Rank-1 Tensor Approximation for High-Order Association in Multi-Target Tracking

Xinchu Shi, Haibin Ling, Yu Pang, Weiming Hu, Peng Chu, Junliang Xing

Abstract—High-order motion information is important in multi-target tracking (MTT) especially when dealing with large inter-target ambiguities. Such high-order information can be naturally modeled as a multi-dimensional assignment (MDA) problem, whose global solution is however intractable in general. In this paper, we propose a novel framework to the problem by reshaping MTT as a rank-1 tensor approximation problem (R1TA). We first show that MDA and R1TA share the same objective function and similar constraints. This discovery opens a door to use high-order tensor analysis for MTT and suggests the exploration of R1TA. In particular, we develop a tensor power iteration algorithm to effectively capture high-order motion information as well as appearance variation. The proposed algorithm is evaluated on a diverse set of datasets including aerial video sequences containing ariel borne dense highway scenes, top-view pedestrian trajectories, multiple similar objects, normal view pedestrians and vehicles. The effectiveness of the proposed algorithm is clearly demonstrated in these experiments.

Index Terms—Multi-target tracking, multi-dimensional assignment, rank-1 tensor approximation, data association

1 Introduction

Multiple target tracking (MTT) aims to locating targets and inferring their trajectories across a temporal sequence of video frames. An accurate and robust solution for MTT is crucial for many applications ranging from visual surveillance, human-computer interaction to computer-aided medical intervention. Research in MTT has a long history, with early works on radar and sonar target tracking [6], [11]. Recently, driven by the great progress in object detection [18], [24], tracking-by-detection [3] has gained popularity and attracted many researchers in MTT. In this paradigm, the targets in each frame are detected beforehand by either background subtraction or a pre-trained object detector. A data association procedure then joins the temporal detections into target trajectories. In this paper, we follow this paradigm and focus on target association.

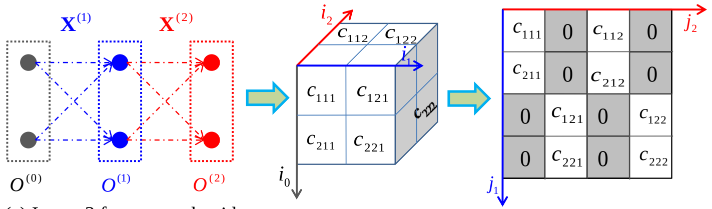
Despite recent advances in tracking-by-detection, accurate and robust tracking is still a challenging task. The task is relatively easy when targets are isolated or can be distinguished from each other. However, it is non-trivial in many practical scenarios such as with crowded targets, similar target appearances and fast motion. The ambiguities in these scenarios confuse association algorithms, especially when only pairs of neighbouring frames are considered.

A great amount of effort has been devoted toward reducing the ambiguities in order to improve tracking performance. Natural ways are to include appearance changes over time and to include high order spatial-temporal information, which encodes rich motion information. This suggests that association should be conducted in a "global" way beyond pairwise between-frame matching. MTT can be formulated as a multi-frame data association problem, which naturally leads to the *multi-dimensional assignment* (MDA) problem [16], [19], [42]. For two consecutive frames, the two-dimensional assignment, also known as the linear assignment or bipartite matching, is a special case of MDA and can be solved efficiently in polynomial time (e.g., by the Hungarian algorithm). In contrast, finding the global solution for MDA is usually intractable for three or higher dimensional cases. This drives researchers to seek alternative approximate methods such as semi-definite programming [48] and Lagrange relaxation [19]. Another way of tackling the problem is to introduce simplifying assumptions, for example to formulate association as a network flow problem. This is achieved by assuming that the global trajectory affinity can be decomposed into pairwise ones in a certain way as shown in [16]. Such network formulation has efficient solutions such as push-relabel [63] and successive shortest path [8], [41]. The price paid by these methods, however, is the limitation of using pair-wise affinity and thus the loss of rich high-order statistics such as the trajectory smoothness.

This work proposes a novel tensor-based framework for high-order association in MTT. The framework is inspired by our discovery of a close correlation between MDA and rank-1 tensor approximation. In particular, reshaping the affinity tensor and assignment variables of the original MDA, we reach a new formulation that is equivalent to the *rank-1 tensor approximation* (R1TA) problem in terms of objective functions. The key to the reformulation is to convert the traditional target-indexed affinity tensor to a (local) assignment-indexed one. An example is shown in Fig. 1.

This discovery opens the door of using R1TA algorithms for MTT, with some adaptations for handling special constraints. We design an efficient tensor power iteration solution by including the assignment constraints inherited from the MDA formulation. The iterative solution is com-

- *X. Shi and J. Xing are with the National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China.*
  W. Hu is with the CAS Center for Excellence in Brain Science and Intelligence Technology, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences; University of Chinese Academy of Sciences.
  E-mail: {xcshi, wmhu, jlxing}@nlpr.ia.ac.cn
- *H. Ling, Y. Pang and P. Chu are with the Department of Computer and Information Sciences, Temple University, Philadelphia, USA.*
  E-mail: {hbling, ypang, peng.chu}@temple.edu.



(a) Input: 3 frames, each with 2 targets ($K=2, I=2$)

(b) 3-rd order affinity tensor

(c) 2-nd order affinity tensor

Fig. 1. Affinity tensor reshaping. (a) Three frames, each with two objects to be associated. (b) The affinity tensor indexed by targets, i.e., $c_{i_0 i_1 i_2}$ denotes the affinity of the trajectory formed by the $\{i_0, i_1, i_2\}$-th targets in the three frames respectively. (c) The reshaped tensor indexed by the local (between-frame) assignment.

putationally efficient. A study of its convergence property is provided. The proposed framework, as summarized in Fig. 2, allows the integration of the high-order discriminative information into MTT in a principled way. Such information can be encoded by the trajectory affinity tensor (i.e., $\mathcal{C}$ and $\mathcal{A}$ in Fig. 1). This encoding is important, but has not been sufficiently investigated in previous research.

The implementation is based on a hierarchical association strategy [28] including the low-level and high-level procedures. In the low-level association, successive target detections are associated into tracklets, while in the high-level association, tracklets are further linked into long trajectories. The proposed R1TA-based power iteration is applied to both associations to make the tracking self-contained.

To summarize, the main contributions of this paper are: (1) the discovery of the close correlation between *multi-target tracking* (MTT) and *rank-1 tensor approximation* (R1TA), which provides a novel perspective for studying MTT; (2) an efficient tensor power iteration algorithm for solving the R1TA problem associated with MTT; and (3) a self-contained R1TA-based MTT framework.

To evaluate the proposed MTT algorithm, it is applied to many benchmark datasets involving diverse association and tracking scenarios: the CLIF dataset containing aerial video sequences of dense highway scenes [1], the PSU dataset [26] containing top-view pedestrian trajectories, the SMTT dataset [22] designed for evaluating the tracking of multiple similar objects, the widely used pedestrian tracking benchmark 2D MOT 2015 [32] and the vehicle tracking benchmark KITTI-Car [27]. The effectiveness of our algorithm compared with the state-of-the-art is clearly demonstrated in these experiments.

Some preliminary parts of this work were presented in CVPR'13 [50], however there are important extensions in this work. First, a more detailed and rigorous derivation of the MDA-R1TA equivalence is presented in this paper. Second, the high-level association is supplemented to make the approach self-contained. Finally, a more thorough experimental evaluation is carried out, in both benchmarks involved and state-of-the-art compared.

The paper is structured as follows: After discussing the related work in Section 2, we give an overview in Section 3. Then Section 4 highlights the close relation between MDA and the rank-1 tensor approximation. Section 5 presents the tensor power iteration solutions for the problem. In Section 6, the implementation details of the proposed algorithms are introduced. Finally, experimental results are presented in Section 7, and Section 8 concludes the whole paper.

2 Related Work

2.1 Multi-target tracking

Multiple target tracking has been an active research area for decades, and there are many related works. Early research interests focus on radar and sonar target tracking. There are many classic approaches such as multiple hypothesis tracking (MHT) [45] and joint probabilistic data association filter (JPDAF) [25]. More related work can be found in [6], [11], [17]. This paper focuses on the visual multi-target tracking. We review research that is closely related to this focus. A general survey can be referred to [35].

MTT algorithms can be divided into two groups: online and offline. Online tracking algorithms use only observations up to the current frame to estimate the current target states. There are two main categories of online tracking. One category includes the filtering-based methods [10], [12], [40], [45], [46]. These methods generally make the Markov assumption and use probabilistic inference for the target state estimation. The other category contains the sequential association based approaches, such as [4], [33], [43], for linking previous trajectories and current target detections. Online tracking algorithms are widely used in real time tracking applications. However, they typically suffer from the model drift, that is, the inability to recover from early errors.

The offline tracking exploits observations from previous and future frames to estimate the target states. This paradigm typically includes two steps, the first step is preprocessing, in which targets are located or extracted by a specific object detector or the motion detection procedure. The second step is generally an optimization process in the global temporal-spatial target state space. Our study falls in the offline tracking category and focuses on the multi-target association.

2.2 Multiple target association

Multiple target association across a batch of frames is popularly formulated as a multi-dimensional assignment (MDA) problem. As mentioned previously, the global solution to MDA is in general intractable and various approximate solutions have been proposed [19], [48]. By assuming the high-order trajectory affinity can be decomposed into the product of pair-wise ones, MDA can be formulated as a network flow problem. Following this direction, the network flow formulation is widely exploited in MTT [8], [14], [15], [33], [41], [57], [63]. The formulation has sophisticated solutions, such as by linear programming [29], push-relabel model [63], and successive shortest path (SSP) algorithms [8], [41]. The dynamic successive shortest path algorithm [14] solves the association by reusing the computation in the SSP problem. The solution is further optimized via a scheduling strategy to accommodate the online and memory bounded applications such as autonomous driving. The work [57] extends the min-cost flow with quadratic interactions between tracks to

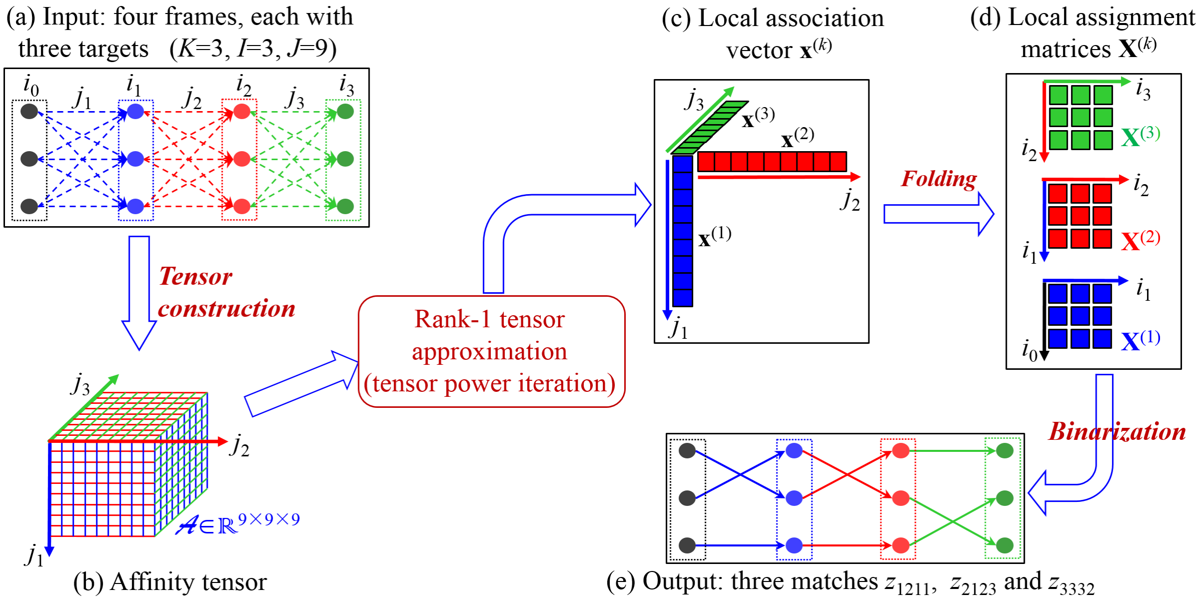

(a) Input: four frames, each with three targets (K=3, I=3, J=9)
i_0
j_1
i_1
j_2
i_2
j_3
i_3
Tensor construction
j_3
j_2
j_1
$\mathcal{A} \in \mathbb{R}^{9\times9\times9}$
(b) Affinity tensor
Rank-1 tensor approximation (tensor power iteration)
(c) Local association vector $\mathbf{x}^{(k)}$
j_3
$\mathbf{x}^{(3)}$
$\mathbf{x}^{(2)}$
j_2
$\mathbf{x}^{(1)}$
j_1
Folding
(d) Local assignment matrices $\mathbf{X}^{(k)}$
i_3
i_2
$\mathbf{X}^{(3)}$
i_2
i_1
$\mathbf{X}^{(2)}$
i_1
i_0
$\mathbf{X}^{(1)}$
Binarization
(e) Output: three matches z_{1211}, z_{2123} and z_{3332}

Fig. 2. The proposed multi-target tracking framework with a toy example. From an input image sequence of $K+1$ frames (a), a K-th order affinity tensor is first constructed (b). Then, the rank-1 tensor approximation algorithm is applied to the tensor to generate K local (between frame) association vectors (c). These vectors are reshaped to obtain local association matrices (d) and binarized to get the final tracking result (e).

capture the contextual cues. A structured prediction SVM is used to learn the tracking parameters. The network flow algorithms have many efficient solutions, but they are unable to explore high-order dynamic information.

