Latent Data Association: Bayesian Model Selection for Multi-target Tracking

Aleksandr V. Segal
Department of Engineering Science,
University of Oxford
avsegal@robots.ox.ac.uk

Ian Reid
Department of Computer Science,
University of Adelaide
ian.reid@adelaide.edu.au

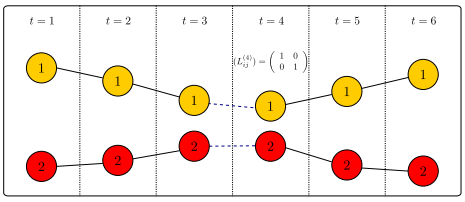
Abstract

We propose a novel parametrization of the data association problem for multi-target tracking. In our formulation, the number of targets is implicitly inferred together with the data association, effectively solving data association and model selection as a single inference problem. The novel formulation allows us to interpret data association and tracking as a single Switching Linear Dynamical System (SLDS). We compute an approximate posterior solution to this problem using a dynamic programming/message passing technique. This inference-based approach allows us to incorporate richer probabilistic models into the tracking system. In particular, we incorporate inference over inliers/outliers and track termination times into the system. We evaluate our approach on publicly available datasets and demonstrate results competitive with, and in some cases exceeding the state of the art.

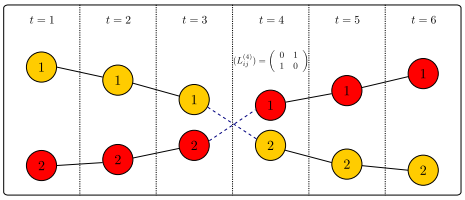
1. Introduction

Multi-target tracking is an important, but stubborn problem in Computer Vision as well as many related fields (notably robotics). The applications range from surveillance, through autonomous navigation, to active scene modeling and understanding. Despite the numerous motivations for solving this problem, it has remained a challenging topic after decades of active research. Historically, it has been difficult for two reasons. The first is the combinatorial space of possible associations between the observations and objects being tracked, and the second is model selection over the number of existing tracks.

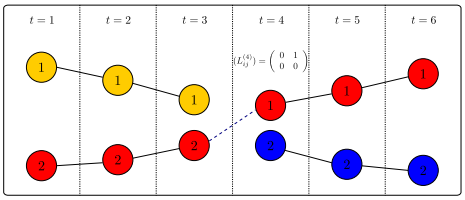
In this paper we propose *Latent Data Association* as an alternative parametrization of the data association problem where the number of underlying target tracks is implicit in the data association. We treat the new parametrization as a special case of a *Switching Linear Dynamical System* (SLDS) [19], and perform approximate inference using a message passing technique.



(a) Assignment #1



(b) Assignment #2



(c) Assignment #3

Figure 1. Illustration of three possible Latent Data Association assignments at $t = 4$. The binary indicator matrix $(L_{ij}^{(4)})$ controls the matching of nodes between $t = 4$ and $t = 3$. Nodes are numbered within each time slice and colored based on their global track membership. Each node represents a single latent track state together with any observations (if they exist).

By treating multi-target tracking as an approximate hybrid inference problem, more complex reasoning about object classification can be incorporated into the same algorithm used for data association and tracking. In this spirit, we take advantage of advances in in the state of the art of object detection and classification [10, 11, 17, 21] by incorporating object/target classification directly into our system.

This is accomplished by adding discrete object category

variables into the tracking model. The outputs of a standard object detector can then be used as observations of the target's category. Using this model allows the classification and tracking problem to be naturally combined into a single system where statistical relationships between target motion (tracking) and target identity (detection and classification) can be exploited.

2. Previous Work

Classical approaches to multi-target tracking were pioneered decades ago assuming point-like targets such as radar returns. Most of these were progressive variations and generalizations of single target tracking in a cluttered environment. The *Probabilistic Data Association Filter* (PDAF) [5] only deals with a single target at a time, but introduced the notion of soft data association based on a weighted mixture of measurements. The *Joint Probabilistic Data Association Filter* (JPDAF) [13] generalizes the PDAF to take into account multiple targets. The *Multiple Hypothesis Tracker* (MHT) [22] keeps a list of all possible data association hypotheses and the resulting filter outputs for each target.

