ComputerScience

BOSTON UNIVERSITY

Boston University

Efficient Track Linking Methods for Track Graphs Using Network-flow and Set-cover Techniques

Zheng Wu[1], Thomas H. Kunz[2], Margrit Betke[1]

[1]Computer Science and [2]Biology Department, Boston University, USA

Problem

Linking track fragments with short-term, long-term or multi-view occlusions

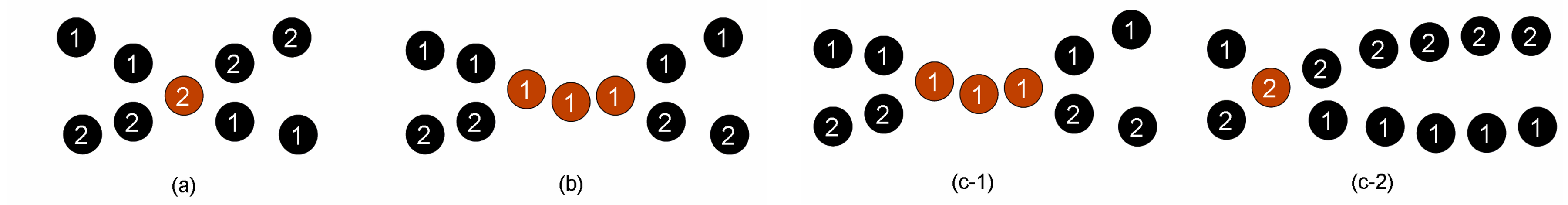


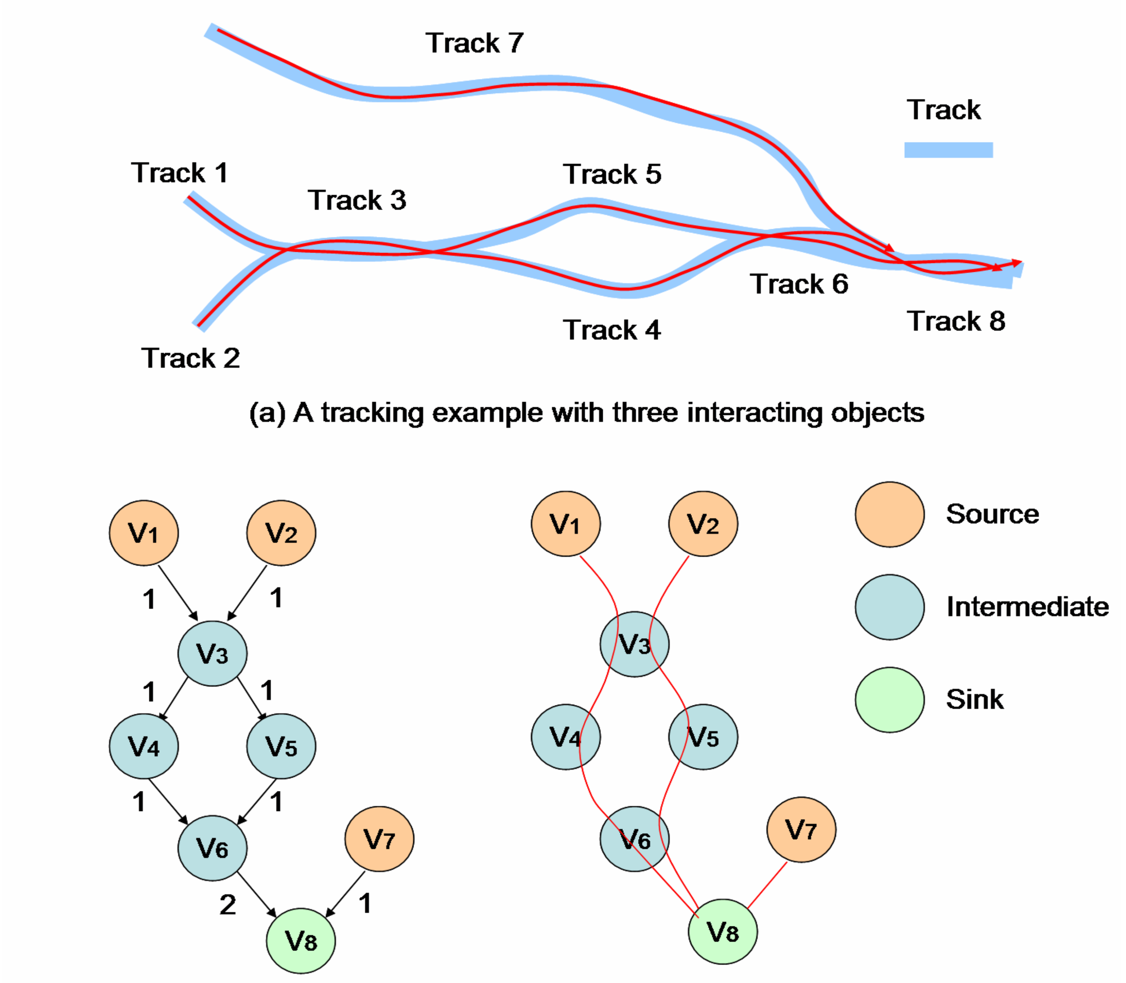
Fig.1 (a) Short-term occlusion, (b) Long-term occlusion, and (c) Occlusion in two camera views. Red nodes represent merged measurements; numbers are labels for objects.

