The Solution Path Algorithm for Identity-Aware Multi-Object Tracking

Shoou-I Yu[1], Deyu Meng[2], Wangmeng Zuo[3], Alexander Hauptmann[1]
[1] Carnegie Mellon University, [2] Xi'an Jiaotong University, [3] Harbin Institute of Technology
iyu@cs.cmu.edu, dymeng@mail.xjtu.edu.cn, wmzuo@hit.edu.cn, alex@cs.cmu.edu

Abstract

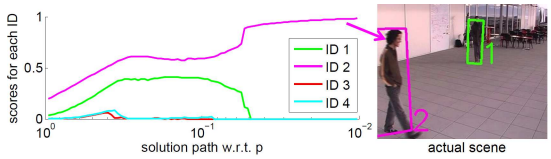
We propose an identity-aware multi-object tracker based on the solution path algorithm. Our tracker not only produces identity-coherent trajectories based on cues such as face recognition, but also has the ability to pinpoint potential tracking errors. The tracker is formulated as a quadratic optimization problem with ℓ_0 norm constraints, which we propose to solve with the solution path algorithm. The algorithm successively solves the same optimization problem but under different ℓ_p norm constraints, where p gradually decreases from 1 to 0. Inspired by the success of the solution path algorithm in various machine learning tasks, this strategy is expected to converge to a better local minimum than directly minimizing the hardly solvable ℓ_0 norm or the roughly approximated ℓ_1 norm constraints. Furthermore, the acquired solution path complies with the "decision making process" of the tracker, which provides more insight to locating potential tracking errors. Experiments show that not only is our proposed tracker effective, but also the solution path enables automatic pinpointing of potential tracking failures, which can be readily utilized in an active learning framework to improve identity-aware multi-object tracking.

1. Introduction

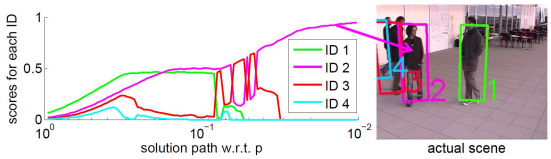
Multi-object tracking in a multi-camera environment serves as a fundamental building block to the analysis of surveillance video. Furthermore, if real-world identity information such as recognized faces can be assigned to each trajectory, this enables subsequent person-specific behavior analysis. Unfortunately, in long-term surveillance scenarios, it is inevitable for multi-object trackers to make errors, such as confusing the identity of two people. Therefore, if we could automatically pinpoint potential tracking errors, then humans can efficiently provide crucial information to guide the tracker to the correct tracking results.

Though many multi-object tracking algorithms have been proposed, most trackers 1) cannot perform identity-aware tracking and 2) cannot identify uncertainty in final tracking output. To this end, we propose an identity-aware multi-object tracking algorithm based on the solution path algorithm, which has been extensively used in multiple tasks such as Lasso [34] and Support Vector Machines [19]. Our tracker is formulated as a quadratic optimization problem with ℓ_0 norm constraints, which enforce the mutual exclusion and the spatial locality constraints, i.e. a person detection can only belong to a single physical individual and a physical individual cannot be at two locations at the same time. Unfortunately, directly solving the optimization problem with ℓ_0 norm constraints effectively is very difficult. An alternative is to solve its ℓ_1 norm convex relaxation, but the results will not be accurate. Therefore, we propose to utilize the solution path algorithm to "bridge" the two solutions under ℓ_1 and ℓ_0 norm constraints. Starting from the solution under the convex ℓ_1 norm constraints, the algorithm successively solves the same optimization problem but under different ℓ_p norm constraints, where p gradually decreases from 1 to 0. The solution path ends as p approaches to 0, and a better solution to our original problem can be obtained.

The solution path algorithm has two key benefits. First, it can optimize loss functions with ℓ_0 norm constraints, which is a common but difficult-to-optimize constraint in sparsity and relaxed combinatorial problems. Second, the solution



(a) Stable solution path: easy observation. It is clearly ID 2.



(b) Turbulent solution path: more confusing observation. Could be ID 2 or ID 3 as they are close together.

Figure 1: Visualization of the solution path of identity (ID) assignments w.r.t. p for two observations. One can identify uncertain observations from the shape of the solution path.

path can be viewed as the "decision making process" of the tracker, which can be utilized to automatically pinpoint uncertainty in tracking for manual correction in an active learning scenario [31]. We emphasize that this uncertainty is different from the confidence of detected people or tracklets used by many trackers [4, 41], which focuses on the confidence of the *input data*. In our case we are interested in estimating the confidence of the *output tracking results*. Figure 1 demonstrates how one could observe the solution path to determine the uncertainty of tracking output. In sum, the main contributions are as follows:

1. We propose a novel optimization paradigm based on the solution path algorithm to solve quadratic programs with ℓ_0 norm constraints.
2. We formulate our identity-aware tracker as a quadratic program with ℓ_0 norm constraints, which models the mutual exclusion and spatial locality constraints.
3. We demonstrate that the solution path acquired can be exploited to locate uncertain tracking output, which can be utilized in an active learning framework to enhance multi-object tracking.

2. Related Work

There has been much effort in improving tracking-by-detection-based [29] multi-object tracking in different aspects, including appearance modeling [21, 22, 4], motion modeling [13, 15] and data association. Data association is the task of grouping person detections into multiple disjoint sets where each set resembles a trajectory of a person. As our paper mainly focuses on the data association step, we discuss this aspect in more detail.