More recently, *Tracking by Detection* (TBD) [1] has become popular. This technique re-frames multi-target tracking as the fusion of an object detector [10, 11, 21] with data association. In contrast to classical methods focusing on radar data with point measurements, TBD literature has focused on tracking objects in video sequences. Out of the recent work, two directions can be identified.

Probabilistic Occupancy Map (POM) based approaches accumulate detections on a discretized grid. The tracking question is formulated as linking compatible detections on the grid into consistent trajectories. Berclaz *et al* [7] form a sparse graph over every hypothetical discrete object locations. Finding tracks is formulated as a network optimization problem with a global solution. Andriyenko *et al* [2] use a relaxed *Integer Linear Program* to achieve an alternative global solution to the problem.

Discretizing the tracking space limits applications (e.g. it is not easy to combine with a moving sensor platform) and forces a compromise between accuracy and the size of the tracking area. Unlike these approaches, we do not make any discretization of the search space. All continuous variables are treated as such and smoothing of the output trajectories is done implicitly via the motion model without any post-processing.

As an alternative to discretization, the second approach can be described as *Detection Partitioning*. In this case, the set of discrete detections is partitioned into tracks without explicitly enumerating what happens to the target in between successive detections. Jiang *et al* [14] formulates data association as a *Linear Program* (LP) over the sparse graph of detections. Zhang *et al* [27] use a network flow approach over an analogous sparse graph. These approaches, and others like them, tend to ignore the traditional observation model by assuming target locations are fully observed, thus requiring a separate post processing step to smooth the resulting trajectories.

Monte Carlo based approaches represent the distribution over the state space as a set of discrete samples. They are both principled and simple to implement, even for complicated non-linear models. In the case of *Particle Filters* (PF), these samples are manipulated so that their distribution tracks the posterior of the filter. The JPDAF can be implemented as a PF [24, 25] in order to track people from a mobile platform using 2D laser range data. Khan *et al* [15] use a *Markov Chain Monte Carlo* (MCMC) based particle filter to incorporate motion priors over target interactions. Breitenstein *et al* [9] introduce the *Detector Confidence Particle Filter* (DCPF) to directly incorporate detector scores as a measure of confidence. PF approaches are particularly prone to the 'curse of dimensionality' and do not scale well as the state space dimension increases.

MCMC can also be used as an independent tracking algorithm by sampling over the joint posterior of the whole problem. Oh *et al* [20] use MCMC in this way to directly sample over partitions of the detections and their posteriors. Recently, Benfold *et al* [6] proposed a real-time global MCMC strategy which simply ignores the continuous state variables of the targets and samples directly over groupings of observations. This has the disadvantage of losing the latent/hidden state space of the targets and so requires post-processing to recover smooth trajectories.

Andriyenko *et al* [3, 4] formulate tracking as a direct optimization problems over splines, and in the latter case discrete track labels. This approach is similar in spirit to ours, but is not amenable to an obvious probabilistic interpretation. Leibe *et al* [16] propose a different batch method where an over-complete set of trajectory hypotheses is pruned down to the most likely non-contradictory set using a *Quadratic Boolean Program*.

Random Finite Sets [18, 26] are a proposed alternative probabilistic calculus designed specifically for dealing with finite sets of targets. Here, a specialized theory is developed for treating a dynamically sized set of target states as a single random variable to be tracked. This is perhaps the most principled approach to multi-target tracking, but unfortunately requires a specialized set of mathematical tools. Our method offers some of the same advantages, but stays within the 'standard' probabilistic framework.

3. Traditional Data Association

Before introducing *Latent Data Association*, we review the classical formulation as a motivation for the subsequent section. We assume a fixed number of tracks and attempt to simultaneously find the target trajectories and the data association of observations to targets.

