Global Optimization for Coupled Detection and Data Association in Multiple Object Tracking

Zheng Wu*, Margrit Betke

Computer Science Department, Boston University, Boston, MA02215, USA

Abstract

We present a novel framework for tracking multiple objects imaged from one or more static cameras, where the problems of object detection and data association are expressed by a single objective function. Particularly, we combine a sparsity-driven detector with the network-flow data association technique. The framework follows the Lagrange dual decomposition strategy, taking advantage of the often complementary nature of the two subproblems. Our coupling formulation avoids the problem of error propagation from which traditional “detection-tracking approaches” to multiple object tracking suffer. We also eschew common heuristics such as “non-maximum suppression” of hypotheses by modeling the joint image likelihood as opposed to applying independent likelihood assumptions. Our coupling algorithm is guaranteed to converge and can resolve the ambiguities in track maintenance due to frequent occlusion and indistinguishable appearance between objects. Furthermore, our method does not have severe scalability issues but can process hundreds of frames at the same time. Our experiments involve challenging, notably distinct datasets and demonstrate that our method can achieve results comparable to or better than those of state-of-art approaches.

Keywords: Sparsity-driven detection, Network-flow data association, 1-bit de-quantization, Dual decomposition, Multiple object tracking

*Corresponding author at: Computer Science Department, 111 Cummington Mall, Boston University, MA02215, USA. Email: wuzheng@bu.edu.

1. Introduction

The interpretation of the motion of large groups of individuals is a challenging problem in computer vision. A complete multi-object tracking system typically consists of three phases: object detection, temporal data association
5 (i.e., the assignment of current observations to object tracks) and an optional step of state estimation for each object. These modules are usually arranged sequentially in the system so each subproblem itself is optimized independently. However, due to the lack of robust methods for object detection in uncontrolled environments and poor scalability to large numbers of objects, current tracking
10 systems do not generalize well on challenging datasets. Most previous efforts have therefore followed two distinct directions of research: 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 detection and data association tasks.

15 The detection-tracking design has the inherent weakness that it requires the output of the detection module to be reliable in order for the data association module to work properly. Detection errors such as “false alarms” and “missed detections” otherwise propagate to the data association module and false matches need to be corrected later. In contrast, we show that error propagation from
20 detection to data association can be avoided if both tasks, detection and data association, are combined into a single module and solved simultaneously by optimizing a single objective function. In addition, we show that temporal information could significantly improve the performance of an object detector in our coupling framework. The coupling idea appears attractive but introduces
25 new challenges as well:

1. What type of objective function should be used? Many existing detection methods have not even been formalized as an optimization problem.
2. How can the new objective function be solved? Many current data association methods are complicated, approximate solutions to intractable
30 problems. A new objective function that couples detection and data as-

sociation 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.

In this paper, 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 [1] for the dense-object tracking scenario when a foreground estimation 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 of 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. 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 [2].

Unlike previous coupling formulations that rely on a coordinate descent technique [3], our overall objective function has a simple form and can be solved through Lagrange dual decomposition. The method distributes the coupling formulation to subproblems and coordinates their local solutions to achieve a globally optimal solution. For each subproblem, a very efficient off-the-shelf algorithm is available. The proposed paradigm also permits distributed computing. Finally, our method was tested both for monocular and multi-view videos, and achieved consistent robustness across several challenging, notably distinct datasets. In summary, our contributions are:

1. A novel and flexible framework for coupling the subproblems of detection and data association of multiple-object tracking in a single objective function, which permits using a straightforward global optimization technique.

2. A new sparsity-driven object detection method that takes binary foreground/background segmentation as input, infers mutual occlusion relationships, and achieves a high detection rate even when objects are severely occluded.
- 65 3. A general mechanism to suppress false alarms without “non-maximum suppression” either in the detection or in the data association stage.

2. Related Work

An important motivation for designing the coupling framework is to perform occlusion reasoning, a major bottleneck of current tracking systems. In this paper, we focus on tracking multiple objects that are imaged with limited resolution from one or more static cameras and may partially occlude each other. It is also possible to encode a scene-specific occlusion model [4] into our formulation.

2.1. Occlusion Reasoning in Detection

75 There are two main challenges for occlusion reasoning in object detection. First, when an object becomes occluded and there is no or only a partial observation of the object on the image plane, model-based detectors must deal with difficult-to-predict uncertainty. Second, the heuristic “non-maximum suppression” technique, adopted by most detection methods, which aims to cluster close hypotheses, also explains away true detections.

80 The first challenge may be addressed by a part-based detector, which may be able to detect certain partially occluded objects, as long as the visible part appears with sufficient resolution [5, 6, 7], and the detector can be adjusted online [8, 9, 10, 11]. Part-based detectors fail when objects are completely occluded or the resolution of an object is too small. Even for pedestrians, an object category well-studied in the computer vision community, the performance of the current state-of-art method [12] drops significantly under partial occlusion and degrades catastrophically for small resolution. Moreover, direct

modeling of occlusion is difficult in general, as the degree of partial occlusion
 90 needs to be explicitly expressed in the object model. However, a detailed object
 model/detector is not necessary for many surveillance applications. Sometimes,
 it is not even useful due to limited resolution or challenging imaging condi-
 tions. As an alternative, when a reasonable background subtraction method is
 available, a common idea is to fit binary shape templates to the observations
 95 with the help of scene knowledge, such as camera calibration or multiview ge-
 ometry [13, 14, 15, 16]. These methods all rely on a background subtraction
 preprocessing step (which can be a difficult problem to develop). Therefore,
 they are sensitive to the quality of background subtraction and the degree of
 partial occlusion.

100 The side effect of the non-maximum suppression technique to filter out true
 detections is particularly undesirable when objects have a large overlap on the
 image plane. Instead of letting this heuristic step make ad-hoc decisions or
 trying to tune parameters for it, a number of recent works have shown that it
 is beneficial to formulate the object detection problem as a global optimization
 105 problem with a Minimum Description Length (MDL) constraint or a context
 prior [17, 18, 16], and let the optimization process determine which hypotheses
 to select. Our detection methods used in the coupling framework fall into this
 category.

2.2. Occlusion Reasoning in Tracking

110 Despite efforts to detect partially occluded objects, missed detection/false
 positives are still inevitable, and any ambiguity could be resolved in the data
 association stage. Research efforts for multiple object tracking typically treat
 occluded objects as missed detection events or track occluded objects all togeth-
 er with a single tracker and iteratively grow or stitch tracklets (track fragments)
 115 before and after occlusions [19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. The main ef-
 fort in this research direction is to design a discriminant similarity measure for
 the tracklet matching problem so that occlusion is implicitly resolved by filling
 gaps after stitching tracklets. It is important to note that all these approaches

follow the “detection-tracking strategy” and therefore rely on good detectors for
120 initialization. The limited ability through tracking to correct the error propa-
gated from the detection stage typically implies that the missed detection/false
positive events are assumed to occur rarely. However, if this assumption does
not apply, hoping that the data association module itself will fix all detection
errors is not promising.

125 Explicit occlusion modeling also appears in an optimization work by An-
driyenko et al. [29], 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. Our formulation
130 avoids these complications, is mathematically rigorous and simpler to optimize.

Another category of occlusion reasoning is to combine both the detector’s
output and motion flow estimation [30, 31]. The intuition is that the dense
motion flow is usually less affected by partial occlusion than an object detector
so that clustering the low-level trajectories can help improve the overall tracking
135 performance when objects are only partially visible. However, motion flow itself
is not able to distinguish individuals in a crowd with coherent motion. It also
significantly increases the amount of computation.