High-order trajectory affinity is important in MTT since it provides global and discriminative information that may be neglected by the pairwise (local) affinities. High-order affinity is effective in modeling motion smoothness [38], [58], such as the constant velocity models defined on the frame triplets [13], [14], [16]. Moreover, a holistic affinity can be used to measure the appearance consistency of the cluster formed by the temporal targets in the trajectory hypothesis, as in [2], [62].

Using the affinity over a temporal window with three or more frames results in an NP-hard association problem, for which there are many sub-optimal approximation algorithms. Collins [16] proposes a block ICM-like method for high-order association. The method iteratively solves two-frame assignments in turn while keeping other assignment variables fixed. After that, Butt and Collins [14] formulate the association into a graph network that uses a third order association affinity. Lagrangian relaxation is employed to obtain a min-cost flow solution.

Zamir et al. [62] apply the general maximum clique partitioning (GMCP) technique to pick the best trajectory candidate iteratively, leading to a greedy and sub-optimal solution. A modified approach, the GMMCP tracker [20], is proposed to solve the joint optimization for all trackers simultaneously, but only for small or middle size problems. Milan et al. [37] perform data association in a continuous state space. The resulting complex non-convex optimization problem is solved locally by gradient descent augmented with heuristic discontinuous jumps. A more elegant discrete-continuous energy is later proposed in [38], where the tracking task is decomposed into two iterative optimization steps, i.e., data association and trajectory fitting. Long-term connectivity between pairs of detections is taken into consideration in [30]. The resulting graph is then solved in a conditional random field. In [5], a variational Bayesian model is introduced for tackling the varying number of targets during MTT. Sampling-based approaches (e.g. Markov Chain Monte Carlo techniques [7], [39], [61])) provide an alternative way of seeking for the global solution, though they typically require high computation costs for very high dimensional state estimation, and the parameter tuning is always a non-trivial task.

With recent emerging deep learning techniques, various deep neural network architectures have been employed in MTT [31], [36], [47], [53], [55], [56]. In [56], a deep convolution neural network is used for hierarchical deep feature learning and appearance affinity estimation. In [31], a two-stage learning scheme is proposed to solve pair-wise data association where the final matching probability is estimated through a trained siamese convolutional neural network and a gradient boosting classifier. In [36], the recurrent neural network is trained end-to-end for online multi-target tracking. In a network flow based approach [47], the pairwise cost functions used in the association are learned in an end-to-end fashion. In [53], a quadruplet architecture of deep neural network is proposed for metric learning, and the minimax label propagation is applied to the association. In [54], the temporal target detections are linked and clustered by solving a minimum cost subgraph multicut problem, and the approach is further extended by learning the pairwise feature based on DeepMatching [55].

3 Overview

In this section, we first formulate the multi-target tracking (MTT) problem as a multi-dimensional assignment (MDA) form, and then provide the intuition and overview of the proposed MTT framework.

3.1 Notations

To reduce the complexity of the formulas and derivations, various notations are used as summarized in Table 1. The notations for the lower order parts of any given structure are consistent. For example, the i-th entry of a vector $\mathbf{a}$ is denoted by a_i, the (i,j)-th entry of a matrix $\mathbf{A}$ by a_{ij} and the $(j_1, j_2, \dots, j_K)$-th entry of a K-th order tensor $\mathcal{A}$ by $a_{j_1:j_K}$. Moreover, when lower-case italic letters $(i, j, \dots)$ are used for indices in summation, they by default run from 1 to the value of the corresponding upper case variable $(I, J, \dots)$. For example, $\sum_i$ means $\sum_{i=1}^{I}$.

TABLE 1
Notations

Notation	Example	Meaning
italic	$a, A, \dots$	scalars
lower-case boldface	$\mathbf{a}, \mathbf{b}, \dots$	vectors
boldface capital	$\mathbf{A}, \mathbf{B}, \dots$	matrices
calligraphic	$\mathcal{A}, \mathcal{B}, \dots$	tensors
blackboard bold	$\mathbb{I}, \mathbb{J}, \dots$	sets
multi-dimensional index	$i_0 : i_K$	$i_0 i_1 \dots i_K$
summation over index sets	$\sum\limits_{\mathbb{I}=\{i_0,\dots,i_K\}}$	$\sum\limits_{i_0=1}^{I} \sum\limits_{i_1=1}^{I} \dots \sum\limits_{i_K=1}^{I}$
ℓ_2 norm of a vector	$\lvert\mathbf{x}\rvert$	$\lVert\mathbf{x}\rVert_2 = (\mathbf{x}^T\mathbf{x})^{1/2}$

3.2 Problem Formulation

Associating multiple targets between two frames can be treated as a *two-dimensional assignment* problem. As an extension, tracking over *multiple frames* can be viewed as a *multi-dimensional assignment* (MDA) [16], [42] problem.

In the rest of the paper, the input for MTT is denoted by $\mathbb{O} = \{\mathbb{O}^{(0)}, \mathbb{O}^{(1)}, \dots, \mathbb{O}^{(K)}\}$, which contains $K+1$ target sets extracted respectively from $K+1$ frames. Each set $\mathbb{O}^{(k)} = \{\mathbf{o}_1^{(k)}, \mathbf{o}_2^{(k)}, \dots, \mathbf{o}_I^{(k)}\}$ has I items to be matched or tracked. Note that to simplify the notation it is assumed that all frames have the same number of targets. This assumption does not affect the algorithm because the set can be padded with dummy targets as used for handling missing targets and false positives.

Given $\mathbb{O}$, it is necessary to find a high-order association that maximizes the overall trajectory affinity subject to the association constraints. In particular, we denote $c_{i_0:i_K} \doteq c_{i_0 i_1 \dots i_K}$ as the affinity for the trajectory composed by sequential targets $\{\mathbf{o}_{i_0}^{(0)}, \mathbf{o}_{i_1}^{(1)}, \dots, \mathbf{o}_{i_K}^{(K)}\}$; $z_{i_0:i_K} \doteq z_{i_0 i_1 \dots i_K}$ indicates whether the trajectory is true (=1) or not (=0); $\mathcal{C} = (c_{i_0:i_K})$ and $\mathcal{Z} = (z_{i_0:i_K})$ are the corresponding $(K+1)$-th order tensors. Fig. 1(a-b) gives a toy example of the problem. Using these notations, MTT is formulated as a $(K+1)$-dimensional assignment problem as follows

$$\arg\max_{\mathcal{Z}} \; f_z(\mathcal{Z}) = \sum_{\mathbb{I}} c_{i_0:i_K} z_{i_0:i_K} = \lVert \mathcal{C} \circ \mathcal{Z} \rVert_1 \, , \tag{1}$$

$$\text{s.t.} \begin{cases} \sum\limits_{\mathbb{I} \setminus \{i_k\}} z_{i_0:i_K} = 1, & \forall k = 0, 1, \dots, K \\ z_{i_0:i_K} \in \{0, 1\}, & \forall i_k = 1, 2, \dots, I. \end{cases} \tag{2}$$

where $\mathbb{I} \doteq \{i_0, i_1, \dots, i_K\}$ is the target index set, '$\circ$' the Hadamard product (element-wise product), '$\setminus$' the set difference, $\lVert \cdot \rVert_1$ the matrix 1-norm (note that both $\mathcal{C}$ and $\mathcal{Z}$ are non-negative), and $\sum\limits_{\mathbb{I}}$ summation over an index set defined as in Table 1.

The above integer assignment problem is generally intractable. It is typically relaxed to a real valued version before the final binarization. We follow this strategy, and by default refer to the relaxed version in the rest of the paper.

3.3 Intuition and Overview of the Framework

The key intuition is to factorize an affinity tensor to a sequence of *local tracking*[1] along the temporal direction. Throughout this paper, *local tracking* is used to indicate target matching between two neighbor frames. In contrast, *global tracking* is used for the association across multiple ($K+1$ in this formulation) frames. While local tracking can be naturally represented by (soft) assignment matrices, the matrix representations are inappropriate for tensor factorization. Instead the vector form of local tracking is used for this task.

In the next section it is shown that the MTT formulation in (1) is equivalent to approximating the affinity tensor (after reshaping) with K local tracking vectors. Based on this idea, an MTT tracking framework is proposed with three major steps:

1) **tensor construction**: a K-th order affinity tensor $\mathcal{A}$ is constructed by reshaping from the original affinity tensor $\mathcal{C}$;
2) **rank-1 tensor approximation**: approximate $\mathcal{A}$ with a sequence of local tracking vectors, denoted by $\mathbb{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(K)}\}$; and
3) **binarization**: binarize the soft local matching vectors to get the final solution via local two-dimensional assignment.

The framework is summarized in Fig. 2 and more details are provided in the following sections.

4 Multi-Target Tracking and Rank-1 Tensor Approximation

4.1 Reformulate MDA

Two issues arise when we connect the $(K+1)$-th order affinity tensor $\mathcal{C}$ with K between-frame local tracking matrices: (1) while it is convenient to use vectors in tensor factorization, MTT between two consecutive frames is essentially a soft assignment matrix; and (2) while there are K two-frame associations, the affinity tensor $\mathcal{C}$ is of order $K+1$. These two issues are addressed by reshaping the original MDA formulation in (1), and we then correlate the new formulation with tensor factorization in the next subsection.

4.1.1 Reshape Global Tracking and Affinity

The local tracking between frames $\mathbb{O}^{(k-1)}$ and $\mathbb{O}^{(k)}$ is described by a *local (assignment) matrix* $\mathbf{X}^{(k)} = (x_{i_{k-1}i_k}^{(k)}) \in \mathbb{R}^{I \times I}$, where $x_{i_{k-1}i_k}^{(k)}$ links the targets $\mathbf{o}_{i_{k-1}}^{(k-1)}$ and $\mathbf{o}_{i_k}^{(k)}$. To address

[1] The local tracking denotes the two-frame association. The terms association, tracking and matching are used interchangeably depending on context.

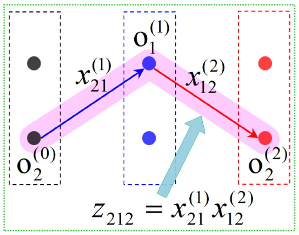


Fig. 3. Reshaping of the global tracking, $z_{212} = x_{21}^{(1)} x_{12}^{(2)} = x_2^{(1)} x_3^{(2)}$.

the first issue above, $\mathbf{X}^{(k)}$ is vectorised to produce a *local (assignment) vector* denoted by $\mathbf{x}^{(k)} = (x_{j_k}^{(k)}) \in \mathbb{R}^J$, where $J \doteq I^2$. For notational convenience, the same scalar symbol x is used for entries in both $\mathbf{X}$ and $\mathbf{x}$ with double subscripts and a single subscript respectively.

Using the above notation, an assignment variable $z_{i_0:i_K}$ is decomposed as

$$z_{i_0:i_K} = x_{i_0 i_1}^{(1)} x_{i_1 i_2}^{(2)} \dots x_{i_{K-1} i_K}^{(K)} = x_{j_1}^{(1)} x_{j_2}^{(2)} \dots x_{j_K}^{(K)} \ , \tag{3}$$

where $j_k = (i_k - 1) \times I + i_{k-1}$ is the index for $\mathbf{x}^{(k)}$ corresponding to the matrix indices (i_{k-1}, i_k). A toy example of a decomposition z_{212} is shown in Fig. 3. The factorization of the global tracking variable $z_{i_0:i_K}$ into the product of local tracking $x_{i_{k-1} i_k}^{(k)}$ is validated by the fact that the global trajectory hypothesis is true if and only if all local associations along it are true. Such decomposition has been widely used in MTT such as in [8], [41], [63].

Let $J = I \times I$ be the number of two-frame association candidates. The second issue is addressed by constructing a K-th order affinity tensor $\mathcal{A} = (a_{j_1:j_K}) \in \mathbb{R}^{J \times J \times \dots \times J}$ from the original $(K+1)$-th order affinity tensor $\mathcal{C}$, and

$$a_{j_1:j_K} = \begin{cases} c_{\overline{j_1}:\overline{j_K}\underline{j_K}}, & \text{if } \underline{j_k} = \overline{j_{k+1}}, \ \forall k = 1, \dots, K-1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\overline{j_k}, \underline{j_k}$ indicate respectively the row and column indices when converting j_k to matrix indices, i.e.,

$$\underline{j_k} = \text{ceil}(j_k / I) \ , \quad \overline{j_k} = j_k - (\underline{j_k} - 1) \times I .$$

Fig. 1 illustrates the conversion from a $2 \times 2 \times 2$ affinity tensor $\mathcal{C}$ to a 4×4 affinity tensor $\mathcal{A}$.

By such reshaping, each element $c_{i_0:i_K}$ in $\mathcal{C}$ has a corresponding element with the same value in the augmented tensor $\mathcal{A}$. Conversely, each non-zero element in $\mathcal{A}$ is copied from an element in $\mathcal{C}$. In this way, the original $(K+1)$-th order tensor $\mathcal{C} \in \mathbb{R}^{I \times I \times \dots \times I}$ is converted to the new K-th order sparse tensor $\mathcal{A} \in \mathbb{R}^{J \times J \times \dots \times J}$ without loss of information.

