Fast Track Matching and Event Detection*

Tao Ding
Dept. of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

Mario Sznaier and Octavia I. Camps
Dept. of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

Abstract

This paper addresses the problems of track stitching and dynamic event detection in a sequence of frames. The input data consists of tracks, possibly fragmented due to occlusion, belonging to multiple targets. The goals are to (i) establish track identity across occlusion, and (ii) detect points where the motion of these targets undergo substantial changes. The main result of the paper is a simple, computationally inexpensive approach that achieves these goals in a unified way. Given a continuous track, the main idea is to detect changes in the dynamics by parsing it into segments according to the complexity of the model required to explain the observed data. Intuitively, changes in this complexity correspond to points where the dynamics change. Since the problem of estimating the complexity of the underlying model can be reduced to estimating the rank of a matrix constructed from the observed data, these changes can be found with a simple algorithm, computationally no more expensive that a sequence of SVDs. Proceeding along the same lines, fragmented tracks corresponding to multiple targets can be linked by searching for sets corresponding to minimal complexity joint models. As we show in the paper, this problem can be reduced to a semi-definite optimization and efficiently solved with commonly available software.

1. Introduction

A common problem in dynamic vision applications involves tracking objects in a sequence of frames. Challenges in designing a robust tracking algorithm arise from several factors, *e.g*. changing appearances, changes in illumination, clutter and occlusion. During the past decade extensive research has been carried out in this area, leading to several techniques that address these effects (see for instance [1, 6, 7, 21, 23, 8, 14, 20] and references therein). In particular, a class of dynamics based trackers has been developed that combine simple dynamic models of the target motion with optimal filtering –(unscented) Kalman, particle– [16, 9, 10] to track in the presence of occlusion. In order to further improve robustness, Camps *et al*. [2] use interpolation theory to learn the dynamics of the target, thus removing a potential source of fragility arising from a mismatch between the assumed and actual dynamics. This approach was later extended by Lim *et al*. in [12] to include changes in the appearance of the target, leading to robust trackers capable of successfully tracking objects through occlusion, even when undergoing substantial appearance changes. The implicit assumption in all these methods is that *the dynamics of the target are linear and do not change,* e.g. the underlying model is linear time invariant (LTI). Nonlinear and time varying dynamics can always be approximated arbitrarily well either by high dimensional or low order piecewise linear time invariant dynamics [18] (Chapter 10),[17]. In principle, low order piecewise LTI dynamics can be identified, as pointed out in [2], by detecting changes between models (corresponding for instance to a different motion modality) by performing a model (in)validation step to establish whether the new data is consistent with the existing model. An advantage of this approach is its ability to unequivocally establish that a change in the dynamics has taken place. However, the computational complexity associated with this process is not small, specially in the case of long sequences[1]. In addition, this approach cannot handle cases where the events occur while the target is occluded, which requires, as a pre-requisite, being able to match tracklets across the occlusion.

The problem of tracklet matching has been addressed in a number of papers: For example, [11] proposed a unified multi-object framework; [25] uses a background layer model; [19] considers a multi-camera setup with major color histogram matching; Chan and *et al*. [3] optimizes join events recognition and track linking using a Dynamic Bayes Net. While successful, these methods are fairly involved.

In this paper we propose a simple, computationally in-

*This work was supported in part by NSF grants ITR 0312558, ECS 050166, and IIS 0713003 and by AFOSR grant FA 9550-05-1-0437.

[1] Roughly speaking, the computational complexity scales as (number of frames)^5.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

expensive approach that allows for both stitching tracklets across occlusion and detecting changes in the dynamics of the target, in a unified way. The main idea is to explain continuous tracks with piecewise linear low order dynamics and to associate the switching points between dynamics to significant events. Then, in this framework, the problem of tracklet matching can be formulated as an optimization problem where tracklets are stitched to minimize the complexity of the model explaining the combined data. Key to this approach are the abilities to: 1) efficiently estimate the complexity of the model explaining the measured data; 2) formulate the complexity as a function of missing data that is amenable to optimization; and 3) determine the switching points between piecewise low order models.

The first point was addressed in [13] where Lublinerman *et al* introduced results from linear systems theory showing that the problem of estimating the complexity of the underlying dynamics of a temporal sequence *without missing data* can be done, without explicitly identifying the model, by estimating the rank of a Hankel matrix constructed from the observed data. Thus, the computational cost of estimating the model complexity explaining the data is no more expensive than computing a series of SVDs. In this paper we show that tracklet stitching can be solved as a Hankel matrix rank optimization problem where the *missing data points* are minimization variables. While this problem is in general NP hard, it can be relaxed to a semi-definite optimization problem that can be solved with comercially available tools such as Matlab. Finally, we also show that significant events – i.e. switching points – can be found by detecting sudden increases in the rank of the Hankel matrix for the observed data, which only requires the computation of a series of SVDs.

