Groupwise Tracking of Crowded Similar-Appearance Targets from Low-Continuity Image Sequences

Hongkai Yu[1*], Youjie Zhou[1*], Jeff Simmons[2], Craig P. Przybyla[2],
Yuewei Lin[1], Xiaochuan Fan[1], Yang Mi[1], and Song Wang[1,†]
[1]University of South Carolina, Columbia, SC 29208, [2]Air Force Research Lab, Dayton, OH 45433

Abstract

Automatic tracking of large-scale crowded targets are of particular importance in many applications, such as crowded people/vehicle tracking in video surveillance, fiber tracking in materials science, and cell tracking in biomedical imaging. This problem becomes very challenging when the targets show similar appearance and the inter-slice/inter-frame continuity is low due to sparse sampling, camera motion and target occlusion. The main challenge comes from the step of association which aims at matching the predictions and the observations of the multiple targets. In this paper we propose a new groupwise method to explore the target group information and employ the within-group correlations for association and tracking. In particular, the within-group association is modeled by a nonrigid 2D Thin-Plate transform and a sequence of group shrinking, group growing and group merging operations are then developed to refine the composition of each group. We apply the proposed method to track large-scale fibers from microscopy material images and compare its performance against several other multi-target tracking methods. We also apply the proposed method to track crowded people from videos with poor inter-frame continuity.

1. Introduction

Automatic tracking of large-scale crowded targets has been attracting more and more attention in the computer vision community for its important applications. In video surveillance, tracking of all the individual persons in a crowd can promote the public security [29] by detecting the group behaviors and individual anomalies. In materials science, accurate tracking of large-scale fibers from a sequence of serial sectioned slices of a composite material can facilitate the fast characterization of its underlying microstructure and accelerate the new material design and development [23]. In biomedical imaging, accurately tracking the motion paths of the large-scale cells can provide important information for computer aided diagnosis [18].

One main challenge of the crowded target tracking lies in the step of *association*, which aims at matching the predictions and the observations of the multiple targets at a new slice/frame. In many applications, such as cell and fiber tracking [23], different targets may share similar appearance, which increases the chances of the mis-association. In addition, many of the above applications may generate image sequences with low inter-frame/inter-slice continuity, which increases the gap between a prediction and its corresponding observation and further increases the mis-association rate. For example, in both biomedical and material imaging, it is highly desired to perform sparse sampling along the image sequence for fast imaging. In crowded human tracking, sudden movement of the camera, which is common when using wearable cameras, together with target occlusions, may also produce video clips with low inter-frame continuity.

In this paper, we develop a new groupwise association method to enable the tracking of large-scale crowded targets with similar appearance from *low continuity* image sequences. We explore the target groups in a way that the targets in a same group show relatively consistent motions and therefore, the tracking of the targets in a same group shows high level of correlation. Specifically, we utilize a 2D nonrigid Thin-Plate Splines (TPS) transform to describe the mapping between the predictions and the associated observations within a same group. However, the group composition of the targets are unknown priorly. To address this issue, we develop a new algorithm that clusters the targets into groups, following by three steps of group refinement: shrinking, growing and merging. The proposed method can handle false positives and false negatives in the observations, i.e., the numbers of predictions and observations may be different in the association.

In this paper we choose the task of fiber tracking from a sequence of microscopy material images for algorithm development. The fiber tracking is a typical crowded track-

[*] Indicates equal contribution.
[†] Corresponding author: songwang@cec.sc.edu.

ing problem that possesses all the challenges as described above: 1) the fibers are highly crowded and of large scale, 2) all the fibers on the 2D slice are of similar appearance and bear a similar ellipsoidal shape, and 3) fast material imaging requires the sampled image sequence to be as sparse as possible, which leads to low inter-slice continuity. Given the ellipsoidal shape of the fiber in each slice, we can simply apply ellipse-detection algorithms to compute the fiber observations and focus our work on addressing the association problem. In the experiment, we test the proposed method on real material image sequences and evaluate the fiber tracking performance against the human annotated ground-truth fiber tracks. In particular, we evaluate the fiber tracking performance at different sampling sparsity along the image sequence. In the experiment, we also apply the proposed method to track crowded people with low inter-frame continuity.

For the remainder of the paper, Section 2 briefly reviews the related work. Section 3 describes the proposed method. Experiment results are presented in Section 4, followed by conclusions in Section 5.

2. Related Work

Both recursive and non-recursive methods were developed for multi-target tracking. *Recursive tracking methods* estimate the state of the target in a new slice (frame for videos) only using the information from previous slices that have been processed. Typical recursive tracking methods include the classical Kalman filter [14, 5, 25], Particle filter [7, 21] and non-parametric mixture Particle filters [27]. When tracking moves to a new slice, these recursive methods first make a prediction of the state for each target, then build association between predictions and observations of multiple targets, and finally correct the states using observations. *Non-recursive tracking methods* assume the availability of the whole image sequence before tracking multiple targets over this sequence. In these methods, observations of multiple targets are first detected on all the slices and then linked across slices along the image sequence for the final tracks by optimizing certain cost functions, such as maximum a posteriori (MAP) [13, 30, 4, 22]. In [10], motion dynamics similarity is incorporated into the cost function, resulting in a non-recursive SMOT tracking method for multi-target video tracking. In [18], a KTH tracking method is developed by searching shortest path in the constructed graph model and this method was successfully used to track living cells in biomedical imaging. In [20], a CEM tracking method was proposed by optimizing a continuous cost function that considers the detection, appearance, motion priors, and physical constraints of the targets.

For recursive methods, association is usually formulated as an explicit step which finds the matching between the predictions and the observations. For non-recursive methods, the association is implied in the cost function and the optimization algorithm – the extracted paths find the associations between the observations across neighboring slices. Most of the existing work on multi-target tracking, especially the non-recursive methods, handle only small number of scattered targets with different appearance [13, 30, 4, 22, 10, 20].

