Multi-target Data Association by Tracklets with Unsupervised Parameter Estimation

Weina Ge and Robert T. Collins
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
{ge, rcollins}@cse.psu.edu

Abstract

We consider multi-target tracking via probabilistic data association among tracklets (trajectory fragments), a mid-level representation that provides good spatio-temporal context for efficient tracking. Model parameter estimation and the search for the best association among tracklets are unified naturally within a Markov Chain Monte Carlo sampling procedure. The proposed approach is able to infer the optimal model parameters for different tracking scenarios in an unsupervised manner.

1 Introduction

Long studied in the radar and remote sensing world, multi-target tracking has been drawing increasing attention in the visual tracking community due to the prevalence of video cameras mounted in public places. The enormous quantity of video demands intelligent algorithms that can adapt to different input sequences. Most existing tracking algorithms require parameter tuning for different scenes. Although methods for automatic parameter estimation exist, they typically require labeled training sequences.

Another challenge in multi-target tracking is the presence of an unknown and ever-changing number of targets. We adopt Markov Chain Monte Carlo Data Association (MCMCDA) to estimate a varying number of trajectories given a set of tracklets extracted from the video sequence. Tracklets are mid-level features that provide more spatial and temporal context than raw sensor detections, while being less demanding to produce than persistent object trajectories. Each tracklet is a partial trajectory extracted by a tracker within a short time period and therefore less prone to drift and occlusion than a long trajectory. The final output of our data association algorithm is a partition of the set of tracklets such that the tracklets belonging to each individual object have been grouped together (see Figure 1).

To summarize, we propose to recover the trajectories of moving foreground objects from a set of short-term tracklets using MCMCDA and to automatically infer the optimal model parameters from unlabeled data. We show that by adopting the Bayesian paradigm, inference of both the optimal parameters and the tracklet partition can be naturally unified. Experimental results also demonstrate the advantage of working at the level of tracklets when objects are closely spaced or occlude each other.
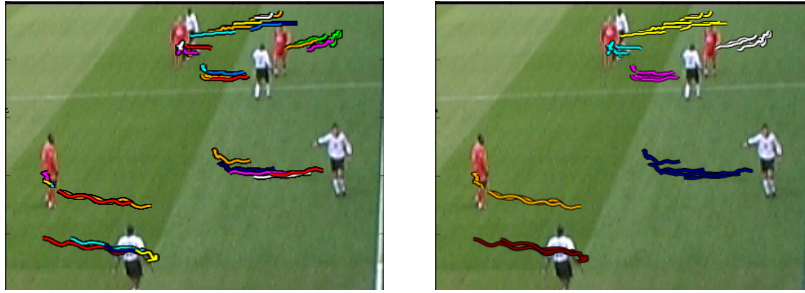
Figure 1: Illustration of multi-target data association by tracklets. Left: unordered collection of raw tracklets extracted from overlapping temporal windows. Right: partition of tracklets into sets associated with individual objects, each drawn in a different color.

1.1 Related work

Multi-target data association is traditionally addressed using the classic multiple hypothesis tracker (MHT) [7] or joint probabilistic data association filter (JPDAF) [1]. MHT maintains, at least in principle, a complete hypothesis tree of feasible data association assignments between object tracks and incoming observations. The full method is computationally infeasible unless combined with (suboptimal) pruning heuristics. JPDAF is a sequential method that updates each known trajectory by a weighted sum of compatible observations in each new frame [9]. As the number of observation grows, the complexity of both methods becomes unmanageable in practice. Recursive Bayesian methods such as the mean-shift or particle filter trackers make a first-order Markovian assumption that the current state of a target only depends on the previous time frame. Many of these basic methods also assume they are tracking a single object in isolation, and the obvious extension to tracking multiple single objects separately runs into problems when the objects are closely spaced or interacting. To model the interaction among targets, various graphical models have been developed. Yang et al. combine individual mean shift trackers in a star-graph and use belief propagation to infer the optimal joint probability [10]. A pairwise Markov random field has been adopted to prevent nearby trackers from claiming the same set of image pixels in [8, 12]. Methods based on MCMC sampling have a computational advantage and can be extended to handle a varying number of targets [8, 12, 13]. However, being limited by the the underlying Markovian assumption, it is hard to achieve optimal tracking results in the long run. To relax this assumption, researchers have been working on approaches that use longer-range temporal information [3].