The paper is organized as follows. Section 2 summarizes the necessary background on Hankel matrices. Sections 3 and 4 present the proposed algorithms and several examples for tracklet matching and event detection, respectively. Finally, Section 5 gives conclusions.

2. Hankel Matrices and Model Order Estimation from Experimental Data

The main idea underlying this paper is to model the evolution of target features as the output of piecewise linear time invariant models whose orders must be efficiently estimated from the available experimental data. Specifically, following [2, 13], we will collect the position of all relevant features of the target in a vector $\mathbf{f}$ and assume that its value at time k is related to its past values $\mathbf{f}_{k-i}$ by a model of the form:

$$\mathbf{f}_k = \sum_{i=1}^{n} \mathbf{g}_i \mathbf{f}_{k-i} + \mathbf{h}_i \mathbf{e}_{k-i}. \quad (1)$$

where $n \leq N_f$ is the number of frames, and where $\mathbf{e}(.)$ is an exogenous stochastic input with appropriate statistics. Note that this can be always assumed without loss of generality, since given N measurements of $\mathbf{f}(.)$ and $\mathbf{e}(.)$, there always exists a linear operator such that (1) is satisfied (Chapter 10 of [18]). The goal is, given measurements y of f corrupted by additive noise:

$$\mathbf{y}_k = \mathbf{f}_k + \eta_k \quad (2)$$

to estimate the *minimum* n such that the model (1) holds. To do this, we consider the **Hankel** matrices for the given input/output sequence defined as:

$$\mathbf{H}_f(k,l) \doteq \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \cdots & \mathbf{f}_l \\ \mathbf{f}_2 & \mathbf{f}_3 & \cdots & \mathbf{f}_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{f}_k & \mathbf{f}_{k+1} & \cdots & \mathbf{f}_{k+l-1} \end{bmatrix}$$

$$\mathbf{H}_e(k,l) \doteq \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_l \\ \mathbf{e}_2 & \mathbf{e}_3 & \cdots & \mathbf{e}_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_k & \mathbf{e}_{k+1} & \cdots & \mathbf{e}_{k+l-1} \end{bmatrix}$$

$$\mathbf{H}_{f,e} \doteq \begin{bmatrix} \mathbf{H}_f(k,l) & \mathbf{H}_e(k,l) \end{bmatrix} \text{ where } l \geq k >> n. \quad (3)$$

Now, recall the following result from Linear Systems Theory relating the minimum order of the model n to the rank of the matrix computed from the (noiseless) experimental data $\mathbf{H}_{f,e}$ [24, 22]:

Fact 1 *Consider the input/output sequence* $\{\mathbf{e}_t, \mathbf{f}_t\}$ *corresponding to the model* (1)*, and the corresponding Hankel matrices,* $\mathbf{H}_f(k,l)$*,* $\mathbf{H}_e(k,l)$*, and* $\mathbf{H}_{fe}$ *where* $l \geq k >> n$*. Under mild conditions [15] the order* n *of the model* (1) *satisfies:*

$$\textit{rank}[\mathbf{H}_{f,e}] = n + \textit{rank}[\mathbf{H}_e] \quad (4)$$

.

Remark 1 *In the sequel, we will assume, by absorbing if necessary the dynamics of the stochastic input* $\mathbf{e}$ *into the dynamics of the plant, that* $\mathbf{e}$ *is an impulse, e.g.* $\mathbf{e}_1 = 1$*,* $\mathbf{e}_i = 0$*,* $i > 1$*. With this assumption we have that* $\textit{rank}[\mathbf{H}_\mathbf{e}] = 1$ *and the fact above reduces to*

$$\textit{rank}[\mathbf{H}_f(k,l)] = n \quad (5)$$

Remark 2 *A potential difficulty here is that, rather than the actual feature positions* $\mathbf{f}$*, only the measurements* $\mathbf{y} = \mathbf{f} + \eta$ *corrupted by noise are available, and it is well known that rank computation is very sensitive to noise. To avoid this difficulty, begin by noting that the Hankel matrices corresponding to the actual and measured position are related by:* $\mathbf{H}_y = \mathbf{H}_f + \mathbf{H}_\eta$*, where* $\mathbf{H}_\eta$ *denotes the Hankel matrix*

associated with the noise sequence $\eta(.)$. Under ergodicity assumptions, $\mathbf{H}_\eta^T\mathbf{H}_\eta$ is an estimate of the covariance matrix of the noise[13]. Thus, noise measurements can be robustly handled by simply replacing rank$(\mathbf{H}_y)$ by $\mathbf{NSV}_{\sigma_\eta}(\mathbf{H}_y)$, the number of singular values larger than σ_η, the standard deviation of the measurement noise.

3. Hankel Matrix Based Track Matching

In this section we address the problem of establishing track identity with missing data due to, for example, temporary occlusion. As part of the process of solving this problem we develop an algorithm that allows for efficiently estimating the missing data that connects tracklets using both past *and* future data. This is in contrast to the use of filters such as Kalman or Particle filters which predict data through the occlusion based only on the available past data.