Recently, new models and methods have been developed for crowd tracking and analysis [2, 19, 32, 26, 1, 31, 12, 24]. Many of these methods employ tracklets or trajectories, which are extracted by optical flow and/or the local appearance matching across frames. These methods require very good inter-slice/inter-frame continuity. Differently, this paper is focused on tracking large-scale, crowded, similar-appearance targets from *low-continuity* image sequences.

3. Proposed Method

As mentioned above, we present the proposed method using large-scale fiber tracking from a material image sequence. For simplicity, we use the Kalman filter to track each fiber, by recursively performing prediction, association and correction along the image sequence. The key contribution of this paper is the development of a new group-wise algorithm for association, which enables the developed method to track along low-continuity image sequences. The pipeline of the proposed tracking algorithm is illustrated in Fig.1.

When the tracking moves to a new slice, we first compute a set of fiber observations from this new slice. As shown in Fig.1, most of the fibers are of an ellipse shape in the 2D slices and can be detected using an ellipse detection algorithm. In this paper, we first apply the EM/MPM algorithm [9] to segment the image slice into fiber and non-fiber regions and then fit an ellipse to each connected component of the fiber region [28]. We take the locations (the center coordinates) of the fitted ellipses as the fiber observations and use them for association and tracking. In addition to the locations, we also record the tight bounding box around each fitted ellipse, which are used by several comparison methods in the later experiment. Given the image noise and blurs, observations contain both false positives and negatives.

In using Kalman filter for tracking each fiber, we define the tracking state $\mathbf{s} = (x, y, v_x, v_y)^T$ to describe the tracked fiber in 2D slices, where $\mathbf{z} = (x, y)^T$ is the fiber location (e.g., ellipse center) and (v_x, v_y) is the fiber velocity (e.g., inter-slice fiber location change). We set the state transition matrix to be $\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and the observation matrix to be $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. Let $\hat{\mathbf{s}}^i$ be the computed prediction

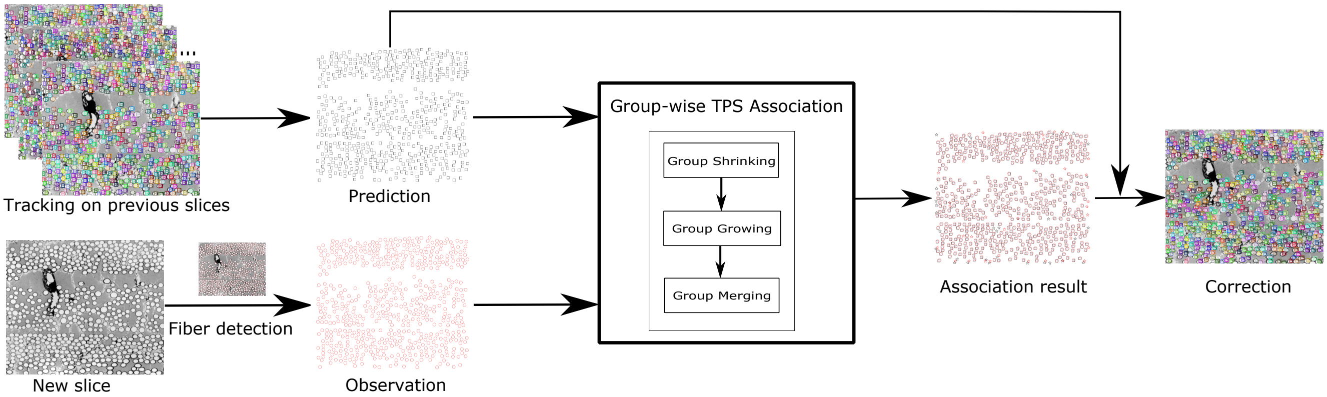

Figure 1. The pipeline of the proposed method used for large-scale fiber tracking. It follows the classical Kalman filtering which sequentially performs prediction, association and correction when moving into a new slice.

of the fiber and $\mathbf{o}^i = [x_\mathbf{o}, y_\mathbf{o}]^T$ to be the observation of the fiber on slice i. We can follow the correction step of Kalman filter to update the state. However, we have large-scale fiber predictions and observations on a slice. In the following, we focus on the step of association that builds a matching between the predictions and observations.

3.1. Thin-Plate Spline Robust Point Matching (TPS-RPM) [8]

In this section, we briefly review the Thin-Plate Spline Robust Point Matching (TPS-RPM) algorithm [8], which we will use to develop the proposed groupwise association algorithm. TPS-RPM can robustly match two sets of 2D points by exploring the correlations among these points. Specifically, let $U = \{\mathbf{u}_p\}_{p=1}^N$ and $V = \{\mathbf{v}_q\}_{q=1}^M$ be two sets of 2D points, i.e., $\mathbf{u}_p = (u_{px}, u_{py})$, $p = 1, 2, \cdots, N$ and $\mathbf{v}_q = (v_{qx}, v_{qy})$, $q = 1, 2, \cdots, M$. The matching between these two sets of points is represented by a matrix $\mathcal{H} = [h_{p,q}]_{N\times M}$, where $h_{p,q} = [0, 1]$ indicates the probability of matching $\mathbf{u}_p$ and $\mathbf{v}_q$. TPS-RPM can jointly determine a non-rigid 2D transform $\mathbf{f} = (f_x, f_y) : \mathbb{R}^2 \to \mathbb{R}^2$ and the matrix $\mathcal{H}$ to minimize a cost function