2.3. Relation to the Proposed Method

As occlusion cannot be resolved solely in the detection or data association
140 phases, 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 [32] that
combines tracking and detection, and models both the approximate articula-
tion of each person as well as the temporal coherency within a walking cycle.
145 An extended part-based model has also been proposed to build a “joint people
detector” that combines a state-of-the-art single person detector with a detector
for pairs of people [11]. While such detailed part-based models offer a princi-
pled way to handle inter-person occlusions, the richness of the representations

requires sufficient resolution so that the part appearance can be properly modeled.
150

Another coupling idea for pedestrian tracking was proposed by Leibe et al. [3], 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 not easy to generalize to other
155 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 subjected to a good initialization as well.

In contrast to the work by Leibe, we base our method on the foundations
160 of Bayesian estimation theory. Our objective function has relatively lower complexity (linear programming) and is straightforward to extend to higher-order cases. Such a formulation has better scalability, which will be demonstrated experimentally. The proposed framework is general in the sense that each sub-problem can be easily substituted by other classic approaches.

165 It is worth mentioning that we adopt the network-flow formulation to model the data association subproblem in this paper. While there are many other variants of this type of modeling in literature [25, 33, 34, 35, 36], we are only interested in its original form here [2] for its low computational costs. We extend the model to account for the multi-category case where multiple layers of the
170 network need to be constructed with a set of new linear constraints. Again, we would like to keep the formulation easy and to be solved as a subproblem, but we do not exclude the possibility to incorporate other complicated models. The overall optimization scheme is a dual decomposition scheme [37], which has been used in solving other computer vision problems, such as MRF-based image
175 segmentation [38] and feature correspondence [39]. Here we demonstrate how it can be used to solve a multi-object tracking problem.

To position the novelty of our work, we emphasize that our framework is based on binary image interpretation so we are less limited by the specifics of the imaging modality or the resolution of the objects. Our main interest is the

180 general multi-object tracking problem, and our proposed methods solve tasks
beyond tracking pedestrians in visible-light cameras. Specifically, we will show
results of our methods applied to tracking bats, runners, and bicycles in videos
recorded by infrared and visible-light cameras, with single or multiple views.
All these visually distinctive scenarios are formulated in the same framework.

185 A preliminary version of this work was described by Wu et al. [16], and was
found to have limitations when severe mutual occlusion happens. To address
these limitations, we here give an enhanced version of a sparsity-driven detec-
tor for binary image interpretation. We also provide additional quantitative
comparisons across several datasets, as well as implementation details and new
190 analysis.

3. Our Approach

3.1. Bayesian Formulation

Figure 1: The graphical model for the multiple object tracking problem. Left: Existing methods model the image likelihood with the independent assumption, that is, the image evidence y_i for each object x_i is generated independently, i.e. $p(\mathbf{Y}|\mathbf{X}) = \prod_i p(y_i|x_i)$; Right: Our method models the image evidence $\mathbf{Y}$ to be jointly determined by **all** objects, i.e. $p(\mathbf{Y}|\mathbf{X})$ cannot be factorized further.

We formulate the multiple object tracking problem as a maximum-a-posteriori estimation problem. Given a collection $\mathbf{Y}$ of image evidence, we estimate the
195 state of all objects $\mathbf{X}$ in the scene as follows:

$$\max_{\mathbf{X}} p(\mathbf{X}|\mathbf{Y})$$

$$\begin{aligned}
& \propto \max_{\mathbf{X}} p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}) \\
& = \max_{\mathbf{X}} \prod_t p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_1) \prod_t p(\mathbf{X}_t|\mathbf{X}_{t-1}) \\
& = \max_{\mathbf{X}} \prod_t p(\mathbf{Y}_t|\mathbf{X}_t) \prod_i p(x_{i,1}) \prod_t p(x_{i,t}|x_{i,t-1}) \tag{1}
\end{aligned}$$

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 relationship between objects and handle occlusions. The graphical model for our generative process is depicted in Fig. 1.

Without modeling the likelihood for the entire image but instead making certain independence assumptions, one can further factorize the first term of Eq. 1, a technique used by most earlier tracking approaches [20], see Fig. 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 [17]. 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. 1, we rewrite the optimization problem as follows:

$$\min_{\mathbf{X}_1, \mathbf{X}_2} [g(\mathbf{X}_1, \mathbf{Y}) + h(\mathbf{X}_2)], \quad s. t. \mathbf{X}_1 = q(\mathbf{X}_2). \tag{2}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are two copies of hidden state variables, 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

Figure 2: Overview of our batch tracking system. We first explain our sparsity-driven detection method in Section 3.2 and the network-flow data association method in Section 3.3. We describe the coupling framework and dual decomposition techniques in Section 3.4. Details on how our system learns the codebook and how we set up the parameters can be found in Section 4.

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 Eq. 2 is to allow two subproblems to utilize different sources of image evidence $\mathbf{Y}_1, \mathbf{Y}_2$:

$$\min_{\mathbf{X}_1, \mathbf{X}_2} g(\mathbf{X}_1, \mathbf{Y}_1) + h(\mathbf{X}_2, \mathbf{Y}_2), \quad s. t. \quad \mathbf{X}_1 = q(\mathbf{X}_2). \quad (3)$$

Eq. 2 is a classic setup in operation 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 [38]. In the remainder of our paper, we will show that the coupling formulation is also useful for solving the tracking problem. We first define functions g and h in Section 3.2 and 3.3 respectively, by giving specific examples of detection and data association methods. An overview of our tracking system is given in Fig. 2.

Figure 3: (a) 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). (b) Two binary shape templates for front and side views of pedestrian. (c) 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. (d) Five binary shape templates for flying bats while each template consists of the same pose in three views.

3.2. Sparsity-driven Detector

Inspired by the sparsity-driven people localization method proposed by Alahi et al. [13], 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

collecting all codewords in discretized 3D space, we build the dictionary $\mathbf{D}$ for a particular category of objects, see Fig. 3. 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
245 dictionary can be performed offline. But for tracking objects in a 3D volume, as in Fig. 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. Details of the dictionary
250 construction for different imaging scenarios will be explained in Sec. 4.

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
255 from zero, where $\mathbf{X}$ is a binary vector to indicate which codeword to select from the dictionary and N is the number of codewords:

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_0, \text{ where } \mathbf{X} \in \{0, 1\}^N. \quad (4)$$

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. 1.
260 It is in general difficult to optimize, so we take the L_1 -norm instead. According to the well-studied sparse signal recovery theory [1], 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
265 step, i.e., $Q(\mathbf{DX})$. A common way to handle quantization is to treat its effect as measurement noise [40]. We here propose to consider it in addition to the random noise that accounts for the degradation of background subtraction or inaccuracy of shape templates. We then express the whole generative process as $\mathbf{Y} \sim \mathbf{DX} + e_r + e_q$, where e_r and e_q account for the two separate noise sources.

270 As long as the noise is sparse, the sparse signal recovery theory still applies.

Note that if there is no mutual occlusion between objects, i.e, every pixel belongs to a unique object, then Eq. 4 is an accurate model in the sense that no quantization effect needs to be considered.