Consider a set of observations $Z = \{Z^{(1)}, \ldots, Z^{(T)}\}$ with $Z^{(t)} = \{z_1^{(t)}, \ldots, z_{N_t}^{(t)}\}$ and t denoting time. Depending on the problem, each observation z_{ti} could include 2D/3D target locations as well as dimensions and other properties. These observations are assumed to be generated by M distinct targets. Each target, $m \in \{1, \ldots, M\}$ follows the trajectory $X_m = (x_m^{(1)}, \ldots, x_m^{(T)})$. The data association problem is classically formulated as finding a correspondence between the targets and observations at each point in time. This is done by introducing a set of discrete decision variables, $D = \{D^{(1)}, \ldots, D^{(T)}\}$, with $D^{(t)} = \{d_i^{(t)}\}$, which control the associations. In this notation, $d_i^{(t)} = j \in \{1, \cdots, M\}$ indicates that the observation $z_i^{(t)}$ is associated with the j^{th} target, with the constraint that no two observations can be assigned to the same target. The value $d_i^{(t)} = 0$ indicates an outlier observation not associated with any particular target. The graphical model for this problem is shown in Figure 2a for reference.

If D is known, it is possible to infer the posterior trajectories, $P(x_m^{(1:T)}|D, Z)$, using a Kalman smoother. With D unknown, however, we are forced to consider all possible data associations. This can be formulated as a posterior

$$P(X_m|Z) = \sum_D P(X_m|D, Z)P(D) \tag{1}$$

or as a MAP problem

$$X_m^* = \underset{X_m, D}{\operatorname{argmax}}\, P(X_m|D, Z)P(D) \tag{2}$$

In either case, an approximation must be made to deal with the combinatorial number of possible values for D. Various search strategies exist for finding a 'good' D, but these are often prone to local minima.

Even if we were to avoid enumerating all values of D in the above, 'proper' Bayesian model selection over the number of tracks, M, still requires this enumeration because the posterior likelihood is given by

$$P(M|Z) = \frac{P(Z|M)P(M)}{P(Z)} \propto \tag{3}$$

$$\propto P(M) \sum_D \int \Big(P(Z|X, D)P(X|M) \Big) dX \tag{4}$$

Whereas for a fixed M we can avoid the enumeration by restricting ourselves to a MAP estimate and local optimization, the same approach cannot be used for model selection. To calculate the probability of a given value of M, we must consider the likelihood of all possible data associations conditioned on the existence of exactly M targets.

4. Latent Data Association

Our *Latent Data Association* parametrization avoids the difficulties of the previous section. While the classical approach attempts to assign observations to previously existing tracks, *Latent Data Association* starts by assuming that each detection is its own track (of length 1) with a permanently associated hidden state variable. The problem of tracking then becomes a question of linking these singleton tracks into longer trajectories. We do this by assigning each track at time t as the continuation of some track at $t-1$. This amounts to a set of discrete variables controlling how to join the tracks after time t with those existing up to time $t-1$. We refer to this form of data association as *latent* because the discrete variables now control associations between adjacent latent state variables. Figure 1 illustrates this parametrization with the tracks being spliced between $t = 3$ and $t = 4$.

To define this model formally, we define a *node* as the set of hidden state variables associated with some track at a specific time instance, as well as any observations of this state. Each node is denoted by the pair $n = (t, i)$, where t is the time index, and i an index within that time slice (illustrated in Fig. 1). For $n = (t, i)$, we define x_{ti} as the unobserved state variables of the node and z_{ti} as the observations (if present).

The binary indication matrix $L_{ij}^{(t)}$ is used to control the latent data associations at time t; setting $L_{ij}^{(t)} = 1$ corresponds to linking node (t, i) with node $(t-1, j)$. If $\forall j$, $L_{ij}^{t} = 0$, we know that node (t, i) is not linked with anything in the past and hence represents the start of a new track. In order to ensure track continuations are always one-to-one, we must enforce the mutual exclusion constraints $\sum_i L_{ij}^{(t)} \leq 1$ and $\sum_j L_{ij}^{(t)} \leq 1$.