4.1.2 New MDA Formulation

In this subsection a new MDA formulation is derived using the reshaped local tracking and new affinities introduced in (3) and (4). There are important connections between the original affinity tensor $\mathcal{C}$ (global tracking tensor $\mathcal{Z}$) and the new tensor $\mathcal{A}$ (local tracking vectors $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(K)}\}$):

1) For each *global tracking* $z_{i_0:i_K}$ and its affinity $c_{i_0:i_K}$, there are corresponding new formulations as $x_{j_1}^{(1)} x_{j_2}^{(2)} \dots x_{j_K}^{(K)}$ and $a_{j_1:j_K}$ respectively. In this way, the element-wise products $c_{i_0:i_K} z_{i_0:i_K}$ in the original MDA optimization (1) are reshaped as $a_{j_1:j_K} x_{j_1}^{(1)} x_{j_2}^{(2)} \dots x_{j_K}^{(K)}$.
2) The corresponding relation between the new assignment (affinity) representation $x_{j_1}^{(1)} x_{j_2}^{(2)} \dots x_{j_K}^{(K)}$ $(a_{j_1:j_K})$ and the old representation $z_{i_0:i_K}$ $(c_{i_0:i_K})$ is not bijective. Some $x_{j_1}^{(1)} x_{j_2}^{(2)} \dots x_{j_K}^{(K)}$ $(a_{j_1:j_K})$ may not have corresponding $z_{i_0:i_K}$ $(c_{i_0:i_K})$.
3) The new representation $a_{j_1:j_K}$ has no counterpart in the old one $c_{i_0:i_K}$ if some interconnection condition $\underline{j_k} = \overline{j_{k+1}}$ in (4) is invalid. Suppose that $\underline{j_k} \neq \overline{j_{k+1}}$. Then the neighbor *local tracking* $x_{j_k}^{(k)}$ and $x_{j_{k+1}}^{(k+1)}$ share no common target in frame $\mathbb{O}^k$, thus there is neither feasible *global tracking* nor real affinity. Such redundant and infeasible elements in $\mathcal{A}$ are masked by 0, and have no influence on the optimization.

These connections enable the conversion of the original MDA optimization (1) into the following equivalent one:

$$\arg\max_{\mathbb{X}} \ f(\mathbb{X}) = \sum_{\mathbb{J}} a_{j_1:j_K} x_{j_1}^{(1)} x_{j_2}^{(2)} \dots x_{j_K}^{(K)} \ , \tag{5}$$

$$\text{s.t.} \begin{cases} \sum\limits_{i_{k-1}} x_{i_{k-1} i_k}^{(k)} = 1, & \forall k = 1, \dots, K \\ \sum\limits_{i_k} x_{i_{k-1} i_k}^{(k)} = 1, & \forall k = 1, \dots, K \\ 0 \leq x_{j_k}^{(k)} \leq 1, & \forall k = 1, \dots, K; \\ & j_k = 1, \dots, J \end{cases} \tag{6}$$

where $\mathbb{X} \doteq \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(K)}\}$ is the set of local tracking vectors, and $\mathbb{J} \doteq \{j_1, j_2, \dots, j_K\}$ is the set of assignment indices. As well as transiting from global tracking tensor to a sequence of local tracking vectors, the new MDA formulation relaxes the integer variables to the real variables.

4.2 Equivalence between Multi-Target Tracking and Rank-1 Tensor Approximation

This subsection contains some preliminaries on tensor algebra. Afterwards, we show the close relation between multi-dimensional assignment and rank-1 tensor approximation: the two optimizations have the same objective function with smart reformulations and relaxations.

4.2.1 Tensor Preliminaries

For a K-th order tensor $\mathcal{A} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_K}$, whose element is $a_{j_1:j_K}$ and $1 \leq j_k \leq J_k$, $k \in \{1, \dots, K\}$, each dimension of the tensor is referred to as a *mode*. Tensors have operations similar to matrix-vector and matrix-matrix multiplication, as defined below.

Definition 1: The *k-mode* product of a K-th order tensor $\mathcal{A} \in \mathbb{R}^{J_1 \times \dots J_{k-1} \times J_k \times J_{k+1} \times \dots J_K}$ and a matrix $\mathbf{X} \in \mathbb{R}^{J_k \times M}$ is a new K-th order tensor $\mathcal{B} \in \mathbb{R}^{J_1 \times \dots J_{k-1} \times M \times J_{k+1} \times \dots J_K}$,

$$\mathcal{B} = \mathcal{A} \times_k \mathbf{X} \ , \tag{7}$$

and the element $b_{j_1 \dots j_{k-1} m j_{k+1} \dots j_K}$ in $\mathcal{B}$ is computed as

$$b_{j_1 \dots j_{k-1} m j_{k+1} \dots j_K} = \sum\nolimits_{j_k} a_{j_1 \dots j_{k-1} j_k j_{k+1} \dots j_K} x_{j_k m} . \tag{8}$$

In particular, the *k-mode* product of the tensor $\mathcal{A}$ and a vector $\mathbf{x} \in \mathbb{R}^{J_k}$, denoted by $\mathcal{A} \times_k \mathbf{x}$, is a $(K-1)$-th order tensor

$$(\mathcal{A} \times_k \mathbf{x})_{j_1 \dots j_{k-1} j_{k+1} \dots j_K} = \sum\nolimits_{j_k} a_{j_1 \dots j_{k-1} j_k \dots j_K} x_{j_k} . \tag{9}$$

With the above definition, the optimization (5) can be formulated as the following tensor-vector product,

$$\arg\max_{\mathbb{X}} \ f(\mathbb{X}) = \mathcal{A} \times_1 \mathbf{x}^{(1)} \times_2 \mathbf{x}^{(2)} \cdots \times_K \mathbf{x}^{(K)}. \tag{10}$$

Next, a close relation is established between the above optimization (10) and the rank-1 tensor approximation problem.

4.2.2 Rank-1 Tensor Approximation

If a tensor $\mathcal{A} \in \mathbb{R}^{J_1 \times \ldots J_k \times \ldots J_K}$ can be computed as the outer product of K vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}, \ldots, \mathbf{x}^{(K)}$ $\left(\mathbf{x}^{(k)} \in \mathbb{R}^{J_k}, 1 \leq k \leq K\right)$, we call $\mathcal{A}$ a rank-1 tensor. Formally, a rank-1 tensor is defined as follows.

Definition 2: A tensor $\mathcal{A} \in \mathbb{R}^{J_1 \times \ldots J_k \times \ldots J_K}$ is a rank-1 tensor, if and only if there exist a set of K vectors $\{\mathbf{x}^{(k)} = (x^{(k)}_{j_k}) \in \mathbb{R}^{J_k}\}_{k=1}^{K}$ such that

$$\mathcal{A} = \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \cdots \otimes \mathbf{x}^{(K)} \ , \tag{11}$$

where $\otimes$ denotes the outer product operator, and

$$\left(\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \cdots \otimes \mathbf{x}^{(K)}\right)_{j_1 j_2 \ldots j_K} \doteq x^{(1)}_{j_1} x^{(2)}_{j_2} \ldots x^{(K)}_{j_K}. \tag{12}$$

The problem of rank-1 approximation of a general tensor $\mathcal{A}$ is formulated as:

Problem 1: (Rank-1 Tensor Approximation (R1TA)) Given a real K-th order tensor $\mathcal{A} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_K}$, find K unit-norm vectors $\mathbb{X} = \{\mathbf{x}^{(k)} = (x^{(k)}_{j_k}) \in \mathbb{R}^{J_k}\}_{k=1}^{K}$ and a scalar λ to minimize the reconstruction error in terms of the square of the Frobenius norm

$$\begin{aligned} h(\lambda, \mathbb{X}) &= \left\| \mathcal{A} - \lambda \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \cdots \otimes \mathbf{x}^{(K)} \right\|_F^2 \\ &= \sum_{\mathbb{J}} \left(a_{j_1:j_K} - \lambda x^{(1)}_{j_1} x^{(2)}_{j_2} \ldots x^{(K)}_{j_K} \right)^2. \end{aligned} \tag{13}$$

Problem 1 can be solved by various techniques such as Lagrange multipliers [21] or least-squares [44]. With some derivations [21], [44], the minimization (13) has the following equivalent form

$$\begin{aligned} &\arg\min_{\mathbb{X}} \left(\min_{\lambda} \left\| \mathcal{A} - \lambda \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \cdots \otimes \mathbf{x}^{(K)} \right\|_F^2 \right) \\ = &\arg\min_{\mathbb{X}} \left(\|\mathcal{A}\|_F^2 - \left| \mathcal{A} \times_1 \mathbf{x}^{(1)} \times_2 \mathbf{x}^{(2)} \ldots \times_K \mathbf{x}^{(K)} \right|^2 \right). \end{aligned} \tag{14}$$

The proof of (14) can be found in [21], [44]. The result naturally leads to the following theorem:

Theorem 1: The minimization of (13) over the unit-norm vectors $\mathbb{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}\}$ is equivalent to maximizing the following function

$$\begin{aligned} g(\mathbb{X}) &= \left| \mathcal{A} \times_1 \mathbf{x}^{(1)} \times_2 \mathbf{x}^{(2)} \cdots \times_K \mathbf{x}^{(K)} \right|^2 \\ &= \left| \sum_{\mathbb{J}} a_{j_1:j_K} x^{(1)}_{j_1} x^{(2)}_{j_2} \ldots x^{(K)}_{j_K} \right|^2 . \end{aligned} \tag{15}$$

$$\text{s.t.} \ \left\| \mathbf{x}^{(k)} \right\|_2^2 = 1, \quad \forall k = 1, \ldots, K \tag{16}$$

4.2.3 From MTT to Rank-1 Tensor Approximation

It is now possible to connect the MDA in MTT with R1TA. It is obvious that the objective function $f(\mathbb{X})$ in the reformulated MDA (10) looks very similar to the objective function $g(\mathbb{X})$ in the R1TA form (15) derived in Theorem 1. The only difference is that $f(\mathbb{X})$ takes plain summation while $g(\mathbb{X})$ uses the squared norm. Fortunately, this difference can be ignored because (1) $\mathcal{A}$ and $\mathbb{X}$ in our problem are guaranteed to be non-negative, and (2) the optimum X is required, therefore the square operation can be removed.

Consequently, the following key equivalence is established:

$$\arg\max_{\mathbb{X}} f(\mathbb{X}) = \arg\max_{\mathbb{X}} g(\mathbb{X}) \ . \tag{17}$$

This equivalence suggests that MTT can be treated as an R1TA problem. This opens a new way for developing MTT algorithms. Despite the equivalence in the objective functions, we note that the constraints over $\mathbb{X}$ in MTT are different from those in the classic R1TA. More specifically, $\mathbb{X}$ in MTT requires the binary variables and is subject to assignment constraints (ℓ_1 or row/column ℓ_1 unit norm), while vectors of $\mathbb{X}$ in R1TA require ℓ_2 unit norm. In the next section we propose a row/column ℓ_1 unit norm tensor power iteration for optimization (10).

An example illustrating the relation between rank-1 tensor approximation and MDA is given in Fig. 2. An affinity tensor $\mathcal{A}$ is constructed based on the global tracking (i.e., trajectory) affinities. Each potential trajectory has a corresponding affinity which is stored as a tensor element. The local tracking vectors generated by R1TA over $\mathcal{A}$ are viewed as the real valued relaxation of the local assignment (i.e., two-frame association) variables. Intuitively, R1TA minimizes the element-wise errors between the trajectory affinity tensor and the reconstructed tensor, which is calculated as the outer product of the approximate vectors. In particular, for a global trajectory with a high affinity, the optimization searches for a sequence of local association vectors so that their outer product along the trajectory matches the high affinity. Consequently, the higher the affinity a trajectory has, the more likely its local association components will be picked up in the final solution. This intuition justifies the underlying rationale for formulating MTT as an R1TA problem.

It is noted that Rank-1 tensor approximation has connections with variational inference in which high-order graphical models are approximated as the product of tractable factors. Both of them aim to the optimization of some objectives, and take the factorization representation to approximate the complex high-dimensional (high-order) elements. However, there are two important differences between them. First, the former is the approximation of the known tensor and the latter is the approximation of the unknown probability distribution. Second, the optimization solutions are different. The former formulates the problem into the tensor methodology, while the latter resorts to the probability theory and approximates the objective distribution to the simplified distribution.

5 Tensor Power Iteration

The MTT-R1TA equivalence allows us to borrow R1TA algorithms for MTT, as summarized in the flow chart in Fig. 2. One of the most popular algorithms for solving R1TA is the tensor power iteration algorithm [21], which was derived as the least squares solution to the R1TA problem.

