The Way They Move: Tracking Multiple Targets with Similar Appearance

Caglayan Dicle
Northeastern University
dicle.c@husky.neu.edu

Mario Sznaier
Northeastern University
msznaier@coe.neu.edu

Octavia Camps
Northeastern University
camps@coe.nue.edu

Abstract

We introduce a computationally efficient algorithm for multi-object tracking by detection that addresses four main challenges: appearance similarity among targets, missing data due to targets being out of the field of view or occluded behind other objects, crossing trajectories, and camera motion. The proposed method uses motion dynamics as a cue to distinguish targets with similar appearance, minimize target mis-identification and recover missing data. Computational efficiency is achieved by using a Generalized Linear Assignment (GLA) coupled with efficient procedures to recover missing data and estimate the complexity of the underlying dynamics. The proposed approach works with tracklets of arbitrary length and does not assume a dynamical model a priori, yet it captures the overall motion dynamics of the targets. Experiments using challenging videos show that this framework can handle complex target motions, non-stationary cameras and long occlusions, on scenarios where appearance cues are not available or poor.

1. Introduction

Recent advances in the accuracy and efficiency of object detectors [14, 17], particularly pedestrian detectors, have inspired and fueled multi-target tracking approaches by detection. These techniques proceed by detecting the targets frame by frame using a high quality object detector and then associating these detections by using online or offline trackers [7, 32, 33, 35]. Often, these associations are based on appearance and location similarity, while the start and end of the tracks are handled using "source" and "sink" nodes. These approaches achieve very good results for scenarios, such as pedestrian tracking, where the appearance of the targets is discriminative, the targets display simple motion patterns, and source and sink nodes can be naturally placed at the boundaries of the field of view. [7, 12, 22, 35]. However, these algorithms do not perform as well when targets have similar appearance, do not move with the assumed dynamics, or come out in the middle of the field of view as in the example shown in Figure 1. While there are trackers that rely less on appearance [3, 9, 10, 13, 31], they often require tuning of a large number of parameters and expertise to adapt the algorithms to these more challenging scenarios.

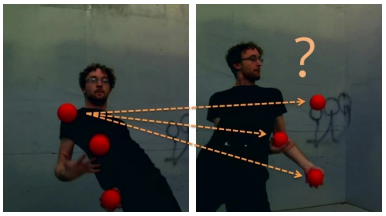


Figure 1. It is hard to say which ball is which. Their appearance does not help, but their motion aids to disambiguate them.

It is also possible to track solely based on dynamics using Kalman [21] or particle [20] filters to predict the target location and associate the closest detection to this prediction. However, these approaches must assume a dynamic model a priori and have trouble distinguishing close to each other targets. Alternatively, Ding *et al*. [15] showed that it is possible to use dynamics to compare tracks and disambiguate between targets without assuming a motion model a priori. Instead, comparisons are based on the complexity of the underlying dynamics which is estimated by minimizing the rank of a Hankel matrix constructed directly from the available data, potentially fragmented and corrupted by noise. Since rank minimization is an NP hard problem [15] used a convex relaxation and a generic interior point (IP) method to first complete the matrix such that it has minimum nuclear norm (a surrogate for low rank), followed by a singular value decomposition (SVD) and singular value thresholding to minimize rank. However, the computational and memory complexity of IP methods is $O(l^6)$ and $O(l^4)$, respectively, where l is the length of the trajectory. Thus, until now this approach has been limited to stitching short trajectories of a few targets.

In this paper we propose a framework to use motion dynamics for multi-target tracking by detection that can efficiently handle large numbers of targets, long trajectories, missing data, and arbitrary motions. This is accomplished by i) formulating the problem as a generalized linear assignment (GLA) of tracklets which are incrementally associated

into longer trajectories based on their dynamics-based similarity, and ii) using efficient algorithms to estimate these similarity measures.

1.1. Contributions

The benefit of using a GLA approach is that it avoids the need to make an a priori commitment about the start and termination nodes for location or time of the considered trajectories.

We explore two algorithms that differ on the method they use to estimate dynamics similarity. The first method replaces the IP optimization step used in [15] with an alternating direction method of multipliers (ADMM) [4] with computational complexity $O(l^3)$ and low memory requirements. Similarly to IP methods, this approach solves a relaxation instead of the original problem and requires setting a parameter weighting the noise penalty and a singular value threshold to estimate the rank which are both difficult to choose. The second method IHTLS (iterative Hankel Total Least Squares) method is a new algorithm that we propose to directly estimate the rank of incomplete, noisy Hankel matrices. The algorithm is based on a noise cleaning algorithm for Hankel matrices introduced in [25] that we modified to handle missing data and estimate rank. The advantages of IHTLS is that it solves the original problem instead of a convex relaxation, it does not require choosing a singular values threshold and it has computational complexity $O((l-n)n^3)$, where $n < l$ is the rank of the Hankel matrix. The disadvantage of this approach is that it uses a Newton's method to solve a non-convex problem and hence can be trapped in local minima. However, experimental evidence shows that in practice it converges to the true optimum.