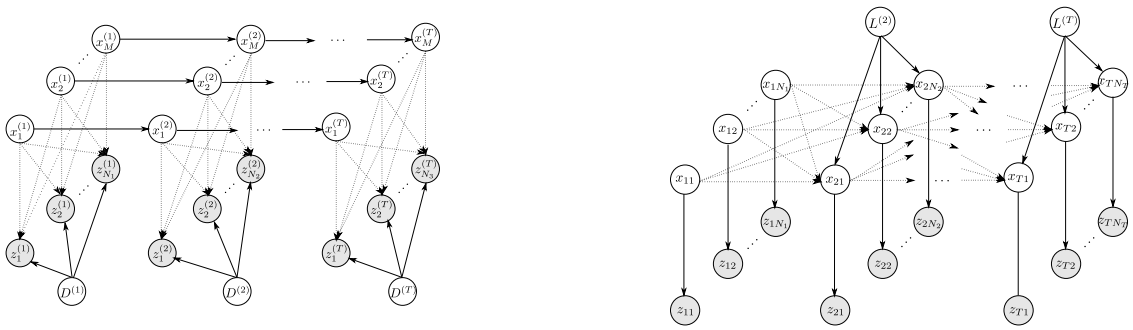
Given these definitions, the set of nodes combined with a value for each $L^{(t)}$ matrix forms a graph structure, seen in Fig. 1 , where each connected component represents an independent track. This parametrization of the problem subsumes standard data association as well as model selection over the number of tracks; any number of tracks and any data association can be represented with a suitable value for $L = \{L^{(t)}\}$.

By fixing the set of latent data association indicators, we partition the nodes into independent tracks. Within each such track, we have the standard motion and observation models. Each observation z_{ti} is generated from the associated target state x_{ti} according to an observation model, $P(z_{ti}|x_{ti})$. The motion model between any two nodes is specified conditional on these nodes being connected:

$$P(x_{ti}|x_{t-1,j}, L_{ij}^{(t)} = 1) \tag{5}$$

The associated graphical model is shown in Fig. 2b .

If we assume linear motion and observation models, the model forms an SLDS [19] where the discrete $L^{(t)}$ variables control the relationships between continuous variables in the Markov Chain. This SLDS can be used to implicitly



(a) Classical data association

(b) Latent Data Association

Figure 2. Graphical models contrasting Latent Data Association with the classical approach. Dashed lines represent dependencies controlled by the data association variables $D^{(t)}$ or latent data association variables $L^{(t)}$ respectively.

solve the data association problem together with model selection over the number of targets.

5. Approximate Inference

We propose an iterative approximate inference technique to solve the SLDS introduced in the previous section. The goal in this section is to pick $L^* = \operatorname{argmax}_L P(Z|L)$ and compute the smooth trajectories $X^* = \operatorname{argmax}_X P(X|Z, L^*)$. Our technique is based on re-using the computation already required for smoothing to also optimize over L.

If the value of L were known, the problem could be reduced to smoothing trajectories based on the partitioned observations. Although many equivalent formulations are possible, we use the notation of a message passing algorithm to describe the smoothing process with L fixed. For a node (t, i), we define $\mathtt{pr}_{ti}$ as the index of the previous node (at $t-1$) in the same track and $\mathtt{nx}_{ti}$ as the index of the next node (at $t+1$). As a shorthand, we also define $x_{ti}^{\mathtt{pr}} \equiv x_{t-1,\mathtt{pr}_{ti}}$. The forward and backward messages respectively can then be defined recursively as

$$\overrightarrow{\mu}_{ti}(x_{ti}) = \int_{x_{ti}^{\mathtt{pr}}} \overrightarrow{\mu}_{t-1,\mathtt{pr}_{ti}} \cdot P(z_{ti}|x_{ti}) \cdot P(x_{ti}|x_{ti}^{\mathtt{pr}}) \quad (6)$$

$$\overleftarrow{\mu}_{ti}(x_{ti}^{\mathtt{pr}}) = \int_{x_{ti}} \overleftarrow{\mu}_{t+1,\mathtt{nx}_{ti}} \cdot P(z_{ti}|x_{ti}) \cdot P(x_{ti}|x_{ti}^{\mathtt{pr}}) \quad (7)$$

After computing both sets of messages, all information about each node will be contained in

$$B_{ti}(x_{ti}, x_{ti}^{\mathtt{pr}}) = \overrightarrow{\mu}_{t-1,\mathtt{pr}_{ti}} \cdot P(z_{ti}|x_{ti}) \cdot P(x_{ti}|x_{ti}^{\mathtt{pr}}) \cdot \overleftarrow{\mu}_{t+1,\mathtt{nx}_{ti}} \quad (8)$$

Note that B_{ti} is proportional to the marginal posterior over $(x_{ti}, x_{ti}^{\mathtt{pr}})$, but does not necessarily integrate to one.

