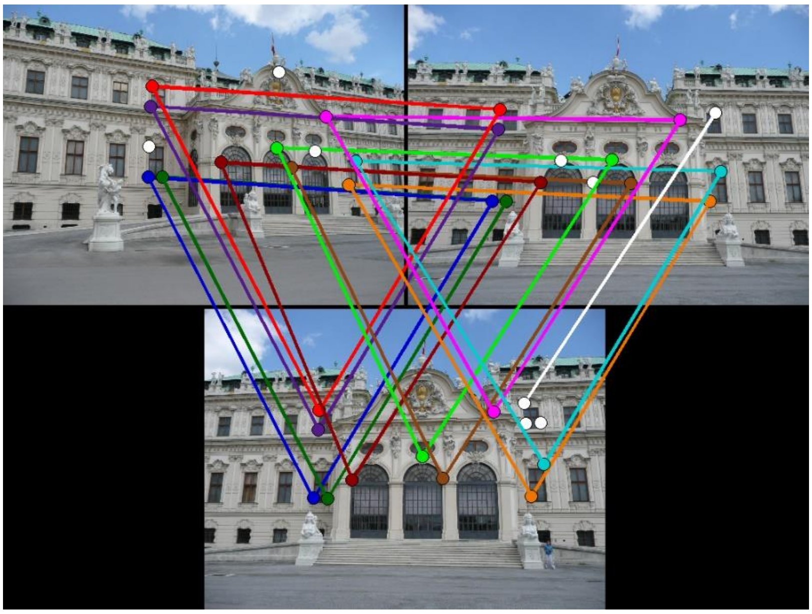
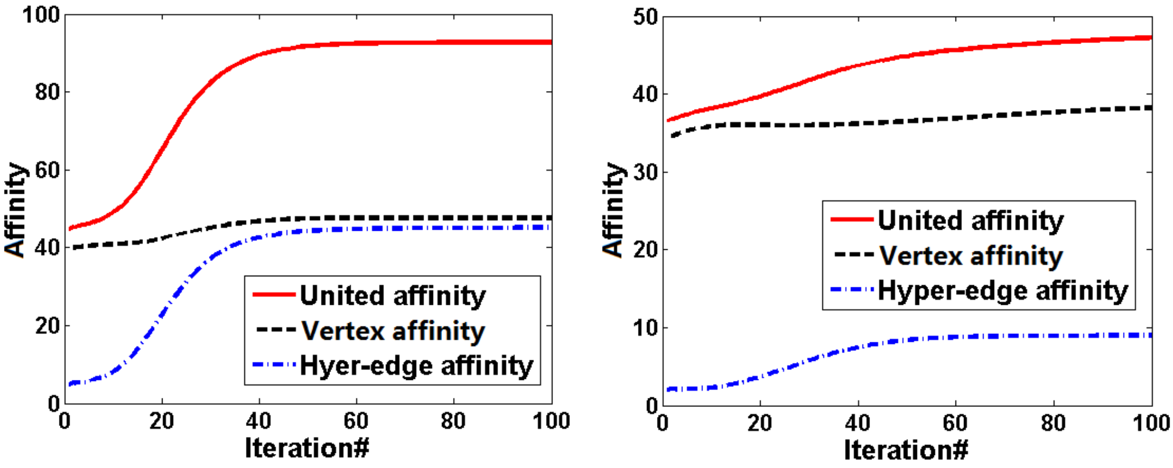


Fig. 17. The matching results of the proposed method across three graphs on the Building sequence: The vertices and matches are color coded, and correct matches appear in the same color as the vertices that they connect. White circles denote outliers. Best viewed in color.

Table 12. Matching accuracy (%) on the Building sequence

	Match-Sync [34]	Tensor-MDA	Tensor-HGM	Tensor-MGM
4-graph	76.5	82.8	87.3	92.8
6-graph	82.8	75.3	58.7	93.0
8-graph	77.9	75.0	53.2	88.3
10-graph	87.3	73.5	66.3	90.7



\ (a) (b)

Fig. 18. The curves of the united affinity, the vertex affinity, and the hyperedge affinity as functions of the number of iterations: (a) Curves for matching the house images; (b) Curves for matching the motorbike images.

1) Convergence: To test the convergence of the proposed dual L_1 normalized hyper context-aware tensor power iteration, the variation in affinity during the iteration process is shown in Fig. 18. There are two examples: one is the 4-graph matching on the CMU-House sequence and the other is the 4-graph matching on the WILLOW-

Motorbike sequence. The united affinity and the individual affinities (i.e., the vertex and hyper-edge affinities) are included in the figure. It is clear that the affinity gradually increases during the iteration and the proposed method converges.

