Multi-target Tracking by Rank-1 Tensor Approximation

Xinchu Shi[1,2], Haibin Ling[2], Junliang Xing[1], Weiming Hu[1]
[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2]Department of Computer and Information Science, Temple University, Philadelphia, USA
{xcshi, jlxing, wmhu}@nlpr.ia.ac.cn, hbling@temple.edu

Abstract

In this paper we formulate multi-target tracking (MTT) as a rank-1 tensor approximation problem and propose an ℓ_1 norm tensor power iteration solution. In particular, a high order tensor is constructed based on trajectories in the time window, with each tensor element as the affinity of the corresponding trajectory candidate. The local assignment variables are the ℓ_1 normalized vectors, which are used to approximate the rank-1 tensor. Our approach provides a flexible and effective formulation where both pairwise and high-order association energies can be used expediently. We also show the close relation between our formulation and the multi-dimensional assignment (MDA) model. To solve the optimization in the rank-1 tensor approximation, we propose an algorithm that iteratively powers the intermediate solution followed by an ℓ_1 normalization. Aside from effectively capturing high-order motion information, the proposed solver runs efficiently with proved convergence. The experimental validations are conducted on two challenging datasets and our method demonstrates promising performances on both.

1. Introduction

Multi-target tracking (MTT) is critical for many applications, ranging from vision-based surveillance to human-computer interaction. Roughly speaking, existing approaches can be sorted into two categories: sequential tracking and association-based tracking. The former tracks multiple targets with observations till the current frame, while the latter collects a batch of evidences within a time span and treat tracking as a multi-frame multiple target association[1] problem. Sequential tracking is suitable for online tasks [6, 24], but sometimes meets problems when dealing with target occlusions. By contrast, association-based tracking [21, 26, 25, 26, 18, 16, 5, 2] recently becomes popular, since it uses batch observations to reduce the association ambiguity and takes benefit from recent advances in object detection [9, 13].

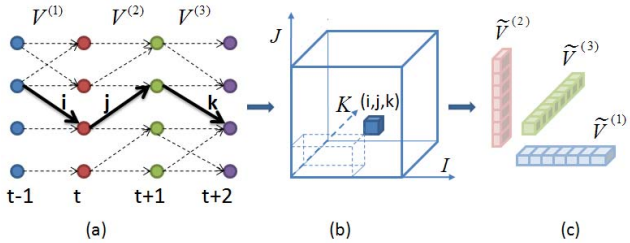

Figure 1. Relation of tensor and association, with a 4-frame association as an example. (a) Association candidates. Each global trajectory constitutes of 3 local associations. (b) 3-order trajectory tensor. Each trajectory has a corresponding tensor item computed from the trajectory affinity. (c) Rank-1 tensor approximation. Resulting vectors $\widetilde{V}^{(1)}, \widetilde{V}^{(2)}, \widetilde{V}^{(3)}$ are real-value solutions of local assignment variables $V^{(1)}, V^{(2)}, V^{(3)}$.

Many association-based models can be formulated as a *multi-dimensional assignment* (MDA) problem [19, 11]. However, the integer optimization in MDA is NP-hard for three or higher dimensional association in general. Some alternative works evade the global association by using hierarchical strategies [15, 7], the optimum local associations are achieved first and used to obtain longer tracks later. Network flow methods [18, 26, 5] formulate MDA as the max-flow/min-cut optimization, and globally optimal solution with polynomial time complexity is available. These methods are however restricted to the use of pairwise cost and could not benefit from rich multiple-frame cues.

In this work, we propose a tensor based approach for multi-frame multi-target association. First, we construct a high-order tensor from all trajectory candidates over a time span, as illustrated in Figure 1. Then, we show that the rank-1 approximation of this tensor has the same energy formulation as the multi-dimensional assignment. Finally, an ℓ_1 tensor power iteration with row/column unit norm is introduced to solve the approximation problem.

The proposed tensor-based solution has two major advantages. First, it enables us to capture information across multiple frames up to the entire trajectory. As a result, our algorithm easily integrates powerful and discriminative

[1]Through the paper, (multi-frame) association and (multi-dimensional) assignment have the same meaning, they are used alternately later.

cues such as high order motion information and high order appearance variation. Second, the proposed iterative solution has low computation complexity and its convergence proof is provided in this paper. To validate the proposed method, we apply it to multi-target tracking using two challenging datasets, one containing wide area motion sequences and the other containing public area surveillance videos. Promising results of the proposed approach are observed on both datasets.

The rest of the paper is structured as follows. Related work is given in Section 2. Tensor formulation and our approach are described in Section 3 and Section 4 respectively. Experimental results are presented in Section 5, and Section 6 concludes the paper.

2. Related work

Study of data association has a long history, with early research focusing on radar target tracking [3], where Multiple Hypothesis Tracking (MHT) [21] is the classic method. With a batch of observations, MHT finds all possible association combinations and selects the most likely association set as the optimal solution. In general, MHT optimization is an NP-hard problem and the computation is prohibitive when the numbers of objects and frames are large.

Multiple target association across multiple frames can be formulated as the multiple dimensional assignment problem. Suppose a sequence of K frames with each frame has N observations, the formulation of MDA is presented as

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

$$\text{s.t.} \begin{cases} \sum_{i_1} \dots \sum_{i_{j-1}} \sum_{i_{j+1}} \dots \sum_{i_K} x_{i_1 \dots i_K} = 1, 1 \le j \le K \\ x_{i_1 \dots i_K} \in \{0,1\}, 1 \le i_1, \dots, i_K \le N \end{cases} \tag{2}$$

where $a_{i_1 \dots i_K}$ is the affinity of the trajectory $\{i_1, \dots, i_K\}$ whose label $x_{i_1 \dots i_K}$ is 1 when the trajectory is true and 0 otherwise; i_j denotes the observation index in j-th frame.