3.1. Track stitching: Estimating missing data.

Consider first the problem of estimating the missing data connecting two segments of the same track. Formally, this can be stated as:

Problem 1 *Given two segments of a track, $\{\mathbf{y}_i^o\}$, $1 \leq i \leq r$ and $\{\mathbf{y}_j^o\}$, $s+1 \leq j \leq N_F$, with $r < s$ and $s - r \ll N_F$, estimate the missing values $\mathbf{y}_k^*$, $r+1 \leq k \leq s$ that are maximally consistent with the existing data, in the sense that the complete sequence is explained by the lowest possible order model.*

From Fact 1 it follows that the missing values $\mathbf{y}^*$ can be optimally estimated by minimizing the rank of the corresponding Hankel matrix $\mathbf{H}_y$ formed by combining $\mathbf{y}^o$ and $\mathbf{y}^*$. Unfortunately, rank minimization problems are known to be generically NP-hard. Thus, motivated by the recent development in the field [4, 5], we will replace the rank minimization step by a convex relaxation that only entails solving a tractable, convex Linear Matrix Inequality (LMI) optimization, leading to the Algorithm given in Fig. 1.

3.2. Multiple Track Matching and Stitching

The ideas discussed above can also be used to match and stitch tracks across occlusion. The main idea is to group tracks according to the complexity of the *simplest* model that explain the joint data. Specifically, given two measurement matrices $\mathcal{W}_i$ and $\mathcal{W}_j$ corresponding to two tracklets, and where $N_{S_j} > N_{F_i}$ the starting and ending frame indexes in $\mathcal{W}_j$ and $\mathcal{W}_i$, respectively, a similarity measure between tracks can be defined proceeding as follows:

1. Use Algorithm 1 to stitch the tracklets. Define $\mathcal{W}_{ij} \doteq \begin{bmatrix}\mathcal{W}_i & \mathcal{W}^* & \mathcal{W}_j\end{bmatrix}$ where $\mathcal{W}^*$ denotes the estimates of the missing measurements.

Algoritm 1: HANKEL BASED TRACK STITCHING

Input: N observed values of $\mathbf{y}$,
N_p, estimated number of missing points.
Output: Estimates $\mathbf{y}^*$ of the missing data.

1. Form a Hankel matrix $H_{\hat{y}}$, where $\hat{y}$ is the sequence formed by combining y and y^*,
$$\mathbf{H}_\mathbf{y} \doteq \begin{bmatrix} \hat{\mathbf{y}}_1 & \hat{\mathbf{y}}_2 & \cdots & \hat{\mathbf{y}}_{\frac{N_F}{2}} \\ \hat{\mathbf{y}}_2 & \hat{\mathbf{y}}_3 & \cdots & \hat{\mathbf{y}}_{\frac{N_F}{2}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{y}}_{\frac{N_F}{2}} & \cdots & \cdots & \hat{\mathbf{y}}_{N_F} \end{bmatrix},$$
where $N_F \doteq N + N_p$.
2. Obtain the best prediction $\mathbf{y}^*$ by solving the following LMI optimization problem in $\mathbf{H}_\mathbf{y}$, $\mathbf{X}$ and $\mathbf{Z}$
$$\begin{aligned} &\text{minimize} \quad Tr(\mathbf{X}) + Tr(\mathbf{Z}) \\ &\text{subject to} \quad \begin{bmatrix} \mathbf{X} & \mathbf{H}_y \\ \mathbf{H}_y{}^T & \mathbf{Z} \end{bmatrix} \geq 0 \\ &\qquad \{\mathbf{y}^* \in \mathcal{R}^2\} \end{aligned}$$
where $\mathbf{X}^T = \mathbf{X}$, $\mathbf{Z}^T = \mathbf{Z}$ are additional free optimization variables, and $\mathbf{H}_y \in \mathcal{R}^{2N_F \times \frac{N_F}{2}}$.

Figure 1. Algorithm for track stitching.

2. The similarity measure $\Gamma_{i,j}$ between tracklets $\{i, j\}$ is defined as:
$$\Gamma_{i,j} \doteq \begin{cases} -\infty, & \text{if temporal conflict exists;} \\ \frac{\mathbf{NSV}_{\sigma_\eta}(\mathbf{H}_{W_i}) + \mathbf{NSV}_{\sigma_\eta}(\mathbf{H}_{W_j})}{\mathbf{NSV}_{\sigma_\eta}(\mathbf{H}_{W_{i,j}})} - 1 \end{cases} \quad (6)$$

Intuitively, if $\mathcal{W}_i$ and $\mathcal{W}_j$ are samples of the same trajectory, then $\text{rank}(\mathbf{H}_{W_i}) = \text{rank}(\mathbf{H}_{W_j}) = \text{rank}(\mathbf{H}_{W_{ij}})$ and hence $\Gamma_{i,j} = 1$. On the other hand if $\mathcal{W}_i$ and $\mathcal{W}_j$ are uncorrelated, $\Gamma_{i,j} \approx 0$. The definition above formalizes this idea, using $\mathbf{NSV}_\sigma$ in lieu of rank(.) to improve robustness against measurement noise. Once $\Gamma_{i,j}$ is computed for all pairs that do not exhibit temporal conflicts (e.g. one track starting before the end of the second), tracks can be matched by simply looking for the pairs (i, j) corresponding to the largest values $\Gamma_{i,j}$. These ideas are summarized in *Algorithm 2* shown in Fig. 2.