Contributions

- For short-term occlusion, our method automatically constructs the track graph and resolves it by computing the minimum flow and a series of bipartite graph matches.
- For long-term occlusion, we convert the problem of track graph reasoning to the standard set cover problem, for which a logarithmic approximation solution exists.
- For occlusions in multiple views, our method constructs the track graph for each view independently and resolves the track graphs jointly by solving a joint-set-cover problem. To the best of our knowledge, we are *the first* to formulate the track linking problem in both the *temporal* (across-time) and *spatial* (across-camera) domains.

Track Graph

Define track graph *G=(V,E)* , where *V* is set of vertices that represent individual or merged tracks. *E* is set of directed edges that represent merging or splitting events. *f* is the *flow* on the edge indicates how many objects are involved during the merging or splitting event. t_v is the *track capacity* that shows how many objects the track contains.



(a) A tracking example with three interacting objects

(b) Track graph with associated flows (c) Resolved track graph after linking

Track Graph Construction

- *Building the structure of graph*: tracking forward and backward to produce the tracklets and merge/split hypothesis. For example, if multiple trackers predict a collision in image in next frame, all trackers are terminated and a new tracker is initiated to track the overlapped measurements.
- *Determine the flow* : solving a *minimum-flow* problem where the lower bound on the capacity of each edge is one. False alarm tracklets are assumed to be identified and excluded during the building stage.

Local Linking Method

Designed for scenario in Fig.1 (a)

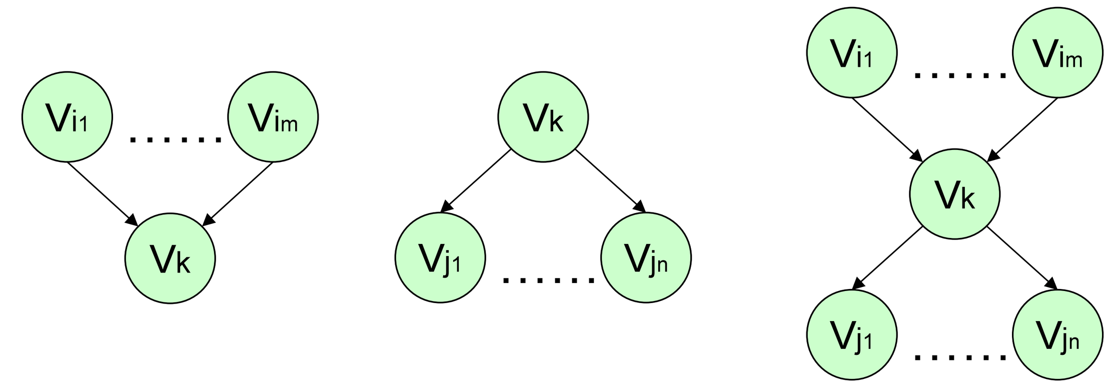


Fig. 2 The local structure of track graph can only have three types shown above. Only the last type needs a nontrivial reasoning which is a *bipartite matching* problem.

Global Linking Method

Designed for scenario in Fig.1 (b)

A *generalized set-cover* formulation:

For a given track graph, we enumerate all possible paths from source set *S* to sink set *T*, where each path consists of a sequence $\{v_{i1}, v_{i2} \dots v_{in}\}$ of vertices visited. The set of all paths is denoted as *P*. A weight w_p is associated with a path *p* that measures the likelihood of the path being a true trajectory, or equivalently the cost of the path. The objective is to select a subset *P'* of *P* such that the sum of the costs of all selected path is minimum. Each vertex *v* from *G* has to be in some path at least t_v times, where t_v is the track-capacity of *v* computed from the minimum flow during the construction of the graph.