Two-frame association is a special case of MDA, and exact solutions with polynomial time such as Hungarian algorithm are available. However, the solution is NP-hard when the association is computed over three or more frames. It is impractical to achieve the global optimum solution when no assumption is used. However, there exist some approximate solutions, by using semi-definite programming [22], Lagrange relaxation [11] etc. When the cost of the trajectory is decomposed as the product of pairwise terms, MDA can be formulated as a network flow problem, which can be solved by using linear programming [16], shortest path algorithms [5], etc. Such network flow formulation, while having global optimal solutions with polynomial time complexity, is limited to use pairwise affinity and misses high order kinematic information.

Sampling-based approaches (eg. Markov Chain Monte Carlo Data Association [17, 4]) provide an alternative to find the global solution. However, they typically require large computational cost, especially for the high-dimensional state estimation in MDA. Furthermore, tuning the parameters to obtain a fast convergence is always non-trivial. Other approaches for solving MDA include greedy search [23, 25] and hierarchical target association [15, 7]. Our work is closely related to the iterative approximate solution proposed in [8], which iteratively solves two-frame assignments in turn while keeping all other assignments fixed. Our approach shares a similar procedure, but we use the tensor framework and propose an analytical iterative solution. In addition, the association ambiguity is retained in our iteration, which reduces the association errors.

3. Tensor formulation

In this section, we give a brief introduction about tensor and its rank-1 approximation. A tensor is the high dimensional generalization of a matrix. For a K-order tensor[2] $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_K}$, each element is represented as $s_{i_1 \dots i_k \dots i_K}$ and $1 \le i_k \le I_k$. In the tensor terminology, each dimension of a tensor is associated with a *mode*. Like matrix-vector and matrix-matrix multiplication, tensor has similar operations, we give the following definition.

Definition 1 *The n-mode* product *of a tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times \dots I_{n-1} \times I_n \times \dots I_K}$ and a matrix $\mathbf{E} \in \mathbb{R}^{I_n \times J_n}$, denoted by $\mathcal{S} \otimes_n \mathbf{E}$, is a new tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots I_{n-1} \times J_n \times \dots I_K}$. The notation is represented as*

$$\begin{aligned} &\mathcal{B} = \mathcal{S} \otimes_n \mathbf{E}\,, \\ &b_{i_1 \dots i_{n-1} j_n i_{n+1} \dots i_K} = \sum_{i_n=1}^{I_n} s_{i_1 \dots i_{n-1} i_n \dots i_K} e_{i_n j_n}\,. \end{aligned} \tag{3}$$

In particular, the n-*mode* product of $\mathcal{S}$ and a vector $\Pi \in \mathbb{R}^{I_n \times 1}$, denoted by $\mathcal{S} \otimes_n \Pi$, is the $K-1$ order tensor

$$(\mathcal{S} \otimes_n \Pi)_{i_1 \dots i_{n-1} i_{n+1} \dots i_K} = \sum_{i_n=1}^{I_n} s_{i_1 \dots i_{n-1} i_n \dots i_K} \pi_{i_n}. \tag{4}$$

3.1. Rank-1 tensor approximation

Before we introduce Rank-1 tensor approximation, the notation of Rank-1 tensor is given first. If the K order tensor $\mathcal{S}$ is computed as the outer product of K vectors $\Pi^{(1)}, \Pi^{(2)}, \dots, \Pi^{(K)}$, we call $\mathcal{S}$ a rank-1 tensor. Specifically, we denote the rank-1 tensor as

$$\mathcal{S} = \Pi^{(1)} * \Pi^{(2)} * \dots * \Pi^{(K)}, \tag{5}$$

[2]Through the whole paper, by default we use font such as $\mathcal{S}$ for a tensor, $\mathbf{S}$ a matrix, S a vector and s a scalar number.

and

$$\left(\Pi^{(1)} * \Pi^{(2)} * \ldots * \Pi^{(K)}\right)_{i_1 i_2 \ldots i_K} = \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} \ldots \pi^{(K)}_{i_K}, \quad (6)$$

where $\Pi^{(k)}$ denotes the k-th ($1 \leq k \leq K$) vector and $\pi^{(k)}_{i_k}$ denotes the i_k-th ($1 \leq i_k \leq I_k$) element of $\Pi^{(k)}$.

With the above definition, the problem of rank-1 approximation for tensor $\mathcal{S}$ is formulated as following:

Problem 1 *Given a real K order tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, find K unit-norm vectors $\Pi = \{\Pi^{(1)}, \Pi^{(2)}, \ldots, \Pi^{(K)}\}$ and a scalar λ to minimize the Frobenius norm square*

$$\begin{aligned} \min_{\Pi} f(\lambda, \Pi) &= \min_{\lambda, \Pi} \|\mathcal{S} - \lambda \Pi^{(1)} * \Pi^{(2)} * \ldots * \Pi^{(K)}\|_F^2 \\ &= \min_{\lambda, \Pi} \sum_{i_1 \ldots i_K} \left(s_{i_1 \ldots i_K} - \lambda \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} \ldots \pi^{(K)}_{i_K}\right)^2 . \end{aligned} \quad (7)$$

Problem 1 can be solved with various techniques such as Lagrange multipliers [10] or least-squares [20]. With some derivations ([10, 20]), the optimization in (7) has the following equivalent form

$$\begin{aligned} &\min_{\lambda, \Pi} \left\|\mathcal{S} - \lambda \Pi^{(1)} * \Pi^{(2)} * \ldots * \Pi^{(K)}\right\|_F^2 \\ &= \min_{\Pi} \left(\|\mathcal{S}\|_F^2 - \left|\mathcal{S} \otimes_1 \Pi^{(1)} \otimes_2 \Pi^{(2)} \ldots \otimes_K \Pi^{(K)}\right|^2\right) . \end{aligned} \quad (8)$$