3.3. Experiments

In this section we present several examples illustrating Hankel based tracklet stitching and matching in the presence of occlusion. The corresponding videos are provided as additional material.

Example 1: Occlusion with ground truth. In this example, we use Motion History Images [26] to locate moving vehicles in a traffic video sequence, where occlusion was artificially generated by blacking out frames 501 to 515. Additionally, for performance evaluation purposes, the true po-

Algoritm 2: HANKEL MATRIX BASED TRACK MATCHING

Input: measurements matrices $\mathcal{W}_i$;
n_T, total tracklet number;
noise standard deviation σ_η.
Output: Similarity matrix $\mathbf{\Gamma}$

for all $i \neq j \in \{1, \cdots, n_T\}$ **do**
  if temporal conflict exists
    Set $\mathbf{\Gamma}_{i,j}$ as -∞.
  else
    Apply **Algorithm 1** to $\mathcal{W}_i$ and $\mathcal{W}_i$ to find $\mathcal{W}_{i,j}$.
    Use (6) to calculate $\mathbf{\Gamma}_{i,j}$.
  end if
end for
Find the best matches guided by $\mathbf{\Gamma}$.

Figure 2. Algorithm for tracklet matching.

i	$\Gamma_{i,1}$	$\Gamma_{i,2}$	$\Gamma_{i,3}$	$\Gamma_{i,4}$
1	NA			
2	$-\infty$	NA		
3	$1^\dagger$	-0.17	NA	
4	-0.29	$0.14^\dagger$	$-\infty$	NA

(a)

i	$\Gamma_{i,1}$	$\Gamma_{i,2}$	$\Gamma_{i,3}$	$\Gamma_{i,4}$
1	NA			
2	$-\infty$	NA		
3	$0^\dagger$	-0.69	NA	
4	-0.73	$0^\dagger$	$-\infty$	NA

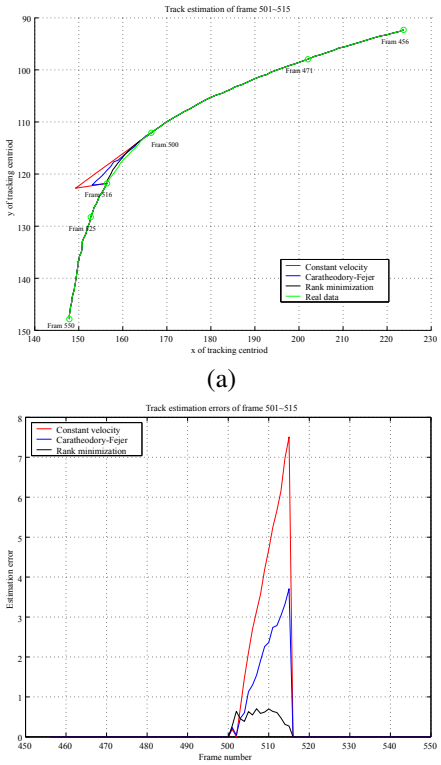
(b)

Table 1. Similarity matrices for examples (a) 2 and (b) 3.

sitions of the targets were also found without the occlusion. The performance of Hankel based stitching using data from frames 471 to 500 and 516 to 525, was compared against using simple (constant velocity) and identified (using CF identification with frames 460 to 500) dynamics to generate the missing data during the occlusion period (501 to 515). Fig. 3 shows the tracking results for frames before the occlusion (456 and 500), and after the occlusion (516 and 550) where it is seen that the estimated track using Hankel matrix rank minimization (green) closely follows the target. On the other hand, tracks using constant velocity (red) and CF identification (blue) drift off the target. Fig. 4 shows quantitatively the computed position and the error with respect to the true values.

Example 2: Multi-target track matching under occlusion (1). Fig. 5 (top) shows 34 frames of a partially occluded sequence of two balls with different dynamic behavior. Applying *Algorithm 2* to the 4 tracklets (using $\sigma_\eta = 3.5$) yields $\mathbf{NSV}(\mathcal{W}_1) = \mathbf{NSV}(\mathcal{W}_3) = 1$, $\mathbf{NSV}(\mathcal{W}_2) = \mathbf{NSV}(\mathcal{W}_4) = 4$, $\mathbf{NSV}(\mathcal{W}_{1,3}) = 1$, $\mathbf{NSV}(\mathcal{W}_{2,4}) = 7$, $\mathbf{NSV}(\mathcal{W}_{1,4}) = 7$, and $\mathbf{NSV}(\mathcal{W}_{2,3}) = 6$. The resulting similarity matrix Γ is given in Table 1 (a). As shown in Fig. 5 grouping tracks according to this matrix indeed leads to the correct assignments.

