OCCLUSION REASONING FOR MULTIPLE OBJECT VISUAL TRACKING

ZHENG WU

Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

BOSTON

UNIVERSITY

BOSTON UNIVERSITY
GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

OCCLUSION REASONING FOR MULTIPLE OBJECT VISUAL TRACKING

by

ZHENG WU

B.S., Zhejiang University, China, 2003
M.S., Zhejiang University, China, 2006

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
2013

© Copyright by
ZHENG WU
2012

Approved by

First Reader

Margrit Betke, PhD
Professor of Computer Science

Second Reader

Stan Sclaroff, PhD
Professor of Computer Science

Third Reader

David Castañón, PhD
Professor of Electrical and Computer Engineering

Acknowledgments

I want to especially thank my advisor, Prof. Margrit Betke for her mentorship in every aspect of my research life. Margrit has been very patient to instruct me with every piece of my work in the past six years, providing extremely professional guidance and kind encouragement to help shape my research career. I would also like to acknowledge all thesis committee members, Prof. Stan Sclaroff, Prof. David Castañón, Prof. Hao Jiang, Prof. Steve Homer and Dr. Fatih Porikli, for taking their valuable time to participate my defense and providing insightful suggestions. I thank all my collaborators in Biology department, especially Prof. Tom Kunz and Nathan Fuller, to give me an extraordinary fieldwork experience as a computer scientist.

I thank all IVC members, past and present, for all interesting discussions on research and life in general. They are not only wise collaborators but also sincere friends, without whom this thesis would not have been possible. Jingbin, Quan, Rui, Taipeng are my first four friends in this country who have offered me tremendous help in life. Ashwin is the most interesting Indian friend I have ever known who shares a lot in common with me. Vitaly, Murat, Bill, John, Vassilis were my seniors who broadened my view of graduate life when I was a fresh Phd student. It was also an enjoyable experience to work with Margrit's "special force:" Sam, Chris, Diane, Gordon, Mikhail, Danna, Erik, and other young fellows: Qinxun, Shugao, Fatih and Kun.

I own great thanks to my family for always being supportive. My family is my strongest motive to keep moving in my career, and it is a great reason worth fighting for. My dear wife, Yuan, has been very kind to understand and encourage me every day for my research work. As the best annotator (and of course the best woman) I know in my life, she provided a lot of high-quality annotations for my experiment, which, hopefully, will be influential for the computer vision community.

OCCLUSION REASONING FOR MULTIPLE OBJECT VISUAL TRACKING

(Order No.)

ZHENG WU

Boston University, Graduate School of Arts and Sciences, 2013

Major Professor: Margrit Betke, Professor of Computer Science

ABSTRACT

Occlusion reasoning for visual object tracking in uncontrolled environments is a challenging problem. It becomes significantly more difficult when dense groups of indistinguishable objects are present in the scene that cause frequent inter-object interactions and occlusions. We present several practical solutions that tackle the inter-object occlusions for video surveillance applications.

In particular, this thesis proposes three methods. First, we propose "reconstruction-tracking," an online multi-camera spatial-temporal data association method for tracking large groups of objects imaged with low resolution. As a variant of the well-known Multiple-Hypothesis-Tracker, our approach localizes the positions of objects in 3D space with possibly occluded observations from multiple camera views and performs temporal data association in 3D. Second, we develop "track linking," a class of offline batch processing algorithms for long-term occlusions, where the decision has to be made based on the observations from the entire tracking sequence. We construct a graph representation to characterize occlusion events and propose an efficient graph-based/combinatorial algorithm to resolve occlusions.

Third, we propose a novel Bayesian framework where detection and data association are combined into a single module and solved jointly. Almost all traditional tracking systems address the detection and data association tasks separately in sequential order. Such a design implies that the output of the detector has to be reliable in order to make the data association work. Our framework takes advantage of the often complementary nature of the two subproblems, which not only avoids the error propagation issue from which traditional "detection-tracking approaches" suffer but also eschews common heuristics such as "non-maximum suppression" of hypotheses by modeling the likelihood of the entire image.

The thesis describes a substantial number of experiments, involving challenging, notably distinct simulated and real data, including infrared and visible-light data sets recorded ourselves or taken from data sets publicly available. In these videos, the number of objects ranges from a dozen to a hundred per frame in both monocular and multiple views. The experiments demonstrate that our approaches achieve results comparable to those of state-of-the-art approaches.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Main Contributions 3

1.3 Organization of the Thesis 5

1.4 List of Related Papers 6

2 Tracking in Multiple Views **8**

2.1 Related Work 8

2.2 Reconstruction-Tracking Method 14

2.3 Experiments 23

2.4 Summary and Discussion 30

3 Track Linking on Track Graph **32**

3.1 Related Work 32

3.2 Track Linking Methods 35

3.3 Experiments 46

3.4 Summary and Discussion 52

4 Coupled Detection and Association **54**

4.1 Related Work 55

4.2 The Coupling Framework 58

4.3 Experiments 71

4.4 Summary and Discussion 85

5 Conclusions and Future Work **87**

5.1 Main Contributions 87

5.2 Limitations and Future Work 89

References **92**

Curriculum Vitae **100**

List of Tables

2.1 Notation for reconstruction-tracking method 14
2.2 Quantitative results for multi-view tracking algorithm 29

3.1 Summary of related work and proposed track linking methods 35
3.2 Notation for track linking method 36
3.3 Statistics of synthetic datasets and comparison results 48
3.4 Quantitative results for track linking on infrared videos – I 50
3.5 Quantitative results for track linking on infrared videos – II 51

4.1 Notation for coupled detection and association method 58
4.2 Quantitative results for coupling algorithm 82

5.1 Summary of detection approaches 88
5.2 Summary of tracking approaches 89

List of Figures

1·1 Overview of multiple object tracking system 2

2·1 Example of occlusion reasoning via across-view association 18
2·2 Example of greedy search for multidimensional assignment problem 20
2·3 Example of the across-time multidimensional assignment problem 23
2·4 Sample frames from the swallow dataset 24
2·5 Data collection for the emergence of Brazilian free-tailed bats 25
2·6 Two sources of error for spatial data association 30

3·1 Example of three different occlusion scenarios 34
3·2 Example of track graph representation 37
3·3 Example of track linking with a bipartite matching formulation 40
3·4 Example of track linking with network flow formulation 41
3·5 Example of the stochastic process to generate tracklets 45
3·6 Sample trajectories from the synthetic dataset 47
3·7 Tracking performance on simulation dataset 48
3·8 Sample infrared video frames and system-generated trajectories 49
3·9 Tracking performance on infrared videos – I 52
3·10 Tracking performance on infrared videos – II 52

4·1 Graphical model for multiple object tracking problem 59
4·2 Example of constructing dictionary for planar motion 62
4·3 Example of constructing dictionary for 3D motion 62
4·4 Example of one-bit quantization effect 64
4·5 Example of minimum-cost network-flow data association 66

4·6 Sample box signals from the synthetic dataset 72
4·7 Comparison of LDND and LDQD on the synthetic dataset 73
4·8 Detection performance on PETS2009 dataset 76
4·9 Sample frames and detection results on PETS2009 dataset 77
4·10 People counting results on PETS2009 dataset 78
4·11 Example of performance of the coupling algorithm at each iteration 84
4·12 Sample frames and tracking results for the coupling algorithm 86

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
CP	Coupling algorithm
DLT	Direct Linear Triangulation
FPR	False Positive Rate
IDS	ID Switch
IGRASP	Iterative Greedy Randomized Adaptive Search Procedure
LDND	Linear Denoising Decoder
LDQD	Linear Dequantization Decoder
ML	Mostly Lost
MMR	Mismatch Rate
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MODA	Multiple Object Detection Accuracy
MODP	Multiple Object Detection Precision
MR	Miss Rate
MT	Mostly Tracked
RT	Reconstruction-tracking algorithm
SDD	Sparsity-driven Detector

Chapter 1

Introduction

1.1 Motivation

A lot of efforts have been made in computer vision to interpret the motion of large groups of individuals. Applications range from video security surveillance to behavioral studies, from medical image analysis to monitoring of wild animals. They all rely on the performance of a robust multiple object tracking system. The performance and accuracy of multi-object tracking systems is still far from being satisfactory for two major reasons: finding a general object detection method still remains an open question, and the scalability to handle dozens or even hundreds of objects based on existing techniques is quite poor.

One cause of the difficulties is the occlusion/interaction event that breaks many assumptions held by the existing systems. After all, if the objects in the scene are well separated without interaction or occlusion, it seems not so challenging to track all of them. A lot of difficult tracking scenarios involve occlusion, including self-occlusion, inter-object occlusion, or static occluders in the scene. It makes tracking even more difficult if objects do not have distinctive appearance among each other. Recently, "Occlusion" and "Confusion" are categorized to be two of the most difficult cases related to multiple object visual tracking [26]. Thus, we believe that improving occlusion reasoning is the crucial step in attaining improved tracking performance, and therefore it is the focus of this thesis.

In general, a complete multi-object tracking system typically consists of three components, as illustrated in Fig. 1·1: object detection, temporal data association, i.e., the assignment of current observations to object tracks, and state estimation of each object. Within this classic framework, previous works typically perform occlusion reasoning from

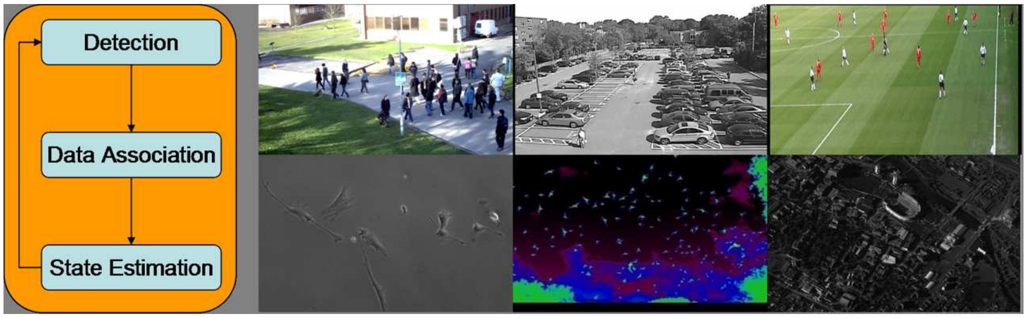


Figure 1·1: For each time step, a typical tracking system needs to sequentially solve object detection, data association and object state estimation. Samples images from different tracking applications (in left-to-right top-to-bottom order: PETS2009 [64], COP2007 [1],VS-PETS2003 [84],BU-Cell [88],BU-Bat [18] and CLIF2007[27]) are shown on the right, where object interaction/occlusion is frequent.

two aspects: building a stronger object detector that accounts for partial visibility and modeling the missed detection event in data association. State-of-the-art object detection methods are usually class-specific and require sufficient image resolution in order to extract dense features [37] from the object. Even for well-studied categories of objects such as pedestrians, current techniques are still sensitive to occlusion and their performance drops catastrophically if the object is only partially visible in the image [33].

Our research in developing methods for occlusion reasoning that support the task of data association aims to be independent of a particular image-understanding application. Therefore, this thesis focuses on data association and detection methods for occlusion reasoning that are not dependent on the class of the object of interest. In our experiments, the objects are typically imaged at low resolution, which excludes the possibility of building a complicated object appearance model for tracking. Furthermore, we are interested in accounting for inter-object occlusion and interaction. Methods that model self-occlusion for articulated objects [78, 74] and methods that learn scene occluders [1, 70] are complementary to our approach. The analysis of the outputs from our tracking algorithms, 3D trajectories of flying bats, birds and insects, and 2D trajectories of people and animals, is expected to have broad impact on the understanding of group behavior [18, 51, 82, 54] and trajectory-based abnormality detection in surveillance studies [3, 4, 21, 85].

1.2 Main Contributions

In order to resolve the ambiguity in maintaining tracks due to occlusion events, there are two research approaches concerning the data association aspect: accumulating observations from multiple views or accumulating them from additional frames in a batch-processing way. When multiple camera views are available, we also need to consider a spatial (*across-view*) data association problem: the determination of corresponding observations of the same object from multiple views. When batch processing is possible, a proper formulation should provide an efficient algorithm to handle the much larger or possibly overwhelming data to process, compared to the data demands of sequential processing. However, most previous works on this topic either underestimate the spatial data association problem in general or resort to a computationally expensive algorithm to solve the underlying optimization problem. In contrast, our work addresses the spatial and temporal data association problem in a multi-view setting, and we propose a new framework to model occlusion events for batch processing that leads to various efficient algorithms that address the short-term, long-term, and multi-view occlusion scenarios, respectively.

Another novel aspect of our approach to improve occlusion reasoning is our idea to consider both detection and data association modules at the same time . Although it might be easier to maintain each module separately from a system point of view, we suggest there are good reasons to combine these two modules. Indeed, how to detect multiple objects from images still remains one of the fundamental research problems that the computer vision community works on. First, without knowing the number of objects in the image, the detector is typically designed to produce a sufficiently large number of candidate detections and then heavily relies on the data association method to identify the false alarms among them. Second, severe occlusion creates challenges as the image evidence (pixels) from the occluded region is usually shared and explained by multiple detections. This makes it fairly difficult to estimate the right number of objects or reason about the occluders and occludees. Despite the trend in the research community of attempting to improve the accuracy of an object detector by using more powerful machine learning tools, we argue that there are two major drawbacks in the detection approaches of current tracking systems: 1) The detection phase is completely separated from the task of data association. Therefore, any type of detection error is propagated and must be fixed later. 2) The projected images of multiple objects in the scene are assumed to occur independently so that the occlusion relationship on the image plane is not modeled properly. Instead, we would like to couple the detection and data association into a single mathematical objective function. Therefore, the subproblems, detection and data association, can benefit from each other, which leads to a more robust and smoothed solution. From a theoretical point of view, such a combination can also be derived from a Bayesian estimation framework, where the key difference compared to previous work is how to factorize the observation likelihood term. In particular, we choose a sparsity-driven detection formulation as our detector that models image likelihood jointly for binary image observations, and combine it with a classic network-flow data association technique. The coupled objective function is further solved by a dual decomposition algorithm.

In summary, the main contributions of the thesis are:

(a) For sequential tracking in multiple views, we propose a "reconstruction-tracking" algorithm that performs spatial-temporal data association [90, 89]. For the reconstruction step, we are the first to propose adding a sparsity constraint to reduce false alarms, known as the "ghost" effects in stereoscopy (Chapter 2).

(b) For batch processing, we develop a unified framework to perform "track linking" with a graph representation [87], known as the "track graph" [60]. Depending on the complexity of occlusion, we propose several different efficient algorithms by converting the original linking problem into network flow, set-cover and joint set-cover problems, respectively (Chapter 3).

(c) For coupling the detection and data association problems, we propose a novel Bayesian framework that combines a sparsity-driven detection method and network-flow data association method into a single objective function. The sparsity-driven detector is able to suppress hypotheses and recover occlusion relationships jointly. To handle the scalability, we adopt a dual decomposition method that allows tracking up to hundreds of objects in a batch process [91] (Chapter. 4).

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows:

Chapter 2 describes our multi-camera, multi-object tracking algorithm. We show how to sequentially solve the two "across-view" and "across-time" data association steps for tracking dense groups of objects moving in free 3D space, which we call the "reconstruction-tracking" method. The underlying combinatorial formulation is adapted from the multi-dimensional assignment problem, and we propose a modified greedy randomized adaptive search procedure to solve it. Despite its success for tracking objects in sparse density, we point out some limitations of this approach when applied to more challenging tracking scenarios at the end of this chapter.

Chapter 3 describes our track linking algorithm. We show how to construct a graph representation that characterizes the occlusion/interaction events in video sequences and how to resolve the occlusion relationship later using a combinatorial algorithm. Depending on the space-time characteristics of the occlusion events, we formulate the resolving process as a bipartite matching, minimum-cost flow, or set-cover problem. At the end, we also give a Bayesian interpretation to justify the proposed approaches.

Chapter 4 explains our novel Bayesian coupling framework that combines detection and data association into a single objective function. Under this framework, we first present our sparsity-driven object detector that works with binary image input, both for monocular and multi-view videos. It not only overcomes the limitation of our baseline tracker described in Chapter 2, but also simultaneously infers the occlusion relationship. We further combine the sparsity-driven detection method with a network flow association method for tracking. We show the strength of the coupling framework by presenting its performance across several challenging, notably distinct datasets. Our algorithms achieve consistent robustness and outperform state-of-the-art techniques.

Chapter 5 summarizes and discusses the key contributions of the thesis work. Some extensions and generalization of our approaches to other computer vision problems are also discussed.

Each chapter is more or less self-contained and has its own literature review and experiment section. A reader who is interested in only one category of approaches could look up the related chapter without extensively going through other chapters.

1.4 List of Related Papers

This thesis is based in part on the following publications with extended formulations and expanded experiments:

- Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. "Coupling Detection and Data Association for Multiple Object Tracking," in Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, June, 2012 [91].

- Z. Wu, M. Betke and T. H. Kunz. "Efficient Track Linking Methods for Track Graphs Using Network-flow and Set-cover Techniques," in Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Springs, Colorado, June, 2011 [87].

- Z. Wu, N. I. Hristov, T. H. Kunz, and M. Betke. "Tracking-Reconstruction or Reconstruction-Tracking? Comparison of Two Multiple Hypothesis Tracking Approaches to Interpret 3D Object Motion from Several Camera Views," in Proceeding of IEEE Workshop on Motion and Video Computing (WMVC), Utah, December, 2009 [90].

- Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke. "Tracking a Large Number of Objects from Multiple Views," in Proceeding of the 12th International

Conference on Computer Vision (ICCV), Kyoto, Japan, September, 2009 [89].

The material in this thesis is based upon work partially supported by the National Science Foundation under IIS-0910908, IIS-0855065, IIS-0308213, IIS-0713229, and Office of Naval Research under ONR 024-205-1927-5 and the Air Force Office of Scientific Research.

Chapter 2

Tracking in Multiple Views

In this chapter, we tackle the issue of occlusion with a multi-camera setup. Cameras are assumed to be calibrated with overlapping fields of view. Videos that capture the motion of objects are recorded with a relatively high frame rate. We assume an appropriate detection algorithm has been developed so that possible 2D locations of objects have been identified in the images. We also assume, however, that inaccurate segmentations and merged measurements due to occlusion and object interaction are not identified in the detection stage. Our focus is to rely on the data association module to maintain the trackers during occlusion events. We first review state-of-the-art data association techniques as well as customized approaches in multi-camera environments in Sec. 2.1. Our detailed multi-object multi-view approach is explained in Sec. 2.2 with supporting experiments in Sec. 2.3. We conclude this chapter in Sec. 2.4 by discussing the strengths and limitations of the proposed approach.

2.1 Related Work

2.1.1 Classic Data Association Approach

The purpose of data association in a tracking system is to ensure the correct correspondence between objects and observations. Otherwise, the state estimates obtained via algorithms such as recursive Bayesian filtering will be based on inaccurately associated observations and the object identity will not be maintained consistently. The radar literature describes some fundamental algorithms for tracking multiple targets within a dynamic system [10], such as Multiple Hypothesis Tracking (MHT) and Joint Probabilistic Data Association (JPDA). MHT [69] enumerates all possible combinations through time by building a hypothesis tree, and picks the best one, i.e., with the highest likelihood, as its solution. In practice, it requires a lot of heuristics to prune the hypothesis tree to avoid its exponential growth [29]. On the other hand, JPDA only looks for correspondences between two frames and does not pursuit the best solution but computes the expectation of track states over all the hypotheses. These Bayesian probabilistic methods need to integrate filtering techniques, such as a Kalman filter [22] or a particle filter [58]. Extension of these methods that accommodate extended object measurements also emerged recently in order to recover object pose and reduce the uncertainty of data association at low frame rates [38].

The probabilistic association methods have their integer optimization counterparts in linear network optimization problems [15]. The most popular formulation is the bipartite matching problem (or 2D assignment problem) [81], where many polynomial-time algorithms exist such as the Hungarian method, Auction method, and JVC method [11, 17]. The minimum-cost flow formulation proposed by Zhang et al. [96] for multiple pedestrian tracking can also be classified into this category since the 2D assignment problem can be considered a special case of the minimum-cost flow problem. A similar linear programming formulation was also presented by Jiang et al. [48] but they augmented the global cost function with a pairwise distance measure. However, because they used the Manhattan metric, the optimization still remains linear and does not increase the complexity compared to bipartite matching. In contrast, the discrete optimization version of MHT, known as the multidimensional assignment problem [66], is NP-hard. It can be seen as finding a weighted maximum matching on a hypergraph, where a hyperedge must connect more than two vertices at the same time. Therefore, it is a generalization of bipartite matching to N-partite matching. To solve this NP-hard problem, the popular semi-definite programming (SDP) technique was adopted by Shafique et al. [73] who relaxed the original discrete optimization to a rank-constrained continuous optimization. Alternatively, an iterative Lagrange relaxation procedure was applied by Deb et al. [30] to the dual problem. The procedure halted its iterations when the duality gap was sufficiently small.

Despite efforts to handle the underlying NP-hard combinatorial optimization, methods described above all inherit the enumerative nature introduced by the "hard assignment" that explicitly assigns observations to tracks exclusively and completely. In contrast, the novel idea of "soft assignment," also known as Probabilistic Multiple Hypothesis Tracking (PMHT), was originally developed by Streit et al. [77], which treated the assignments themselves as random variables or non-observed "missing data" and converted the data association problem to a soft clustering problem or incomplete data estimation problem. Both the work by Gauvrit et al. [41] on passive SONAR and Yu et al. [94] on pedestrian tracking are along this direction. The main issue with these approaches is that the inference algorithm used typically, EM or variational EM, has relatively slow convergence and is sensitive to the initial estimate of the model parameters. When a large number of objects needs to be tracked, many model parameters must be estimated. As a result, the problems of sensitivity to initial starting points and slow convergence present a challenge to applying these EM-type algorithms.

Sampling based algorithms form another category of data association methods that gained popularity recently, partially because of the advance of Monte Carlo theory applied to practical image understanding problems. Oh et al. [61] first proposed a general framework to sample the data association hypotheses directly with Markov Chain Monte Carlo (MCMC) sampling. It is a batch processing method and able to handle object arrival and departure at the same time. For sequential tracking, Kevin et al. [75] defined the dimension of state space to be correlated with the varying number of objects in the scene and applied Reversible Jump Markov Chain Monte Carlo (RJMCMC) sampling that allows transition between state spaces of different dimensions. Although theoretically it is difficult to conclude whether a sampling-based method outperforms the deterministic combinatorial optimization method or not, the sampling-based method does have the flexibility to deal with more complicated region tracking scenarios. Khan et al. [51] introduced a probabilistic model to associate merged and split measurements using a MCMC-based particle filter. Yu and Medioni [93] also extended the general framework by Oh et al. [61] to find the best spatial and temporal association of regions to track with Data-Driven Markov Chain Monte

Carlo (DDMCMC) sampling. However, as for most sampling techniques, determining how to achieve fast convergence is always a nontrivial task [56].

Finally, we want to note that all these classic probabilistic and determinatistic approaches were originally designed for *temporal* data association, that is, to match the measurements obtained from different time frames. Most of them treat occlusion events as a sign of missed detections or merged measurements. For missed detections, temporal data association serves as an interpolation for time series data. For merged measurements (occluded objects have extended images overlapping on the image plane), temporal data association has to relax a common constraint that forces each tracker to be matched exclusively to one measurement. Although each of these two ideas has its strength for different object image resolutions, they all suffer from long-term occlusion events.

2.1.2 Multi-view Data Association Approach

For situations when many objects emerge at the same time in the image of the scene and occlusion occurs frequently, single-view approaches are not so promising. An alternative way is to use more than one camera to provide information from different views [40, 55, 89, 59, 36, 50, 62]. It involves another type of data association task, that is to find the correspondence of objects across cameras views. We call such task *spatial* or "across-view" data association as opposed to *temporal* or "across-time" data association. Using multiple views is advantageous because when occlusion occurs in a certain view, it might not happen in other views. In addition to occlusion reasoning, multi-view tracking also assists generating 3D trajectories of an object's motion based on epipolar geometry [46].

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 the 2D tracks of the 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-failure," 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. Existing works on human tracking from multiple camera views have compared the two schemes [80, 55] and have generally favored the reconstruction-tracking scheme.

For the reconstruction-tracking scheme, tracking is performed in 3D [36, 59, 98, 80], using reconstructed 3D object features, or in 2D [32, 55], 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 is gathered from carefully calibrated cameras, the 3D position can typically be estimated quite accurately. Obtaining *accurate* position estimates, however, is not the main challenge of the reconstruction-tracking scheme; instead, the main challenge is the correct interpretation of *ambiguous* position estimates, which might be caused by incorrect across-view correspondences. Such ambiguity becomes significantly worse when correspondences need to be established for tracking dense crowds of objects.

The complexity of the multi-view tracking algorithm is also determined by the motion pattern of the objects. Most of the previous multi-view methods for pedestrians tracking adopt a planar motion assumption and use the planar homography to simplify the across-view correspondence problem [59, 36, 50]. Occlusion can then be resolved even if the object is completely occluded in some views. But it cannot be applied to scenarios where the planar motion assumption does not apply. For objects moving in 3D space, we developed a 3D tracking mechanism that circumvents having to intepret occlusion, and a track-to-track scheme that combines data association information from each camera and corrects the track lost or track switch errors [89, 90].

Another interesting approach that explicitly models the occlusion process given accurate camera geometry information was proposed by Otsuka and Mukawa [62]. Silhouettes of objects were extracted and visual cones were constructed to represent a measurement. A variant of Multiple-Hypothesis-Tracking was adapted to predict when and how an occlusion event was going to happen. Obviously, such an approach is only applicable in highly controlled environments with sufficient coverage of overlapping fields of view from many different viewpoints.