These algorithms were tested and compared against state-of-the-art approaches [8, 13] on a set of videos with challenging scenarios where targets are difficult to discriminate based on appearance alone. The dataset, annotated with ground truth target locations, is available for public use at our website. The experiments show that multi-target tracking using IHTLS performs faster and more accurately than when using ADMM and that both techniques perform significantly better and faster than the state of the art, where performance is measured using the MOTA metric.

Finally, it should be noted that the proposed methods are complementary to appearance-based methods: they can improve the performance of appearance-based methods when visual discrimination is possible, yet retain target identities when such information is not available.

1.2. Other Related Work

Often, multi-target tracking is formulated as a network flow problem [35] or variations of it [6, 8, 26, 27, 31]. Most methods rely to a large degree on target appearance and assume simple motion models a priori that work well in settings where the targets are pedestrians or vehicles. However, their performance deteriorates in more challenging scenarios as shown in Figure 2 where the algorithm from [8] fails to track the balls shown in Figure 1.

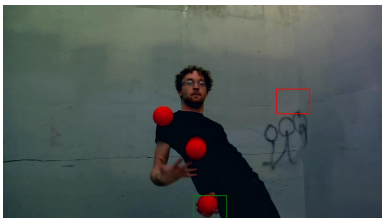


Figure 2. The method of [8] fails due to complex dynamics and unexpected start and ending locations for the targets trajectories.

A drawback of using a network flow formulation is that it requires setting up beginning and ending locations and/or times of the targets which, depending on the scenario, may be hard to pinpoint in advance[1]. An alternative to network flow are max-clique type formulations. Brendel *et al*. [12] used a maximum weighted set formulation but they only consider two-frames relations. Recently, Zamir *et al*. [34] proposed using a General Maximum Clique Partitioning formulation picking one-best candidate from each tracklet to achieve global association. The GLA formulation used here is similar to the Linear Assignment formulation of [13, 8, 35], but it has the advantage that allows tracks to start and terminate anywhere in position and time. In spirit, we are also similar to [19] and [34] since we also solve the associations iteratively from easy to hard. Like [34] our algorithm can operate at the tracklet level, but we use all the data on a tracklet rather than a selected portion. In addition, we do not assume any priors for the target motion as in [13, 31]. This allows our algorithm to capture long term dynamics as targets may change their motion behavior over time. Lastly, the dynamics based similarity measure used by our algorithm can deal with different types of scenes, target and camera dynamics (including zooming) while providing a natural way to "inpaint" missing data.

There are few multi-target trackers for non-human targets, and they either rely on appearance [12] or prior motion models [9]. One of the goals of our work is to ease these requirements. More recently, [13] and [3] formulate the multi-target tracking problem using higher order motion models which are one-step better than models previously used but still limited. The computational complexity of [13] is $O(d^{2.5})$ where d is the number of detections and the algorithm in [3] requires a large set of parameters. In contrast, our method can handle arbitrarily high order motions with a run time $O(N^2)$, where $N << d$ is the number of tracklets and requires only two parameters.

[1] For example, a person coming out of a car in the middle of the field of view.

2. Dynamics-based Multi-tracklet Association

Given a set of short tracklets, possibly of different lengths and with no appearance information, we want to associate together those that belong to the same trajectory.

There are four challenges that makes this association task difficult 1. Lack of appearance information, 2. Object crossings, 3. Object occlusions, and 4. Camera motion. To address these challenges, we formulate the multi-target tracking problem as a Generalized Linear Assignment (GLA) using a dynamics-based similarity measure.

2.1. Generalized Linear Assignment Problem

Given N tracklets $\{\alpha_1, \ldots, \alpha_N\}$, the Linear Assignment (LA) Problem is stated as the optimization problem:

$$\max_X \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}X_{ij} \qquad (1)$$
$$st. \sum_{i=1}^{N} X_{ij} = 1 \; ; \sum_{j=1}^{N} X_{ij} = 1 \; ; X_{ij} \in \{0,1\}$$

where P_{ij} is a suitable similarity measure between α_i and α_j, such that P is a predecessor-successor matrix, i.e. P_{ij} is $-\infty$ if α_j cannot follow α_i in time. The optimization variable X_{ij} indicates that α_i is the predecessor of α_j when $X_{ij} = 1$ and that they will be merged considering their gap.

