Boston University Computer Science Technical Report No. BUCS-TR-2010-030*

Tracking-Reconstruction or Reconstruction-Tracking? Comparison of Two Multiple Hypothesis Tracking Approaches to Interpret 3D Object Motion from Several Camera Views

Zheng Wu[1], Nickolay I. Hristov[2], Thomas H. Kunz[3], and Margrit Betke[1]
[1] Department of Computer Science, Boston University
[2] Department of Life Sciences, Winston Salem State University, and Center for Design Innovation, University of North Carolina
[3] Department of Biology, Boston University

Abstract

We developed two methods for tracking multiple objects using several camera views. The methods use the Multiple Hypothesis Tracking (MHT) framework to solve both the across-view data association problem (i.e., finding object correspondences across several views) and the across-time data association problem (i.e., the assignment of current object measurements to previously established object tracks). The "tracking-reconstruction method" establishes two-dimensional (2D) objects tracks for each view and then reconstructs their three-dimensional (3D) motion trajectories. The "reconstruction-tracking method" assembles 2D object measurements from all views, reconstructs 3D object positions, and then matches these 3D positions to previously established 3D object tracks to compute 3D motion trajectories. For both methods, we propose techniques for pruning the number of association hypotheses and for gathering track fragments. We tested and compared the performance of our methods on thermal infrared video of bats using several performance measures. Our analysis of video sequences with different levels of densities of flying bats reveals that the reconstruction-tracking method produces fewer track fragments than the tracking-reconstruction method but creates more false positive 3D tracks.

1. Introduction

Multi-object tracking remains a difficult problem in computer vision because occlusion is prevalent in typical multi-object imaging scenarios. Ambiguity in data association (i.e., the process of matching currently measured objects with established object tracks) must be resolved. Disambiguating measurement-to-track associations for all objects in a scene may not be possible within one time step, especially if the objects have similar appearance. However, popular "sequential tracking methods" (e.g., the Joint Probabilistic Data Association (JPDA) method [1]) must, in one time step, process the set of candidate assignments and decide on the most likely measurement-to-track associations. If the requirement for such sequential, time-step-by-time-step decisions can be relaxed, the likelihood of candidate associations typically can be estimated more accurately. Uncertainties in the current time step may be resolved when evidence for or against a hypothesized association has been collected in subsequent frames. This approach is called "look-ahead" or "deferred-logic tracking," and the classic method is Multiple Hypothesis Tracking (MHT) [16].

The MHT method builds a tree of possible measurement-to-track associations, evaluates the

*This report appeared in the proceedings of the *IEEE Workshop on Motion and Video Computing (WMVC)*, Snowbird, Utah, December 2009. The material is based upon work supported by the National Science Foundation under Grant Nos. 0326483 and 0910908.

probability of each candidate association, and solves the NP-hard problem of finding the association with the highest probability by explicit enumeration or combinatorial optimization. The MHT method becomes impractical when the number of objects in the scene is large. Thus, techniques for pruning the number of association hypotheses have been used [16, 4]. An important technique for pruning the hypotheses tree to a fixed depth T is to use a sliding time-window of duration T during which hypotheses can be resolved. In this paper, we propose two approaches for tracking multiple objects from several camera views that use the MHT framework with the sliding-window pruning technique.

Tracking multiple objects in several camera views is challenging because data association must be performed not only across time, as in single-view tracking, but also across views. Two strategies can be used to solve the multi-view multi-object tracking task that differ in the order of the association processes: (1) The "tracking-reconstruction method" processes the across-time associations first and establishes 2D objects tracks for each view. It then reconstructs 3D motion trajectories. (2) The "reconstruction-tracking method" processes the across-view associations first by reconstructing the 3D positions of candidate measurements. It then matches the 3D positions to previously established 3D object tracks.

The tracking-reconstruction method can be interpreted as a track-to-track fusion process that benefits from deferring assignment decisions, as in Multiple Hypothesis Tracking. When, over time, information about the 2D track is accumulated, the ambiguity in matching tracks across views becomes smaller. The method is suitable when a distributed system architecture is required to prevent "one-point-failures" (which may occur in a centralized system used by the reconstruction-tracking method). The reconstruction-tracking method can be seen as a feature-to-feature fusion process, where the features are 3D object positions processed from 2D image measurements. The reconstruction-tracking method is often implemented without a deferred-logic tracking approach, so that decisions are made sequentially. This is advantageous because sequential approaches are conceptually easier and computationally less expensive. Existing work on human tracking from multiple camera views have compared the two schemes [18, 10] and have generally favored the reconstruction-tracking scheme [11, 7, 18]. To the best of our knowledge, the computer vision literature on multi-view tracking does not include analyses that compare the two schemes for imaging scenarios with dense groups of objects.

In this paper, we propose two tracking methods that use the reconstruction-tracking and tracking-reconstruction approaches, respectively. We show that each method has its advantages and disadvantages, especially in imaging scenarios where objects look similar and are thus difficult to distinguish. We focus on thermal infrared video recordings and address scenarios where dozens of objects appear simultaneously in the field of view of three cameras and are imaged at low spatial resolution. In particular, we tested and compared the performance of our two tracking methods on a thermal infrared video of a large group of bats flying out of a cave. Our data set is challenging because bats appear similar, move extremely fast, and do not fly in straight lines, but may choose any heading direction within 3D space. Because the 3D movement directions of bats are more general than those of people, we cannot take advantage of the constraint, which is often used in computer vision research and works well for tracking people's heads [7] or feet [9], that the image of the ground plane of the scene in each camera view is related by a homography.