$$E_{TPS-RPM}(\mathcal{H}, \mathbf{f}) = \sum_{p=1}^{N}\sum_{q=1}^{M} h_{pq} \parallel \mathbf{f}(\mathbf{u}_p) - \mathbf{v}_q \parallel^2 + \\ + \alpha\phi(\mathbf{f}) + \beta \sum_{p=1}^{N}\sum_{q=1}^{M} h_{pq} \log h_{pq} - \gamma \sum_{p=1}^{N}\sum_{q=1}^{M} h_{pq}, \quad (1)$$

where $\phi(\mathbf{f}) = \iint [L(f_x) + L(f_y)]\, dxdy$ is the TPS bending energy [3, 8] with $L(\cdot) = \left(\frac{\partial^2}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2}{\partial y^2}\right)^2$ and it reflects the smoothness of the 2D mapping $\mathbf{f}$ – the smaller the $\phi(\mathbf{f})$, the smoother the mapping $\mathbf{f}$. The cost function is alternately minimized in terms of $\mathcal{H}$ and $\mathbf{f}$ until convergence. Finally the obtained matrix $\mathcal{H}$ is thresholded to build the point matching between U and V. By introducing the last two terms in the cost function, TPS-RPM can handle the noise and identify points without matchings.

3.2. Groupwise TPS Association - Initialization

When Kalman tracking moves into a new slice i, we have a set of N fiber predictions $\left\{\hat{\mathbf{s}}_p^i\right\}_{p=1}^N$ derived from the previous slices and a set of M fiber observations $\left\{\mathbf{o}_q^i\right\}_{q=1}^M$ detected on the new slice. For simplicity, we drop the superscript i and denote the predictions and observations as $\{\hat{\mathbf{s}}_p\}_{p=1}^N$ and $\{\mathbf{o}_q\}_{q=1}^M$, respectively, when it does not introduce ambiguity.

Fibers are usually implanted in bundles. On one hand, the fibers in different bundles are not correlated and the association of fibers in different bundles cannot be well modeled by a single TPS transform. On the other hand, for the fibers in the same bundle, they show good proximity and parallelism in 3D space and therefore, a smooth TPS transform, such as the one computed using TPS-RPM, may well describe such a within-bundle fiber association, especially in sparsely sampled image sequences. The problem is that the bundle compositions are unknown priorly. In this section, we develop a new approach that can simultaneously explore the fiber-bundle composition and the fiber association.

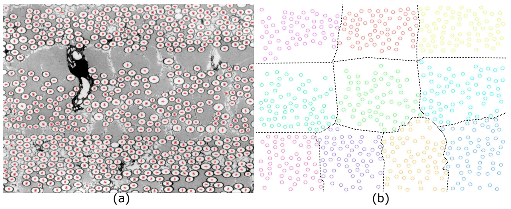

Figure 2. An illustration of clustering predictions into a set of compact groups. (a) Predictions (in red) and (b) clustered groups.

Without knowing the bundle composition, we first cluster all the predictions into smaller groups, as shown in Fig. 2

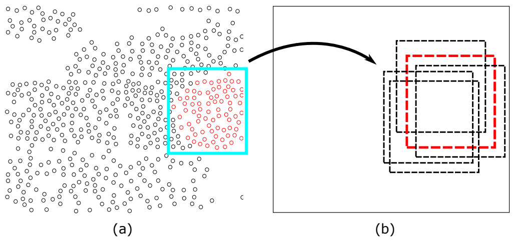

Figure 3. An illustration of finding the initial matching. (a) One group of predictions (in red) and (b) Sliding windows across the slice with observations: the optimal window with the best matching is shown in red.

by using the K-means algorithm. While each fiber prediction $\hat{\mathbf{s}}_p$ is a 4D vector made up of a 2D location and a 2D velocity, we only cluster in terms of the locations $\{\hat{\mathbf{z}}_p = (\hat{x}_p, \hat{y}_p)\}_{p=1}^N$. However, the observations are not divided into corresponding groups as for the predictions. We use a sliding-window strategy to address this issue. As shown in Fig. 3, for each group of predictions, shown by red circles, we derive its bounding box (in blue). Then we apply the sliding window of the size of the bounding box, dilated by 5 pixels, on the slice with the observations and perform TPS-RPM between the group of prediction against the observations in each of the sliding window. The sliding window that leads to the minimum cost $E_{TPS-RPM}$ is taken as the optimal window, shown by the red box in Fig. 3, and the matched observations in this window are taken as the matching to the considered group of predictions. Similarly, we only consider 2D location $\hat{\mathbf{z}}_p$ when applying the TPS-RPM for matching.

After applying such a sliding-window based matching for each group of predictions, we find its corresponding matched group of the observations and construct an initial association between the predictions and the observations. However, K-means clustering cannot guarantee the predictions from a same group are all from a same bundle and the initial association from TPS-RPM in such a group may not be reliable. In the following, we propose an algorithm to refine this initial association result.

3.3. Groupwise TPS Association - Refinement

We develop a three-step algorithm, including *group shrinking*, *group growing* and *group merging,* for refining the initial groupwise association.

Group shrinking further removes the outlier matchings in each group. Without loss of generality, let $(\hat{\mathbf{s}}_p, \mathbf{o}_p)$, $p = 1, 2, \cdots, m$ be one matched group of predictions and observations, as shown in Fig. 4(a). To identify and remove the outlier matching pairs from this group, we calculate the TPS bending energy as in Eq. (1) for this matching, in a

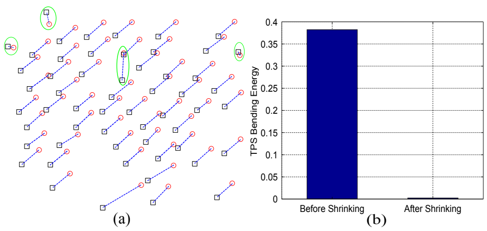

Figure 4. An illustration of the group shrinking. (a) Initial matching of one group of the predictions (black boxes) and observations (red circles), with matching pairs linked by dashed lines. Outlier matchings that are removed in group shrinking are highlighted in green vertical ellipses. (b) TPS bending energies before and after removing the outlier matchings.