More recently, MCMCDA has been adopted for multi-target tracking [5, 6, 11]. This approach has the advantage of searching for the globally optimal solution while still being computationally manageable, and provides a principled way to incorporate prior knowledge. Automatic parameter estimation is proposed in [5] as a linear programming problem, but labeled sequences are required. Our paper is inspired by these recent advances in MCMCDA, however we propose a purely Bayesian approach that infers the model parameters from unlabeled data by sampling, while simultaneously estimating the optimal solution for the data association.

2 Multi-target tracking as data association

Denote the set of observations within the time interval $[1, T]$ as $Z = \{Z_1, Z_{1+\Delta t}, \ldots, Z_T\}$ where Z_t is the set of observations from an unknown number of targets obtained at time frame t and Δt controls the sampling rate. The data association view of multi-target tracking aims at finding the optimal partition of the observation set $\omega^* = \{\tau_0, \tau_1, \ldots, \tau_K\}$, such that τ_0 is the set of false alarms, τ_k is the trajectory of target k, and K is the estimated number of targets that appear within the entire time interval. We constrain each observation to be associated with at most one trajectory, and constrain a valid trajectory to have at least two observations to distinguish between a single observation and a false alarm.

In the Bayesian framework, we take the the maximum a posterior (MAP) estimator of the posterior distribution as the optimal solution for the partition ω, i.e.

$$\omega^* = \arg\max_{\omega}(p(\omega|Z)) \stackrel{\text{Bayes' rule}}{\longleftrightarrow} \arg\max_{\omega}(p(Z|\omega)p(\omega)) \tag{1}$$

where $p(Z|\omega)$ is the likelihood function that models how well the partition fits the observations and $p(\omega)$ expresses our prior knowledge about desirable properties of good trajectories. This prior is often parameterized as $p(\omega|\lambda)$, with $\lambda = \{\lambda_i\}$ being a vector of model parameters. The values of λ are crucial to the algorithm's performance. In [11], these parameters are estimated from labeled video sequences by solving a linear system of equations. However, as the authors pointed out, the ground truth data often generate contradictory equations and thus heuristics have to be used to form a solvable system. One of our main contributions is to show how to infer λ from unlabeled data in a principled way by a Bayesian hierarchical model with hyperprior $p(\lambda|\theta)$. The hyperparameters θ are set to yield non-informative priors because we want our method to adapt to different tracking scenarios with the minimal amount of human supervision. However, it is easy to modify the priors to incorporate domain knowledge. Instead of manually setting the model parameters λ, we treat them as unknowns as well, and infer both λ and ω as

$$(\omega^*, \lambda^*) = \arg\max_{\omega, \lambda}(p(\omega, \lambda|Z)) \Longleftrightarrow \arg\max_{\omega, \lambda}(p(Z|\omega, \lambda)p(\omega|\lambda)p(\lambda|\theta)) \tag{2}$$

To find the solution of Eqn. 2 is extremely challenging due to the combinatorial solution space of ω. MCMC sampling has been shown to be a powerful computing approach for solving such complicated problems [5, 6, 11]. We show how to extend MCMCDA with inference of the model parameters λ in Section 3. The rest of this section will first explain how we extract the features from the video sequence and use them in our models.

2.1 Feature Extraction and Modeling

We define a set of basic features similar to [11], adapted for use with tracklet observations as input. We use simple, single-target trackers such as mean-shift or particle filtering to generate tracklets. Indeed, the strength of our approach is that it does not depend on how the initial tracklets are produced, since we automatically estimate our parameters from the data itself. Tracklets are initialized by a foreground object detector that runs on every 10th frame. The object detector fits a rectangular cover to the foreground map generated by a background subtraction algorithm, in a manner similar to [13]. For each detection, a tracker is initialized to track the rectangular region for up to d subsequent frames (e.g.