This naturally leads to the following theorem:

Theorem 1 *The minimization of the function (7) over the unit-norm vectors $\Pi = \{\Pi^{(1)}, \Pi^{(2)}, \ldots, \Pi^{(K)}\}$ is equivalent to the maximization over $g(\Pi)$ defined as*

$$\begin{aligned} g(\Pi) &= \left|\mathcal{S} \otimes_1 \Pi^{(1)} \otimes_2 \Pi^{(2)} \ldots \otimes_K \Pi^{(K)}\right|^2 \\ &= \left(\sum_{i_1 i_2 \ldots i_K} s_{i_1 i_2 \ldots i_K} \pi^{(1)}_{i_1} \pi^{(2)}_{i_2} \ldots \pi^{(K)}_{i_K}\right)^2 . \end{aligned} \quad (9)$$

To maximize (9), tensor power iteration ([10, 20]) is proposed with a sound convergence proof. Though there is no guarantee for the algorithm to reach the global optimum, it always attains satisfactory solutions in graph matching observed in [12]. It also gives a solution very close to the optimum when initialized cleverly ([20]).

3.2. Relations to Multi-Dimensional Assignment

In this section, we show the rank-1 tensor approximation has the similar optimization formulation with MDA, with an appropriate tensor item definition.

First, we reformulate the Eq. (1). Each trajectory (global association) is decomposed as a sequence of edges (two-frame association), which is formulated as

$$x_{i_1 i_2 \ldots i_K} = e_{i_1 i_2} e_{i_2 i_3} \ldots e_{i_{K-1} i_K} , \quad (10)$$

subjects to the new constraints:

$$\begin{cases} \sum\limits_{i_n} e_{i_n i_{n+1}} = 1, & n \in \{1, 2, ..., K-1\} \\ \sum\limits_{i_{n+1}} e_{i_n i_{n+1}} = 1, & n \in \{1, 2, ..., K-1\} \\ e_{i_n i_{n+1}} \in \{0, 1\}, & n \in \{1, 2, ..., K-1\} \end{cases} \quad (11)$$

where $e_{i_n i_{n+1}}$ is the element of the two-frame association matrix $\mathbf{E}^{(n)} = (e_{i_n i_{n+1}})$. Similarly, the affinity representation is reformulated as Eq. (12). Note the affinity remains depending on the entire trajectory but with a different subscript, i.e.

$$a_{i_1 i_2 \ldots i_K} = s_{i_1 i_2, i_2 i_3, \ldots, i_{K-1} i_K} , \quad (12)$$

which will be defined soon.

We vectorize (e.g. by column concatenation) $\mathbf{E}^{(n)}$ into a vector, which we called local assignment variable $\Pi^{(n)}$, the optimization problem (1) is rewritten as

$$\max \sum_{l_1}^{N^2} \sum_{l_2}^{N^2} \ldots \sum_{l_{K-1}}^{N^2} s_{l_1 l_2 \ldots l_{K-1}} \pi^{(1)}_{l_1} \pi^{(2)}_{l_2} \ldots \pi^{(K-1)}_{l_{K-1}} , \quad (13)$$

where $s_{l_1 l_2 \ldots l_{K-1}}$ is defined as

$$s_{l_1 l_2 \ldots l_{K-1}} = \begin{cases} a_{\overline{l_1}\, \underline{l_2} \ldots \overline{l_{K-1}}\, \underline{l_{K-1}}}, & \text{if } \Omega \text{ is true} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

In (14), "$\overline{l}$" and "$\underline{l}$" denote the row and column index of the l-th element of the vector Π, in the matrix $\mathbf{E}$; and Ω is the condition set defined as

$$\Omega : \{\overline{l_2} = \underline{l_1}, \overline{l_3} = \underline{l_2}, ..., \overline{l_{K-1}} = \underline{l_{K-2}}\} . \quad (15)$$

By placing constraints on row and column indices of consecutive two-frame associations using (14), formulations (1) and (13) are equivalent to each other. Also, we note that the affinity values s and the association variables π in (13) are non-negative, thus the equivalence between (9) and (13) is self-evident.

However, the constraints over Π in (9) and that in (13) are different. Specifically, Π in tensor approximation must have the ℓ_2 unit norm, while Π in MDA consumes integer values with the row and column ℓ_1 unit norm. In the next section, we propose a row/column ℓ_1 unit norm tensor power iteration. With this extension, the tensor approximation can be considered as the counterpart of MDA in the real-value domain.

An example illustrating the relation between rank-1 tensor approximation and MDA is given in Figure 1. It shows that (1) tensor elements correspond to global associations; and (2) vectors approximating the rank-1 tensor are real solutions of local assignment variables. Finally, for an intuitive view of Eqn. (7), it aims to minimize the element-wise

Algorithm 1 Tensor based multi-target association

Input. M frame observation sequence.
t_0: Start frame, K:Number of a batch frames
Output: target associations
while $t_0 + K - 1 \leq M$ **do**
  Collect a batch of K frames observations $\Phi = \{O(t_0), O(t_0+1), ..., O(t_0+K-1)\}$
  Generate two-frame association hypotheses.
  Generate global trajectory hypotheses.
  Compute the trajectory affinities and construct the $K-1$ order tensor $\mathcal{S}$.
  Initialize the approximate vectors.
  ℓ_1 row/column unit norm tensor power iteration.
  Solution: Discretize the approximate vectors.
  $t_0 \leftarrow t_0 + K - 1$
end while