For this paper, we build upon our previous work on multi-view tracking [19], which is based on a sequential approach. Here instead we apply deferred-logic tracking to both the reconstruction-tracking and tracking-reconstruction approaches. We use the MHT Multidimensional Assignment Formulation by Poore [15] to process track initiation, maintenance, and termination. For our tracking-reconstruction method, we propose a greedy matching procedure with a spatial-temporal constraint for track-to-track fusion. We relax the constraint of one-to-one correspondence across views because of potential long-term occlusion in a single view. For our reconstruction-tracking method we propose a heuristic approach to reduce "phantom" effects caused by false positive 3D trajectories.

Existing work that aims at improving tracking performance for single-view scenarios [14, 12] is helpful for both reconstruction-tracking and tracking-reconstruction approaches. However, a comparison of multi-view and single-view tracking approaches is beyond the scope of this paper.

2. Related Work

Research on multi-object tracking has a long history in computer vision. The first systems analyzed video data collected using a single camera. In recent years, imaging systems that use several cameras have become attractive because they can provide an analysis of 3D object trajectories and stereoscopic reasoning for assessing occlusion. Because the 3D positions of the objects cannot be measured directly but need to be inferred from 2D measurements, tracking-reconstruction and reconstruction-tracking approaches have been developed, which track the objects in 2D or 3D.

Existing systems [6, 7, 10, 11, 13, 18] use the sequential reconstruction-tracking scheme. Tracking is performed in 3D [7, 11, 13, 18], using reconstructed 3D object features, or in 2D [6, 10], using the 2D projections of reconstructed 3D points into the image plane of each camera. The former approach, tracking in 3D, is a reasonable choice if the 3D positions of objects or object features can be predicted accurately. If the information about an object gathered from several rather than only two camera views is fused and the cameras are spatially calibrated, the 3D position estimates can typically be made quite accurately. Obtaining *accurate* position estimates, however, is not the main challenge of multi-object video analysis; instead, the main challenge is the correct interpretation of *ambiguous* position estimates, which are caused by occlusion. Experiments reported for existing systems involved tracking a single object [18] or a few objects (less than 5) [6, 10], and it is not clear how across-time ambiguity in 2D or across-view ambiguity in 3D would affect the tracking performance of these methods in dense object scenarios.

The differences between previously published work on 3D tracking from several camera views and our two tracking approaches are:

- Our methods are explicitly designed to track dense groups of objects.
- We use a deferred-logic (not a sequential) tracking framework. Our MHT approach allows us to evaluate the information obtained from several cameras during a window in time.
- Our approach does not assume that object motion is restricted to occur on a ground plane.
- Our approach fuses information about object position only and does not attempt to fuse additional information about object appearance.
- Our test data include imaging scenarios where many objects appear in the scene with low resolution at the same time (groups of 10 to 30 individuals).

Because deferred-logic approaches, by definition, have access to more information than sequential approaches, a comparison of the performance between our method and any sequential approach described in the literature would not be meaningful. Thus, the performance analysis in our paper focuses on the comparison between the two methods we propose and further discusses their advantages and disadvantages.

3. Two 3D Multiple Hypothesis Tracking Methods

We use the Multiple-Hypothesis-Tracking (MHT) method as the tracking framework for both tracking-reconstruction (TR) and reconstruction-tracking (RT) approaches. The difference is TR uses 2D measurements as input for MHT to generate 2D trajectory, while RT uses 3D reconstructed points as input for MHT to generate 3D trajectory. We first revisit the MHT method in its multidimensional assignment formulation [15] with a fixed-duration, sliding time-window (Sec. 3.1). We then describe our reconstruction-tracking method (Sec. 3.2) and our tracking-reconstruction method (Sec. 3.3).

3.1. Revisit of Multiple Hypothesis Tracking

The Multiple-Hypothesis-Tracking method is generally posed as the problem of maximizing the aposteriori probability of measurement-to-track associations, where the current time step is T, the set of measurements at time step k is $Z(k)$, for $k = 1, ..., T$, and the number of measurements at time step k is $M_k = |Z(k)|$. The data association problem can then be formulated as the problem of finding a partition of the measurement set $Z = (Z(1), ..., Z(T))$ into a track set

$\mathcal{T}$ that maximizes the aposteriori probability $p(\mathcal{T}|Z)$ of measurement-to-track associations:

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} p(\mathcal{T}|Z) \quad (1)$$

$$= \arg\max_{\{\mathcal{T}_n\}} \prod_n P(Z_{\mathcal{T}_n}|\mathcal{T}_n)p(\mathcal{T}_n), \quad (2)$$

where $Z_{\mathcal{T}_n}$ is the sequence of measurements assigned to track $\mathcal{T}_n$ and the variable $\mathcal{T}_0$ represents the false positive tracks. The explicit assumption of this partitioning formulation is that the tracks do not overlap, i.e., measurements are assigned to one and only one track. In the case of multi-camera 3D tracking, the assumption holds because the measurements are reconstructed 3D object positions which must be disjoint. In the case of 2D tracking, 3D object trajectories projected onto the image plane typically have overlapping 2D measurements due to occlusions. For a trajectory of an object that is occluded at some point in time, the partitioning formulation would yield a pair of 2D tracks: a track that ends when the object is occluded and a new track that starts when the object is in view again. The challenge is then to automatically interpret that the two tracks successively describe the movement trajectory of the same object.