$d = 30$). Each tracklet is thus a sequence of rectangles that delineates the location and size of a candidate object, and tracklets for the same object overlap temporally to be resilient to missed detections in some frames.

Let the tracklet for the jth detected object rectangle that is initialized at frame t be $Z_t^j = \{(X_{t_i}^j, S_{t_i}^j, V_{t_i}^j) : i \in [0, d]\}$, where $X = (x, y)$ is the coordinate of the object center, $S = (w, h)$ is the width and height of the object, and $V = (dx, dy)$ is the velocity vector normalized w.r.t. object size. To recover the true trajectories of foreground objects is equivalent to finding a subset of tracklets that belong to each foreground object and stitching them together in an optimal way. The estimated trajectory for each object, τ_k, is represented as $\{\tau_{k_1}, \tau_{k_2}, \dots, \tau_{k_{|\tau_k|}}\}$, where τ_{k_i} denotes the ith tracklet in the trajectory.

We extract four tracklet-level measures to model the likelihood of belonging to the same trajectory based on spatial, motion, and appearance consistencies. In this scheme, a distance function D_j defines the similarity of two rectangles at one time instance based on feature j. This rectangle-level measure is aggregated into a tracklet-level distance measure $f_j(Z_1, Z_2)$ as follows: if two tracklets Z_1 and Z_2 overlap temporally, the distance measures between rectangles in the overlapping frames are averaged; otherwise, we compute the distance between the ending rectangle of Z_1 and the starting rectangle of Z_2 to allow missing detections and gaps between tracklets. This tracklet-level distance f_j is further aggregated to a trajectory-level distance M_{jk} based on the pairwise distances between pairs of successive tracklets in the trajectory. This layered aggregation scheme provides more accurate and stable measures in a trajectory context than purely frame-wise measures. We use a general exponential model to define the likelihood function for a single trajectory τ_k given the observed tracklet features

$$\ell_j(\tau_k) = \prod_{i=1}^{|\tau_k|-1} \ell_i(\tau_{k_{i+1}} | \tau_{k_i}) = \prod_{i=1}^{|\tau_k|-1} \lambda_j e^{-\lambda_j f_j(\tau_{k_{i+1}}, \tau_{k_i})} = \lambda_j^{|\tau_k|-1} e^{-\lambda_j M_{j_k}} \quad (3)$$

$$\text{where } M_{j_k} = \sum_{i=1}^{|\tau_k|-1} f_j(\tau_{k_{i+1}}, \tau_{k_i}) \quad (4)$$

We now define the D_j distance functions for each feature M_j. M_1: *Color Appearance.* We measure appearance similarity between two tracklets by Earth Mover's Distance (EMD) [4]. D_1 is the EMD distance between color histograms extracted from two rectangular regions. M_2: *Object Size.* Rectangles with quite different sizes are unlikely to come from the same object. Hence, we define $D_2 = ||S_1 - S_2|| / \max(w_1, w_2)$ as the normalized difference between object sizes. M_3: *Spatial Proximity.* The spatial proximity among tracklets within the same trajectory is measured by Euclidean difference of object location of the two tracklets normalized w.r.t. the object size, i.e., $D_3 = ||X_1 - X_2|| / \max(w_1, w_2)$. M_4: *Velocity Coherence.* The velocity distance is measured by $D_4 = ||V_1 - V_2||$, as we do not want to merge two tracklets into one trajectory if they are going in two different directions, even if they are spatially close to each other and have similar appearance.

