

# Dynamic Subspace-Based Coordinated Multicamera Tracking

Mustafa Ayazoglu, Binlong Li, Caglayan Dicle, Mario Sznajder, and Octavia I. Camps \*  
Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115  
<http://robustsystems.ece.neu.edu>

## Abstract

*This paper considers the problem of sustained multicamera tracking in the presence of occlusion and changes in the target motion model. The key insight of the proposed method is the fact that, under mild conditions, the 2D trajectories of the target in the image planes of each of the cameras are constrained to evolve in the same subspace. This observation allows for identifying, at each time instant, a single (piecewise) linear model that explains all the available 2D measurements. In turn, this model can be used in the context of a modified particle filter to predict future target locations. In the case where the target is occluded to some of the cameras, the missing measurements can be estimated using the facts that they must lie both in the subspace spanned by previous measurements and satisfy epipolar constraints. Hence, by exploiting both dynamical and geometrical constraints the proposed method can robustly handle substantial occlusion, without the need for performing 3D reconstruction, calibrated cameras or constraints on sensor separation. The performance of the proposed tracker is illustrated with several challenging examples involving targets that substantially change appearance and motion models while occluded to some of the cameras.*

## 1. Introduction

Distributed surveillance systems use multiple cameras to cover wider areas and to provide different viewpoints of targets. Intuitively, the additional information provided by using multiple cameras with overlapping field of views can help a tracking system to overcome occlusion and clutter, specially when there are multiple similar targets in the scene. For example, if targets are visible by more than one camera at the same time, it is possible to disambiguate between them by using constraints imposed by the cameras' epipolar geometry. Furthermore, even if a target appears

largely occluded to some sensors, the system might recover by using the other sensors where the target is visible. Finally, examining data from spatially distributed sensors can reveal activity patterns not apparent to single or closely clustered sensors.

In order to fully take advantage of the information available from multiple sensors, multi-camera tracking systems must maintain consistent identity labels of the targets across the different views. However, this is, in general, a difficult problem because the appearance of the targets can be quite different when observed with cameras that have very different viewpoints or are widely separated (wide baseline stereo). Previous approaches to the “correspondence across views” problem include matching features such as color and apparent height [3, 7, 23, 9], using 3D information from camera calibration [7, 2, 1, 8, 10, 27], using the epipolar constraint to search for correspondences [22, 26, 11], modeling the relationship between the appearance of a target in different views through a linear time invariant system [21] or computing homographies between views [16, 17, 6, 4, 19, 15].

In this paper we propose a new approach that exploits geometric and dynamics constraints to improve robustness to occlusion, even in the presence of appearance or dynamics changes. The new method requires neither camera calibration, video rectification, nor 3D reconstruction and does not need to assume a dynamic motion model for the targets. Instead, it is based on the fact that, under mild conditions, the trajectories of a target in images captured from different viewpoints are constrained to evolve in a low dimensional subspace, directly determined from the tracking data.

The paper is organized as follows. Section 2 presents background material from realization theory and Hankel matrices which are the basis for the proposed approach. Section 3 describes a method using Hankel matrices to predict and propagate new measurements through a particle filter. Section 4 describes the proposed algorithm for multicamera tracking in the presence of occlusion. Section 5 presents experimental results comparing the performance of the proposed approach to previous methods. Finally, section 6 gives final remarks and discusses future work.

\*This work was supported in part by NSF grants IIS-0713003 and ECCS-0901433, AFOSR grant FA9550-09-1-0253, and the Alert DHS Center of Excellence under Award Number 2008-ST-061-ED0001.

## 2. Background

### 2.1. Autoregressive Models and the Hankel Matrix

Consider a vector dynamical process described by an  $n^{th}$  order autoregressive model of the form:

$$\mathbf{y}_k = a_1 \mathbf{y}_{k-1} + a_2 \mathbf{y}_{k-2} + \dots + a_n \mathbf{y}_{k-n} \quad (1)$$

where  $\mathbf{y}_k \in \mathbb{R}^d$ . To every trajectory  $\mathbf{y}_k$  of this model, one can associate a Hankel matrix defined as:

$$H_y^{s,r} \doteq \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \dots & \mathbf{y}_r \\ \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_{r+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_s & \mathbf{y}_{s+1} & \dots & \mathbf{y}_{r+s} \end{bmatrix} \quad (2)$$

It is a well known result from realization theory [12, 20], that, under mild conditions, given a sequence of measurements  $\{\mathbf{y}_k\}$  generated by (1), the order  $n$  of the model is related to the rank of the corresponding Hankel matrix by:

$$\text{rank}(H_y^{s,r}) = n$$

provided that  $r, s \geq n$ . Simple algebra using (1) shows that the vector  $[\mathbf{a}^T \ -1]^T$ , where  $\mathbf{a} = [a_n \ a_{n-1} \ \dots \ a_1]^T$  is the regressor vector, is orthogonal to the subspace spanned by the rows of the Hankel matrix, that is

$$H_y^{s,n} \begin{bmatrix} \mathbf{a} \\ -1 \end{bmatrix} = 0$$

Thus, the regressor vector  $\mathbf{a}$  can be estimated from the experimental data by least square error minimization:

$$\hat{\mathbf{a}} = (H_y^{s,n-1T} H_y^{s,n-1})^{-1} H_y^{s,n-1T} \begin{bmatrix} \mathbf{y}_n \\ \vdots \\ \mathbf{y}_{s+n} \end{bmatrix} \quad (3)$$