By replacing the L_0 -norm with the L_1 -norm in Eq. 4, the original formulation
275 can be converted to the following linear programming problem:

$$\begin{aligned}
& \min_{\mathbf{X}, \mathbf{U}} \mathbf{1}^T \mathbf{U} \\
s. \ t. \quad & -\mathbf{D}\mathbf{X} - \mathbf{U} + \mathbf{Y} \leq 0, \\
& \mathbf{D}\mathbf{X} - \mathbf{U} - \mathbf{Y} \leq 0, \\
& \mathbf{0} \leq \mathbf{X} \leq \mathbf{1},
\end{aligned} \tag{5}$$

where $\mathbf{U}$ is an auxiliary variable. Note the above formulation with the L_1 -norm is a relaxed version of the original problem because $\mathbf{X}$ is continuous in Eq. 5. A branch-and-bound method can be applied to further yield the exact integer solution. The L_1 -norm introduces sparsity in the solution, which is a desirable
280 property as we want to use a minimal number of hypotheses to explain the image observation. We refer to the solution of Eq. 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
285 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 K versions of shape templates
290 shown in Fig. 3 used in our experiments:

$$\min_{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}} \left\| \mathbf{Y} - \sum_{k=1}^K \mathbf{D}_k \mathbf{X}^{(k)} \right\|_1, \ s. \ t. \ \sum_{k=1}^K \mathbf{X}^{(k)} \leq \mathbf{1}, \ \mathbf{X}^{(k)} \in \{0, 1\}^N. \tag{6}$$

The 1-bit quantization described in the above generative process is very crude. In our detection context, when severe occlusion between objects exists,

Figure 4: Two binary signals d_1 and d_2 (codewords) are overlaid 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 Eq. 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.

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.

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 $\hat{\mathbf{Y}}$, as it can explain how many objects are involved in the occlusion at that pixel (Fig. 5). By definition, this occlusion layer variable only takes non-negative integer values. We extend Eq. 5 to a new linear programming problem that simultaneously estimates occlusion layers for 1-bit dequantization and sparse signal recovery as follows:

$$\begin{aligned}
& \min_{\mathbf{X}, \hat{\mathbf{Y}}} \quad [\|\hat{\mathbf{Y}} - \mathbf{D}\mathbf{X}\|_1 + \beta\|\hat{\mathbf{Y}} - \mathbf{Y}\|_1] \\
& s. t. \quad \hat{y}_i \geq y_i, \quad \forall i : y_i > 0 \\
& \quad \hat{y}_j = 0, \quad \forall j : y_j = 0 \\
& \quad \text{and } \hat{\mathbf{Y}} \in \mathbb{Z}_+^M, \mathbf{X} \in \{0, 1\}^N,
\end{aligned} \tag{7}$$

Figure 5: Illustration of the proposed LDQD method. The first image shows a binary image $\mathbf{Y}$ after background subtraction. The second image shows the estimated occlusion layers $\hat{\mathbf{Y}}$ found by LDQD. A brighter pixel value suggests a higher value for the associated occlusion variable, i.e., more objects are occluded at that pixel location. The third image gives the final detection results on a sample image from the PETS2009 S1L1-1357 dataset.

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 $\mathbb{Z}_+$ is the set of non-negative integers. By linear relaxation, the above problem can be converted to a linear programming problem similar to Eq. 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 Eq. 7 as the “Linear Dequantization Decoder (LDQD).” By default, in this paper we use the augmented formulation Eq. 7 as our sparsity-driven detector (SDD).

3.3. 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. 6(a) represent all possible locations at each time frame stacked in columns. Each edge 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 [2]. 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. However, one can link the sink node with the source node to form an equivalent *transportation* problem without explicitly specifying the amount of flow to be pushed. 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 [41, 42]. The minimization problem is given as follows:

$$\min_{\mathbf{f} \geq 0} \sum_i \sum_j c_{i,j} f_{i,j}, \text{ s. t. } \sum_i f_{i,n} = \sum_j f_{n,j}, \forall n. \quad (8)$$

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

Figure 6: 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 duplicate the network into multiple layers, while each layer corresponds to one specific category of object. A flow cannot travel across different layers since any track is uniquely associated with one object category.

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 Sec. 3.4.

Other variants of the network flow model exist in the tracking literature [25,
33, 34], all of which were designed for single-category object tracking. Here
we extend the classic network flow model to a multi-layer network model in
order to handle the multi-object multi-category tracking problem. The multi-
object multi-category tracking problem is the most general tracking problem
but seldom discussed in the literature. In many applications, there are multiple

categories of objects of interest (pedestrians, bicycles, cars, etc.). In addition
365 to handling the traditional constraints of the multi-object single-category prob-
lem, here we also need to ensure a track is uniquely associated with one object
category. We create multiple layers in the network with duplicated structures,
where each layer describes the multi-object tracking problem for a specific cate-
370 gory. We do not allow any transition edge to cross the layers so a flow will never
travel across different categories, as shown in Fig. 6(b). However, the mutual
exclusion constraint across different categories also needs to be satisfied, i.e.,
tracks with different object categories cannot occupy the same location at the
same time. We express such constraints as a set of linear inequality constraints:
 $\sum_l f_{l,n^+,n^-}^{(t)} \leq 1$, where there are L layers to model L categories of objects, as
375 shown in Fig. 6(b); $f_{l,n^+,n^-}^{(t)}$ represents the flow going through the n -th node in
the l -th layer at frame t (we give detailed definitions of different flow variables
in the next section). These constraints are typically called “side constraints.”
Note that side constraints do not exist in the standard minimum-cost flow for-
mulation (Eq. 8), and therefore standard network-flow solvers cannot be used.
380 Fortunately, the modified optimization problem here can still be relaxed to a
linear programming problem where efficient solvers exist.

3.4. The Coupling Algorithm

To couple our sparsity-driven detector and the classic network-flow data
association method, we propose a joint objective function, where $\sum_t \|\hat{\mathbf{Y}}_t -$
385 $\mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1$ approximates the negative logarithm of the image likeli-
hood $p(\mathbf{Y}|\mathbf{X})$ 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. 1. We separate the
set of flow variables $\mathbf{f}$ into four subsets (we use one layer network to simplify the
explanation): $f_{in,n^+}^{(t)}$ is associated with the edge that connects the source node
390 to the n -th node at frame t ; $f_{n^-,out}^{(t)}$ is associated with the edge that connects
the n -th node to the sink node at frame t ; $f_{n^+,n^-}^{(t)}$ is associated with the edge
that connects the duplicated n -th nodes at frame t ; $f_{m^-,n^+}^{(t)}$ 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 Eq. 8 and
 395 using Eq. 7, we have the following coupled minimization problem:

$$\begin{aligned}
 \min_{\mathbf{X}_t, \mathbf{f}, \hat{\mathbf{Y}}_t, t=1, \dots, T} \quad & [\sum_t \|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1 + \sum_{t,n} c_{in, n^+}^{(t)} f_{in, n^+}^{(t)} + \sum_{t,n} c_{n^-, out}^{(t)} f_{n^-, out}^{(t)} \\
 & + \sum_{t,n} c_{n^+, n^-}^{(t)} f_{n^+, n^-}^{(t)} + \sum_{t,m,n} c_{m^-, n^+}^{(t)} f_{m^-, n^+}^{(t)}] \\
 \text{s. t.} \quad & f_{in, n^+}^{(t)} + \sum_m f_{m^-, n^+}^{(t)} = \sum_k f_{n^-, k^+}^{(t)} + f_{n^-, out}^{(t)} \\
 & \sum_{t,n} f_{in, n^+}^{(t)} = \sum_{t,n} f_{n^-, out}^{(t)} \\
 & x_{t,n} = f_{n^+, n^-}^{(t)}, \quad \forall t, \forall n \\
 & \hat{y}_{t,i} \geq y_{t,i}, \quad \forall t, \forall i : y_{t,i} > 0 \\
 & \hat{y}_{t,j} = 0, \quad \forall t, \forall j : y_{t,j} = 0 \\
 & \mathbf{f} \geq 0, \hat{\mathbf{Y}}_t \in \mathbb{Z}_+^M \text{ and } \mathbf{X}_t \in \{0, 1\}^N.
 \end{aligned} \tag{9}$$

The selection variable $\mathbf{X}_t$ indicates the presence of an object at a particular location in discretized space at time t . For K templates, as indicated in Eq. 6, $\mathbf{X}_t$ here can be replaced by $\sum_{k=1}^K \mathbf{X}_t^{(k)}$ with linear constraints: $\sum_{k=1}^K \mathbf{X}_t^{(k)} \leq \mathbf{1}$. The flow variable $\mathbf{f}$ is used in the minimum-cost flow problem, where $f_{i,j} = 1$
 400 means there is a match between detections at location i and j , which belong to the same track. The cost function (9) is the summation of two *local* terms to minimize; the first term represents the costs of sparsity-driven object detection (Sec. 3.2) and the second term measures the costs of temporal data association in the minimum-cost flow formulation (Sec. 3.3). The first set of constraints
 405 with flow variables ensures a balance of flow. The second set of constraints with flow and selection variables ensures *consistency* between the two local variables $\mathbf{X}$ and $\mathbf{f}$. 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., $f_{n^+, n^-}^{(t)} = 1$.