Equation (1) is the max-flow formulation used by [8, 35] where the constraints enforce that each tracklet has to be assigned to one predecessor and one successor. In general, the problem is augmented with source and sink nodes to simulate the entrance and termination of the tracklets. To avoid this requirement, we use the Generalized Linear Assignment (GLA) [28, 29] formulation (2):

$$\max_X \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}X_{ij} \qquad (2)$$
$$st. \sum_{i=1}^{N} X_{ij} \leq 1 \; ; \sum_{j=1}^{N} X_{ij} \leq 1 \; ; X_{ij} \in \{0,1\}$$

The main difference between LA and GLA, is that in the latter, tracklets are not forced to begin, terminate or associate with any other tracklet. While at a first glance, this seems to be a small change, it has two important consequences: 1. avoids setting up sink source nodes and learning tracklet entrance and termination probabilities; and 2. GLA is an NP-Complete problem which is harder to solve than the LA problem. However, under very mild constraints it can be approximately solved using the deterministic annealing "softassign" algorithm [18]. In our experiments, it was observed that the dynamics-based similarity measure we use reduces possible ambiguities and leads softassign to fast and accurate convergence (on average converges in 10 iterations and never takes more than 100 iterations).

2.2. Tracklet Dynamics and Similarity Measure

A tracklet α consists of an ordered sequence of measurements y_k, $s \leq k \leq e$, where s and e are the starting and ending times, respectively. The underlying dynamics of the tracklet can be represented using a linear regressor, since linear regressors are universal approximators [11]:

$$y_k = \sum_{i=1}^{n} a_i y_{k-i} \;\; , \;\; k \geq s+n \qquad (3)$$

for a high enough value of n. The order of the regressor n measures the "complexity" of the underlying dynamics and in the absence of noise, $n = rank(H_\alpha^{(m)})$ where $H_\alpha^{(m)}$ is the Hankel matrix with $m \geq n$ columns:

$$H_\alpha^{(m)} \doteq \begin{bmatrix} y_s & y_{s+1} & \cdots & y_{s+m-1} \\ y_{s+1} & y_{s+2} & \cdots & y_{s+m} \\ \vdots & \vdots & \vdots & \vdots \\ y_{e-m+1} & y_{e-m} & \cdots & y_e \end{bmatrix} \qquad (4)$$

Then, the dynamics-based similarity P_{ij} between two tracklets α_i, α_j is defined [15] as:

$$P_{ij} \doteq \begin{cases} -\infty & \text{if } \alpha_i \text{ and } \alpha_j \text{ conflict} \\ \frac{rank(H_{\alpha_i}) + rank(H_{\alpha_j})}{\min_{\beta_i^j} rank(H_{\alpha_{ij}})} - 1 & \text{otherwise} \end{cases} \qquad (5)$$

where $\alpha_{ij} = [\alpha_i \; \beta_i^j \; \alpha_j]$ is the joint tracklet padded with tracklet β_i^j at the gap between α_i and α_j values.

The intuition behind the above similarity measure is that if two tracklets are portions of the same trajectory they can be approximated by a single, relatively low order regressor. On the other hand, if two tracklets belong to different trajectories, explaining a merged/joined trajectory requires a higher order regressor than the regressors of each tracklet[2]. Thus, intuitively, if $rank(H_{\alpha_i}) = r_i$ and $rank(H_{\alpha_j}) = r_j$, then $rank(H_{\alpha_{ij}}) = r_{ij} \leq (r_i + r_j)$. Accordingly, if α_i and α_j belong to the same trajectory, then $r_i = r_j = r_{ij}$ and $P_{ij} = 1$, but if α_i and α_j are not related $P_{ij} \approx 0$.

2.3. Dynamics-based Similarity Computation

A major challenge in computing (5) is that one has to estimate the rank of noisy and incomplete structured matrices. [15] addressed this problem by solving a convex relaxation: $\min_{\beta_i^j} \|H_{\alpha_{ij}}\|_*$ with the additional Hankel structural constraints[3]: $H \in \mathcal{S}_\mathcal{H}$ where $\mathcal{S}_\mathcal{H}$ is the set of Hankel matrices and where $\|\|_*$ denotes the nuclear norm using IP methods, and then estimating the rank of $H_{\alpha_{ij}}$ by thresholding its singular values. However, this requires computing a Hessian

[2] This assumes that the object does not drastically change dynamics between tracklets. This is a fair assumption for tracklets that are close to each other in time and akin to assume that appearance will stay similar over-time and change slowly in appearance-based target tracking [7, 12, 19].