Note in passing that since the matrix  $H_y^{s,n-1}$  has rank  $n$ , then  $(H_y^{s,n-1T} H_y^{s,n-1})$  is invertible. Finally, given measurements  $\mathbf{y}_{s+1:s+n}$ , the expected value of the next measurement can be predicted from:

$$\hat{\mathbf{y}}_{s+n+1} = [\mathbf{y}_{s+1} \ \dots \ \mathbf{y}_{s+n}] \cdot \hat{\mathbf{a}} \quad (4)$$

### 3. Dynamic Subspace-based Tracking

The dynamics encapsulated in the temporal ordering of video data play an important role in computer vision. They have been used in several applications, including tracking, human recognition from gait, activity recognition, and dynamic texture. The main idea is to use the dynamics to model the temporal evolution of a measurement vector  $\mathbf{y}_k \in \mathbb{R}^m$  and predict its future values. For example, in a tracking application, the measurement vector  $\mathbf{y}_k$

can represent the coordinates of a target or the pixel values of its image at time  $k$ . Then, tracking can be posed as a Bayesian recursive filtering problem. Given measurements  $\mathbf{y}_{0:k-1} = \mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{k-1}$  predictions are made by

$$p(\mathbf{x}_k | \mathbf{y}_{0:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{0:k-1}) d\mathbf{x}_{k-1}$$

where  $\mathbf{x}_k \in \mathbb{R}^n$  is the state variable describing the target state at time  $k$ . The predictions are then corrected once a new measurement is available:

$$p(\mathbf{x}_k | \mathbf{y}_{0:k}) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{0:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{k-1})}$$

Traditionally, tracking systems assume a simple state transition dynamic model  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ , such as random walk, constant velocity or constant acceleration, and perform Bayesian inference through a Kalman [14] or a Particle filter [13]. Unfortunately, mismatches between the assumed and true dynamics can result in significant filter drifting causing tracking failure in the presence of prolonged occlusions. This problem can be solved by identifying the dynamics from experimental data [18, 5]. However, identification techniques assume that the dynamics are time invariant. In this paper we propose a simple method that continuously estimates the dynamics from the measured data, and hence can adapt to slowly time varying dynamics, which can be easily incorporated into a particle filter.

#### 3.1. Hankel-based Particle Filter

The Hankel-based measurement prediction mechanism described in section 2.1 can be used in conjunction with a particle filter to compute the drift of the particles as shown in Algorithm 1. The update of the particle weights is done based on the likelihood under a chosen observation model.

---

#### Algorithm 1: Hankel-Updated Particle Filter

---

The posterior  $p(\mathbf{y}_{k-1} | \mathbf{y}_{0:k-2})$  is represented by a set of sample particles  $\{s_{k-1}^{(n)}, n = 1, \dots, N\}$ , with weights  $\{\pi_{k-1}^{(n)}, n = 1, \dots, N\}$ . At time step  $k$ , construct a “new” sample set  $\{s_k^{(n)}, \pi_k^{(n)}, n = 1, \dots, N\}$  as follows:

1. Obtain next measurement estimate  $\hat{\mathbf{y}}_k$  using (4).
2. Compute the instantaneous drift

$$\mathbf{v}_k = \hat{\mathbf{y}}_k - \mathbf{y}_{k-1}$$

3. Drift the particles using the deterministic drift  $\mathbf{v}_k$

4. Re-sample the particles from  $\{s_{k-1}^{(n)}, n = 1, \dots, N\}$ , according to  $\{\pi_{k-1}^{(n)}, n = 1, \dots, N\}$ , and denote the newly selected particles  $\{s_k^{(n)}, n = 1, \dots, N\}$
5. Diffuse the samples
6. Recompute the weights for the particles according to a confidence measure based on the likelihood under the observation model and normalize the weights so they sum one. (For example, based on the error in matching against a target template at the particle location).
7. Output the particle with the highest confidence value as the output of the tracker and keep the other particles for the next iteration.

## 4. Multicamera Hankel Tracking

In this section we present a new algorithm for stereo tracking that exploits dynamic and geometric constraints to increase robustness to occlusion and clutter. While the algorithm is described for a two camera stereo setup, the approach is general and can be directly applied to more than two cameras.

The main ideas of the proposed algorithm are as follows:

- 1) As long as the target is visible from all viewpoints, track the target *simultaneously* in all cameras by performing a global optimization to predict its location on the 2D images through a dynamics based constraint (derived in section 4.1); and
- 2) If occlusion is detected, continue joint tracking in the views where the target is visible, and use dynamics and epipolar constraints to cope in the occluded views until the target is visible again.

### 4.1. The Trajectory Dynamic Subspace

In this section we present the key observation that motivates this paper: in the absence of noise, all the 2D affine projections of the 3D trajectory of a target, captured simultaneously by a set of affine cameras, lie on a single subspace orthogonal to a vector that can be estimated using data from a single camera.