Many data association methods have been proposed. A popular way to formulate the tracking problem is designing a linear optimization function for tracking [20, 41, 30, 8, 15, 7, 24, 36] which can be solved efficiently with linear programming solvers or minimum cost network flow. However, linear objective functions cannot model higher-order relations such as the one used in our tracker: finding a labeling such that the labels of an observation is similar to its k nearest neighbors. This requires modeling higher-order relations, and the lack of this ability may cause the linear methods to be not as stable. Therefore, [25], [11] and [12] optimizes second-order relations based on quadratic boolean programming, Lagrangian Relaxation and the Frank-Wolfe algorithm respectively. Another line of work utilizes continuous energy minimization [2] and discrete-continuous optimization [3] to perform tracking. Also, [37, 27] utilizes Conditional Random Fields for data association.

However, it is not straightforward to perform identity-aware tracking with many existing trackers, because these trackers do not utilize identity information during the data association process. One naïve fix is a two step process, where the trajectories are first acquired with any existing multi-object tracker and then the identities are assigned afterwards. However, problems occur when a trajectory has identity-switches and could be assigned more than one identity, thus leading to identity-incoherent trajectories. Therefore, an identity-aware tracker should incorporate identity information directly into the data association process to create identity-coherent trajectories.

Three existing identity-aware trackers are described below. First, [7, 40] utilizes the number on sports jerseys or face recognition for identity-aware tracking, but the tracker is formulated as a linear program and suffers from the drawbacks mentioned in the previous paragraphs. Also, in multi-camera scenarios, discretization of the tracking space is required to aggregate the detections from multiple cameras, and some precision may be lost. Second, [14] is based on network flow hence also a linear program, where the identity-aware edge weights were assigned by a target-specific appearance model learned with structured learning. Finally, [39] is formulated as a quadratic program, but [39] does not include the spatial locality constraint in the loss function, which might produce unreasonable tracking results where the same person is at multiple places at the same time.

Finally, most methods do not address the uncertainty of a tracker's prediction. [7] mentioned that the non-integer results acquired from their linear-programming-based tracker can also be interpreted as uncertainty of tracking output, but no experiments were performed in this direction. Active learning for efficient manual refinement of tracking results has been explored in [35], which utilizes a *single* object tracker to track multiple objects. In our case, we propose a *multi*-object tracker which can aid active learning.

3. Tracker Formulation

Following the tracking-by-detection paradigm [29], the main input to our tracker are person detections from all video frames and sparse face recognition information, which can be viewed as sparse labels. Our tracker's goal is then to assign identity labels to all person detections except for the false positives. In the following paragraphs, we will describe how we formulate our tracker as a quadratic loss function with ℓ_0 norm constraints. Then, in Section 4, we will describe how the loss function is solved with the solution path algorithm.

3.1. Notations

The notations used throughout this section are defined as follows. For a matrix $\mathbf{B}$, we denote $\mathbf{B}_{ij}$ as the element located at row i and column j of $\mathbf{B}$. We denote $\mathbf{B}_i$ as the i-th row of $\mathbf{B}$. For a vector $\mathbf{b}$, we denote $\mathbf{b}_i$ as the i-th element of $\mathbf{b}$. Given a vector $\mathbf{x} \in \mathbb{R}^b$, the ℓ_p norm of $\mathbf{x}$ is defined as $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{b} |\mathbf{x}_i|^p\right)^{\frac{1}{p}}$. $Tr(\cdot)$ denotes the trace operator.

A person detection result is referred to as an *observation*. Suppose there are n observations generated by the person

detector. These observations could be from a single camera or multiple cameras. As single camera is a special case of multiple cameras, we will only discuss the multi-camera case, where we map all observations from different cameras to a unified 3D coordinate system. Let $\mathbf{P} \in \mathbb{R}^{n\times 3}$, where $\mathbf{P}_i$ corresponds to the (x, y, z) location of observation i. Let $\mathbf{t} \in \mathbb{R}^n$, where $\mathbf{t}_i$ corresponds to the time when observation i was observed. Let $\mathbf{H} \in \mathbb{R}^{n\times d}$ be a matrix representing all the d-dimensional color histograms of each observation. Let c be the number of tracked individuals, which can be determined by either a pre-defined gallery of faces or the number of unique individuals identified by the face recognition algorithm. Note that the number of trajectories found can be larger than c, as each individual can enter and exit the scene multiple times. We denote the velocity required to go from observation i to j as $v_{ij} = \frac{\max(||\mathbf{P}_i - \mathbf{P}_j||_2 - \delta, 0)}{|\mathbf{t}_i - \mathbf{t}_j| + \epsilon}$. ϵ is a small number to prevent division by zero, and δ is the maximum localization error, which is used to compensate for camera calibration errors.

The multi-object tracking task is to assign each of the n (non-false-positive) observations a class label, where each class corresponds to one individual. The learned label assignments are encoded in the label matrix $\mathbf{F} \in \mathbb{R}^{n\times c}$. $0 \leq \mathbf{F}_{ij} \leq 1$, and a larger value corresponds to higher chance of observation i belonging to class j. The set of available face recognition information (and annotations from active learning) are represented by $\mathcal{Y} = \{(i, j) \mid \text{observation } i \text{ is recognized as individual } j\}$.

3.2. Modeling Appearance and Spatial Affinity

Following [39], we encode the appearance and spatial information of person detections using a manifold learning approach. This method is advantageous in that we are taking into account appearance and spatial information from multiple other observations to decide an observation's label.