The MHT formulation requires a model for the prior probabilities of false positive and missed detections. For the imaging scenarios we address, we can assume a perfect detection rate and that the false positive detections are uniformly distributed in the field of view. We also assume a uniform distribution of the prior probability $p(\mathcal{T}_n)$. The likelihood of the nth track can then be written as

$$P(Z_{\mathcal{T}_n}|\mathcal{T}_n) = \prod_{k=1}^{T} p(z^k_{i_k} \mid \hat{x}^k_n)p(\hat{x}^k_n \mid \hat{x}^{k-1}_n), \quad (3)$$

where measurement $z^k_{i_k} \in Z(k)$ for $k = 1, ..., T$; $\hat{x}^k_n$ is object state and can be estimated using Kalman smoothing, i.e, estimate $\hat{x}^k_n$ given $(z^1_{i_1}, z^2_{i_2}, ...z^T_{i_T})$ with a series of forward and backward recursions [3].

We use binary variable $b_{i_1 i_2 ... i_T}$ to indicate whether measurement sequence $(z^1_{i_1}, z^2_{i_2}, ...z^T_{i_T})$ forms a potential track or not. It can be shown that the MHT problem formulation in Eq. 1, estimating the probability of a set of measurement-to-track assignments, can be stated as the following multidimensional assignment problem:

$$c = \min \sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} ... \sum_{i_T=1}^{M_T} c_{i_1 i_2 ... i_T}\, b_{i_1 i_2 ... i_T} \quad (4)$$

$$\text{s. t.} \sum_{i_2=1}^{M_2} \sum_{i_3=1}^{M_3} ... \sum_{i_T=1}^{M_T} b_{i_1 i_2 ... i_T} = 1; \quad i_1 = 1, 2, ..., M_1$$

$$\sum_{i_1=1}^{M_1} \sum_{i_3=1}^{M_3} ... \sum_{i_T=1}^{M_T} b_{i_1 i_2 ... i_T} = 1; \quad i_2 = 1, 2, ..., M_2$$

$$\vdots$$

$$\sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} ... \sum_{i_{T-1}=1}^{M_{T-1}} b_{i_1 i_2 ... i_T} = 1; \quad i_T = 1, 2, ..., M_T.$$

We compute the cost of an assignment by estimating the negative log likelihood of a measurement sequence: $c_{i_1 i_2 ... i_T} = -\log P(Z_{\mathcal{T}_n}|\mathcal{T}_n)$.

The multidimensional assignment problem in Eq. 4 is NP-hard for $T > 2$. We use the Greedy Randomized Adaptive Local Search Procedure (GRASP) [17] to efficiently obtain a suboptimal solution. Before applying GRASP, we use the following pruning techniques:

- Gating: Each established track maintains its own validation region or gate so that only measurements that fall within this gate need to be considered.
- Clustering: Tracks that do not compete for measurements form a cluster (i.e., a measurement in one cluster is not located in any validation region of a track in another cluster). Combinatorial optimization is applied within each cluster independently.

In the pseudocode below, we describe our proposed variant of the original MHT method, which automatically considers track initiation, maintenance and termination.

3.2. Reconstruction-Tracking Method

Our Reconstruction-Tracking Method first reconstructs the 3D positions of objects in the current scene and then applies 3D tracking to predict the next 3D object positions. Two approaches are used to perform the reconstruction step and find the

MULTIPLE HYPOTHESIS TRACKING WITH WIDTH-T SLIDING TIME WINDOW:
Input: Set Z of measurements from time t_0 to $t_0 + T$ and set $\mathcal{T}$ of tracks maintained up to time $t_0 + T - 1$

1. Remove from set Z the measurements that have been assigned to the tracks in set $\mathcal{T}$.
2. Build the multiple hypotheses tree with gating; assign measurements recorded at time T to tracks in $\mathcal{T}$; form additional candidate tracks with the measurements that remained in Z.
3. Cluster the hypotheses into disjoint trees and formulate a multidimensional assignment problem for each cluster.
4. Solve each problem using GRASP and return a set $\mathcal{T}^*$ of tracks.
5. Classify the tracks in sets $\mathcal{T}^*$ and $\mathcal{T}$:
   - If a track $\mathcal{T}_n$ in $\mathcal{T}^*$ is an extension of some track in $\mathcal{T}$, then it is interpreted as a *continuing track*.
   - If a track $\mathcal{T}_n$ in $\mathcal{T}^*$ has no overlap with any track in $\mathcal{T}$, then it is interpreted as a *new track* initiated at time t_0.
   - The remaining tracks in $\mathcal{T}$ are considered *terminated tracks* that end at time $t_0 + T - 1$;

Output: New, continuing, and terminated tracks up to time $t_0 + T$

across-view associations (i.e., the measurements from different views that describe the same object). The first approach [5] minimizes a linear combination of costs for all possible associations with the one-to-one match constraint that a measurement in one view can only be matched with exactly one measurement in another view. The optimization problem can be formulated as in Eq. 4 and yields solutions that do not correctly interpret occlusions and clutter when the one-to-one match constraint is violated as in the imaging scenario shown in Fig. 1. The alternative method is to perform triangulation [1] for every possible match without considering the one-to-one constraint. In our method, we apply gating and only consider matches whose *"reconstruction residual"* is below a given threshold. We compute the reconstruction residual as the root mean squared distance between the 2D measurements and the 2D projections in all camera views of the reconstructed 3D point. The pseudo code for the reconstruction step of our RECONSTRUCTION-TRACKING METHOD is given below.

RECONSTRUCTION-TRACKING METHOD
Reconstruction Step at Time t
Input: Set $\mathcal{T}$ of currently maintained 3D tracks and set $\{z_{s,i_s}\}_{s \in S}, i_s = 1, .., n_s$, of 2D point measurements from S views, where n_s is the number of measurements in view s.