Algorithm 1 Tensor power iteration with ℓ_1 unit norm

```
Input: $K$-th order tensor $\mathcal{A} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$.
Output: $\ell_1$ unit norm vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)}$.
Initialize $\mathbf{x}^{(1)(0)}, \ldots, \mathbf{x}^{(K)(0)}$
$n = 0$
repeat
    $\widehat{\mathbf{x}}^{(1)(n+1)} = \left(\mathcal{A} \times_2 \mathbf{x}^{(2)(n)} \cdots \times_K \mathbf{x}^{(K)(n)}\right) \circ \mathbf{x}^{(1)(n)}$
    $\mathbf{x}^{(1)(n+1)} = \widehat{\mathbf{x}}^{(1)(n+1)} / \left\|\widehat{\mathbf{x}}^{(1)(n+1)}\right\|_1$
    $\vdots$
    $\widehat{\mathbf{x}}^{(r)(n+1)} = \left(\mathcal{A} \ldots \times_{r-1} \mathbf{x}^{(r-1)(n+1)} \times_{r+1} \mathbf{x}^{(r+1)(n)} \ldots \times_K \mathbf{x}^{(K)(n)}\right) \circ \mathbf{x}^{(r)(n)}$
    $\mathbf{x}^{(r)(n+1)} = \widehat{\mathbf{x}}^{(r)(n+1)} / \left\|\widehat{\mathbf{x}}^{(r)(n+1)}\right\|_1$
    $\vdots$
    $\widehat{\mathbf{x}}^{(K)(n+1)} = \left(\mathcal{A} \times_1 \mathbf{x}^{(1)(n+1)} \ldots \times_{K-1} \mathbf{x}^{(K-1)(n+1)}\right) \circ \mathbf{x}^{(K)(n)}$
    $\mathbf{x}^{(K)(n+1)} = \widehat{\mathbf{x}}^{(K)(n+1)} / \left\|\widehat{\mathbf{x}}^{(K)(n+1)}\right\|_1$
    $n = n + 1$
until convergence
return $\mathbf{x}^{(k)} = \mathbf{x}^{(k)(n)}, k = 1, 2, \ldots, K$
```

The original tensor power iteration algorithm cannot be directly applied to the R1TA formulation because the constraints are different. In the following, we first describe an ℓ_1 unit norm tensor power iteration algorithm for solving the optimization (15), and then design a power iteration solution for the optimization (10) with row/column constraints (6).

5.1 ℓ_1 Unit Norm Power Iteration

We assume all tensor elements $a_{j_1:j_K}$ and the approximation vectors $\mathbf{x}^{(k)}, k = 1, 2, \ldots, K$, are non-negative. Thus, the optimization (15) with the ℓ_1 norm constraint is formulated as

$$\max_{\mathbb{X}} g(\mathbb{X}) = \max_{\mathbb{X}} \sum_{\mathbb{J}} a_{j_1:j_K} x^{(1)}_{j_1} x^{(2)}_{j_2} \ldots x^{(K)}_{j_K}, \tag{18}$$

$$\text{s.t.} \begin{cases} \sum_{j_k} x^{(k)}_{j_k} = 1, \quad k = 1, 2, \ldots, K. \\ 0 \leq x^{(k)}_{j_k} \leq 1, \quad k = 1, 2, \ldots, K. \\ \qquad\qquad\quad j_k = 1, 2, \ldots, J. \end{cases} \tag{19}$$

A block-update iteration algorithm is proposed. Each block $\mathbf{x}^{(k)}$ is updated in turn. Denote the vector $\mathbf{x}^{(k)}$ after the n-th iteration as $\mathbf{x}^{(k)(n)}$ with elements $x^{(k)(n)}_{j_k}$. The iteration for block $\mathbf{x}^{(1)(n)}$ is

$$x^{(1)(n+1)}_{j_1} = \frac{x^{(1)(n)}_{j_1}}{C^{(1)(n)}} \sum_{\mathbb{J} \backslash \{j_1\}} a_{j_1:j_K} x^{(2)(n)}_{j_2} \ldots x^{(K)(n)}_{j_K}, \tag{20}$$

where $C^{(1)(n)} = \sum_{\mathbb{J}} a_{j_1:j_K} x^{(1)(n)}_{j_1} \ldots x^{(K)(n)}_{j_K}$ is the ℓ_1 normalization factor. The iterations of other block vectors $\mathbf{x}^{(k)} (k \neq 1)$ have similar formulations as in Eq. (20). The complete tensor power iteration solution for optimization (18) is shown in Algorithm 1.

The power iteration algorithm for classic R1TA can be derived as the least squares solution [21], [44], which has been proved to be convergent. Our ℓ_1 unit norm version is inspired by and similar to that proposed in [23]. In the following, we prove its convergence property. First, we have the following proposition:

Proposition 1: For an iteration in (20), we have

$$\begin{aligned} & g(\mathbf{x}^{(1)(n+1)}, \mathbf{x}^{(2)(n)}, \ldots, \mathbf{x}^{(K)(n)}) \\ \geq\ & g(\mathbf{x}^{(1)(n)}, \mathbf{x}^{(2)(n)}, \ldots, \mathbf{x}^{(K)(n)}). \end{aligned} \tag{21}$$

Proof. We first make two auxiliary vectors $\mathbf{w} = (w_1, \ldots, w_{J_1})^T$ and $\mathbf{u} = (u_1, \ldots, u_{J_1})^T$ by

$$\begin{cases} w_{j_1} = \sum_{\mathbb{J} \backslash \{j_1\}} a_{j_1:j_K} x^{(2)(n)}_{j_2} \ldots x^{(K)(n)}_{j_K}, \\ u_{j_1} = \sqrt{x^{(1)(n)}_{j_1}}. \end{cases} \tag{22}$$

With above notations, the objective in the optimization (18) is computed as

$$\begin{aligned} & g(\mathbf{x}^{(1)(n)}, \mathbf{x}^{(2)(n)}, \ldots, \mathbf{x}^{(K)(n)}) \\ & = \sum_{\mathbb{J}} a_{j_1:j_K} u_{j_1} u_{j_1} x^{(2)(n)}_{j_2} \ldots x^{(K)(n)}_{j_K} \\ & = \left\langle \mathbf{u}, \mathbf{u} \circ \mathbf{w} \right\rangle, \end{aligned} \tag{23}$$

where "$\langle \cdot \rangle$" denote the inner product. With the norm constraint $\|\mathbf{u}\|_2^2 = \left\|\mathbf{x}^{(1)(n)}\right\|_1 = 1$, the Cauchy-Schwarz inequality gives

$$\begin{aligned} g(\mathbf{x}^{(1)(n)}, \mathbf{x}^{(2)(n)}, \ldots, \mathbf{x}^{(K)(n)}) & = \langle \mathbf{u}, \mathbf{u} \circ \mathbf{w} \rangle \\ & \leq |\mathbf{u}| \, |\mathbf{u} \circ \mathbf{w}| \\ & = |\mathbf{u} \circ \mathbf{w}|. \end{aligned} \tag{24}$$

With the iteration formulation (20), there is

$$\begin{aligned} & g(\mathbf{x}^{(1)(n+1)}, \mathbf{x}^{(2)(n)}, \ldots, \mathbf{x}^{(K)(n)}) \\ & = \langle \mathbf{x}^{(1)(n+1)}, \mathbf{w} \rangle \\ & = \frac{1}{C^{(1)(n)}} \langle \mathbf{x}^{(1)(n)} \circ \mathbf{w}, \mathbf{w} \rangle \\ & = \frac{1}{C^{(1)(n)}} \langle \mathbf{u} \circ \mathbf{w}, \mathbf{u} \circ \mathbf{w} \rangle \\ & = \frac{|\mathbf{u} \circ \mathbf{w}|^2}{g\left(\mathbf{x}^{(1)(n)}, \mathbf{x}^{(2)(n)}, \ldots, \mathbf{x}^{(K)(n)}\right)}. \end{aligned} \tag{25}$$

The inequality (21) is proved by combining formulation (24) and (25). Suppose that the maximum value across the tensor elements is $a_{j_1^*:j_K^*}$, the maximum of the optimization objective in (18) is not larger than $a_{j_1^*:j_K^*}$. Therefore, the objective is bounded. ■

The convergence proofs for the other block iterations have similar derivations and are therefore skipped. Since each iteration update leads to a better score, the proposed algorithm converges.

5.2 Row/Column ℓ_1 Unit Norm Power Iteration

The power iteration in Algorithm 1 deals with ℓ_1 unit norm constraints. However, the MDA optimization (1) requests row/column ℓ_1 unit norm constraints. To solve this constrained optimization problem in MDA, we design a new power iteration algorithm, listed as Algorithm 2, which extends Algorithm 1 with different normalization steps. Specifically, the new power iteration solution uses the same block iteration as Algorithm 1 by using different normalization step to accommodate the row/column ℓ_1 unit norm

Algorithm 2 Tensor power iteration with row/column ℓ_1 unit norm

```
1:  Input: the affinity tensor $\mathcal{A} : a_{j_{1:j_K}}$.
2:  Output: assignment matrices $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}$.
3:  Initialize $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(K)}$
4:  repeat
5:      for $k = 1, \dots, K$ do
6:          for $j_k = 1, \dots, J_k$ do
7:              $x^{(k)}_{j_k} = x^{(k)}_{j_k} \sum_{\mathbb{J}\setminus\{j_k\}} a_{j_{1:K}} x^{(1)}_{j_1} \dots x^{(K)}_{j_K}$.
8:          end for
9:          repeat
10:             for $i_{k-1} = 1, \dots, I$ do
11:                 $C = \sum_{i_k} x^{(k)}_{i_{k-1}i_k}$
12:                 $x^{(k)}_{i_{k-1}i_k} = x^{(k)}_{i_{k-1}i_k}/C, \quad i_k = 1, 2, \cdots, I$
13:             end for
14:             for $i_k = 1, \dots, I$ do
15:                 $C = \sum_{i_{k-1}} x^{(k)}_{i_{k-1}i_k}$
16:                 $x^{(k)}_{i_{k-1}i_k} = x^{(k)}_{i_{k-1}i_k}/C, \quad i_{k-1} = 1, 2, \cdots, I$
17:             end for
18:         until convergence
19:     end for
20: until convergence or maximum number of iterations
21: Reshape $\mathbf{x}^{(k)}$ to $\mathbf{X}^{(k)}, k = 1, 2, \dots, K$
22: Discretize $\mathbf{X}^{(k)}, k = 1, 2, \dots, K$, using the Hungarian algorithm
23: return $\mathbf{X}^{(k)}, k = 1, 2, \dots, K$
```

constraint. With the alternative row and column normalization, a local assignment matrix converges to a doubly stochastic matrix according to the Sinkhorn's theorem [52]. The last step in Algorithm 2 is the Hungarian assignment to discretize the assignment variables.

Unfortunately, we currently can not give the convergence proof for Algorithm 2. The difficulty is caused by the alternative column and row normalization, which is presented as line 9∼18 in Algorithm 2. The repeated column/row normalizations complicate the solution. Though no theoretical proof is available, the optimization objective appears to converge in all the experimental validations, which will be detailed in the experiments.

There are several important observations to be made about the proposed algorithm. Firstly, the hard assignment in the original MDA task is relaxed into the soft assignment in the R1TA formulation. The real valued R1TA solution can be interpreted as a set of association probabilities. Secondly, the association ambiguity still exists, but it is reduced during the proposed power iteration. Finally, the key iteration in the Algorithm 2 is step 7, in which a local association decision $x^{(k)}_{j_k}$ is influenced by all the possible global trajectories passing through $x^{(k)}_{j_k}$. A benefit of such an iteration is the encoding of the powerful high order information carried by the trajectory affinities. Such high order information provides more discriminative semantic support than its pairwise counterpart.

6 Application to Multi-Target Tracking

To apply the proposed R1TA algorithms for MTT, several components need to be defined, including hypothesis generation, affinity definition, initialization and termination. This section describes these components in details.

TABLE 2
Symbols in MTT

Symbols	Meaning
$\mathbf{o}_i^{(k)}$	$i-$th target in the $k-$th frame
$\mathbf{a}_i^{(k)}, \mathbf{p}_i^{(k)}, t_i^{(k)}$	Appearance, position and time of $\mathbf{o}_i^{(k)}$
$\tau_i : \{\mathbf{o}_i^{(s_i)}, ..., \mathbf{o}_i^{(e_i)}\}$	A tracklet hypothesis
$\tau_0(\tau_\infty)$	The virtual source (sink) tracklet
x_{ij}	The soft association value between τ_i and τ_j
$t_i^{(s_i)}(t_i^{(e_i)})$	The start (end) time of τ_i
$\mathbf{z}_i^{(k)}=\mathbf{p}_i^{(k)}-\mathbf{p}_i^{(k-1)}$	The velocity of τ_i at frame k
$\mathbb{T} : \{\tau_1, \tau_2, ..., \tau_N\}$	The tracklet set
Ψ_i^j	Trajectory hypotheses passing through local tracklet association $\tau_i \to \tau_j$
Γ	The set of all trajectory hypotheses
a_l	The affinity of the l-th trajectory $\Gamma(l)$
A_l	Local associations in $\Gamma(l)$. For instance, $A_l = \{x_{12}, x_{23}\}$ for $\Gamma(l) = \{\tau_1, \tau_2, \tau_3\}$
T_0	Time threshold of tracklet association

Before getting into details, note that some symbols used in the following sections are listed in Table 2.

6.1 Hierarchical association framework

For the tracking application, the widely used hierarchical association framework is applied. In the two-level associations, raw detections are first linked into short tracklets, which are then associated to obtain long trajectories.

In the low-level association, a batch mode is used. Each batch contains 4∼15 frames depending on the number of tracked targets. Any two consecutive batches contain a common frame for tracklet growing. The tensor power iteration presented in Algorithm 2 is used in this stage.

The high-level association is different to and more difficult than its low-level counterpart. One could imagine the basic tracklet as the target detection and follow up the same association used in the low-level process, yet the tracklets are not as temporally homogeneous as the detections. This is a known issue in high-level tracklet association.