**Theorem 1:** There exist a vector of the form  $[\mathbf{a}^T \ -1]^T \in \mathbb{R}^{n+1}$  that is orthogonal to the subspaces spanned by the rows of each of the Hankel matrices corresponding to the 2D affine projections of the 3D trajectory of a target.

**Proof:** Let the autoregressive model (1) describe the 3D motion of a target, where  $\mathbf{y}_k = \mathbf{P}_k$  represents the 3D homogeneous coordinates of a point  $\mathbf{P}$  belonging to the target, at time  $k$ , measured with respect to an arbitrary, fixed reference frame. Denote by  $H_y$  its associated block Hankel matrix and  $\mathbf{a}$  its auto-regressor vector:

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{k-n} & \dots & \mathbf{P}_{k-1} \end{bmatrix} \mathbf{a} \quad (5)$$

Now consider  $M$  affine cameras, with  $2 \times 4$  projection matrices  $\Pi_i$ ,  $i = 1, \dots, M$ . Assuming that the point  $\mathbf{P}$  is visible from camera  $i$  at time  $k$ , its 2D projections are given by:

$$\mathbf{p}_k^{(i)} = \Pi_i \mathbf{P}_k, \quad i = 1, \dots, M \quad (6)$$

Substituting (5) recursively in (6) we have:

$$\mathbf{p}_k^{(i)} = \Pi_i \begin{bmatrix} \mathbf{P}_{k-n} & \dots & \mathbf{P}_{k-1} \end{bmatrix} \mathbf{a} \quad (7)$$

$$= \begin{bmatrix} \mathbf{p}_{k-n}^{(i)} & \dots & \mathbf{p}_{k-1}^{(i)} \end{bmatrix} \mathbf{a} \quad (8)$$

or, equivalently:

$$H_{y^{(i)}}^{s,n} \begin{bmatrix} \mathbf{a} \\ -1 \end{bmatrix} = 0 \quad i = 1, \dots, M$$

where  $H_{y^{(i)}}^{s,n}$  denotes the Hankel matrix associated with the affine 2D projections of the 3D trajectory on the image plane of the  $i^{th}$  camera. *q.e.d.*

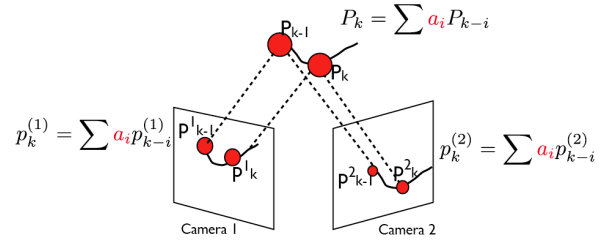


Figure 1. Multicamera Regressors: The 3D trajectory of each target is explained with an autoregression model. The corresponding 2D trajectories are also explained by autoregression models using the *same* regressor vector as their corresponding 3D trajectory.

**Remark 1** From the result above it follows that the projected trajectories are described by the same regressor  $\mathbf{a}$  which can be estimated from the Hankel matrices of the cameras that see  $\mathbf{P}$  as illustrated in Figure 1.

### 4.2. Dynamics-based Multicamera Measurement Predictions

Consider the last  $N$  measurements of the target location<sup>1</sup>  $\mathbf{y}_j^{(i)} \in \mathbb{R}^d$ ,  $j = k - N + 1, \dots, k$ , in views  $i = 1, \dots, M$ . From Theorem 1, these trajectories lie on the subspace orthogonal to  $[\mathbf{a}^T \ -1]^T$  such that

$$H_{y^{(i)}}^{s,n} \begin{bmatrix} \mathbf{a} \\ -1 \end{bmatrix} = 0 \quad i = 1, \dots, M$$

<sup>1</sup>In our implementation we use the centroid of the target found through normalized correlation with an adaptive template. However, it is possible to apply the algorithm to local features such as corners or SIFT features.

or equivalently

$$H_{y^{(*)}}^{s,n} \begin{bmatrix} \mathbf{a} \\ -1 \end{bmatrix} = 0 \quad (9)$$

where  $\mathbf{a}$  is the  $n^{th}$  order regressor common to all the trajectories, and where  $\mathbf{y}_k^{(*)} \in \mathbf{R}^{Md}$  is formed by stacking the measurements of the target location at time  $k$  in images taken from  $M$  different viewpoints on top each other:

$$\mathbf{y}_k^{(*)} = \begin{bmatrix} \mathbf{y}_k^{(1)} \\ \mathbf{y}_k^{(2)} \\ \vdots \\ \mathbf{y}_k^{(M)} \end{bmatrix},$$

Here the dimensions  $n$  and  $s = N - n$  are chosen so that the rank of the matrix  $H_{y^{(*)}}^{s,n}$  is  $n$  and all the available measurements are used. Thus, we can use the dynamics based constraint (9) to *simultaneously* predict the measurements at time  $k + 1$  in *all*  $M$  views,  $\hat{\mathbf{y}}_{k+1}$  by finding the common regressor vector  $\mathbf{a}$  as described in Algorithm 2. The predictions  $\hat{\mathbf{y}}_{k+1}$  can then be used in step 2 in Algorithm 1.