Algorithm 2 Tensor power iteration with ℓ_1 unit norm

Input: $K-1$ order tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_{K-1}}$.
Output: ℓ_1 unit norm vectors $\Pi^{(1)}, ..., \Pi^{(K-1)}$.
Initialization: $\Pi_0^{(1)}, ..., \Pi_0^{(K-1)}$; Iteration Num: $j \leftarrow 0$.
repeat
  $\hat{\Pi}_{j+1}^{(1)} = (\mathcal{S} \otimes_2 \Pi_j^{(2)} \otimes_3 \Pi_j^{(3)} ... \otimes_{K-1} \Pi_j^{(K-1)}) \circ \Pi_j^{(1)}$
  $\Pi_{j+1}^{(1)} = \widehat{\Pi}_{j+1}^{(1)} / \|\widehat{\Pi}_{j+1}^{(1)}\|_1$
  $\hat{\Pi}_{j+1}^{(2)} = (\mathcal{S} \otimes_1 \Pi_{j+1}^{(1)} \otimes_3 \Pi_j^{(3)} ... \otimes_{K-1} \Pi_j^{(K-1)}) \circ \Pi_j^{(2)}$
  $\Pi_{j+1}^{(2)} = \widehat{\Pi}_{j+1}^{(2)} / \|\widehat{\Pi}_{j+1}^{(2)}\|_1$
  ...
  $\hat{\Pi}_{j+1}^{(K-1)} = (\mathcal{S} \otimes_1 \Pi_{j+1}^{(1)} \otimes_2 \Pi_{j+1}^{(2)} ... \otimes_{K-2} \Pi_{j+1}^{(K-2)}) \circ \Pi_j^{(K-1)}$
  $\Pi_{j+1}^{(K-1)} = \widehat{\Pi}_{j+1}^{(K-1)} / \|\widehat{\Pi}_{j+1}^{(K-1)}\|_1$
  $j \leftarrow j + 1$
until convergence

reconstruction error between the the trajectory tensor and the reconstructed tensor, which is calculated as the outer product of local assignment vectors. In particular, for a trajectory with a high affinity, the optimization tries to make a high-value outer product to match its affinity. Consequently, the higher the affinity a trajectory has, the more likely it will be picked up in the final solution.

4. Tensor Based Multi-Target Association

In this section, we introduce the tensor based multi-target tracking approach, and Algorithm 1 outlines the framework. Generally, multi-target association is performed with the batch way. When K frame observations are available, association hypotheses (trajectories) are generated first. With all these hypotheses, a tensor is constructed by computing the trajectory affinities. Then, the most important step is the ℓ_1 norm tensor power solution. We highlight the tensor-based multi-target association in the following parts and present details about object detection in the experiment part.

4.1. Relaxed Optimization

We now relax the optimization (13) by allowing different numbers of objects for different frames, resulting in different numbers of local associations, denoted as $I_k, 1 \leq k \leq K-1$. Further, we use "soft" association variables to make the optimization more feasible. We now have

$$\max \sum_{l_1}^{I_1} \sum_{l_2}^{I_2} ... \sum_{l_{K-1}}^{I_{K-1}} s_{l_1 l_2 ... l_{K-1}} \pi_{l_1}^{(1)} \pi_{l_2}^{(2)} ... \pi_{l_{K-1}}^{(K-1)}, \quad (16)$$

$$s.t. \begin{cases} \sum\limits_{i_n} e_{i_n i_{n+1}} = 1, \ \ n \in \{1, 2, ..., K-1\} \\ \sum\limits_{i_{n+1}} e_{i_n i_{n+1}} = 1, \ \ n \in \{1, 2, ..., K-1\} \\ 0 \leq e_{i_n i_{n+1}} \leq 1, \ \ n \in \{1, 2, ..., K-1\} \end{cases} \quad (17)$$

Note that the original tensor power iteration implied in Theorem 1 is designed for ℓ_2 unit norm, thus not suitable for solving (16). Instead, we propose a new algorithm in the following to adapt the ℓ_1 row/column unit norm constraint.

4.2. ℓ_1 Unit Norm Power Iteration

To address the issue raised from the ℓ_1 norm constraint, we advocate an ℓ_1 *unit norm power iteration* algorithm to solve (16). The basic idea is to iteratively update the solution by tensor powering followed by an ℓ_1 unit normalization. The procedure for general rank-1 tensor approximation is presented in Algorithm 2, where "$\circ$" indicates the Hadamard product (element-wise product). Detailed convergence proof is presented in the following.

We first assume the tensor item $s_{l_1 l_2 ... l_{K-1}}$ and the approximate vectors Π are non-negative, thus the optimization of (9) is equivalent to the following optimization

$$\max_{\Pi} g(\Pi) = \max_{\Pi} \sum_{l_1 ... l_{K-1}} s_{l_1 ... l_{K-1}} \pi_{l_1}^{(1)} ... \pi_{i_{K-1}}^{(K-1)}. \quad (18)$$

For clear expression, we denote the k-th vector at the n-th iteration as $\Pi^{(k)}(n)$, which has elements $\pi_{l_k}^{(k)}(n)$. Consider the iteration on $\Pi^{(1)}(n)$, with all other vectors fixed, we has following proposition.

Proposition 1 *For an iteration step (19),*

$$\pi_{l_1}^{(1)}(n+1) = \frac{\pi_{l_1}^{(1)}(n)}{C^{(1)}} \sum_{l_2 ... l_{K-1}} s_{l_1 ... l_{K-1}} \pi_{l_2}^{(2)}(n) \ldots \pi_{l_{K-1}}^{(K-1)}(n), \quad (19)$$

where $C^{(1)} = \sum\limits_{l_1 ... l_{K-1}} s_{l_1 ... l_{K-1}} \pi_{l_1}^{(1)}(n) \ldots \pi_{l_{K-1}}^{(K-1)}(n)$ *is the* ℓ_1 *normalization factor, we have*

$$g(\Pi^{(1)}(n+1), \Pi^{(2)}(n), \ldots, \Pi^{(K-1)}(n)) \geq g(\Pi^{(1)}(n), \Pi^{(2)}(n), \ldots, \Pi^{(K-1)}(n)). \quad (20)$$