Finally, there also exists work that addresses tracking objects in a camera network with non-overlapping fields of view. Establishing across-view correspondence in this context, also known as the re-identification problem, focuses on how to build a discriminant descriptor for objects and how to utilize the topology of the camera network for re-entry prediction [76, 49, 35]. As such a camera setup is not necessary to help inter-object occlusion reasoning, we refer readers to related literature and focus on cameras with overlapping fields of view in this thesis.

Relation to existing work. The objects in our videos move in free 3D space and are imaged with low resolution. This scenario is more general than scenarios involving planar motion, which have been studied in the computer vision literature extensively. Our reconstruction-tracking method follows the multidimensional assignment formulation for both the spatial and the temporal association problem. Based on multiview geometry, the cost function to evaluate each spatial data association hypothesis requires information from all views. This inevitably introduces a hard combinatorial problem. The formulation is further extended to handle merged measurements due to overlapped projections from multiple objects, and solved iteratively.

2.2 Reconstruction-Tracking Method

For now, we assume an appropriate detection method has been provided to return a set of measurements from each camera at each time step. We also assume the motion of each object in the scene can be well described by a linear dynamic system so that a Kalman filter can be applied for to estimate the state of a track. Therefore, occlusion reasoning relies on data association, both temporally (across-time) and spatially (across-views). Multiple cameras are deployed to share a large overlapping field of view to maximize the visibility of objects in all views. The basic idea is to collect observations/measurements from all cameras, reconstruct the 3D positions of objects by triangulation, and apply recursive Bayesian tracking in 3D space. We call such an approach "reconstruction-tracking."

Table 2.1: Notation for reconstruction-tracking method

y^t_{s,i_s}	the (i_s)-th observation/measurement at time t from camera s
$Y_{i_1 i_2 \ldots i_N}$	N measurements $y_{1,i_1}, y_{2,i_2}, \ldots, y_{N,i_N}$
$x_{i_1 i_2 \ldots i_N}$	a binary variable to associate measurements $Y_{i_1 i_2 \ldots i_N}$ to a unique object
$c_{i_1 i_2 \ldots i_N}$	the cost to associate measurements $Y_{i_1 i_2 \ldots i_N}$ to a unique object
$Z_{i_1 i_2 \ldots i_T}$	T 3D reconstructed measurements $z_{1,i_1}, z_{2,i_2}, \ldots, z_{T,i_T}$
A	state transition matrix for a linear dynamic system
H_s	observation matrix in camera s
$\mathbf{x}_a$	the state (position) vector of object a
P_{D_s}	detection rate in camera s
Φ_s	volume of field of view in camera s
u_s, v_s	measurement of image coordinates in camera s
F	set of all possible across view associations
M_c	set of confirmed associations without dummy measurements
M_s	set of suspicious associations with dummy measurement in each tuple

2.2.1 Multidimensional Assignment Formulation

In this section, we define the state $\mathbf{X}$ of an object of interest by its position $\mathbf{x}$ and velocity $d\mathbf{x}$ in 3D space. Its evolving process follows a constant velocity. The measurement returned by our detection method is a 2D point observation of an object on the image plane or a false alarm. Given N calibrated and synchronized cameras that share overlapping fields of view and n_s measurements in the field of view of camera s, the state $\mathbf{X}^{(t)}_a$ of an object of interest a at time t and its observations can be assumed to evolve in time according to the equations

$$\begin{cases} \mathbf{X}_a^{(t+1)} = A\mathbf{X}_a^{(t)} + v^{(t)}, \\ y_{s,i_s}^{(t)} = H_s\,\mathbf{x}_a^{(t)} + w_s^{(t)}, \quad \text{for } s = 1, ..., N,\ i_s = 1, ..., n_s; \end{cases} \tag{2.1}$$

where $v^{(t)}$ and $w_s^{(t)}$ are independent zero-mean Gaussian noise processes with respective covariances $Q(t)$ and $R_s(t)$, A is the state transition matrix with a constant velocity assumption, and H_s the projection matrix for camera s. Each point measurement $y_{s,i_s}^{(t)}$ is either the projection of some object a in camera s plus additive Gaussian noise $\mathcal{N}(0, R_s(t))$, or a false-positive detection, which is assumed to occur uniformly likely within the field of view of camera s.

In order to model missed detections, for each camera, we define the probability of an object being detected is $\mathrm{P}_{D_s} < 1$. We add "dummy" measurements $y_{s,0}^{(t)}$ to handle the case of missed detections, accordingly. In particular, when object a is not detected in camera s at time t, a dummy measurement $y_{s,0}^{(t)}$ from camera s is associated with object a.

We use the notation $Y_{i_1 i_2 \ldots i_N}$ to indicate that the measurements $y_{1,i_1}, y_{2,i_2}, \ldots, y_{N,i_N}$ originate from a common object in the scene at time t. For simplicity, we omit the time superscript for now. The likelihood that $Y_{i_1 i_2 \ldots i_N}$ describes object state $\mathbf{x}_a$ is given as

$$p(Y_{i_1 i_2 \ldots i_N}|\mathbf{x}_a) \;=\; \prod_{s=1}^{N}\{[1-\mathrm{P}_{D_s}]^{1-u(i_s)} \times\ [\mathrm{P}_{D_s}\, p(y_{s,i_s}|\mathbf{x}_a)]^{u(i_s)}\} \tag{2.2}$$

where $u(i_s)$ is an indicator function defined as 0 if $i_s = 0$ and 1 otherwise. The conditional probability density of a measurement y_{s,i_s} originating from object a, is

$$p(y_{s,i_s}|\mathbf{x}_a) = \mathcal{N}(y_{s,i_s}; H_s\,\mathbf{x}_a, R_s). \tag{2.3}$$

The likelihood that $Y_{i_1 i_2 \ldots i_N}$ is unrelated to object a or related to dummy object $\oslash$ is

$$p(Y_{i_1 i_2 \ldots i_N}|\oslash) = \prod_{s=1}^{N} [\frac{1}{\Phi_s}]^{u(i_s)}, \tag{2.4}$$

where Φ_s is the volume of the field of view of camera s. Since we do not know the true state

$\mathbf{x}_a$, we replace it with a least-square solution as follows. state $\hat{\mathbf{x}}_a$ to be the reconstructed 3D position based on the corresponding measurements $y_{1,i_1}, y_{2,i_2}, ..., y_{N,i_N}$ in the N views. If we assume each measurement y_{s,i_s} is expressed as image coordinates (u_s, v_s) and the state of the object in 3D is expressed as a homogeneous coordinate $\mathbf{x} = (x, y, z, 1)^T$, then for each measurement there are two linear constraints:

$$\begin{cases} u_s(H_s^{(3)}\mathbf{x}) - H_s^{(1)} = 0 \\ v_s(H_s^{(3)}\mathbf{x}) - H_s^{(2)} = 0 \end{cases} \tag{2.5}$$

where $H_s^{(i)}$ is the i-th row of matrix H_s. To find $\hat{\mathbf{x}}_a$, given the measurements from N views, the Direct Linear Transformation (DLT) method [46] solves the overdetermined linear system in Eq.2.5 with $2N$ constraints.

We now can define the cost of associating N-tuple $Y_{i_1 i_2 \ldots i_N}$ to object a as the negative log-likelihood ratio

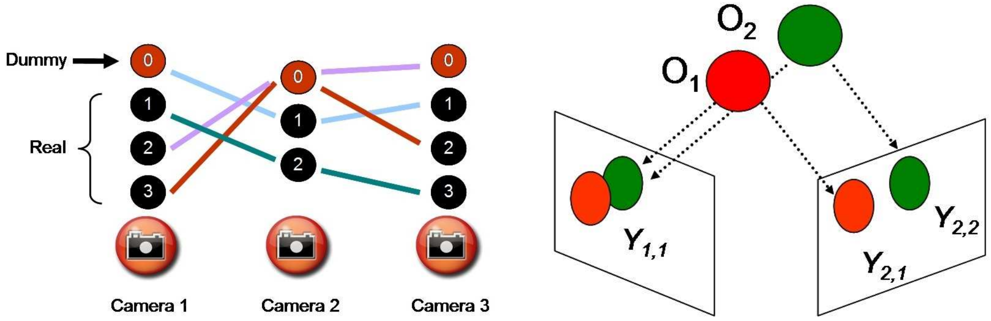
$$\begin{aligned} c_{i_1 i_2 \ldots i_N} &= -\ln \frac{p\,(Y_{i_1 i_2 \ldots i_N} |\, a)}{p\,(Y^t_{i_1 i_2 \ldots i_N} |\, \oslash)} \\ &= \sum_{s=1}^{N} \{[u(i_s) - 1] \ln(1 - \mathrm{P}_{D_s}) - u(i_s) \ln \left(\frac{\mathrm{P}_{D_s} \Phi_s}{|2\pi R_s|^{1/2}} \right) \\ &\qquad + u(i_s)[\frac{1}{2}(y_{s,i_s} - H_s \hat{\mathbf{x}}_a)^T R_s^{-1} (y_{s,i_s} - H_s \hat{\mathbf{x}}_a)]\} \end{aligned} \tag{2.6}$$

We use the binary variable $x_{i_1 i_2 \ldots i_N}$ to indicate if $Y_{i_1 i_2 \ldots i_N}$ is associated with a candidate object or not. Assuming that such associations are independent, our goal is to find the most likely set of N-tuples that minimizes the linear cost function:

$$\begin{aligned} \min \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \dots \sum_{i_N=0}^{n_N} c_{i_1 i_2 \dots i_N}\, x_{i_1 i_2 \dots i_N} \qquad (2.7) \\ \text{s. t.} \sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} \dots \sum_{i_N=0}^{n_N} x_{i_1 i_2 \dots i_N} = 1; \quad i_1 = 1, 2, \dots, n_1 \\ \sum_{i_1=0}^{n_1} \sum_{i_3=0}^{n_3} \dots \sum_{i_N=0}^{n_N} x_{i_1 i_2 \dots i_N} = 1; \quad i_2 = 1, 2, \dots, n_2 \\ \vdots \\ \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \dots \sum_{i_{N-1}=0}^{n_{N-1}} x_{i_1 i_2 \dots i_N} = 1; \quad i_N = 1, 2, \dots, n_N. \end{aligned}$$

The above cost function has been proposed in the radar tracking literature [67, 30]. The equality constraints imply every detection has to be explained and the matching is one-to-one between real measurements. Each measurement is either assigned to some object or claimed to be a false-positive detection. However, due to occlusion, multiple objects might share the same projection, i.e., a centroid point taken from the merged "object blobs," as shown in Fig. 2·1. We therefore have to allow a real but merged measurement to be matched more than once. In another words, we need to identify possible occluded objects and relax the one-to-one matching constraint for those objects.

Eq. 2.7 is known as a generalized multidimensional assignment problem, which is NP-hard when the dimension $N \geq 3$. The processing time for the optimal solution is unacceptable in dense tracking scenarios, even if a branch-and-bound search method is used, because such a method is inevitably enumerative in nature. The alternative is to search for a sub-optimal solution to this combinatorial problem, using greedy approaches [71], Lagrangian relaxation [67, 30], simulated annealing or tabu search. We propose an iterative greedy randomized adaptive search procedure (IGRASP), which randomly picks a greedy solution as a starting point and performs local search in feasible solution space.



(a) (b)

Figure 2·1: (a) Each detection is either matched to a real object, or declared to be a false alarm. If a detection is missed for an object, a dummy measurement is matched instead. (b) From a single view, two objects o_1 and o_2 occlude each other and yield a single measurement $y_{1,1}$. A single-view tracker may lose track of one of the objects. If two views are available, the objects o_1 and o_2 can be matched to their respective measurements $y_{2,1}$ and $y_{2,2}$. Stereoscopic reasoning reveals that $y_{1,1}$ is the image of both objects. Therefore, the real measurement $y_{1,1}$ should be matched more than once.

2.2.2 Iterative Greedy Randomized Adaptive Search Procedure

We first briefly outline the generic Greedy Randomized Adaptive Search Procedure (GRASP), as we applied it to the multidimensional assignment problem of Eq. 2.7. This required adjusting the procedure to our multi-view scenario. GRASP is a multi-start local search method with random initialization [71]. It consists of a randomized greedy step and a local search step at each iteration. In the randomized greedy step, a restricted candidate list is constructed greedily from the remaining feasible assignments, from which an assignment is selected randomly and added to the solution set. In the local search step, we adopt the so-called 2-assignment-exchange operation between real measurement assignments. That is, for two tuples $Z_{i_1 \ldots i_j \ldots i_N}$ and $Z_{i'_1 \ldots i'_j \ldots i'_N}$ from the feasible solution, we exchange the assignment to $Z_{i_1 \ldots i'_j \ldots i_N}$ and $Z_{i'_1 \ldots i_j \ldots i'_N}$ if such an operation decreases the total cost in Eq. 2.7. The tuples and their indices to exchange are selected to be the most profitable pair at the current iteration. The exchange takes place recursively until no exchange can be made anymore. Details of the GRASP implementation and other possible greedy constructions and assignment exchange strategies can be found in the work by Robertson [71].

We adopt a technique similar to "gating" during the initialization step to reduce the number of possible candidate tuples as follows. Given a pair of calibrated views, our technique establishes the correspondence of the two projected images of an object using epipolar geometry. Thus, we only need to evaluate the candidate tuples that lie within the neighborhood of corresponding epipolar lines. Specifically, all candidate points from the second view that can be matched to a 2D point y (expressed in homogeneous coordinates) in the first view should be on the epipolar line computed by Fy, where F is the fundamental matrix that captures the geometric relationship between two cameras [46]. A user-defined threshold is adopted to prune candidate points that are far away from this line so the total possible number of pairings can be reduced significantly. This pruning step in building the multidimensional assignment problem, which we call epipolar-neighborhood search, becomes crucial for the overall efficiency of our method.

```
GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE:
Compute the costs for all possible associations and prune the candidates by an epipolar-neighborhood search
For $i = 1, ..., maxIter$
    1. Randomly construct a feasible greedy solution,
    2. Recursively improve the feasible solution by a local search,
    3. Update the best solution by comparing the total costs,
End
Output the best solution found so far.
```

To relax the one-to-one matching constraint, measurements that overlap due to occlusion or imperfect segmentation during the detection stage and thus are interpreted as a single measurement (centroid of merged "object blobs"), can be assigned to multiple objects, as shown in Fig. 2·1(b). We extend the generic GRASP algorithm to an iterative process, where at each iteration, an updated multidimensional assignment problem is solved that involves measurement previously identified as false alarms. One toy example to demonstrate such an iterative procedure is given in Fig. 2·2.

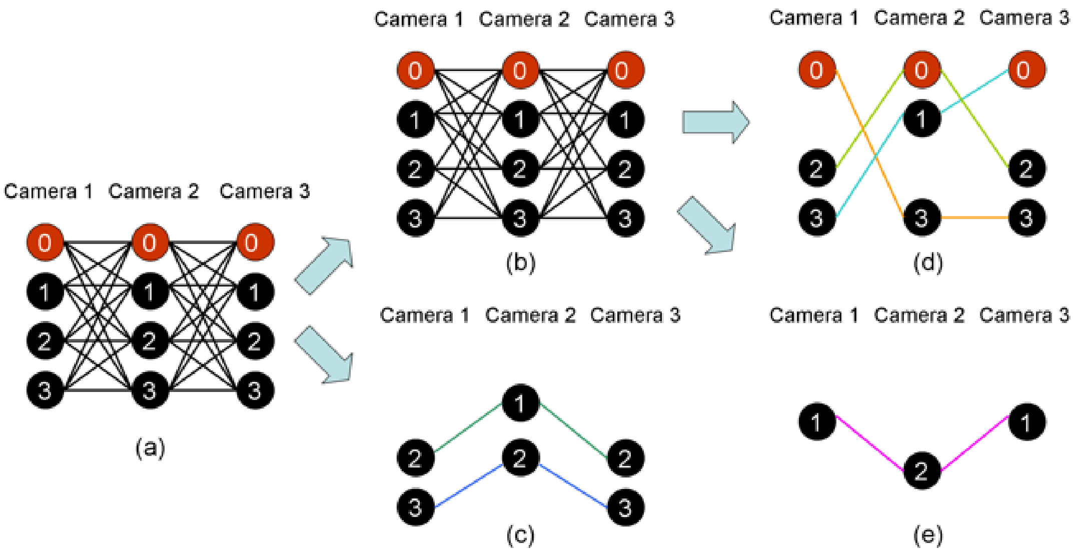


Figure 2·2: The greedy solution for a multi-view tracking example with 3 views, each of which receives three measurements. (a) A 3-partite graph corresponding to Eqn. 2.7, where each hyperedge is a possible association tuple. (b) The residual graph with two confirmed associations extracted in (c) after the first iteration of IGRASP. (d) The suspicious associations after solving the multidimensional assignment problem corresponding to the residual graph (b), with (e) as a new confirmed association after the second iteration of IGRASP. If no further confirmed association can be generated from the residual graph with respect to (c) and (e), the final greedy solution to the original problem (a) is the union of (c), (d) and (e).

We denote the set of all possible N-tuples as $F = Z_1 \times ... \times Z_s \times ... \times Z_N$, where Z_s is the set of all the measurements in view s plus the "dummy" measurement. Solving Eq. 2.7 yields a set of possibly suboptimal assignments Z^*, where a specific assignment in this solution can be expressed as $\{Z_{i_1 i_2 \ldots i_N} | x_{i_1 i_2 \ldots i_N} = 1\}$. We divide the set of assignments into two subsets:

1. **Confirmed associations:**

   $M_c = \{Z_{i_1 i_2 \ldots i_N} | x_{i_1 i_2 \ldots i_N} = 1; i_1 \neq 0; \ldots; i_N \neq 0\}$.

2. **Suspicious associations:** $M_s = Z^* \setminus M_c$.

Suspicious associations contain dummy measurements $z_{s,0}$ that indicate an object is not detected in some view and measurements associated with the dummy measurement are false positive detections. Thus, associations in set M_s have at least one zero-index in their subscripts. From set M_s, we construct another assignment problem that is described by Eq. 2.7, except with the already confirmed assignments in M_c removed from the feasible assignment set F. Occluded objects then get a second chance to match the measurements, especially aiming for possible merged measurements. In addition, costs for candidate association tuples without a zero-index measurement are increased by a scaling factor so that it becomes more and more difficult to generate confirmed associations as the algorithm iterates. Now the algorithm can generate another two subsets from the result and iterate until a maximum number of iterations is reached or M_c in the current iteration is empty. We summarize the Iterative GRASP in the pseudocode below.

```
ITERATIVE GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE (IGRASP):
Building Phase
Initialization by computing the costs for all possible associations in set $F$.

Solving Phase
For $i = 1, ..., maxIter$
    1. Formulate multidimensional assignment problem on set $F$ according to Eqn. 2.7, where cost coefficients for tuples without a zero-index measurement are increased by a scaling factor $\gamma > 1$.
    2. Run standard GRASP to obtain a suboptimal solution.
    3. Partition the computed solution into confirmed set $M_c$ and suspicious set $M_s$.
    4. If    Set $M_c$ is empty,    terminate; else    $F = F \setminus M_c$
End
Output the best solution found so far.
```

2.2.3 Reconstruction-tracking Algorithm

Thus far we described a method to solve multi-view data association in a single time step. The solution allows us to estimate the current 3D position of each object in the scene using Eq. 2.5, which estimates the 3D position in a least-squares sense [46]. Once the 3D locations of objects are reconstructed, similarly to Eqn. 2.7, the problem of temporal data association can also be formulated as a multidimensional assignment problem, as shown in Fig. 2·3. The state definition of each object remains the same as described earlier, but the measurement is taken as the reconstructed 3D point. The cost for each matching hypothesis is taken as the negative log-likelihood evaluated by Kalman filtering. We refer to the work by Poore [66] for the detailed derivation of the function describing the cost of a hypothesis and the objective function, which sums these costs.

The tracking algorithm is implemented with a sliding-window scheme. At each time step t, a new $(T+1)$-dimensional assignment problem is formulated with the set of established tracks at time $t-1$ and T sets of new measurements up to time $t+T-1$. Each established track carries its estimated state and noise covariance at the end, which will be used to initialize the Kalman filter that evaluates a particular matching hypothesis. Once the assignment problem is solved, the tracks are extended to time t and their state vectors and covariance matrices are updated with Kalman smoothing. To complete the steps of track initiation, continuation, and termination, we outline our reconstruction-tracking algorithm, which forms our baseline algorithm for multi-object multi-view sequential tracking, as follows.

RECONSTRUCTION-TRACKING ALGORITHM

Tracking with deferred logic. At each time step t:

Input: A set of measurements $\{y^{(t)}_{s,i_s}\}$ from N cameras with T frames, and M established tracks from time $t-1$:

1. For each of T frames, reconstruct 3D positions of objects by solving a generalized N-dimensional assignment problem according to Eqn. 2.7.
2. Combine T frames of reconstructed measurements and M active tracks to a $(T+1)$-dimensional assignment problem [66] and solve it. The solution gives a set of tracklets of length $(T+1)$.
3.
    - If a tracklet's head is from one of the M established tracks, extend it with the tracklet.
    - If a tracklet's head is a dummy measurement, initialize a new track with this tracklet.
    - If an established track does not have its extension in tracklets, it is a lost track. Track coasting technique is applied.

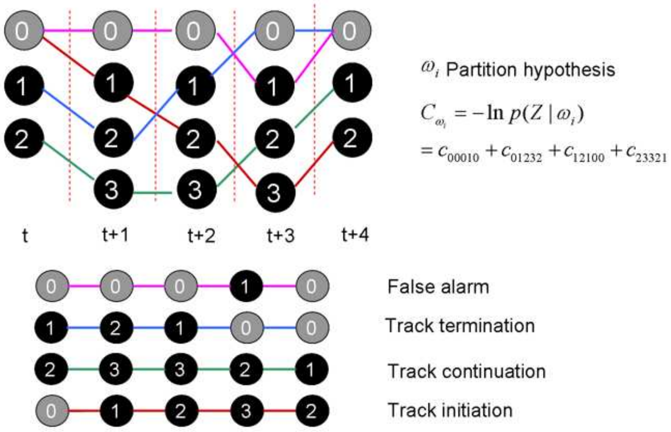

ω_i Partition hypothesis
$C_{\omega_i} = -\ln p(Z \mid \omega_i)$
$= c_{00010} + c_{01232} + c_{12100} + c_{23321}$
t
t+1
t+2
t+3
t+4
False alarm
Track termination
Track continuation
Track initiation

Figure 2·3: Example of solving the multidimensional assignment problem in the temporal domain. Current active tracks are listed in the first column of the sliding window. Track initialization, continuation, and termination are implemented by checking the position of the zero-index dummy measurement in the solution.

2.3 Experiments

In this section, we first describe two datasets collected for understanding the behavior of flying animals, which require both tracking and reconstruction techniques. Then we give a quantitative analysis of our reconstruction-tracking approach applied to two fully-annotated infrared video sequences.

2.3.1 Data Collection

Observing the flight behavior of large groups of bats or birds is fascinating – their fast, collective movements provide some of the most impressive displays of nature. Quantitative studies of cooperative animal behavior have typically been limited to sparse groups of only a few individuals. The limitations of these studies are mainly due to the lack of tools to obtain accurate 3D positions of individuals in dense formations. Although important progress has been made recently [9], a robust solution to 3D tracking, reconstruction, and data association still needs to be developed. Thus, our automatic multi-object multi-view tracker is expected to have great impact in related fields by providing thousands of trajectories for group behavior studies. In this thesis, videos of two different species of flying animals, barn swallows and Brazilian free-tailed Bats, were collected to test our multi-object multi-view tracking approach.

A recording of swallows in visible-light video was provided by Prof. Ty Hedrick, University of North Carolina at Chapel Hill, which contains 475 frames for each of three cameras. The average distance between swallows and cameras is around 50 meters. The sequence is relatively easy to analyze because object density is low. Point measurements are obtained by background subtraction and by selecting the centroid points from connected foreground components. Our tracker can produce high-quality trajectories without difficulty in finding the right data associations both across view and across time. A qualitative result with sample frames is shown in Fig. 2·4.

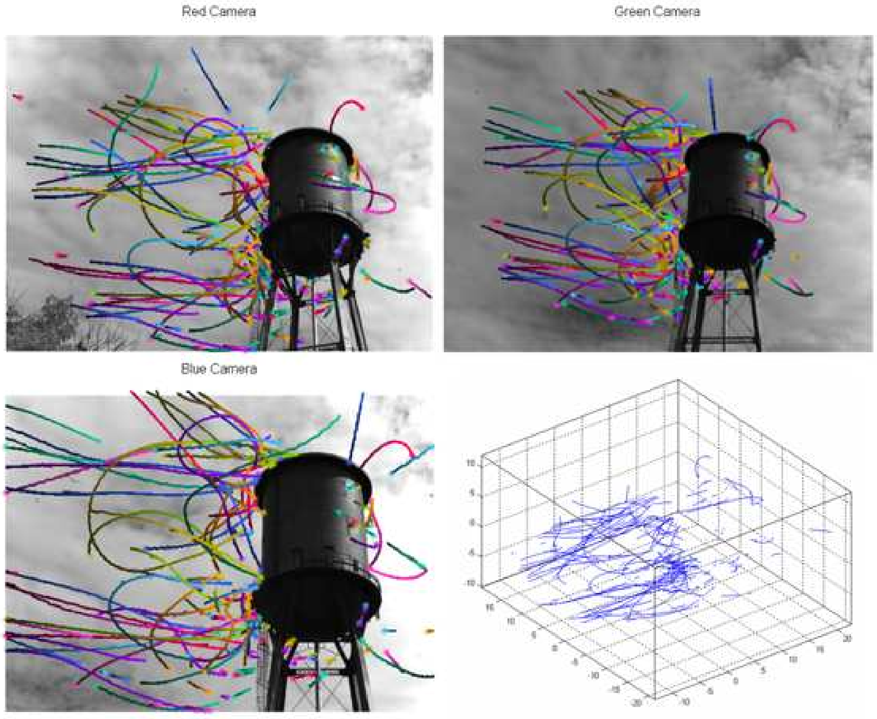


Figure 2·4: Stereoscopy reconstructed 3D flight paths of swallows and the three camera views of the sequence overlayed with the trajectories backprojected onto each image plane. Corresponding paths across views are shown in the same color. The brightness is proportional to the depth in the scene.