[3] That is, the matrices must have block-constant off-diagonals.

with $O(l^4)$ memory and $O(l^6)$ computational complexity and hence does not scale well for long trajectories.

Recently, Li *et al.* [23] introduced a different similarity score, based on the subspace angle between the column spaces of the Hankel matrices, that is efficient to compute and does not require estimating rank. However, since the subspace angle is invariant to the initial conditions of the trajectories, it cannot be used to distinguish two targets with the same dynamics and hence is not suitable for this application. We propose two alternatives to address these issues:

Rank estimation using ADMM An alternative to using IP methods is to solve a similar convex relaxation using ADMM [4]:

$$\min_{\beta_i^j} \|A_{\alpha_{ij}}\|_* + \lambda \|E_{\alpha_{ij}}\|_1$$

such that $H = A + E \in \mathcal{S}_{\mathcal{H}}$. In contrast to IP methods, ADMM does not require computing the Hessian, and it is solved by computing a number of SVDs. The procedure has very modest memory requirements and computational cost $O(l^3)$ at each iteration. Thus, it scales well as the length of the trajectories increase.

Rank estimation using IHTLS We propose as a second alternative a new algorithm to estimate the rank of an incomplete Hankel matrix corrupted with additive noise. The algorithm is based on two simple modifications of the Hankel Total Least Squares (HTLS) algorithm [25] which allow us to handle missing data and to estimate rank.

The first modification is the introduction of an "indicator" binary vector to flag missing data and allow its recovery while performing inpainting to stitch tracklets with gaps. The second modification is to run this algorithm, iteratively, for increasing rank values to find the optimal rank. Thus, like the approaches in [15] and in [4] the algorithm, described in detail below, does not only estimate the rank, but also cleans and completes the data while respecting the structural constraint that $H \in \mathcal{S}_{\mathcal{H}}$. However, it does so without relaxing the original problem, at the lower computational cost of $O((l-n)n^3)$, where l is the length of the tracklet and $n < l$ is the rank of the matrix [30].

More formally, consider a trajectory of length l, $\alpha = \{y_s, \ldots, y_e\}$, with known dynamics complexity $n < l$. Let $\hat{\alpha} = \{\hat{y}_s, \ldots, \hat{y}_e\}$ and $\eta = \{\eta_s, \ldots, \eta_e\}$ be the noiseless measurements and the noise, respectively. Let ω be a binary "indicator" l-length vector where 1s and 0s indicate available and missing measurements, respectively. For simplicity of notation, let $[A|b] = [H_\alpha^{(n)}|b]$ and $[E|f] = [-H_\eta^{(n)}|f]$, where $b = \begin{bmatrix} y_{s+n}^T & \ldots & y_e^T \end{bmatrix}^T$ and $f = -\begin{bmatrix} \eta_{s+n}^T & \ldots & \eta_e^T \end{bmatrix}^T$ Then, from (3) and (4) there exist a regressor x such that: $(A+E)x = (b+f)$ and α can be estimated by solving the following modified Total Least Squares (TLS) problem [25],

$$\min_{x,E,f} \; ||\Omega \circ [E|f]||_F^2 \qquad (6)$$
$$st.\ (A+E)x = b + f \;\; ; \;\; [A|b], [E|f] \in \mathcal{S}_{\mathcal{H}}$$

where $\circ$ is the Hadamard product and $\Omega = H_\omega^{(n+1)}$ is introduced to recover the missing data. It should be noted that the general TLS problem (6) without missing data has a closed form solution for general matrices which can be found by computing the SVD of $[A|b]$ [24]. However, adding the structural constraints precludes a close form solution since SVDs do not preserve the Hankel structure. Thus, as shown below, we will follow [25] and solve the HTLS problem by using Newton's method which converges in a few iterations.

In the sequel, for the sake of simplicity, we present the solution for the one scalar measurements. Generalization to the multi-dimensional case is straight forward.

Since $[E|f]$ and $[A|b]$ are Hankel matrices with constant off-diagonals, we can rewrite (6) as,

$$\min_{\eta,x} \; ||WD\eta||_2^2 \;\; st. \;\; r(\eta,x) \doteq b + f - (A+E)x = 0 \quad (7)$$

where D is a diagonal matrix with the number of times each η_i appears in the Hankel matrix $H_\eta^{(n+1)}$ and $W = diag(\omega)$. Then, following the method of [25] we can combine the constraint with the minimization problem:

$$\min_{\eta,x} \; \left|\left| \begin{pmatrix} \pi r(\eta,x) \\ WD\eta \end{pmatrix} \right|\right|_2^2 \qquad (8)$$

where π is a large penalty constant. Next, write $r(\eta,x)$

$$r(\eta,x) = b + P_1\eta - (A+E)x \qquad (9)$$

where $P_1 = [0_{m\times n} \; I_{m\times m}]$ and $m = l - n$. Linearizing $r(\eta,x)$ we have,

$$r(\eta+\delta\eta, x+\delta x) \approx r(\eta,x) + P_1\delta\eta - (A+E)\delta x - \delta E x$$