#### Algorithm 2:

##### Hankel-based Multicamera Measurement Prediction

Given the last  $N$  measurements in views  $i = 1, \dots, M$ ,  $\mathbf{y}_{k-N+1:k}^{(i)} \in \mathbf{R}^d$ , predict the next measurements  $\mathbf{y}_{k+1}^{(i)} \in \mathbf{R}^d$  in all views as follows:

1. Form  $N$  stacked vectors  $\mathbf{y}_{k-N+1:k}^{(*)} \in \mathbf{R}^{Md}$ .
2. Form a Hankel matrix for vectors  $\mathbf{y}_{k-N+1:k}^{(*)}$  as square as possible,  $H_{y^{(*)}}$ .
3. Estimate the rank of  $H_{y^{(*)}}$ ,  $n$ :
  - (a) Compute the SVD  $H_{y^{(*)}} = UDV^T$
  - (b)  $\Lambda = \text{diag}(D)$
  - (c)  $Thrs = \text{sum}(\Lambda)$
  - (d)  $n = \text{sum}(\text{cumulative sum}(\Lambda) > \gamma Thrs)$
4. Reshape  $H_{y^{(*)}}$  so it has  $n$  columns
5. Estimate  $\mathbf{y}_{k+1}^{(*)}$ , using (4)

where  $\gamma^2$  is an upper bound on the (relative) variance of the measurement noise.

### 4.3. Occlusion Handling

When the target is occluded in a view, it is possible to use a combination of dynamic and epipolar geometry constraints to estimate the current location of the target in the occluded view and use this estimate to predict the location of the target in the next frame in all views, as described next.

Without loss of generality, assume a stereo system with two cameras and that the target is occluded in camera 2 at time  $k$ . Further, assume that there are available  $N + 1$  measurements from camera 1,  $\mathbf{y}_{k-N+1:k}^{(1)}$  and  $N > n$  measurements from camera 2,  $\mathbf{y}_{k-N+1:k-1}^{(2)}$ . From Theorem 1, we have

$$\begin{bmatrix} \mathbf{y}_k^{(1)} \\ \mathbf{y}_k^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_{k-n}^{(1)} & \mathbf{y}_{k-n-1}^{(1)} & \cdots & \mathbf{y}_{k-1}^{(1)} \\ \mathbf{y}_{k-n}^{(2)} & \mathbf{y}_{k-n-1}^{(2)} & \cdots & \mathbf{y}_{k-1}^{(2)} \end{bmatrix} \mathbf{a}$$

and since corresponding points are constrained to be in the associated epipolar lines we have

$$\begin{bmatrix} \mathbf{y}_k^{(1)T} & 1 \end{bmatrix} F \begin{bmatrix} \mathbf{y}_k^{(2)} \\ 1 \end{bmatrix} = 0$$

where  $F$  is the fundamental matrix. Combining these two constraints and using all the available measurements we have

$$A\mathbf{x} = \mathbf{b}$$

where

$$A = \begin{bmatrix} H_{\mathbf{y}^{(1)}}^{s,n} & 0 & 0 \\ H_{\mathbf{y}^{(2)}}^{t,n} & 0 & 0 \\ \hline Y_{k-1-n:k-1}^{(2)} & -I_{2 \times 2} \\ \hline 0_{1 \times n} & \ell_1 & \ell_2 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \text{vect}\left(Y_{k-n+1:k}^{(1)}\right) \\ \text{vect}\left(Y_{k-n:k-1}^{(2)}\right) \\ \hline 0_{2 \times 1} \\ \hline -\ell_3 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{y}_k^{(2)} \end{bmatrix}; \quad Y_{j:k}^{(i)} = \begin{bmatrix} \mathbf{y}_j^{(i)T} & \mathbf{y}_{j+1}^{(i)T} & \cdots & \mathbf{y}_k^{(i)T} \end{bmatrix}^T$$

$$\begin{bmatrix} \ell_1 & \ell_2 & \ell_3 \end{bmatrix} = \begin{bmatrix} \mathbf{y}_k^{(1)T} & 1 \end{bmatrix} F$$

and where, for a matrix  $M$ ,  $\text{vect}(M)$  denotes the vector obtained by stacking the columns of  $M$  on top of each other. Thus, a least squares estimate of  $\mathbf{y}_k^{(2)}$  is given by:

$$\begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{y}}_k^{(2)} \end{bmatrix} = (A^T A)^{-1} A^T \mathbf{b}$$

where  $A^T A$  is invertible, since by construction  $A$  has full column rank. Finally, the measure  $\mathbf{y}_k^{(1)}$  and the estimate  $\mathbf{y}_k^{(2)}$  can be used to predict the measurements at time  $k + 1$  in both views using Algorithm 2.