Proof. We make two notations $W = (w_1, \ldots, w_{I_1})^T$ and $U = (u_1, \ldots, u_{I_1})^T$, and I_1 is the length of $\Pi^{(1)}$. The definitions of two notations are given as

$$\begin{cases} w_{l_1} &= \sum\limits_{l_2 \ldots l_K} s_{l_1 l_2 \ldots l_k} \pi_{l_2}^{(2)}(n) \ldots \pi_{l_{K-1}}^{(K-1)}(n) \\ \pi_{l_1}^{(1)}(n) &= u_{l_1} * u_{l_1} \end{cases} \tag{21}$$

With above notations, we have following equation

$$\begin{aligned} & g(\Pi^{(1)}(n), \Pi^{(2)}(n), ..., \Pi^{(K-1)}(n)) \\ &= \sum_{l_1 \ldots l_{K-1}} s_{l_1 l_2 \ldots l_{K-1}} u_{l_1} u_{l_1} \pi_{l_2}^{(2)}(n) ... \pi_{l_{K-1}}^{(K-1)}(n) \\ &= \langle U, U \circ W \rangle \ , \end{aligned} \tag{22}$$

where '$\langle \cdot \rangle$' and '$\circ$' denote the inner product and the Hadamard product respectively. With the norm constraint $\|U\|_2^2 = \|\Pi^{(1)}(n)\|_1 = 1$, the Cauchy-Schwarz inequality gives

$$\begin{aligned} g(\Pi^{(1)}(n), ..., \Pi^{(K-1)}(n)) &= \langle U, U \circ W \rangle \\ \leq \|U\|_2 \, \|U^{\circ} W\|_2 &= \|U^{\circ} W\|_2 \, . \end{aligned} \tag{23}$$

With the formulation (19), the new score is presented as

$$\begin{aligned} & g(\Pi^{(1)}(n+1), \Pi^{(2)}(n), ..., \Pi^{(K-1)}(n)) \\ &= \left\langle \Pi^{(1)}(n+1), W \right\rangle \\ &= \tfrac{1}{C^{(1)}} \left\langle \Pi^{(1)}(n) \circ W, W \right\rangle \\ &= \tfrac{1}{C^{(1)}} \left\langle U \circ W, U \circ W \right\rangle = \tfrac{\|U \circ W\|_2^2}{g(\Pi^{(1)}(n), ..., \Pi^{(K-1)}(n))} \end{aligned} \tag{24}$$

By combining formulation (23) and (24), we prove the inequality (20). ■

The convergence proof on the iterations of vectors $\Pi^{(k)}(n)$ $(1 < k < K)$ has the similar form, thus is ignored here. Combine results for all k, we have the proposed ℓ_1 unit norm iteration algorithm converges to a (local) extreme.

Algorithm 2 gives the ℓ_1 unit norm vector solution. We note the solution in (16) has matrix row and column ℓ_1 unit norm, so we make an adaption in the normalization. Set the association problem (16) as an example, the ℓ_1 unit norm tensor iteration for $\Pi^{(1)}$ has the formulation as

$$\hat{\Pi}_{j+1}^{(1)} = (\mathcal{S} \otimes_2 \Pi_j^{(2)} \otimes_3 \Pi_j^{(3)} ... \otimes_{K-1} \Pi_j^{(K-1)}) \circ \Pi_j^{(1)}. \tag{25}$$

Followed by row ℓ_1 normalization

$$e_{pq} = \frac{\widehat{e}_{pq}}{\left\| \widehat{\mathbf{E}}(p,:) \right\|_1} = \frac{\widehat{e}_{pq}}{\sum_q \widehat{e}_{pq}} \ , \ p \in \{1, 2, ...\}, \tag{26}$$

where $\widehat{e}_{pq}$ is the element of $\widehat{\mathbf{E}}$, which is the folded matrix of vector $\widehat{\Pi}_{j+1}^{(1)}$. e_{pq} has the similar meaning. The iterations of other vectors follow the similar normalization. The row/column ℓ_1 normalization operation has no effect to the total convergence, which is illustrated in Appendix A.

4.3. Hypothesis generation

We follow the traditional approaches to set a bound for association generation. Basically, we make an association hypothesis between two object candidates from two consecutive frames only when they are spatially close to each other. This strategy is popular in multi-target tracking, since making all associations neither practical nor meaningful. For implementation, we select a distance threshold to guarantee the inclusion of all true associations. Consequentially, the threshold is application dependent.

Finally, an important issue in hypothesis generation is the management of special events, such as target entrance (reappearance) and exit (occlusion). For handling this issue, in each frame we include two dummy targets, a source and a sink, to generate the entrance and exit association for each real target.

4.4. Tensor computation

The tensor $\mathcal{S}$ is constructed based on the global association hypothesis, with the trajectory affinities as the tensor elements. Given different affinity representations, there are different optimization formulations. For example, if the affinity $s_{l_1 l_2 \ldots l_{K-1}}$ can be decomposed as $s_{l_1 l_2 \ldots l_{K-1}} = \sum_{i=1,\ldots,K-1} s_{l_i}^*$, where s^* denotes the two-frame association affinity, the objective (16) can be reformulated as

$$\begin{aligned} & \sum_{l_1}^{I_1} \sum_{l_2}^{I_2} \cdots \sum_{l_{K-1}}^{I_{K-1}} s_{l_1 l_2 \ldots l_{K-1}} \pi_{l_1}^{(1)} \pi_{l_2}^{(2)} \ldots \pi_{l_{K-1}}^{(K-1)} \\ = \ & \Big(\prod_{k \neq 1} I_k \Big) \sum_{l_1}^{I_1} s_{l_1}^* \pi_{l_1}^{(1)} + \ldots + \Big(\prod_{k \neq K-1} I_k \Big) \sum_{l_{K-1}}^{I_{K-1}} s_{l_{K-1}}^* \pi_{l_{K-1}}^{(K-1)} . \end{aligned} \tag{27}$$

Consequently, the two-dimensional assignment is a special case of the tensor framework.