2) Vertex affinity: The proposed method has the advantage that it allows the exploration of high-level vertex affinity which is not available in the state of the art methods. Although we utilize a simple affinity measure which is sensitive to the factors such as large deformations of graphs, the state of the art results are still obtained.

3) Affinity combination: It was observed from the experiments that both the Tensor-MDA and Tensor-HGM have good performances. By combining these two complementary affinities, the proposed method obtains a much better result. This indicates the necessity of incorporating high-order information across both multi-graphs and hyper-edges. In addition, the proposed method is adaptable to diverse edge or hyper-edge affinities, such as the pairwise edge similarity, the third or higher order hyper-edge affinity, and even the hybrid of different order hyper-edge affinities.

4) Runtime: Table 13 shows, for the CMU-House/hotel dataset, the runtimes of our dual L_1-normalized hyper-context aware tensor power iteration-based graph matching method for matching 2 graphs, 4 graphs, 8 graphs and 12 graphs, where the runtimes include the times required to build the affinity tensors. It is seen that the runtimes for the CMU-House and CMU-Hotel sequences are similar. When the number of graphs increases, the runtime inevitably increases.

Table 13. The runtimes for our multi-graph matching method on the CMU-House/Hotel dataset

Number of graphs		2 graphs	4 graphs	8 graphs	12 graphs
Runtime	CMU-House	0.036 seconds	0.09 seconds	3.29 seconds	10.21 minutes
	CMU-Hotel	0.037 seconds	0.11 seconds	3.31 seconds	10.18 minutes

8.3. Discussion on parameter tuning

In our work, some parameters, such as the regularized factor α, are dependent on the scenarios. To yield state of the art results, we chose these parameters differently for different datasets and some different sequences, as in previous work, such as [58, 61]. The reason is that the scenarios for the datasets, for example aerial videos and ground surveillance videos, are quite different. The scenario-dependent parameters should be set different values in order to obtain more accurate results than the competing methods which also vary the parameter values for different sequences [58, 61].

On the CLIF dataset, the PSU dataset, the PETS 2009 dataset, and the TUD-Stadtmitte dataset, for our tracking method, the parameters for the sequences in the same dataset have the same values because the scenarios in the dataset are similar. On the MOT16 challenge benchmark dataset, only the MOT16-13 and MOT16-14 sequences have different values for the parameters, while all the other sequences have the same values of the parameters. Our methods yield the state of the art multi-object tracking results on the CLIF dataset, the PSU dataset, the PETS 2009 dataset, the TUD-Stadtmitte dataset, and the MOT16 challenge benchmark dataset, and yield the state of the art multi-graph matching results on the CMU-House/Hotel and Building sequences, the Motorbike and Winebottle sequences, and the Duck and Car sequences. These partly indicate the generalization capabilities of our methods.

9. Conclusion

In this paper, the multi-dimensional assignment task has been formulated as the row and column constrained tensor approximation problem. A dual L_1-normalized context/hyper-context aware tensor power iteration optimization method has been proposed. In this method, temporal affinity and association contexts or hyper contexts are included in a combined optimization. Various types of pairwise contexts have been modeled. This optimization method has been applied to association-based multi-object tracking. Contextual cues and high-order motion information have been used simultaneously to alleviate the association ambiguity. The tensor power iteration method has also been applied to multi-graph matching. High-order vertex affinities and hyper-edge affinities have been explored to leverage graph matching accuracy and consistency. The experiments on diverse datasets have illustrated the effectiveness of the proposed methods.

Acknowledgements

This work is supported by the NSFC-general technology collaborative Fund for basic research (Grant No. U1636218), the Natural Science Foundation of China (Grant No. 61751212, 61721004, 61772225), Beijing Natural Science Foundation (Grant No. L172051), the Key Research Program of Frontier Sciences, CAS, Grant No. QYZDJ-SSW-JSC040, the CAS External cooperation key project, and the National Natural Science Foundation of Guangdong (No. 2018B030311046).

References

[1] S. Ali, V. Reilly, and M. Shah. “Motion and appearance contexts for tracking and re-acquiring targets in aerial videos,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-6, 2007.

[2] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, “Multiple object tracking using k-shortest paths optimization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806-1819, 2011.

[3] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” *EURASIP Journal on Image & Video Processing*, vol. 2008, Article ID 246309, pp. 1-10, 2008.

[4] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L.V. Gool, “Online multi-person tracking-by-detection from a single, uncalibrated camera,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820-1833, 2010.

[5] W. Brendel, M. Amer, and S. Todorovic, “Multiobject tracking as maximum weight independent set,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1820-1833, June 2011.

[6] A. Butt and R. Collins, “Multi-target tracking by Lagrangian relaxation to min-cost network flow,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1846-1853, 2013.