At this point, we have computed the posterior over X by assuming a fixed value of L. To optimize over L we consider the marginal likelihood of a given track, computed by integrating out all relevant X variables. This quantity can be efficiently retrieved from any node along the track as

$$m_{ti} = \int_{x_{ti}, x_{ti}^{\mathtt{pr}}} B_{ti} \quad (9)$$

Eq. 9 allows us to maximize the marginal likelihood of all tracks present at t over L^t while holding $L^{(t')}$ fixed for $t' \neq t$:

$$L^{*(t)} = \operatorname*{argmax}_{L^{(t)}} P(Z|L) = \operatorname*{argmax}_{L^{(t)}} \prod_i m_{ti} \quad (10)$$

This optimization can be solved as a Linear Assignment Problem (LAP) between nodes at t and $t-1$ formulated via the (constrained) binary indicator matrix $L^{(t)}$:

$$\max_{L^{(t)}} \sum_{ij} L_{ij}^{(t)} \cdot \log m_{tij} \quad (11)$$

$$m_{tij} \equiv \int_X \overrightarrow{\mu}_{t-1,j} \cdot P(z_{ti}|x_{ti}) \cdot P(x_{ti}|x_{t-1,j}) \cdot \overleftarrow{\mu}_{t+1,\mathtt{nx}_{ti}} \quad (12)$$

Note that m_{tij} is the hypothetical value of m_{ti} if we had torn the node (t, i) from its current assignment and attached it to node $(t-1, j)$ instead.

Picking a new value of $L^{(t)}$ according to Eq. 11 does not affect any of the forward messages before time t or any of the backward messages after time t – these only depend on values of $L^{(t')}$ for $t' < t$ and $t' > t$ respectively. This allows us to interleave optimization over $L^{(t)}$ into the standard message passing procedure. We use the messages $\{\overrightarrow{\mu}_{t-1}\}$ and $\{\overleftarrow{\mu}_{t+1}\}$ to update $L^{(t)}$, and subsequently use the new value of $L^{(t)}$ to compute the forward messages $\{\overrightarrow{\mu}_t\}$. Virtual nodes with no observations are added at time t for any nodes from $t-1$ which were left unassigned. The process is repeated going forward; at each point increasing $P(Z|L)$. The backward pass of the algorithm remains unchanged from a standard smoother. This modified forward-backward procedure is repeated until convergence. An outline of the inference procedure is listed in Fig. 3.

procedure FORWARDMESSAGEPASS
  for $t = 1 \dots T$ **do**
    remove all virtual nodes at t
    for all $n = (t, i), n' = (t-1, j)$ **do**
      compute c_{tij} using Eq. 12
    end for
    re-estimate $L^{(t)}$ using Eq. 11
    add virtual nodes at t
    for all $n = (t, i)$ **do**
      update forward message $\overrightarrow{\mu_{ti}}$ using Eq. 6
    end for
  end for
end procedure

Figure 3. Approximate message passing procedure used for inference in the forward direction.

6. Pedestrian Tracking by Detection with Latent Data Association

Up to this point we have described the Latent Data Association parametrization and inference algorithm in general terms. We now introduce the practical implementation and extensions used for the presented evaluations. To this end we describe the observation and state space models for both 2D and 3D tracking, as well as extensions to handle false positive detections and track length priors. Fig. 4 illustrates the graphical model for a single node with the modifications described in this section.

Since every detection now corresponds to a track, outliers must correspond to outlier tracks, leading to an extra discrete state variable, $c_{ti} \in \{\texttt{pedestrian}, \texttt{outlier}\}$, representing the target class. To go with the class model, a prior $P(c_{ti})$ and transition model $P(c_{ti}|c_{t-1,j}, L_{ij}^{(t)} = 1)$ must be defined. In our evaluation, we use only two classes, but in principle the formulation allows for more.

The pedestrian detectors we use are discriminative, so no generative model exists to explain the observations based on the target class. To compensate, we train the observation model for the detector. The score of each detector firing is treated as a real-valued observation, s_{ti}, conditioned on the class. Kernel Density Estimation (Gaussian kernel with a width of 0.05) is used to estimate the distributions

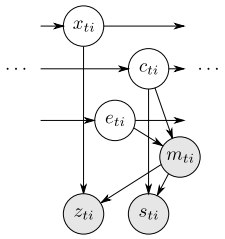


Figure 4. Graphical model of the variables in the extended model for a single node $n = (t, i)$.