Appearance Affinity Modeling: Based on the assumption that two observations with similar appearance are likely to belong to the same individual, we find nearest neighbors for each data point with the following criteria to build the manifold structure. Observation j is a suitable candidate for a nearest neighbor of observation i if 1) j is reachable with reasonable velocity, i.e. $v_{ij} \leq V$, 2) i and j should not be too far apart in time, i.e. $|\mathbf{t}_i - \mathbf{t}_j| \leq T$, and 3) both observations should look similar, i.e. the similarity of color histograms $\mathbf{H}_i$ and $\mathbf{H}_j$ should be larger than a threshold τ. Here, V is the maximum velocity a person can achieve. T limits how far we look for nearest neighbors in the time axis, because if two observations are separated too far in terms of time, even if they have very similar appearance, they may still not belong to the same individual. The similarity between two histograms is computed with the exponential-χ^2 metric: $\chi^2(\mathbf{H}_i, \mathbf{H}_j) = exp\left(-\frac{1}{2}\sum_{l=1}^{d}\frac{(\mathbf{H}_{il} - \mathbf{H}_{jl})^2}{\mathbf{H}_{il} + \mathbf{H}_{jl}}\right)$. For observation i, let $\mathcal{Q}_i$ be the set of up to k most similar observations which satisfy the aforementioned criteria 1, 2 and 3. We can then compute the sparse affinity matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ as follows. If $j \in \mathcal{Q}_i$, then $\mathbf{W}_{ij} = \chi^2(\mathbf{H}_i, \mathbf{H}_j)$. Otherwise $\mathbf{W}_{ij} = 0$. The diagonal degree matrix $\mathbf{D}$ of $\mathbf{W}$ is computed, i.e. $\mathbf{D}_{ii} = \sum_{l=1}^{n}\mathbf{W}_{il}$. Then, the Laplacian matrix which captures the manifold structure in the appearance space is $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

Spatial Affinity Modeling: Other than modeling observations with similar appearances, observations which are a few centimeters apart in neighboring frames are also very likely to belong to the same individual. This is reasonable in a multi-camera scenario, where detections from different cameras correspond to the same person, but due to calibration errors these person detections will not all project to the same 3D point. In this case, regardless of the appearance differences which may be due to non-color-calibrated cameras, these observations should belong to the same individual. We encode this information with another Laplacian matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$. Let $\mathcal{K}_i$ be the set of observations which are less than distance $\tilde{D}$ away and less than $\tilde{T}$ frames away from point i. We then compute the affinity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ from $\mathcal{K}_i$ by setting $\mathbf{A}_{ij} = 1$ if $j \in \mathcal{K}_i$ and $\mathbf{A}_{ij} = 0$ otherwise. Let $\hat{\mathbf{D}} \in \mathbb{R}^{n\times n}$ be the diagonal degree matrix of $\mathbf{A}$, i.e. $\hat{\mathbf{D}}_{ii} = \sum_{l=1}^{n}\mathbf{A}_{il}$. We then compute the normalized Laplacian matrix [28] $\mathbf{K} = \mathbf{I} - \hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{A}\hat{\mathbf{D}}^{-\frac{1}{2}}$, which encodes spatial affinity.

A preliminary loss function encoding the appearance and spatial affinity is as follows:

$$\min_{\mathbf{F}} \; Tr\left(\mathbf{F}^T(\mathbf{L} + \mathbf{K})\mathbf{F}\right) \; s.t. \; \forall (a, b) \in \mathcal{Y}, \mathbf{F}_{ab} = 1. \quad (1)$$

Minimizing the quadratic term of the loss function will lead to finding a labeling such that the inferred labels have similar labeling as its neighbors [6, 39]. The constraints enforce the final solution to be consistent with the face recognition results, which also prevents the trivial solution for the quadratic term. This constraint also assumes that all face recognition results are correct, which is reasonable as face recognition systems are approaching human-level performance [26]. However, this may not be an accurate assumption in all scenarios, so analysis on how face recognition errors affect tracking performance are detailed in the supplementary materials. In the next section, we add the mutual exclusion and spatial locality constraints into our loss function.

3.3. Modeling the Mutual Exclusion and Spatial Locality Constraint

We model the mutual exclusion and spatial locality constraint by adding constraints on the label matrix $\mathbf{F}$. The mutual exclusion constraint means that an observation can only belong to at most a single class, which corresponds to each row of $\mathbf{F}$ having at most one non-zero value. Mathematically, this is $\|\mathbf{F}_i\|_0 \leq 1, \; 1 \leq i \leq n$. This formulation

can also handle false positive observations by having all elements in one row of $\mathbf{F}$ be all zeros, i.e. this observation does not belong to any class.

The spatial locality constraint enforces that a person cannot be at multiple places at the same time. We incorporate the constraint by modeling pairwise observation constraints. Given a pair of observations (i, j), if the velocity v_{ij} required to move from one observation to the other is too large, then it is highly unlikely that the pair belong to the same person. Mathematically, if two observations (i, j) cannot belong to the same class, then the elements $\mathbf{F}_{il}$ and $\mathbf{F}_{jl}$ cannot both be non-zero for all $1 \leq l \leq c$. This constraint can also be modeled with a ℓ_0 norm constraint, i.e. $\left\|\begin{bmatrix}\mathbf{F}_{il} & \mathbf{F}_{jl}\end{bmatrix}\right\|_0 \leq 1, 1 \leq l \leq c$. We denote $\mathcal{T} = \{(i, j) \mid v_{ij} > V\}$ as all the observation pairs which are unlikely to be of the same individual.

The final loss function is as follows:

$$\begin{aligned} &\min_{\mathbf{F}} \; Tr\left(\mathbf{F}^T(\mathbf{L}+\mathbf{K})\mathbf{F}\right) \\ &s.t.\ \forall(a,b) \in \mathcal{Y}, F_{ab} = 1,\ \|\mathbf{F}_r\|_0 \leq 1,\ 1 \leq r \leq n \\ &\forall(i,j) \in \mathcal{T}, \left\|\begin{bmatrix}\mathbf{F}_{il} & \mathbf{F}_{jl}\end{bmatrix}\right\|_0 \leq 1, 1 \leq l \leq c. \end{aligned} \tag{2}$$

The three constraints in the loss function correspond to face recognition, mutual exclusion and spatial locality constraints. However, due to the ℓ_0 norm constraints, this loss function is difficult to optimize. Therefore, we utilized the solution path algorithm to solve Equation 2.