Then, there exist a matrix $X \in \mathcal{R}^{m\times(m+n-1)}$ that satisfies,

$$Ex = XP_0\eta \qquad (10)$$

where $P_0 = [I_{(m+n-1)\times(m+n-1)} \; 0_{(m+n-1)\times 1}]$. Finally, we can write (8) as,

$$\min_{\delta\eta,\delta x} \; \left|\left| \begin{pmatrix} \pi(P_1 - XP_0) & -\pi(A+E) \\ WD & 0 \end{pmatrix} \begin{pmatrix} \delta\eta \\ \delta x \end{pmatrix} + \begin{pmatrix} \pi r \\ WD\eta \end{pmatrix} \right|\right|_2^2 \qquad (11)$$

which can be solved by least squares leading to the procedure shown in Algorithm 1. Finally, Algorithm 2 summarizes the rank estimation procedure.

Figure 3. Tracking results for *crowd, slalom, juggling, and TUD-crossing* sequences using IHTLS. Dashed bounding boxes indicate that the algorithm performed inpainting to recover missing data.

Algorithm 1: HTLS with Missing Data

Input: α sequence of length l, ω sampling sequence of length l, desired rank n
Output: $\hat{\alpha}$ inpainted and cleaned sequence, η noise/perturbation, x AR coefficients

Form $[A|b]_{(l-n+1)\times(n+1)}$
Solve min $||Ax-b||_2^2$ for x
Form P_1 and WD from ω
$\eta = 0$
while $\left|\left|\begin{pmatrix}\delta\eta\\ \delta x\end{pmatrix}\right|\right| > \theta$ **do**
  Form XP_0 from x and form $[E|f]_{(l-n+1)\times(n+1)}$
  Compute $r = b + f - (A+E)x$
  Form $M = \begin{pmatrix}\pi(P_1 - XP_0) & -\pi(A+E)\\ WD & 0\end{pmatrix}$
  Solve min $\left|\left|M\begin{pmatrix}\delta\eta\\ \delta x\end{pmatrix} + \begin{pmatrix}\pi r\\ WD\eta\end{pmatrix}\right|\right|_2^2$ for $\delta\eta$, δx
  Update $x = x + \delta x$,$\eta = \eta + \delta\eta$

Algorithm 2: Iterative Hankel Total Least Squares

Input: α sequence of length l, η_{max} maximum average error, ω sampling sequence
Output: $\hat{\alpha}$ inpainted and cleaned sequence, n estimated rank for the sequence

$n = 0$, $\mu_\eta =$ huge
Form $\Omega_{(l-n)\times(n+1)} = H_\omega^{(n+1)}$
while $\mu_\eta > \eta_{max}$ **do**
  n = n + 1;
  Solve HTLS problem, $min||\Omega \circ [E|f]||_F$ st. $(A+E)x = b + f$
  Form $[E|f]_{(l-n)\times(n+1)} = \mathcal{H}(\eta)$
  Compute average error, $\mu_\eta = \frac{||\Omega\circ[E|f]||_F}{||\Omega||_1}$

3. Implementation Details

We used a simple heuristic to generate short tracklets of length 3 or longer by stitching non-conflicting detections. A detection pair is non-conflicting if the ratio of their distance is smaller than 0.3 to the second closest detection. To speed up the algorithm we processed the input sequences with non-overlapping windows. We did three passes with offset windows. A time window of $(40-60)$ frames worked well for all videos with a final pass twice the window size. This scheme helped us to merge the trajectories from easiest to hardest – i.e. we fill the gaps that are proportional to the average tracklet length. As tracklets got larger, longer gaps were filled iteratively. Lastly, before terminating the algorithm all tracklets of length shorter than 5 were eliminated.