Based on Eqn. 3, we are set to define the likelihood function of K estimated trajectories given observations from the entire sequence Z as

$$p(Z|\omega, \lambda) = \prod_{k=1}^{K} \ell(\tau_k) = \prod_{k=1}^{K} \prod_{j=1}^{4} \ell_j(\tau_k) = \prod_{j=1}^{4} \lambda_j^{\sum_k |\tau_k|-K} e^{-\lambda_j M_j}, \ M_j = \sum_k M_{j_k} \quad (5)$$

We also incorporate prior knowledge about desirable properties of trajectories by computing the following features.

M_5: *False Alarms.* To avoid a trivial configuration of ω where all the tracklets are considered as false alarms, we define the penalty function:

$$p_f(\omega) = \lambda_5 e^{-\lambda_5 M_5}, \quad \text{where } M_5 = |\tau_0| \tag{6}$$

M_6: *Trajectory Length.* Let $F(\tau_k)$ be the set of frames covered by a trajectory τ_k and let $DF(\tau_k) = \max(F(\tau_k)) - \min(F(\tau_k))$. We encourage long trajectories by the following exponential model

$$p_l(\omega) = \prod_{k=1}^{K} p_l(\tau_k) = \prod_{k=1}^{K} \lambda_6 e^{-\frac{\lambda_6}{DF(\tau_k)}} = \lambda_6^K e^{-\lambda_6 M_6}, \; M_6 = \sum_k DF(\tau_k)^{-1} \tag{7}$$

M_7, M_8: *Merge Pairs and Spatial Overlap.* In practice, we extract tracklets from temporally overlapping windows, and therefore each trajectory is expected to be fragmented into multiple overlapping tracklets. Candidate merge pairs are two tracklets with a particular parent/child structure, to be described in the next section. If eventually they are not merged, we call them *dangling merge pairs*. To encourage merging overlapping tracklets rather than starting new trajectories, we penalize large numbers of dangling merge pairs as well as spatial overlap between different trajectories.

$$p_g(\omega) = \lambda_7 e^{-\lambda_7 M_7}, \; M_7 = |G|, \; G \text{ is the set of dangling merge pairs} \tag{8}$$

$$p_o(\omega) = \prod_{k=1}^{K} p_o(\tau_k) = \prod_{k=1}^{K} \lambda_8 e^{-\lambda_8 M_{8_k}} = \lambda_8^K e^{-\lambda_8 M_8}, \; M_8 = \sum_k M_{8_k} \tag{9}$$

where M_{8_k} is the amount of spatial overlap between object rectangles, aggregated into a track-level measure in a similar manner as discussed for Eqn.3. With Eqns. 6-9, the prior probability is defined as follows:

$$p(\omega|\lambda) = p_f(\omega) p_l(\omega) p_g(\omega) p_o(\omega) = \lambda_5 \lambda_6^K \lambda_7 \lambda_8^K e^{-\Sigma_{j=5}^{8} \lambda_j M_j} \tag{10}$$

The last piece of the model is the hyperprior for the model parameters λ, since we treat them as parameters to be estimated by the algorithm along with ω. Because we model both the likelihood function and the prior distribution by exponential distributions, for computational efficiency we choose the Gamma distribution for all our hyperpriors.

$$p(\lambda|\theta) \sim \texttt{Gamma}(\theta^0, \theta^1) \propto \lambda^{\theta^0 - 1} e^{-\frac{1}{\theta^1}\lambda} \tag{11}$$

where $\theta^0 = \{\theta_i^0 : i = 1, \ldots, 8\}$ is the vector of shape parameters for the Gamma distribution and θ^1 is the vector of scale parameters. The hyperparameters θ that govern the Gamma distributions are chosen to yield non-informative priors, assuming we do not have prior knowledge about the tracking scenario.