matrix form [6]

$$\phi\left((\hat{\mathbf{s}}_p, \mathbf{o}_p)\,|p = 1, 2, \cdots, m\right) = \frac{1}{8\pi}\left(\mathbf{x_o}^T\mathbf{L}\mathbf{x_o} + \mathbf{y_o}^T\mathbf{L}\mathbf{y_o}\right), \tag{2}$$

where $\mathbf{L}$ is the $m \times m$ upper-left block of the matrix $\begin{pmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix}^{-1}$, $\mathbf{K}$ is the $m \times m$ TPS kernel matrix with element $k_{pq} = k(\hat{\mathbf{z}}_p, \hat{\mathbf{z}}_q) = \|\hat{\mathbf{z}}_p - \hat{\mathbf{z}}_q\|^2 \log \|\hat{\mathbf{z}}_p - \hat{\mathbf{z}}_q\|$, $\mathbf{P} = (\mathbf{1}, \hat{\mathbf{x}}, \hat{\mathbf{y}})$ with $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ being the concatenated vectors for all the x and y coordinates of the predictions $\{\hat{\mathbf{s}}_p\}_{p=1}^m$ (i.e., $\{\hat{\mathbf{z}}_p\}_{p=1}^m$) respectively, and $\mathbf{x_o}$ and $\mathbf{y_o}$ are the concatenated vectors for all the x and y coordinates of the observations $\{\mathbf{o}_p\}_{p=1}^m$, respectively. We calculate the leave-one-pair-out TPS bending energy $\phi_j = \phi\left((\hat{\mathbf{s}}_p, \mathbf{o}_p)\,|p = 1, 2, \cdots, m; p \neq j\right)$, $j = 1, 2, \cdots, m$ and remove the j^*-th pair that leads to the largest decrease of bending energy, i.e., $j^* = \arg\min_j \phi_j$. We repeat this process until a specified percentage (δ) of pairs are removed from each group, as shown in Fig. 4. Note that, by prespecifying the percentage δ, the step of group shrinking may remove true positive matchings from a group. This will be handled in the later steps of group growing and group merging.

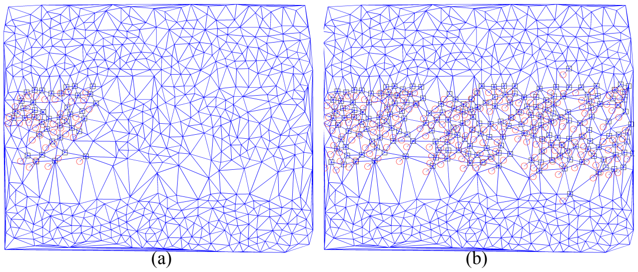

Figure 5. An illustration of the group growing. (a) Delaunay triangulation of the predictions and a matching group before group growing. (b) The same matching group after the group growing. Matched pairs in the group are shown by the dashed-line linked boxes (predictions) an circles (observations).

After the group shrinking, we can construct a TPS mapping for the matching in each group. Without loss of generality, let $(\hat{\mathbf{s}}_q, \mathbf{o}_q)$, $q = 1, 2, \cdots, n$ be one matched group of predictions and observations after group shrinking. We derive the TPS transform $\mathbf{f} = (f_x, f_y)$ such that $\mathbf{o}_q$= $\mathbf{f}(\hat{\mathbf{z}}_q)$, $q = 1, 2, \cdots, n$. This transform is in the form of [6]

$$\begin{cases} f_x(\hat{\mathbf{z}}) = & a_1 + a_2\hat{x} + a_3\hat{y} + \sum_{q=1}^{n} c_q k(\hat{\mathbf{z}}, \hat{\mathbf{z}}_q) \\ f_y(\hat{\mathbf{z}}) = & b_1 + b_2\hat{x} + b_3\hat{y} + \sum_{q=1}^{n} d_q k(\hat{\mathbf{z}}, \hat{\mathbf{z}}_q), \end{cases} \quad (3)$$

where $\hat{\mathbf{z}} = (\hat{x}, \hat{y})^T$ is any location in the domain of prediction and $\hat{\mathbf{z}}_q$ is the location of the prediction $\hat{\mathbf{s}}_q$. The parameters $\mathbf{a} = (a_1, a_2, a_3)^T$, $\mathbf{b} = (b_1, b_2, b_3)^T$, $\mathbf{c} = (c_1, c_2, \cdots, c_n)^T$ and $\mathbf{d} = (d_1, d_2, \cdots, d_n)^T$ can be computed by

$$\begin{pmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c} & \mathbf{d} \\ \mathbf{a} & \mathbf{b} \end{pmatrix} = \begin{pmatrix} \mathbf{x_o} & \mathbf{y_o} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

where $\mathbf{K}$, $\mathbf{P}$, $\mathbf{x_o}$, $\mathbf{y_o}$ and $k(\cdot, \cdot)$ are defined as in Eq. (2), but using the n matching pairs after the group shrinking.

In *group growing*, we first construct a Delaunay triangulation [17] by taking all the predictions as the vertices, as shown in Fig. 5(a). Group growing is performed for each matching group (after the group shrinking) independently. Without loss of generality, let's consider a matching group $(\hat{\mathbf{s}}_q, \mathbf{o}_q)$, $q = 1, 2, \cdots, n$ shown by the dashed-line linked boxes (predictions) and circles (observations) in Fig. 5(a). The iterative growing of this group takes the following steps:

1. Label the predictions $\hat{\mathbf{s}}_q$, $q = 1, 2, \cdots, n$ as "processed" and all the other predictions as "unprocessed". Compute the TPS transform $\mathbf{f}$ using Eq. (3).