4. Solution Path Algorithm Optimization

The solution path algorithm acts like a bridge between two solutions: the easily accessible solution under ℓ_1 norm constraints, and the hard to obtain solution under ℓ_0 norm constraints. This is achieved by computing the solution path w.r.t. ℓ_p, $0 \leq p \leq 1$. Initialized with the solution under ℓ_1 norm constraints, the algorithm successively solves the same optimization problem but under different ℓ_p norm constraints, where p gradually decreases from 1 to 0. The intuition is that if the p from the previous iteration does not differ too much with the current p, the solution from the previous parameter setting could be a good initialization for the current parameter setting. Since our algorithm has a good initialization (ℓ_1 norm, convex global solution), and also p decreases slowly, the solution path algorithm is more likely to converge to a better local minimum compared to direct minimization under the ℓ_0 norm constraints or inaccurately solving the problem under ℓ_1 constraints.

More specifically, we denote $p^{(m)}$ as the p used during the m-th iteration of the path algorithm, and $p^{(1)} = 1$, $p^{(M)} \to 0$, and $p^{(m-1)} > p^{(m)}$ for $2 \leq m \leq M$. At iteration m, we compute the solution $\mathbf{F}^{(m)}$ of the loss function in Equation 2 under $\ell_{p^{(m)}}$ norm constraints with block coordinate descent. $\mathbf{F}^{(m-1)}$ is used to initialize iteration m's optimization process. The solution to $\mathbf{F}^{(1)}$ under ℓ_1 norm constraints is solved with random initial values because Equation 2 under ℓ_1 norm constraints is convex. The set $\mathcal{F} = \left\{\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \ldots, \mathbf{F}^{(M)}\right\}$ denotes the solution path. We now describe how we utilize block coordinate descent to compute $\mathbf{F}^{(m)}$.

4.1. Block Coordinate Descent

To optimize for $\mathbf{F}^{(m)}$, we utilize block coordinate descent [10] by only updating the variables for a single observation i while keeping all other observations fixed. A random ordering is utilized to select which observation to update. For notational clarity, we denote $\mathbf{G} = \mathbf{F}^{(m)}$ and p refers to $p^{(m)}$. We denote the i-th row of $\mathbf{G}$ as $\mathbf{G}_i = \begin{bmatrix}\mathbf{G}_{i1}, & \ldots, & \mathbf{G}_{ic}\end{bmatrix}$. We fix all other variables and only optimize for $\mathbf{G}_i$, thus the loss function is simplified from Equation 2 to the following quadratic function:

$$\begin{aligned} &\min_{\mathbf{G}_i} \frac{1}{2}\|\mathbf{G}_i - \mathbf{a}\|_2^2 \;\; s.t. \;\; \|\mathbf{G}_i\|_p \leq 1, \\ &(i, \forall j) \in \mathcal{T}, \left\|\begin{bmatrix}\mathbf{G}_{il}, & \mathbf{G}_{jl}\end{bmatrix}\right\|_p \leq 1, 1 \leq l \leq c, \end{aligned} \tag{3}$$

where $\mathbf{a} \in \mathbb{R}^c$. Each element l in $\mathbf{a}$ is computed as follows: $\mathbf{a}_l = \frac{\sum_{j=1}^{n}(\mathbf{W}_{ij}+\mathbf{A}_{ij})\mathbf{G}_{jl}}{\sum_{j=1}^{n}(\mathbf{W}_{ij}+\mathbf{A}_{ij})}$. $\mathbf{a}$ encodes the label information of the neighbors of observation i. $\mathbf{W}$ and $\mathbf{A}$ are the similarity matrices defined in Section 3.2. If the i-th row contains a recognized face, then $\mathbf{G}_i$ is not updated.

We further relax Equation 3. Due to the spatial locality constraint, $\mathbf{G}_{il} \leq \left(1 - \mathbf{G}_{jl}^p\right)^{\frac{1}{p}}$ for all j where $(i, j) \in \mathcal{T}$. As $\mathbf{G}_{jl}$ for all j are fixed, we can combine all such constraints and acquire an upper bound of $\mathbf{G}_{il}$ which we denote as $\mathbf{u}_l$. Let $\mathbf{u} \in \mathbb{R}^c$ represent the upper bound of each element in $\mathbf{G}_i$. Then, we clip the values in each dimension of $\mathbf{a}$ with $\mathbf{u}$ to acquire $\mathbf{a}'$, i.e. $\mathbf{a}'_l = \min(\mathbf{a}_l, \mathbf{u}_l)$. We arrive at the following relaxed loss function:

$$\min_{\mathbf{G}_i} \|\mathbf{G}_i - \mathbf{a}'\|_2^2 \;\; s.t. \;\; \|\mathbf{G}_i\|_p \leq 1. \tag{4}$$

If $\|\mathbf{a}'\|_p \leq 1$, then the solution of $\mathbf{G}_i = \mathbf{a}'$. Otherwise, we propose to solve Equation 4 by our proposed iterative projection method.

4.2. Iterative Projection Method

We propose an iterative projection method (IPM) to find a local minimum of Equation 4 when $\|\mathbf{a}'\|_p > 1$. For convenience, we refer to the region which satisfies $\left\{\mathbf{v} \mid \|\mathbf{v}\|_p \leq 1\right\}$ as the ℓ_p *norm ball*. At the beginning, given $\mathbf{a}'$ with $\|\mathbf{a}'\|_p > 1$, we first draw a line between $\mathbf{a}'$ and the origin. As shown in Figure 2, let $\mathbf{G}_i^{(1)} \in \mathbb{R}^c$ be the place where the line intersects the boundary of the ℓ_p norm ball in iteration 1, i.e. $\left\|\mathbf{G}_i^{(1)}\right\|_p = 1$. This intersection can be found

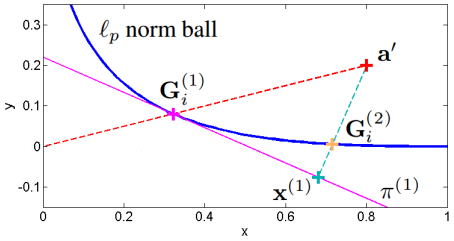


Figure 2: First iteration of iterative projection method. $\mathbf{G}_i^{(2)}$ is derived from $\mathbf{G}_i^{(1)}$, which is derived from $\mathbf{a}'$.