4. Experiments

In order to evaluate the proposed approach, we collected a set of challenging videos with multiple targets with identical or very similar appearance to assemble the *Similar MultiObject Tracking* (SMOT) dataset, available at our website. The SMOT dataset consists of eight videos four of which were downloaded from YouTube while the remaining ones

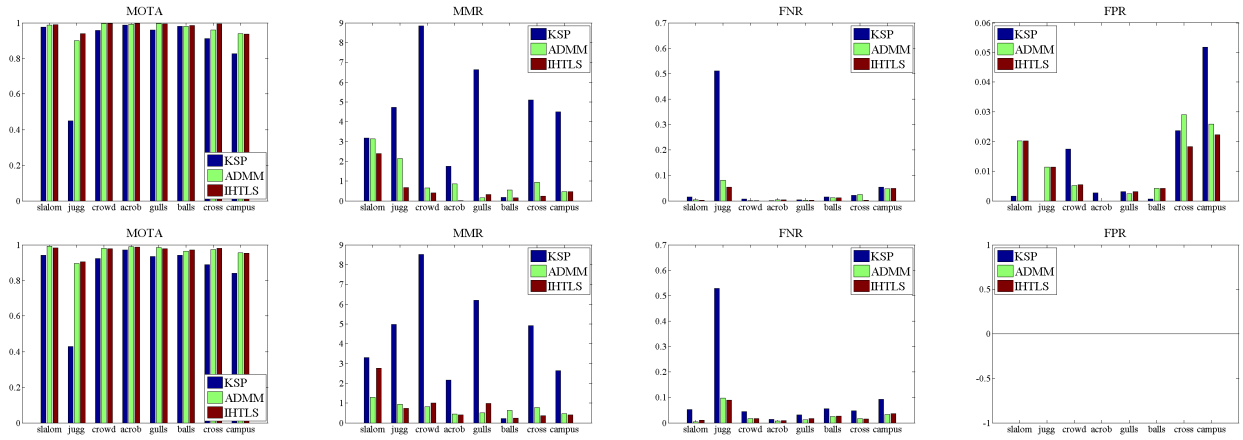


Figure 4. Results for MOTA, MMR, FNR and FPR Top: with 20% outliers. Bottom: with 12% missing data.

are from [1], [8] and [2]. In order to decouple the performance of the trackers from the performance of the detectors, all detections were hand labeled using the video labeling tool from [16], and then during the experiments, detections were randomly deleted and added in a controlled way.

4.1. SMOT Sequences

Slalom has three skiers racing down a slalom. They perform complex zig-zag motions and get close to each other very frequently. One skier escapes out of the field of view for a long time. It exhibits camera motion and zooming.

Juggling is a 3-ball juggling scene and is the hardest sequence in the dataset. The juggler adds artistic motions to the performance with alternating tricks. The motion of the balls, juggler and the camera combined makes this sequence incredibly hard even for a human to keep track of the balls.

Acrobats is a short sequence from Cirque De Soleil acrobats from the Academy Awards 2012. In this sequence all the acrobats are dressed the same; they lineup in the air and get occluded several times.

Seagulls shows a flock of seagulls taking off at sea. This is another extremely difficult sequence where seagulls fly close to each other and get occluded very frequently.

TUD-Campus and ***TUD-Crossing*** are from [2] and are often used to evaluate pedestrian tracking algorithms. They have long occlusions, closely moving pedestrians.

Crowd is from the crowd UCF dataset [1]. It is an overcrowded surveillance scene where the detections are the heads of the people. Due to the density of the crowd, there are frequent occlusions among the closely moving targets.

Balls is from [8]. These are a approximately 50 randomly bouncing identical ping pong balls.

4.2. Performance Evaluation

Methods. We compared the performance of the tracking algorithms using dynamics based similarity measures estimated using IP [15], ADMM [5], and IHTLS methods and the algorithm proposed in [8] (We will refer to this algorithm as KSP). IP was implemented in CVX as described in [15] and we used the code provided by the authors for ADMM and KSP. The code for IHTLS was implemented using Matlab. Additionally, we ran experiments to compare against [13]. However, since the code for this algorithm is not available, we could only compare by running our algorithms on their data. Furthermore, since this data does not provide information about the image boundaries, it was not possible to test KSP in this experiment.

Parameters. For ADMM, λ was set to 0.1. The singular value threshold η_{max} was set between $(0.3-3)$ for different videos but kept constant across algorithms. For KSP, we used a 64×64 grid and set the borders of the image as source and sink locations as it is required by the algorithm. Finally, a maximum depth of $\{1, 2, 3\}$ was used whichever performed best for the sequence.

Set of experiments. We conducted two set of experiments with increasing random false positives and increasing random false negatives. Each test was run ten times and the resulting performances were averaged. In the first set, we increased the number of false positives by injecting uniformly distributed false detections. In the second set, we introduced false negatives by uniformly removing true detections. The input for all the algorithms was a set of (x, y) detections for each time instant. We want to emphasize that no other information was used in the tests. All methods were tested against increasing false positives and false negatives with the exception of IP. This method was tested only for the zero false positive and zero false negative case due to its very high computational cost (see Figure 6).