2. From all the "unprocessed" predictions that are adjacent to the matching group in the Delaunay triangulation graph, identify the nearest one and denote it as $\hat{\mathbf{s}}_a$ with the location $\hat{\mathbf{z}}_a$. If all predictions adjacent to the matching group have been "processed", exit and return the matching group as the group growing result.

3. Apply TPS transform $\mathbf{f}$ to $\hat{\mathbf{z}}_a$ and search for the observation $\mathbf{o}_a$ that is nearest to $\mathbf{f}(\hat{\mathbf{z}}_a)$.

4. Check the consistency of the pair $(\hat{\mathbf{s}}_a, \mathbf{o}_a)$ against the matching group. If the consistency conditions are satisfied, we update the matching group by including the pair $(\hat{\mathbf{s}}_a, \mathbf{o}_a)$. Relabel the predictions in the updated matching group as "processed" and all the other predictions as "unprocessed". Recalculate the TPS transform $\mathbf{f}$ using the updated matching group and go back to Step 2. If any consistency condition is not satisfied, simply label $\hat{\mathbf{s}}_a$ as "processed" and go back to Step 2.

Figure 5(b) shows a group-growing result, starting from the matching group given in Fig. 5(a).

In this paper, we define two consistency conditions between a pair $(\hat{\mathbf{s}}_a, \mathbf{o}_a)$ and a matching group $(\hat{\mathbf{s}}_q, \mathbf{o}_q)$, $q = 1, 2, \cdots, n$. First, we compute the distribution of the prediction-observation gap $(\mathbf{o}_q - \hat{\mathbf{z}}_q)$, $q = 1, 2, \cdots, n$ in the matching group and examine whether the gap $(\mathbf{o}_a - \hat{\mathbf{z}}_a)$ shows high likelihood in this distribution. More specifically, the gap is a 2D vector and we estimate two Gaussian distributions for the magnitude and slope angle, respectively. The first consistency condition is that the gap $(\mathbf{o}_a - \hat{\mathbf{z}}_a)$ falls in L_T times the standard deviations in both magnitude and slope angle distributions. Second, adding a new pair to a matching group may increase the TPS bending energy for the matching group. A small bending-energy increase, i.e., $\phi\left((\hat{\mathbf{s}}_q, \mathbf{o}_q)\,|q = a, 1, 2, \cdots, n\right) - \phi\left((\hat{\mathbf{s}}_q, \mathbf{o}_q)\,|q = 1, 2, \cdots, n\right) \leq \Delta_\phi$, is taken as the other consistency condition. If both consistency conditions are satisfied, we update the matching group by including the new pair as stated in Step 4. Note that Gaussian distributions used in consistency conditions are also updated when the matching group is updated in the group growing.

After applying the group growing independently to all the matching groups, one prediction may be matched to different observations in different matching groups and vice versa. We perform a *group merging* for the final association by applying two rounds of majority voting. In the first round, for each prediction $\hat{\mathbf{s}}_p$, the number of the votes an observation $\mathbf{o}_q$ receives is the number of matching groups that contain the pair $(\hat{\mathbf{s}}_p, \mathbf{o}_q)$ after group growing. The observation with the largest number of votes is matched to $\hat{\mathbf{s}}_p$. In the second round, for each observation $\mathbf{o}_q$ we vote similarly for its corresponding prediction by only considering the matching pairs that are kept after the first round of voting. After these two rounds of voting, the resulting matching pairs are guaranteed to be one-on-one: No two observations are matched to a same prediction and vice versa. We take these final matching pairs as the final association.

The whole groupwise TPS association algorithm is summarized in Algorithm 1.

Algorithm 1 Groupwise TPS association algorithm.

```
Input $\{\hat{\mathbf{s}}_p^i\}_{p=1}^{N}$: $N$ fiber predictions on slice $i$
      $\{\mathbf{o}_q^i\}_{q=1}^{M}$: $M$ fiber observations on slice $i$
Divide predictions to groups using $K$-means
FOR each group
    TPS-RPM for initial association
    Group shrinking to remove outlier matchings
    Group growing to include consistent matching pairs
END FOR
Group merging for final association
```

4. Experiments

In the experiments, we apply the proposed method to track large-scale fibers from S200, an amorphous SiNC matrix reinforced by continuous Nicalon fibers. Three sets of data, denoted as Data 1, Data 2 and Data 3, are collected, each of which is a 100-slice image sequence with dense inter-slice distance $1\mu m$. The size of each slice is 1292×968. A sample slice is shown in Fig. 2(a), which contains hundreds of crowded fibers. On the collected data, we annotate the locations of fibers on each slice and link them across slices as the ground truth for performance evaluation.

We use five widely used metrics [15, 20] for evaluating the fiber tracking performance: Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Identity Switches (IDSW), Mostly Tracked (MT) and Mostly Lost (ML), all of which measure the co-alignment between the tracked fibers and the annotated ground-truth fibers. Among these metrics, MOTA is a comprehensive one that considers both IDSW, false positives and false negatives. IDSW, MT and ML only reflect the tracking performance from specific perspectives. In computing these metrics, we use a threshold of 20 pixels between the tracked fiber and the ground-truth fiber on each slice to count the hit/miss on the slice. MT is the number of ground-truth fibers that are hit in no less than 80% of slices while ML is the number of ground-truth fibers that are hit in no more than 20% of slices. For MOTA and MT, the higher the better, and for MOTP, IDSW and ML, the lower the better. Note that, we follow the distance-based MOTP definition [15] and a lower MOTP indicates a better tracking.