Input: locations $\mathbf{P}$, timing $\mathbf{t}$, histograms $\mathbf{H}$, $p^{(1)}, \ldots, p^{(M)}$
Output: solution path $\mathcal{F}$

Compute Laplacian matrices $\mathbf{L}$, $\mathbf{K}$; // (Sec. 3.2)
// Sol. of convex ℓ_1 norm constraint as initialization
$\mathcal{F} \leftarrow \{\mathbf{F}^{(1)}\}$; // Solution path set
$m \leftarrow 2$; // path algorithm iteration count
repeat // Solution path algorithm (Sec. 4)
  $\mathbf{G} \leftarrow \mathbf{F}^{(m-1)}$ // Initialize with $\mathbf{F}^{(m-1)}$
  // Solve Equation 2 under $\ell_{p^{(m)}}$ norm constraint
  repeat // Block coordinate descent (Sec. 4.1)
    for $\forall$ *observations* i **do**
      Update $\mathbf{G}_i$ by solving Equation 3 with IPM (Sec. 4.2)
    end
  until *convergence*;
  $\mathbf{F}^{(m)} \leftarrow \mathbf{G}$;
  Add $\mathbf{F}^{(m)}$ into $\mathcal{F}$;
  $m \leftarrow m + 1$;
until $m \geq M$;
return $\mathcal{F}$

Algorithm 1: Solution path tracking algorithm.

efficiently with binary search. Now, at iteration $l > 1$, given $\mathbf{G}_i^{(l-1)}$ from iteration $l-1$, we compute $\mathbf{G}_i^{(l)}$ as follows. We first compute the tangent plane $\pi^{(l-1)}$ of the ℓ_p norm ball at $\mathbf{G}_i^{(l-1)}$. Next we project $\mathbf{a}'$ onto $\pi^{(l-1)}$ and denote the projection as $\mathbf{x}^{(l-1)}$. We then draw a line between $\mathbf{a}'$ and $\mathbf{x}^{(l-1)}$ and find the intersection of the line with the ℓ_p norm ball. We denote the intersection as $\mathbf{G}_i^{(l)}$.These steps are repeated till convergence. The loss function monotonically decreases with such an update rule, which is proved in the supplementary materials.

4.3. Wrapping Up

Once we acquire the final $\mathbf{F}^{(M)}$ which satisfies the exclusion constraints, we compute the trajectories by simply connecting neighboring observations belonging to the same class. At one time instant, if there are multiple observations belonging to a person, which is common in multi-camera scenarios, then the weighted average location is computed. The weights are based on the scores in $\mathbf{F}^{(M)}$. A simple filtering process is also utilized to remove sporadic predictions. Algorithm 1 summarizes our method. The computational complexity of our method is detailed in the supplementary materials.

5. Experiments[1]

Data Sets: We utilized three data sets: *terrace1*, *nursing home short* and *nursing home long* to evaluate our tracker. *terrace1* [17] is a 4 camera sequence consisting of 9 people walking around in a 7.5m by 11m rectangle for around 5,000 frames. The scene is very crowded as shown in Figure 3. The *nursing home short* and *nursing home long* data sets are 15 camera sequences recorded in a nursing home [38]. *nursing home short* is originally from [39] and consists of 13 people performing their daily activities in a nursing home for around 11,000 frames. This data set is challenging in that the indoor environment is very complex, including many occlusion caused by walls and long corridors. The *nursing home long* data set is an extension of *nursing home short* and consists of 7 hours 45 minutes of video per the 15 cameras. Ground truth was annotated every minute, and 49 individuals were identified.

Evaluation Metrics: Multiple evaluation metrics were utilized. First, the *Multiple Object Tracking Accuracy* (MOTA[2]) from the CLEAR metrics [9] was used. This is the most popular metric used to evaluate multi-object trackers. It takes into account the number of true positives (TP), false positives (FP), missed detections (MD) and identity switches (ID-S). Following [39], tracking results and ground truth are matched if they are less than 1 meter apart. Following [7], MOTA is computed as follows: $\text{MOTA} = 1 - \frac{\#\text{ FP}+\#\text{ MD}+\log_{10}(\#\text{ ID-S})}{\#\text{ ground truth}}$.

However, the TP count in MOTA does not take into account the identity of a person, which is unreasonable for identity aware tracking. Therefore, we compute identity-aware true positives (I-TP), which means that a detection is only a true positive if 1) it is less than 1 meter from the ground-truth and 2) the identities match. Similarly, we can compute I-FP and I-MD, which enables us to compute classification-based metrics such as micro-precision ($MP = \frac{\#\text{ I-TP}}{\#\text{ I-TP} + \#\text{ I-FP}}$), micro-recall ($MR = \frac{\#\text{ I-TP}}{\#\text{ I-TP} + \#\text{ I-MD}}$) and micro-F1 ($\frac{2 \times \text{MP} \times \text{MR}}{\text{MP+MR}}$) for each tracker. The *micro*-based performance evaluation takes into account the length (in terms of time) of each person's trajectory, thus a person who appears more often has larger influence on the final scores.

[1]Source code and data sets used in the paper are released here: https://sites.google.com/site/solutionpath2016/

[2]Code from https://github.com/glisanti/CLEAR-MOT [5].