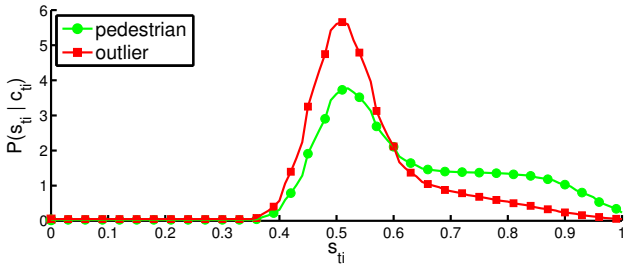


Figure 5. Learned model of the object detector firing score conditioned on object class, $P(s_{ti}|c_{ti} = \{\texttt{pedestrian}, \texttt{outlier}\})$.

$P(s_{ti}|c_{ti})$. The distribution is trained by matching detector firings with ground truth annotations over sequences out of the PETS'09 dataset [12] (the `S2.L1` sequence is excluded since it is used for evaluation). Fig. 5 shows the conditional distributions of the trained model.

In practice, a lot of information is contained in the missing detections – a track with very few detections is more likely to be an outlier than one with many consistent detections. To incorporate this negative information, we include detector failure into the observation model. The indicator variable $m_{ti} = 1$ is used to denote a missing observation at node $n = (t, i)$. In this case n is a virtual node and the z_{ti} and s_{ti} observation variables are ignored. We allow missing observations to occur with probability dependent on the underlying class.

Finally, we include a track length prior. Because of the detector failure model, we cannot assume a track continues on indefinitely after its last observation – doing so would imply a very large number of missing observations and make all tracks likely to be outliers. Instead, we give each target track a fixed probability of terminating at every time instance after its last observation. We introduce the indicator variable e_{ti} to mark that the track has ended. Once this variable transition from 0 to 1, a transition in the other direction is not possible. If $e_{ti} = 1$, we require that $m_{ti} = 1$; once a track ends, it cannot have any additional observations. Otherwise the behavior of m_{ti} is as described above.

7. Modified Inference Procedure

Incorporating the changes of Sec. 6 into the approximate inference procedure described in Sec. 5 is not difficult since all of the modifications can be represented as additional discrete components in the Markov chain. Furthermore, Eq. 8 and Eq. 9 do not depend on the Markov chain being continuous; analogous equations hold for a discrete chain if the marginalization integrals are replaced with sums.

We run discrete message passing over e_{ti} and c_{ti} and compute the track log-likelihood of the data by adding the log-likelihoods obtained from Eq. 9 applied to the discrete

and continuous Markov chains independently. As before, we update $L^{(t)}$ by solving the LAP in Eq. 11 with the cost of each assignment based on the combined track log-likelihood.

8. Evaluation

Experimental validation was performed using four publicly available video sequences comprising over 1200 frames from two standard pedestrian tracking datasets (TUD [1] and PETS'09 [12]). 2D tracking was used for the TUD datasets and 3D tracking for the PETS sequence. We ran 2D tracking on `TUD-Stadtmitte` despite the available camera calibration because the oblique viewing angle makes accurate estimation of ground plane positions difficult. Raw detections, ground truth annotations, and tracking area specifications provided by Andriyenko *et al* [4] were used for all evaluations. Results are presented in terms of the CLEAR MOT [8] metrics for tracking performance and precision-recall curves for classification accuracy. We also include the number of fragmentations (FM), mostly tracked targets (MT), and identity switches (IDS). All evaluations use a 50% intersection over union threshold for matching 2D bounding boxes.

A constant-velocity motion model with direct linear observations was used within each track:

$$x_0 \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{13}$$

$$x_{t+1} \sim \mathcal{N}(A \cdot x_t, \Sigma_{\texttt{mot}}) \tag{14}$$

$$z_t \sim \mathcal{N}(B \cdot x_t, \Sigma_{\texttt{obs}}) \tag{15}$$

In the above, A implements the constant-velocity model and the B selects the bounding box position and dimensions out of the state space.