410 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 gener-

alization 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 (10) to the minimization problem (9):

$$\begin{aligned}
L(\boldsymbol{\lambda}) = & \min_{\mathbf{X}, \mathbf{f}, \hat{\mathbf{Y}}} \left[\sum_t \|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1 + \boldsymbol{\lambda}_t^T \mathbf{X}_t \right. \\
& + \sum_{t,n} c_{in,n^+}^{(t)} f_{in,n^+}^{(t)} + \sum_{t,n} c_{n^-,out}^{(t)} f_{n^-,out}^{(t)} + \sum_{t,n} (c_{n^+,n^-}^{(t)} - \lambda_{t,n}) f_{n^+,n^-}^{(t)} \\
& \left. + \sum_{t,m,n} c_{m^-,n^+}^{(t)} f_{m^-,n^+}^{(t)} \right] \quad (10)
\end{aligned}$$

Here λ_t is a vector of the Lagrangian multipliers associated with the coupling constraints $x_{t,n} = f_{n^+,n^-}^{(t)}$ defined in Eq. 9. According to the subgradient method for dual decomposition [43], the so-called master problem can be separated into $(T+1)$ independent subproblems as follows (constraints are omitted), where T is the number of frames:

$$\begin{aligned}
g_t(\boldsymbol{\lambda}) &= \min_{\mathbf{X}_t, \hat{\mathbf{Y}}_t} [\|\hat{\mathbf{Y}}_t - \mathbf{D}\mathbf{X}_t\|_1 + \beta \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_1 + \boldsymbol{\lambda}_t^T \mathbf{X}_t] \\
h(\boldsymbol{\lambda}) &= \min_{\mathbf{f}} \left[\sum_{t,n} c_{in,n^+}^{(t)} f_{in,n^+}^{(t)} + \sum_{t,n} c_{n^-,out}^{(t)} f_{n^-,out}^{(t)} + \sum_{t,n} (c_{n^+,n^-}^{(t)} - \lambda_{t,n}) f_{n^+,n^-}^{(t)} \right. \\
& \quad \left. + \sum_{t,m,n} c_{m^-,n^+}^{(t)} f_{m^-,n^+}^{(t)} \right]
\end{aligned}$$

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 chose 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 as well as convergence proof were described by Bertsekas [43]. In summary, the dual decomposition technique then yields the following Coupling Algorithm:

COUPLING ALGORITHM FOR TRACKING

For $k = 1, 2, \dots, \text{maxIteration}$, **do**

- Solve T sparsity-driven detection problems with the primal-simplex algorithm:
 $\mathbf{X}_t \leftarrow \arg \min g_t(\boldsymbol{\lambda})$.
- Solve the minimum-cost flow data-association problem with the network-simplex algorithm:
 $\mathbf{f} \leftarrow \arg \min h(\boldsymbol{\lambda})$.
- **If** $x_{t,n} = f_{n^+,n^-}^{(t)}$ for all n, t , **Then Return** $\mathbf{X}_t, \mathbf{f}$
- Update dual variables $\lambda_{t,n} = \lambda_{t,n} + \alpha_k(x_{t,n} - f_{n^+,n^-}^{(t)})$, $\alpha_k = \frac{1}{k}$ (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 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

¹Cplex: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>

output. By changing the value of λ dynamically through dual decomposition, false alarms can be suppressed and detections missed due to occlusions can be
450 recovered.

4. Experiments and Discussion

We first present and discuss results of our sparsity-driven detectors LDND and LDQD on the sequences from the PETS2009 dataset [44]. Additional extensive experiments with these detectors on a synthetic dataset have shown
455 that LDQD has superior performance over LDND if the amount of occlusion is large [45]. We here then report results of our coupling algorithm applied to various infrared video sequences as well as visible-light sequences and give a quantitative analysis and comparison to current state-of-the-art methods.

4.1. Evaluation of the Sparsity-driven Detector

460 **Evaluation Datasets and Metrics.** We tested our sparsity-driven detector (LDND and LDQD) on the PETS2009 dataset for people localization. To compare with other reported results in the literature, we selected three subsets that represent different crowd densities (S1L1-1357, S1L1-1359, S1L2-1406) from PETS2009. We chose only the first view of each sequence, as previous
465 works in the literature did when testing on these sequences. The performance was measured by precision and recall at different levels of a miss/hit threshold, as well as Average Frame Error for counting results. Details of definitions of these metrics have been discussed by Ellis et al. [46].

Implementation Details. To obtain the binary image evidence, we applied an
470 adaptive Gaussian mixture estimation method for background subtraction [47]. The ground plane was discretized with a grid size of $0.3 \text{ m} \times 0.3 \text{ m}$, which is approximately half of the space a pedestrian could occupy. To speed up the computation, we rescaled the binary image to a 320×240 pixel resolution. Two shape templates were used, as described in Fig. 3, which were obtained
475 through K-means clustering explained in Sec. 4.2. We further used 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 was approximately 500 codewords, each approximately representing 20,000 pixels. The regularization parameter β was chosen to be 0.1.

Figure 7: (a) Evaluation results for Precision vs. Hit/Miss threshold on PET-S2009 S1L1-13-57 (b) Evaluation results for Recall vs Hit/Miss threshold on PETS2009 S1L1-13-57. In the official rules of the PETS evaluation, the hit/miss threshold is set to 0.5.

Comparison with state-of-the-art methods. We compared our detector with a sampling-based method for which results had been reported on the same data. The MCMC sampler [48] 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 does it consider the quantization effect explicitly. Moreover, the sampling process converges very slowly (30 s per frame in their Matlab implementation) and is vulnerable to be trapped in local minima.

The evaluation results at different levels of hit/miss thresholds are shown in Fig. 7. The superior performance of the MCMC sampler against several gradient-feature-based methods was previously reported by Ge [48]. As expected, our sparsity-driven detector (SDD) outperformed the MCMC sampler in both accuracy and efficiency. The Cplex solver typically found the optimal integer solution in less than one second. The LDND detector without considering the quantization effect achieved a similar precision but a much lower recall rate compared to the LDQD detector because of severe partial occlusion. As shown in Fig. 5, our LDQD detector did not only localize the pedestrians but it produced an estimate of occlusion layers as well.

We also compared our method to a similar approach by Alahi et al. [13], who measured the reconstruction error with the L_2 norm. To be able to compare to their reported results, we followed 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 Table 1.

Figure 8: Regions of interest in PETS2009 dataset. People passing through R0, R1 and R2 are counted.