We also recorded the emergence of a colony of Brazilian free-tailed bats from a natural cave in Blanco County, Texas. We used three FLIR SC6000 thermal infrared cameras with a resolution of 640 × 512 pixels at a frame rate of 125 Hz, as shown in Fig. 2·5. The cameras were placed at a distance around 15 meters from the cave in order to capture the entire group of flying bats from different viewpoints with overlapping fields of view. All cameras were synchronized and spatially calibrated with a large baseline.

We do not have sufficient appearance information to distinguish between bats or swallows, which look very similar to each other. The size of the projection of each target ranges from 10 to 40 pixels, depending on the distance of the target to the camera. In addition to qualitative evaluation of our tracking system on the swallow sequence, we also established ground truth by manually labeling two subsets (Infrared S1, S2) of different densities from infrared bats video, which includes about 30 and 100 bats per frame, respectively. The first subset with low density comprises of 1,100 frames for each view, while the second one comprises of 200 frames.

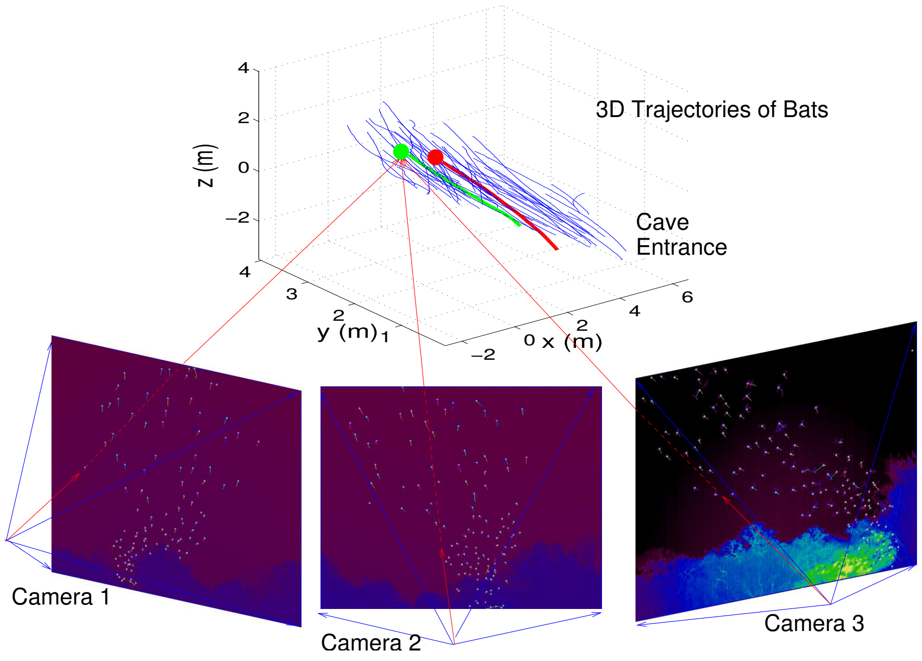


Figure 2·5: The emergence of Brazilian free-tailed bats. Hundreds of bats were automatically tracked and the trajectories were reconstructed.

2.3.2 Quantitative Evaluation on Infrared Video Datasets

Two versions of our "Reconstruction-tracking" algorithm were implemented, which we denote as "RT-1" and "RT-2." These two differ only in the cost function to evaluate the likelihood of a given *temporal* data association hypothesis. For "RT-1," the cost function is defined by the negative likelihood ratio through Kalman filtering [22][1], where the measurements are 3D locations of reconstructed points after solving the *spatial* data association problem. For "RT-2," the cost function is the same except the measurements are 2D locations of the detections on the image planes in all views. As the accuracy of 3D reconstruction depends on the quality of camera calibration as well as the distance between targets and cameras, it is possible that the 3D reconstruction could be off by meters in the physical world even if the right spatial data association is found. Therefore, "RT-2" circumvents the need to have accurate triangulation in stereoscopy. On the other hand, as "RT-2" needs to work with 2D measurements directly, it is sensitive to the detection quality, especially when multiple objects occlude each other and yield an overlapped measurement.

Important Parameter Settings. To initialize the Kalman filter of a newly appearing object, the initial state of an object is taken as the measurement in the current frame (position) and the displacement between measurements from the first two frames it appears in (velocity). The covariance matrices are initialized as identity matrices. To initialize the tracking process of a tracked object at the first frame of a sliding window, which consists of 5 consecutive frames, state and covariance parameters are set based on the estimates carried at the end of its track in the previous instantiation of the sliding window.

The parameter that defines the "gate" in spatial data association is set to be 20 pixels. It is the maximum distance allowed from a given point to its epipolar line. Larger threshold settings are disadvantageous because they would introduce additional candidate

---

[1] We use the toolbox by Murphy K. `http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html`

association hypotheses. Note that evaluating the cost of each hypothesis is the bottleneck of the whole system. The set of all hypotheses could also be separated into disjoint subsets by clustering before optimization, as suggested by Cox and Hingorani [29]. The maximum number of missed detections allowed is also a critical parameter to determine the problem size, It is set to 1 for spatial data association in three views and 2 for temporal data association throughout our experiments. For the IGRASP algorithm, the maximum number of iterations is set to 10 with a scaling factor $\gamma = 1.05$ (see Sec. 2.2.2). These two parameters should be adjusted when the density of objects varies. Additional iterations and a lower scaling factor would be advantageous if the object density is higher than present in our dataset.

Quantitative Evaluation Metric.

For quantitative evaluation, we use the "USC metrics" by Wu [86] and the "CLEAR MOT" metrics by Bernadin and Stiefelhagen [14]. Because we use these metrics throughout this thesis, we here briefly explain how they are computed.

Given a set of system-generated tracks S and a set of ground-truth tracks G, a list of possible matches is constructed at each time step t, where a possible match pair (s, g) is determined if the matching cost between the two is above a hit/miss threshold. In this chapter, we use the Euclidean distance as the matching cost. Once such a list is constructed, an assignment problem is solved to find the optimal one-to-one matches. The number of matched pairs in the solution is denoted as c_t. The distance between each matched pair is denoted as d_t^i. The number of system-generated tracks that are not matched (false positives) is fp_t; the number of ground-truth tracks present in the current frame is g_t and the number of ground-truth tracks that are not matched (miss) is m_t. The number of system-generated tracks that are matched to different ground-truth tracks compared to the matches made at previous time step (mismatch or ID switch) is mme_t. Given these quantities for all the frames, the CLEAR MOT metrics that include Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) [14] are computed as follows:

- **Miss Rate (MR)**: $\frac{\sum_t m_t}{\sum_t g_t}$;
- **False Positive Rate (FPR)**: $\frac{\sum_t fp_t}{\sum_t g_t}$;
- **Mismatch Rate (MMR)**: $\frac{\sum_t mme_t}{\sum_t g_t}$;
- **Multiple Object Tracking Accuracy (MOTA)**: MOTA takes into account false positives, missed targets, and identity mismatches. The final score to summarize tracking accuracy is computed as 1-MR-FPR-MMR.
- **Multiple Object Tracking Precision (MOTP)**: $\frac{\sum_{t,i} d_t^i}{\sum_t c_t}$

The USC metrics [86] are computed as:

- **Mostly Tracked (MT)**: the number of objects for which $\geq 80\%$ of the trajectory is tracked, i.e., 80% of a ground-truth track has been matched to some non-empty set of system-generated tracks;
- **Mostly Lost (ML)**: the number of objects for which $\leq 20\%$ of their trajectories is tracked;
- **ID Switch (IDS)**: the number of identity switches $\sum_t mme_t$.

In order to compute a match between ground-truth trajectories and system-generated trajectories, 0.3 m is chosen as the miss/hit threshold for the infrared data of bats. This threshold is close to the physical size of this species when the wings are extended. In addition to MOTA, we compute the average Euclidean distances in 3D between two sets of trajectories for MOTP that measures the average precision.

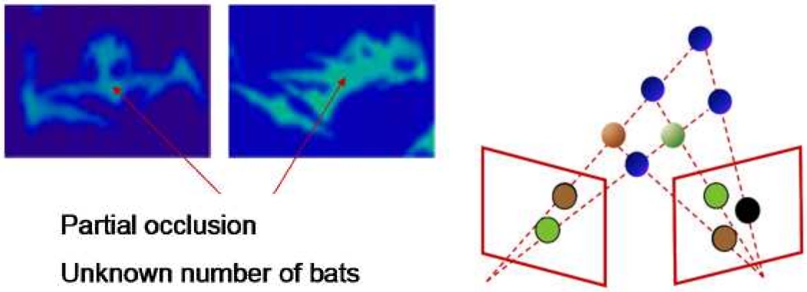
Table 2.2 gives the quantitative evaluation of the proposed two versions of the reconstruction-tracking algorithm. Both algorithms work reasonably well for the sequence of low object density. But the performance drops catastrophically when dealing with the extremely dense scenario. Between the two versions of the reconstruction-tracking algorithm, “RT-1”

Data	Method	GT	MT	ML	IDS	MOTA	MOTP
Infrared S1	RT-1	207	200	0	35	0.65	8.5 cm
(1100 frames)	RT-2	207	195	0	72	0.65	9.0 cm
Infrared S2	RT-1	203	147	5	158	-0.31	10.1 cm
(200 frames)	RT-2	203	152	2	609	-0.40	10.9 cm

Table 2.2: Quantitative results of our reconstruction-tracking algorithm on Bats dataset. GT:Ground Truth; MT: Mostly Tracked; ML: Mostly Lost; IDS: ID Switch.

clearly has superior performance, which suggests that tracking in 3D is much more reliable than 2D as long as the reconstruction is accurate enough. As the occlusion introduces uncertainty on 2D measurements, "RT-2" that works directly with merged measurements in 2D is more sensitive to the frequency of occlusion, which results in a high ID switch error rate.

The negative MOTA scores are caused by incorrect spatial data associations that occurred in the first step of the reconstruction-tracking algorithm (note that the false positive rate defined in the CLEAR metric is **not** bounded to be at most one). There are mainly two issues to be addressed. First, although a point representation is good enough for the objects in our experiment, an extended measurement should be considered when the projections of multiple objects yield a single merged blob, as shown in Fig. 2·6 (a). A better detection method should extract the right number of points and accurate positions of these points from the merged measurement. Second, even if the 2D measurement is accurate, a "ghost effect" might show up during the triangulation step, i.e., multiple hypotheses in 3D locations would generate the same 2D measurements on the image planes. Such ambiguity cannot be resolved purely from the knowledge of camera geometry. Therefore, additional constraints should be added in order to suppress these errors and reduce the false positive detection rate. We will revisit this issue in Chapter 4.



(a) (b)

Figure 2·6: Two sources of error for spatial data association. (a) Uncertainty in the merged measurement. Each connected component contains an unknown number of objects, and the optimal point to represent to each object's location is not clear. (b) Ghost effect created by triangulation. All blue points in the figure perfectly match camera geometry, but they are all false alarms.

2.4 Summary and Discussion

In this chapter, we propose a sequential spatial-temporal data association method for multi-view multi-object tracking. Occlusion could be resolved by solving spatial (across-view) association and occluded objects can be localized in 3D through stereoscopy. In particular, we adapt the traditional multidimensional assignment formulation, a variant of the Multiple-Hypothesis-Tracking (MHT) algorithm, to our spatial data association task. In order to allow many-to-one matching for merged measurements due to inter-object occlusion, we propose an iterative greedy algorithm (IGRASP) to identify those potentially merged measurements and recover occluded objects. Once the 3D locations of objects are reconstructed, a variant of MHT is applied again to perform temporal data association as well as maintain track initialization, continuation, and termination.

We compare the proposed method with two different implementations (RT-1 and RT-2) and test on visible-light videos of swallows and infrared videos of bats, where objects with small resolution are moving in free 3D space. Our tracking algorithm is able to track most objects in sparse or median densities and produce 3D trajectories for further data analysis. However, quantitative results suggest that such algorithms work poorly on a dense sequence where the benefit of multi-view geometry reaches its limit. The “ghost effect” is introduced during the reconstruction step where multiple hypotheses perfectly satisfy camera geometry constraints and therefore cannot be distinguished from each other. Such phenomenon could be eliminated through tracking if it only happens sporadically. Unfortunately, in our challenging infrared video data of bats, the phenomenon exists persistently and cannot be resolved purely through the data association step. We will revisit this problem in Chapter 4.

Chapter 3

Track Linking on Track Graph

In this chapter, we adopt a batch processing method where we treat objects involved in the occlusion event as a single target to track, known as “track linking.” It is a generalization of traditional measurement-to-measurement association: here, the matching involves trajectory segments (tracklets). Each tracklet typically carries much more information than the measurements considered in the previous chapter (e.g., centroid positions). Occlusion ambiguity is resolved by optimizing a cost function that considers the smoothness of object motion and appearance over several frames. With this approach, tracklets may be stitched together and full trajectories may be recovered. This idea can be applied to both single-view and multi-view settings.

We first review classic tracklet stitching techniques in Sec. 3.1. Our detailed track linking approach is explained in Sec. 3.2 with supporting experiments in Sec. 3.3. We conclude this chapter in Sec. 3.4 by discussing the strength and limitation of the proposed approach.

3.1 Related Work

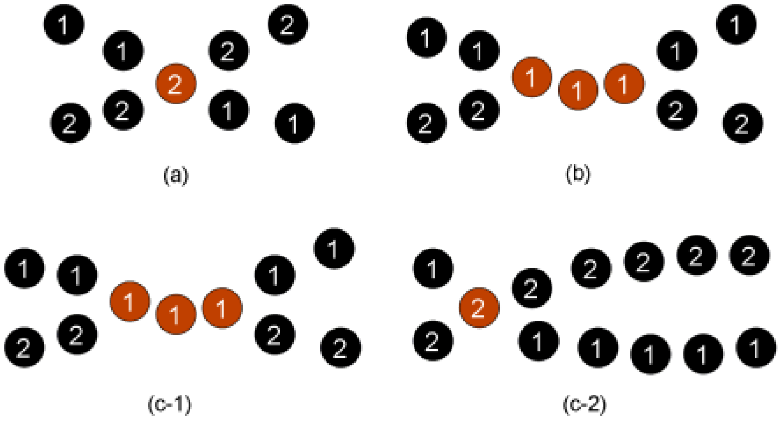
Most of data association works described in the previous chapter use a instantaneous measurement as the matching unit. Track linking, as a batch process, is a generalization of instantaneous measurement-to-measurement association: here, the matching involves trajectory segments or “tracklets,” which are typically generated by a low-level tracker. The advantages of using tracklets are twofold. First, the complexity of most data association methods usually grows quickly when many frames are processed in a batch mode. By matching tracklets, especially long tracklets, the time span of the sequence in a batch that a system can handle efficiently typically significantly increases. Second, each tracklet already carries filtered information and, therefore, the descriptor for each tracklet is much more informative than a simple instantaneous measurement can be [68, 8].

Static scene occlusions or inter-object occlusions are the main causes that break a complete trajectory into pieces. In order to stitch pieces that occur before and after occlusion events, a common assumption is adopted in track linking that a complete track should obey certain smoothness properties, either in its appearance or motion. Most existing techniques that work with tracklets simply extend a measurement-to-measurement association method by redesigning the similarity function under the same mathematical framework, such as the 2D assignment problem [47, 63], MCMC sampling [43] or network-flow optimization [25].

Instead of organizing temporal data-association hierarchically, where, at each level, *local* links between track fragments are produced [47, 63, 92], Nillius et al. [60] solved the problem *globally* by processing the track graph that represented all object interactions. Their method used the “junction-tree algorithm” for loopy graph inference to maintain track identities. Unfortunately, the size of the state space defined for each node in the graph that models object interaction grows exponentially as the number of objects involved in the interaction increases. Since the state space, i.e., the permutation space over the object identities, is large, their method has to incorporate some heuristics to make it practical, especially when objects interaction is frequent.

Track linking also plays an important role in medical applications, such as cell analysis in time-lapse microscopy [54]. Due to frequent interactions, highly nonrigid deformations and cluttered background, it is not easy to develop a robust low-level tracker in these applications. An additional linking procedure has to be performed using the spatial-temporal context. An interesting problem under consideration here is how to identify mitosis events in a low-frame-rate video where objects undergo splitting as a physical process.

Relation to existing work. Usually a track linking method needs to compare features



(a) (b)

(c-1) (c-2)

Figure 3·1: Three different inter-object occlusion scenarios: (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. Short-term occlusion (a) is usually easy to resolve if objects have distinctive motion patterns. Long-term occlusion (b) is more difficult to explain since motion information about the objects (i.e. linear dynamics) typically only characterizes them for a short time period. If multiple views are available (here two), a long-term occlusion in one view (c-1) may be resolved by analyzing the status of the objects in another view where the occlusion does not occur or only occurs for a short time (c-2). Throughout this chapter, we do not assume objects are significantly distinctive in appearance or motion characteristics. Such an assumption would simplify the problem of occlusion reasoning, but cannot be made for our data.

extracted from tracklets to decide if a stitch should be made. The feature is *local* if it only represents the information carried within the tracklet under consideration. The feature can also be *global* if it depends on the whole trajectory formed by **all** the tracklets along the path. Most previous track linking methods use local features only. We will show that a global feature is more appropriate if the occlusion process is complicated. Previous efforts can also be categorized according to their stitching strategy which either follows a non-iterative or an iterative process. For a typical iterative process, tracklets are linked as a pair at each iteration and the complete path is formed incrementally [47, 63, 92, 54]. For a non-iterative process, a global optimization problem is formulated, whose solution provides all the paths at the same time [60, 25]. The choice of the linking strategy depends on the characteristics of the occlusion events, as shown in Fig 3·1.

In this chapter, we employ both iterative and non-iterative linking strategies to handle different types of occlusion events. All these linking processes are based on a graph representation, and we propose a simple forward-backward algorithm to create such a graph. Furthermore, we also introduce a new strategy for linking tracklets that involves matches across camera views. The strategy can be seen as a "track-to-track" fusion scheme, a complementary method for multi-view multi-object tracking described in the previous chapter. Finally, we justify all these linking methods as performing maximum-likelihood estimation. A summary of related work and our methods is given in Table 3.1.

Table 3.1: Summary of related work and proposed track linking methods

Method	Feature	Strategy	Track Merge/Split
Huang et al [47]	local	iterative	no
Li et al [54] Xing et al [92] Perera et al [63]	local	iterative	yes
G. Castañón and L. Finn [25]	local	non-iterative	no
Nillius et al [60]	local	non-iterative	yes
Our local linking Minimum-flow+Bipartite matching	local	iterative	yes
Our network linking Minimum-cost flow	local	non-iterative	yes
Our global linking Weighted set-cover	global	non-iterative	yes
Our multiview linking Weighted set-cover	global	non-iterative	yes

3.2 Track Linking Methods

In this section, we present several linking strategies with the same underlying data representation, which we call a "track graph." We first describe the construction of such a graph with a forward-backward tracking scheme, and then develop four linking methods according to the characteristics of the inter-object occlusion events.

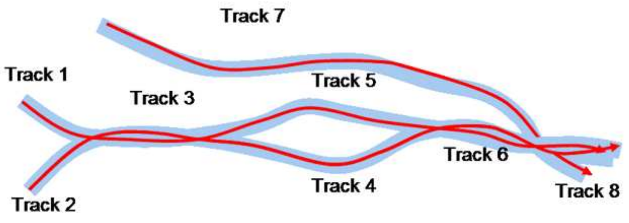
Table 3.2: Notation for track linking method

$\mathbf{Z}$	the collection of all tracklets
$\mathbf{X}$	the states of all objects
$\mathbf{M}$	the mapping matrices for edges on bipartite graph
$\mathbf{G}$	track graph
$\mathcal{T}_i$	(i)-th tracklet produced by low-level tracker
v_i	(i)-th vertex in track graph that corresponds to a tracklet $\mathcal{T}_i$
$e_{i,j}$	edge in track graph that shows a link between $\mathcal{T}_i$ and $\mathcal{T}_j$
$f_{i,j}$	the flow variable for $e_{i,j}$ to represent the number of interacting objects
$c_{i,j}$	the cost associated with $e_{i,j}$ to measure the likelihood of linking v_i and v_j
H	hypothesis of merging or splitting event
p	path in the track graph
x_p	the integer variable to represent path p is selected x_p times

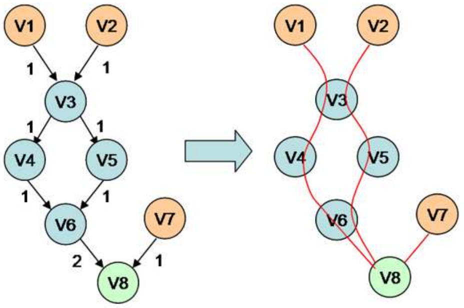
3.2.1 Track Graph Representation

A track graph $\mathbf{G} = (V, E)$ is defined over sets of vertices V that represent individual or merged tracks and edges E that represent merging or splitting events. A merged track is produced when multiple objects are treated as a single object due to either a close interaction between objects or an overlapped projection of moving objects in 3D space. The directed edge $e_{i,j}$ from vertex v_i to v_j represents that track v_i is merged with track v_j if v_j is a merged track, or that v_i is split to track v_j if v_i is a merged track, as shown in Fig. 3·2.

For simplicity, we assume each individual track is part of a complete trajectory corresponding to a true object, but the number of objects is unknown. The *flow* on the edge indicates how many objects are involved during the merging or splitting event. The vertex that has only incoming edges is called *sink*; the vertex that has only outgoing edges is called *source.* The set of all source vertices is denoted by S, and the set of all sink vertices by T. Each vertex has its *track-capacity* to represent single or multiple objects. For a source vertex, its associated track-capacity is the sum of outgoing flows; for a sink vertex, its associated track-capacity is the sum of incoming flows; for other intermediate vertices, the sum of incoming flows is equal to the sum of outgoing flows for balance. For tracking in a single view, an isolated vertex that has no incoming or outgoing edges has capacity



(a)



(b)

Figure 3·2: A tracking example that consists of three interacting objects and eight system-generated tracks (a) and the corresponding track graph (b). The track graph represents two objects that occlude each other for a while, then move apart, then merge again, and finally interact with a third object. The track graph is particularly useful to visualize such frequent track-merging and track-splitting events. Our local and global linking algorithms process the track graph (b-left) and produce the resolved graph (b-right), where each red arrow connects multiple vertices (i.e., tracks) and maintains the identity of the tracked object.

one. We remove these isolated vertices in preprocessing, as they do not require occlusion reasoning.

3.2.2 Algorithm to Construct Track Graph

The algorithm first processes the sequence forward in time to generate basic tracks and merge hypotheses. It then goes backward to break some tracks when necessary, and generate split hypotheses. Finally it defines the vertices and edges of the track graph. Here we use a Kalman filter to produce the tracklets where the state of an object is described by its 2D location and velocity. The same definition of the state vector is also used to describe multiple objects in a group if they are interacting with each other.

1. **Tracking Forward:** A new tracker is initiated when a measurement cannot be associated with an existing tracker. Each existing tracker chooses the measurement nearest to its position estimate, which is predicted by a Kalman filter, as its current observation. If a measurement is determined to be associated with multiple trackers, each of these trackers terminates itself, and a new tracker is initiated for this measurement. Meanwhile, a track-merge hypothesis H_m is generated and added to the list of hypotheses. An existing tracker also terminates itself if it is not associated with any measurement for a certain number of frames.

2. **Tracking Backward:** If a track is not initiated within the entrance zone of the scene (e.g., the image boundary), then it must be a track that is split from a previously merged track. Its position is predicted backward in time to find a nearest measurement. The track that originally occupies this measurement is denoted as a merged track. Meanwhile, a track split hypothesis H_s is generated and added to the list of hypotheses.

3. **Building Track Graph:** The list of merge/split hypotheses is sorted according to time. A vertex of the track graph is created for each track on this list. For each merge hypothesis H_m that merges track $\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, ...\mathcal{T}_{i_m}$ to track $\mathcal{T}_j$, corresponding edges from vertices $v_{i_1}, v_{i_2}, ..., v_{i_m}$ to v_j are added to the track graph. For each split hypothesis H_s that splits track $\mathcal{T}_i$ into track $\mathcal{T}_{j_1}, \mathcal{T}_{j_2}, ...\mathcal{T}_{j_n}$, the corresponding edges from vertex v_i to $v_{j_1}, v_{j_2}, ..., v_{j_n}$ are added.

3.2.3 Linking Strategy on Track Graph

We propose several linking strategies to process the track graph, which we call "local linking," "network linking," "global linking," and "multi-view linking." If occlusion occurs for a short period of time or the local feature computed from each tracklet is sufficiently discriminant, we can use a local or network linking strategy. If occlusion occurs frequently for a long period of time or a global feature needs to be computed to describe the whole trajectory, we can apply a global linking strategy. When tracklets from multiple views are available, we can also apply a multi-view linking strategy.

Local Linking

The local linking strategy mainly consists of two steps: determining the flow on the track graph and iteratively stitching pairs of tracklets.