1. For each track $\mathcal{T}_i \in \mathcal{T}$, compute its validation region $V_{s,i}$ in each view.
2. For each 2D point z_{s,i_s}, compute its epipolar lines in the other views.
3. Create candidate tuple $(z_{1,i_1}, z_{2,i_2}, ..., z_{S,i_S})$, where $i_1 \in \{1, .., n_1\}, .., i_S \in \{1, .., n_S\}$, by selecting one 2D point measurement z_{s,i_s} from each view such that:
   - Each z_{s,i_s} is located within its respective gate.
   - Each z_{s,i_s} is located near its epipolar line.
   - The reconstruction residual is below threshold ρ.
4. For the remaining unassigned 2D points, create a candidate tuple if they are located near their respective epipolar lines and if their reconstruction residual is below threshold ρ.

Output: Set Z of 3D points, reconstructed from candidate tuples, that will be interpreted as the set of input measurements for the tracking step of the reconstruction-tracking method.

Our RECONSTRUCTION-TRACKING METHOD generally computes 3D points that correctly correspond to 3D object positions. Due to false across-view associations during the reconstruction step, this method may also generate "phantom" points that do not correspond to any real objects in the scene. As we will demonstrate in Sec. 4, many of these phantom points can be eliminated during the 3D tracking step. Phantoms reported in the multi-object tracking literature (e.g. [7]) are typically considered an accidental occurrence. In the analysis of our infrared video data,

[1]We selected the Direct Linear Transformation (DLT) algorithm [8] to perform the triangulation because of its efficiency and sufficient accuracy. Other methods may replace DLT in our framework.

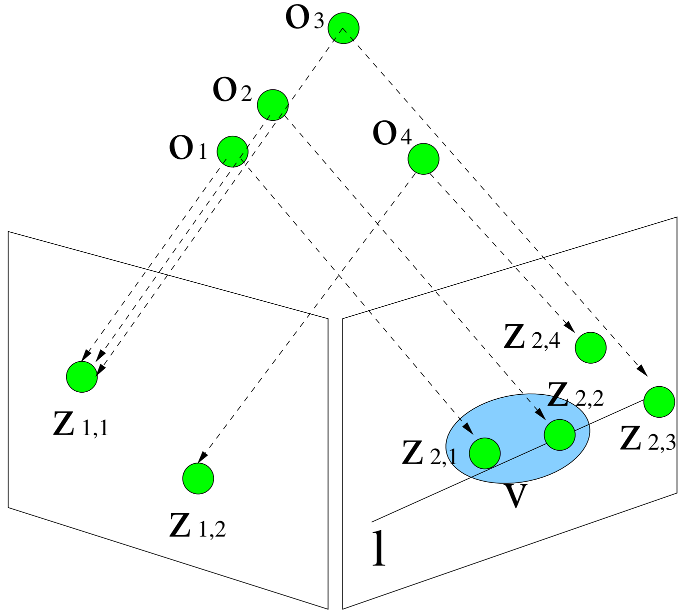

Figure 1. Across-view association without one-to-one matches. The single 2D measurement $z_{1,1}$ in the left view represents the overlapping projections of three objects o_1, o_2, and o_3 in the scene. Line l is the epipolar line in the right view that corresponds to the projection of all possible 3D objects that could be imaged as $z_{1,1}$ in the left view. Validation region V is the projected gate of the 3D track of object o_1 in the right view. In the across-view assignment process, $z_{1,1}$ may only be matched to measurements in the right view that are near line l and inside region V. These are $z_{2,1}$ and $z_{2,2}$, but not $z_{2,3}$ and $z_{2,4}$.

in which a very large number of similar objects are imaged at low spatial resolution, false across-view associations, however, are inevitable, especially if data obtained at only a single time step are analyzed [19].

To reduce the number of candidate across-view associations that are evaluated at each time step, we make use of the traditional gating technique, which here utilizes the epipolar geometry as follows. Given a set of calibrated camera views, the projected images of objects lie on corresponding epipolar lines (or near these lines if there are inaccuracies in the calibration) and they should fall in the validation regions of the respective views, which are determined by the established 3D tracks (see Fig. 1). The validation region or gate is defined as

$$V_i^t(\gamma) = \{z : [z - H_i\hat{z}_t]' S_t^{-1} [z - H_i\hat{z}_t] < \gamma\} \quad (5)$$

where $\hat{z}_t$ is the predicted 3D position, S_t is the covariance matrix at time t (both of which can be evaluated using a standard Kalman filter), H_i is the projection matrix in the ith view, and γ an error threshold.

3.3. Tracking-Reconstruction Method

Our TRACKING-RECONSTRUCTION METHOD applies 2D tracking in each view independently and reconstructs 3D trajectories through track-to-track associations. It avoids creating redundant 3D phantom points, a drawback of the RECONSTRUCTION-TRACKING METHOD, but it has the disadvantage that occlusion negatively affects its 2D tracking performance (i.e., occlusion cannot impact the tracking performance of RECONSTRUCTION-TRACKING METHOD to the same extent, since it tracks in 3D). When an object is occluded in one view, the TRACKING-RECONSTRUCTION METHOD may not correctly connect the 2D track of the object before and after the occlusion, which then, in the reconstruction step of the method, may lead to an undesirable fragmentation of the 3D trajectory.

To reduce the occurrence of fragmented 3D trajectories, our TRACKING-RECONSTRUCTION METHOD analyzes and fuses the 2D track information from several views. Our method may reason that a long 2D track in one view may correspond to several short tracks in another view. In the most challenging case, an object may be occluded in each camera view at some point in time, leading to the occurrence of track fragments or "tracklets" in each view. The timing of the occlusion is typically not the same in any two views, which means that the 2D tracklets in different views that correspond to the same object usually have different start and end times (Fig. 2). Our TRACKING-RECONSTRUCTION METHOD first breaks long tracks into short tracklets so that candidate tracklets to be matched are aligned in time (Fig. 2). Tracklets are then matched in a greedy way where each tracklet can be matched multiple times. The resulting matched 2D tracklets are reconstructed into 3D trajectory pieces that are then linked into complete 3D trajectories. The pseudo code and technical details of this reconstruction and data association step of our TRACKING-RECONSTRUCTION METHOD are given below.