## 5. Implementation Details and Evaluation

To test the performance of the proposed approach we implemented using MATLAB two wide-baseline stereo particle filter tracker systems, “StereoHankel” and “3D-Assisted”. Both trackers estimate the Fundamental matrix from 8 point correspondences (selected by hand), perform background subtraction using a mixture of Gaussians model [25] as a preprocessing step and use the adaptive template matching particle filter tracker IVT [24] to locate the target in the 2D images. The “StereoHankel” tracker works directly with the 2D images and does not recover 3D information. In addition, it uses Algorithms 1 and 2 to simultaneously predict the location of the target in both views. Occlusion is detected when the normalized cross correlation with the current template is below a threshold. Finally, it also uses the occlusion handling technique described in the previous section on the centroid of the template. The tracker “3D-Assisted”, on the other hand, uses an (uncalibrated) 3D reconstruction of the topmost point of the target obtained from the 2D target tracked positions and the Fundamental matrix. This tracker predicts the next 2D measurements and handles occlusion by using a 3D constant velocity model in conjunction with a Kalman filter to first predict the next 3D position of the target and then back-projecting it to both 2D views. Thus, the main difference between the trackers is on how they handle occlusion: 3D-Assisted does it through geometry, while StereoHankel does it through dynamics.

### 5.1. Experiments

We chose to use videos from the PETS 2001, 2003 and 2006 databases to compare the performance of the two trackers. These datasets are very challenging in terms of significant lighting variations, long lasting occlusions, scene activity, variation of view-points and target motions. In all cases we set  $\gamma = 0.95$  in Algorithm 2, corresponding to a 5% estimated noise level. In Figures 2 to 5 the blue tracks show when the tracker is in charge, and the green tracks show when the predictor is in charge.

In the first two examples (PETS2001, Scenarios 1 and 2) the target is a car that changes dynamics while occluded in one of the views. In scenario 1, a car enters the occlusion at full speed, brakes while occluded and stops shortly after the occlusion. In scenario 2, a car moves towards a long occlusion and turns while occluded. Figures 2 and 3 show sample frames of the tracking output for both trackers. In both examples, 3D-Assisted fails to recover after the occlusion, while StereoHankel successfully tracks the target through the occlusions and correctly interpolates the occluded trajectory.

The third example (PETS2003, Scenario 3) is a video of a soccer match where the target (a player) is very small, and the separation between the cameras is almost as large as the soccer field (110 meters wide). Additional challenges in

Table 1. Frame rate for HankelStereo and 3D-Assisted Trackers.

Scenario	HankelStereo	HankelStereo	3D-Assisted
	(Occ. Handling)	(No Occ. Handling)	
Sc. 1	2.1 fps	2.8 fps	2.4 fps
Sc. 2	1.9 fps	2.8 fps	2.4 fps
Sc. 3	1.5 fps	1.9 fps	1.5 fps
Sc. 4	2.6 fps	2.9 fps	3.0 fps

this video are due to the fact that the target leaves the field of view and enters it back, and the erratic motion of the target (stops suddenly, changes directions, runs backwards). Figure 4 shows sample frames of the tracking output for both trackers. While the 3D-Assisted tracker does not miss the target and is able to recover after the target comes back into the field of view of camera 1, its accuracy is poor. On the other hand, the proposed StereoHankel keeps a close track on the target through the occlusion and after it comes back, in spite of the changes in dynamics and the temporary disappearance of the target.

The last example (PETS2006, Scenario 4) presents several challenges: the viewpoints are very different (180 degrees from each other), the occlusion is due to another moving target with similar appearance, and the scene is quite cluttered. Figure 5 shows sample frames of the tracking output for both trackers. Once again, the 3D-Assisted fails to track while StereoHankel successfully tracks the target through the occlusion in spite of the changes in dynamics and the disappearance of the target.

### 5.2. Computational Time

Table 1 compares the frame rate for the HankelStereo tracker with and without occlusion handling and the 3D-Assisted tracker, implemented in MATLAB 2010a running on a Intel Xeon CPU, 2.53GHz (2 processors). The table shows that the computational times for the HankelStereo and the 3D-Assisted tracker are very similar and that our occlusion handling mechanism does not represent a significant computational burden.

## 6. Conclusion

In this paper we proposed a new tracking algorithm for multicamera systems capable of exploiting both geometric and dynamical constraints to improve robustness to occlusion, even in the presence of appearance or dynamics changes. This new method requires neither camera calibration, video rectification, nor 3D reconstruction and does not need to assume a dynamic motion model for the targets. Instead, it is based on the fact that, under mild conditions, the trajectories of a target in images captured from different viewpoints are constrained to evolve in a common



low dimensional subspace. This fact allows for identifying, at each step, a single model that explains all the available 2D measurements and can be used in the context of a modified particle filter to predict future target locations. In cases where the target is occluded to some of the cameras, the missing measurements can be estimated using the facts that they must lie both in the subspace spanned by previous measurements and satisfy epipolar constraints. Combining these constraints allows for jointly finding the model and estimating the occluded data by simply solving a least-squares problem. The ability of the proposed tracker to robustly handle occlusion was illustrated with several challenging examples involving targets that substantially change appearance and motion models while occluded to some of the cameras. Research is currently underway seeking to extend these results to the case of perspective projections, with the main difficulty here being the non-linear nature of the map projecting the 3D coordinates to the 2D image planes.