Flow Computation. To determine the number of objects involved in the merging/splitting events, we choose a path-reducing min-flow algorithm to compute the track-capacity of each vertex and flow for each edge. The number of objects in the track graph is equivalent to the amount of flow passing through the network. Since each track represents at least one object, we have a lower bound on the capacity of the edges in the track graph. This is not sufficient to uniquely determine the actual number of objects and resolve the ambiguity caused by occlusion, i.e., an arbitrary number of objects can "hide" in any merged track. For single-view tracking, we require our algorithm to select the smallest number of objects that can explain the graph. We thus convert our problem into a minimum-flow problem where the lower bound on the capacity of each edge is one. We use a polynomial-time algorithm that iteratively searches for a "reducing path" (as opposed to the "augmenting path" in the max-flow Ford-Fulkerson method [28]) and updates the residual network:

1. **Finding a feasible flow:** Starting from the source vertices, keep pushing flow through the graph G until the lower-bound capacity $c(u, v)$ (one in our case) of every edge $e_{u,v}$ is satisfied, which returns a feasible flow f. Determine the residual graph G_f to be the network with capacity $c_f(u, v) = f(u, v) - c(u, v)$.

2. **Path-reducing step:** If G_f has a path p from one source node in S to one sink node in T, reduce the edge capacity of G_f along path p by $c_f(p) = \min\{c_f(u, v) | (u, v) \in p\}$, and subtract $c_f(p)$ units along p from flow f. Repeat this step until no valid path can be found in residual graph G_f. The result flow f is the minimum flow.

Stitching Process. Once the track-capacity is computed, the vertices of the graph are first sorted according to time, which is the initiation time of its corresponding track. The local linking algorithm processes each vertex sequentially until all vertices have been matched. Here a bipartite matching problem is constructed to model the linking problem where multiple tracklets merge and split later, as shown in Fig. 3·3.

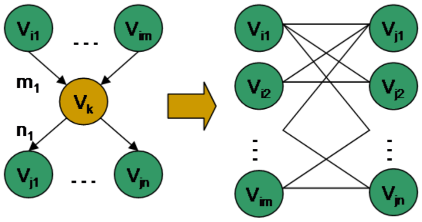


Figure 3·3: Example of track linking with a bipartite matching formulation. For each local graph structure that represents a merge-and-split event, we convert the linking problem into a bipartite matching problem, where tracklets before merging need to be matched to tracklets after splitting. A weight/cost has to be computed between each pair of tracklets and measures the similarity between the two, and the goal is to minimize the overall sum of assignment cost.

- For a merge hypothesis $H_m : \{(\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, ...\mathcal{T}_{i_m}) \vdash \mathcal{T}_k\}$, we extend each individual track with the merged track and smooth the connected trajectory. A new set of tracks is created $(\mathcal{T}_{i_1}\mathcal{T}_k, \mathcal{T}_{i_2}\mathcal{T}_k, ..., \mathcal{T}_{i_m}\mathcal{T}_k)$.

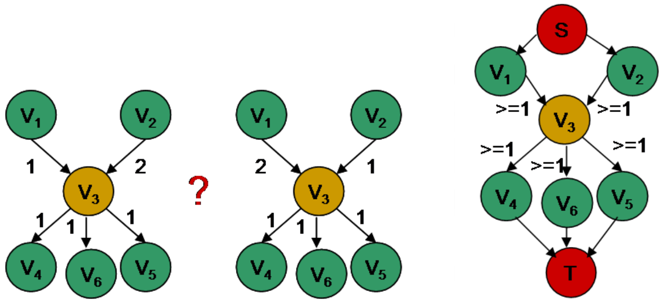
- For a split hypothesis $H_s : \{\mathcal{T}_k \vdash (\mathcal{T}_{j_1}, \mathcal{T}_{j_2}, ...\mathcal{T}_{j_n})\}$, we extend each split track reversely with the merged track and smooth the connected trajectory. The tracks are now $(\mathcal{T}_k\mathcal{T}_{j1}, \mathcal{T}_k\mathcal{T}_{j2}, ..., \mathcal{T}_k\mathcal{T}_{jn})$,

- For a merge hypothesis immediately followed by a split hypothesis, we search for the best match between two sets of tracks $H_a : \{(\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, ...\mathcal{T}_{i_m}) \vdash (\mathcal{T}_{j1}, \mathcal{T}_{j2}, ...\mathcal{T}_{jn})\}$, which is a bipartite matching problem shown in Fig. 3·3. The flow $f_{i \to k}$ determines the number of times track $\mathcal{T}_i$ has to be matched, and the flow $f_{k \to j}$ determines the number of times track $\mathcal{T}_j$ has to be matched. The matching cost between a pair of tracks $(\mathcal{T}_i, \mathcal{T}_j)$ depends on the specific application. Once we find the best match, we first link each track $\mathcal{T}_i$ with track $\mathcal{T}_k$ and then link that with $\mathcal{T}_j$, which is the match of $\mathcal{T}_i$, resulting in $\mathcal{T}_i\mathcal{T}_k\mathcal{T}_j$.

The above procedure repeats until all the hypotheses are processed. Since each linking operation makes a locally optimal choice based on the *local* feature, the result of the algorithm is only locally optimal.

Network Linking

The main issue of the local linking procedure is that there could be multiple solutions that all have the same amount of flow going through the graph but have different configurations, as shown in Fig. 3·4(a). A better solution is to combine the flow determination and matching process together, and formulate it as a minimum-cost flow problem. The idea of such a formulation was also explored by Castañón et al. [25]. Since their linking task was not designed for inter-object occlusion scenarios, there are no merged tracks in their graph representation. Note we still use *local* features in the network linking approach, and the cost function for the whole trajectory is additive, i.e., the total cost is the summation of pairwise linking costs of adjacent tracklets.



(a) (b)

Figure 3·4: Network linking. (a) Two tracks merged into one and split to three tracks. There is ambiguity in which of the two tracks before merging carries two objects (track capacity). (b) Instead of resolving the track graph iteratively, optimizing a global cost function by the minimum-cost flow avoids the ambiguity of track capacity determination. Flows are encouraged to pass edges with lower cost. Here, "S" and "T" nodes are virtual nodes that represent track initiation and termination.

The network linking approach creates an augmented graph, as shown in Fig. 3·4(b). In the augmented graph, all source nodes in the original track graph are connected to a virtual track initiation node, whereas all sink nodes are connected to a virtual track termination node. All edge capacities still have a lower bound of one to ensure that every edge is visited. For each edge $e_{i,j}$, a cost $c_{i,j}$ is defined to measure how likely tracklets $\mathcal{T}_i$ and $\mathcal{T}_j$ are on the same path. Then the objective function is to select the paths on the augmented track graph such that all lower bounds are satisfied and the total additive cost along the paths is minimum. This is exactly the minimum-cost flow problem

$$\begin{aligned} \min \quad & \sum_{i,j} c_{i,j} f_{i,j} & \\ \text{s. t.} \quad & \sum_i f_{i,v} = \sum_j f_{v,j}, & \forall v \in V \\ & f_{i,j} \geq 1, & \forall e_{i,j} \in E, \end{aligned} \tag{3.1}$$

which can be solved by many polynomial-time algorithms such as the push-relabel algorithm [28].

Global Linking

Global linking may connect several trajectory segments together at the same time, and the cost along a flow path is not decomposable. Instead, a *global* feature is computed from all the tracklets along the trajectory. We convert this problem to a weighted set-cover problem as follows.

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 $\{v_{i_1} v_{i_1} ... v_{i_p}\}$ of vertices visited. To connect our formulation to the standard set-cover problem, we ignore the order between the vertices of the sequences. The set of all paths is denoted as P. A weight w_p is associated with a path p that measures the negative log-likelihood of the path being a true trajectory, or equivalently the "cost" of the path based on a global feature such as motion smoothness. The objective function then is defined as selecting a subset P' of P such that the sum of the costs of all selected paths is minimum. Each vertex $v \in V$ has to be on some path at least once. Mathematically, this is equivalent to the following linear integer programming problem, where x_p is an integer variable to indicate if path p is selected x_p times:

$$\begin{aligned} & \min \sum_{p \in P} w_p x_p \\ \text{s. t.} \quad & \sum_{p: v \in p} x_p \geq 1, \quad \forall v \in V \\ & x_p \geq 0 \text{ and } x_p \text{ is integer.} \end{aligned} \tag{3.2}$$

To solve the set-cover problem, the deterministic greedy method achieves an approximation ratio of $\mathcal{H}(s)$, where s is the size of the largest set, and $\mathcal{H}(n) = \sum_{i=1}^{n} 1/i \approx \log(n)$ is the n-th harmonic number [57].

Linking in Multi-view

A more general scenario of track linking is to link tracklets from multiple views with a global linking cost. For ease of notation, we here consider only two views, but the method can be extended to an arbitrary number of views. We formulate the multi-view global linking problem as a *joint*-set-cover problem. Specifically, we generate a track graph for each view independently as $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. For each graph G_i, $i = 1, 2$, we enumerate all valid paths in set P_i. We define a_p and b_q to measure the respective likelihood of path $p \in P_1$ and $q \in P_2$ being true trajectories. Our goal is to choose a subset $P_i' \subseteq P_i$ to achieve a cover on V_i for each view, subject to the additional constraint that enforces any selected path $p \in P_i'$ has a corresponding path $q \in P_j'$ with an across-view matching cost $c_{p,q}$. We seek the solution that achieves the minimum weighted sum. Mathematically, it can be formulated as the following linear integer programming problem, where $z_{p,q}$ is a binary variable to indicate if a path pair $(p, q), p \in P_1, q \in P_2$ is selected or not:

$$
\begin{aligned}
&\min \sum_{p} a_p \sum_{q} z_{p,q} + \sum_{q} b_q \sum_{p} z_{p,q} + \sum_{p} \sum_{q} c_{p,q} z_{p,q} \\
\text{s. t. } &\sum_{p:u \in p} \sum_{q} z_{p,q} \geq 1, \forall u \in V_1 \\
&\sum_{q:v \in q} \sum_{p} z_{p,q} \geq 1, \forall v \in V_2 \\
&z_{p,q} \geq 0 \text{ and } z_{p,q} \text{ is integer}
\end{aligned} \tag{3.3}
$$

It is easy to see the joint-set-cover problem defined in Eq. 3.3 can be reduced to a standard weighted set-cover problem.

Proof For each pair of sets $p \in P_1, q \in P_2$, we create a joint set $o = p \cup q$ with an associated weight $w = a_p + b_q + c_{p,q}$. The new set of o is denoted as O and the new vertex set as $V = V_1 \cup V_2$. Now we need to find a subset $O' \subseteq O$ that is a cover on V with a minimum weighted sum, which is the weighted set-cover problem. ▮

In case some object does not appear in the field of a particular view, e.g., set $p \in P_1$ has no matching set $q \in P_2$, we add all pairs (p, q_0) to the joint set O, where $p \in P_1$ and q_0 is a "dummy" placeholder, and assign a large matching cost so that these elements have a low priority of being selected.

3.2.4 A Bayesian Justification for Track Linking

Given a collection of tracklets, the linking process can be formulated as a Bayesian estimation problem. To explain how tracklets can be produced given the true states of objects, we first associate each object with a state (position) vector X_i of length T, where T is the time span of the entire sequence. Each tracklet $\mathcal{T}_j$ is represented as a measurement vector of length T, where the entries outside of the time span of this tracklet are zeroed out. The relationship between an object to its tracklets is represented as edges in a bipartite graph, as shown in Fig. 3·5. Note that every solution for resolving the track graph can be uniquely represented by such a bipartite graph. For each edge in the bipartite graph, a diagonal matrix M is constructed to select part of the trajectory from state X_i. Then, given a bipartite graph represented as a collection of matrices $\mathbf{M}$, any tracklet measurement Z_j can be seen as generated from a combination of selected parts from all trajectories, for example, the mean of $\sum_i M_{i,j} X_i$ plus Gaussian noise. We define the state space of the bipartite graph for data association $\mathbf{M}$ as the space of feasible solutions that can resolve the track graph. The constraint given by the lower bound in the network-flow formulation or by the minimum cover used in the set-cover formulation is ensured because of the graph being bipartite.

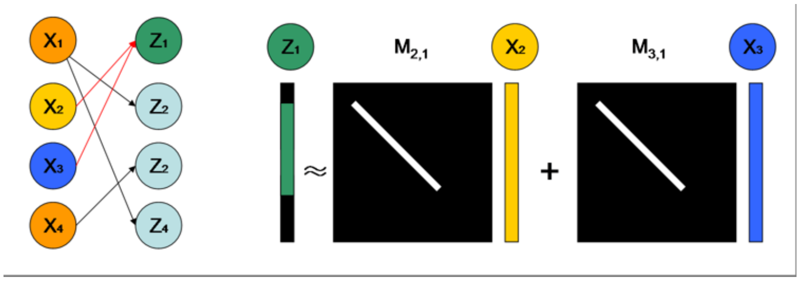


Figure 3·5: An example of a bipartite graph to generate tracklets from four objects whose states are $X_1, ..., X_4$. Nodes $Z_1, ..., Z_4$ are four tracklets. Here node Z_1 represents two merged tracklets shared by two objects of states X_2 and X_3. We associate each edge of the graph with a diagonal mapping matrix M that selects part of the state vector from its corresponding object. So the final observation Z_j of a tracklet can be represented concisely as vector $\sum_i M_{i,j} X_i$ corrupted by some random noise e. Here, $Z_1 = M_{2,1} X_2 + M_{3,1} X_3 + e$.

The linking methods described in the previous section are essentially maximizing the likelihood of the data, i.e., $\max_{\mathbf{M},\mathbf{X}} p(\mathbf{Z}|\mathbf{M}, \mathbf{X})$. They all simplify the object dependencies such that each object generates its own trajectory fragments independently. Therefore, the likelihood term is factorized into $\prod_i p(\mathbf{Z}^i|\mathbf{M}^i, X_i)$, where $\mathbf{Z}^i$ is the set of tracklets associated to object i through matrices $\mathbf{M}^i$. For global linking with the set-cover formulation, $\mathbf{Z}^i$ is the subset to select, and $-\ln p(\mathbf{Z}^i|\mathbf{M}^i, X_i)$ is the weight for the subset. For network linking, $p(\mathbf{Z}^i|\mathbf{M}^i, X_i)$ is decomposed into $p(Z^i_1|M^i_1, X_i) \prod_n p(Z^i_{n+1}|M^i_{n+1}, X_i) p(M^i_{n+1}|M^i_n, X_i)$, which only considers the pairwise similarity between adjacent tracklets. The negative log-likelihood then can be transformed to the cost $c_{i,j}$ defined on the network.

A stronger formulation could incorporate domain knowledge into the prior distribution and maximize the posterior: $p(\mathbf{M}, \mathbf{X}|\mathbf{Z}) \propto p(\mathbf{Z}|\mathbf{M}, \mathbf{X})p(\mathbf{M})p(\mathbf{X})$. For example, we can choose $p(\mathbf{M})$ to favor simple data associations where fewer objects merge or split in the scene, and use $p(\mathbf{X})$ to model object arrival and departure rate.

3.3 Experiments

To test the scalability and robustness of our various linking methods, we first conduct a quantitative evaluation on synthetic data. Then we test on the infrared video sequences introduced in the previous chapter, and compare the results with the results of other traditional sequential tracking methods.

3.3.1 Quantitative Evaluation on Synthetic Datasets

We randomly generate colored spheres with 10-unit radii, moving at constant speed in a 500^3-unit 3D space (Fig. 3·6). Each sphere carries a unique color as label, and the arrival time of each sphere is drawn uniformly from the interval $[1, T_{\max}]$ with $T_{\max} = 250$ frames. We create two virtual cameras for viewing the spheres from directions differing by 45^o. The motion model of each sphere is $X^{(t)} = FX^{(t-1)} + W^{(t)}$ and $Z^{(t)} = HX^{(t)} + V^{(t)}$ with 6D state X (3D position and velocity), 2D observation Z (virtual view of sphere), state transition matrix F, projection matrix H, and zero-mean Gaussian noise processes W and V with respective covariance matrices $\text{diag}(1, 1, 1, 0.1, 0.1, 0.1)$, and $\text{diag}(1, 1)$. We generate 6 datasets (D1-D6) with increasing density. Each dataset contains 5 sequences, each with 250 frames per view, resulting in a total of 15,000 test frames. Key statistics of the synthetic data are summarized in Table 3.3, rows 1–5. Row 5 shows the average number of errors (missed detections, false alarms, and track switches) that correspond to a 0.01 MOTA score. In order to compute the MOTA metric, a match between the ground truth and the system-generated track is uniquely determined by the color of the sphere.

The track graph representation is constructed by forward-backward nearest-neighbor filtering (Sec. 3.2.2). All linking methods use the same set of tracklets from the track graph

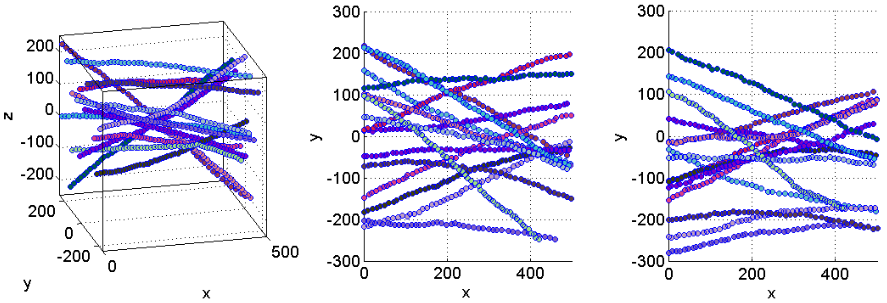


Figure 3·6: Fifteen sample trajectories in 3D space (left), randomly generated by the simulator, and their images in two views (middle and right) with numerous occlusions. We use matching colors to visualize corresponding trajectories.

as input. For the local linking method, the cost of pairing two tracklets is chosen to be the standard deviation of the linear-regression residual over the observed 2D coordinates, assuming that the motion is along a straight line for short periods. In case a long tracklet may present nonlinear motion pattern, we only extract at most 10 measurements right before or after interactions. For the global linking method, the cost function that measures how likely several tracklets can form a smooth trajectory is evaluated by Kalman filtering. The initialization parameter setting is similar to that described in Sec. 2.3.2. For the multi-view linking method, the across-view cost function is defined as the reconstruction error according to the epipolar geometry, which is a least-square solution to the triangulation. In our implementation of the dynamic Bayesian network method by Nillius et al. [60], we follow their recommendation to restrict the dependence between two vertices (here the number of objects involved in an occlusion event and the frequency of such events) within 20 frames. Details of the heuristics can be found in the paper by Nillius et al. [60].

We measure the performance of each linking method using the CLEAR MOTA metric described in Sec. 2.3.2 (Table 3.3, rows 6–10), for which a **small** difference in a score can reveal a **significant** difference in tracking accuracy (see row 5). Not surprisingly, the performance for all methods decreases as the density of objects in the scene increases. Both the global and multi-view linking methods outperform the other linking strategies. The

Table 3.3: Statistics of synthetic datasets and comparison results

	D1	D2	D3	D4	D5	D6
Avg. Objs / frame	8.8	17.8	27.1	36.2	45.5	54.6
Max. Objs / frame	16	27	38	51	59	73
Occlusions / frame	0.13	0.60	1.53	2.69	3.87	5.48
# errors, 0.01 MOTA	23	46	70	95	120	145
Nillius et al. [60]	0.92	**0.92**	0.86	0.82	NA	NA
Local Linking	0.90	0.83	0.75	0.68	0.59	0.53
Network Linking	**0.95**	0.91	0.87	0.82	0.81	0.80
Global Linking	**0.95**	**0.92**	**0.89**	**0.85**	**0.83**	**0.81**
Multi-view Linking	**0.95**	**0.92**	**0.89**	**0.85**	0.81	0.78

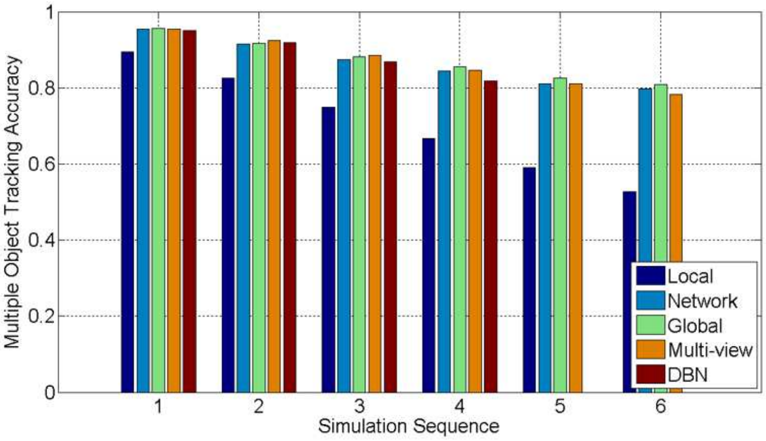


Figure 3·7: Comparison of MOTA metric on simulation datasets with increasing density. The MOTA score is averaged over all test sequences for each density category. The compared methods are local linking, network linking, global linking, multi-view linking and the dynamic Bayesian network method proposed by Nillius et al. [60].

method by Nillius et al. [60], also a global approach, achieves comparable performance but failed to handle very dense scenarios (no reports for D5, D6). It is simply too slow because its state space is too large even with their proposed heuristics applied [60]. For a vertex with n incoming and n outgoing edges, our global linking method enumerates n^2 paths passing this vertex. In contrast, the method by Nillius et al. [60] must evaluate $n!$ possibilities of matching between incoming and outgoing edges.

Although the multi-view linking method shows a good performance by using additional 3D geometric information, it starts to degrade in the dense scenarios of our simulation

(D5, D6), where the size of the proposed joint-set cover problem is much larger than each single set-cover problem. In this case, the additional benefits that geometric information provides are compromised by the inaccuracy of the greedy solution. An advantage of the multi-view linking method is that, as a byproduct, it gives the trajectory correspondences between views, which can be further used for 3D path reconstruction.

3.3.2 Quantitative Evaluation on Infrared Video Datasets

We also test our track linking algorithms on real datasets for infrared video analysis of bats, as described in Sec. 2.3.1. Our data contains a long sequence of 1,100 frames from three views with low density, and a short sequence of 200 frames with high density. We apply background subtraction to detect bats in each image, followed by labeling of connected components. The position of each bat is located by finding the pixel with the highest intensity value within the connected component. Because of occlusion, a single component might correspond to the overlapping images of multiple bats.

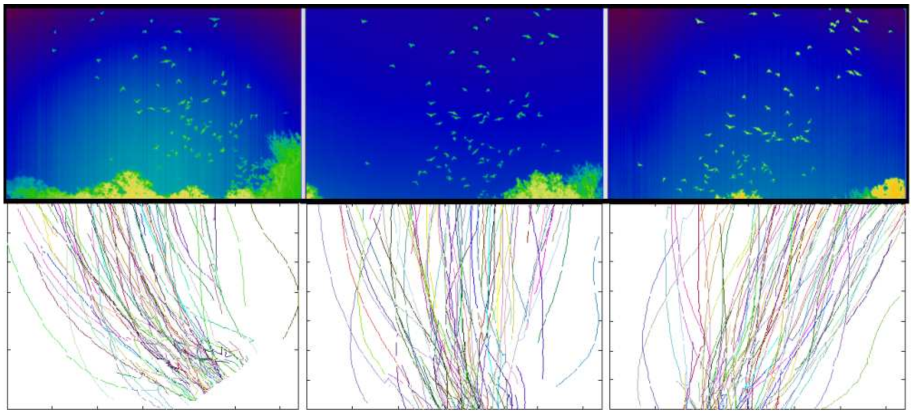


Figure 3·8: Corresponding infrared video frames from three cameras (top) and system-generated trajectories (bottom) from bat dataset.

For dataset of bats, we compare the performance of five track-linking approaches as well as the two classic measurement-level sequential approaches JPDA and MHT. We implement both JPDA and MHT in their standard forms that do not model the occlusion events. Quantitative results are shown in Table 3.4 and Table 3.5. The metrics are “Mostly

Table 3.4: Quantitative results for track linking on infrared videos with objects in low density.

Data	Method	MT	ML	MOTA	MR	FPR	IDS
	JPDA	165	0	0.796	0.129	0.074	**17**
Bats B1	MHT	172	0	0.836	0.118	**0.043**	82
View 1	Local Linking	**202**	0	**0.863**	**0.039**	0.095	67
211 objects	Network Linking	178	1	0.842	0.091	0.066	**17**
1100 frames	Global Linking	175	0	0.839	0.092	0.068	22
	Multiview Linking	187	3	0.819	0.085	0.094	54
	DBN [60]	179	5	0.826	0.087	0.085	49
	JPDA	166	0	0.813	0.122	0.064	41
Bats B1	MHT	184	1	**0.871**	0.089	**0.033**	150
View 2	Local Linking	**203**	0	0.869	**0.025**	0.103	74
211 objects	Network Linking	179	3	0.851	0.087	0.061	**34**
1100 frames	Global Linking	183	1	0.860	0.077	0.061	35
	Multiview Linking	191	3	0.849	0.070	0.079	52
	DBN [60]	183	1	0.841	0.079	0.077	52
	JPDA	179	0	0.854	0.129	0.067	20
Bats B1	MHT	187	0	0.887	0.094	**0.034**	93
View 3	Local Linking	**200**	0	**0.910**	**0.027**	0.109	27
209 objects	Network Linking	191	0	0.888	0.092	0.064	**16**
1100 frames	Global Linking	192	0	0.889	0.082	0.065	18
	Multiview Linking	196	0	0.897	0.074	0.083	29
	DBN [60]	198	1	0.902	0.084	0.082	31

Tracked (MT)," "Mostly Lost (ML)," "Multiple Object Tracking Accuracy (MOTA)," "Miss Rate (MR)," "False Positive Rate (FPR)," and "ID Switches (IDS)." Details of definitions of these metrics can be found in Sec. 2.3.2. In order to compute these metrics, the user-defined threshold for hit/miss used by the MOTA metric is set to 10 pixels. We use 5-scanback for MHT and one-scanback for JPDA. We use the same cost functions for the track-linking methods as for the synthetic data.

The linking approaches are in general superior to the sequential methods since the tracks are constructed based on all information in the sequence. Both of the two sequential methods degrade significantly when tested on datasets with high density, while the linking methods are less sensitive to the density. The dynamic Bayesian network approach proposed by Nillius et al. [60] can only work with simple track graphs, and, as a result, fails to run on dense object sequences. The local, global and network linking approaches are relatively more efficient in their computations, and therefore could be applied on the

Table 3.5: Quantitative results for track linking on infrared videos with objects in high density.