We denote $\mathcal{T}_{i_1 i_2 \ldots i_S}$ as an association of S tracklets $\mathcal{T}_{1,i_1}, \mathcal{T}_{2,i_2}, \ldots, \mathcal{T}_{S,i_S}$ with the same length L from S views. The cost of the association is defined as:

$$c(\mathcal{T}_{i_1 i_2 \ldots i_S}) = \frac{1}{L} \sum_{l=1}^{L} \sum_{s=1}^{S} ||z_{s,l} - H_s x_l||, \quad (6)$$

where $z_{s,l}$ is the lth 2D measurement along the track $\mathcal{T}_{s,i_s}$ in view s, H_s is the projection matrix of view s, and x_l is the reconstructed 3D point based on $(z_{1,l}, z_{2,l}, \ldots, z_{S,l})$.

From the 2D tracklets $\mathcal{T}^{(t)}_{s,i_s}, s = 1, \ldots, S$, from S views that are aligned in time at time instance t, our method reconstructs the corresponding 3D trajectory $\mathcal{T}^{(t)}_{i_t}$ if the association cost $c(\mathcal{T}^{(t)}_{i_1 i_2 \ldots i_S})$ is below a threshold τ. We use the same threshold τ and triangulation method as in the RECONSTRUCTION-TRACKING METHOD.

Our method attempts to link the 3D trajectory pieces $\mathcal{T}^{(t)}_{i_t}$, $t = 1, \ldots, N$, into longer 3D trajectories iteratively. It links two consecutive pieces $\mathcal{T}^{(t)}_{i_t}$ and $\mathcal{T}^{(t+1)}_{i_{t+1}}$ if (1) the start time of $\mathcal{T}^{(t+1)}_{i_{t+1}}$ is the next time step after the end time of $\mathcal{T}^{(t)}_{i_t}$; (2) the spatial distance between the end point of $\mathcal{T}^{(t)}_{i_t}$ and the start point of $\mathcal{T}^{(t+1)}_{i_{t+1}}$ is sufficiently small; (3) the linked trajectory is sufficiently smooth (we assume the object does not make drastic changes in direction).

TRACKING-RECONSTRUCTION METHOD
Reconstruction Step
Input: Sets $\mathcal{T}_s$, $s = 1, \ldots, S$, of 2D tracks from S views.

- **Breaking Phase:** Break each $\mathcal{T}_{s,i_s} \in \mathcal{T}_s$, $i_s = 1, \ldots, M_s$, into tracklets $\{\mathcal{T}^{(t)}_{s,i_s}\}$ at times $\{t_i\}$, where t_i is the start or end time of some track $\mathcal{T}_{r,i_r}$ in view r ($r \neq s$).
- **Association Phase:** For each $\mathcal{T}^{(t)}_{s,i_s} \in \mathcal{T}_{s,i_s}$, $t = 1, \ldots, N$, find its corresponding tracklets in other views with the same start and end times and compute its association cost $c(\mathcal{T}^{(t)}_{i_1 i_2 \ldots i_S})$ based on Eq. 6. If the cost is below threshold τ, reconstruct the 3D trajectory fragment $\mathcal{T}^{(t)}_{i_t}$.
- **Linking Phase:** Iteratively link trajectory fragments $\mathcal{T}^{(t)}_{i_t}$ into long trajectories until no more fragments can be linked.

Output: 3D trajectories.

---

4. Experiments and Results

We compared our two methods described in Sec. 3 for infrared video analysis of free-ranging bats. We processed the video of the emergence of a colony of Brazilian free-tailed bats from a natural cave in Blanco County, Texas [19]. The data was collected with three FLIR SC6000 thermal infrared cameras with a resolution of 640×512 pixels at a frame rate of 125 Hz. Our task was to track each bat in the emergence column of the colony and reconstruct their 3D flight trajectories (Fig. 3). Brazilian free-tailed bats can fly as fast as 30 mph, which at our frame rate resulted in significant displacements of the position of the same bat between two frames. In imaging scenarios where objects are displaced significantly from frame to frame, kernel-based trackers are not recommended [18]. The association problem is even more challenging in our case because we do not have sufficient appearance information to distinguish between bats, which look very similar to each other.

To detect the 2D position of each bat in each image frame, we used a method [2] that applies adaptive background subtraction followed by labeling of connected components. The size of the projection of a bat ranges from 10 to 40 pixels, depending on its distance to the camera. The position of a bat is represented