3   MCMCDA with unsupervised parameter estimation

Because of the combinatorial solution space of ω, to find even a good approximate partition of tracklets into trajectories is extremely challenging. We use MCMC sampling techniques as a stochastic mode seeking procedure, and extend the previous approaches
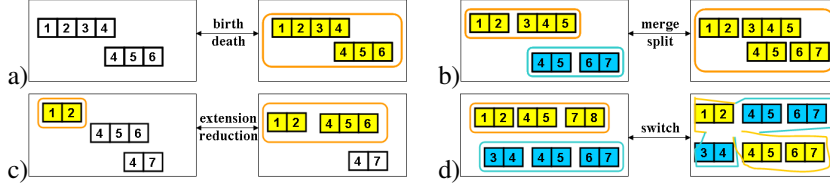
Figure 2: Different move types in MCMCDA. The tracklets in the same color belong to the same trajectory and false alarms are drawn in white.

in [6, 11] with a fully Bayesian treatment that simultaneously estimates the model parameters λ along with the observation partition ω.

From Eqn. 5, 10, and 11, we can derive the full conditional distributions for ω and λ.

$$
\begin{aligned}
p(\lambda_1|-) &\propto p(Z|\omega,\lambda_1)p(\lambda_1) \propto \lambda_1^{\sum_k |\tau_k|-K} e^{-\lambda_1 M_1} \lambda_1^{\theta_1^0-1} e^{-\frac{1}{\theta_1^1}\lambda_1} \\
&= \lambda_1^{(\alpha_1-1)} e^{-\frac{1}{\beta_1}\lambda_1} \sim \texttt{Gamma}(\alpha_1,\beta_1)
\end{aligned}
\tag{12}
$$

where $\alpha_1 = (\theta_1^0 + \sum_k |\tau_k| - K)$, $\beta_1 = (\frac{1}{\theta_1^1} + M_1)^{-1}$. Similar derivations show that all $p(\lambda_j|-) \sim \texttt{Gamma}(\alpha_j,\beta_j)$. where

$$
\alpha = \{(\theta_{1\ldots4}^0 + \sum_k |\tau_k| - K), (\theta_5^0+1), (\theta_6^0+K), (\theta_7^0+1), (\theta_8^0+K)\}, \beta = (\frac{1}{\theta^1}+M)^{-1} \tag{13}
$$

We see that by adopting exponential models and conjugate Gamma priors, the full conditional distributions for λ are also Gamma distributions, which can be efficiently sampled using a Gibbs sampler [2].

A similar derivation leads to the full conditional distribution for ω

$$
p(\omega|-) \propto p(Z|\omega,\lambda)p(\omega|\lambda) \propto \prod_{j=1}^{4} \lambda_j^{\sum_k|\tau_k|-K}\lambda_6^K\lambda_8^K e^{-\lambda M^T} \tag{14}
$$

This is not a known distribution, so we resort to the Metropolis Hastings algorithm [2]. By design, a series of reversible proposal moves yields a Markov chain that is irreducible, aperiodic, and that converges to a stationary distribution by the ergodic theorem [6]. In our case, the stationary distribution $\pi(\omega)$ is defined in Eqn.14, and the acceptance ratio is computed as

$$
A(\omega,\omega') = \min(1, \frac{\pi(\omega')q(\omega',\omega)}{\pi(\omega)q(\omega,\omega')}) \tag{15}
$$

The proposal distributions $q(\omega,\omega')$ consists of four pairs of reversible moves, as illustrated in Fig.2. To describe the constructions of the proposals, we first introduce a neighborhood tree structure of observations similar to [6], designed to make the search space manageable. Tracklet Z_1 is the parent of tracklet Z_2 if their initial frame numbers are within the maximal allowed missing frames T_{max} and if their spatial distance falls below a threshold controlled by the maximal speed of the targets. The probability of proposing each type of move, $p_\omega(m)$, is essentially a uniform distribution, but adapted to the current configuration of ω for better efficiency. For example, if the number of trajectories $K=0$,