[7] R. Collins, “Multitarget data association with higher-order motion models,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1744-1751, 2012.

[8] N. Dalal and B. Triggs., “Histograms of oriented gradients for human detection,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[10] W. Ge, R. Collins, and R. Ruback, “Vision-based analysis of small groups in pedestrian crowds,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1003-1016, 2012.

[11] D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” *Physical Review E Statistical Physics Plasmas Fluids & Related Interdisciplinary Topics*, vol. 51, no. 5, pp. 4282-4286, 1995.

[12] H. Jiang, S. Fels, and J. Little, “A linear programming approach for multiple object tracking,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[13] Z. Khan, T. Balch, and F. Dellaert, “MCMC-based particle filtering for tracking a variable number of interacting targets,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805-1819, 2005.

[14] M. Luber, J. Stork, G. Tipaldi, and K. Arras, “People tracking with human motion predictions from social forces,” *in Proc. of IEEE Conference Robotics and Automation*, pp. 464-469, 2010.

[15] S. Pellegrini, A. Ess, K. Schindler, and L.V. Gool, “You’ll never walk alone: modeling social behavior for multi-target tracking,” *in Proc. of IEEE International Conference on Computer Vision*, pp. 261-268, 2009.

[16] S. Pellegrini, A. Ess, and L.V. Gool, “Improving data association by joint modeling of pedestrian trajectories and groupings,” *in Proc. of European Conference on Computer Vision*, pp. 452-465, 2010.

[17] H. Pirsiavash, D. Ramanan, and C. Fowlkes, “Globally-optimal greedy algorithms for tracking a variable number of objects,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1201-1208, June 2011.

[18] P. Scovanner and M. Tappen, “Learning pedestrian dynamics from the real world,” *in Proc. of IEEE International Conference on Computer Vision*, pp. 381-388, 2009.

[19] A. Sole-Ribalta and F. Serratosa, “Graduated assignment algorithm for multiple graph matching based on a common labeling,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 1, pp. 1350001-1 -1350001-17, Feb. 2013.

[20] X. Shi, H. Ling, W. Hu, J. Xing, and Y. Zhang, “Tensor power iteration for multi-graph matching,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5062-5070, 2016.

[21] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg, “Who are you with and where are you going?” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1345-1352, 2011.

[22] The Columbus Large Image Format CLIF dataset 2006. www.sdms.afrl.af.mil/index.php?collection=clif

[23] B. Yang and R. Nevatia, “Multi-target tracking by online learning of non-linear motion patterns and robust appearance models,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1918-1925, 2012.

[24] B. Yang and R. Nevatia, “An online learned CRF model for multi-target tracking,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2034-2041, 2012.

[25] L. Zhang, Y. Li, and R. Nevatia, “Global data association for multiobject tracking using network flows”, *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1-8, 2008.

[26] P. Regalia and E. Kofidis, “The higher-order power method revisited: convergence proofs and effective initialization,” *in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2709-2712, 2000.

[27] L. De Lathauwer, B. De Moor, and J. Vandewalle, “On the best rank-1 and rank-(r1, r2,..., rn) approximation of higher-order tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324-1342, 2000.

[28] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509-522, 2002.

[29] O. Duchenne, F. Bach, I. Kweon, and J. Ponce, “A tensor-based algorithm for high-order graph matching,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2383-2395, Dec. 2011.

[30] J. Lee, K.M. Lee, and M. Cho, “Hyper-graph matching via reweighted random walks,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1633-1640, 2011.

[31] C. Huang, B. Wu, and R. Nevatia, “Robust object tracking by hierarchical association of detection responses,” In: D. Forsyth, Torr P., Zisserman A. (eds), *European Conference on Computer Vision, Lecture Notes in Computer Science*, vol. 5303, pp. 788-801, Springer, 2008.

[32] http://vasc.ri.cmu.edu/idb/html/motion/

[33] M. Cho, K. Alahari, and J. Ponce, “Learning graphs to match,” *in Proc. of IEEE International Conference on Computer Vision*, pp. 25-32, 2013.

[34] D. Pachauri, R. Kondor, and S. Vikas, “Solving the multi-way matching problem by permutation synchronization,” *in Proc. of Annual Conference on Neural Information Processing Systems*, pp. 1860-1868, 2013.

[35] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and M. Chu, “Graduated consistency-regularized optimization for multi-graph matching,” In: D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (eds), *European Conference on Computer Vision*, Lecture Notes in Computer Science, vol. 8689, Springer, pp. 407-422, 2014.

[36] J. Yan, Y. Tian, H. Zha, X. Yang, and Y. Zhang, “Joint optimization for consistent multiple graph matching,” *in Proc. of IEEE International Conference on Computer Vision*, pp. 1649-1656, 2013.