Data	Method	R0	R1	R2
		AFE	AFE	AFE
S1L1	Alahi et al [13]	4.2	2.3	1.8
-1359	Our LDQD	1.5	0.6	1.0
S1L2	Alahi et al [13]	n/a	6.5	4.0
-1406	Our LDQD	6.0	3.6	1.9

Table 1: Counting results on PETS2009 dataset. Performance is measured by the Average Frame Error (AFE), whose ideal value is 0.

4.2. Evaluation for the Coupling Algorithm

Evaluation Datasets and Metrics. We used the standard CLEAR MOT metrics [49] to evaluate the tracking performance of our CP algorithm. The Multiple Object Tracking Accuracy (MOTA) metric combines the false positive rate, miss rate, and identity-switch rate into a single number with ideal value 100%; the Multiple Object Tracking Precision (MOTP) measures the average distance between ground truth and tracker output. For evaluating 2D tracking performance, precision is defined according to the region overlap criterion with ideal value 1. For 3D, precision is defined according to the Euclidean distance in 3D space based on camera calibration information with ideal value 0. To better assess the quality, we additionally report the numbers of Mostly Tracked (MT, $\geq 80\%$) trajectories, Mostly Lost (ML, $\leq 20\%$) trajectories, track fragmentations (FM, the number of times that a ground truth trajectory is interrupted), and identity switches (IDS, the number of times that a tracked trajectory changes its matched ground truth).

Our coupling algorithm was evaluated on 11 video sequences from challenging benchmarks that vary in object category, density, motion pattern, imaging modality and camera views (see Table 2): 1) Two sequences of flying bats with 3 views from BU-TIV [50]. We chose 0.3 m in 3D space as the miss/hit threshold which approximates the physical size of the object. 2) The infrared multi-

Table 2: Dataset Summary.

Data	Camera	Frames	Category	Objects	Density	Views	Tracking Space	Hit/Miss
BU Bat-1	Infrared	1100	Bat	207	Sparse	3	3D Space	$\leq 0.3\ m$
BU Bat-2	Infrared	200	Bat	203	Dense	3	3D Space	$\leq 0.3\ m$
BU Marathon-1	Infrared	3000	Runner, Car Motor, Bicycle	179	Median	1	Image Plane	≥ 0.5
PETS S2L1	Visible	795	Pedestrian	23	Sparse	1	Ground Plane	$\leq 1\ m$
PETS S1L1-1357	Visible	221	Pedestrian	46	Dense	1	Ground Plane	$\leq 1\ m$
PETS S1L1-1359	Visible	241	Pedestrian	36	Median	1	Ground Plane	$\leq 1\ m$
PETS S1L2-1406	Visible	201	Pedestrian	43	Dense	1	Ground Plane	$\leq 1\ m$
PETS S2L2	Visible	436	Pedestrian	75	Dense	1	Ground Plane	$\leq 1\ m$
EPFL Terrace-1	Visible	5000	Pedestrian	9	Median	4	Ground Plane	$\leq 1\ m$
UCF PNNL ParkingLot-1	Visible	1000	Pedestrian	14	Median	1	Image Plane	≥ 0.5
Oxford Towncenter	Visible	4500	Pedestrian	230	Sparse	1	Image Plane	≥ 0.5

category multi-object tracking sequence BU-Marathon from BU-TIV [50] that captures a street view of pedestrians, bicycles, cars and motors. The ground truth was given on the image plane, so we chose 0.5 for the region overlap threshold. To address the multi-category issue, in addition to the miss/hit threshold, the system output was not paired with the ground truth if the category labels were not the same. 3) The first view of five sequences S2L1, S2L2, S1L1-1357, S1L1-1359, S1L2-1406 from PETS2009. We used the ground truth annotation provided by Andriyenko et al. [29]. To enable comparisons with previously published results, we restricted our evaluation to objects moving in the constrained area defined by Andriyenko et al. [29], and chose 1 m on the ground plane as the miss/hit threshold for computing the metrics. 4) Four views from the EPFL Terrace-1 dataset. The ground truth was originally given on the ground plane, so we chose 1 m on the ground plane as the miss/hit threshold. 5) PNNL-Parking-Lot-1 from UCF dataset [8]² and TownCenter from Oxford dataset [51]. We chose 0.5 for the region overlap threshold to determine the

²We modified the ground truth that had been originally published with the UCF-PNNL dataset because many partially occluded pedestrians were not labeled. The results presented here are based on the process of the first 747 frames in order to align with previously published results [26].

miss/hit on image plane.

Implementation Details. We here describe how our method learns the shape templates and dictionaries (Fig. 3) and how to set up the network (Fig. 6). To develop shape templates, we assume a pedestrian can occupy a cylinder
555 with radius 30 cm and height 180 cm, and that a flying animal can occupy a sphere volume of 15-cm radius. The shape variation is learned through K-means clustering on a training set that comprises of 200 unoccluded examples, which results in two shape templates (frontal and side view) for pedestrians and five templates for bats. The learned template from a typical K-means clustering is
560 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 dictionaries may potentially improve the performance, we did not find it necessary given the relatively small resolution of the objects in
565 most test sequences. Similar steps were applied to the BU-Marathon sequence, where binary templates of pedestrians, cars, bicycles, and motors were learned. Once the shape templates have been obtained, the size and location of the projected object in the binary image (codeword) can be determined with the camera calibration information [13]. If such 3D-to-2D information is not available (like
570 BU-Marathon and UCF-PNNL), we can estimate the size of object at each grid point on the image plane with a few training samples and a linear regressor.

To set up the network used in data association, we needed to define the cost on the edges. As shown in Fig. 6, there are four types of edges: edges between the duplicated nodes within a time frame, edges between nodes across time,
575 edges starting from the source node and edges terminating at the sink node. We call the cost defined on these edges “detection cost,” “transition cost,” “initiation cost” and “termination cost.” The detection cost was 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
580 codeword, which can be seen as a measure to support the presence of an object at a particular position. For the infrared dataset, we used the Euclidean distance

between nodes as the transition cost. For datasets captured by visible-light cameras, we first computed the color histogram of each codeword. Given two instantiated codewords from consecutive frames, we computed the histogram intersection distance to represent the transition cost. We use the histogram comparison method here in replacement of the normalized correlation method used in our previous work [16] mainly because we can benefit from the integral image method to speed up the computation. 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 Eq. 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. The initiation and termination costs represent our prior knowledge of the moving objects and the scene. They altogether determine the minimum detection evidence needed to validate the existence of a moving object. We assign a small constant to the edges where objects are most likely to appear, such as image border or entrance/exit of the scene. Additional implementation details have been described by Wu [45].

Important parameter settings. There are a few of user-defined parameters that needed to be determined by experiment. The weighting parameter β in Eq. 7 that governs the dequantization quality was set to 0.1 for all the sequences with rigid objects and 10 for the infrared dataset of bats, according to the detection performance on a few of training images. Generally, the need to estimate the dequantization effect for datasets with multi-view support is not as strong as for the single-view datasets. To ensure the numerical balance between the L_1 -norm term and the network-flow term in Eq. 9, we rescaled the L_1 -norm by another weighting parameter γ , which is set to 0.01 for all datasets. These weighting parameters are not sensitive in general, and we did not extensively tune them throughout our experiments. Once reasonable values for these

weighting parameters are found, a wide range of values nearby may apply as well.