---

Algorithm 1 MCMCDA with parameter estimation

Input: Z, n_{mc}, ω_0, θ Output: λ^*, ω^*

Initialization: $\omega \leftarrow \omega_0$, $\lambda \sim \texttt{Gamma}(\theta^0, \theta^1)$, $(\lambda^*, \omega^*) = (\lambda, \omega_0)$

for $n = 1$ to n_{mc}

    update ω sample a move m from the distribution $p_\omega(m)$

        propose ω' from the move specific proposal $p_m(\omega|\lambda)$

        sample $U \sim \texttt{Uniform}(0,1)$

        $\omega \leftarrow \omega'$ if $\log(U) < \log(A(\omega, \omega'))$

    update λ update α, β according to Eqn. 13

        sample $\lambda \sim \texttt{Gamma}(\alpha, \beta)$

        $(\lambda^*, \omega^*) \leftarrow (\lambda, \omega)$ if $p(\omega, \lambda|Z) > p(\omega^*, \lambda^*|Z)$

---

only the birth move is allowed.

Birth/Death. Every birth move proposes a new trajectory by sampling uniformly at random (u.a.r.) from the current set of free tracklets τ_0 in ω. We then extend τ'_{K+1} by recursively appending a child tracklet of the current ending tracklet with probability γ. The child tracklet is chosen based on consistency between the child and parent tracklets as defined in the likelihood function. Hence, we define $ext(child) \propto (1 - \log(\ell(child|parent)))^{-1}$, with ℓ as given by Eqn. 3. The birth move is rejected if $|\tau_{K+1}| < 2$ because we cannot distinguish between a false alarm and a trajectory with only a single tracklet. For the (reverse) death move, we choose k u.a.r. from $\{1, \ldots K\}$ and delete τ_k from ω, adding the tracklets associated with τ_k back to the set of false alarms τ_0.

Extension/Reduction. In an extension move, a trajectory τ_k is selected u.a.r. from ω and extended by the same recursive procedure as in the birth move. In a reduction move, we pick a tracklet τ_k u.a.r. and then select a break point i from $\{2, \ldots, |\tau_k|\}$ according to the probability $b_k(i) \propto -\log(\ell(\tau_{k_{i+1}}|\tau_{k_i}))$, which is inversely proportional to the consistency measure, so the trajectory is likely to break at its weakest link. The tracklets after the break point are added back to the false alarm set. The same operations are performed backwards in time in a similar manner.

Split/Merge. The split move is similar to a reduction, but instead of freeing up the chain of tracklets after the break point, it becomes a new trajectory. Specifically, we u.a.r. select τ_k and select a break point from $\{2, \ldots, |\tau_k| - 1\}$. To propose a merge move, we pick a pair of trajectories (τ_i, τ_j) u.a.r. from the set of all possible merge pairs $G = \{(\tau_i, \tau_j) : \tau_j(t_1) \in child(\tau_i(t_{|\tau_i|}))\}$ and append τ_j to the end of τ_i (the rectangles in the temporal overlap between the tracklets are averaged).

Switch. This move is included to help explore the solution space. It is essentially the same as a series of birth/death and split/merge moves. We select a pair of trajectories $(\tau_i(t_p), \tau_j(t_q))$ u.a.r. from the set of switchable trajectories $\{(\tau_{i_p}, \tau_{j_q}) : \tau_{j_{q+1}} \in child(\tau_{i_p}) \,\&\, \tau_{i_{p+1}} \in child(\tau_{j_q})\}$. The tail sections of the two trajectories after their switch points are swapped.

To summarize, by introducing the hyperpriors θ over the model parameters λ, our algorithm is able to estimate λ and the object trajectories ω in a unified Bayesian framework so that the tracking method can adapt to different videos automatically. The MAP estimation is computed by MCMC sampling, where λ can be sampled easily by a simple Gibbs sampler, thanks to the choice of the conjugate Gamma prior, and ω is sampled using the Metropolis Hastings algorithm with reversible moves. The complete algorithm

is summarized in Algorithm 1.