Baselines: As our main focus is on identity-aware tracking, existing trackers which could not generate identity-coherent trajectories as discussed in Section 2 were not compared.

Multi-Commodity Network Flow (MCNF): MCNF [7] creates multiple layers of the network-flow tracking graph, where each layer corresponds to an identity group. Face recognition and global appearance templates were used for identification. Two different person localization methods were used in our paper. The original paper [8, 7] utilized Probabilistic Occupancy Maps (POM) [17] for localization, which we found to be ineffective on the *nursing home short* sequence. Therefore, we also provide POM-like person localization results computed from person detections (*MCNF with PD*). This is computed by first discretizing the tracking space into many cells. Then the score of each cell is computed according to the density of person detections in the cell. We solved the data association problem with Gurobi [1].

Lagrangian Relaxation (LR): [14] imposes mutual exclusion constraints for identity-aware tracking in a network flow framework very similar to MCNF, where each identity has their own identity specific edges. The weights of the Lagrange multipliers, which enforce the mutual exclusion constraint over mutual-exclusive edges in the graph, are learned with Lagrangian Relaxation. To fairly compare different data association methods, our LR-based tracker utilizes the same appearance information used by all our other trackers, thus the structured learning and densely sampled windows proposed in [14] were not used. Specifically, LR uses the same POM-like input and network as MCNF.

Non-Negative Discretization (NND): NND [39] formulates identity-aware tracking as a constrained quadratic optimization problem and solves it with nonnegative matrix optimization techniques. The biggest problem with NND is that it does not incorporate the spatial locality constraint in the optimization step, and the solutions acquired may suggest that the same person is at multiple places at the same time. NND also requires the start and end locations of each track, which we could not provide as it is nearly impossible to identify all start and end locations of each track, especially on the 7 hour 45 minute long 15 camera *nursing home long* sequence.

Implementation Details: We utilized off-the-shelf models [16, 18] for person detection, which were further mapped into a global 3D coordinate system based on the provided camera calibration parameters. Color histograms for person detections were computed following [39]. We split the bounding box horizontally into pyramids of regions [23] and computed the HSV color histogram for each region. Given L layers, we will have $2^L - 1$ partitions for each template. L was 3 in our experiments. For POM, background subtraction was performed with [33]. Face information was acquired from the PittPatt software. The software also provides a tool to group detected faces into multiple clusters, where each cluster only contains faces of a single individual, but there could be multiple clusters representing the same single individual. Therefore, for evaluation purposes, these clusters were manually mapped into their corresponding grouth-truth identities.

Figure 3: Snapshots of tracking results on *terrace1* data set.

For our method, the parameters for all three data sets were as follows. The number of nearest neighbors used for appearance-based manifold construction was $k = 25$, and the threshold for color histogram similarity was $\tau = 0.85$. The window to search for appearance-based nearest neighbors was $T = 8$ seconds. The fastest velocity one could walk was $V = 3$ m/s. The parameters to create $\mathbf{K}$ were very conservatively set to $\tilde{D} = 20$ cm and $\tilde{T} = 6$ frames. The maximum localization error was $\delta = 125$cm. Also, the step size for p was 1.05, i.e. $p^{(m+1)} \leftarrow p^{(m)}/1.05$, and there were a total of $M = 94$ different p values. $p^{(M)} = 0.01$.

5.1. Tracking Results

Quantitative tracking results are shown in Table 1, and qualitative results[3] are shown in Figure 3. For quantitative results, our tracker is run 5 times and the performance of the run with median F1-score is reported to take into account the randomness in our method. In terms of both evaluation metrics, our tracker outperforms the state-of-the-art on all three data sets, except on the MOTA metric for *terrace1* where our method is slightly lower. For the other methods, NND performs very poorly on crowded sequences such as *terrace1* as it does not take the spatial locality constraint into account. MCNF performs well on *terrace1* as POM is effective in that sequence. However, POM creates many false positives in the complex indoor nursing home environment, which has non-ideal camera coverage that causes ambiguities in POM localization, thus leading to poor POM-based tracking on *nursing home short*. Nevertheless, if person detections (PD) were used, then the *MCNF with PD* run still performs well.

We also tested the performance of our method under different step sizes s (used in $p^{(m+1)} \leftarrow p^{(m)}/s$) for varying values of p on the *terrace1* sequence as shown in Figure 4.

[3]Due to space constraints, qualitative results for the nursing home sequences are in the supplementary materials.

Method	Micro-Precision	Micro-Recall	**Micro-F1**	TP	FN	FP	ID-S	**MOTA**
MCNF with POM	0.593	0.532	0.561	21864	3298	644	197	**0.844**
LR with POM	0.609	0.478	0.535	19216	5996	521	147	0.743
NND	0.613	0.238	0.343	8035	17267	1771	57	0.249
Ours	0.752	0.705	**0.727**	22263	2996	1407	100	0.826

(a) Tracking performance on *terrace1* sequence: 4 cameras, 5000 frames each, 9 individuals.

Method	Micro-Precision	Micro-Recall	**Micro-F1**	TP	FN	FP	ID-S	**MOTA**
MCNF with POM	0.117	0.238	0.157	23493	9769	44452	757	-0.594
MCNF with PD	0.746	0.578	0.652	19941	13749	5927	329	0.422
LR with PD	0.802	0.565	0.663	19415	14408	4203	196	0.453
NND	0.861	0.726	0.787	25628	8364	3100	27	0.663
Ours	0.871	0.755	**0.809**	26531	7458	3004	30	**0.692**

(b) Tracking performance on *nursing home short* sequence: 15 cameras, 11310 frames each, 13 individuals.