Evaluation. We report the performance of the algorithms using four different measures. We use the standard

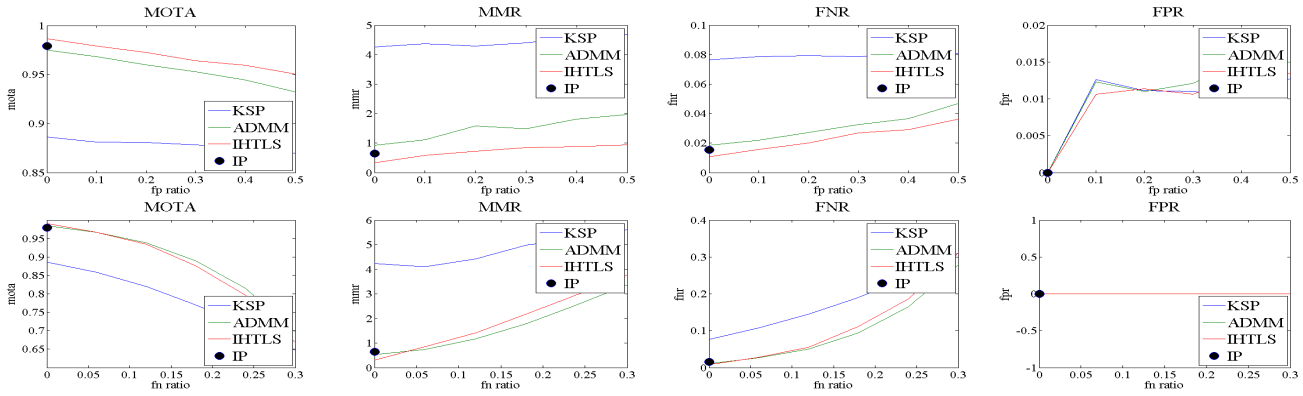


Figure 5. Results for MOTA, MMR, FNR and FPR. Top: for increasing false positives. Bottom: for increasing missing data.

MOTA measure,

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + mm_t)}{\sum_t gt_t} \quad (12)$$

where fn_t, fp_t, mm_t and gt_t are false negatives(misses), false positives, mismatches and ground truth at frame t. We called a detection a "match" if an (x, y) hypothesis was within the half width radius of the true object location.

In addition, we used three other measures to better evaluate the robustness of the algorithms to detection quality. These measures are the False Negative Ratio(FNR), False Positive Ratio(FPR) and MissMatch Ratio(MMR):

$$FNR = \frac{\sum_t fn_t}{\sum_t gt_t}, \; FPR = \frac{\sum_t fn_t}{\sum_t pt_t}, \; MMR = \frac{\sum_t fn_t}{\sum_t tt_t}, \quad (13)$$

where pt_t are the tt_t number of false positives injected to the frame t and the ground truth number of tracks at frame t. Lastly, we report average run time performance for all the algorithms.

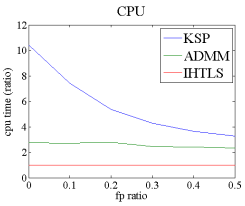
4.3. Results

Figure 3 shows sample frames of the tracking results using the IHTLS algorithm for four of the videos in the dataset, where dashed boxes indicate that the algorithm inpainted data due to occlusions.

ADMM, IHTLS and KSP were run 10 times on each video, for each level of noise. The noise was varied from 0 to 50% input outliers and from 0 to 30% missing input data. Figure 4 shows plots for MOTA, FNR, FPR and MMR scores for all the videos for noise levels of 20% outliers and 12% missing data showing that *juggling* is the most difficult video. Figure 5 shows plots of the average performances across all videos and noise levels. For the MOTA measure, IHTLS has the best overall performance, followed closely by ADMM, and KSP performed the worst. By looking at the other measures, it can be seen that the biggest difference in performance is due to mismatch errors. KSP has the largest mismatch ratio as it often jumps from one trajectory to another without doing inpainting. As a result, the trajectories obtained by KSP tend to be shorter than the ones found by ADMM and IHTLS. Table 1 gives MOTA scores comparisons with [13]. Finally, the average execution times are shown in Figure 6. Once again, IHTLS has the better performance, with running times up to close 5 times faster than KSP.

Table 1. MOTA score comparison with [13]

	MDA	**ADMM**	**IHTLS**
PSU-sparse	1.00	0.98	0.99
PSU-dense	0.87	0.94	0.97



	sec/frame	**sec/track**
KSP	0.217	5.726
ADMM	0.120	3.986
IHTLS	0.048	1.396
IP	56.341	3891.341

Figure 6. Run time comparisons: The plots show the ratios of running times with respect to IHTLS run times. The results are the average of 10 runs for each false positive rate across 8 videos.