In the 2D case, the continuous state space is composed of the bounding box center and the log of the dimensions. Dimensions are tracked in log-space to help compensate for perspective effects. Both the position, $\mathbf{p}$, and log-dimensions, $\mathbf{d}$, have an associated velocity ($\dot{\mathbf{p}}$ and $\dot{\mathbf{d}}$) resulting in an 8D state space: $(\mathbf{p}_x, \mathbf{p}_y, \mathbf{d}_x, \mathbf{d}_y, \dot{\mathbf{p}}_x, \dot{\mathbf{p}}_y, \dot{\mathbf{d}}_x, \dot{\mathbf{d}}_y)$. The position prior is centered in the image with mean log-dimensions of $\log(320)$ by $\log(240)$. The standard deviation (s.d.) is 400px for the position and 1.0 for the log-dimensions. We incorporate a correlation coefficient of 0.99 between the prior log-dimensions. The velocity prior is zero-mean with an s.d. of 5px for the center location and 0.01 for the log-dimensions. The motion model adds isotropic noise with an s.d. of 10^{-4}px, 10^{-4}, 0.5px/s, and 10^{-2}, for the $\mathbf{p}$, $\mathbf{d}$, $\dot{\mathbf{p}}$, and $\dot{\mathbf{d}}$ components respectively. The observation model is unbiased with an s.d. of 10px for $\mathbf{p}$ and 0.1 for $\mathbf{d}$.

For 3D tracking, object position is tracked on the ground plane together with the bounding box dimensions (width and height are tracked; depth is assumed equal to width). We again use a constant-velocity model for the ground plane position, but assume the dimensions follow a random walk with no velocity (unlike in the 2D tracking case, we expect the 3D dimensions to stay relatively constant). The 3D state space consists of $(\mathbf{p}_x, \mathbf{p}_y, \mathbf{d}_x, \mathbf{d}_y, \dot{\mathbf{p}}_x, \dot{\mathbf{p}}_y)$. The prior is zero mean for $\mathbf{p}$ and $\dot{\mathbf{p}}$ with an s.d. of 40m and 0.25m/s respectively. The prior for $(\mathbf{d}_x, \mathbf{d}_y)$ has mean $(0.7\text{m}, 1.7\text{m})$ with an s.d. of 0.2m. The constant velocity motion model adds isotropic noise with an s.d. of 10^{-4}m, 0.05m/s, and 0.01m for the three components of the state space respectively. We assume observation noise with an s.d. of 0.15m for the position and 0.20m for the dimensions.

The discrete model parameters are the same for both 2D and 3D tracking. We use a uniform prior over $P(c_0)$, and a transition model such that $P(\{c_t = c_{t-1}\}) = 1 - 10^{-6}$. The missing detections probability, $P(m_t|e_t = 0, c_t)$, is 0.6 for pedestrians and 0.7 for outliers. The track termination probability, $P(e_t = 1|e_{t-1} = 0, c_t)$, is set so there is a 0.0025(pedestrian) and 0.18(outlier) chance of terminating after one second.

All parameters were determined empirically and are scaled based on the time between frames, Δt, when appropriate. We note in particular the discrete Markov transition matrix, T, is adjusted to become $T^{\Delta t}$ for frame rate invariance.

Because our system keeps track of object sizes as well as location, the size of the bounding boxes output by the detector vs the size of the labeled ground truth plays an important role in the performance of the system. Since these two differ substantially in the PETS and TUD datasets, we scale the width of the bounding boxes output by our system by 0.75 to better match the ground truth labeling.

Our tracking results are shown in Tab. 1 and are competitive with the state of the art. We note that despite the widespread use of the CLEAR MOT metrics, direct comparison of published algorithms is still difficult as many authors differ in the precise evaluation methods used (2D v.s. 3D metrics, different regions of interest, etc.). Despite this, we have attempted to make an informative evaluation against recently published results – we do not imply a head-to-head comparison. Where 2D evaluations are available, we list those published by the authors. To compare with Andriyenko *et al* [4], we have run our own 2D evaluation scripts on the their data where possible, as well as listing their published results. Only 3D ground tracks were available for the `TUD-Stadtmitte` sequence. In this case we assumed average 3D pedestrian dimensions and projected these into 2D bounding boxes.