Inspired by the tensor power iteration algorithm, we propose a similar approach to high-level association. First, for each tracklet all possible trajectory candidates passing through it are counted. In this way, we can obtain all trajectory candidates and compute the affinity tensor. Second, for each tracklet-tracklet association candidate, we compute the affinity summation weighted by association values of all trajectories passing through this local association. Finally, the alternative row and column normalization is applied to obtain the constrained tracklet-tracklet association probability. With the symbols presented in Table 2, the proposed high-level tracklet association algorithm is presented as Algorithm 3.

An illustration is presented in Fig. 4. In the example, there are five real tracklets. For tracklet τ_1, the forward association set is $\Phi_1 = \{2, 3, \infty\}$, which means τ_1 can be associated with τ_2, τ_3 and τ_∞. In this case, the first association candidate is $\Phi_1(1) = 2$ and the set of trajectories passing through association $\tau_1 \to \tau_2$ is Ψ_1^2. While Ψ_1^2 has three elements

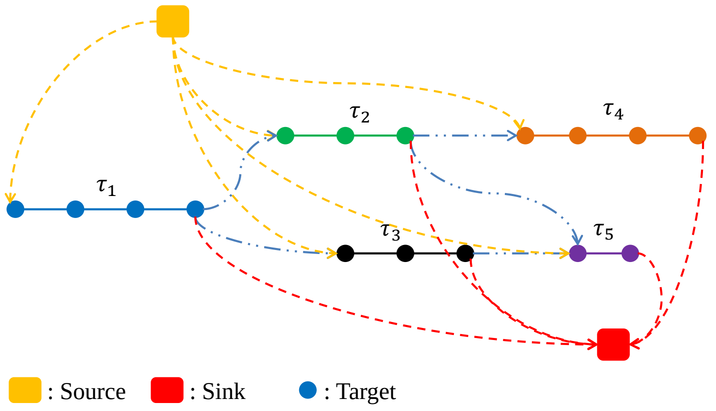


Fig. 4. An example of high-level association. The orange (red) square denotes the source (sink) tracklet; the colored disk denotes the target; the solid line denotes the real tracklet and the virtual arrow denotes the association hypothesis between two tracklets. Best viewed in color.

$\{0, 1, 2, \infty\}$, $\{0, 1, 2, 4, \infty\}$ and $\{0, 1, 2, 5, \infty\}$. Suppose the three trajectories have affinities as a_1, a_2 and a_3 respectively, then the affinity score S_1 is computed as

$$\begin{aligned} S_1 = & a_1 x_{01} x_{12} x_{2\infty} + a_2 x_{01} x_{12} x_{24} x_{4\infty} \\ & + a_3 x_{01} x_{12} x_{25} x_{5\infty}. \end{aligned} \tag{26}$$

The scores S_2 for association $\tau_1 \to \tau_3$ and S_3 for association $\tau_1 \to \tau_\infty$ can be computed in the same way.

6.2 Hypothesis Generation

The gate strategy used in traditional approaches (e.g., [11], [45]) for association hypothesis generation is followed. In the low-level process, two object detections from two neighbouring frames are associated as a hypothesis only if they are spatially close to each other. In the high-level process tracklets are associated if they are temporally close to each other. This strategy is popular and effective for general tracking applications where the camera is fixed and targets possess limited spatial offsets (i.e., velocities). In our implementation, a distance threshold is set to guarantee the inclusion of all true associations, and a time threshold is set to exclude many spurious associations. Generally, the thresholds are application dependent.

An important issue in hypothesis generation is the management of special events, i.e., target entrance (reappearance) and exit (disappearance). For handling this issue, in each frame we include two dummy targets, a source and a sink, to generate the appearance and disappearance hypotheses. Each target in the previous frame associates with the sink target in the current frame to form the disappearance hypothesis, and each target in the current frame associates with the source target in the previous frame to construct the appearance hypothesis. In this case, the one-to-one mapping constraint (2) does not hold for the dummy targets.

Algorithm 3 High-level tracklet association

```
Input: Tracklet set $\mathbb{T} : \{\tau_1, \tau_2, ..., \tau_N\}$.
Output: Trajectory set $\Pi$.
Set the forward association set $\Phi_i = \{\infty\}$ for $\tau_i$;
Set the backward association set $\Theta_i = \{0\}$ for $\tau_i$.
// Find all association candidates
for $i = 1, \ldots, N$ do
    for $j = 1, \ldots, N$ do
        if $(0 < t_j^{(s_j)} - t_i^{(e_i)} < T_0)$ then
            $\Phi_i = \Phi_i \cup \{j\}; \Theta_j = \Theta_j \cup \{i\}$.
        end if
    end for
end for
// Find trajectory candidates and compute the affinities.
Initialize trajectory hypotheses set $\Gamma = \{\{0, 1\}, ..., \{0, N\}\}$
repeat
    for $k = 1 : \mathrm{card}(\Gamma)$ do
        Suppose the last element in subset $\Gamma(k)$ as $l$.
        if $l \neq \infty$ then
            Grow the trajectory hypothesis $\Gamma(k)$ into branches with
            $\Phi_l$ like the tree growing[2], and update $\Gamma$
        end if
    end for
until $\Gamma$ can not be extended
Compute the affinities for all trajectory hypotheses in $\Gamma$.
// Power iteration solution
Initialize the association value $x_{ij}$ for any two tracklets $\tau_i$ and
$\tau_j$
for $iter = 1 : \mathrm{max}_{iter}$ do
    for $i = 1 : N$ do
        $S_j = 0, j = 1, 2, ..., N$        // initialization
        for $k = 1 : \mathrm{card}(\Phi_i)$ do
            Suppose $j = \Phi_i(k)$, then the affinity score $S_j$ is defined
            as $S_j = \sum_{\Gamma(l) \in \Psi_i^j} a_l \times \prod_{x_{ef} \in A_l} x_{ef}$
        end for
        $x_{ij} = \frac{S_j}{\sum_j S_j}, j = 1, 2, ..., N$
    end for
    $x_{ij} = \frac{x_{ij}}{\sum_i x_{ij}}, i = 1, 2, ..., N$
    Alternate row/column normalization on $x_{ij}$ to make it a
    double stochastic matrix.
end for
Discretize $x_{ij}$ to make it binary matrix.
If $x_{ij} = 1$, tracklets $\tau_i$ and $\tau_j$ are linked together. In this way,
we get the final long trajectory.
```

6.3 Affinity Computation

The proposed high-order association framework can be applied to various kinds of affinity models, as is shown in the experiments. Different affinity models are designed in the two-stage associations and depending on the applications.

6.3.1 Low-level Association Affinity

In the low-level association, two kinds of models are applied: the snake-based model and the constant-velocity model, to compute the affinity of the tracklet hypotheses.

For a tracklet hypothesis $\tau : \{\mathbf{o}_{i_0}^{(0)}, \mathbf{o}_{i_1}^{(1)} \ldots, \mathbf{o}_{i_K}^{(K)}\}$, its affinity is $c_{i_0:i_K}$, and the velocity of τ at frame k is represented as $\mathbf{z}_{i_{k-1}i_k}^{(k)} \in \mathbb{R}^2$ (i.e., $\mathbf{z}_{i_{k-1}i_k}^{(k)} = \mathbf{p}_{i_k}^{(k)} - \mathbf{p}_{i_{k-1}}^{(k-1)}$).

(1) The **snake-based affinity**, which has been applied for point set tracking in [16], encourages small and smooth

[2] The trajectory hypothesis generation has a tree structure, The sequence of tracklets in a trajectory hypothesis form a path from the root node to the leaf node of a trajectory tree.

motion up to the second order as defined below:

$$c_{i_0:i_K} = E_0 - \sum_{k=1}^{K} \left|\mathbf{z}^{(k)}_{i_{k-1}i_k}\right| - \alpha \sum_{k=1}^{K-1} \left|\mathbf{z}^{(k+1)}_{i_k i_{k+1}} - \mathbf{z}^{(k)}_{i_{k-1}i_k}\right|, \tag{27}$$

where α is the weighting parameter and E_0 is a constant to keep the affinities positive. The first component measures the total spatial offsets of the consecutive target detections and penalizes any large jump in position between two targets. The second component measures the velocity variation of the targets at two successive frames.

(2) The **constant velocity affinity**, which measures the motion similarity between consecutive velocity vectors, is computed according to (28),

$$\begin{aligned} c_{i_0:i_K} = \prod_{k=1}^{K-1} \exp\Bigg(& \frac{\langle \mathbf{z}^{(k)}_{i_{k-1}i_k}, \mathbf{z}^{(k+1)}_{i_k i_{k+1}} \rangle}{\left|\mathbf{z}^{(k)}_{i_{k-1}i_k}\right| \left|\mathbf{z}^{(k+1)}_{i_k i_{k+1}}\right|} + \\ & \frac{2\left|\mathbf{z}^{(k)}_{i_{k-1}i_k}\right| \left|\mathbf{z}^{(k+1)}_{i_k i_{k+1}}\right|}{\left|\mathbf{z}^{(k)}_{i_{k-1}i_k}\right|^2 + \left|\mathbf{z}^{(k+1)}_{i_k i_{k+1}}\right|^2} \Bigg), \end{aligned} \tag{28}$$

where the exponential operation ensures that the affinity value is positive. The two components represent respectively the orientation consistency and the magnitude consistency between two neighbor velocities. A similar model is employed in [49] for point set tracking.

6.3.2 *High-level Association Affinity*

For a trajectory hypothesis $\Gamma(l) : \{\tau_1, \tau_2, ..., \tau_M\}$ where $\tau_i, \tau_{i+1} (1 \leq i < M)$ are the sequential tracklets, the local association affinity s_i between consecutive tracklets τ_i and τ_{i+1} is defined by

$$s_i = (sa_i + sd_i)\, st_i, \tag{29}$$

where sa_i, sd_i and st_i are the appearance affinity, spatial (distance) affinity and temporal affinity respectively. The three items are defined as

$$\begin{aligned} sa_i &= \sum\nolimits_b \min\left(h^i_b, h^{i+1}_b\right) \\ sd_i &= \frac{1}{2} \exp\left(\frac{\left|\mathbf{p}^{(s_{i+1})}_{i+1} - \mathbf{p}^{(e_i)}_i - \Delta t \mathbf{z}^{(e_i)}_{ii}\right|^2}{-2\left|\mathbf{z}^{(e_i)}_{ii}\right|^2} \right) \\ &+ \frac{1}{2} \exp\left(\frac{\left|\mathbf{p}^{(s_{i+1})}_{i+1} - \mathbf{p}^{(e_i)}_i - \Delta t \mathbf{z}^{(s_{i+1})}_{i+1i+1}\right|^2}{-2\left|\mathbf{z}^{(s_{i+1})}_{i+1i+1}\right|^2} \right), \\ st_i &= \begin{cases} \exp(-\frac{\Delta t}{T_0}), & \text{if } \ 0 < \Delta t < T_0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{30}$$

In (30), h^i_b (h^{i+1}_b) is the value in the b-th bin of the average color histogram of the tracklet τ_i (τ_{i+1}), $\Delta t = t^{(s_{i+1})}_{i+1} - t^{(e_i)}_i$ is the time gap between τ_i and τ_{i+1}, and T_0 is the temporal threshold for possible tracklet associations.

Given the pairwise tracklet-tracklet association affinity (29), the affinity of the trajectory $\Gamma(l)$ is defined by $a_l = \sum_{i=1}^{M-1} s_i$. Note that other kinds of high-order affinities can be explored too. An affinity design is good if the better the association, the higher the affinity value. The main focus of this paper is to introduce the R1TA based MIT framework. Further exploration of possible affinities is left to future work.

6.4 Initialization and Termination

Since the MDA problem is in general NP-hard, the tensor power iteration algorithms can not guarantee a global optimum. Different initial points may lead to different local solutions. The best initial solution may be application dependent. This paper focusses on general MTT problems without the prior information. A uniform initialization is employed. Specifically, for a target with M association candidates, the probabilities for associating with them are initially all set at the same value, $1/M$.

Other initialization schemes such as the weighted initialization, where the more plausible candidates for association are given higher initial value, could be used. A robust method for the local optimization is to start the iteration from multiple initial points, then select the one with the best convergence value. In the experiments, only the uniform initialization scheme is applied. This simple strategy produces promising results.

The optimization iteration is terminated when the predefined maximum number of iterations (set to 100 for all the experiments) is reached. Finally, the solution given by tensor power iteration is real valued. It must be binarized to meet the binary assignment constraints. To resolve the conflicts between different local association candidates, the real valued solutions is treated as the costs of corresponding association candidates and formulate them into a bipartite graph matching problem. Then, we apply the Hungarian algorithm to obtain the binary output.

7 Experiments

To validate the effectiveness of the tensor approximation based tracking framework, a series of experiments are performed on diverse and challenging datasets. In this section, we first evaluate the performance on data association using point set and small target datasets. Then, the two-level tracking framework is validated with the pedestrian sequences.