615 **Comparison with baselines and state-of-the-art methods.** We designed a baseline method (**SDD-Net**) that applies our SDD detector and solves the data association sequentially, which can be considered to run our Coupling Algorithm in one iteration only for planar motion cases. We also report pedestrian tracking results from our preliminary work, **CP(LDND)**, that couples our LDND and
 620 network-flow. Our new method that couples LDQD and network-flow is denoted as **CP(LDQD)**. Both methods use exactly the same parameter settings in the same coupling framework, except that the choices of detector are different. For tracking bats in 3D space, we extended the reconstruction-tracking method (RT [52]) by applying our SDD detector on a dense set of triangulation hypotheses of 3D points (**SDD-RT**), and used the multi-dimensional assignment
 625 formulation to solve the data association step. We report tracking results after 20 iterations of the Coupling Algorithm in Table 3 and Table 4 to demonstrate the improved performance.

To better assess the challenges of the datasets and the overall tracking performance of our method, we also compared it with several state-of-the-art methods,
 630 see Table 3 and Table 4. A variant of the flow-based method, KSP [53], has a problem setup similar to ours – it also requires discretization of the ground plane. The occlusion-modeling (OM) method [29] combines explicit occlusion reasoning, a full-body SVM classifier, tracklet stitching, and initialization via
 635 extended Kalman filtering. The GMCP method [26] uses a graph representation to assist a complicated tracklet stitching process. The Occlusion Geodesics (OG) online method [54] exploits the spatio-temporal evolution of occlusion regions, detector reliability, and target motion prediction to handle missed detections. The Constrained Sequential Labeling (CSL) method [30] uses supervoxels or
 640 dense point trajectories as a coherent spatiotemporal basis for handling occlusion. The Simultaneous Detection-tracklet Classification and Clustering (CL^2) method [31] associates objects in the joint space of tracklets and motion trajectories. Both CSL and CL^2 methods require long-term motion flow estimation

Table 3: Quantitative results of the coupling algorithm. Most existing trackers sequentially apply the detection and data association modules, while our CP method couples them. MOTA is ideally 1, MOTP 1 or 0 cm.

Data	Method	MT	ML	FM	IDS	MOTA	MOTP
BU	RT [52]	96.6%	0	105	97	64.1%	8.9 cm
Bat-1	our SDD-RT (baseline)	95.2%	0	145	126	78.9%	5.7 cm
	our CP(LDQD)	97.1%	0	115	126	88.1%	5.4 cm
BU	RT [52]	71.9%	2.5%	274	355	-32.0%	10.0 cm
Bat-2	our SDD-RT (baseline)	61.1%	3.0%	454	444	44.9%	7.7 cm
	our CP(LDQD)	84.2%	0.5%	176	156	80.5%	6.2 cm
BU	our SDD-Net (baseline)	60.9%	12.3%	172	158	62.1%	76.1%
Marathon	our CP(LDQD)	61.5%	13.4%	153	142	65.1%	76.0%
EPFL	our SDD-Net (baseline)	100%	0	13	53	69.5%	23.3 cm
Terrace-1	our CP(LDND)	55.6%	0	55	61	70.4%	25.6 cm
	our CP(LDQD)	100%	0	18	31	96.1%	22.3 cm
UCF	our SDD-Net (baseline)	78.6%	0	31	23	80.5%	70.5%
PNNL	our CP(LDND)	57.1%	7.1%	37	28	74.4%	71.3%
	our CP(LDQD)	92.9%	0	22	22	86.6%	70.9%
	GMCP [26]	85.7%	7.1%	9	3	86.5%	84.4%
	Joint-Learn-2 [11]	78.6%	7.1%	11	16	81.1%	80.6%
Oxford	our SDD-Net (baseline)	53.0%	8.3%	1226	1382	17.5%	67.7%
Towncenter	our CP(LDND)	33.9%	10.4%	1406	618	45.5%	68.0%
	our CP(LDQD)	45.2%	9.6%	1312	890	42.5%	68.0%

so they can provide more detailed object segmentation on top of the bound-
ing box localization. The joint people detection (*Joint – Learn*) method [11]
learns pairwise mutual occlusion and trains the people detector with a tracker
in the loop. Results of these competing methods were taken from publications
or directly from the authors. Note that many existing works rely on a strong
pedestrian detector, and the chosen detector (typically part-based) is only appli-
cable to relatively high-resolution images recorded by visible-light cameras. To
position our contribution, we here emphasize that although we were interested
in designing a pedestrian detector that would work for various imaging modal-
ities and for images with limited resolution, our main focus was a demonstration
of our coupling framework and its general applicability to different datasets.

Throughout our experiments with various, notably distinct datasets, our
coupling algorithm proved more reliable than competing methods based on the

MOTA, MT, and ML scores and comparably accurate based on the MOTP scores (Table 3)³. The superior performance is best demonstrated in sequences containing dense crowds of objects with frequent partial occlusions. With the
660 same detection method (SDD), our coupling algorithm was able to achieve high-quality results and outperform the baseline methods (SDD-Net, SDD-RT) consistently in MOTA. For the two variants of SDD detection methods, the strength of using the occlusion-sensitive detector (LDQD) over our previous work (LD-ND) is also demonstrated by the results on the pedestrian tracking sequences.

665 The results on the infrared data of bats highlight that our SDD detection method can successfully suppress ghost reconstructions (false triangulation) in 3D space, a phenomenon presented in our previous study for tracking in 3D space [52]. Although the reconstruction-tracking (RT) algorithm could successfully track most of the objects, it had a high false positive rate because of the
670 persistent ghost effect during the reconstruction step. This issue is not addressed sufficiently in existing literature, probably because only sparse tracking scenarios have 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 in RT [52] with the proposed sparsity-driven
675 detector, we were able to improve the overall performance MOTA score by almost 77%. Moreover, the performance improvement between SDD-RT and CP shows the important impact of our coupling idea.

Optimization Issues The variables in the Coupling Algorithm can be optimized separately. In particular, each detection subproblem can be solved inde-
680 pendently through parallel computing. The complexity of the L_1 minimization (Fig. 3) depends on the size of the dictionary, which is determined by the grid size of the ground plane or the number of valid triangulation candidates, the number of shape templates and the image size. After the preprocessing step
685 that reduced the dictionary size as described earlier, once the problem was con-

³Video results: <https://www.youtube.com/watch?v=IVE3zYxH1MY&feature=youtu.be>

Figure 9: Performance of the coupling algorithm on PETS-S1L1-1359 at each iteration with different initialization. The MOTA score changes quickly at the first few iterations.

structured, the actual time to solve the linear programming problem was less than
 a second per frame in our implementation. The data association subproblem
 could also be solved efficiently even for a large network with one million nodes in
 our experiment. This is because our networks were sparse and the complexity of
 the minimum-cost flow algorithm was 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 was adopted and the primal optimal basis was saved to initialize the
 optimization in the next iteration. Furthermore, we found simple rounding of
 the LP solution was sufficiently accurate, so additional efforts to pursue the
 exact integer solution were not needed.

We found that the Coupling Algorithm does not need to run many iterations
 before it reaches a good solution. We monitored the tracking quality at each
 iteration of the subgradient method used by the Coupling Algorithm with two
 different initializations. If we first apply 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 call this a “good initialization.” If we do not

set the detection cost, no flow will be pushed at the first iteration, and we
 705 call this a “bad initialization.” As shown in Fig. 9, it is always beneficial to
 use a “good initialization” if we are confident in the majority of our detection
 results. Despite the difference in 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
 710 has also been observed in other optimization work [55]. Although we set the
 maximum iterations to be 20 in our experiments, in practice, an earlier stop is
 sufficient for producing a good suboptimal solution.