To test the tracking performance under sparsely sampled image sequences, we downsample the original image sequence. In particular, we skip $C \geq 0$ slices before taking the next slice in the original sequence, until the end of original sequence is reached, to construct such sparsely sampled image sequences. For convenience, we name parameter C the *sparsity*: The larger the parameter C, the lower the inter-slice continuity of the constructed image sequence. One issue is that, such constructed image sequences with large C are much shorter than the original image sequence and the tracking performance obtained on a single such short sequence may not be statistically reliable. To alleviate this issue, for a given sparsity C we construct $C + 1$ image sequences, starting from original slice $1, 2, \cdots, C+1$ respectively. These $C+1$ image sequences do not share any slice. We perform tracking on each of them independently and then average their performances, e.g., MOTA and MOTP, as the performance of tracking under the sparsity C. Note that when $C = 0$, tracking is directly performed and evaluated on a single image sequence: the original densely sampled image sequence. In our experiments, we continuously vary the sparsity C from 0 to 19 and examine the tracking performance under different sparsity.

To justify the effectiveness of the proposed method, we compare its performance against three baseline Kalman filter methods and four other state-of-the-art multi-target tracking methods. For the three baseline methods, *Kalman-NN, Kalman-Hung* and *Kalman-Global*, they all follow the same Kalman filter setting as in the proposed *Kalman-Groupwise* method. The difference lies in the step of association. *Kalman-NN* uses the nearest neighboring (NN) search for association. More specifically, the pair of the prediction and observation with the minimum distance is identified and associated. Then we exclude this identified pair and repeat the same NN search on the remaining predictions and observations, until either predictions or observations are empty. *Kalman-Hung* computes the association using Hungarian algorithm [16] for a minimum-total-distance bipartite matching. Because the number of predictions and observations are usually different, we introduce dummy nodes into Hungarian algorithm. The distance to a dummy node is set to 40 pixels in our experiments. *Kalman-Global* computes the association by directly applying TPS-RPM [8] to match the predictions and observations in a global fashion. The four other multi-target tracking algorithms used for comparison are DPNMS [22], SMOT [10], CEM [20] and KTH [18]. As reviewed in Section 2, these four are all non-recursive tracking methods.

For the Kalman-based methods, including the three baseline methods and the proposed Kalman-Groupwise, the initial state covariance is set to be a diagonal matrix with diagonal elements 10^3. The transition noise covariance is set to be a diagonal matrix with diagonal elements 10^{-3}. The observation noise covariance is set to be a diagonal matrix with diagonal elements 10^{-3}. In the proposed Kalman-Groupwise, predictions are always clustered to 10 groups and the percentage of removed fiber pairs in group shrinking is set to $\delta = 30\%$. The consistency thresholds in group growing are set to $L_T = 3$ and $\Delta_\phi = 0.01$. For the four non-recursive methods, we use the code downloaded from their authors' websites. For DPNMS, SMOT and CEM, the observations are the bounding boxes of the detected ellipses as described in Section 3. For KTH, no source code is available and we only have its binary executable file, which has its own integrated image segmentation and target detection components. For these four comparison methods, we use their default parameters for experiments.

4.1. Experiment Results on Fiber Tracking

The top row of Fig. 6 shows the MOTA and MOTP of the proposed Kalman-Groupwise method and the three baseline Kalman filter tracking methods on the three datasets (image sequences), under different sparsity C. We can see that, when the sparsity C is low, both the proposed

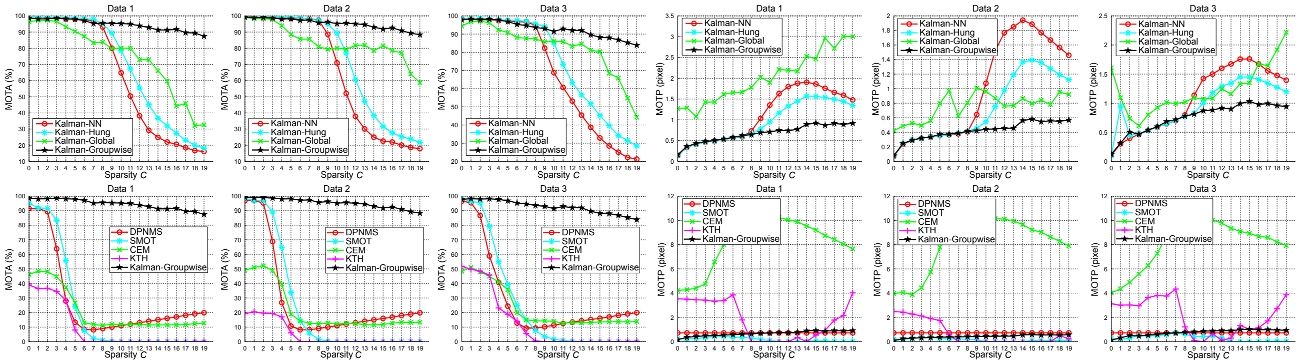

Figure 6. MOTA and MOTP performance of the proposed Kalman-Groupwise and the comparison methods, under different sparsity C.

Table 1. IDSW, MT and ML performance under different sparsity C. The performance is the average over all three image sequences.