[37] J. Yan, M. Chu, H. Zha, and X. Yang, “Multi-graph matching via affinity optimization with graduated consistency regularization,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1228-1242, June 2016.

[38] M. Cho, J. Lee, and K. Lee, “Reweighted random walks for graph matching,” *in Proc. of European Conference on Computer Vision*, pp. 492-505, Sep. 2010.

[39] F. Zhou and F. De la Torre, “Factorized graph matching,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1774-1789, Sep. 2016.

[40] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, “Consistency driven alternating optimization for multigraph matching: a unified approach,” *IEEE Trans. on Image Processing*, vo. 24, no. 3, pp. 994-1009, 2015.

[41] K. Shafique, M. Lee, and N. Haering, “A rank constrained continuous formulation of multi-frame multi-target tracking problem,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.

[42] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, “A generalized SD assignment algorithm for multisensor-multitarget state estimation,” *IEEE Trans. on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523-538, April 1997.

[43] B. Benfold and I. Reid, “Stable multi-target tracking in real-time surveillance video,” *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3457-3464, June 2011.

[44] S. Oh, S. Russell, and S. Sastry, “Markov chain Monte Carlo data association for multi-target tracking,” *IEEE Trans. on Automatic Control*, vol. 54, no. 3, pp. 481-497, March 2009.

[45] A. Zamir, A. Dehghan, and M. Shah, “GMCP-tracker: Global multiobject tracking using generalized minimum clique graphs,” *in Proc. of European Conference on Computer Vision*, pp. 343-356, 2012.

[46] S. Gold and J. Rangarajan, “A graduated assignment algorithm for graph matching,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377-388, 1996.

[47] M. Leordeanu, M. Hebert, and R. Sukthankar “An integer projected fixed point method for graph matching and MAP inference,” *in Proc. of Annual Conference on Neural Information Processing Systems*, pp. 1114-1122, 2009.

[48] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” *in Proc. of IEEE International Conference on Computer Vision*, pp. 1482-1489, Oct. 2005.

[49] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," *in Proc. of Annual Conference on Neural Information Processing Systems*, pp. 313-320, 2007.

[50] M. Zaslavskiy, F. Bach, and J.-P. Vert, "A path following algorithm for the graph matching problem," *IEEE Trans. on Pattern analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2227-2242, 2009.

[51] Z. Liu, H. Qiao, X. Yang, and S. Hoi, "Graph matching by simplified convex-concave relaxation procedure," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 169-186, 2014.

[52] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.

[53] A. Sole-Ribalta and F. Serratosa, "Models and algorithms for computing the common labelling of a set of attributed graphs," *Computer Vision and Image Understanding*, vol. 115, no. 7, pp. 929-945, 2011.

[54] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *in Proc. of IEEE International Conference on Computer Vision*, pp. 2980-2988, 2017.

[55] E. Ristani, E. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *in Proc. of. European Conference on Computer Vision Workshops*, pp. 17-35, 2016.

[56] X. Shi, H. Ling, W. Hu, C. Yuan, and J. Xing, "Multi-target tracking with motion context in tensor power iteration," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3518-3525, 2014.

[57] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3539-3548, 2016.

[58] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple object tracking with high performance detection and appearance feature," *in Proc. of European Conference on Computer Vision Workshops*, pp. 36-42, 2016.

[59] B. Lee, E. Erdenee, S. Jin, M. Nam, Y. Jung, and P. Rhee, "Multi-class multi-object tracking using changing point detection," *in Proc. of European Conference on Computer Vision Workshops*, pp. 68-83, 2016.

[60] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," *in Proc. of IEEE International Conference on Computer Vision*, pp. 3029-3037, 2015.

[61] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *in Proc. of IEEE International Conference on Image Processing*, pp. 3645-3649, Sept. 2017.

[62] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *in Proc. of IEEE International Conference on Image Processing*, pp. 3464-3468, Sept. 2016.

[63] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," *in Proc. of IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1-6, Sept. 2017.

[64] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autogressive networks for online multi-object tracking," *in Proc. of IEEE Winter Conference on Applications of Computer Vision*, pp. 466-475, 2018.

[65] M. Leordeanu, A. Zanfir, and C. Sminchisescu, "Semi-supervised learning and optimization for hypergraph matching," *in Proc. of IEEE International Conference on Computer Vision*, pp. 2274-2281, Nov. 2011.

[66] Q. Nguyen, A. Gautier, and M. Hein, "A flexible tensor block coordinate ascent scheme for hypergraph matching," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5270-5278, 2015.

[67] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios, "Dense non-rigid surface registration using high-order graph matching," *in Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 382-389, 2010.