Finally, we report the running time of our algorithm for the PETS2009
 dataset (Table 5) to illustrate the impact of increasing object density. The re-
 715 ported running time is separated into three steps: preprocessing time, problem-
 building time and problem-solving time. The preprocessing step performs fore-
 ground estimation and sets up the sparsity-driven detector. The problem-
 building step sets up the coupled minimization problem. The problem-solving
 step performs the iterative dual decomposition algorithm. The execution time
 720 of each step of our Matlab implementation was measured on an Intel(R) X-
 eon(R) 2.40 GHz processor. We found that the main bottleneck is setting up
 the minimization problem, while the actual problem solving step is very efficient
 when we use a highly optimized linear programming solver (Cplex).

5. Summary and Discussion

725 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 (Eq. 1), taking advantage of the often complemen-
 tary nature of the two subproblems. As a concrete example, our Coupling
 Algorithm combines a sparsity-driven detection method and a network-flow
 730 data-association method within this framework (Eq. 10). 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 (Eq. 11), a coupled objective function is optimized iteratively with off-the-shelf efficient algorithms for each sub-problem. 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 is 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. A significant portion of this time was spent on the construction of the optimization problem. Additional efforts should be made to speed up the implementation in scenarios where objects may have large variations of poses or where 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 or the foreground estimation is difficult. This is the main reason that causes the misalignment of the produced bounding box. Combining gradient features with binary shape templates has been proven to be effective in the object detection literature [56]. Output from these detectors could be used to introduce a bias on which codeword to select. Finally, in future work, we would like to consider incorporating more prior knowledge into our framework, such as the social group model described by Qin and Shelton [57].

Acknowledgment

This material is based upon work partially supported by the National Science

Foundation (IIS-0910908, IIS-0855065, 1421943, 1337866) and the Office of Naval Research (024-205-1927-5). We would like to thank Nathan Fuller, Diane Theriault and Prof. Thomas H. Kunz for help with data collection, and Prof. Stan Sclaroff, Prof. David Castañón and Ashwin Thangali for feedback on this work.

References

- [1] E. J. Candes, T. Tao, Decoding by linear programming, *IEEE Trans. Information Theory* 51 (12) (2005) 4203–4215.
- [2] D. A. Castañón, Efficient algorithms for finding the k best paths through a trellis, *IEEE Trans. AES* 26 (2) (1990) 405–410.
- [3] B. Leibe, K. Schindler, N. Cornelis, L. V. Gool, Coupled object detection and tracking from static cameras and moving vehicles, *IEEE Trans. PAMI* 30 (10) (2008) 1683–1698.
- [4] V. Abalavsky, S. Sclaroff, Layered graphical models for tracking partially-occluded objects, *IEEE Trans. PAMI* 33 (9) (2011) 1758–1775.
- [5] B. Wu, Part based object detection, segmentation, and tracking by boosting simple feature based weak classifiers, Ph.D. thesis, University of South California, USA (2008).
- [6] D. M. Ross Girshick, Pedro Felzenszwalb, Object detection with grammar models, in: *NIPS*, 2011, pp. 442–450.
- [7] X. Wang, T. X. Han, S. Yan, An hog-lbp human detector with partial occlusion handling, in: *ICCV*, 2009, pp. 32–39.
- [8] G. Shu, A. Dehghan, O. Oreifej, E. Hand, M. Shah, Part-based multiple-person tracking with partial occlusion handling, in: *CVPR*, 2012, pp. 1815 – 1821.
- [9] M. Yang, F. Lv, W. Xu, Y. Gong, Detection driven adaptive multi-cue integration for multiple human tracking, in: *ICCV*, 2009, pp. 1554 – 1561.

- [10] B. Yang, R. Nevatia, Online learned discriminative part-based appearance models for multi-human tracking, in: ECCV, 2012, pp. 484–498.
- 790 [11] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, B. Schiele, Learning people detectors for tracking in crowded scenes, in: ICCV, 2013, pp. 1049 – 1056.
- [12] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Trans. PAMI 99 (4) (2011) 743 – 761.
- 795 [13] A. Alahi, L. Jacques, Y. Boursier, P. Vandergheynst, Sparsity-driven people localization algorithm: Evaluation in crowded scenes environments, in: IEEE Wksp. PETS, 2009, pp. 1–8.
- [14] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multicamera people tracking with a probabilistic occupancy map, IEEE Trans. PAMI 30 (2) (2008) 267–282.
- 800 [15] W. Ge, R. T. Collins, Crowd detection with a multiview sampler, in: ECCV, 2010, pp. 324–337.
- [16] Z. Wu, A. Thangali, S. Sclaroff, M. Betke, Coupling detection and data association for multiple object tracking, in: CVPR, 2012, pp. 1948 – 1955.
- 805 [17] C. Desai, D. Ramanan, C. Fowlkes., Discriminative models for multi-class object layout, in: CVPR, 2009, pp. 229 – 236.
- [18] P. K. Olga Barinova, Victor Lempitsky, On detection of multiple object instances using hough transforms, IEEE Trans. PAMI 34 (9) (2012) 1773 – 1784.
- 810 [19] Z. Wu, M. Betke, T. H. Kunz., Efficient track linking methods for track graphs using network-flow and set-cover techniques, in: CVPR, 2011, pp. 1185 – 1192.

- [20] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows, in: CVPR, 2008, pp. 1–8.
- [21] Z. Khan, T. Balch, F. Dellaert, MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements, IEEE Trans. PAMI 28 (12) (2006) 1960 – 1972.
- [22] Q. Yu, G. Medioni, I. Cohen, Multiple target tracking using spatio-temporal Markov Chain Monte Carlo data association, in: CVPR, 2007, pp. 1–8.
- [23] C. Huang, B. Wu, R. Nevatia, Robust object tracking by hierarchical association of detection responses, in: ECCV, 2008, pp. 788–801.
- [24] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, W. Hu, Multi-object tracking through simultaneous long occlusions and split-merge conditions, in: CVPR, 2006, pp. 666–673.
- [25] G. Castanon, L. Finn, Multi-target tracklet stitching through network flows, in: Proc. IEEE Aerospace Conference, 2011, pp. 1–7.
- [26] A. R. Zamir, A. Dehghan, M. Shah, Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs, in: ECCV, 2012, pp. 343–356.
- [27] B. Yang, R. Nevatia, An online learned CRF model for multi-target tracking, in: CVPR, 2012, pp. 2034 – 2041.
- [28] C. Dicle, O. Camps, M. Sznajda, The way they move: Tracking multiple targets with similar appearance, in: ICCV, 2013, pp. 2304 – 2311.
- [29] A. Andriyenko, S. Roth, K. Schindler, An analytical formulation of global occlusion reasoning for multi-target tracking, in: Proc. 11th IEEE Int’l. Workshop on Visual Surveillance, 2011, pp. 1839 – 1846.
- [30] S. Chen, A. Fern, S. Todorovic, Multi-object tracking via constrained sequential labeling, in: CVPR, 2014, pp. 1130 – 1137.