7.1 Sequences

In the experiments, the proposed algorithm is validated on various challenging video sequences. In particular, the Columbus Large Image Format (CLIF) dataset [1], the PSU dataset [26] and the Similar Multi-object Tracking (SMTT) dataset [22] are used to evaluate the effectiveness of multi-target association. The CLIF sequences are wide area aerial images, which portray high-speed and crowded traffic scenarios. Three CLIF sequences, namely CLIF-1, CLIF-2 and CLIF-3, each with 100 frames, are used in the experiments. The vehicle targets are acquired via background subtraction and object detection. Details can be found in [51]. The PSU dataset includes six sequences in two subsets: PSU-dense and PSU-sparse, which are the trajectories from pedestrians walking in an atrium. Both PSU-sparse and PSU-dense contain three different frame-rate sequences. The last number in each sequence name denotes the frame rate (e.g., '3' in 'PSU-dense 3'). The SMTT dataset consists of eight videos in which the targets have similar appearances. In

TABLE 3
Statistics of datasets used in the experiments

Dataset. name	#seq. tested	#targets per seq.	fps	Motion pattern	Detection quality
CLIF	3	100 $\sim$ 300	2	very fast	noisy
PSU	6	3 $\sim$ 30	1 $\sim$ 3	fast	clean
SMTT	3	5 $\sim$ 40	high	moderate	noisy
Pedestrian	13	5 $\sim$ 40	high	moderate	noisy
KITTI-Car	29	5 $\sim$ 100	high	fast	noisy

the experiments three sequences with many objects, namely seagulls, balls and crowd, were chosen as test data.

Multi-target tracking is explored in pedestrian and car tracking datasets. The pedestrian datasets include the PETS09 S2L1 and TUD-Stadtmitte sequences, as well as the widely used 2D MOT 2015 benchmark [32]. The KITTI-Car benchmark [27] contains 29 test sequences with complicated traffic scenarios.

All these datasets are challenging due to factors such as crowded scenarios, weak or unavailable appearance information, low frame-rate and/or fast motion and noisy object detections. An overview of these sequences is given in Table 3.

7.2 Trackers and Evaluation Metrics

In the evaluation, we compare the proposed approach with several state-of-the-art MTT trackers including the ICM tracker [16], the Network-Flow tracker [41] and the IHTLS tracker [22].

The same affinity representation is used in both our approach and the ICM tracker. The snake-based affinity is used for the PSU dataset, the SMTT dataset and the pedestrian dataset, and the constant velocity affinity is used in the CLIF tracking. The Network-Flow tracker and the IHTLS tracker make use of their default affinities. The parameters E_0 and α in the snake-based affinity are set to 1000 and 0.5 respectively. In the low-level association, a batch size of 6 frames is chosen in both our approach and the ICM tracker for the CLIF and the PSU sequences. A batch size of 10 is chosen for the SMTT and pedestrian sequences.

Three sets of evaluation metrics are used in the experiments. First, the *correct match percentage* P_c and the *wrong match percentage* P_w are used to evaluate the low-level association performance[2]. The P_c and P_w are defined as

$$P_c = 100 \times \frac{\sum_k cm(k)}{\sum_k gt(k)}, \quad P_w = 100 \times \frac{\sum_k wm(k)}{\sum_k gt(k)}, \tag{31}$$

where $cm(k)$ and $wm(k)$ represent respectively the numbers of correct and wrong (i.e., ID switch) associations in the k-th frame, and $gt(k)$ denotes the ground truth association number in the same frame.

The two other sets of metrics are applied to evaluate pedestrian tracking performance. The first set is the CLEAR MOT metric [9]. The second set of metrics [34] evaluates the numbers of mostly/partially tracked (MT/PT), mostly lost (ML) trajectories, numbers of fragments (Frag) and ID switches

7.3 Multi-target Association

The quantitative association results of the four approaches on the CLIF, PSU and SMTT datasets are presented in Table 4. It can be seen that our approach performs the best among the four trackers on almost all sequences, followed by the ICM tracker. Overall, all trackers perform weakly on the complex sequences, such as the crowded CLIF, PSU-dense and the low frame-rate PSU-dense 1 and PSU-sparse 1. This is not a surprise since there exist large association ambiguities in crowded and/or fast motion scenarios.

Aside from achieving the best performances in most sequences, our algorithm has a notably high performance on the CLIF dataset. This is attributed mainly to its capability to effectively encode high order motion information, which is very important for wide area aerial-borne videos where vehicles are small and hardly distinguishable from each other. It is observed that our method performs worse on CLIF-3 than on the other two sequences. This is attributed to poor vehicle detection, such as frequent missing detections and false positives. The raw detection and frame-between association results can be seen in the supplementary files.

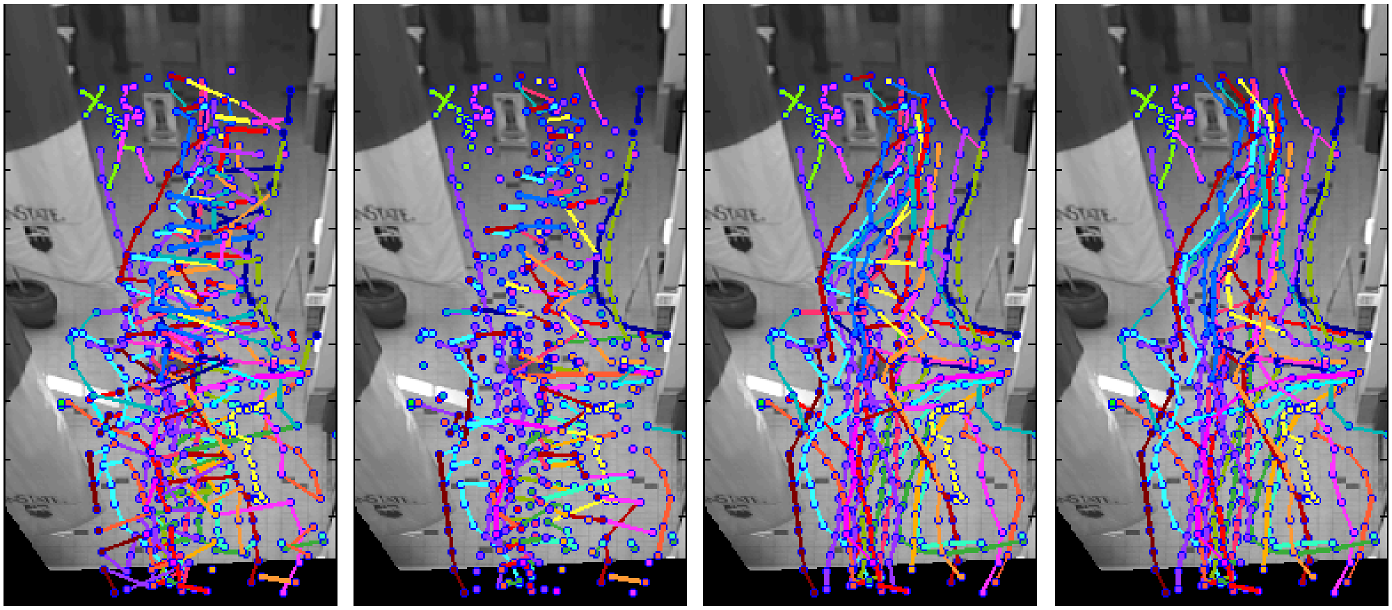
The ICM tracker uses a block update strategy like ours and has a performance level closest to ours. The two approaches take the similar block update strategy. The ICM tracker iterates its association solution with binary values using the Hungarian algorithm, while our approach searches for a solution in a relaxed real valued domain with tensor power iteration. The real valued tensor power iteration avoids hard decisions, which may introduce association errors in the very early stages, as in the ICM algorithm. Furthermore, it is observed that the performance gap between our approach and the ICM tracker is larger on the difficult sequences, such as PSU-dense 1 and PSU-sparse 1, than on the easy ones. This observation demonstrates that the proposed approach can alleviate the association ambiguity efficiently.

IHTLS follows the hierarchical association strategy. A double threshold local association is applied in the low-level processing stage. The double threshold association yields a favourable performance in the sparse or slow motion scenarios, where the association ambiguities are small and the associations are reliable. In contrast, in fast motion and crowded scenes, the double threshold association is insufficient. As for the Network-Flow algorithm, the pairwise association cost limits its success in complex applications such as crowded and high-speed scenarios.

Some qualitative association results of all four algorithms on the PSU sequence are presented in Fig. 5. Our approach performs the best with the fewest mismatches. The Network-Flow algorithm generates some disordered local associations, due to the lack of high order motion information. There are numerous isolated targets in the trajectories produced by IHTLS. These may be attributed to the somewhat inflexible association decision, such as the double threshold strategy used in IHTLS.

The association performances of all algorithms on a CLIF sequence are shown in Fig. 6, where the between-frame association results are presented. It can be seen that the

[2] The two metrics are not fully dependent on each other and the sum of P_c and P_w is not exactly 1, as shown in the experiments. This is attributed to the noisy false positives. In addition, the missing associations are not counted here.



Network-Flow (221) IHTLS (168) ICM (70) Proposed (29)

Fig. 5. Association results on one episode (20 frames) in PSU-dense-1. Numbers of mismatches are listed on the side of the corresponding algorithms. All trajectories are color-coded with respect to the ground truth. Edges of good trajectories appear in the same color.

TABLE 4
Quantitative evaluation of MTT algorithms (%).

		Correct match percentage				Wrong match percentage			
		Flow	IHTLS	ICM	Ours	Flow	IHTLS	ICM	Ours
CLIF	CLIF-1	54.60	11.50	71.10	**85.54**	45.24	39.37	28.77	**14.87**
	CLIF-2	60.83	31.10	57.93	**81.29**	37.53	25.78	43.69	**22.76**
	CLIF-3	55.16	13.83	65.68	**74.71**	44.33	**14.60**	35.49	25.15
PSU	PSU-sparse 1	94.57	88.22	98.87	**99.45**	5.43	6.54	0.97	**0.50**
	PSU-sparse 2	99.72	97.55	99.97	**99.99**	0.28	1.13	0.01	**0.00**
	PSU-sparse 3	99.96	98.57	99.95	**99.99**	0.04	0.70	**0.00**	**0.00**
	PSU-dense 1	78.65	71.23	93.63	**96.98**	21.35	20.78	6.26	**3.01**
	PSU-dense 2	98.64	94.66	99.74	**99.78**	1.36	3.99	0.24	**0.20**
	PSU-dense 3	99.77	96.78	99.91	**99.94**	0.23	1.90	0.08	**0.05**
SMTT	balls	98.75	98.66	99.91	**100.0**	1.21	0.33	0.07	**0.00**
	seagulls	95.79	99.76	99.94	**99.96**	4.14	0.15	**0.00**	**0.00**
	crowd	99.11	99.83	99.98	**99.99**	0.89	0.16	**0.01**	**0.01**

scenario is very challenging, because of the large number of targets, the noisy vehicle detections and the crowded and similar target distractions. The proposed approach obtains the best performance with the fewest mismatches and the most correct matches. By contrast, other trackers have many missing associations.

TABLE 5
Tracking results on PETS09 S2L1

Tracker	Rec	Prec	TA	TP	MT	PT	Frag	IDS
[59]	91.8	**99.0**	-	-	89.5	10.5	9	**0**
[41]	94.0	97.4	88.9	80.9	89.5	10.5	13	10
[22]	96.3	92.7	85.4	**86.5**	94.7	5.3	49	16
[16]	97.2	97.4	93.5	83.2	94.7	5.3	21	12
Ours	**98.2**	98.3	**96.0**	82.0	**100.0**	**0.00**	**6**	2

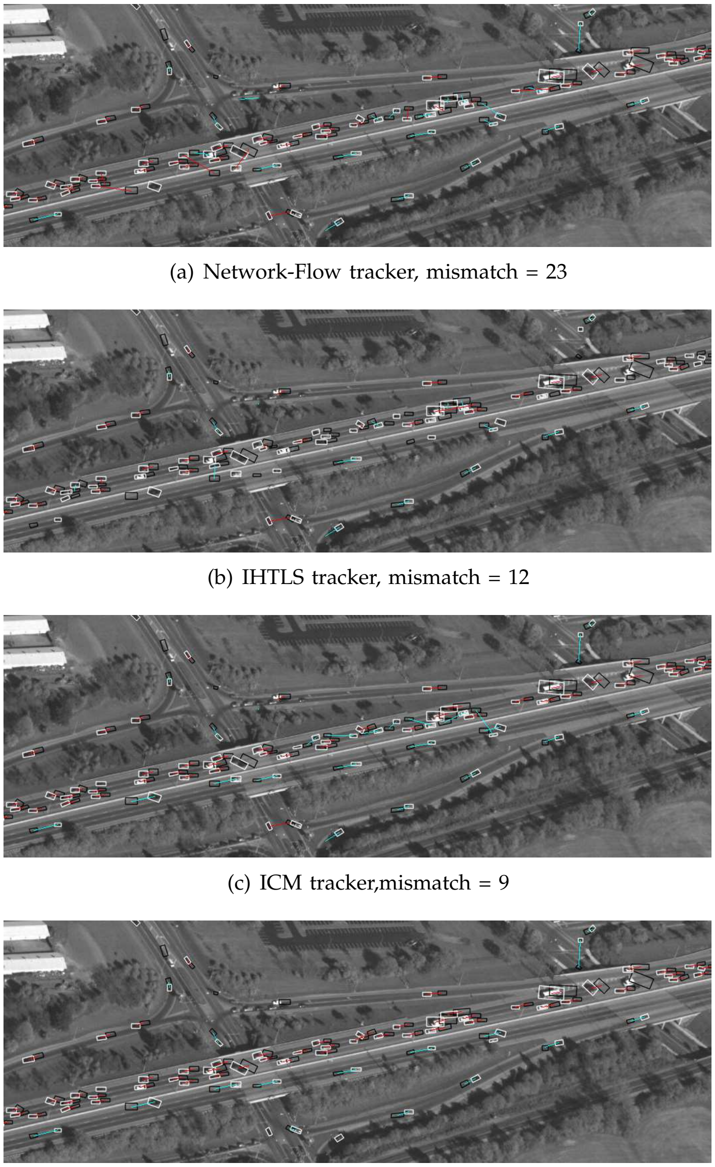
7.4 Multi-target Tracking

Multiple object tracking is performed with hierarchical data associations, followed by the post-process such as isolated target filtering and trajectory smoothing[3]. The proposed tracking approach is validated on two different tasks, namely pedestrian tracking and vehicle tracking.