Example 3: Multi-target track matching under occlusion (2). This example, shown in the bottom portion of Fig. 5, consists of 47 frames from a partially occluded sequence of two bouncing balls. Applying *Algorithm 2* to the 4 tracklets (using $\sigma_\eta = 2.7$) yields $\mathbf{NSV}(\mathcal{W}_1) = \mathbf{NSV}(\mathcal{W}_3) = 2$, $\mathbf{NSV}(\mathcal{W}_2) = \mathbf{NSV}(\mathcal{W}_4) = 2$, $\mathbf{NSV}(\mathcal{W}_{1,3}) = 4$, $\mathbf{NSV}(\mathcal{W}_{2,4}) = 4$, $\mathbf{NSV}(\mathcal{W}_{1,4}) = 15$, and $\mathbf{NSV}(\mathcal{W}_{2,3}) = 13$. The resulting similarity matrix Γ is given in Table 1(b). As shown in Fig. 5 grouping tracks according to this matrix again leads to the correct assignments.



(a)

(b)

Figure 4. Comparison of track estimation using constant velocity, CF identification and Hankel matrix rank minimization. (a) Estimated tracks. (b) Estimation error during occlusion.

i	$\Gamma_{i,1}$	$\Gamma_{i,2}$	$\Gamma_{i,3}$	$\Gamma_{i,4}$
1	NA			
2	$-\infty$	NA		
3	$1^\dagger$	-0.38	NA	
4	0	$0.33^\dagger$	$-\infty$	NA

Table 2. Similarity matrix for the car and ball example.

The next two examples illustrate the ability of the method to exploit dynamical information to match partially overlapping tracks.

Example 4: Partially overlapping, occluded tracks. This example consists of 49 frames of the sequence shown in the top portion of Fig. 6, containing two moving objects: a ball and a car, the latter appearing only after frame 16. The

Figure 3. Multi-target track matching: comparison with constant velocity and CF interpolation.

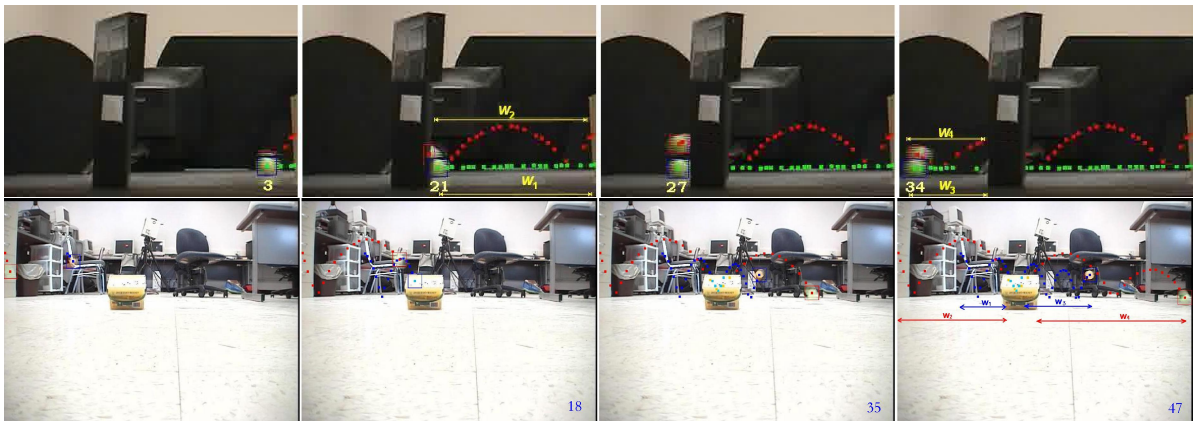
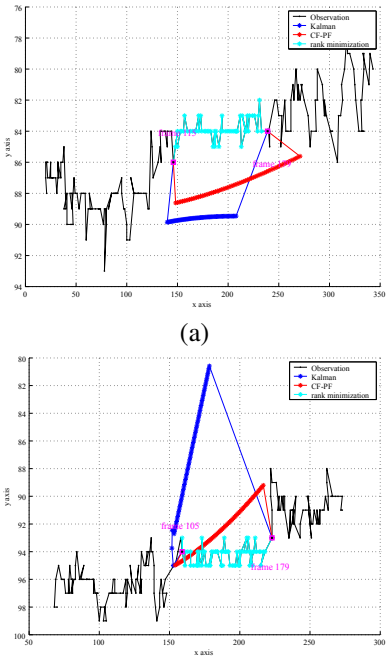


Figure 5. Multi-target track matching.