When the affinity is computed as the product of pairwise costs, i.e. $s_{l_1 l_2 \ldots l_{K-1}} = s_{l_1}^* s_{l_2}^* \ldots s_{l_{K-1}}^*$, the score in (16) can be rewritten as

$$\begin{aligned} & \sum_{l_1}^{I_1} \cdots \sum_{l_{K-1}}^{I_{K-1}} s_{l_1 \ldots l_{K-1}} \pi_{l_1}^{(1)} \ldots \pi_{l_{K-1}}^{(K-1)} \\ = \ & \sum_{l_1}^{I_1} s_{l_1}^* \pi_{l_1}^{(1)} \sum_{l_2}^{I_2} s_{l_2}^* \pi_{l_2}^{(2)} \cdots \sum_{l_{K-1}}^{I_{K-1}} s_{l_{K-1}}^* \pi_{l_{K-1}}^{(K-1)} \end{aligned} \tag{28}$$

We note (28) is the same as the formulation in network flow ([5]), thus network flow is a special case of the proposed tensor framework.

To summarize, tensor approximation provides a flexible framework to take advantage of global and local association energy. Detailed representation about the affinity computation is presented in the experiment section.

4.5. Initialization and termination

The initial point is important for tensor iteration. In our work, we make the uniform initialization. For exam-

ple, when one target has 4 association candidates, the initial value for each candidate is $1/4$. The algorithm terminates when the predetermined iteration number is reached, or when the gain of the objective function is below a threshold.

The solution given by tensor power iteration is real-value, which must be discretized to meet the integer and one-to-one mapping constraints in the assignment. To leverage the conflicts between different local association candidates, we treat the real-value solutions as the costs for corresponding association candidates and feed them into a bipartite problem. Then, we apply the Hungarian algorithm to obtain the binary output.

5. Experiments

We test the proposed approach on two challenging datasets. One is the wide aerial motion imagery (WAMI): CLIF [1], with which we implement multiple vehicle tracking. The other is the pedestrian walking sequences used in [8, 14], we called it PSUdata.

5.1. CLIF data

Our first experiment is conducted on the Columbus Large Image Format (CLIF) dataset ([1]) for multiple moving targets tracking. The dataset is challenging in the following aspects: 1) large image format (4016×2672), 2) large camera and target motion, 3) tiny target occupy (4∼70 pixels), 4) similar target appearances (gray image), 5) low frame rate sampling ($\leq$2 fps) and 6) a large mount of targets (dozens to hundreds).

In this experiment, we compare our approach with the other two approaches. One is the *iterated conditional modes* (ICM)-like algorithm presented in ([8]), which we denote as " ICM-like". The other is the classical Hungarian assignment. Three methods are tested on three sequences, each constitutes of 100 frames (50 seconds). One sequence describes the heavy traffic scene, with more than 200 vehicles on the road. The other two sequences are more sparse, but there are still more than 80 targets in each frame. We note there is no ground truth about CLIF dataset, therefore we spend much time on labeling the three sequences.

Vehicle detection in the wide area surveillance (WAS) constitutes of two components, motion detection and object classification. First, background modeling with median filter is performed and used to obtain foreground blobs. Second, the trained SVM classifier using HOG [9] features is applied to remove false positives.

The affinity of a trajectory is defined as

$$s_{l_1 l_2 \ldots l_{K-1}} = a_{l_1} a_{l_2} \ldots a_{l_{K-1}} m_{l_1 l_2 \ldots l_{K-1}}, \quad (29)$$

where a_{l_t} is the affinity of the local association l_t and is computed using histogram appearance and bounding box area features; $m_{l_1 l_2 \ldots l_{K-1}}$ is the global motion affinity defined as

$$m_{l_1 \ldots l_{K-1}} \propto \prod_{t=1}^{K-2} \exp\left(\frac{U_{l_t} U_{l_{t+1}}^T}{\|U_{l_t}\| \|U_{l_{t+1}}\|} + \frac{2\|U_{l_t}\| \|U_{l_{t+1}}\|}{\|U_{l_t}\|^2 + \|U_{l_{t+1}}\|^2}\right), \quad (30)$$

where U_{l_t} is the velocity vector of association l_t. We set K as 5 in both our approach and ICM-like.

We use the same affinity (29) in our approach and ICM-like, and a_{l_t} is used as the cost in Hungarian algorithm. The quantitative results are presented in Table 1. The *correct match percentage* P_c and *wrong match percentage* P_w are computed as equation (31)

$$P_c = 100 \times \frac{\sum_t cm(t)}{\sum_t g(t)}, \; P_w = 100 \times \frac{\sum_t wm(t)}{\sum_t g(t)}, \quad (31)$$

where $cm(t)$ and $wm(t)$ represent numbers of correct and wrong associations (ID switch) in frame t, $g(t)$ denotes the number of ground truth targets at frame t.

Among the three algorithms, Hungarian assignment performs the worst, since it is the two-dimensional local solution and the high-order motion is not captured. Further, our approach performs better than ICM-like. One reason lies in that the association ambiguity is retained in the iteration process in our approach till final decision.

Figure 2 gives the multi-vehicle tracking trajectories of our approach. It can be seen that the application is especially difficult, with a large amount of tiny and confusing targets. Despite the difficulty, most trajectories provided by our algorithm are correct.

Table 1. Evaluation results on the CLIF dataset. Sparse scene: Seq1 and Seq 2. Dense scene: Seq3.