First, pedestrian tracking is performed on two sequences, PETS09 S2L1 and TUD-Stadtmitte. The pedestrian detection results in [59], [60] are used as the association inputs. For comparison, their tracking results are listed in the experiments. T_0 in (30) is set as 30 for both sequences. Tracklets with large time gaps are not linked because such associations are likely to be erroneous. In the comparison, the ICM tracker takes the same settings used in our approach. The Network-Flow tracker and the IHTLS tracker take their default settings used for pedestrian tracking. Quantitative results are presented in Table 5 and Table 6. It can be seen that the proposed tracking approach achieves the best results overall.

The results of the proposed tracker on two pedestrian

[3] Finally, short trajectories with less than 5 instances are removed. The remaining trajectories are smoothed with spline fitting.



(a) Network-Flow tracker, mismatch = 23

(b) IHTLS tracker, mismatch = 12

(c) ICM tracker,mismatch = 9

(d) Proposed tracker, mismatch = 3

Fig. 6. Association results between two neighbor CLIF frames (only a small section is shown for better visualization). White (black) rectangles denote the vehicle detections in the current (last) frame, red and blue lines represent the associations on two opposed directions.

TABLE 6
Tracking results on TUD-Stadtmitte

Tracker	Rec	Prec	TA	TP	MT	PT	Frag	IDS
[60]	**87.0**	96.7	-	-	70.0	30.0	**1**	**0**
[41]	83.8	96.5	75.9	82.6	**80.0**	**20.0**	10	8
[22]	83.6	93.2	69.8	83.7	60	40	15	13
[16]	84.6	97.9	78.7	86.6	80	20	7	3
Ours	86.2	**99.9**	**84.8**	**89.6**	70.0	30.0	4	1

sequences are shown in Fig. 7. Object tracking for the two sequences is difficult because of frequent target interactions such as crossings and collisions, as well as the crowded scenarios. It can be seen that our approach yields excellent results for both pedestrian tracking sequences. More results can be seen in the supplemental videos.

The proposed tracker is evaluated on the 2D MOT 2015 benchmark [32]. We compare the proposed algorithm with the Network Flow approach [41], the ITHLS tracker [22], the CEM tracker [37], the DCO tracker [38], the SiamaeseCNN tracker [31] and the RNN based tracker [36]. The results are presented in Table 7. Generally, the SiameseCNN tracker has the best performance. Note that it applies the deep neural network for learning the discriminative affinity model. Our approach takes the second place with the plain affinity model based on motion, temporal and appearance information.

The pedestrian datasets are very different from the PSU and CLIF sequences. With the noisy pedestrian detection output and slow target motion, the high order motion representation is affected by errors. In this case the hight order affinity model is not much powerful as that used in the PSU and CLIF datasets. Further, the pedestrian targets have larger spatial occupancy than points and vehicles in CLIF, and the appearance features of the pedestrian detections are much more discriminative. The methods with elaborate pairwise affinity models computed on discriminative appearances achieve good performance.

The proposed algorithm is tested on car tracking using the KITTI-Car benchmark [27], which consists of 21 training sequences and 29 test sequences. The proposed algorithm is compared with the Network Flow approach [41], the CEM tracker, the DCO tracker, the modeSSP* tracker [33] and the LP-SSVM tracker [57]. In the implementation, the regionlet detection is used as the association input. The results are presented in Table 8. It can be seen that the LP-SSVM tracker has the best performance, while our algorithm has a result comparable with that of the modeSSP* tracker in this evaluation. The LP-SSVM tracker extends the min-cost flow tracking framework with two improvements, which are the introduction of the contextual interaction and the tracking parameter learning. Generally, the LP-SSVM tracking shows the importance of the tracking parameter learning which depends on similar scenario training.

For the two benchmarks, our approach has a much better result on the KITTI-Car than on the 2D MOT 2015. The underlying reason is that the car target has much faster and more constrained motion pattern, and thus the designed high order affinity is much more powerful and effective for the association task.

7.5 Algorithm analysis

Convergence analysis: To study the convergence property of the proposed iterative solution. It is shown how the optimization energy evolves over iterations in Fig. 8. It can be seen that the energy has been improved consistently along the iterations.

We also present how the association performance varies along with the optimization energy objective; the result on the PSU-dense 1 is presented in Fig. 9. It can be seen that the total association energy (i.e., f) steadily climbs to a stable value, while the algorithm obtains better association results with more iterations. Furthermore, it illustrates that the snake-based affinity used in the PSU datasets is well designed.

Affinity design: The affinity model plays a fundamental role in the association, and our approach has the flexibility to explore different kinds of high order affinities. In this

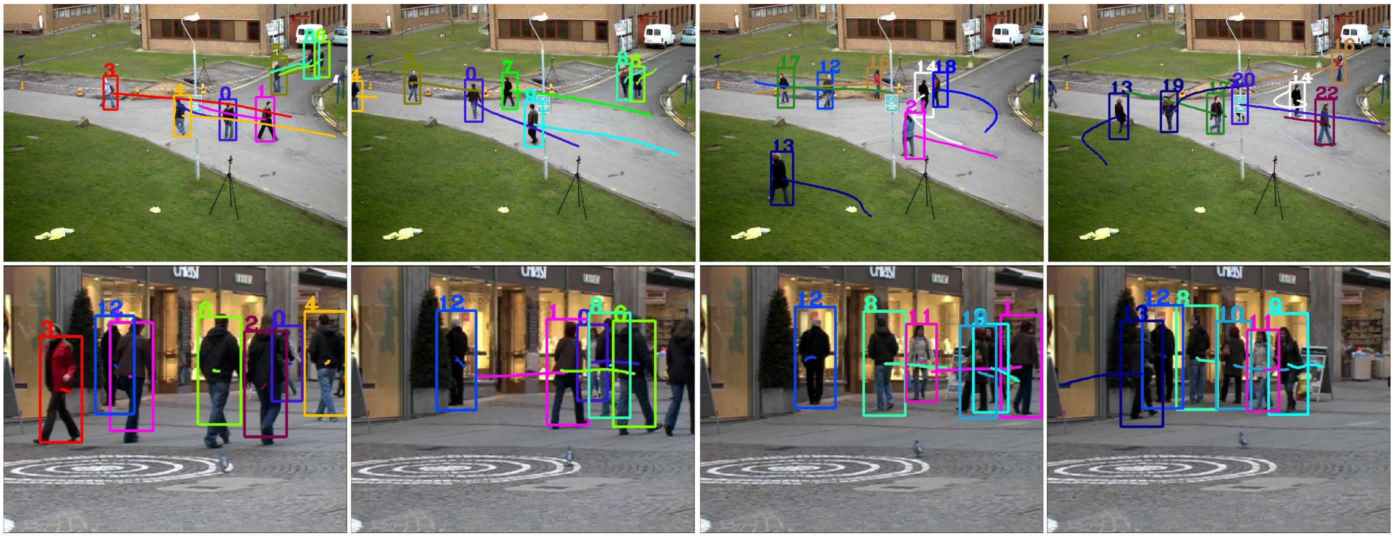


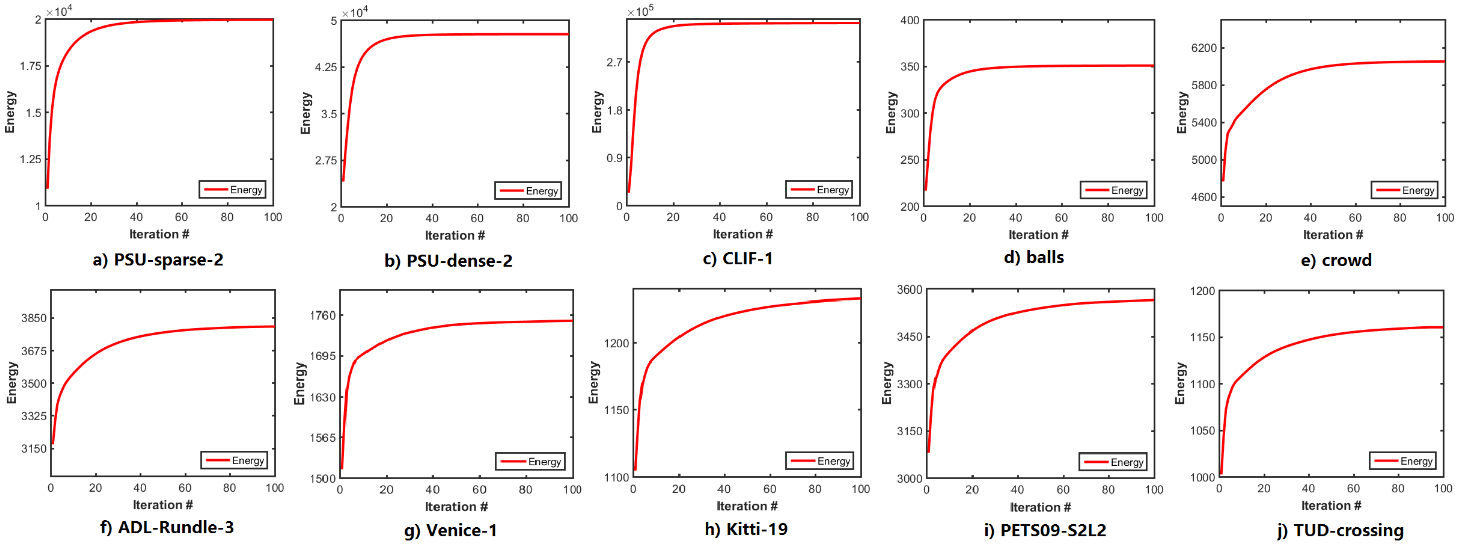
Fig. 7. Tracking results of our approach on two pedestrian sequences. Top: PETS09 S2L1, Bottom: TUD-Stadtmitte.

TABLE 7
Tracking results on 2D MOT 2015

Tracker	MOTA	IDF1	MT	ML	FP	FN	IDS	Frag	Hz
DP_NMS [41]	14.5	19.7	6.0%	**40.8%**	13171	34814	4537	3090	**444.8**
ITHLS [22]	18.2	**0.0**	2.8%	54.8%	8780	40130	1148	2132	2.7
CEM [37]	19.3	**0.0**	**8.5%**	46.5%	14180	**34591**	813	1023	1.1
DCO [38]	19.6	31.5	5.1%	54.9%	10652	38232	**521**	**819**	0.3
SiameseCNN [31]	**29.0**	34.3	**8.5%**	48.4%	**5160**	37798	639	1316	52.8
RNN_LSTM [36]	19.0	17.1	5.5%	45.6%	11578	36706	1490	2081	165.2
Ours	24.3	24.1	5.5%	46.6%	6664	38582	1271	1304	24.0

TABLE 8
Tracking results on KITTI-Car

Tracker	MOTA	MOTP	MT	ML	FP	FN	IDS	Frag	Runtime	Environment
DP_MCF [41]	38.33	78.41	18.00%	36.15%	**70**	18425	2716	3225	0.01s	1 core @ 2.5 Ghz (python)
CEM [37]	51.94	77.11	20.00%	31.54%	807	15598	125	**396**	0.09s	1 core @ > 3.5 Ghz (Matlab + C/C++)
DCO [38]	37.28	74.36	15.54%	30.92%	4458	16891	220	612	0.03s	1 core @ > 3.5 Ghz (Matlab + C/C++)
modeSSP* [31]	72.69	78.75	48.77%	8.77%	1918	7360	114	858	0.01s	1 core @ 2.7 Ghz (Python)
LP-SSVM* [57]	**77.63**	77.80	**56.31%**	**8.46%**	1239	**6393**	**62**	539	0.02s	1 core @ 2.5 Ghz (Matlab + C/C++)
Ours	71.18	**79.15**	47.85%	11.69%	1915	7579	418	947	0.04s	1 core @ 2.5 Ghz (Matlab + C/C++)



a) PSU-sparse-2 b) PSU-dense-2 c) CLIF-1 d) balls e) crowd

f) ADL-Rundle-3 g) Venice-1 h) Kitti-19 i) PETS09-S2L2 j) TUD-crossing

Fig. 8. The energy variation curves of the proposed iterative solution on different sequences.

work, we adopt different affinities to accommodate different scenarios. Two kinds of affinities are applied in the low-level associations.

The performance of two proposed affinities is tested on three datasets to validate the impacts of the affinity model. The results are presented in Table 9. It can be seen that the two affinities have different performances across the sequences. The constant velocity affinity performs well on the

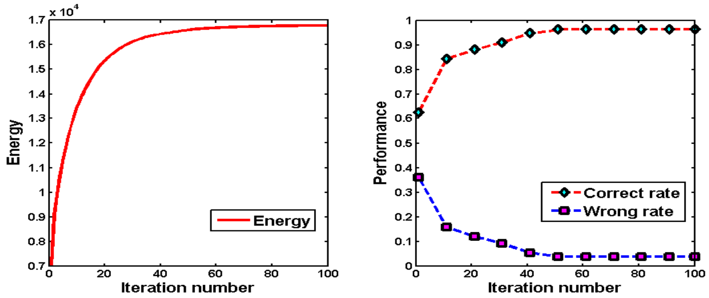


Fig. 9. The total affinity (left) and performances (right) versus the number of iterations for one batch in the PSU dataset.