## References

- [1] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3), 2003. 1
- [2] M. Black and T. Ellis. Multiple camera image tracking. In *Performance Evaluation of Tracking and Surveillance Conference (PETS)*, 2001. 1
- [3] Q. Cai and J. K. Aggarwal. Tracking human motion in structured environments using a distributed camera system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):1241–1247, 2000. 1
- [4] S. Calderara, R. Cucchiara, and A. Prati. Bayesian-competitive consistent labeling for people surveillance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):354–360, 2008. 1
- [5] O. I. Camps, H. Lim, C. Mazzaro, and M. Sznaiar. A caratheodory-fejer approach to robust multiframe tracking. In *International Conference on Computer Vision*, pages 1048–1055, 2003. 2
- [6] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *IEEE Computer Vision and Pattern Recognition*, 2000. 1
- [7] T. H. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *International Conference on Computer Vision*, 2001. 1
- [8] R. Collins, O. Amidi, and T. Kanade. An active camera system for acquiring multi-view video. In *Int. Conf. on Image Processing*, volume I, pages 517–520, 2002. 1
- [9] D. Comaniciu, F. Berton, and V. Ramesh. Adaptive resolution system for distributed surveillance. *Real Time Imaging*, 8:427–437, 2002. 1
- [10] S. L. Dockstader and A. M. Tekalp. Multiple camera tracking of interacting and occluded human motion. In *Proceedings of the IEEE*, volume 89, pages 1441–1455, October 2001. 1
- [11] A. Gaschler, D. Burshchka, and G. Hager. Epipolar-based stereo tracking without explicit 3d reconstruction. In *ICPR*, 2010. 1
- [12] B. Ho and R. Kalman. Effective construction of linear, state-variable models from input/output functions. *Regelungstechnik*, 14:545–548, 1966. 2
- [13] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 2
- [14] 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. 2
- [15] S. M. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(3):505–519, 2009. 1
- [16] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common frame. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):758–767, 2000. 1
- [17] L. Lee and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767, August 2000. 1
- [18] H. Lim, O. I. Camps, M. Sznaiar, and V. Morariu. Dynamic appearance modelling for human tracking. In *IEEE Computer Vision and Pattern Recognition*, pages 751–757, 2006. 2
- [19] S. Lim and L. Davis. An ease-of-use stereo-based particle filter for tracking under occlusion. In A. E. et al, editor, *Lecture Notes in Computer Science*, volume 4814, pages 225–239. Springer Verlag, 2007. 1
- [20] M. Moonen, B. D. Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state space models. *International Journal of Control*, 49:219–232, 1989. 2
- [21] V. Morariu and O. I. Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *IEEE Computer Vision and Pattern Recognition*, pages 537–544, 2006. 1
- [22] K. Ni and F. Dellaert. Stereo tracking and three-point/onepoint algorithms—a robust approach in visual odometry. In *ICPR*, pages 2777–2780, 2006. 1
- [23] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. V. Gool. Color-based object tracking in multi-camera environments. In *DAGM*, Springer LNCS 2781, pages 591–599, 2003. 1
- [24] D. Ross, J. Lim, R.-S. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1–3):125–141, 2007. 5
- [25] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. 5
- [26] D. Stoyanov, G. Mylonas, F. Deligianni, A. Darzi, and G. Yang. Soft-tissue motion tracking and structure estimation for robotic assisted mis procedures. In *Lecture Notes in Computer Science*, volume 3750, pages 139–146. Springer Verlag, 2005. 1
- [27] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke. Tracking a large number of objects from multiple views. In *ICCV*, 2009. 1