Data	Method	MT	ML	MOTA	MR	FPR	IDS
	JPDA	136	5	0.738	0.194	0.066	31
Bats B2	MHT	125	7	0.778	0.201	**0.019**	32
View 1	Local Linking	**188**	2	0.792	**0.073**	0.131	53
211 objects	Network Linking	172	4	0.836	0.121	0.041	**14**
200 frames	Global Linking	171	3	**0.837**	0.119	0.042	24
	Multiview Linking	167	6	0.788	0.144	0.065	50
	JPDA	144	3	0.733	0.208	0.055	57
Bats B2	MHT	139	3	0.795	0.183	**0.019**	48
View 2	Local Linking	**203**	0	0.793	**0.046**	0.153	124
212 objects	Network Linking	173	3	0.815	0.127	0.055	**42**
200 frames	Global Linking	172	2	**0.823**	0.127	0.048	46
	Multiview Linking	164	4	0.762	0.136	0.098	71
	JPDA	135	5	0.726	0.221	0.051	22
Bats B2	MHT	128	11	0.765	0.214	**0.019**	40
View 3	Local Linking	**198**	3	0.807	**0.074**	0.114	92
221 objects	Network Linking	168	5	0.801	0.144	0.052	36
200 frames	Global Linking	172	4	**0.814**	0.132	0.052	**35**
	Multiview Linking	167	6	0.784	0.128	0.082	85

large-scale datasets.

The MOTA difference between the global and local linking strategies is not as conclusive as it is in the simulation. On the one hand, the local linking approach obtains lower miss rates but presents significantly higher false positive rates, which also implies a higher frequency in ID switches. On the other hand, both network linking and global linking methods show better robustness across different evaluation metrics than the local linking method. The multi-view linking approach does not perform as well on the real as on the synthetic data. This may be a result of inaccuracies of camera calibration and errors in the detection step. Nonetheless, it is important to note that the multi-view approach is particularly relevant for imaging situations in which local or global information is sparse, e.g., objects look alike and move in highly nonlinear patterns. In these situations, stereoscopic geometry might be the only useful information to help tracking through occlusion.

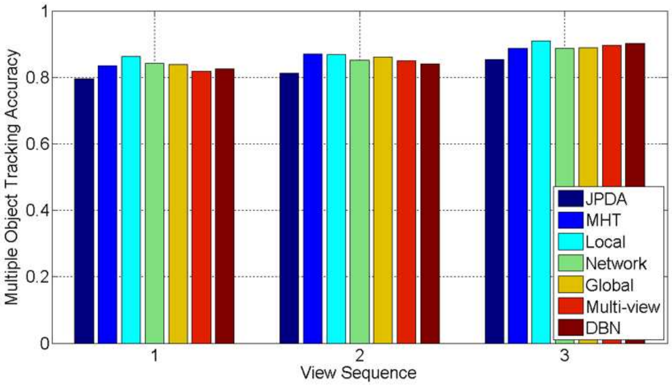


Figure 3·9: Comparison of MOTA metric on infrared dataset of bats with low density. The compared methods are sequential filtering methods JPDA and MHT, local linking, network linking, global linking, multi-view linking and the dynamic Bayesian network method proposed by Nillius et al. [60].

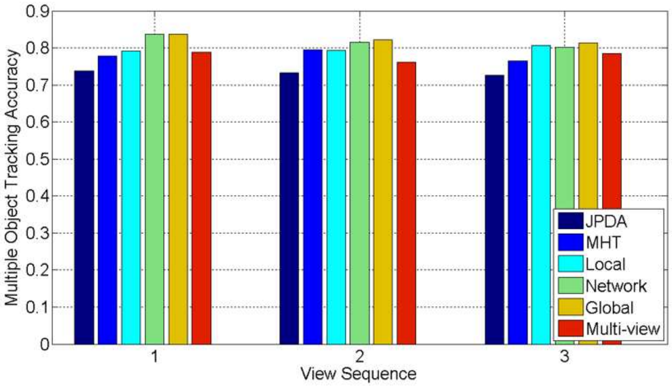


Figure 3·10: Comparison of MOTA metric on infrared bats dataset with dense density. The compared methods are JPDA, MHT, local-linking, network-linking, global-linking, and multiview-linking.

3.4 Summary and Discussion

In this chapter, we proposed a track linking framework for reasoning about both short-term and long-term occlusions. All linking strategies are unified under the same framework called “track graph” that describes the track merging and splitting events caused by occlusion. To explain short-term occlusions, when local information is sufficient to distinguish objects, the process links trajectory segments through a series of locally optimal bipartite-graph matches or a minimum-cost flow formulation. To resolve long-term occlusions, when global features are needed to characterize objects, the linking process computes a logarithmic approximation solution to the set-cover problem. If multiple views are available, our method builds a track graph independently for each view, and then simultaneously links track segments from each graph, solving a joint-set-cover problem for which a logarithmic approximation also exists. Through experiments on different datasets, we show that our proposed techniques make the track graph a particularly useful tool for tracking large groups of individuals in images.

The track graph is an interesting representation which could be useful to visualize object interactions through time. A potential problem is that if interactions happen frequently between two objects, the number of paths in the graph will grow exponentially which makes it difficult to apply global reasoning. In practice, the best reasoning might require a combination of local and global solutions. From the Bayesian point of view, another extension is to consider maximum-a-posteriori estimation instead of maximum-likelihood estimation. The prior distribution on the structure of the track graph could be useful to identify false alarm tracklets as well as abnormal merging/splitting events in the cell imaging application. With the descriptive modeling of producing tracklets in Sec. 3.2.4, it is also possible to apply sampling-based techniques when the problem size is too large for discrete optimization.

Chapter 4

Coupled Detection and Association

Our previous efforts tackle the occlusion problem purely through a data association step. The success of these methods either relies on additional views or accuracy of tracklets produced by low-level trackers. In this chapter, we address the occlusion problem in **both** the detection and data association steps. In contrast to the traditional "detection-tracking" system, the combined decision is more accurate and robust in that it can significantly reduce the risk of having errors propagate from noisy detector output to data association. Detection errors such as false alarms or missed detections due to occlusion could be corrected by feeding temporal information through tracking. This coupling idea appears attractive but introduces new challenges as well:

1. What type of objective function should be used? Many existing detection methods have not even been formalized with an objective function.
2. How can the new objective function be solved? Many current data association methods are complicated and approximate solutions to intractable problems. A new objective function that couples detection and data association might be even more difficult to optimize.
3. How can scalability of the proposed method be ensured? Computer vision systems face demands for being able to track large numbers of objects in dense formations. Given such large input sizes, an efficient algorithm to optimize the new objective function must be found.

Here we address all the questions above with a formulation of a coupling function and a method to optimize it. In particular, we propose a detection method with the classic sparse-signal recovery technique [23] for the dense-object tracking scenario when a background subtraction technique is available. This method can be used to detect objects moving on the ground plane as well as objects moving in free 3D space. The sparsity constraint is important here because it can significantly reduce the number of false alarms and serves as a replacement for the heuristic technique of non-maximum suppression of hypotheses. We have to take care, however, that the approach does not lead to overly sparse results, that is, missed detections. Estimation of occlusion relationships is also naturally embedded into this detector. To further boost the detection accuracy, we also impose a smoothness constraint from the data association aspect where we assume the state of each object follows a first-order Markov process and adopt the classical network flow formulation [24].

Unlike many coupling problems that rely on coordinate descent techniques, our overall objective function has a simple form and can be solved through a Lagrange dual decomposition that permits distributed computing. The method distributes the coupling formulation to subproblems and coordinates their local solutions to achieve a globally optimal solution. For each subproblem, efficient off-the-shelf algorithms are available. The framework is novel and also flexible in the sense that other modeling choices for each of the subproblems are possible.

We first review related work that helps occlusion reasoning from detection or tracking aspects in Sec. 4.1. In Sec. 4.2, we describe the proposed coupling framework and introduce our sparsity-driven detector with supporting experiments in Sec. 4.3. We conclude this chapter in Sec. 4.4 by discussing the strength and possible extensions of the proposed framework.

4.1 Related Work

Most previous efforts have followed two distinct directions of research for occlusion reasoning in tracking: building stronger object detectors and designing better data association methods. As a result, almost all existing tracking systems use a “detection-tracking design” with two separate modules to address the occlusion reasoning task independently.

4.1.1 Occlusion Reasoning in Detection

There are two main challenges for occlusion reasoning in object detection. First, when occlusion occurs, the occluded object is not observable on image plane, which introduces unpredicted uncertainty for most model-based detectors. A part-based detector may be able to handle certain partially occluded objects when sufficient pixel resolution is available [86, 45], but it fails when objects are completely occluded or the resolution of an object is too small. Even for pedestrians, a well-studied object category in the computer vision community, the performance of the state-of-the-art method drops significantly under partial occlusion and degrades catastrophically for lower resolution [33]. Moreover, directly modeling the occlusion process is difficult in general, as the degree of partial occlusion needs to be explicitly expressed in the object model. However, a detailed object model is probably not so necessary for many surveillance applications. Sometimes, it is not even useful due to limited resolution or challenging imaging conditions. As an alternative, when a reasonable background subtraction method is available, a common idea is to fit binary shape templates to the observations with the help of scene knowledge, such as camera calibration or multiview geometry [2, 40, 44, 91]. These methods all rely on a background subtraction preprocessing step, which itself could be a difficult research problem. Therefore, they are sensitive to the quality of background subtraction and the degree of partial occlusion.

Second, the heuristic “non-maximum suppression” technique, adopted in most detection methods that aim to cluster close hypotheses, often explains away true detections. This side effect is particularly undesirable when objects appear with large overlap on the image plane. Instead of tuning parameters for this heuristic step, a number of recent works have shown that it is beneficial to formulate object detection as a global optimization problem constrained by the Minimum Description Length or a context prior [31, 12, 91], and let the optimization process determine which hypotheses to select without applying any ad-hoc decisions. Our detection methods used in the coupling framework fall into this category.

4.1.2 Occlusion Reasoning in Tracking

Despite efforts in detecting partially occluded objects, missed detection/false positives are still inevitable, and such ambiguity may be resolved in the data association stage. As we have seen in the previous chapter, research efforts that address multiple object tracking typically treat occluded objects as missed detection events or track occluded objects all together with a single tracker, and iteratively grow or stitch tracklets before and after occlusions [87, 96, 51, 93, 47, 63, 25]. However, all these approaches follow the "detection-tracking strategy" and therefore rely on good detectors for initialization. This limits the generalization of these approaches to more challenging data where missed detections or false alarms are not rare events. Thus, without solving the detection problem first, hoping data association itself will fix all detection errors is not promising.

Explicit occlusion modeling also appears in a recent work by Andriyenko et al. [6], who integrated an occlusion model in their global objective function. As the objective function becomes more and more complicated, it becomes highly non-convex, and the optimization relies on good initialization as well as ad-hoc sampling heuristics to avoid local minima. Instead, our formulation is mathematically rigorous and much simpler to optimize.

4.1.3 Coupling Techniques

As the occlusion problem cannot be resolved solely by the detection or data association algorithms, a natural extension is to consider combining these two subproblems into a single framework and take advantage of the often complementary nature of the two subproblems. A generative part-based model was proposed by Andriluka et al. [5] that combines tracking and detection of pedestrians. It models both the approximate articulation of each person as well as the temporal coherency within a walking cycle. While such a detailed part-based model offers a principled way to handle inter-person occlusions, the richness of the representation requires sufficient resolution so that the part appearance can be properly modeled. Another coupling idea for pedestrian tracking was proposed by Leibe et al. [53], who coupled the two through a quadratic Boolean function and optimized it according to the Minimum Length Description criterion. The objective function is closer to our formulation but it is not easy to generalize to other choices/combinations of detection and data association methods. Under this formulation, a suboptimal solution was obtained through EM-type alternating minimizations, and therefore the quality is subject to a good initialization as well. We instead base our method on the foundations of Bayesian estimation theory. Our objective function has relatively lower complexity and is straightforward to extend to model higher-order relationships between objects. The proposed formulation has better scalability, and is very general in the sense that the solution to each subproblem can be easily substituted by other classic approaches.

4.2 The Coupling Framework

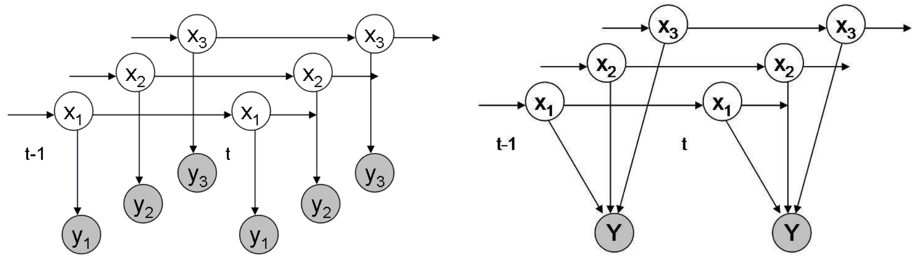
Our coupling framework can be derived from Bayesian estimation theory, where different choices of modeling the image likelihood and motion prior can lead to various coupling objective functions. In particular, we give a concrete example that uses a sparsity-driven detector and a network-flow association method within this framework, which shows a significant improvement over state-of-the-art approaches on challenging video sequences.

Table 4.1: Notation for coupled detection and association method

$\mathbf{Y}$	binary observation (vector) of the entire image
$\hat{\mathbf{Y}}$	non-negative integer observation (vector) of the entire image
$\mathbf{D}$	dictionary matrix of all codewords
d_i	codeword (entire image) for an object at encoded position i in 3D
$\mathbf{X}$	binary vector to select a subset of codewords from dictionary
$c_{i,j}^{(t)}$	transitional cost for moving from vertex v_i to vertex v_j at time t
$f_{i,j}^{(t)}$	flow variable to show an object moving from v_i to v_j at time t

4.2.1 Bayesian Formulation for Multiple Object Tracking

We formulate the multiple object tracking problem as a maximum-a-posteriori estimation problem. Given a collection $\mathbf{Y}$ of image evidence for the entire sequence, we estimate the



(a) Traditional Model (b) Our Model

Figure 4·1: The graphical model for the multiple object tracking problem. (a) The image likelihood is modeled with an independence assumption, that is, the image evidence y_i for each object x_i is generated independently, i.e., $p(\mathbf{Y}|\mathbf{X}) = \prod_i p(y_i|x_i)$; (b) The image evidence $\mathbf{Y}$ is jointly determined by **all** objects, i.e., $p(\mathbf{Y}|\mathbf{X})$ cannot be factorized further.

state of all objects $\mathbf{X}$ in the scene as follows:

$$\begin{aligned}
& \max_{\mathbf{X}} p(\mathbf{X}|\mathbf{Y}) \\
\propto\ & \max_{\mathbf{X}} p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}) \\
=\ & \max_{\mathbf{X}} \prod_{t=1}^{T} p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_1)\prod_{t=2}^{T} p(\mathbf{X}_t|\mathbf{X}_{t-1}) \\
=\ & \max_{\mathbf{X}} \prod_{t=1}^{T} p(\mathbf{Y}_t|\mathbf{X}_t)\prod_{i=1}^{M} p(x_{i,1})\prod_{t=2}^{T} p(x_{i,t}|x_{i,t-1}) \qquad (4.1)
\end{aligned}$$

Here, $p(\mathbf{Y}_t|\mathbf{X}_t)$ is the image likelihood conditioned on **all** objects. The joint state of all objects is governed by a Markov process and objects are independent from each other, so $p(\mathbf{X})$ can be factorized with respect to each individual object. We do not further factorize the image likelihood because all objects jointly generate the image. This enables us to model spatial relationships between objects and handle occlusions. The graphical model for our generative process is depicted in Fig. 4·1(b).

Without modeling the likelihood for the entire image but instead making certain independence assumptions, one can further factorize the first term of Eq. 4.1, a technique used by most earlier tracking approaches, see Fig. 4·1. A side effect of the independence assumption is that it yields ad-hoc choices (e.g., non-maximum suppression) because the number of objects is also a hidden variable to be inferred. In contrast, if the likelihood for the entire image is modeled, context and the relationship between objects are naturally brought into consideration. This observation has been recognized widely for the topic of scene recognition [31]. Directly estimating the joint hidden states is difficult here because we do not even know the dimension of the joint state. We propose a decomposition technique to tackle the MAP estimation problem. After taking the negative logarithm of Eq. 4.1, we rewrite the optimization problem as follows:

$$\begin{aligned} &\min_{\mathbf{X}_1,\mathbf{X}_2} g(\mathbf{X}_1,\mathbf{Y}) + h(\mathbf{X}_2) \\ s.\,t. \qquad &\mathbf{X}_1 = q(\mathbf{X}_2), \end{aligned} \tag{4.2}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are two copies of hidden state variables (the dimension of the state variable needs to be determined during the inference since we do not know the number of objects yet), g is the function that models the detection problem, h the function that models the data association problem and q the function that enforces the agreement between the solutions $\mathbf{X}_1$ and $\mathbf{X}_2$ of the two subproblems. More specifically, $g(\mathbf{X}_1,\mathbf{Y})$ is minimized to estimate the states $\mathbf{X}_1$ of objects from image evidence $\mathbf{Y}$ and $h(\mathbf{X}_2)$ is minimized to infer the states $\mathbf{X}_2$ of objects from motion or other types of prior knowledge. Both coupling variables $\mathbf{X}_1$ and $\mathbf{X}_2$ could be discrete or continuous. If a filtering technique that works in the continuous domain is used to solve the data association subproblem, q here could be a quantization mapping. A more general extension to Eqn. 4.2 is to allow two subproblems to utilize different sources of image evidence $\mathbf{Y}_1, \mathbf{Y}_2$:

$$\begin{aligned} &\min_{\mathbf{X}_1,\mathbf{X}_2} g(\mathbf{X}_1,\mathbf{Y}_1) + h(\mathbf{X}_2,\mathbf{Y}_2) \\ s.\,t. \qquad &\mathbf{X}_1 = q(\mathbf{X}_2), \end{aligned} \tag{4.3}$$

Eq. 4.2 is a classic setup in operations research: a minimization problem with a coupling constraint. This type of formulation has been applied to the labeling problem, e.g., MRF-based image segmentation [52]. In the remainder of this chapter, we show that the coupling formulation is also useful for solving the tracking problem. We first define functions g and h in Section 4.2.2 and 4.2.3 respectively, by giving specific examples of detection and data association methods.

4.2.2 Sparsity-driven Detector

Inspired by the sparsity-driven people localization method proposed Alahi et al. [2], we propose the following L_1-norm minimization formulation as our object detector. First we discretize the space in which objects move. If camera information is available, then for each possible location in 3D, we can reproject the object to the image plane. The reprojected foreground image can be seen as a template or a "codeword." The codeword can be just an image in the single-view case, or a concatenation of images in the multiple-view case. By construction, each codeword has encoded scale and shape information by re-scaling and translating templates in the image plane. By collecting all codewords in discretized 3D space, we build the dictionary $\mathbf{D}$ for a particular category of objects, see Fig. 4·2. The length of each codeword is the size of the observed image(s), while the number of entries in the dictionary is determined by the discretization. Usually, the step of creating the codeword dictionary can be performed offline. But for tracking objects in a 3D volume, as in Fig. 4·3, the discretization of the entire volume is infeasible. In this case, we only consider valid triangulations formed from 2D detections using epipolar geometry and build the dictionary on the fly. Here a triangulation is valid if the reconstruction error is within a certain tolerance.

Given the binary foreground image $\mathbf{Y}$ after background subtraction, we want to find the best way to instantiate the codewords from the dictionary such that the generated image is as close to observation $\mathbf{Y}$ as possible. Mathematically, we want to minimize the following L_0-norm, defined as the Hamming distance from zero, where $\mathbf{X}$ is an binary vector to indicate which codeword to select from the dictionary and N the number of codewords:

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_0, \text{ where } \mathbf{X} \in \{0,1\}^N. \tag{4.4}$$

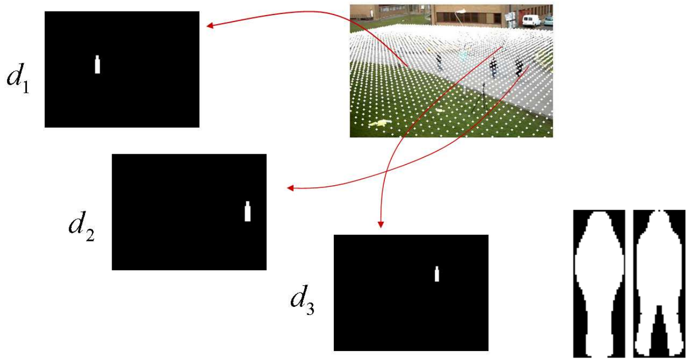


Figure 4·2: For objects that move on the ground plane, our method discretizes the plane into a grid, where the binary image of the instantiation of an object at each grid point is a codeword (e.g., d_1, d_2, and d_3). Two binary shape templates for front and side views of pedestrian are used.

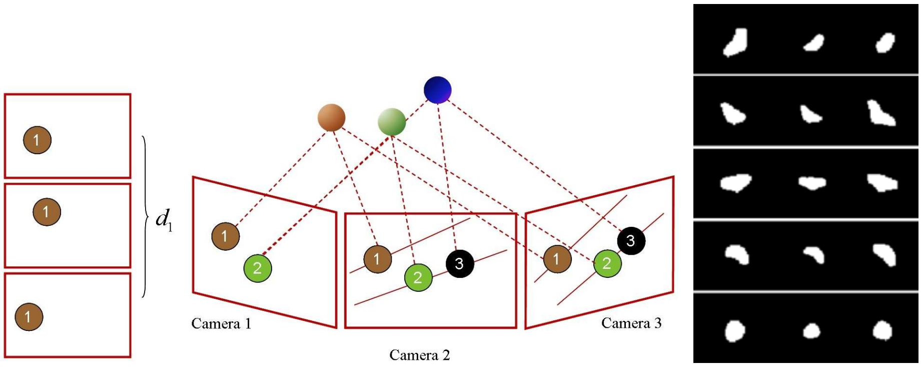


Figure 4·3: For objects that move in a 3D volume, our method constructs the pool of candidate locations in 3D by triangulation, keeping the reconstruction error below a threshold. The images of the re-projection of each candidate object is one codeword. Five binary shape templates for flying bats are used, while each template consists of the same pose in three views.

Because of the way we construct the dictionary, the selection variable $\mathbf{X}$ also encodes the positions of objects in 3D. The L_0-norm can be seen as our approximation to the negative logarithm of image likelihood $p(\mathbf{Y}|\mathbf{X})$ defined in Eq. 4.1. It is in general difficult to optimize, so we take the L_1-norm instead. According to the well-studied sparse signal recovery theory [23], the recovery of $\mathbf{X}$ using the L_1-norm is “almost” accurate if $\mathbf{X}$ is sparse (only has a few of non-zero entries). Because of occlusion, the real imaging process we model here should actually be a linear combination of codewords followed by a 1-bit quantization step, i.e., $Q(\mathbf{DX})$. A common way to handle quantization is to treat its effect as noise [20], in addition to the random noise that accounts for the degradation of background subtraction or inaccuracy of shape templates. Therefore, the whole generative process can be expressed as $\mathbf{Y} \sim \mathbf{DX} + e_r + e_q$, where e_r and e_q are random noise and quantization effect respectively. As long as the noise is sparse, the sparse signal recovery theory still applies.

By replacing the L_0-norm with the L_1-norm in the Eq. (4.4), the original formulation can be converted to the following linear programming problem:

$$\begin{aligned} & \min_{\mathbf{X},\mathbf{U}} \ \mathbf{1}^T\mathbf{U} \\ s.\,t. \quad & -\mathbf{DX} - \mathbf{U} + \mathbf{Y} \leq 0, \\ & \mathbf{DX} - \mathbf{U} - \mathbf{Y} \leq 0, \\ & \mathbf{0} \leq \mathbf{X} \leq \mathbf{1}, \end{aligned} \tag{4.5}$$

where $\mathbf{U}$ is an auxiliary variable. Notice the above formulation with the L_1-norm is a relaxed version of the original problem because $\mathbf{X}$ is continuous in Eq. 4.5. A branch-and-bound method can be applied to further get the exact integer solution. The L_1-norm introduces sparsity in the solution, which is a desirable property as we want to use a minimal number of hypotheses to explain the image observation. We refer to the solution of Eqn. 4.5 as the “Linear Denoising Decoder (LDND).”