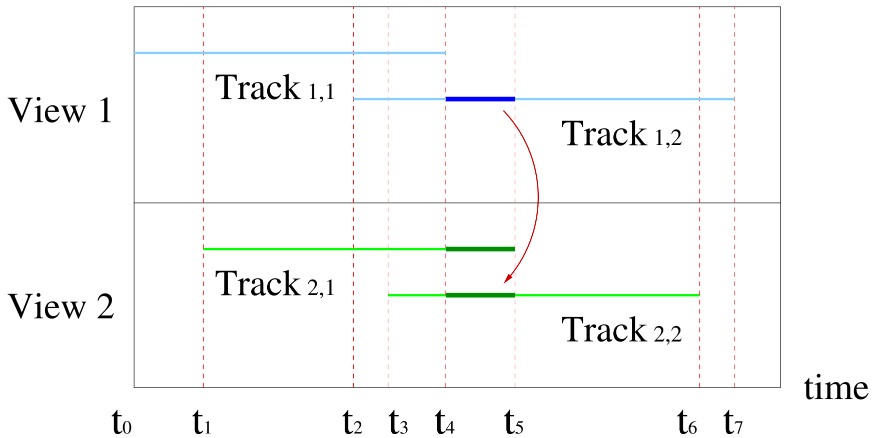

Figure 2. Preprocessing of 2D tracks before track-to-track association and reconstruction. The 2D tracks are broken into fragments based on the start and end points of all tracks. Here the two tracks in each view are respectively broken into 4 and 5 (view 1) and 4 and 3 (view 2) tracklets based on time indices $t_0, ..., t_7$. The subsequent association step matches the 9 tracklets in view 1 to the 7 tracklets in view 2. The red arrow shows a candidate match.

by the pixel with the highest intensity value within its connected component. Missed detections occur due to inter-object occlusion; false positive detections occur due to misinterpretation of background clutter. We used the same set of 2D positions measurements, including false positive detections, as input to the two tracking methods, so that we could conduct a reasonable comparison of their performance.

To evaluate the performance of our two tracking methods, we manually established the ground-truth 3D flight trajectories by visual inspection and compared them to the corresponding system-generated tracks. To evaluate the accuracy of a system-generated track $\mathcal{T}_i$, we measured the Euclidean distance of each object position x^i on $\mathcal{T}_i$ to the corresponding position x^j on the ground-truth trajectory $\mathcal{G}_j$. We adopted the track distance definition by Perera et al. [14],

$$D(\mathcal{T}_i, \mathcal{G}_j) = \frac{1}{|O(\mathcal{T}_i, \mathcal{G}_j)|} \sum_{t \in O(\mathcal{T}_i, \mathcal{G}_j)} ||x_t^i - x_t^j||, \quad (7)$$

which measures the sum of these distances for all time instances in the time-index set $O(\mathcal{T}_i, \mathcal{G}_j)$ for which both tracks include comparable object positions. The distance D can be interpreted as an error measure for the average distance between computed positions and true object positions.

The full motion trajectory of an object may not have been detected by our methods with a single track. In this situation, a set of consecutive tracks may collectively describe the object motion. For a ground-truth trajectory $\mathcal{G}_j$, we define the set $\mathcal{S}^*(\mathcal{G}_j)$ of associated system-generated tracks to include only those tracks that do not share a time index, i.e. their time-index set O is empty, and that minimize the sum of the track distances D between each system-generated track $\mathcal{T}_i$ and the ground-truth trajectory $\mathcal{G}_j$:

$$\mathcal{S}^*(\mathcal{G}_j) = \arg\min_{\mathcal{S}(\mathcal{G}_j)} \sum_{\mathcal{T}_i \in \mathcal{S}(\mathcal{G}_j)} D(\mathcal{T}_i, \mathcal{G}_j) \quad (8)$$

subject to $O(\mathcal{T}_a, \mathcal{T}_b) = \emptyset$, for all $\mathcal{T}_a, \mathcal{T}_b \in \mathcal{S}(\mathcal{G}_j)$.

Once we have identified the tracks $\mathcal{S}^*(\mathcal{G}_j)$ that collectively match the ground-truth trajectory $\mathcal{G}_j$, we can measure accuracy, completeness, and fragmentation of our results. Ideally, our results yield $|\mathcal{S}^*(\mathcal{G}_j)| = |\{\mathcal{T}_i\}| = 1$ (i.e., only one track $\mathcal{T}_i$ is associated with the ground-truth trajectory $\mathcal{G}_j$ (no fragmentation)) and $|O(\mathcal{T}_i, \mathcal{G}_j)| = |\mathcal{G}_j|$ (i.e., the number of time-indices of the object positions of $\mathcal{T}_i$ and $\mathcal{G}_j$ match (completeness)). We also count the number of tracks that are not matched with any ground-truth trajectory. This number indicates how many false positive or phantom tracks our methods produced. Our metrics to evaluate tracking performance are then defined by:

Track completeness:

$$\text{TC} = \frac{\sum_j \sum_{\mathcal{T}_i \in \mathcal{S}^*(\mathcal{G}_j)} |O(\mathcal{T}_i, \mathcal{G}_j)|}{\sum_j |\mathcal{G}_j|}, \quad (9)$$

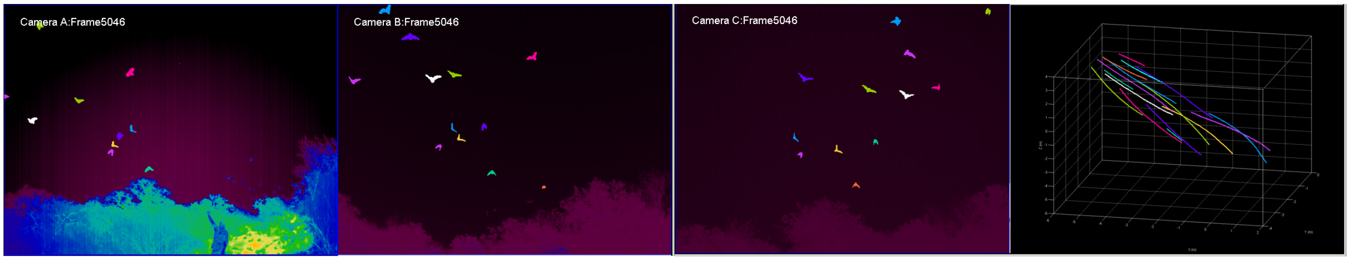

Figure 3. Results of tracking bats in flight in videos recorded by three infrared thermal cameras, which were placed near the entrance of a cave. Left: A false-color visualization of three synchronized frames with the tracked bats marked by distinct colors (the background-color differences in the thermal images are due to a lack of radiometric calibration of the cameras). Right: Visualization of the 3D trajectories of the group of bats. We used the same color to represent a specific bat in all three views and to display its tracked trajectory.