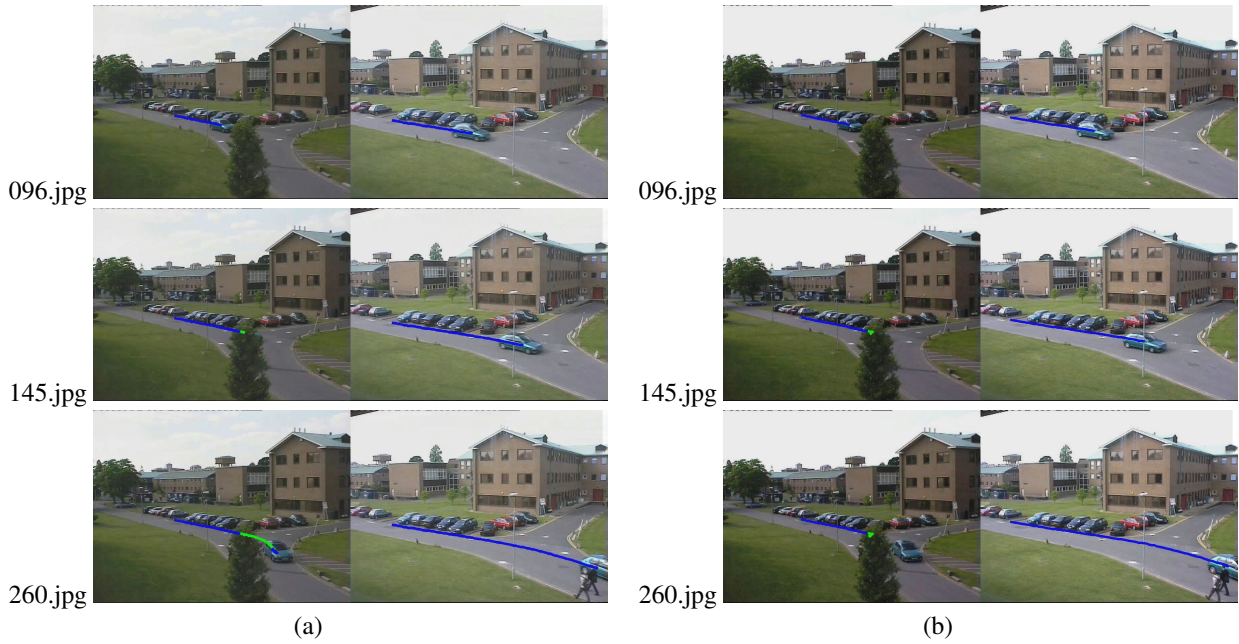


Figure 2. PETS2001, Scenario 1: Braking car. The target slows down while occluded in camera 1. (a) StereoHankel successfully handles the occlusion and interpolates the occluding trajectory. (b) 3D-Assisted fails to track due to the occlusion.

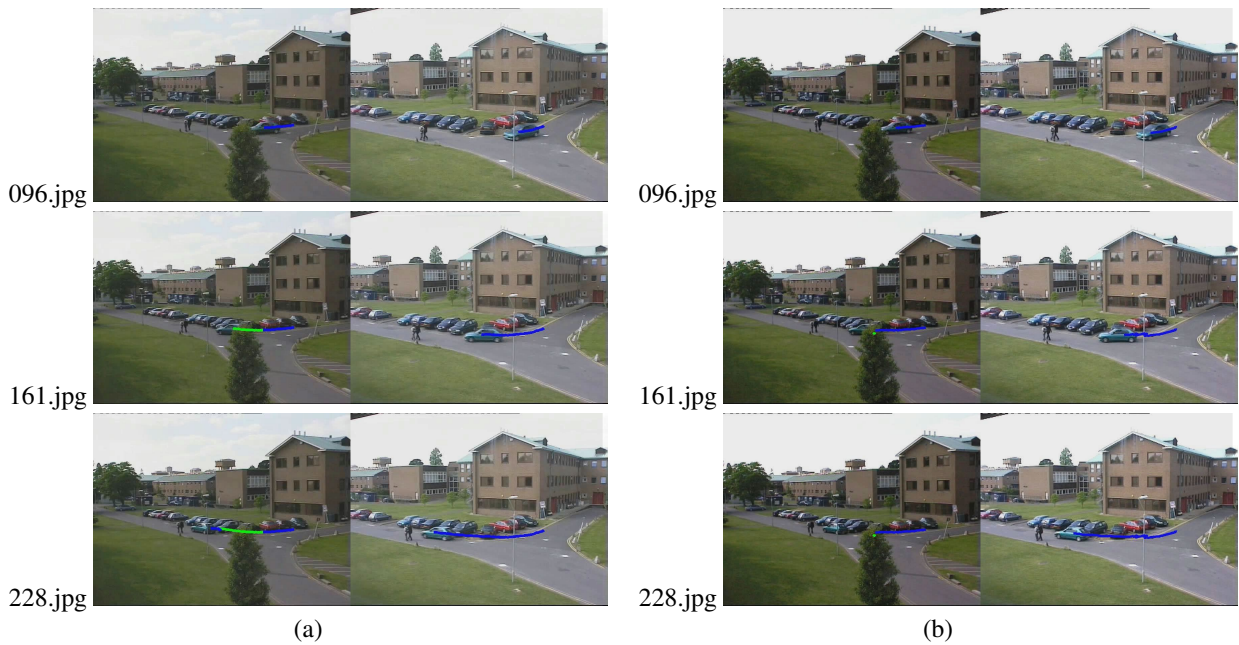


Figure 3. PETS2001, Scenario 2: Turning car. The target turns direction while occluded in camera 1, changing significantly both dynamics and appearance. (a) StereoHankel successfully handles the occlusion and interpolates the occluding trajectory. (b) 3D-Assisted fails to track due to the occlusion.