- [31] K. Fragkiadaki, W. Zhang, G. Zhang, J. Shi, Two-granularity tracking: mediating trajectory and detection graphs for tracking under occlusions, in: ECCV, 2012, pp. 552 – 565.
- [32] M. Andriluka, S. Roth, B. Schiele, People-tracking-by-detection and people-detection-by-tracking, in: CVPR, 2008, pp. 1–8.
- [33] J. Henriques, R. Caseiro, J. Batista, Globally optimal solution to multi-object tracking with merged measurements, in: ICCV, 2011, pp. 2470 – 2477.
- [34] H. B. Shitrit, J. Berclaz, F. Fleuret, P. Fua, Multi-commodity network flow for tracking multiple people, IEEE Trans. PAMI 36 (8) (2006) 1614 – 1627.
- [35] A. Butt, R. Collins, Multi-target tracking by lagrangian relaxation to min-cost network flow, in: CVPR, 2013, pp. 1846–1853.
- [36] L. Leal-Taixe, G. Pons-Moll, B. Rosenhahn, Branch-and-price global optimization for multi-view multi-target tracking, in: CVPR, 2012, pp. 1987–1994.
- [37] S. Boyd, L. Xiao, A. Mutapcic, J. Mattingley, Notes on decomposition methods, http://http://see.stanford.edu/materials/lsoctee364b/08-decomposition_notes.pdf (2007).
- [38] N. Komodakis, N. Paragios, G. Tziritas, MRF optimization via dual decomposition: Message-passing revisited, in: ICCV, 2007, pp. 1–8.
- [39] L. Torresani, V. Kolmogorov, C. Rother, Feature correspondence via graph matching: Models and global optimization, in: ECCV, 2008, pp. 596–609.
- [40] P. T. Boufounos, R. G. Baraniuk, 1-bit compressive sensing, in: Proc. Conf. on Information Science and Systems (CISS), 2008, pp. 16 – 21.
- [41] A. Andriyenko, K. Schindler, Globally optimal multi-target tracking on a hexagonal lattice, in: ECCV, 2010, pp. 466–479.

- [42] H. Pirsiaavash, D. Ramanan, C. Fowlkes, Globally-optimal greedy algorithms for tracking a variable number of objects, in: CVPR, 2011, pp. 1201–1208.
- 870 [43] D. P. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, MA, 1999.
- [44] 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
875 a crowd, and detection of separate flows and specific crowd events, <http://www.cvg.rdg.ac.uk/PETS2009> (Jun. 2009).
- [45] Z. Wu, Occlusion reasoning for multiple object visual tracking, Ph.D. thesis, Boston University, USA (2012).
- [46] A. Ellis, A. Shahrokni, J. M. Ferryman, Pets2009 and winter-pets 2009
880 results: a combined evaluation, in: IEEE Wksp. PETS, 2009, pp. 1–8.
- [47] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, in: ICPR, 2004, pp. 28–31.
- [48] W. Ge, Bayesian analysis of small groups in crowds, Ph.D. thesis, The Pennsylvania State University, USA (2010).
- 885 [49] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: The CLEAR MOT metrics, EURASIP Journal on Image and Video Processing 2008 (1).
- [50] Z. Wu, N. Fuller, D. Theriault, M. Betke, A thermal infrared video benchmark for visual analysis, in: Proc. of 10th IEEE Workshop on Perception
890 Beyond the Visible Spectrum, 2014, pp. 1–8.
- [51] B. Benfold, I. Reid, Stable multi-target tracking in real-time surveillance video, in: CVPR, 2011, pp. 3457–3464.

- [52] Z. Wu, N. I. Hristov, T. H. Kunz, M. Betke, Tracking-reconstruction or reconstruction-tracking? comparison of two multiple hypothesis tracking approaches to interpret 3d object motion from several camera views, in: IEEE WMVC, Utah, 2009, pp. 1–8.
- [53] J. Berclaz, F. Fleuret, E. Turetken, P. Fua, Multiple object tracking using k-shortest paths optimization, IEEE Trans. PAMI 33 (9) (2011) 1806–1819.
- [54] H. Possegger, T. Mauthner, P. M. Roth, H. Bischof, Occlusion geodesics for online multi-object tracking, in: CVPR, 2014, pp. 1306 – 1313.
- [55] B. Savchynskyy, J., S. Schmidt, C. Schnörr, A study of Nesterov’s scheme for Lagrangian decomposition and map labeling, in: CVPR, 2011, pp. 1817 – 1823.
- [56] Q. Yuan, Learning a family of detectors, Ph.D. thesis, Boston University, USA (2010).
- [57] Z. Qin, C. R. Shelton, Improving multi-target tracking via social grouping, in: CVPR, 2012, pp. 1972–1978.

Table 4: Quantitative results of the coupling algorithm on PETS2009 Dataset.

Data	Method	MT	ML	FM	IDS	MOTA	MOTP
PETS	our SDD-Net (baseline)	87.0%	0	17	14	83.3%	67.5%
S2L1	our CP(LDND)	95.7%	0	12	5	90.5%	71.0%
	our CP(LDQD)	100%	0	15	9	91.0%	67.2%
	OM [29]	91.3%	0	12	19	88.6%	76.9%
	KSP [53]	73.9%	8.7%	22	13	80.3%	72.0%
	GMCP [26]	-	-	-	8	90.3%	69.0%
	OG [54]	100%	0	16	9	98.1%	80.5%
	CSL [30]	100%	0	53	18	91.5%	74.7%
	CL^2 [31]	34.8%	4.3%	53	56	70.5%	66.4%
PETS	our SDD-Net (baseline)	79.7%	2.7%	89	177	53.8%	65.1%
S2L2	our CP(LDND)	40.5%	6.8%	117	100	61.6%	65.2%
	our CP(LDQD)	78.4%	2.7%	104	155	65.9%	65.7%
	OM [29]	41.3%	10.7%	87	120	57.2%	59.7%
	KSP [53]	9.3%	53.3%	38	22	24.2%	60.9%
	OG [54]	41.5%	0	315	181	66.0%	64.8%
	CSL [30]	24.3%	9.5%	631	711	37.0%	65.9%
	CL^2 [31]	8.1%	41.9%	104	86	26.0%	65.5%
	Joint-Learn-2 [11]	-	-	-	-	56.9%	74.4%
PETS	our SDD-Net (baseline)	71.7%	4.3%	45	56	68.6%	60.9%
S1L1-1357	our CP(LDND)	32.6%	8.7%	39	31	61.7%	63.7%
	our CP(LDQD)	73.9%	2.2%	47	44	75.7%	65.2%
	CL^2 [31]	13.0%	28.3%	74	44	40.5%	65.9%
PETS	our SDD-Net (baseline)	86.1%	13.9%	11	24	78.0%	62.0%
S1L1-1359	our CP(LDND)	58.3%	19.4%	23	12	70.1%	58.0%
	our CP(LDQD)	88.9%	5.6%	21	14	88.9%	64.0%
	OM [29]	50.0%	22.2%	20	27	57.8%	61.9%
	KSP [53]	44.4%	38.9%	8	4	51.5%	64.8%
	CSL [30]	47.2%	11.1%	109	159	49.4%	62.3%
	CL^2 [31]	19.4%	22.2%	42	24	44.8%	67.5%
PETS	our SDD-Net (baseline)	65.1%	0	90	151	50.6%	46.9%
S1L2-1406	our CP(LDND)	14.0%	60.5%	19	13	22.5%	57.0%
	our CP(LDQD)	58.1%	0	86	138	56.0%	45.6%
	Joint-Learn-2 [11]	18.6%	44.2%	26	30	33.0%	47.9%

Table 5: Runtime results of the coupling algorithm on PETS2009 Dataset.

Data	People Density (per frame)	Prep. Time (per frame)	Building Time (per frame)	Solving Time (per frame per iteration)
PETS S2L1	5.0	0.56 s	0.60 s	0.03 s
PETS S2L2	19.2	1.20 s	1.79 s	0.45 s
PETS S1L1-1357	17.3	0.91 s	1.34 s	0.28 s
PETS S1L1-1359	10.7	0.79 s	1.00 s	0.15 s
PETS S1L2-1406	18.7	0.93 s	2.43 s	0.64 s

Figure 10: Tracking results for the Coupling Algorithm. Each of the six rows shows sample frames/trajectories from BU-Marathon, BU-Bat-2(3 views), PET-S S1L1-1357, S2L2, EPFL-Terrace-1(4 views), and UCF-PNNL-ParkingLot1 sequences, respectively. The same color/label is associated with the same object.