5. Conclusion

Motion dynamics provide strong cues while tracking targets with identical or very similar appearance. We presented two efficient tracking algorithms that measure the dynamic similarity of tracklets and recover missing data due to long occlusions. This measure, coupled with a GLA framework, can be used to successfully track multiple targets under adverse conditions including lack of appearance cues, occlusions, camera motion and complex dynamics. It should be emphasized that the proposed method does not preclude the use of appearance but rather complements it. Not surprisingly, there are situations where the proposed framework

will fail to retain associations. One example is when two bodies have an elastic collision, in which case one body transfers its motion to the other. However, these type of situations require a better understanding of the physics of the scene and are beyond the scope of this paper.

References

[1] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*, pages 1–6. IEEE, 2007. 6

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, pages 1–8. IEEE, 2008. 6

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 1, 2

[4] M. Ayazoglu, M. Sznaier, and O. Camps. Fast algorithms for structured robust principal component analysis. In *CVPR*, pages 1704–1711, 2012. 2, 4

[5] M. Ayazoglu, M. Sznaier, and O. Camps. Fast algorithms for structured robust principal component analysis. In *CVPR*, pages 1704–1711. IEEE, 2012. 6

[6] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, pages 137–144. IEEE, 2011. 2

[7] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, pages 3457–3464. IEEE, 2011. 1, 3

[8] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. PAMI*, 33(9):1806–1819, 2011. 2, 3, 6

[9] M. Betke, D. Hirsh, A. Bagchi, N. Hristov, N. Makris, and T. Kunz. Tracking large variable numbers of objects in clutter. In *CVPR*, pages 1–8. IEEE, 2007. 1, 2

[10] E. Brau, K. Barnard, R. Palanivelu, D. Dunatunga, T. Tsukamoto, and P. Lee. A generative statistical model for tracking multiple smooth trajectories. In *CVPR*, pages 1137–1144. IEEE, 2011. 1

[11] L. Breiman. Hinging hyperplanes for regression, classification and function approximation. *IEEE Trans. Inf. Theory*, pages 999–101, 1993. 3

[12] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, pages 1273–1280. IEEE, 2011. 1, 2, 3

[13] R. Collins. Multitarget data association with higher-order motion models. In *CVPR*, pages 1744–1751. IEEE, 2012. 1, 2, 6, 7

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893. IEEE, 2005. 1

[15] T. Ding, M. Sznaier, and O. Camps. Fast track matching and event detection. In *CVPR*, pages 1–8. IEEE, 2008. 1, 2, 3, 4, 6

[16] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. PAMI*, 34(4):743–761, 2012. 6

[17] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8. IEEE, 2008. 1

[18] S. Gold, A. Rangarajan, et al. Softmax to softassign: Neural network algorithms for combinatorial optimization. *J. of Artificial Neural Nets.*, 2(4):381–399, 1995. 3

[19] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. *ECCV*, pages 788–801, 2008. 2, 3

[20] M. Isard and A. Blake. CONDENSATION conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998. 1

[21] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME Ser. D: J. Basic Eng.*, 83:95–108, March 1961. 1

[22] C. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, pages 685–692. IEEE, 2010. 1

[23] B. Li, O. Camps, and M. Sznaier. Cross-view activity recognition using hankelets. In *CVPR*, June 2012. 4

[24] I. Markovsky and S. Van Huffel. Overview of total least-squares methods. *Sig. Proc.*, 87(10):2283–2302, 2007. 4

[25] H. Park, L. Zhang, and J. Rosen. Low rank approximation of a hankel matrix by structured total least norm. *BIT Numerical Mathematics*, 39(4):757–779, 1999. 2, 4

[26] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268. IEEE, 2009. 2

[27] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208. IEEE, 2011. 2

[28] G. Ross and R. Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming*, 8(1):91–103, 1975. 3

[29] D. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62(1):461–474, 1993. 3

[30] S. Van Huffel, H. Park, and J. Rosen. Formulation and solution of structured total least norm problems for parameter estimation. *Sig. Proc.*, 44(10):2464–2474, 1996. 4

[31] Z. Wu, T. Kunz, and M. Betke. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *CVPR*, pages 1185–1192. IEEE, 2011. 1, 2

[32] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR*, pages 1200–1207. IEEE, 2009. 1

[33] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model. In *CVPR*, pages 1233–1240. IEEE, 2011. 1

[34] A. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. ECCV, 2012. 2

[35] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, pages 1–8. IEEE, 2008. 1, 2, 3