	Correct match percentage			Wrong match percentage		
	Ours	ICM	Hun	Ours	ICM	Hun
Seq1	**91.1**	83.1	75.8	**11.9**	16.5	25.2
Seq2	**92.1**	89.6	86.8	**9.4**	10.3	11.3
Seq3	**91.4**	87.3	83.3	**9.4**	12.9	16.5

5.2. PSUdata

PSUdata constitutes of two sequences of trajectories from pedestrians walking in an atrium. One is relatively sparse, with about 3∼5 people per frame. The other is a dense sequence with more than 20 people per frame, thus more difficult. A major challenge of PSUdata is the absence of appearance information, which is often used by pedestrian tracking algorithms.

In this experiment, we compare the proposed tensor-based association with ICM-like. We use the same affinity in [8] defined as

$$\begin{aligned} s &= \mathrm{E}_0 - \mathrm{E}_{cont} - \mathrm{E}_{curv} \\ &= \mathrm{E}_0 - \eta \sum_{i=2}^{K} \|p_i - p_{i-1}\| - \sum_{i=2}^{K-1} \|p_{i+1} + p_{i-1} - 2p_i\|, \end{aligned} \quad (32)$$

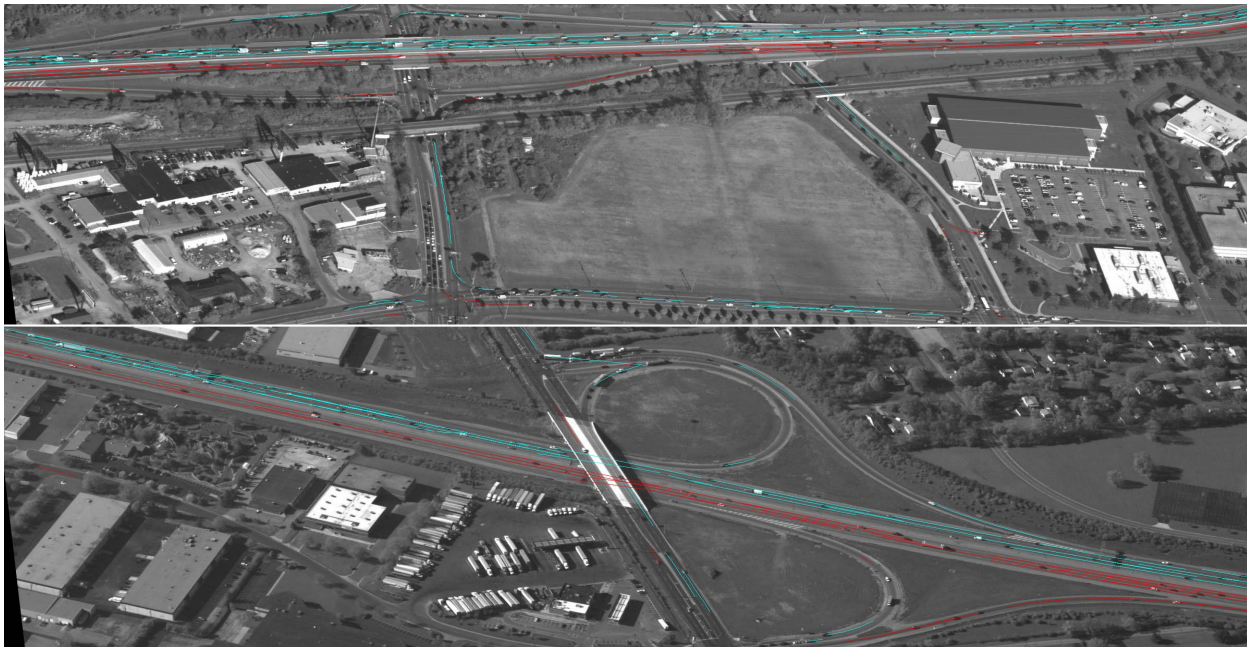

Figure 2. Vehicle tracking with the proposed approach. Top: dense traffic scene. Bottom: sparse scene. Red line indicates tracking results with a rightward motion direction and blue is the opposite. Best viewed with color printing and enlarged size.

Table 2. Evaluation results on the sparse scene of the PSUdata

	Correct match percentage		Wrong match percentage	
	Ours	ICM	Ours	ICM
3fps	**99.99**	99.95	0.00	0.00
2fps	**99.98**	99.97	**0.00**	0.01
1fps	**99.45**	98.87	**0.50**	0.97

Table 3. Evaluation results on the dense scene of the PSUdata

	Correct match percentage		Wrong match percentage	
	Ours	ICM	Ours	ICM
3fps	**99.94**	99.91	**0.05**	0.08
2fps	**99.78**	99.74	**0.20**	0.24
1fps	**96.98**	93.63	**3.01**	6.26

where η is the weighting parameter (set as 0.5); p_i is the target position in frame i; E_0 is a large constant to make the affinity positive; E_{cont} penalizes large position jumps between successive point pair; and E_{curv} defines the constant-velocity motion model. We use the 6 frames as a batch in both our approach and ICM-like.

The quantitative results[3] are presented in Tables 2 and 3, where we used the same metric (31) to evaluate the performance of two approaches. It can be seen that our method performs better than ICM-like in most cases, especially in the difficult 1fps dense scenarios. The large motion offsets of targets in the low frame-rate dense scene cause multiple association possibilities, which confuse the association algorithms. Our approach deals with the problem by retaining the association ambiguity till the final binarization stage, thus acquires a better result than does ICM-like. The performance gains of our approach in sparse and higher frame-rate sequences are small, since the results of ICM-like are close to saturation, and there are less association ambiguities for the data too.

[3]Our implementation of ICM-like generates similar but not identical results as the original one [8]. We list results from our implementation since [8] only reports wrong percentage but not correct percentage.