Metrics		Kalman-NN	Kalman-Hung	Kalman-Global	DPNMS[22]	SMOT[10]	CEM[20]	KTH[18]	Kalman-Groupwise
IDSW	$C = 0$	9.0	6.3	4.5	780.5	62.8	63.0	258.3	4.3
	$C = 5$	6.3	3.3	41.0	5135.7	21.9	128.2	3.5	2.6
	$C = 10$	596.6	209.9	118.6	2921.7	0.02	89.7	0	5.0
	$C = 15$	1162.4	937.9	229.9	1871.0	0.7	118.2	0	21.3
	$C = 19$	1100.5	999.1	453.0	1413.3	0	129.6	0.1	43.4
MT	$C = 0$	376.3	377.0	366.0	370.5	363.0	113.8	84.0	376.3
	$C = 5$	371.1	374.9	345.5	368.6	68.1	38.1	16.1	373.9
	$C = 10$	309.2	337.6	313.0	360.1	0.7	2.3	0	364.6
	$C = 15$	268.5	271.9	301.8	360.6	0	3.6	0	354.6
	$C = 19$	307.5	297.5	280.2	362.0	0	5.2	0	347.5
ML	$C = 0$	0.3	0.3	4.0	0.8	1.0	115.5	140.8	0.3
	$C = 5$	1.6	1.8	8.5	0.8	214.4	166.0	324.6	2.7
	$C = 10$	3.9	5.4	7.7	0.8	372.7	198.6	375.0	6.2
	$C = 15$	8.8	11.2	11.4	1.6	374.3	231.0	375.0	12.4
	$C = 19$	1.5	1.7	3.5	0.8	373.5	117.7	370.6	5.5

method and the three baseline methods produce satisfactory fiber tracking, with very high MOTA and very low MOTP. With the increase of the sparsity, the performance of all these four methods drops. However, the proposed Kalman-Groupwise's performance, in terms of both MOTA and MOTP, drops much slower than the three baseline methods. Even if $C = 19$, i.e., increasing the inter-slice distance by 19 times, the proposed method can still achieve very high MOTA performance ($> 80\%$).

The bottom row of Fig. 6 shows the MOTA and MOTP of the proposed method and the four non-recursive tracking methods: DPNMS, SMOT, CEM and KTH. All the four comparison methods show low MOTA values when the sparsity increases, because the crowded targets with similar appearance and the low continuity between slices break some basic assumptions made in these methods. In terms of MOTP, the proposed Kalman-Groupwise, SMOT, and DPNMS are better (e.g., with lower MOTP values) than CEM and KTH.

Table 1 shows the IDSW, MT and ML metrics of the proposed Kalman-Groupwise and all the comparison methods. The performance shown in this table is the average over all three test image sequences. In general, the proposed Kalman-Groupwise shows competitive performance than these comparison methods when the sparsity C is high. In particular, compared to other comparison methods, the proposed Kalman-Groupwise is the only one that always keeps small IDSW, high MT and low ML when increasing sparsity C. DPNMS shows very high MT and low ML but suffers from very high IDSW. This indicates that DPNMS makes many mis-associations between frames.

For the running time, the proposed method processes 0.023 slices per second on a workstation with a 4-core 2.6GHz Intel CPU and 8GB memory. It can be accelerated substantially by parallelizing the matching on different fiber groups. We also study the selection of the cluster numbers in group initialization. We tried the group numbers from 8 to 16 on one fiber dataset with sparsity C=19. The obtained mean MOTA is 85.6% with standard deviation 5.8%, indicating that the proposed method is not very sensitive to the choice of cluster number.

4.2. Crowded Human Tracking

The proposed method can be used for tracking crowded people from videos. Analogue to fiber bundles, crowded people usually move in groups where people in the same group usually move toward similar directions with similar velocities and their association can be modeled by a TPS transform. While videos usually show good inter-frame continuity, there are several important cases where such inter-frame continuity may get very poor. First, the camera may move suddenly and then get back to look at the

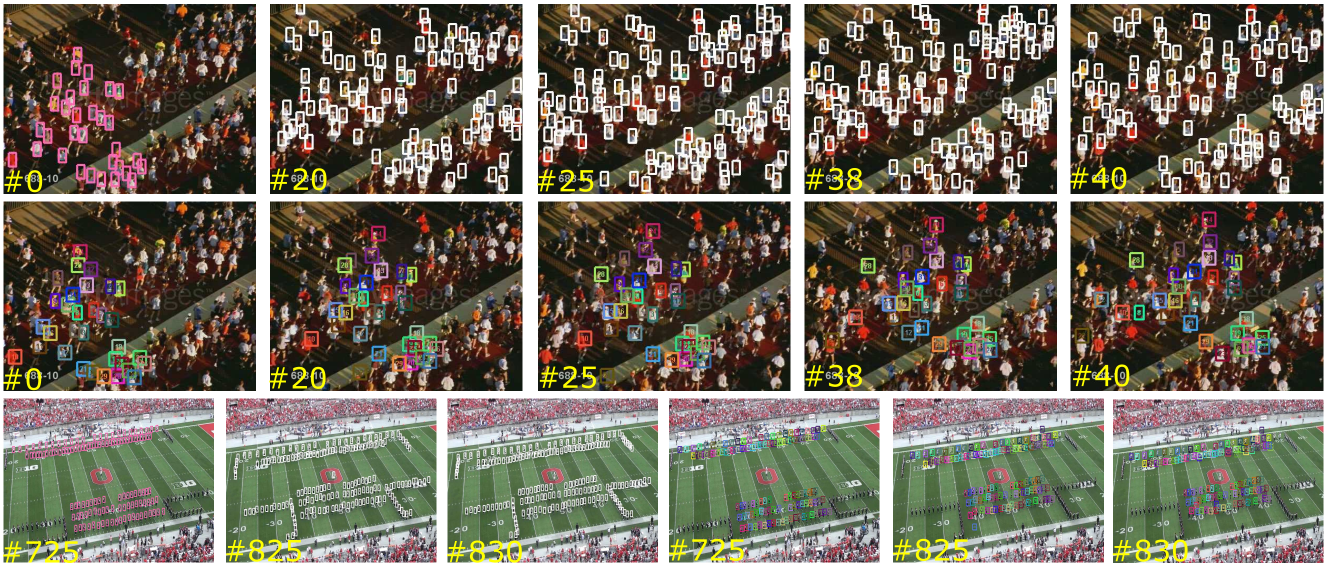

Figure 7. An illustration of crowded human tracking. Sparsely sampled Marathon video (five frames in row 1) and the tracking results (row 2) using the proposed method. Third row: Sparsely sampled Band video (left three frames) and the tracking results (right three frames) using the proposed method. People to be tracked are labeled in pink in the leftmost frame for both data. White boxes are observations and the boxes with an identifical color and number across frames represent a resulting track.