In case we need to consider shape variations of the objects, we just enrich our dictionary by providing multiple templates that model these variations. The shape templates for a specific category of object can be learned from training examples through unsupervised clustering. We then impose a uniqueness constraint on our selection variable $\mathbf{X}$, i.e, the system can only choose one of the multiple templates to explain our image evidence as a valid solution. The following modified minimization formulation supports multiple versions of shape template shown in Fig. 4·2 and Fig. 4·3 used in our experiments:

$$\begin{aligned} &\min_{\mathbf{X}} \|\mathbf{Y} - \sum_i \mathbf{D}_i \mathbf{X}_i\|_0, \\ s.\,t. \qquad &\sum_i \mathbf{X}_i \leq \mathbf{1}, \mathbf{X}_i \in \{0,1\}^N; \end{aligned} \tag{4.6}$$

The 1-bit quantization described in the above generative process is very crude. In our detection context, when severe occlusion between objects exists, the noise that accounts for the quantization effect is not sparse anymore. As a result, the L_1-norm approximation is not applicable. A simple example to demonstrate the quantization effect is given in Fig. 4·4. One-bit dequantization in general is an ill-posed problem even for the noise-free case, as the magnitude of the original signal is completely lost. Here we express the value of the dequantized signal at each pixel by an "occlusion layer" variable, as it can explain how many objects are involved in the occlusion at that pixel. By definition, this occlusion layer variable only takes non-negative integer values. We extend Eqn. 4.5 to a new linear programming problem that simultaneously estimates occlusion layers for 1-bit

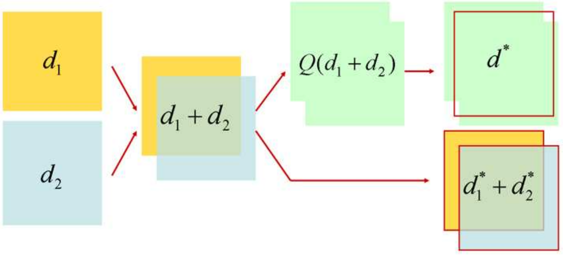


Figure 4·4: Two binary signals d_1 and d_2 (codewords) are overlayed on top of each other which simulates the occlusion effect. From background subtraction, a 1-bit quantized measurement $Q(d_1 + d_2)$ is obtained. By solving the minimization problem in Eqn. 4.4, the "best" recovered signal is just one binary signal d^* that tries to cover most nonzero entries as much as possible, and the remaining uncovered part is considered sparse random noise. Clearly, if we were able to obtain the dequantized measurement $d_1 + d_2$, it is much easier to recover the original two signals by using the same minimization formulation.

dequantization and sparse signal recovery as follows:

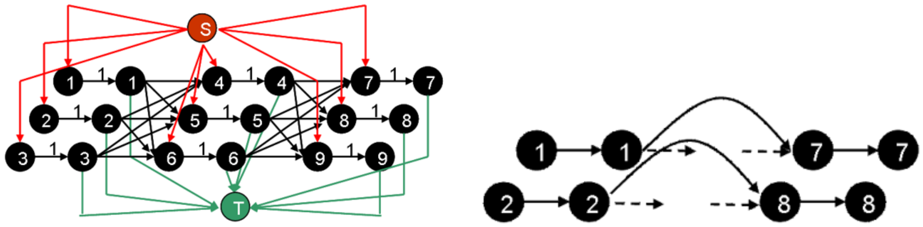
$$
\begin{aligned}
\min_{\mathbf{X},\hat{\mathbf{Y}}} \quad & \|\hat{\mathbf{Y}} - \mathbf{D}\mathbf{X}\|_1 + \beta\|\hat{\mathbf{Y}} - \mathbf{Y}\|_1, \\
s.\,t. \quad & \hat{y}_i \geq y_i, \quad \forall i : y_i > 0 \\
& \hat{y}_j = 0, \quad \forall j : y_j = 0 \\
& \text{and } \hat{\mathbf{Y}} \in \{\mathcal{Z}^+\}^N, \mathbf{X} \in \{0,1\}^N, \qquad (4.7)
\end{aligned}
$$

where $\hat{\mathbf{Y}}$ is the occlusion layer to be estimated, which has to preserve the quantization correctness: $Q(\hat{\mathbf{Y}}) = \mathbf{Y}$. The new appended term in the objective function is the L_1 regularization that penalizes the difference between the dequantized and quantized measurements. The parameter β weighs the two terms in the objective function, and $\mathcal{Z}^+$ is the set of non-negative integers. By linear relaxation, the above problem can be converted to a linear programming problem similar to Eqn. 4.5, where many off-the-shelf LP solvers could be used. We experimented with both an optimal branch-and-bound method and a simple rounding approach that yielded an integer solution. We did not find strong evidence that the branch-and-bound method produced significantly better results, so we ended up using the simple rounding in our experiments.

Regularization is necessary to ensure that the estimation of the two sets of variables is not ill-posed. The weighting parameter β controls the quality of dequantization which we determined by experiment. We refer to the solution of Eqn. 4.7 as "Linear Dequantization Decoder (LDQD)." From now on, we use the augmented formulation Eqn. 4.7 as our sparsity-driven detector (SDD).

4.2.3 Network-flow Data Association

The classical network-flow data association method represents every detection returned from the detector in every frame as a node in a network and every potential match between detections across time as an arc with an associated cost. We increase the size of network by setting **all** possible locations of objects in the scene as the nodes. The black circles in Fig. 4·5 represent all possible locations at each time frame stacked in columns. Each edge



(a) (b)

Figure 4·5: Data association as a minimum-cost network-flow problem. (a) A flow of amount 1 along a path from the source S (track initiation) to the sink T (track termination) represents a single object. Here, three candidate detections, (1,2,3), (4,5,6) and (7,8,9), were made in each of three frames. Duplicate nodes with capacity-one arcs ensure mutually disjoint paths are computed. Here, up to 27 paths can be represented. (b) An extension of network used in (a) is to add "jumping" edges in order to represent a path with miss detection for a few frames. A flow going from 1 directly to 7 without passing any of (4,5,6) means the object disappears at that time frame.

represents a potential move from one location to another and there is a cost associated on each edge in the graph. It adds two special vertices, "source" and "sink," to represent track initiation and termination. To ensure multiple tracks do not share the same detection, nodes in each time step are duplicated, and a single, unit-capacity, zero-cost arc is added between them [24]. By enforcing the upper bound on the flow of this edge to be one, the paths or the flows going through the graph are guaranteed to be mutually exclusive. The goal is to push the right amount of flow into the network that corresponds to the trajectories of objects, i.e., sequences of associated detections so that the total cost along the flows is minimum; this is a standard min-cost flow problem. As the number of objects present is unknown a priori, the method needs to search for the amount of flow that produces the minimum cost. It is important to notice that the network flow data association assumes the cost function over a track is additive, i.e., it is a summation of edge cost along the path. Other simple extensions to capture missed detections or model higher order motion information such as velocity are possible with an increased number of edges [7, 65]. We here select the network-flow formulation as our data association method because several efficient algorithms exist [15]. The minimization problem is given as follows:

$$\begin{aligned} \min_{\mathbf{f}} \quad & \sum_i \sum_j c_{i,j} f_{i,j} \\ \text{s. t.} \quad & \sum_i f_{i,n} = \sum_j f_{n,j}, \quad \forall\, n \in V \\ & f_{i,j} \geq 0 \ . \end{aligned} \tag{4.8}$$

where $c_{i,j}$ is the cost associated with each edge that links node i and j; $f_{i,j}$ is the flow variable associated with each edge, whose optimal value is always integer for such a network. The constraint set ensures the conservation property that the amount of incoming flows is the same as the amount of outgoing flows at each candidate detection node. Notice that, if all costs defined on edges have positive value, there will be no flow pushed into the network. The network starts to function properly only when strong detection evidence shows up. In this case, each node will be associated with a negative detection score that might make some path in the network have total negative cost. Such dynamic updating of the cost within the network is the key ingredient of our coupling framework, which will be explained in the next section.

4.2.4 The Coupling Algorithm

To couple our sparsity-driven detector and network-flow data association methods, we propose a joint objective function, where $\sum_t \|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1$ approximates the negative logarithm of the image likelihood $p(\mathbf{Y}|\mathbf{X})$ and the sum of flow costs $\sum_i \sum_j c_{i,j} f_{i,j}$ approximates the negative logarithm of the Markov motion prior $p(\mathbf{X})$ described in Eq. 4.1. We separate the set of flow variables $\mathbf{f}$ into four subsets: $f^{(t)}_{in,n^+}$ is associated with the edge that connects the source node to the n-th node at frame t; $f^{(t)}_{n^-,out}$ is associated with the edge that connects the n-th node to the sink node at frame t; $f^{(t)}_{n^+,n^-}$ is associated with the edge that connects the duplicated n-th nodes at frame t; $f^{(t)}_{m^-,n^+}$ is associated with the edge that connects the m-th node at frame t to the n-th node at frame $t+1$. By rearranging the variables in the network-flow problem given in Eqn. 4.8 and using Eqn. 4.7, we have the following coupled minimization problem:

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{f},\hat{\mathbf{Y}}} \quad & \sum_t \|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta\|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1 \\
& + \sum_t \sum_n c^{(t)}_{in,\, n^+} f^{(t)}_{in,\, n^+} + \sum_t \sum_n c^{(t)}_{n^-,\, out} f^{(t)}_{n^-,\, out} \\
& + \sum_t \sum_n c^{(t)}_{n^+,\, n^-} f^{(t)}_{n^+,\, n^-} + \sum_t \sum_m \sum_n c^{(t)}_{m^-,\, n^+} f^{(t)}_{m^-,\, n^+} && (4.9)
\end{aligned}
$$

$$
\begin{aligned}
\text{s. t.} \quad & f^{(t)}_{in,\, n^+} + \sum_m f^{(t)}_{m^-,\, n^+} = \sum_k f^{(t)}_{n^-,\, k^+} + f^{(t)}_{n^-,\, out} = f^{(t)}_{n^+,\, n^-}, \quad \forall t, \forall n && (4.10) \\
& \sum_t \sum_n f^{(t)}_{in,\, n^+} = \sum_t \sum_n f^{(t)}_{n^-,\, out} && (4.11) \\
& x_{t,n} = f^{(t)}_{n^+,\, n^-}, \quad \forall t, \forall n && (4.12) \\
& \hat{y}_{t,i} \geq y_{t,i}, \quad \forall t, \forall i : y_{t,i} > 0 \\
& \hat{y}_{t,j} = 0, \quad \forall t, \forall j : y_{t,j} = 0 \\
& \mathbf{f} \geq 0, \hat{\mathbf{Y}}_t \in \{\mathcal{Z}^+\}^N \text{ and } \mathbf{X}_t \in \{0,1\}^N.
\end{aligned}
$$

The selection variable **X** indicates the presence of an object at a particular location in discretized space. The flow variable **f** is used in the minimum-cost flow problem, where $f_{i,j} = 1$ means there is a match between detections at location i and j, which belong to the same track. The cost function (4.9) is the summation of two *local* terms to minimize; the first term represents the costs of sparsity-driven object detection (Sec. 4.2.2) and the second term measures the costs of temporal data association in the minimum-cost flow formulation (Sec. 4.2.3). The first set of constraints (4.10 and 4.11) ensures a balance of flow. The second set of constraints (4.12) ensures *consistency* between the two local variables **X** and **f**. In other words, if there is a true detection at location n at time t, i.e, $x_{t,n} = 1$, there must be a flow going through the same location at the same time, i.e, $f^{(t)}_{n^+,n^-} = 1$.

Since this is a linear integer programming problem, we can apply a general LP solver to find the optimal solution. This limits the scalability and generalization when hundreds of frames need to be computed or another high-order form of the objective function needs to be considered. Instead, because of the special structure of the objective function, we can decompose the problem into two kinds of subproblems, each of which can be solved with an efficient algorithm, and ensure to coordinate the separate minimizers until an agreement is achieved. This approach can be pursued by formulating the Lagrangian dual problem (4.13) to the minimization problem (4.9):

$$\begin{aligned}
L(\boldsymbol{\lambda}) \quad &= \quad \min_{\mathbf{X},\mathbf{f},\hat{\mathbf{Y}}} \Big(\sum_t \|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1 + \boldsymbol{\lambda}_t^T \mathbf{X}_t \\
&\quad + \sum_t \sum_n c^{(t)}_{in,n^+} f^{(t)}_{in,n^+} + \sum_t \sum_n c^{(t)}_{n^-,out} f^{(t)}_{n^-,out} \qquad (4.13) \\
&\quad + \sum_t \sum_n (c^{(t)}_{n^+,n^-} - \lambda_{t,n}) f^{(t)}_{n^+,n^-} + \sum_t \sum_m \sum_n c^{(t)}_{m^-,n^+} f^{(t)}_{m^-,n^+} \Big) \\
\text{s. t.} \quad & f^{(t)}_{in,n^+} + \sum_m f^{(t)}_{m^-,n^+} = \sum_k f^{(t)}_{n^-,k^+} + f^{(t)}_{n^-,out} = f^{(t)}_{n^+,n^-}, \quad \forall t, \forall n \\
& \sum_t \sum_n f^{(t)}_{in,n^+} = \sum_t \sum_n f^{(t)}_{n^-,out} \\
& \hat{y}_{t,i} \geq y_{t,i}, \qquad \forall t, \forall i : y_{t,i} > 0 \\
& \hat{y}_{t,j} = 0, \qquad \forall t, \forall j : y_{t,j} = 0 \\
& \mathbf{f} \geq 0, \hat{\mathbf{Y}}_t \in \{\mathcal{Z}^+\}^N \text{ and } \mathbf{X}_t \in \{0,1\}^N. \qquad (4.14)
\end{aligned}$$

It can be separated into $(T+1)$ independent subproblems, where T is the number of frames:

$$\begin{aligned}
g_t(\boldsymbol{\lambda}) \quad &= \quad \min_{\mathbf{X}_t \in \{0,1\}^N, \hat{\mathbf{Y}}_t \in \{\mathcal{Z}^+\}^N} \quad \|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1 + \boldsymbol{\lambda}_t^T \mathbf{X}_t \\
\text{s. t.} \quad & \hat{y}_{t,i} \geq y_{t,i}, \qquad \forall t, \forall i : y_{t,i} > 0 \\
& \hat{y}_{t,j} = 0, \qquad \forall t, \forall j : y_{t,j} = 0 \qquad (4.15) \\
h(\boldsymbol{\lambda}) \quad &= \quad \min_{\mathbf{f} \geq 0} \sum_t \sum_n c^{(t)}_{in,n^+} f^{(t)}_{in,n^+} + \sum_t \sum_n c^{(t)}_{n^-,out} f^{(t)}_{n^-,out} \\
&\quad + \sum_t \sum_n (c^{(t)}_{n^+,n^-} - \lambda_{t,n}) f^{(t)}_{n^+,n^-} + \sum_t \sum_m \sum_n c^{(t)}_{m^-,n^+} f^{(t)}_{m^-,n^+} \\
\text{s. t.} \quad & f^{(t)}_{in,n^+} + \sum_m f^{(t)}_{m^-,n^+} = \sum_k f^{(t)}_{n^-,k^+} + f^{(t)}_{n^-,out} = f^{(t)}_{n^+,n^-}, \quad \forall t, \forall n \\
& \sum_t \sum_n f^{(t)}_{in,n^+} = \sum_t \sum_n f^{(t)}_{n^-,out} \qquad (4.16)
\end{aligned}$$

Now the dual problem is to maximize $\sum_t g_t(\boldsymbol{\lambda}) + h(\boldsymbol{\lambda})$ with variable $\boldsymbol{\lambda}$. Here we use a subgradient method to solve the “master problem.” The Lagrange multiplier $\boldsymbol{\lambda}$ will be updated in each iteration, which can be seen as a perturbation to the original subproblem without the Lagrange term. Therefore, a re-optimization technique should be considered so we do not need to solve subproblems at each iteration from scratch. In our problem, the perturbation only occurs at the objective function and the constraint set remains the same. A primal method would be suitable for this case since the solution from the previous iteration remains feasible. As a result, the primal method can reuse the previous solution as a starting point without the need to search for a starting feasible solution. In particular, we choose the Cplex implementation [1] of the primal-simplex algorithm to solve the first T subproblems with parallel computing, and the network-simplex algorithm to solve the min-cost flow subproblem. Details of the dual decomposition technique are described by Bertsekas [16].

In summary, the dual decomposition technique then yields the following Coupling Algorithm:

```
COUPLING ALGORITHM FOR TRACKING
For $k = 1, 2, ..., K$ (max iterations), do
    • Solve $T$ sparsity-driven detection problems with the primal-simplex algorithm:
      $\mathbf{X}_t \leftarrow \arg\min g_t(\mathbf{X}_t, \boldsymbol{\lambda})$.
    • Solve the minimum-cost flow data-association problem with the network-simplex algorithm:
      $\mathbf{f} \leftarrow \arg\min h(\mathbf{f}, \boldsymbol{\lambda})$.
    • If $x_{t,n} = f^{(t)}_{n^+,n^-}$ for all $n, t$, Then Return $\mathbf{X}_t, \mathbf{f}$
    • Update dual variables $\lambda_{t,n} = \lambda_{t,n} + \alpha_k(x_{t,n} - f^{(t)}_{n^+,n^-})$, $\alpha_k = \frac{1}{k}$ (step size).
Return $\mathbf{X}_t, \mathbf{f}$
```

The Coupling Algorithm performs as desired in our tracking context: The Lagrange multiplier λ serves as a weighting parameter. For the detection subproblem, a higher value of λ implies a lower preference for detection at a particular location. For the data association subproblem, a higher value of λ leads to a lower edge cost, so it attracts

---

[1] Cplex is available from http://www-01.ibm.com/software/integration/optimization/cplex-optimizer

flows passing through that edge. When agreement is achieved, the optimal global solution is obtained for the primal objective function. The detection output is guaranteed to be smooth because of the influence of data association. The flow computation produces tracks as the final output. By changing the value of λ dynamically through dual decomposition, false alarms can be suppressed and detections missed due to occlusions can be recovered.

4.3 Experiments

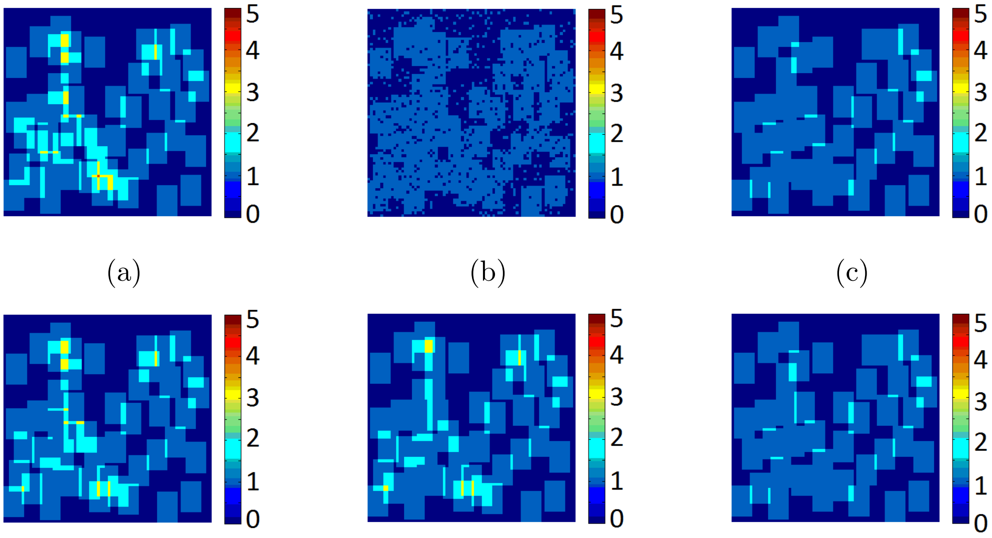
In this section, we first test our two versions of the sparsity-driven detector LDND and LDQD on synthetic data as well as the PETS2009 dataset. Then the coupling algorithm is applied on pedestrian sequences and infrared video sequences of flying bats with quantitative analysis and comparison to the state-of-the-art methods.

4.3.1 Quantitative Evaluation of the Sparsity-driven Detector

We first test our sparsity-driven detector on a simulated dataset. The task is designed to simulate the occlusion process in real data that incorporates the quantization effect and random noise, while the goal is to justify the necessity of applying a sparsity prior and a dequantization estimator at the same time. We are also interested in the robustness of our detector with respect to the noise level, the sparseness of the signal, and the amount of occlusion.

We generate rectangular binary boxes with size of 12×8 pixels randomly positioned in a square image of size 80×80. We call these rectangle binary boxes "box signals." After multiple potentially overlapping boxes are placed in the image (Fig. 4·6(a)), a measurement is taken after 1-bit quantizing the image corrupted by random noise (Fig. 4·6(b)). The noise is chosen to be uniformly distributed in the image and the sign of binary pixels is flipped randomly. Given such a measurement, our detector is adopted to recover the original 2D box signals including the number of boxes as well as their positions in the image.

To set up the minimization problem, each column of dictionary is a binary 80×80 image with one box at a particular location. The measurement to be generated is controlled by the



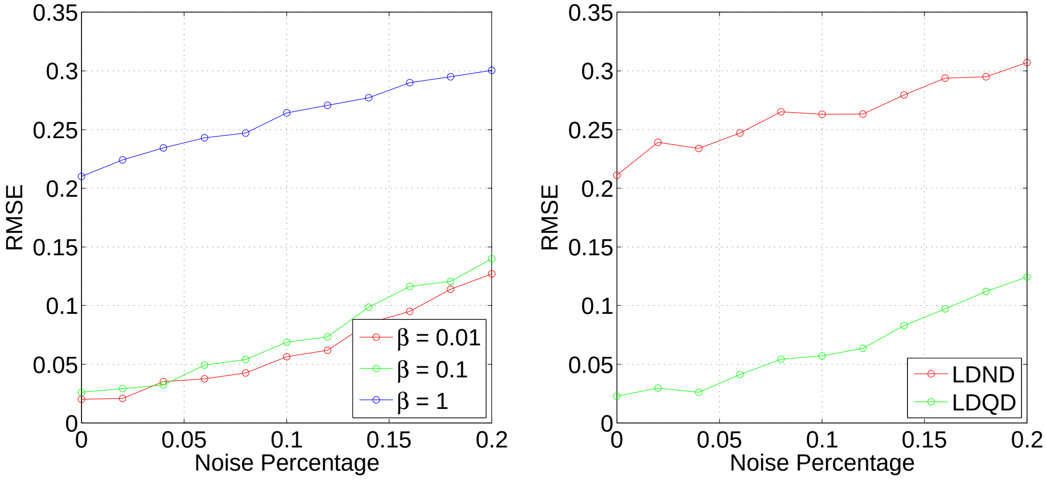
(a) (b) (c)

(d) (e) (f)

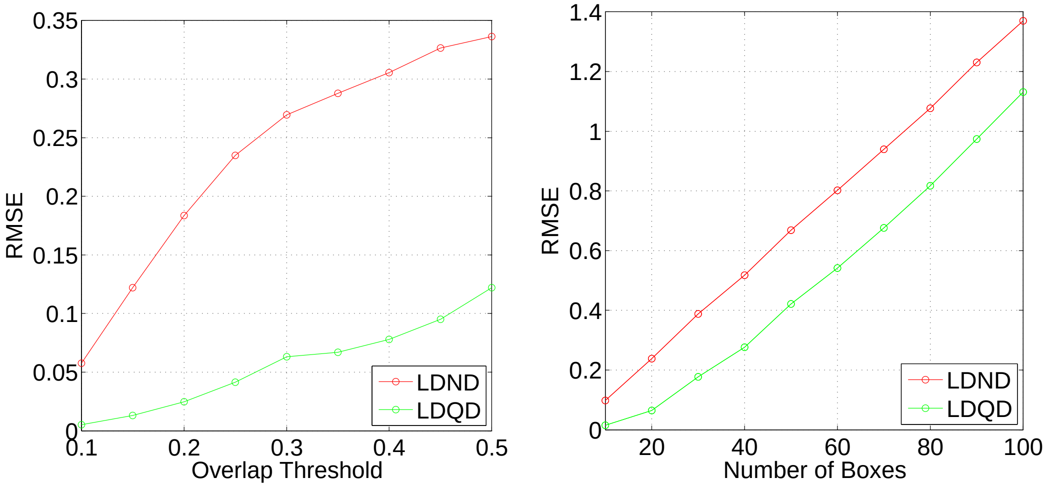
Figure 4·6: Sample images and results from the synthetic dataset. (a) The original noise-free box signals. A higher value at a pixel implies more boxes overlap at that pixel. (b) The quantized measurement with 20% of pixels corrupted by uniformly distributed random noise. (c) The estimated box signals by LDND without considering the quantization effect. (d) The estimated box signals with the weighing parameter $\beta = 0.01$ in LDQD (Eqn. 4.7). (e) $\beta = 0.1$. (f) $\beta = 1$. Visually, the reconstructed image (d) looks most similar to the original image (a).

noise level (percentage of pixels to be corrupted), the number of boxes, and the maximum overlap ratio allowed among boxes. One example of the simulated data is shown in Fig. 4·6, where 50 boxes are randomly positioned with maximum overlap ratio of 30% according to the VOC criterion [37].

We compare the performance of the Linear Denoising Decoder (LDND, Eqn. 4.5) and the Linear Dequantization Decoder (LDQD, Eqn. 4.7) on simulated data. One hundred testing samples are randomly generated for each parameter setting that is controlled by the three parameters mentioned above: the noise level, the number of boxes, and the maximum overlap ratio allowed among boxes. The overall performance is measured by the root mean square error (RMSE) between ground truth 2D image (Fig. 4·6(a)) and reconstructed image (Fig. 4·6(c)-(f)). As shown in Fig. 4·7, it is clear that LDQD consistently outperforms



(a) RMSE vs. Regularization β (b) RMSE vs. Noise Level



(c) RMSE vs. Maximum Overlap (d) RMES vs. Number of Boxes

Figure 4·7: Comparison of LDND and LDQD on the synthetic dataset. The performance is measured by the root-mean-square-error with respect to different parameters of the simulator. (a) The performance with different values of β. (b) The performance with varying noise level. The number of boxes is 30. The maximum overlap ratio is 0.3. $\beta = 0.01$ for LDQD (Eqn. 4.7). (c) The performance with varying overlap ratio. Higher overlap ratio means boxes have larger overlap between each other. The number of boxes is 30. The noise percentage is 10%. $\beta = 0.01$ for LDQD. (d) The performance with varying number of boxes. Higher value means more boxes can overlap with each other. The maximum overlap ratio is 0.8. The noise percentage is 10%. $\beta = 0.01$ for LDQD.

LDND by explicitly modeling the quantization effect. A proper value for the weighting parameter β used in LDQD can be determined empirically by experiment. In scenarios with relatively fewer overlaps, the recovery by LDQD with a small regularization weight is almost exact even if the measurement is corrupted by a significant amount of noise. The quality of the recovered signal seems less sensitive to the random noise, which suggests that our background subtraction step and the shape templates used in our real dataset do not need to be perfect. However, the amount of overlap due to occlusions is a more sensitive factor, which justifies the necessity to explicitly model the quantization effect in LDQD.