Method	Micro-Precision	Micro-Recall	**Micro-F1**	TP	FN	FP	ID-S	**MOTA**
MCNF with PD	0.743	0.418	0.535	265	347	71	25	0.342
LR with PD	0.787	0.405	0.535	261	360	52	16	0.351
NND	0.588	0.505	0.543	314	281	174	42	0.283
Ours	0.650	0.581	**0.614**	375	236	152	26	**0.389**

(c) Tracking performance on *nursing home long* sequence: 15 cameras, 7 hours 45 minutes each, 49 individuals.

Table 1: Tracking performance on 3 sequences. POM: Probabilistic Occupancy Map [17] as input. PD: Person detection as input. *MCNF with POM* was not run on the *nursing home long* sequence because it already performs poorly on the *nursing home short* sequence.

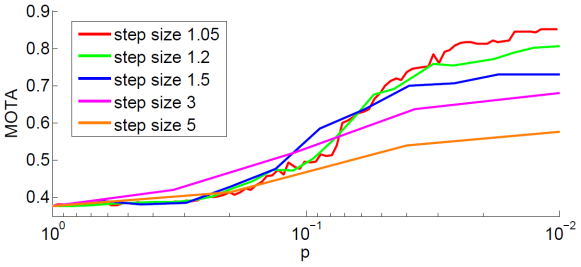


Figure 4: Tracking performance on *terrace1* data set under different step sizes and ℓ_p norm constraints.

We can see that smaller step size leads to better MOTA scores, which shows that directly optimizing under ℓ_0 norm constraints (i.e. very large step size) may converge to bad local minimum. Also, we see big performance drop if one only solved the tracking problem under ℓ_1 norm constraints, which only achieves MOTA 0.378. However, by using the solution path algorithm to compute the solution under ℓ_0 norm constraints, we were able to improve MOTA to 0.83.

5.2. Analyzing the Solution Path

We demonstrate how to utilized the solution path to locate potential tracking errors. The solution path records the class membership values in the label matrix $\mathbf{F}$ for all values of p. If the solution path for an observation shows high scores for multiple individuals as shown in Figure 1b, this may indicate that the tracker is uncertain, which can be captured with the entropy measure [32]. Specifically, for iteration m,

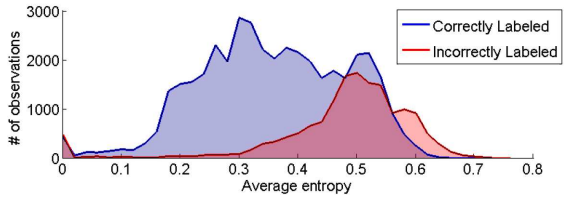


Figure 5: Entropy histogram of correctly and incorrectly labeled observations for *terrace1*.

we denote $\mathbf{G}_i = \mathbf{F}_i^{(m)}$ as the score distribution for the i-th observation. According to the mutual exclusion constraint, $\|\mathbf{G}_i\|_{p^{(m)}} \leq 1$, i.e. $\sum_{j=1}^{c} \mathbf{G}_{ij}^{p^{(m)}} \leq 1$. Then, we can view $\mathbf{G}_{ij}^{p^{(m)}}$ for all $1 \leq j \leq c$ as a probability distribution[4] and compute the entropy as follows: $e_i^{(m)} = -\sum_{j=1}^{c} \mathbf{G}_{ij}^{p^{(m)}} \log\left(\mathbf{G}_{ij}^{p^{(m)}}\right)$. $e_i^{(m)}$ captures the spread of the score distribution for observation i at iteration m. We compute the uncertainty of an observation by summing the entropy of the observation for iterations where $p \in [0.01, 0.1]$, as we observe most fluctuation in this range, i.e. $\bar{e}_i = \sum_{p^{(m)} \in [0.01,0.1]} e_i^{(m)}$.

There are other methods to compute uncertainty, such as computing the residual error for each observation. However, the residual error for a sample is only a single number, but our method provides richer information as the decision process for the whole solution path is taken into account. One

[4]The sum of $\sum_{j=1}^{c} \mathbf{G}_{ij}^{p^{(m)}}$ may not always be 1 if the constraint is not tight, but it is usually 1 in most cases.

may argue that we can also utilize the intermediate results of other optimization methods and treat them as the decision making process. However, these unconverged intermediate results do not have an obvious physical meaning. For our case, the solution path consists of converged solutions from multiple unique optimization problems. Each solution has clear physical meanings: the class membership hypothesis given the current strictness (value of p) of the mutual exclusion and spatial locality constraint. In sum, our entropy measure provides deeper insights to the tracking process.

To validate our entropy metric, we plotted the histogram of $\bar{e}_i$ for both correctly and incorrectly assigned observations. The histogram shown in Figure 5 shows that incorrect observations tend to have larger entropy, thus supporting our claims. In the next section, we demonstrate the usage of the entropy measure for improving multi-object tracking in an active learning scenario.

5.3. Active Learning for Multi-Object Tracking

In challenging scenarios, it is inevitable for trackers to make mistakes, and it would be very useful if the tracker can pinpoint potential errors for human verification and labeling. However, human verification is very expensive, thus one should first present to the human annotators the instances which will improve the classifier the most if the labels of the instances were acquired. The task of automatically pinpointing such instances is called active learning [31]. A widely used heuristic is *uncertainty sampling*, i.e. selecting the instances of which the classifier is least certain. Uncertainty sampling is a good fit to our entropy measure, which reflects the uncertainty of our tracker's output.