4 Experimental Results

We first illustrate our algorithm using a sequence from the EU Caviar Project[1]. The supplied ground truth trajectories are broken up to create a set of overlapping tracklets of length 30 frames, starting at every 10th frame. We obtain the optimal parameter λ^* from the ground truth tracklets and show how the estimated trajectories are affected by perturbing the parameters. In Figure 3(c), we plot the estimated number of trajectories against λ_8, the parameter for the spatial overlap term. Even such a crude measure shows the importance of setting proper parameter values. If the value of λ_8 is too small, the overlapping tracklets do not get merged properly and are instead hypothesized as new trajectories (Figure 3(a)). The appropriate parameter has to be set to achieve optimal association (Figure 3(b)). Note that there are also correlations among the model parameters that make them difficult to fine tune, and therefore adaptive methods are desired to determine the optimal parameter values automatically.
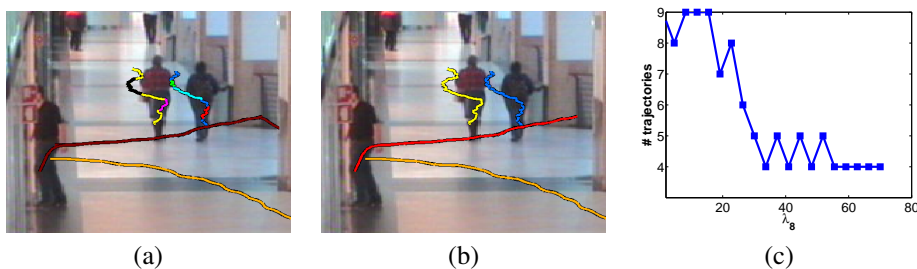

(a)   (b)   (c)

Figure 3: Illustration of multi-target data association by tracklets and the influence of one model parameter. Left: a "bad" partition of tracklets where more than four trajectories were estimated because λ_8 is small. Middle: a "good" partition of tracklets into sets associated with individual objects, each drawn in a different color. The orange one is the trajectory of a person not currently visible. Right: The total estimated number of targets varies while changing even one single model parameter.

We next show the learnt parameters and estimated trajectories for real scenes. The first test sequence is a challenging multi-target soccer sequence[2]. Players were automatically detected at every 10th frame via background subtraction and used to seed simple correlation-based template trackers to generate tracklets. The second sequence is captured using a Sony camcorder at an outdoor art's festival. The task becomes more challenging here due to lower camera elevation angle and higher crowd density, which lead to more occlusion and more complex trajectory dynamics. We use an edge-based head detector for detection and color-based particle filter for tracklet generation. Figure 4 shows the estimated trajectories for each case and the inferred model parameters. The algorithm

---

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[2] http://www.cvg.cs.rdg.ac.uk/VSPETS/vspets-db.html

is able to use noisy tracklets (Figure 4(a,e)) generated by a variety of simple trackers to recover reasonable trajectories under challenging situations.

5   Conclusion

In this paper, we use tracklet features for multi-target tracking, which provides greater spatio-temporal context for data association. By introducing a hierarchical Bayesian model, we propose a principled method for unsupervised learning of model parameters and object trajectories. The MAP solution is made computationally tractable by MCMC sampling techniques. This algorithm could be extended easily to an online version by using sliding temporal windows.

6   Acknowledgements

References

[1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, San Diego, CA, 1988.

[2] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.

[3] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 1–8, 2007.

[4] E. Levina and P. Bickel. The Earth Mover's Distance is the Mallows distance: Some insights from statistics. In *International Conference on Computer Vision*, pages II: 251–256, 2001.

[5] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, H. Wang, B. Yang, L. Sun, and S. Yang. Automatic player detection, labeling and tracking in broadcast soccer video. In *Proc. British Machine Vision Conf.*, pages –, 2007.

[6] S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo data association for general multiple-target tracking problems. In *Proc. IEEE Int. Conf. on Decision and Control*, pages 735–742, 2004.

[7] D. B. Reid. An alogirhthm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854, 1979.

[8] K. Smith, D. Gatica-Perez, and J. Odobez. Using particles to track varying numbers of objects. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 962–969, 2005.

[9] L. D. Stone, C. A. Barlow, and T. L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, Norwood, MA, 1999.