We also test our sparsity-driven detector on PETS2009 [64] dataset for people localization. To compare with other reported results in the literature, three subsets (S1L1-1357, S1L2-1406, S2L1) from PETS2009 are selected. Only the first view of each sequence is chosen, which is used by most previous methods testing on these sequences. The performance is measured by four metrics: Multiple Object Detection Accuracy (MODA), Multiple Object Detection Precision (MODP), Precision, and Recall. Similar to MOTA, MODA accounts for all possible errors such as miss detection and false alarms. MODP measures the relative accuracy of alignment between ground truth and the predicted bounding box on image plane. Details of definitions of these metrics have been provided by Ellis et al. [34].

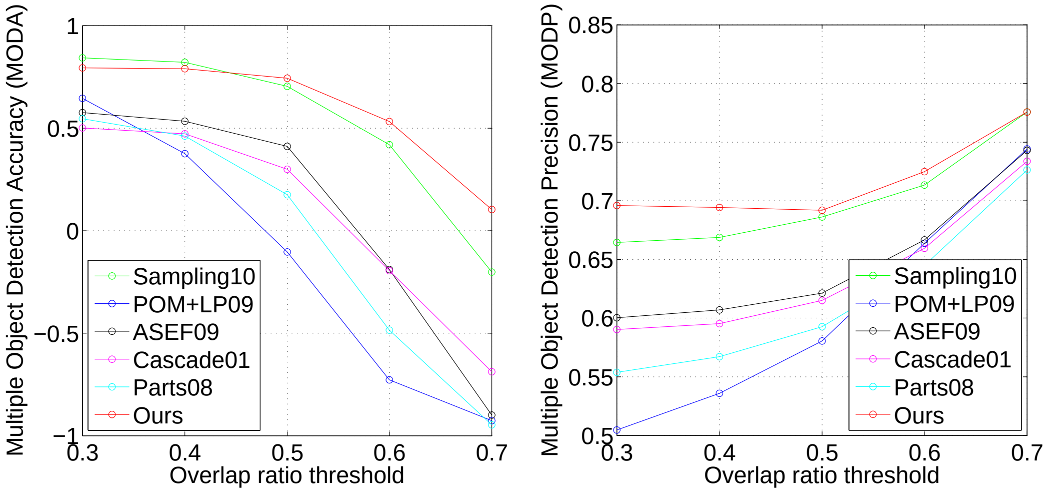
Implementation details. To obtain the binary image evidence, we run an adaptive Gaussian mixture estimation method for background subtraction [97]. The ground plane is discretized with a grid size of 0.3 m × 0.3 m, which is approximately half of the space a pedestrian could occupy. To speed up the computation, we rescale the binary image to a 320×240 pixel resolution. Two shape templates are used as described in Fig. 4·2. We further use two heuristics to reduce the size of the dictionary before running the LP solver. First, if a codeword does not receive sufficient support from the image, i.e., 50% of the foreground pixels are not detected in the grid, the corresponding column in the dictionary is removed. Second, the original length of the codeword is 320×240, the size of the image. However, a large portion of pixels in the image will not be covered by any codeword in the dictionary, either because the pixel is outside of the monitoring region or possible codewords that can cover this pixel have been removed in the first step. The corresponding entries for these “uncoverable” pixels are removed from the rows of the dictionary. The final size of the dictionary constructed for this dataset is approximately 500 codewords, each approximately representing 20,000 pixels. The regularization parameter β is chosen to be 0.1.

Comparison with the state-of-the-art methods. We compared our detector against several state-of-the-art methods for which results have been reported on these sequences[2]. The MCMC sampler [42] has a flavor similar to our method in that it samples shape templates from a much richer set to synthesize a binary image and compare it against an image computed from background subtraction. Such a method does not enforce sparsity on its solution, nor considers the quantization effect explicitly. Moreover, the sampling process converges very slowly (30 s per frame in their Matlab implementation) while it is vulnerable to be trapped in local minima. The Average Synthetic Exact Filter (ASEF) method [19] is a correlation-based method that captures the gradient information around the silhouette of the object. Such a filter does not consider possible occlusions so it tends to be sensitive to the loss of gradient information and fail on partially visible objects. The POM+LP method [13] is a complete tracking system which also requires discretization of space and a binary shape template. The presence of an object is modeled in a probabilistic way called “probabilistic occupancy map” and relies on tracking to identify true detections or false alarms. We will revisit this method in our tracking experiments in Sec. 4.3.2. Finally, two popular classifier-based detectors, Cascade [83] and Part-based Model [39], which are designed for general object detection purposes, are also compared. The evaluation results at different levels of hit/miss thresholds are shown in Fig. 4·8. The performance of related methods was previously reported by Ge [42].

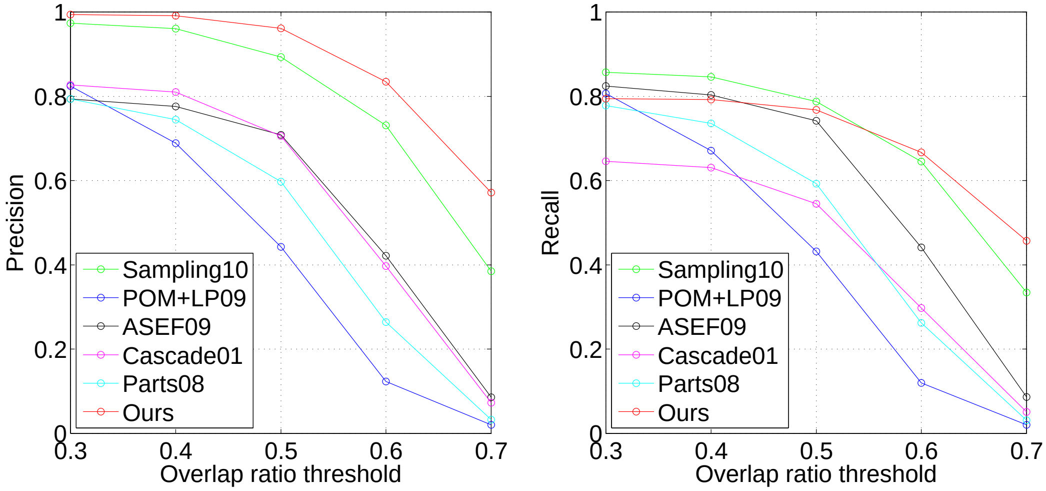
We also compare our method with a similar approach by Alahi et al. [2], but where

---

[2] The ground truth is provided through `http://www.gris.informatik.tu-darmstadt.de/~aandriye`



(a) MODA vs. Hit/Miss Threshold (b) MODP vs. Hit/Miss Threshold



(c) Precision vs. Hit/Miss Threshold (d) Recall vs. Hit/Miss Threshold

Figure 4·8: (a). Evaluation results for MODA on PETS2009 S2L1. Our method is plotted in red. (b) Evaluation results for MODP on PETS2009 S2L1 (c) Evaluation results for Precision on PETS2009 S1L1-13-57 (d) Evaluation results for Recall on PETS2009 S1L1-13-57. In official rules of PETS evaluation, the hit/miss threshold is set to 0.5.

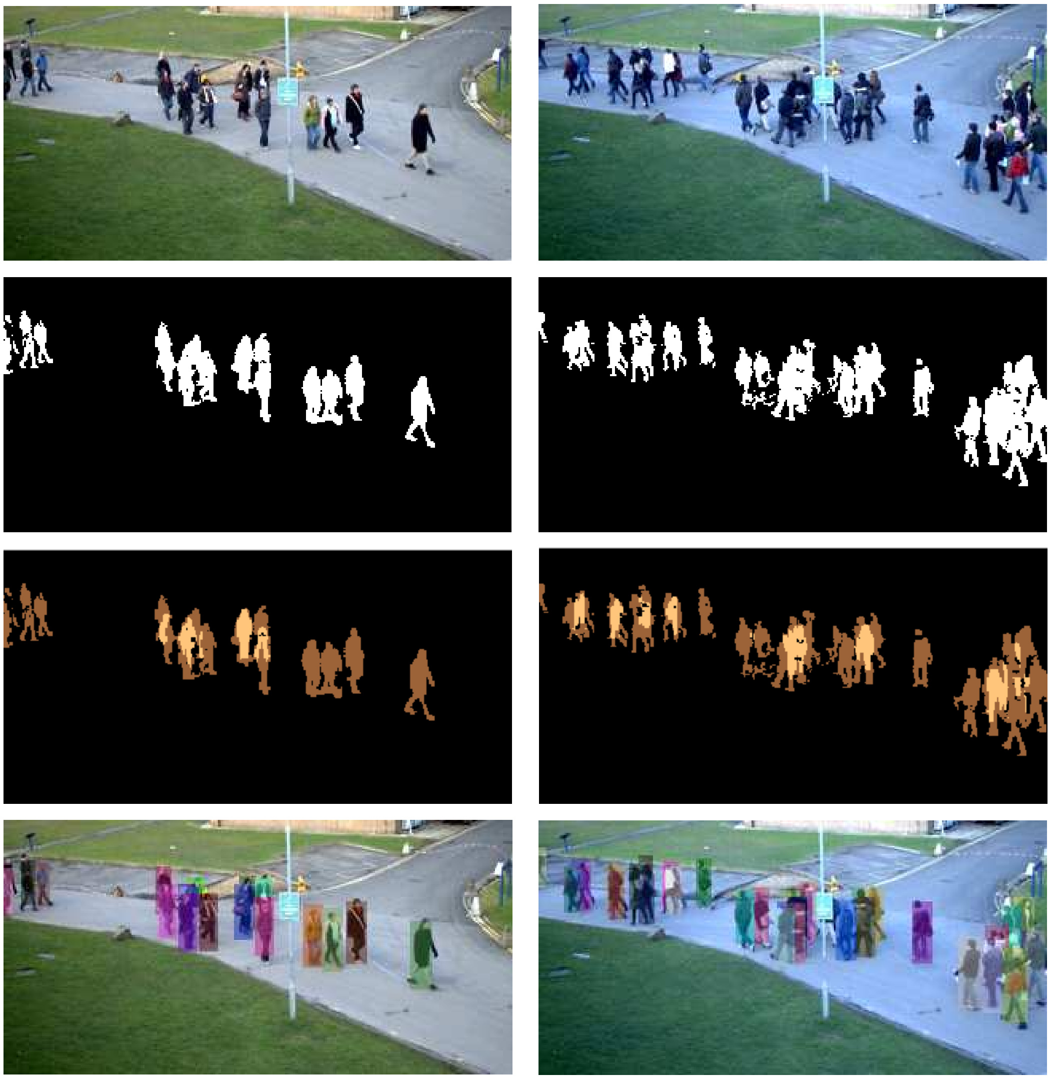
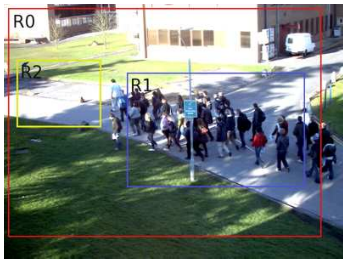


Figure 4·9: Sample frames and detection results on PETS2009 dataset. The original sample frames from PETS2009 S1L1-1359 and S1L1-1357 are listed in the first row. The second row shows the binary images after background subtraction. The third row shows the estimated occlusion layers by our detector. The fourth row gives the final detection results.

we measure the reconstruction error with the L_2 norm. In order to compare with their reported results, we follow the official rule in PETS2009 for the people counting task, where only the numbers of people passing through specified regions are computed. The quality of the counting algorithm is evaluated by computing the Average Frame Error (AFE). Superior performance of our method is reported in Fig. 4·10.



Data	Method	R0 AFE	R1 AFE	R2 AFE
S1L1 -1359	Alahi et al [2]	4.2	2.3	1.8
	Our LDQD	**1.5**	**0.6**	**1.0**
S1L2 -1406	Alahi et al [2]	n/a	6.5	4.0
	Our LDQD	6.0	**3.6**	**1.9**

Figure 4·10: People counting results on PETS2009 dataset. People passing through the "R0," "R1," and "R2" regions are counted. The performance is measured by computing the Average Frame Error (AFE), whose ideal value is 0.

As expected, methods that use the background subtraction technique in general produce better results than those that are classifier-based methods. Among those that work with binary images as input, our sparsity-driven detector (SDD) consistently outperforms competing algorithms in all metrics evaluated. As shown in the third row in Fig. 4·9, our detector does not only localize the pedestrians but it also produces an estimate of occlusion layers as well.

4.3.2 Quantitative Evaluation for the Coupling Algorithm

To test our coupling algorithm that combines the sparsity-driven detector (SDD) and network-flow data association method, we evaluate its performance on the PETS2009 dataset [64] for pedestrian tracking, as well as the infrared BU-Bats dataset described in Sec. 2.3.1. Four sequences with the first view from the PETS2009 benchmark are selected: S2L1 (795 frames), S2L2 (436 frames), S1L1-1357 (221 frames) and S1L1-1359 (241 frames). We use the ground truth annotation provided by Andriyenko et al. [6]. To enable comparisons with previously published results, we restrict our evaluation to objects moving in the constrained area defined by Andiryenko et al. [6] shown in Fig. 4·12. We use CLEAR MOT metrics and USC metrics for evaluation, as described in earlier chapters. In order to compute MOTA, we choose 1 m on the ground plane as the miss/hit threshold for the PETS data and 0.3 m for the infrared data. MOTP measures the average distance between ground-truth trajectories and system-generated trajectories. We compute Euclidean distances for the 3D case (infrared data) and the overlap ratio between ground-truth and system-generated bounding boxes for the 2D case (PETS data).

Implementation details. We here describe the implementation details on how the shape templates are learned (Fig. 4·2) and how to set up the network (Fig. 4·5). To develop shape templates, we assume a pedestrian can occupy a cylinder with radius 30 cm and height 180 cm (Fig. 4·2), and that a flying animal can occupy a sphere volume of 15-cm radius (Fig. 4·3). The shape variation is learned through K-mean clustering on a training set that comprises of 200 unoccluded examples, which results in two shape templates for pedestrians and five templates for bats. The learned template from a typical K-mean clustering is a real-valued representation and we further binarize it to a binary shape template. The number of clusters K is chosen empirically to balance the size of the dictionary and the accuracy of performance. Although increasing the number of templates used in our two dictionaries could potentially improve the performance, we do not find it necessary given the relatively small resolution of the objects in the test data.

To set up the network used in data association, we need to define the cost on the edges. As shown in Fig. 4·5, there are two types of edges: edges between the duplicated nodes within a time frame, and edges between nodes across time, including the “jumping edges.” We call the cost defined on the first type of edge the “detection cost,” and the cost on the second type the “transition cost.” The detection cost is computed as $-\ln \frac{\rho}{1-\rho}$, where ρ is the ratio between the number of foreground pixels that can be explained by a codeword and the number of foreground pixels in that codeword, which can be seen as a measure to support the presence of an object at a particular position. For the PETS dataset, we first compute the histogram for the foreground pixels of each codeword. Given two codewords from consecutive frames, we compute the histogram intersection distance to represent the transition cost. For the infrared dataset of bats, we use the Euclidean distance between nodes as the transition cost. Notice that given the topology of the network, the transition cost computed using histogram intersection actually depends on the image data, so the result of the data association procedure is not independent of the image evidence but rather follows the generalized coupling framework in Eqn. 4.3. Without the detection cost that potentially has a negative value, the network-flow minimizer simply chooses a zero flow as the best output since all transition costs are non-negative. As a result, a drastic cost update (by subtracting $\lambda_{i,j}$) then occurs in subsequent iterations of the Coupling Algorithm. Finally, to reduce the number of edges, we do not allow transitions that would model a pedestrian's unrealistic move more than 2 m (7 fps for PETS) in one time frame or a bat's move more than 30 cm (125 fps for Bats). "Jumping edges" are only introduced within three time frames.

We also develop a sliding-window scheme to handle long sequences. The length of a sliding-window is limited to the availability of system memory but should not be too short. Throughout our experiments, at least 100 frames are processed each time with 20 frames overlap between adjacent subsequences. A bipartite matching is solved to link trajectories generated from the first and second batch.

Important parameter settings. There are a few user-defined parameters that need to be determined by experiment. The weighting parameter β in Eqn. 4.7 that governs the dequantization quality is set to 0.1 for the PETS dataset and 10 for the infrared dataset of bats, according to the detection performance on the training set. The infrared dataset has multi-view support so that the need to estimate the dequantization effect is not as strong as for the single-view PETS dataset. To ensure the numerical balance between the L_1-norm term and the network-flow term in Eqn. 4.9, the L_1-norm is re-scaled by another weighting parameter γ, which is set to 0.01 for both datasets. These weighting parameters are not sensitive in general. Once reasonable values for these weighting parameters are found, a wide range of values nearby could apply as well.

Comparison with the state-of-the-art methods. Our quantitative evaluation provides the tracking results on six sequences and compares them to the results of five related approaches, see Table 4.2. The occlusion-modeling (OM) method [6] achieves the best performance on the PETS dataset so far by combining explicit occlusion reasoning, a full-body SVM classifier, tracklet stitching, and initialization via extended Kalman filtering. Two versions of flow-based methods (POM+LP, ILP) have a problem set up similar to ours – both of which require discretization of the ground plane and background subtraction. They run the detection and network-flow data association modules sequentially and do not take advantage of the complementary nature of the two subproblems. We further extend the reconstruction-tracking method (RT-1), proposed in Chapter 2, by applying our SDD detector on a dense set of hypotheses of 3D points (SDD-RT-1). To address the noisy measurement issue, described at the end of Chapter 2, a dense set of 2D measurements is returned by sampling points from each connected component, which significantly increases the size of the set of valid triangulation hypotheses. However, running our SDD on this increased set will return a sparse set of detections.

As shown in Table 4.2, our coupling algorithm is more reliable than competing methods based on the MOTA, MT, and ML scores and comparably accurate based on the MOTP scores. We want to emphasize that S2L2, S1L1-1357 and S1L1-1359 test sequences contain crowds of people with frequent occlusions and were originally intended only for testing methods for density estimation or single object tracking. To the best of our knowledge, very few tracking results have ever been reported on these sequences. Our coupling algorithm can achieve high-quality results and outperform the competing method (OM) consistently in MOTA. We also found that our algorithm is robust because any variations of the parameters of our systems that we tested resulted in a change of the MOTA score that was only (±3%).

Data	Method	#O	MT	ML	MOTA	MOTP
PETS S2L1	OM [6]	23	20	1	0.88	**0.76**
	POM+LP [13]	23	n/a	n/a	≤ 0.6	≤ 0.5
	ILP [7]	23	1	8	0.26	0.67
	our CP	23	**23**	**0**	**0.91** ± 0.03	0.70 ± 0.02
PETS S2L2	OM [6]	75	25	8	0.60	0.61
	our CP	75	**40**	**1**	**0.61** ± 0.03	**0.62** ± 0.02
PETS S1L1-1357	our CP	46	28	0	0.68 ± 0.03	0.56 ± 0.02
PETS S1L1-1359	OM [6]	36	20	7	0.64	**0.67**
	our CP	36	**31**	**2**	**0.86** ± 0.03	0.65 ± 0.02
Infrared S1	RT-1 (Sec.2.3.2)	207	200	0	0.65	8.5 cm
	SDD-RT-1	207	198	0	0.80	4.5 cm
	our CP	207	**201**	0	**0.90**	**4.2** cm
Infrared S2	RT-1 (Sec.2.3.2)	203	147	5	-0.31	10.1 cm
	SDD-RT-1	203	128	6	0.47	7.8 cm
	our CP	203	**171**	**1**	**0.81**	**6.2** cm

Table 4.2: Quantitative results for the coupling algorithm. The OM, ILP, RT, and S-RT trackers sequentially apply the detection and data association modules, while our CP method couples them. MOTA is ideally 1, MOTP also 1 or 0 cm. Results are extracted from published papers. The scores for POM+LP [13] was read from a chart and were based on a different source of ground truth.

The experiments with the infrared data of bats highlight that our SDD detection method can successfully suppress ghost reconstructions in 3D space. Although the reconstruction-tracking algorithm can successfully track most of the objects, it also has a high false positive rate because of the persistent ghost effect during the reconstruction step. This issue is not addressed enough in existing literature probably because only a sparse tracking scenario has been considered so far. Once large groups of objects are under consideration, the need to eliminate ghost effects starts to emerge. By replacing the spatial data association step with our sparsity-driven detector, the overall performance MOTA score can be improved by almost 80%. Moreover, the performance improvement between SDD-RT and CP shows the important impact of our coupling idea.

The variables in the Coupling Algorithm can be optimized separately. In particular, each detection subproblem can be solved independently through parallel computing. The complexity of the L_1 minimization depends on the size of the dictionary, which is determined by the grid size of ground plane or the number of valid triangulation candidates, the number of shape templates and the image size. After the preprocessing step to reduce the dictionary size described earlier, once the problem is constructed, the actual time to solve the linear programming problem is less than a second per frame in our implementation. The data association subproblem can also be solved efficiently even for a large network with one million nodes in our experiment. This is because the network is sparse in our application and the complexity of the minimum-cost flow algorithm is mainly governed by the number of edges. At each iteration of dual decomposition, a re-optimization technique could be applied if available. For our L_1 minimization subproblem, the primal simplex method is adopted and the primal optimal basis is saved to initialize the optimization in the next iteration. Furthermore, we find simple rounding of the LP solution is sufficiently accurate, so additional efforts to pursue the exact integer solution are not needed.

We also find that the Coupling Algorithm does not need to run many iterations before it reaches a good solution. We monitor the tracking quality at each iteration of the subgradient method used by the Coupling Algorithm with two different initializations. If we first run the SDD detector and initialize the network with detection costs only on those nodes selected by our detector, we can expect to see nonzero flows pushed into the network at the first iteration of the Coupling Algorithm. We refer such an initialization scheme as a "good initialization." If we do not set the detection cost, no flow will be pushed at the first iteration, and we call such a scheme a "bad initialization." As shown in Fig. 4·11, it is always beneficial to use a "good initialization" if we are confident in the majority of our detection results. Despite the difference on initialization, the subgradient method used by the Coupling Algorithm always presents fast improvement at the first few rounds of iterations but with relatively slow convergence. This kind of behavior is also observed in other optimization work [72]. In practice, an early stop (25 iterations in our experiments) is sufficient for producing a good suboptimal solution. Other heuristic stopping criteria could also be used. Trackers, such as ILP, RT, and S-RT, which apply the detection and

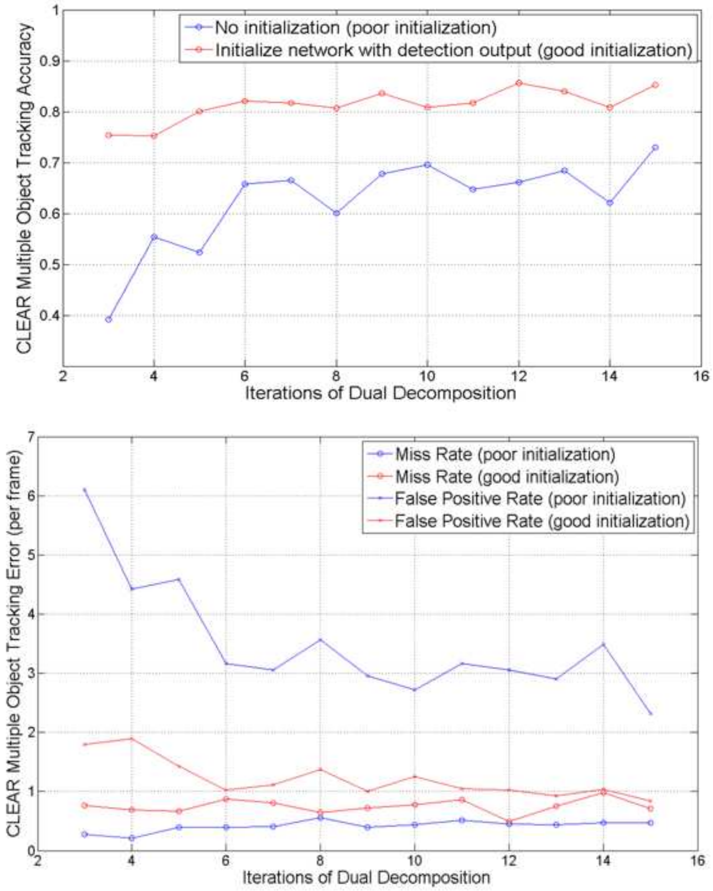


Figure 4·11: Performance of the coupling algorithm on PETS-S1L1-1359 at each iteration with different initialization. The MOTA score, false positive rate, miss detection rate change quickly at the first few iterations. Unlike the traditional "detection-tracking scheme" that reports a fixed detection rate for the detector, here we have a dynamic performance on detection.

data association modules sequentially could be considered to perform the first iteration of our coupling algorithm. The results in Fig. 4·11 seem to indicate that the performance of these trackers may increase significantly with additional iterations, if they were placed within our coupling framework.

4.4 Summary and Discussion

In this chapter, we presented a novel multiple-object tracking framework that couples object detection and data association. The new objective function was derived from Bayesian estimation theory (4.1), taking advantage of the often complementary nature of the two subproblems. As a concrete example, our Coupling Algorithm combines a sparsity-driven detection method and a network-flow data-association method within this framework (4.13). Our sparsity-driven detection enables us to model the likelihood of the entire image so we could eschew common heuristics such as non-maximum suppression. Moreover, the sparsity constraint also successfully reduces the "ghost effect" that can occur in 3D multi-view tracking. An extension of such detector that considers both sparseness and quantization is used to infer the occlusion relationship, which is represented by occlusion layers, to detect partially visible objects purely from binary images. Through dual decomposition (4.16), a coupled objective function is optimized iteratively with off-the-shelf efficient algorithms for each subproblem. The experiments with both monocular and multi-view datasets show that coupling detection and data association can improve tracking performance compared to the results of sequentially applying each module.