(a)

(b)

Figure 8. Comparison of track estimation during occlusion using Kalman Filter, CF identification and Particle Filtering, and Hankel matrix rank minimization. (a) Faster person. (b) Slower person.

similarity matrix Γ shown in Table 2 shows that $\mathcal{W}_{1,3}$ and $\mathcal{W}_{2,4}$ are the most consistent tracklets.

Example 5: Partially overlapping, occluded tracks, moving camera. This example, shown in Fig. 7 consists of 244 frames of two moving persons, captured with a moving camera, with occlusion in frames 110-180. As shown in the figure the proposed algorithm correctly stitches the tracklets, even though they are partially overlapping and even though the man overtakes the woman while occluded by the vehicle. Notice that the predicted positions of the targets, using a Kalman filter and CF identification with Particle filter, lag considerably with respect to the true positions of the targets while the proposed approach do not. A plot of the estimated tracks during the occlusion are also given in Fig. 8.

4. A Rank Criterion for Fast Event Detection

In this section we turn our attention to the problem of efficiently detecting dynamic events. Fact 1 can be used to perform fast detection of changes in the motion modality of the target by simply searching for points where the rank of the Hankel matrix abruptly changes after having remained approximately constant. This corresponds to a formalization of the intuitive fact that trying to explain two different modalities (distinguished either by different dynamics or different inputs) using a single model will require considerable more complexity than that required to explain each modality alone. It is worth emphasizing that the approach outlined above does not require explicitly finding the models (computationally expensive).

Figure 6. Track matching with partially overlapping tracks of a car and a ball.

Figure 7. Track matching with overlapping tracks and a moving camera: comparison between Kalman filter (blue circle and triangle), CF and Particle filter (yellow circle and triangle), and our approach (cyan circle and red triangle).

4.1. Event Detection Examples

The efficiency of this approach is illustrated next with several examples. The corresponding videos are provided as additional material.

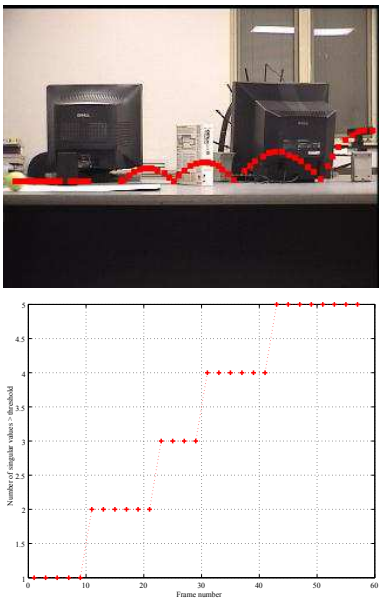


Figure 9. Detecting events using the rank criterion. Top row: input sequence showing dynamic events at frames 10, 23, 34 and 43. Second row: corresponding **NSV** plot

Example 1: Bouncing and rolling ball. Consider the bouncing ball shown in Fig. 9. The dynamics change at frames 9, 22, 33 and 43, with the first 3 changes due to impact with the table and the last to the transition from bouncing to rolling motion. All these changes are clearly evidenced by the jumps in **NSV**($\mathbf{H}$) shown in the bottom plot.

Example 2: Change in human activity: This example consists of 78 frames of the sequence shown in Fig. 10(a) of a moving person that abruptly switches from walking to jumping, starting at frame 51. As illustrated in Fig. 10(b), this change is clearly shown in the plot of NSV($\mathbf{H}$).

Example 3: Normal versus abnormal car slowdown. This example consists of two sequences showing a car undergoing deceleration, as a result of a crash (Fig. 10 (c)) and during normal braking (Fig. 10(e)). The NSV plot corresponding to the crash, shown in Fig. 10(d), exhibits a large jump starting around frame 45, indicating the occurrence of a dynamic event. For comparison, the NSV plot for the normal deceleration (Fig. 10(f)), has a much smaller jump, around frame 50, as the car slows down to a stop.

Example 4: Crash detection at street intersection. In this example, for simplicity, we only chose 10 cars and marked their tracks using MHI and template matching as shown in Fig. 11. The corresponding NSV plots ($\sigma_\eta = 1$) are shown in Fig. 12. The plots show that the ranks of the Hankel matrices for cars 1, 4, and 8 have dramatic changes at the crash time. On the other hand, the Hankel matrices for cars not involved in the accident do not change rank.