CLIF sequences but not on other sequences. The underlying reason for the affinity design is that, in the high-way traffic application, cars follow very consistent motion patterns, so we make use of the highly constrained constant-velocity model. While in the moderate speed and unrestricted scenarios, the flexible snake-based model is appropriate since it balances the spatial proximity and motion smoothness.

Detection dependence: Target detection is the base of the tracking-by-detection algorithm. The false positive, missing and noisy detections may bring a lot of association ambiguities. For one thing, the better the detection output, the better the association performance of our approach. For another, the association results give the feedback for the detection refinement. In our approach, several strategies are employed to advance the detection and association performance. First, short tracks are filtered out to reduce the false positives. Second, the short-term target missing can be re-detected by temporal prediction in the final tracklet association. Finally, the spline fitting is applied to smooth the trajectories.

Our association can greatly refine the raw detection, as can be observed from Table 5, Table 6 and Table 8. In these experiments, all the algorithms have the same raw detections as the association input. On the detection precision and tracking precision metrics which evaluate the detection accuracy of the trajectories, our approach achieves very promising results. Since the proposed hierarchical association amends the detection, reducing false alarms by the iterative optimization and smoothing the detections with fitting.

Running time: The computation time depends on the number of high-order trajectory hypotheses. For instance, for a batch of $K+1$ frames, each frame has N targets and every target has M association candidates in the next frame. In this case, there are a total of NM^K trajectory hypotheses. If the solution is iterated L times, the computation complexity of one batch association is $O(LK^2NM^K)$. Generally, the running time fluctuates in applications. In the experiments, the iteration number of the solution is set to 100 for all the sequences, and the running time of the proposed approach in the low-level association are presented in Table 10.

Generally, our approach runs faster than the ICM tracker, especially for the crowded sequences such as PSU-dense 1, AVG-Center and the crowd. The main computations in the ICM tracker are the iterative pairwise assignment optimization and the repeated global trajectory search.

8 Conclusion

This work focusses on the high-order data association optimization without pairwise relaxations. The main contributions are concluded from several aspects. First, we bridge the tensor algebra and MDA optimization by using the close relationship between the rank-1 tensor approximation (R1TA) and the MDA optimization. Second, an l1 norm constraint tensor power iteration solution is proposed. The convergence of the R1TA problem is studied. We further put forward the row/column constrained power iteration solution for the MDA task. Finally, the proposed multi-frame data association solution is extended to the high-level association where tracklets are not temporally aligned.

The effectiveness of the proposed algorithm is validated using various datasets, ranging from the point set association, similar target association to crowded pedestrian and car tracking. Generally, our solution yields remarkable performances, especially for the fast and textureless target motion where large association ambiguities exists. It mainly attributes to two important benefits in the proposed algorithm: the high order association affinity and the soft iteration solution. Even with simple hand-crafted features, the proposed approach has favourable performance on the pedestrian and car tracking. In future work, we will investigate different high-order association affinities in the R1TA framework, for example, the feature learning based on the deep neural network.

Acknowledgments: We would like to express our sincere appreciation to Professor Steve Maybank for his valuable suggestion and careful revision on the wordings and grammar in the paper. This work is supported by Beijing Natural Science Foundation (Grant No. L172051), the Natural Science Foundation of China (Grant No. 61502492, 61751212, 61721004), the NSFC-general technology collaborative Fund for basic research (Grant No. U1636218), the Key Research Program of Frontier Sciences, CAS, Grant No. QYZDJ-SSW-JSC040, and the CAS External cooperation key project. H. Ling was supported in part by US NSF (grants 1814745, 1407156, and 1350521).

References

[1] The CLIF dataset. www.sdms.afrl.af.mil/index.php?collection=clif 2006. 2, 10

[2] K.C.A. Kumar and C. Vleeschouwer. Discriminative label propagation for multi-object tracking with sporadic appearance features. Proc. IEEE Conf. Computer Vision, 2013. 3

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008. 1

[4] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2014. 2

[5] Y. Ban, S. Ba, X. Alameda-Pineda and R. Horaud. Tracking multiple persons based on a variational bayesian model. Proc. European Conference on Computer Vision, 2016. 3

[6] Y. Bar-Shalom and T. Fortmann. Tracking and data association. Academic Press, 1988. 1, 2

[7] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2011. 3

[8] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 33, no. 9, pp. 1806-1819, Sep. 2011. 1, 2, 5

[9] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. on Image & Video Proc.*, 2008. 11

TABLE 9
Quantitative evaluation of MTT algorithms (%).

		Correct match percentage		Wrong match percentage	
		Snake-based	Constant-velocity	Snake-based	Constant-velocity
CLIF	CLIF-1	75.11	**85.54**	24.39	**14.87**
	CLIF-2	66.77	**81.29**	36.54	**22.76**
	CLIF-3	71.20	**74.71**	29.98	**25.15**
PSU	PSU-sparse 1	**99.45**	90.41	**0.50**	9.43
	PSU-sparse 2	**99.99**	95.36	**0.00**	4.60
	PSU-sparse 3	**99.99**	95.99	**0.00**	3.99
	PSU-dense 1	**96.98**	80.74	**3.01**	18.60
	PSU-dense 2	**99.78**	95.84	**0.20**	4.02
	PSU-dense 3	**99.94**	95.84	**0.05**	2.62
SMTT	balls	**100.00**	97.94	**0.00**	2.03
	seagulls	**99.96**	99.91	**0.00**	0.04
	crowd	**99.99**	99.95	**0.01**	0.04

TABLE 10
Running time comparison between our approach and ICM tracker. (sec)

Seq	CLIF-1	CLIF-2	CLIF-3	Venice-1	KITTI-19	Jelmoli	TUD-Crossing	ETH-Crossing	Linthescher	AVG-Center
Ours	445	116	86	38	137	14	2	3	7	498
ICM	1113	100	93	248	82	37	1	2	9	4270

Seq	balls	seagulls	crowd	Rundle-1	Rundle-3	KITTI-16	PETS09-S2L2	PSU-dense 1	PSU-dense 2	PSU-sparse 1
Our	4	198	1566	39	135	22	89	936	320	13
ICM	1	410	3465	228	433	32	106	23452	842	15

[10] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. Proc. Workshop Motion and Video Computing, Dec. 2002. 2

[11] S. Blackman and R. Popoli. Design and analysis of modern tracking systems. Artech House, Norwood, MA, 1999. 1, 2, 9

[12] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multi-person tracking-by-detection from a single, uncalibrated camera. IEEE Trans. Pattern Analysis and Machine Interlligence, vol. 33, no. 9, pp. 1820-1833, Sep. 2011. 2

[13] A. Butt and R. T. Collins. Multiple target tracking using frame triplets. Proc. Asian Conf. Computer Vision, 2012. 3

[14] A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2013. 2, 3

[15] V. Chari, S.L. Julien, I. Laptev, J. Sivic. On pairwise costs for network flow multi-object tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition 2015. 2

[16] R. T. Collins. Multitarget data association with higher-order motion models. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012. 1, 3, 4, 9, 11, 12, 13

[17] I. Cox. A review of statistical data association techniques for motion correspondence. Int'l J. Computer Vision, vol. 10, no. 1, pp. 53-66, 1993. 2

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005. 1

[19] S. Deb, M. Yeddanapudi, K. Pittipati, and Y. Bar-Shalom. A generalized S-D assignment algorithm for multi-sensor multi-target state estimation. IEEE Trans. Aerospace and Electronic Systems, vol. 33, no. 2, pp. 523-538, Apr. 1997. 1, 2

[20] A. Dehghan, S. Modiri, and M. Shah. GMMCP tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2015. 3

[21] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(R_1, R_2,..., R_N) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl., vol. 21, no. 4, pp. 1324-1342, 2000. 6, 7

[22] C. Dicle, M. Sznaier and O. Camps. The way they move: Tracking multiple targets with similar appearance. Proc. IEEE Int'l Conf. Computer Vision, 2013. 2, 10, 11, 12, 13, 14

[23] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. IEEE Trans. Pattern Analysis and Machine Interlligence, vol. 33, no. 12, pp. 2383-2395, 2011. 7

[24] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. 1

[25] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. Proc. IEEE Conf. Decision and Control, vol. 19, pp. 807-812, Dec. 1980. 2

[26] W. Ge, R. T. Collins, and R. Ruback. Vision-based analysis of small groups in pedestrian crowds. IEEE Trans. Pattern Analysis and Machine Interlligence, vol. 34, no. 5, pp. 1003-1016, May. 2012. 2, 10

[27] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012. 2, 11, 13

[28] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. Proc. European Conf. Computer Vision, 2008. 2

[29] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2007. 2

[30] N. Le, A. Heili, and J. Odobez. Long-term time-sensitive costs for crf-based tracking by detection. Proc. European Conf. Computer Vision, 2016. 3

[31] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese CNN for robust target association. Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, 2016. 3, 13, 14

[32] L. Leal-Taixé, A. Milan, I. Reid, S. Roth and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 [cs], 2015. 2, 11, 13

[33] P. Lenz, A. Geiger and R. Urtasun. FollowMe: Efficient online min-cost flow tracking with bounded memory and computation. Proc. IEEE Int'l Conf. Computer Vision, 2015. 2, 13

[34] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009. 11

[35] W. Luo, J. Xing, X. Zhang, X. Zhao, and T-K. Kim. Multiple object tracking: A review. arXiv:1409.7618, 2014. 2

[36] A. Milan, S. Rezatofighi, A. Dick, I. Reid and K. Schindler. Online multi-target tracking using recurrent neural networks. Proc. AAAI, 2017. 3, 13, 14

[37] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. IEEE Trans. Pattern Analysis and Machine Interlligence, vol. 36, no. 1, pp. 58-72, Jan. 2014. 3, 13, 14

[38] A.Milan, K. Schindler, and S. Roth. Multi-target tracking by discrete-continuous energy minimization. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pp. 2054-2068, Sep. 2016. 3, 13, 14

[39] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for multi-target tracking. IEEE Trans. Automatic Control, vol. 54, no. 3, pp. 481-497, Mar. 2009. 3

[40] K. Okuma, A. Taleghani, O.D. Freitas, J.J. Little, and D.G. Lowe. A boosted particle filter: Multitarget detection and tracking. Proc. European Conf. Computer Vision, 2004. 2

[41] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy

algorithms for tracking a variable number of objects. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2011. 1, 2, 5, 11, 12, 13, 14

[42] A. Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. Computational Optimization and Applications, vol. 3, no. 1, pp. 27–57, 1994. 1, 4

[43] H. Possegger, T. Mauthner, P.M. Roth, H. Bischof. Occlusion geodesics for online multi-object tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2014. 2

[44] P. Regalia and E. Kofidis. The higher-order power method revisited: Convergence proofs and effective initialization. Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing, 2000. 6, 7

[45] D. Reid. An algorithm for tracking multiple targets. IEEE Trans. Automatic Control, vol. 24, no. 6, pp. 843-854, Dec. 1979. 2, 9

[46] S. Ricardo, P. Fabio, and C. Andrea. Online multi-target tracking with strong and weak detections. European Conf. Computer Vision, 2016. 2

[47] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker. Deep network flow for multi-object tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2017. 3

[48] K. Shafique, M. Lee, and N. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking problem. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008. 1, 2

[49] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 51-65, Jan. 2005. 10

[50] X. Shi, H. Ling, J. Xing, and W. Hu. Multiple target tracking by rank-1 tensor approximation. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2013. 2

[51] X. Shi, H. Ling, E. Blash, and W. Hu. Context-driven moving vehicle detection in wide area motion imagery. Proc. IEEE Int'l Conf. Pattern Recognition, 2012. 10

[52] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann. Math. Statist, vol. 35, pp.876-879, 1964. 8

[53] J. Son, M. Baek, M. Cho, and BY. Han. Multi-object tracking with quadruplet convolutional neural networks. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2017. 3

[54] S. Tang, B. Andres, M. Andriluka and B. Schiele. Subgraph decomposition for multitarget tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2015. 3

[55] S. Tang, B. Andres, M. Andriluka and B. Schiele. Multi-person tracking by multicut and deep matching. European Conf. Computer Vision, 2016. 3

[56] B. Wang, L. Wang, B. Shuai, Z. Zhen, T. Liu, K.C. Chan, and G. Wang. Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, 2016. 3

[57] S. Wang, and C. Fowlkes. Learning optimal parameters for multi-target tracking with contextual interactions. Int'I J. Computer Vision, 2016. 2, 13, 14

[58] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S.Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2014. 3

[59] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012. 12

[60] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012. 12, 13

[61] Q. Yu and G. Medioni. Multiple target tracking by spatiotemporal monte carlo markov chain data association IEEE Trans. Automatic Control, vol. 31, no. 12, pp. 2196-2210, Dec. 2009. 3

[62] A. Zamir, A. Dehghan, and M. Shah. GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs. Proc. European Conf. Computer Vision, 2012. 3

[63] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008. 1, 2, 5