Solver: *Greedy Method* or *Linear Relaxation* (same approximation quality)

Linking in Multiple Views

Designed for scenario in Fig.1 (c)

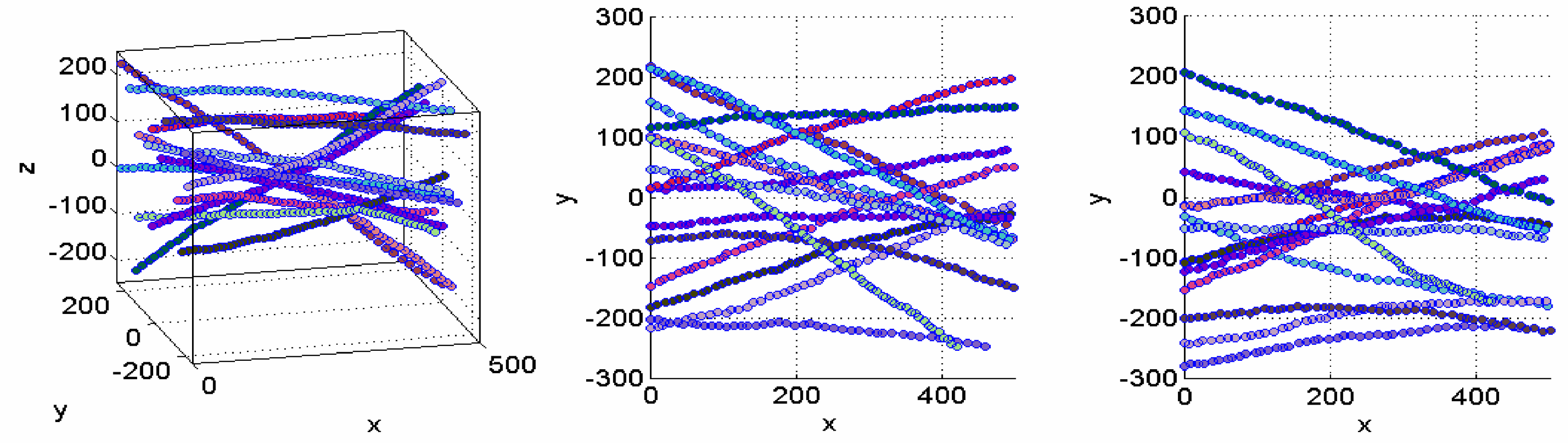
A *joint set-cover* formulation:

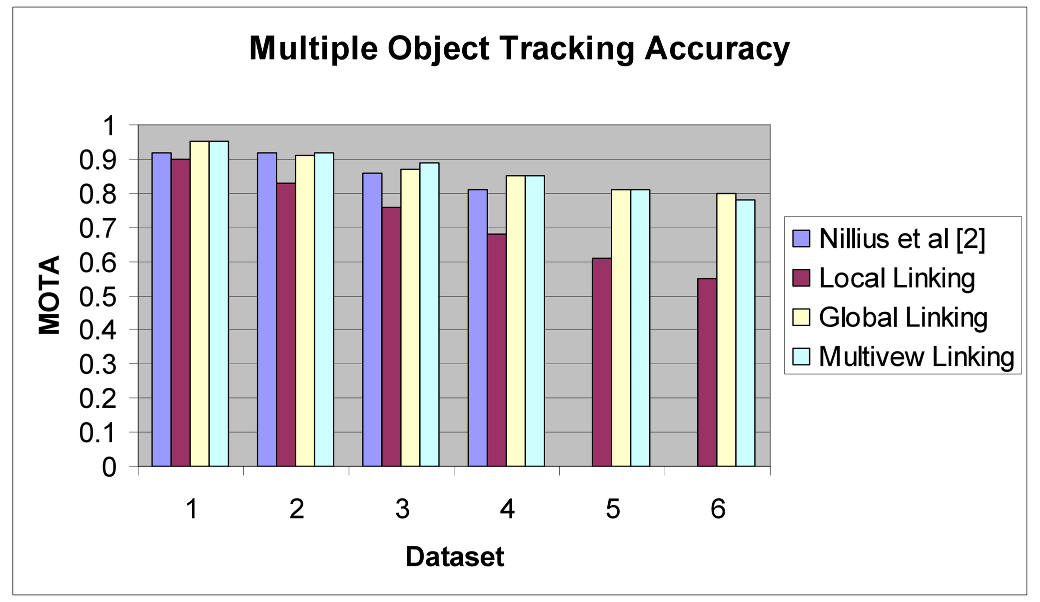
Given two track graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$, enumerate all valid paths as set P_i. Compute a_p and b_q to measure the respective likelihood of path *p* in P_1 and *q* in P_2 being true trajectories. The objective is to choose a subset P_i' to achieve a cover on V_i for each view, subject to the additional constraint that enforces that any selected path *p* in P_i' has a corresponding path *q* in P_j' with matching cost $c_{p,q}$. We seek the solution that achieves the minimum weighted sum.