Example 5: Detecting dynamic events under occlusion. This example illustrates the ability of the proposed methods to detect event changes, even if these events occur while the target is occluded. In this example, a jumping ball exhibits different dynamics in frames 1–42 and 43–59 . The available data consists of 4 tracklets, labeled $\mathcal{W}_{1-4}$ in Fig. 13(a), with estimated noise level $\sigma_\eta = 7.75$. Applying *Algorithm 1* to stitch the track led to the connecting trajectories shown in red in the figure. Finally, the **NSV** plot shown in Fig. 13(b), clearly indicates points at where dynamic events

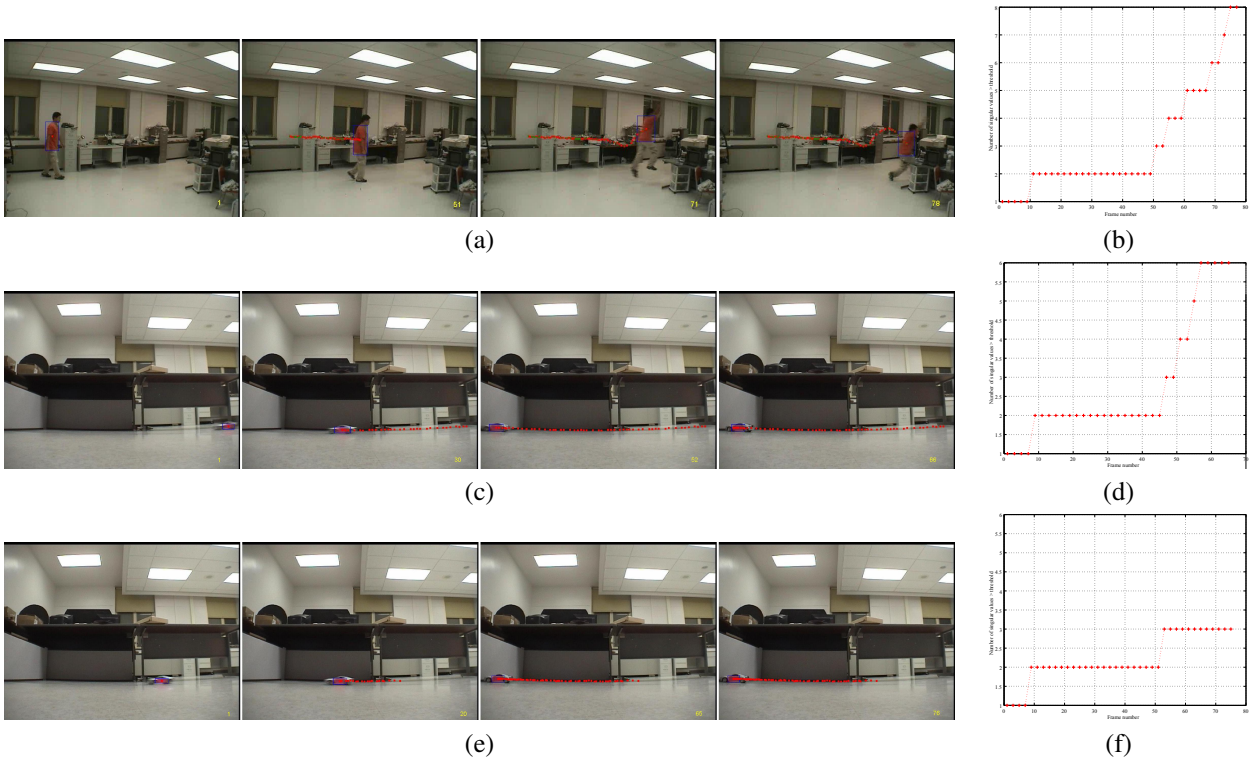


Figure 10. Event detection. (a) transition from walking to jumping. (b) Corresponding **NSV** plot. (c) Car crashing. (d) Corresponding **NSV** plot. (e) Car deceleration. (f) Corresponding **NSV** plot

Figure 11. Crash detection. Tracks of 10 cars in frame 311, 341, 381, and 429.

take place.

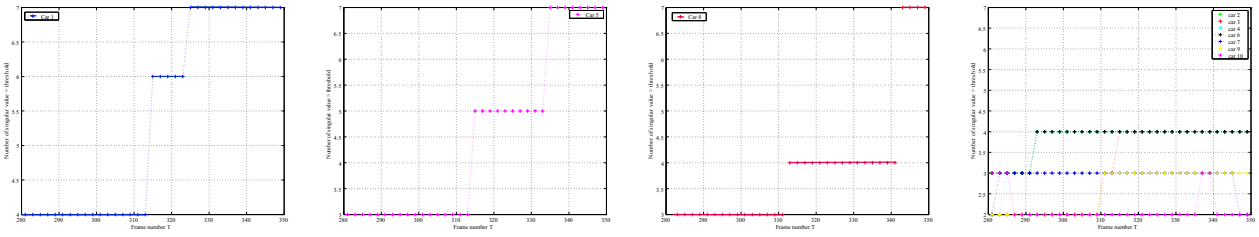
5. Conclusions

In this paper we addressed the problem of multi–target dynamical event detection using fragmented tracks. In order to solve this problem, we introduced algorithms for (i) establishing track identity across occlusion, (ii) estimating missing data and (iii) analyzing the (reconstructed) track to establish points where dynamical events took place. The underlying idea in all cases is that tracks corresponding to a single target who is not undergoing dynamic events can be explained by a model whose complexity is lower than that required to jointly explain different dynamics. The latter situation can be due for instance to having different targets or a single target performing different activities. In turn, by exploiting results from Linear Systems Theory, the problem of estimating the order of the model, can be reduced to computing the rank of a matrix constructed from the experimental data. This observation leads to fast, computationally simple algorithms that do not require finding explicit models. The effectiveness of this technique was illustrated using several examples.