To evaluate the scalability of the proposed method, we need to consider the processing complexity of our system, which largely determined by the size of the dictionary. This is proportional to the number of shape templates, the image size, and the number of grid blocks on the ground plane or valid triangulations. For the datasets we considered, the running time of our system was in the order of a few seconds per frame with a Matlab implementation. Additional efforts should be made to speed up the implementation in scenarios where objects may have large variation of poses or a fine 2D or 3D granularity is needed. Moreover, only binary pixels are used in our detector, which is not sufficient for object localization if objects are in dense formations. Combining gradient features with binary shape templates has been proven to be effective in the object detection literature [95]. Output from these detectors could be used to introduce a bias on which codeword to select.

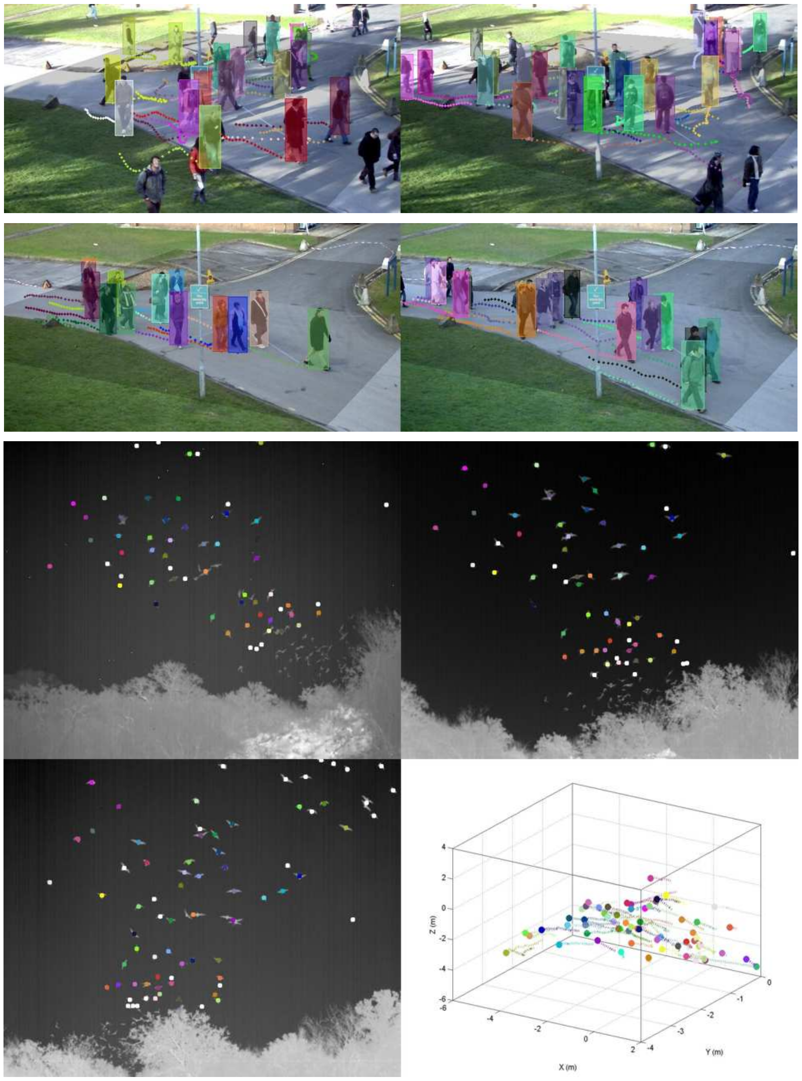


Figure 4·12: Tracking results for the Coupling Algorithm. The first two rows show sample frames and trajectories from PETS S2L2 and S1L1-1359 sequences. The last two rows show sample frames and 3D trajectories from infrared sequence S2.

Chapter 5

Conclusions and Future Work

In this final chapter of the thesis we summarize the main contributions and open issues in the work that we have described. Following that, we will point out some interesting directions for future research.

5.1 Main Contributions

We proposed three categories of algorithms to address the occlusion reasoning problem for tracking large groups of objects imaged with low resolution. In order to recover and track occluded objects, different sources of additional information were used, such as additional views in the reconstruction-tracking algorithm, more temporal evidence in the track linking algorithm, and the interaction between detector and tracker in the coupling algorithm. A summary of our sparsity-driven detectors (LDND and LDQD) and three categories of tracking algorithms is given in Table 5.1 and Table 5.2. Competing methods applied on the same datasets are also listed for comparison. Compared to existing work, our methods lead to computationally tractable formulations such as set-cover, network-flow, or L_1 minimization problems, where efficient off-the-shelf polynomial-time algorithms are available.

To the best of our knowledge, we are the first to present an online multi-object multi-view algorithm (SDD+RT) that is able to track large groups of flying animals and produce high-quality trajectories for further scientific research. We highlight the track graph representation and unify various fast, simple, yet effective linking strategies under the same framework. We also developed a novel coupling framework that combines detection and data association modules, which achieved the best performance on our own and publicly

Table 5.1: Summary of detection approaches

Method	Background model	Sliding window	Non-maximum suppression	Occlusion estimation	Core Algorithm
Our approaches					
LDND	yes	no	no	no	Linear programming
LDQD	yes	no	no	yes	Linear programming
Competing approaches					
MCMC [42]	yes	no	no	no	MCMC sampling
ASEF [19]	no	yes	yes	no	Correlation
Cascade [83]	no	yes	yes	no	Boosting Classifier
LSVM [39]	no	yes	yes	no	SVM Classifier

available datasets.

For object detection, if objects do not have many pose variations, we recommend our LDQD detector that simultaneously optimizes the selection of hypotheses and estimates the occlusion layer. For object tracking, if large groups of objects frequently occlude each other, we recommend the Coupling Algorithm (CP) that combines the LDQD and network-flow association method; if a set of reliable tracklets can be generated by low level trackers, we encourage to use the global set-cover linking algorithm to further reduce track fragmentation errors. In regards to computation complexity for LDQD, our current Matlab implementation with Cplex LP solver processes each video frame in 2 s for planar-motion dense sequence, 5 s for 3D-motion dense sequence on an Intel Xeon 3.2 GHz machine. About 95% of the computation is expended on building the dictionary and the problem reduction, especially for the 3D case where each triangulation involves solving a linear system. These preprocessing steps could be significantly speeded up with GPU support. Once the minimization problems have been set up, our Matlab implementation of the Coupling Algorithm can process each video frame for the dense tracking scenario in 3 s at each iteration of the dual decomposition without parallel computing.

Table 5.2: Summary of tracking approaches

Method	Type	Applicable to long-term occlusion	Applicable to large groups	Core Algorithm
Our approaches				
RT	Sequential	yes	no	Greedy randomized adaptive search
SDD+RT	Sequential	yes	yes	Greedy randomized adaptive search
Loc-linking	Batch	no	no	Minimum-flow Bipartite matching
Net-linking	Batch	yes	yes	Minimum-cost flow
Sc-linking	Batch	yes	yes	Greedy set-cover
Mv-linking	Batch	yes	yes	Greedy set-cover
CP	Batch	yes	yes	Linear programming Minimum-cost flow Dual decomposition
Competing approaches				
JPDA	Sequential	no	no	Probabilistic filter
MHT	Sequential	no	no	Greedy randomized adaptive search
DBN [60]	Batch	yes	no	Junction-tree
POM [13]	Batch	yes	no	Linear programming
OM [6]	Batch	yes	yes	Extended Kalman filter Non-convex minimization

5.2 Limitations and Future Work

The key technical problem in this thesis is how to select a subset from a pool of competing hypotheses. These hypotheses are competing with each other because they share the same image evidence due to occlusions. In the tracking context, this is related to the problem of estimating the number of objects, and the true state of each object. We have attempted an iterative greedy method (Chapter 2), maximum-likelihood estimation (Chapter 3), and maximum-a-posteriori estimation (Chapter 4). These methods are only tested on video sequences where a foreground/background separation can be made. More analysis is required to examine how to extend these methods to a more general setting. Specifically, we need to solve the following technical issues:

How to generate hypotheses. The step of generating hypotheses usually involves proposing a possible location of an object or track, and evaluating its likelihood. Due to the underlying combinatorial nature of the problem, the number of hypotheses might grow exponentially, and evaluating the likelihood for each of them is even more computationally expensive. Although there has been interesting work on how to produce a *single* optimal hypothesis efficiently for the pose estimation problem [79], currently we have seen very little activity in the research area of efficiently producing a set of informative but competing track hypotheses. A data-driven sampling-based technique might be a solution.

How to represent the object. The benefit of using a binary foreground pixel as our feature representation is that the same pixel can be naturally explained by occluder and occludee. This is clearly not the case when a more advanced feature representation is used, for example, based on intensity gradient or color information. However, the hypothesis of an occluded object can still "imagine" its appearance of the occluded part by synthesizing corresponding features from the template. How to incorporate feature detection and synthesis into occlusion reasoning is a fresh new question. Any reasonable solution for this task would be beneficial for a broader class of computer vision problems. Another related question is if it is possible to track objects in 2D only so we do not need to consider camera calibration information. Currently this is a necessary input in our coupling algorithm and sparsity-driven detector. With the 3D-to-2D mapping, we introduce the competition scheme where the hypotheses of 3D locations "compete" for the 2D measurements. If our hypotheses only infer 2D locations of objects, an alternative competition scheme should be developed in order to retrieve a sparse output.

How to speed up the optimization. Although the computation of the optimization part of our experiments is not intensive, there is still room to improve its efficiency. For example, if the object shows a strong cyclic motion pattern, which, for example, a flying bat exhibits, its pose/shape in the next frame will be quite predictable. In this situation, a prediction on which pose template to select can be used as an initial solution instead of searching from scratch. In this thesis, we have not explored how to make use of cyclic pose prediction in tracking and how much it can speed up the optimization. This step could be crucial if a fast sequential tracking algorithm is required.

Finally, throughout this thesis, we have frequently used the assumption that objects in the scene are moving independently, although their observations may be dependent. If the objects present strong group behavior or they are physically connected with kinematic constraints, it is better to introduce high-order dependence among object states. It would be interesting to see how to encode these high-order dependencies in our methods and how important this would be.

References

[1] V. Ablavsky and S. Sclaroff. Layered graphical models for tracking partially-occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1758–1775, 2011.

[2] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments. In *Proceeding of the IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2009.

[3] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Proceeding of the 10th European Conference on Computer Vision (ECCV)*, pages 1–14, 2008.

[4] E. Andrade, S. Blunsden, and R. Fisher. Performance analysis of event detection models in crowded scenes. In *Proceedings of the Workshop "Towards Robust Visual Surveillance Techniques and Systems" at Visual Information Engineering 2006*, pages 427–432, Bangalore, India, September 2006.

[5] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[6] A. Andriyenko, S. Roth, and K. Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *Proceeding of the 11th IEEE Workshop on Visual Surveillance*, pages 1839 – 1846, 2011.

[7] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *Proceeding of 11th European Conference on Computer Vision (ECCV)*, pages 466–479, 2010.

[8] S. Bak, D. P. Chau, J. Badie, E. Corvee, F. Bremond, and M. Thonnat. Multi-target tracking by discriminative analysis on riemanian manifold. In *Proceeding of the 19th International Conference on Image Processing (ICIP)*, 2012.

[9] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, 105(4):1232–1237, 2008.

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

[11] Y. Bar-Shalom and X. R. Li. *Multitarget - Multisensor Tracking: Principles and Techniques.* YBS Publishing, 1995.

[12] O. Barinova, V. Lempitsky, and P. Kohli. On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1773–1784, 2012.

[13] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Proceeding of the IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2009.

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

[15] D. P. Bertsekas. *Linear Network Optimization: Algorithms and Codes.* MIT Press, 1991.

[16] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999.

[17] D. P. Bertsekas and D. A. Castañón. A forward/reverse auction algorithm for asymmetric assignment problems. *Computational Optimization and Applications*, 1(3):277–297, December 1992.

[18] 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 *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[19] D. S. Bolme, B. A. Draper, and J. R. Beveridge. Average of synthetic exact filters. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 21052112, 2009.

[20] P. T. Boufounos and R. G. Baraniuk. 1-bit compressive sensing. In *Proceedings of Conference on Information Science and Systems (CISS)*, March 2008.

[21] G. J. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 594–601, 2006.

[22] R. G. Brown and P. Y. C Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, Inc, 1997.

[23] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transaction on Information Theory*, 51(12):4203–4215, 2005.

[24] D. A. Castañón. Efficient algorithms for finding the k best paths through a trellis. *IEEE Transaction on Aerospace and Electronic Systems*, 26(2):405–410, 1990.

[25] G. Castañón and L. Finn. Multi-target tracklet stitching through network flows. In *Proceeding of the IEEE Aerospace Conference*, March 2011.

[26] D. M. Chu and A. W. M. Smeulders. Thirteen hard cases in visual tracking. In *Proceeding of the IEEE International Confenence on Advanced Video and Signal-Based Surveillance*, pages 103–110, August 2010.

[27] Columbus large image format 2007 data. https://www.sdms.afrl.af.mil/, 2007.

[28] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[29] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, February 1996.

[30] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. A generalized s-d assignment algorithm for multisensor-multitarget state estimation. *IEEE Transaction on Aerospace and Electronic Systems*, 33(2):523–538, 1997.

[31] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[33] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743761, 2012.

[34] A. Ellis, A. Shahrokni, and J. M. Ferryman. Pets2009 and winter-pets 2009 results: a combined evaluation. In *Proceeding of the IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2009.

[35] U. M. Erdem and S. Sclaroff. Event prediction in a hybrid camera network. *ACM Transactions on Sensor Networks (TOSN)*, 8(2), 2012.

[36] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[37] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303 – 338, June 2010.

[38] M. Feldmann, D. Frnken, and W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.

[39] P. Felzenszwalb, R. 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, September 2010.

[40] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008.

[41] H. Gauvrit and J. P Le Cadre. A formulation of multitarget tracking as an incomplete data problem. *IEEE Transaction on Aerospace and Electronic Systems*, 33(4):1242–1257, 1997.

[42] W. Ge. *Bayesian Analysis of Small Groups in Crowds*. PhD thesis, The Pennsylvania State University, USA, 2010.

[43] W. Ge and R. T. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *Proceeding of British Machine Vision Conference (BMVC)*, 2008.

[44] W. Ge and R. T. Collins. Crowd detection with a multiview sampler. In *Proceeding of the 11th European Conference on Computer Vision (ECCV)*, pages 1 – 8, 2010.

[45] R. Girshick, P. Felzenszwalb, and D. Mcallester. Object detection with grammar models. In *Proceeding of the 24th Advances in Neural Information Processing Systems (NIPS)*, pages 442–450, 2011.

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

[47] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceeding of the 10th European Conference on Computer Vision (ECCV)*, pages 788–801, 2008.

[48] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[49] J. Kang, I. Cohen, and G. Medioni. Persistent objects tracking across multiple non-overlapping cameras. In *Proceeding of IEEE Workshop on Motion and Video Computing (WMVC)*, 2005.

[50] S. M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceeding of the 9th European Conference on Computer Vision (ECCV)*, volume 3954, pages 133–146, 2006.

[51] Z. Khan, T. Balch, and F. Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1960 – 1972, December 2006.

[52] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *Proceeding of the International Conference on Computer Vision (ICCV)*, 2007.

[53] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1683–1698, October 2008.

[54] K. Li, M. Chen, T. Kanade, E. Miller, L. Weiss, and P. Campbell. Cell population tracking and lineage construction with spatiotemporal context. *Medical Image Analysis*, 12(1):546–566, October 2008.

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

[56] D. Lin and J. Fisher. Efficient sampling from combinatorial space via bridging. In *Proceeding of the 15th Conferences on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[57] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, pages 383 – 390, 1975.

[58] Isard M. and Blake A. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

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

[60] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking: Linking identities using bayesian network inference. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2187–2194, 2006.

[61] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple target tracking problems. In *Proceeding of the 43rd IEEE Conference on Decision and Control*, pages 1–8, 2004.

[62] K. Otsuka and N. Mukawa. Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 90–97, 2004.

[63] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 666–673, 2006.

[64] Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), datasets for crowd image analysis, crowd count and density estimation, tracking of individual(s) within a crowd, and detection of separate flows and specific crowd events. http://www.cvg.rdg.ac.uk/PETS2009, June 2009.

[65] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2011.

[66] 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.

[67] A. B. Poore and A. J. Robertson. A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems. *Computational Optimization and Applications*, 8(2):129–150, 1997.

[68] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *Proceeding of the 11th European Conference on Computer Vision (ECCV)*, 2010.

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

[70] J. Renno, D. Greenhill, J. Orwell, and G. Jones. Occlusion analysis: Learning and utilising depth maps in object tracking. *Image and Vision Computing*, 26(3):430441, march 2008.

[71] 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.

[72] B. Savchynskyy, J., S. Schmidt, and C. Schnörr. A study of Nesterov's scheme for Lagrangian decomposition and map labeling. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[73] K. Shafique, M. W. Lee, and N. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[74] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 421–428, 2004.

[75] K. Smith, D. Gatica-Perez, and J.M. Odobez. Using particles to track varying numbers of objects. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2005.

[76] B. Song and A. Roy-Chowdhury. Robust tracking in a camera network: A multi-objective optimization framework. *IEEE Journal on Selected Topics in Signal Processing: Special Issue on Distributed Processing in Vision Networks*, 2(4):582 – 596, August 2008.

[77] R. L. Streit and T. E. Luginbuhl. A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning. In *Proceeding of the 6th Joint Service Data Fusion Symposium*, pages 1015–1024, June 1993.

[78] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Proceeding of the 17th Advances in Neural Information Processing Systems (NIPS)*, 2004.

[79] Taipeng Tian. *Efficient Techniques of Recovering 2D Human Body Poses from Images*. PhD thesis, Boston University, USA, 2011.

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

[81] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[82] A. Veeraraghavan, R. Chellappa, and M. Srinivasan. Shape-and-behavior-encoded tracking of bee dances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):463 – 476, March 2008.

[83] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 511518, 2001.

[84] First joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), datasets for outdoor people tracking. ftp://ftp.pets.rdg.ac.uk/pub/VS-PETS, October 2003.

[85] X. G. Wang, K. T. Ma, G. W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using a nonparametric bayesian model. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[86] B. Wu. *Part based Object Detection, Segmentation, and Tracking by Boosting Simple Feature based Weak Classifiers*. PhD thesis, University of South California, USA, 2008.

[87] Z. Wu, M. Betke, and T. H. Kunz. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1185–1192, 2011.

[88] Z. Wu, D. Guraril, J. Y. Wong, and M. Betke. Hierarchical partial matching and segmentation of interacting cells. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012: 15th International Conference, Proceedings*, Nice, France, October 2012.

[89] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke. Tracking a large number of objects from multiple views. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1–8, Kyoto, Japan, September 2009.

[90] Z. Wu, N. I. Hristov, T. H. Kunz, and M. Betke. Tracking-reconstruction or reconstruction-tracking? comparison of two multiple hypothesis tracking approaches to interpret 3d object motion from several camera views. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WMVC)*, pages 1–8, Utah, December 2009.

[91] Z. Wu, A. Thangali, S. Sclaroff, , and M. Betke. Coupling detection and data association for multiple object tracking. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[92] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1200 – 1207, 2009.

[93] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal Markov Chain Monte Carlo data association. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[94] T. Yu, Y. Wu, N. O. Krahnstoever, and P. H. Tu. Distributed data association and filtering for multiple target tracking. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[95] Q. Yuan. *Learning a family of detectors*. PhD thesis, Boston University, USA, 2010.

[96] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[97] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *Proceeding of the 17th International Conference on Pattern Recognition (ICPR)*, pages 28–31, 2004.

[98] D. P. Zou, Q. Zhao, H. S. Wu, and Y. Q. Chen. Reconstructing 3d motion trajectories of particle swarms by global correspondence selection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1578–1585, 2009.

Curriculum Vitae

Zheng Wu
Department of Computer Science
Boston University
111 Cummington St
Boston, MA 02215
wuzheng@cs.bu.edu
http://cs-people.bu.edu/wuzheng

Research Interests

Computer Vision and Machine Learning

Education

Ph.D.	Computer Science, Boston University. Requirements completed 2012, degree conferred 2013. *Occlusion Reasoning for Multiple Object Visual Tracking* Advisor: Margrit Betke
M.S.	Computer Science, Zhejiang University, 2006. *Image Registration by Hierarchical Feature Matching* (in Chinese) Advisor: Miaoliang Zhu
B.S.	Computer Science, Chu Kochen Honors College, Zhejiang University, 2003.

Academic Experience

Research Assistant, Boston University
Department of Computer Science
2006 – present
Image and Video Computing Group

Multiple Object Tracking, 3D Reconstruction, Medical Image Segmentation, mentored undergraduate and masters students, and assisted in grant writing.

- **Infrared Thermal Video Analysis of Wild Animals**
  This project is to census wild animals like Brazilian free-tailed bats for understanding the ecological and economic impact of these animals on terrestrial ecosystems. Several

interesting problems need to be solved, such as automatic counting and tracking a large number of objects in infrared thermal videos from single or multiple cameras; gesture recognition in spatial-temporal domain; statistical analysis on group behavior; multiple-sensor fusion, etc.

- **Tracking of Cell Populations in Response to Physical Stimuli**
  This project is to reveal the behavioral response of fibroblast cells to changes in hydrogel conditions. The system has to track unknown number of cells in time-lapse phase-contrast microscopy images, and identify cell division/death as well as morphology deformation under different stimuli.
- **Hand Gesture Recognition and Tracking**
  This project is to segment and track articulated object like human hand with highly cluttered background to assist American Sign Language Recognition. The challenge is to build a robust and near real-time system to recognize hand gesture in uncontrolled environment and handle self-occlusion of fingers in 2D as well.

Summer Intern, Microsoft Research New England
2010.5 – 2010.8

Large-scale Object Detection, Part-based Model, Active Learning.

Teaching Assistant, Boston University
Department of Computer Science
CS131 Combinatoric Structures, 2011

Lecturer for lab sessions. Designed teaching materials on combinatorial algorithms and proofs.

CS440/640 Artificial Intelligence

Designed and graded homework assignments and held tutoring hours

CS585 Image and Video Computing, 2008

Designed and graded homework assignments and held tutoring hours

Awards

Research Achievement Award for 2011-2012, Computer Science, Boston University
This award is offered annually to one Ph.D. student in the department, based on their research accomplishments.

Invited Talk

"*Occlusion Reasoning for Multiple Object Tracking*", at computer vision group, Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA. May 2011.

Publications

1. **Z. Wu**, D. Guraril, J. Y. Wong and M. Betke. "Hierarchical Partial Matching and Segmentation of Interacting Cells." In Proceeding of the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Nice, France, 2012.

2. **Z. Wu**, A. Thangali, S. Sclaroff, and M. Betke. "Coupling Detection and Data Association for Multiple Object Tracking." In proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, June, 2012.

3. **Z. Wu**, M. Betke and T. H. Kunz. "Efficient Track Linking Methods for Track Graphs Using Network-flow and Set-cover Techniques." In proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Springs, Colorado, June, 2011.

4. D. H. Theriault, **Z. Wu**, N. I. Hristov, S. M. Swartz, K. S. Breuer, T. H. Kunz, and M. Betke. "Reconstruction and analysis of 3D trajectories of Brazilian free-tailed bats in flight." In Proceeding of IEEE Workshop on Visual Observation and Analysis of Animal and Insect Behavior, in conjunction with the 20th International Conference on Pattern Recognition, Istanbul, Turkey, August, 2010.

5. J. Magee, **Z. Wu**, H. Chennamaneni, S. Epstein, D. H. Theriault, and M. Betke. "Towards a Multi-camera Mouse-replacement Interface." In proceeding of the 10th International Workshop on Pattern Recognition in Information Systems (PRIS). In conjunction with ICEIS 2010, Madeira, Portugal, June, 2010.

6. **Z. Wu**, N. I. Hristov, T. H. Kunz, and M. Betke. "Tracking-Reconstruction or Reconstruction-Tracking? Comparison of Two Multiple Hypothesis Tracking Approaches to Interpret 3D Object Motion from Several Camera Views." In Proceeding of IEEE Workshop on Motion and Video Computing (WMVC), Utah, December, 2009 (Oral).

7. **Z. Wu**, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke. "Tracking a Large Number of Objects from Multiple Views." In Proceeding of the 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, September, 2009 (PAMI Student Support).

8. D. House, M. L. Walker, **Z. Wu**, J. Y. Wong, and M. Betke. "Tracking of Cell Populations to Understand their Spatio-Temporal Behavior in Response to Physical Stimuli." In Proceeding of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA), in conjunction with CVPR 2009, Miami, Florida, June, 2009.

9. **Z. Wu**, M. Betke, J. Wang, V. Athitsos, and S. Sclaroff. "Tracking with Dynamic Hidden-State Shape Models." In Proceeding of the 10th European Conference on Computer Vision (ECCV), Marseille, France, October, 2008.

10. D. Wang, C. Cui, **Z. Wu**. “Matching 3D Models with Global Geometric Feature Map.” In Proceeding of IEEE 12th International Conference on Multimedia Modeling, Beijing, January, 2006.

Professional Activities

Reviewer for:

IEEE Transaction on Pattern Analysis and Machine Intelligence.

Journal of Signal, Image, and Video Processing.

Journal of Medical Physics.

International Journal of Computers and Applications.

International Symposium on Biomedical Imaging.

IEEE Conference on Computer Vision and Pattern Recognition

European Conference on Computer Vision

IEEE International Conference on Computer Vision

International Conference on Medical Image Computing and Computer Assisted Intervention

Journal of Field Robotics

Journal of Personal and Ubiquitous Computing

Journal of Universal Access in the Information Society

Boston University Image and Video Computing Group

Group Meeting Co-coordinator, 2008-2009

Student Volunteer, 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2010