Track accuracy:

$$\mathrm{TA} = \frac{\sum_j \sum_{\mathcal{T}_i \in \mathcal{S}^*(\mathcal{G}_j)} D(\mathcal{T}_i, \mathcal{G}_j)}{\sum_j |\mathcal{G}_j|}, \quad (10)$$

Track fragmentation:

$$\mathrm{TF} = \frac{\sum_j |\mathcal{S}^*(\mathcal{G}_j)|}{|\{\mathcal{G}_j | \mathcal{S}^*(\mathcal{G}_j) \neq \emptyset\}|}, \quad (11)$$

Phantom track ratio:

$$\mathrm{PTR} = \frac{|\{\mathcal{T}_i \mid \forall \mathcal{G}_j : \mathcal{T}_i \notin \mathcal{S}^*(\mathcal{G}_j)\}|}{|\{\mathcal{G}_j \mid \mathcal{S}^*(\mathcal{G}_j) \neq \emptyset\}|}. \quad (12)$$

To test the performance of our two tracking methods for different levels of object density, we selected three scenarios of flight activity with approximately 10, 20 and 30 bats per video frame, respectively. For each of the scenarios, the data set contains three 100-frame sequences recorded from three synchronized thermal infrared cameras, respectively. For each tracking method, we used the same set of parameters, such as the width T of the sliding window, the threshold of the reconstruction residual, etc.

A summary of the tracking performance of our two methods is shown in Fig. 4. The track accuracy scores (Fig 4 top right) indicate that the average error in estimating the positions of bats were lower than 8 cm for both methods and for all three density levels of bats. We suggest that the magnitude of the error is small enough for future studies of bat flight behavior. A bat with fully extended wings has a width of 28 cm on average, and thus an 8-cm error may be small enough for reliable analysis of flight behavior. The difference in track accuracy between the two methods (the RECONSTRUCTION-TRACKING METHOD yielded smaller errors) was basically governed by data association accuracy, because we chose the same detection and triangulation procedures for both methods.

The TRACKING-RECONSTRUCTION METHOD generated a lower track completeness score (Fig 4 top left) and a higher track fragmentation score (Fig 4 bottom left) than the RECONSTRUCTION-TRACKING METHOD (in the ideal case both scores are one). The main reason for the difference is that false across-time data associations caused by occlusion in its tracking step affected the across-view 2D track associations in its subsequent reconstruction step.

While for the RECONSTRUCTION-TRACKING METHOD, the across-time 3D data associations were relatively reliable, the ambiguity in across-view data association created phantoms and thus yielded a higher phantom track ratio (Fig 4 bottom right).

Most phantom tracks are short. One way to reduce the phantom track ratio is to use a longer duration T for the sliding time window. This provides additional opportunities for the method to remove short track fragments, and thus identify phantoms. We evaluated the performance of the RECONSTRUCTION-TRACKING METHOD with different sliding window sizes T (Fig. 5). Our results show that the phantom track ratio was reduced when we used a longer-duration sliding window. The other performance measures, however, also decreased because short true tracks were removed incorrectly. Because of this trade-off, the window size that yields desired results with regard to all four metrics can only be chosen

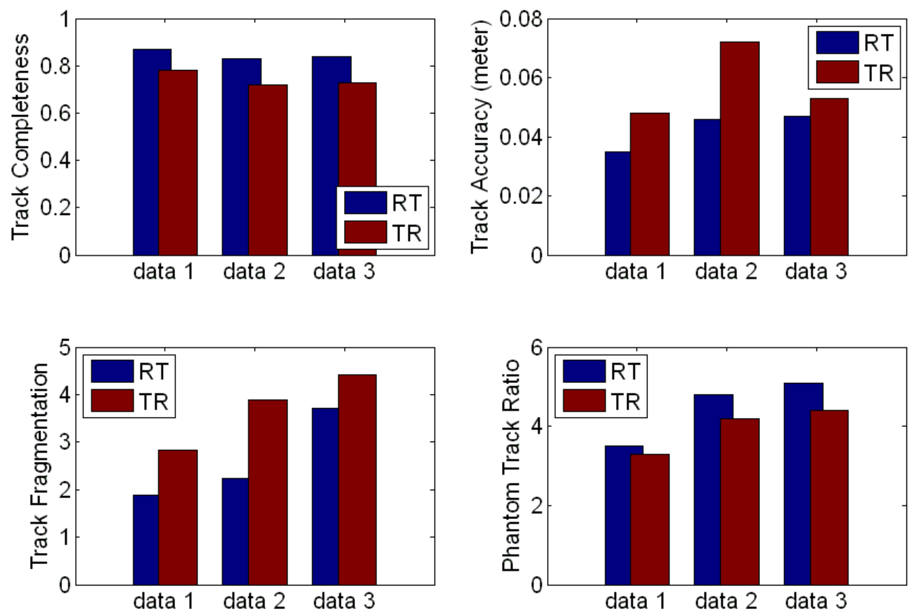

Figure 4. Comparison of our two multi-object multi-view 3D tracking approaches: RECONSTRUCTION-TRACKING (RT) and TRACKING-RECONSTRUCTION (TR). The three test data sets (data 1, 2, and 3) have different levels of object density: approximately 10, 20, and 30 bats per frame, respectively. The tracking performance of both methods is evaluated by four metrics: track completeness, track accuracy, track fragmentation, and phantom track ratio.