References

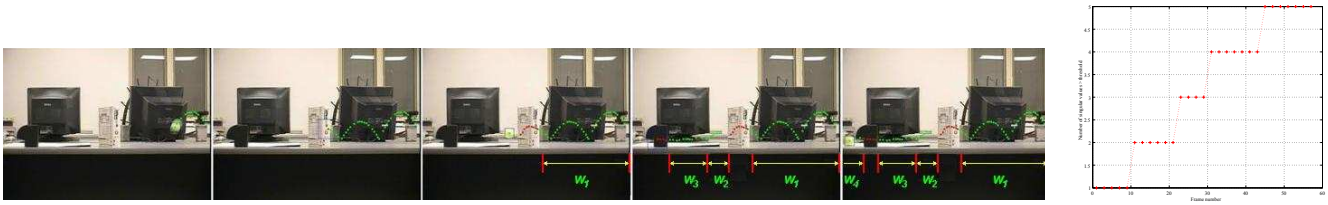
[1] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Comp. Vision*, 26(1):63–84, 1998.

[2] O. Camps, H. Li, M. C. Mazzaro, and M. Sznaier. A caratheodory-fejer approach to robust multiframe tracking, In *Proc. ICCV 2003*, pp. 1048–1055.



(a) (b) (c) (d)

Figure 12. NSV plots of 10 car tracks in example 4. (a) Car 1. (b) Car 5. (c) Car 8. (d) Other cars.



(a) (b)

Figure 13. Event detection: (a) Detecting occluded events. (b) **NSV** plot showing events.

[3] M. T. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, and G. Doretto. Joint recognition of complex events and track matching. In *Proc. CVPR 2006*, vol. 2, pp. 1615–1622.

[4] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proc. 2004 ACC*, pp. 3273–3278.

[5] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *Proc. 2003 ACC*, pp. 2156–2162.

[6] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proc. CVPR 1998*, vol. 1, pp. 22–29.

[7] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. PAMI*, 20(10):1025–1039, 1997.

[8] M. Irani and P. Anandan. Unified approach to moving object detection in 2d and 3d scenes. *IEEE Trans. PAMI*, 20(6):577–589, 1998.

[9] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int. J. of Comp. Vision*, 29(1):5–28, 1998.

[10] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. 1995 ACC*, vol. 1, pp. 1628–1632.

[11] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Proc. 2005 CVPR*, pp. 990–997.

[12] H. Lim, V. I. Morariu, O. I. Camps, and M. Sznaier. Dynamic appearance modeling for human tracking. In *Proc. CVPR 2006*, vol. 1, pp. 751–757.

[13] R. Lublinerman, M. Sznaier, and O. Camps. Dynamics based robust motion segmentation. In *Proc. CVPR 2006*, vol. 1, pp. 17–22.

[14] R. Mann, A.-D. Jepson, and T. El-Maraghi. Trajectory segmentation using dynamic programming. In *Proc. 2002 ICPR*, vol. 1, pp. 331–334.

[15] M. Moonen, B. D. Moor, L. Vandeberghe, and J. Vandewalle. On- and off-line identification of linear state-space models. *Int. J. of Control*, 49:219–232, 1989.

[16] B. North, A. Blake, M. Isard, and J. Rittscher. Leraning and classification of complex dnamics. *IEEE Trans. PAMI*, 22(9):1016–1034, 2000.

[17] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: A tutorial. *European J. of Control*, 13(2–3):242–260, 2007.

[18] R. S. Pena and M. Sznaier. *Robust Systems Theory and Applications*. Wiley & Sons, Inc., 1998.

[19] M. Piccardi and E. D. Cheng. Multi-frame moving object track matching based on an incremental major color spectrum histogram matching algorithm. In *Proc. CVPR 2005*, vol. 3, pp. 19–24.

[20] P.-L. Rosin and G. West. Nonparametric segmentation of curves into various representations. *IEEE Trans. PAMI*, 17(12):1140–1153, 1995.

[21] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR 1994*, vol. 1, pp. 593–600.

[22] N. K. Sinha and B. Juszta. *Modeling and Identification of Dynamic Systems*. Van Nostrand Reinhold Co., 1983.

[23] C. R. Wen, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Trans. PAMI*, 19(7):780–785, 1997.

[24] C. M. Woodside. Estimation of the order of linear systems. *Automatica*, 7(6):727–733, 1971.

[25] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In *Proc. ICCV 2003*, vol. 1, pp.1079–1085.

[26] Z.Yin and R.Collins. Moving object localization in thermal imagery by forward-backward mhi. In *IEEE Proc. OTCBVS'06*, p. 133, New York City, N.Y., June 2006.