[10] M. Yang, Y. Wu, and S. Lao. Intelligent collaborative tracking by mining auxiliary objects. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 17–22, 2004.

[11] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal Markov Chain Monte Carlo data association. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
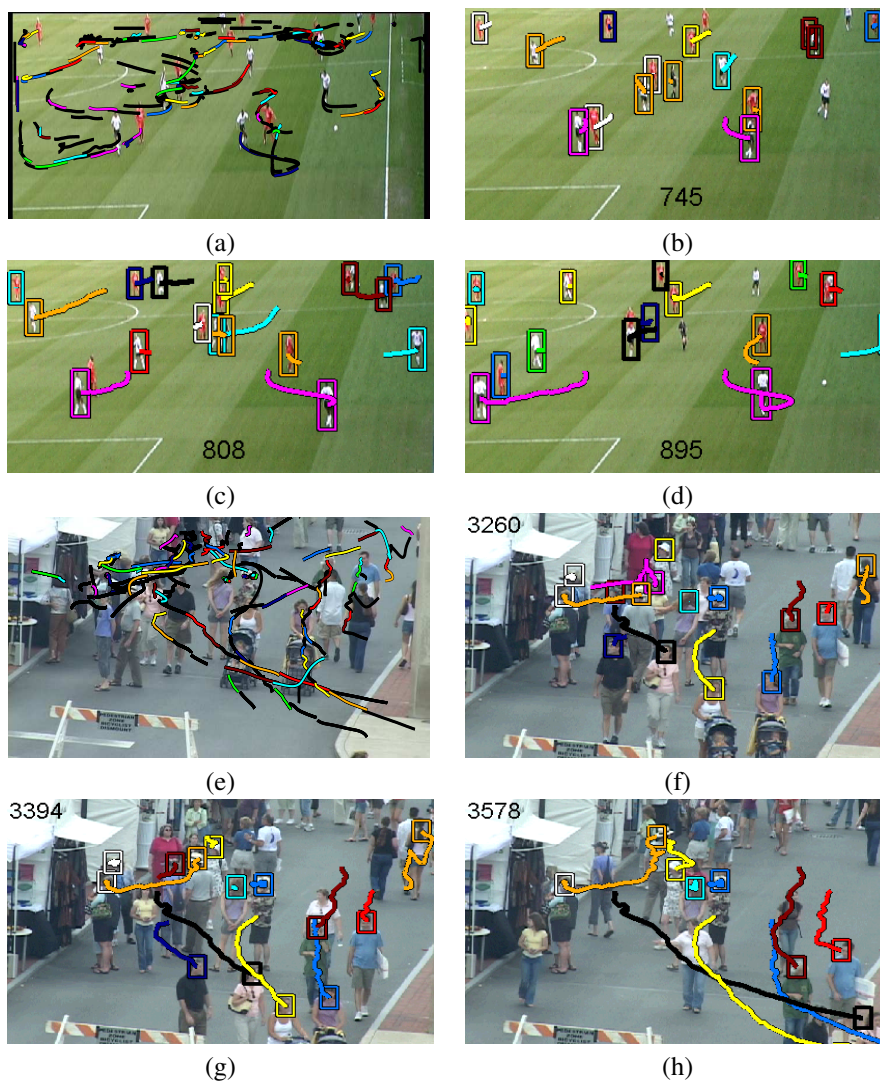
Figure 4: Results for two different scenes. Soccer (a-d): the raw fragmented tracklets and snapshots of trajectories extending, with a single color for each trajectory. ArtFest (e-h): raw tracklets and estimated trajectories for a mix of moving and static objects. The estimated λ for the Soccer and the ArtFest clips are $(11, 22.1, 5.2, 804, 10.1, 3.2, 67, 1343)$ and $(28, 50, 7, 858.9, 12.9, 5.7, 87, 1299)$ respectively.

[12] F. Dellaert Z. Khan, T. R. Balch. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1805–1918, 2005.

[13] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 406–413, 2004.