Table 2. MOTA performance (%) on crowded human tracking.

Video	Kalman-NN	Kalman-Hung	Kalman-Global	DPNMS[22]	SMOT[10]	CEM[20]	Kalman-Groupwise
Marathon	30.6	34.1	14.1	16.5	3.5	25.9	76.5
Band	14.8	20.4	7.4	16.7	1.9	9.3	74.1

same crowd of people. This is very common with the use of wearable cameras, such as Google Glass and GoPro. Second, the crowded targets may be occluded for a while and then re-appear in the view. In both cases, we need to track over low-continuity frames, analogue to the sparse sampled image sequences in fiber tracking.

We use two videos with crowded people, downloaded from internet, to test the proposed method: Marathon and Band, as shown in Fig. 7. Similar to fiber tracking, we simulate the low-continuity videos by sparse sampling: skipping a random number of frames (1 to 100) before taking a new frame for tracking. In Fig. 7, the frame numbers on the original video are shown at the bottom-left corner of the frame. Observations shown as white boxes are obtained by a trained DPM detector [11], followed by manual adjustments and corrections. After the sparse sampling, we choose 32 people to track on Marathon and 124 people to track for Band, as labeled by pink boxes on the starting frame as shown in Fig. 7. Their tracking results on subsequent frames are shown by numbered color boxes: the boxes with an identifical color and number across frames represent one resulting track.

Table 2 shows the MOTA tracking performance on these two sparsely sampled videos by using the proposed Kalman-Groupwise and other comparison methods. KTH is not applicable to this task because we only have its executable file which detects and tracks only objects with a simple shape of cells (e.g., fibers), but not human.

5. Conclusions

In this paper, we proposed a groupwise association algorithm for tracking similar-appearance, crowded targets along low-continuity image sequences. The proposed algorithm divides targets in groups and employs the nonrigid Thin-Plate Splines (TPS) to model the within-group association. Without knowing the group compositions priorly, we applied K-means clustering to initialize the groups and then developed a three-step algorithm, consisting of group shrinking, group growing and group merging, for refining the initial groups. By integrating this association algorithm into Kalman filter, we used it to track large-scale crowded fibers from sparsely sampled material image sequences. We also showed the application of the proposed method to track crowded people from low-continuity videos. Results showed that the proposed method outperforms several baseline Kalman filters and multi-target tracking methods.

6. Acknowledgment

This work was partially supported by AFOSR FA9550-11-1-0327 and UES, Inc./AFRL S-901-486-002. Source code and fiber data used in this paper can be downloaded from `http://cvl.cse.sc.edu/project/cvpr2016.html`.

References

[1] A. Alahi, V. Ramanathan, and L. Fei-Fei. Socially-aware large-scale crowd forecasting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2211–2218. IEEE, 2014. 2

[2] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *European Conference on Computer Vision*, pages 1–14. 2008. 2

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 3

[4] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–8, 2009. 2

[5] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Workshop on Motion and Video Computing*, pages 169–174, 2002. 2

[6] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):567–585, 1989. 4, 5

[7] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision*, pages 1515–1522, 2009. 2

[8] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003. 3, 6

[9] M. L. Comer and E. J. Delp. The em/mpm algorithm for segmentation of textured images: analysis and further experimental results. *IEEE Transactions on Image Processing*, 9(10):1731–1744, 2000. 2

[10] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *IEEE International Conference on Computer Vision*, pages 2304–2311, 2013. 2, 6, 7, 8

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 8

[12] J. F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *IEEE International Conference on Computer Vision*, pages 2470–2477, 2011. 2

[13] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 2

[14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960. 2

[15] B. Keni and S. Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008. 6

[16] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[17] D.-T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980. 5

[18] K. Magnusson, J. Jalden, P. Gilbert, and H. Blau. Global linking of cell tracks using the viterbi algorithm. *IEEE Transactions on Medical Imaging*, 2014. 1, 2, 6, 7

[19] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942, 2009. 2

[20] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):58–72, 2014. 2, 6, 7, 8

[21] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pages 28–39. 2004. 2

[22] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1201–1208, 2011. 2, 6, 7, 8

[23] C. Przybyla, T. Godar, J. Simmons, M. Jackson, L. Zawada, and J. Pearce. Statistical characterization of sic/sic ceramic matrix composites at the filament scale with bayesian segmentation hough transform feature extraction, and pair correlation statistics. In *International SAMPE Technical Conference*, pages 859–878, 2013. 1

[24] Z. Qin and C. R. Shelton. Improving multi-target tracking via social grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1972–1978, 2012. 2

[25] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. 2

[26] J. Shao, C. Loy, and X. Wang. Scene-independent group profiling in crowd. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2226, 2014. 2

[27] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multimodality through mixture tracking. In *IEEE International Conference on Computer Vision*, pages 1110–1116, 2003. 2

[28] Y. Xie and Q. Ji. A new efficient ellipse detection method. In *International Conference on Pattern Recognition*, pages 957–960, 2002. 2

[29] Y. Yuan, J. Fang, and Q. Wang. Online anomaly detection in crowd scenes via structure analysis. *IEEE Transactions on Cybernetics*, 45(3):562–575, 2015. 1

[30] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[31] S. Zhang, Y. Zhu, and A. K. Roy-Chowdhury. Tracking multiple interacting targets in a camera network. *Computer Vision and Image Understanding*, 134:64–73, 2015. 2

[32] B. Zhou, X. Tang, H. Zhang, and X. Wang. Measuring crowd collectiveness. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1586–1599, 2014. 2