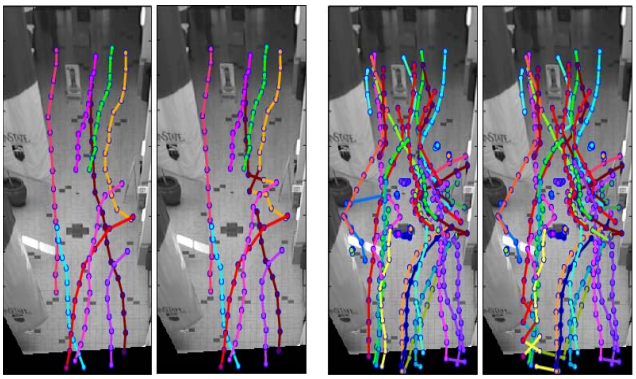

Figure 3. Multi-target association results of two approaches on PSUdata. Left half for sparse data and right half for dense association. Far left: tensor-based association, with 0 ID switch. Second from left: ICM-like association, with 4 ID switches. Second from right: tensor-based association, with 9 ID switches. Far right: ICM-like association, with 23 ID switches.

The qualitative experiment is presented in Figure 3. It can be seen our approach performs better than the ICM-like method in both sparse and dense scenarios.

5.3. Complexity analysis

Both our approach and ICM-like iterate on the global trajectory. The global affinity table is computed firstly. Tensor based association has a computation complexity of $O(fn)$ in each iteration, n is the length of the table (i.e., number of non-zero items) and f is the number of frames. By contrast, ICM-like has a complexity of $O(mn)$, where m is the total number of two-frame association candidates. Because the iteration on each variable in our approach only needs lookup-table operations, while the iteration of ICM-like on each variable needs the global search across the table. Generally, our approach is more efficient since $f \ll m$.

6. Conclusion

In this work, we first consider the global trajectory as the high-order tensor item, and formulate the multiple dimensional assignment task as the (row/column) constrained tensor approximation problem. Further, an ℓ_1 unit norm tensor power iteration algorithm is proposed to solve the optimization, and we provide the convergence proof. The two features in our approach, using global trajectory affinity and maintaining the association ambiguity, advance the global association performance. Experiments on two challenging datasets demonstrate the excellent capability of our approach.

Acknowledgement. This work is partly supported by NSFC (Grant No. 60935002), the National 863 High-Tech R&D Program of China (Grant No. 2012AA012504), the Natural Science Foundation of Beijing (Grant No. 4121003), and the Guangdong Natural Science Foundation (Grant No. S2012020011081). Ling is supported in part by NSF (Grant No. IIS-1218156).

A. Power iteration for row/column unit norm

Given a matrix $\mathbf{E} \in \mathbb{R}^{M\times N}$, constitutes of elements $e_{pq}(1 \le p \le M, 1 \le q \le N)$. We represent the unfolding vector as Π, which is organized as equation (33).

$$\begin{aligned}\Pi &= (\mathbf{E}(1,\cdot), \mathbf{E}(2,\cdot), ..., \mathbf{E}(M,\cdot)) \\ &= (e_{11}, .., e_{1N}, e_{21}, .., e_{2N}, ..., e_{M1}, .., e_{MN}).\end{aligned} \tag{33}$$

We borrow the definition of W in (21), and formulate it as (34), where $W_p \in \mathbb{R}^{1\times N}(1 \le p \le M)$.

$$W = (w_1, w_2, ..., w_{M\times N}) = (W_1, ..., W_M). \tag{34}$$

With the formulation (33) and (34), the score is represented as

$$\begin{aligned}g(\Pi^{(1)}, ..., \Pi^{(K-1)}) &= \sum_{l_1 ... l_{K-1}} s_{l_1 ... l_{K-1}} \pi^{(1)}_{l_1} ... \pi^{(K-1)}_{l_{K-1}} \\ &= \langle \Pi^{(1)}, W\rangle = \sum_{p=1}^{M} \langle \mathbf{E}(p,\cdot), W_p\rangle.\end{aligned} \tag{35}$$

It can be seen the total score constitutes of M partial scores. As each partial score has a raise (no decrease) after the ℓ_1 unit norm iteration, the total score converges to the extreme with each row/column unit iteration.

References

[1] CLIF dataset. www.sdms.afrl.af.mil/index.php?collection=clif2006. 6

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 1

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

[4] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011. 2

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011. 1, 2, 5

[6] 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. *TPAMI*, 33(9):1820–1833, 2010. 1

[7] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011. 1, 2

[8] R. Collins. Multitarget data association with higher-order motion models. In *CVPR*, 2012. 2, 6, 7

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 6

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000. 3

[11] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. A generalized sd assignment algorithm for multisensor-multitarget state estimation. *TAES*, 33(2):523–538, 1997. 1, 2

[12] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *TPAMI*, 33(12):2383–2395, 2011. 3

[13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010. 1

[14] W. Ge, R. Collins, and R. Ruback. Vision-based analysis of small groups in pedestrian crowds. *TPAMI*, 34(5):1003–1016, 2012. 6

[15] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008. 1, 2

[16] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 1, 2

[17] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for multi-target tracking. *TAC*, 54(3):481–497, 2009. 2

[18] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 1

[19] A. Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Comput. Optim. Appl.*, 3(1):27–57, 1994. 1

[20] P. Regalia and E. Kofidis. The higher-order power method revisited: convergence proofs and effective initialization. In *ICASSP*, 2000. 3

[21] D. Reid. An algorithm for tracking multiple targets. *TAC*, 24(6):843–854, 1979. 1, 2

[22] K. Shafique, M. Lee, and N. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking problem. In *CVPR*, 2008. 2

[23] Z. Wu, N. Hristov, T. Hedrick, T. Kunz, and M. Betke. Tracking a large number of objects from multiple views. In *CVPR*, 2009. 2

[24] J. Xing, H. Ai, L. Liu, and S. Lao. Multiple player tracking in sports video: a dual-mode two-way bayesian inference approach with progressive observation modeling. *TIP*, 20(6):1652–67, 2011. 1

[25] A. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012. 1, 2

[26] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 1