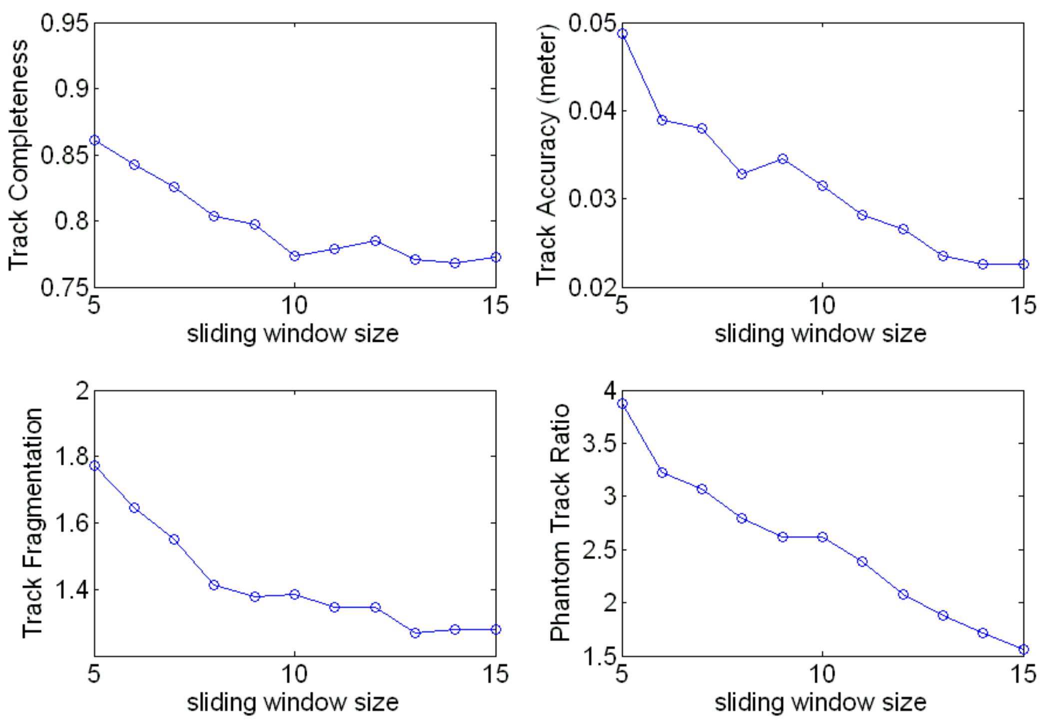

Figure 5. Tracking performance of the RECONSTRUCTION-TRACKING METHOD with different sliding window sizes T.

through experimentation.

5. Conclusion

We proposed and compared two multiple hypotheses tracking methods that use the reconstruction-tracking and tracking-reconstruction approaches respectively. Our analysis of thermal infrared video data of flying bats recorded by a three-camera system simultaneously revealed that the RECONSTRUCTION-TRACKING METHOD produced fewer track fragments than the TRACKING-RECONSTRUCTION METHOD but created more false positive 3D tracks. We do not make a general recommendation of one method over the other, but instead suggest that the TRACKING-RECONSTRUCTION METHOD may be used to interpret imaging scenarios when linking 2D track fragments is not difficult (e.g., because of a high frame rate and infrequent occlusions), while the RECONSTRUCTION-TRACKING METHOD may be used when additional information can reduce the number of false positive 3D tracks.

References

[1] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[2] M. Betke, D. E. Hirsh, A. Bagchi, N. I. Hristov, N. C. Makris, and T. H. Kunz. Tracking large variable numbers of objects in clutter. In *CVPR*, 2007.

[3] R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied Kalman filtering*. John Wiley and Sons, Inc, 1997.

[4] I. Cox and S. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. PAMI*, 18:138–150, February 1996.

[5] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. A generalized s-d assignment algorithm for multisensor-multitarget state estimation. *IEEE Trans. AES*, 1997.

[6] S. L. Dockstader and A. Tekalp. Multiple camera fusion for multi-object tracking. In *IEEE Workshop on Multi-Object Tracking*, 2001.

[7] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *CVPR*, 2008.

[8] R. I. Hartley and A. Zisserman. *Multiview view geometry in computer vision*. Cambridge University Press, 2003.

[9] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *ECCV*, pages 133–146, 2006.

[10] Y. Li, A. Hilton, and J. Illingworth. A relaxation algorithm for real-time multiple view 3d-tracking. *Image Vis Comput*, 20:841–859, 2002.

[11] A. Mittal and L. S. Davis. M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*, 51(3):189–203, 2003.

[12] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking: Linking identities using bayesian network inference. In *CVPR*, 2006.

[13] K. Otsuka and N. Mukawa. Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles. In *CVPR*, pages 90–97, 2004.

[14] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006.

[15] A. B. Poore. Multidimensional assignment formulation of data association problems rising from multitarget and multisensor tracking. *Computational Optimization and Applications*, 3(1):27–57, 1994.

[16] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, 24:843–854, December 1979.

[17] A. J. Robertson. A set of greedy randomized adaptive local search procedure (GRASP) implementations for the multidimensional assignment problem. *Computational Optimization and Applications*, 19(2):145–164, 2001.

[18] A. Tyagi, G. Potamianos, J. Davis, and S. Chu. Fusion of multiple camera views for kernel-based 3d tracking. In *IEEE Workshop on Motion and Video Computing*, 2007.

[19] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke. Tracking a large number of objects from multiple views. In *ICCV*, 2009.