To evaluate our entropy-based sampling method, we performed active learning experiments as follows. The tracker is run iteratively, and after each iteration, the tracker automatically identifies the 5 most confusing observations and requests for their labels. The additional labels are added into $\mathcal{Y}$ and the tracker is rerun. This iteration is repeated 10 times. To select the confusing observations, three methods were utilized: sampling based on entropy values, sampling based on time difference from closest labeled instance (baseline), and sampling based on residual error (baseline). The sampling based on entropy values favors the observations with higher entropy, i.e., the probability of the i-th observation being selected is $\frac{1}{\text{Rank}(\bar{e}_i)}$, where the rank instead of the absolute entropy values were used to favor higher ranked observations. Time difference sampling favors observations which are furthest away in terms of time from any labeled instance, thus having higher likelihood of having an identity switch. Residual error sampling favors observations which have high residual error in the final optimization result. High residual error may indicate that this observation is incorrect. During sampling, we also add a simple filter to avoid sampling all 5 observations from the same region (within 1 meter) and time (within 2 seconds). The results shown in Figure 6 demonstrates that our entropy-based sampling significantly beats the baselines. Time difference sampling also shows some gains, but residual sampling performs poorly because high residual error mostly occurs near observations with recognized faces. Recognized faces have a fixed score of 1, but scores of neighboring observations will be significantly smaller (e.g. 0.3 or less), thus causing large residual error. Labeling observations near recognized faces is not helpful in improving tracking because they are already highly likely to be correct. In sum, our entropy measure for uncertainty sampling is effective in active learning.

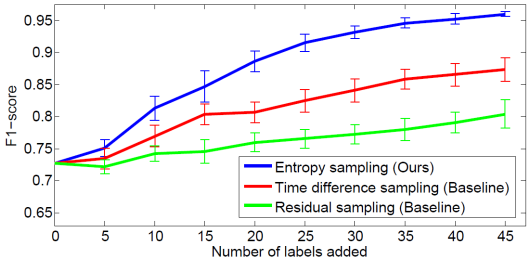


Figure 6: Performance of tracker in each iteration of active learning on *terrace1* data set. Experiments for the two sampling methods were performed 20 times each, and the 95% confidence interval are drawn.

6. Conclusion and Future Work

We propose a multi-object tracker which utilizes the solution path algorithm to solve a quadratic problem with ℓ_0 norm constraints. The ℓ_0 norm constraints enforce the mutual exclusion and spatial locality constraint. The solution path also provides insights into the "decision making process" of the tracker, thus enabling us to identify uncertainty in the tracking output. Experiments show that our tracker is not only effective, but also the uncertainty can be utilized in an active learning framework to more efficiently enhance tracking results. Future work includes 1) applying the solution path algorithm to other problems with ℓ_0 norm constraints, 2) utilizing the computed solution path to locate uncertainty in other tasks, and 3) taking into account face recognition errors during the optimization process.

Acknowledgements

This paper was partially supported by the U.S. Army Research Office (W911NF-13-1-0277) and partially supported by the National Science Foundation under Grant Number IIS-12511827. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ARO and NSF. This work is supported in part by NSFC grant (61271093).

References

[1] Gurobi optimizer reference manual, http://www.gurobi.com, 2012. 6

[2] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR*, 2011. 2

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 2

[4] S.-H. B. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, 2014. 2

[5] A. D. Bagdanov, A. Del Bimbo, F. Dini, G. Lisanti, and I. Masi. Posterity logging of imagery for video surveillance. In *IEEE Multimedia*, 2012. 5

[6] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. In *Neural computation*, 2003. 3

[7] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-commodity network flow for tracking multiple people. In *TPAMI*, 2014. 2, 5, 6

[8] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. In *TPAMI*, 2011. 2, 6

[9] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. In *J. Image Video Process.*, 2008. 5

[10] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009. 4

[11] A. Butt and R. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR*, 2013. 2

[12] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic. On pairwise costs for network flow multi-object tracking. In *CVPR*, 2015. 2

[13] R. T. Collins. Multitarget data association with higher-order motion models. In *CVPR*, 2012. 2

[14] A. Dehghan, Y. Tian, P. H. Torr, and M. Shah. Target identity-aware network flow for online multiple target tracking. In *CVPR*, 2015. 2, 6

[15] C. Dicle, M. Sznaier, and O. Camps. The way they move: Tracking targets with similar appearance. In *ICCV*, 2013. 2

[16] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *TPAMI*, 2010. 6

[17] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. In *TPAMI*, 2008. 5, 6, 7

[18] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/. 6

[19] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *JMLR*, 2004. 1

[20] H. Jiang, S. Fels, and J. J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 2

[21] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010. 2

[22] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011. 2

[23] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 6

[24] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, 2014. 2

[25] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *CVPR*, 2007. 2

[26] C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with GaussianFace. *arXiv preprint arXiv:1404.3840*, 2014. 3

[27] A. Milan, K. Schindler, and S. Roth. Detection-and trajectory-level exclusion in multiple object tracking. In *CVPR*, 2013. 2

[28] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002. 3

[29] K. Okuma, A. Taleghani, N. D. Freitas, O. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004. 2

[30] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2

[31] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 2, 8

[32] C. Shannon. A mathematical theory of communication. In *Bell Systems Technical Journal, vol. 27, pp.379-423,623-656*, 1948. 7

[33] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999. 6

[34] R. J. Tibshirani. *The solution path of the generalized lasso*. Stanford University, 2011. 1

[35] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. In *NIPS*, 2011. 2

[36] X. Wang, E. Türetken, F. Fleuret, and P. Fua. Tracking interacting objects optimally using integer programming. In *ECCV*, 2014. 2

[37] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012. 2

[38] Y. Yang, A. Hauptmann, M.-Y. Chen, Y. Cai, A. Bharucha, and H. Wactlar. Learning to predict health status of geriatric patients from observational data. In *Computational Intelligence in Bioinformatics and Computational Biology*, 2012. 5

[39] S.-I. Yu, Y. Yang, and A. Hauptmann. Harry potter's marauder's map: Localizing and tracking multiple persons-of-interest by nonnegative discretization. In *CVPR*, 2013. 2, 3, 5, 6

[40] M. Zervos, H. BenShitrit, F. Fleuret, and P. Fua. Facial descriptors for identity-preserving multiple people tracking. Technical report epfl-article-187534, 2013. 2

[41] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 2