Proposition: The *joint set-cover* problem can be reduced to a standard set cover problem

Experiment 1: Synthetic Data

Randomly generated spheres using a linear dynamic model with constant velocity plus Gaussian noise. 6 datasets of increasing density (up to 60 objects per frame) with two views were tested.



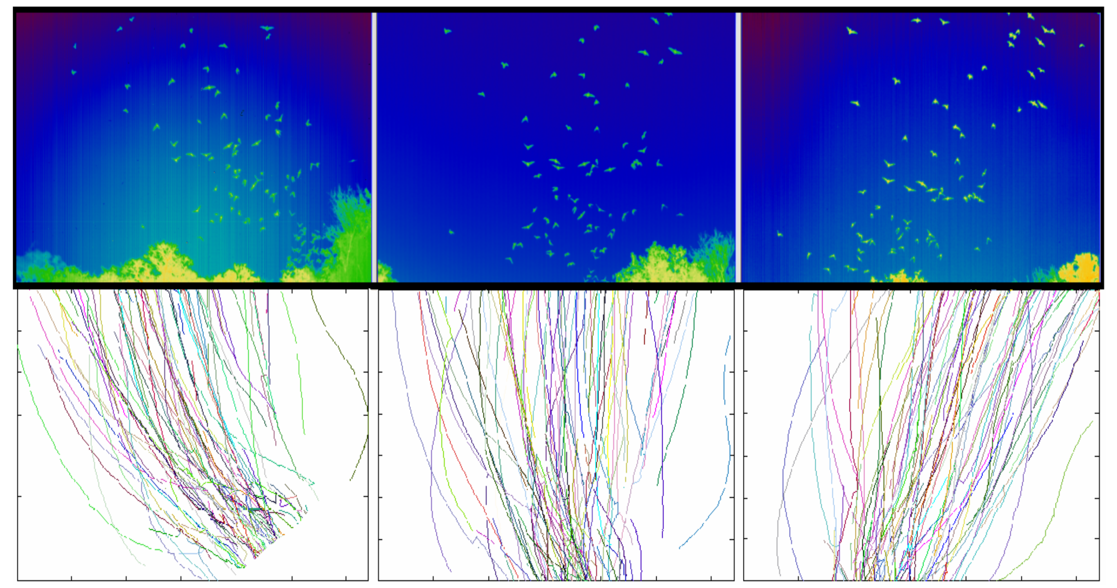


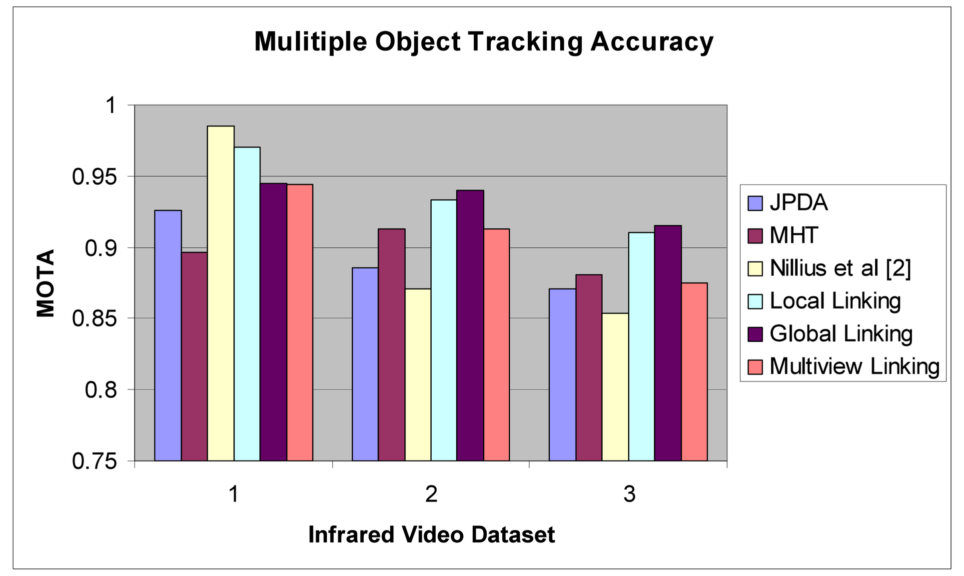
Multiple Object Tracking Accuracy (MOTA)[1] = 1 - Miss Rate - False Positive Rate - Mismatch Rate

Nillius et al [2]: Consider track graph as Bayesian Net and use Junction-Tree algorithm for inference

Experiment 2: Infrared Thermal Video Analysis

3 Datasets of increasing density (up to 100 objects per frame) with three thermal infrared cameras.





[1] K. Bernardin and R. Stiefelhagen. Evaluation multiple object tracking performance: The CLEAR MOT metrics. EURASIP Journal on Image and Video Processing, 2008

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