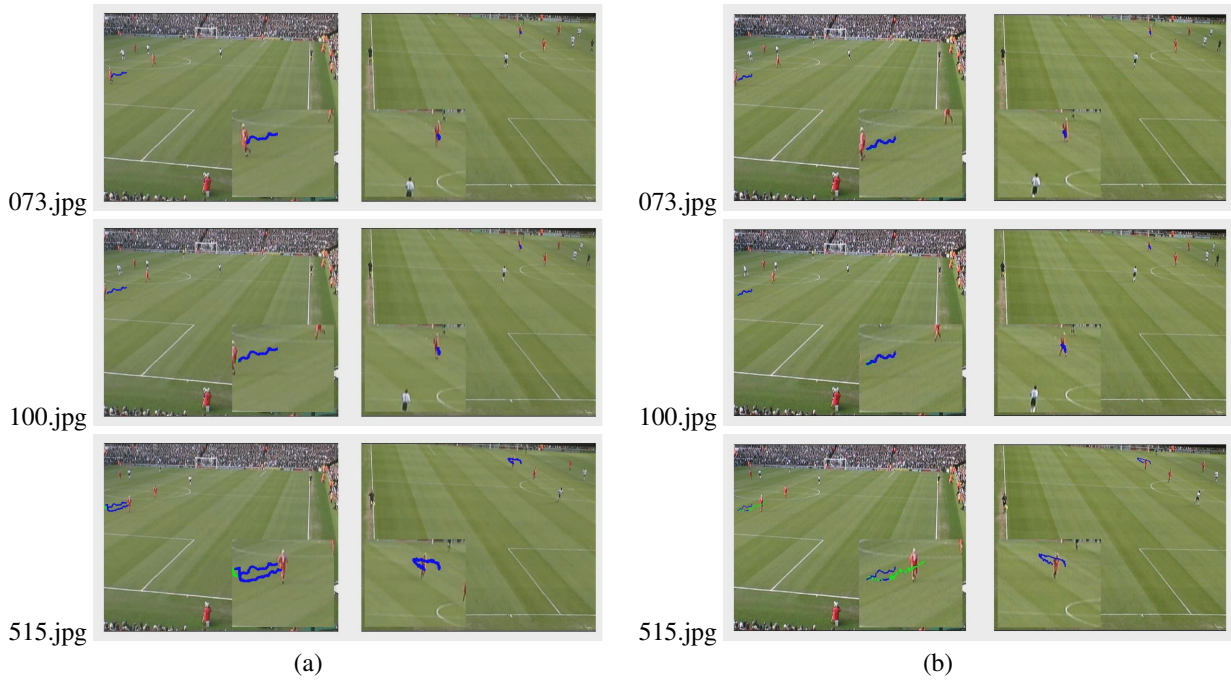


Figure 4. PETS2003, Scenario 3: Soccer player. The target is very small (please look at zoomed in sub-windows), leaves the field of view in camera 1 and changes his motion several times. (a) StereoHankel successfully handles the dynamic changes and the target disappearance. (b) 3D-Assisted tracks poorly as the target is partially occluded and does not recover after the target comes back.

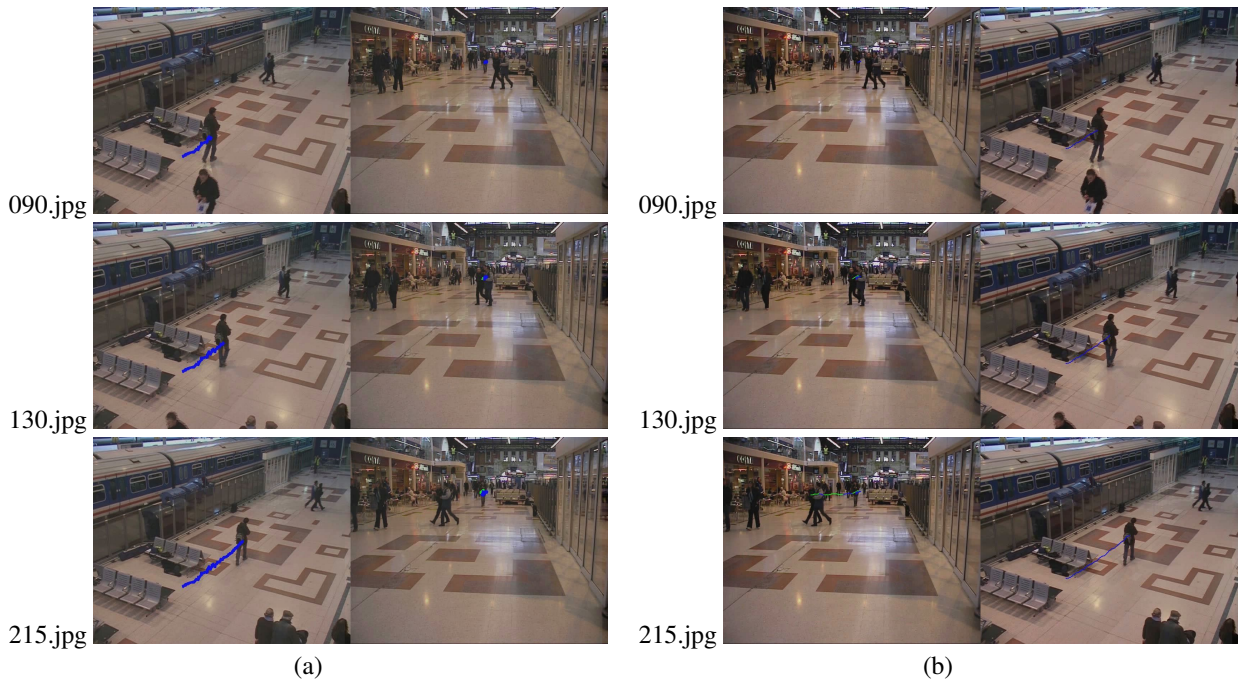


Figure 5. PETS2004, Scenario 4: Transport terminal. The viewpoints of the two cameras are very different and the occlusion is due to another moving target with similar appearance. (a) StereoHankel successfully handles the occlusion. (b) 3D-Assisted gets distracted by the occluder.