Fig. 6 shows the log-likelihood as a function of the number of forward-backward iterations performed. Note the monotonic increase in log-likelihood and convergence in a small number of iterations.

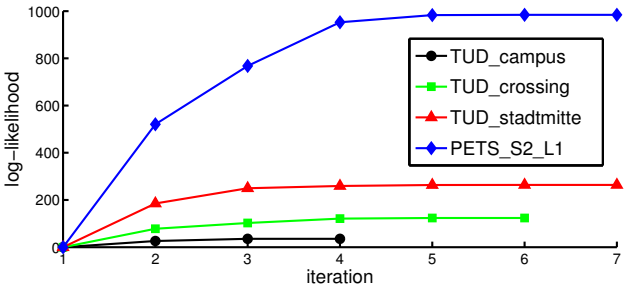


Figure 6. Convergence of the approximate inference algorithm is achieved in under 7 iterations for all evaluated sequences. The plot has been zeroed to the initial log-likelihoods.

Precision Recall curves showing improvement over the baseline detector are show in Fig. 7. These curves are possible because of the probabilistic nature of our approach where each output has an associated posterior pedestrian vs outlier probability. While these figures convey the quantitative measures of performance, we encourage the reader to view the supplementary material to observe the qualitative tracking behavior and performance.

9. Conclusions and Future Work

This paper has proposed a novel parametrization of the data association problem for multi-target tracking that has a number of very useful properties. The key idea behind our formulation is the proposal to perform latent data association, in which we seek associations between latent state variables over time. Associations between observations are then implicit, rather than being explicitly sought as in more traditional formulations. A key advantage of our formulation is that it the number of tracks – which is in fact a model selection problem – is determined automatically during inference. We have shown how this new parametrization can be solved using a factored approximate message passing algorithm, that the solution admits a probabilistic interpretation and that it permits easy extension to multi-category tracking in which visual identities and motion models are mutually beneficial. Finally we have compared our system against various state-of-the-art methods and shown that it is competitive in terms of performance as well as offering the advantages described above.

An intriguing possibility for future work is to deal with a moving camera. Indeed we believe that our framework is sufficient to incorporate fixtures and a vehicle state to yield a general SLAM environment containing both static and moving objects.

Acknowledgment

This work was supported by the Engineering and Physical Science Research Council [grant number EP/H050795] and the Australian Research Council, grant DP130104413.

Algorithm		MOTA	MOTP	IDS	MT	FM
proposed		0.82	0.74	0	5	3
Breitenstein2011[4]	[9]	0.67	0.73	2	–	–

(a) TUD-Campus

Algorithm		MOTA	MOTP	IDS	MT	FM
proposed		0.74	0.76	2	7	12
Zamir2012[4]	[23]	0.92	0.76	0	–	–
Breitenstein2011[4]	[9]	0.71	0.84	2	–	–

(b) TUD-Crossing

Algorithm		MOTA	MOTP	IDS	MT	FM
proposed		0.73	0.71	2	4	1
Zamir2012[4]	[23]	0.78	0.63	0	–	–
proposed[2]		0.63	0.73	4	4	1
Andriyenko2012[2,3]	[4]	0.61	0.68	3	6	1

(c) TUD-Stadtmitte

Algorithm		MOTA	MOTP	IDS	MT	FM
proposed		0.90	0.75	6	17	21
Zamir2012[4]	[23]	0.90	0.69	8	–	–
Andriyenko2012[3]	[4]	0.79	0.66	29	17	56
Andriyenko2012[1,4]	[4]	0.89	0.56	–	–	–
Breitenstein2011[1,4]	[9]	0.56	0.80	–	–	–
proposed[2]		0.92	0.75	4	18	18
Andriyenko2012[2,3]	[4]	0.83	0.65	24	18	43

(d) PETS'09 S2.L1 (View 1)

[1] evaluated by PETS'09 workshop
[2] cropped to tracking region of Andriyenko *et al* [3, 4]
[3] our own 2D evaluations using authors' provided output data
[4] results as published by authors

Table 1. A comparison using various tracking metrics. We use a threshold of $P(c_{ti} = \texttt{pedestrian}) \geq 0.50$) for all evaluations of our algorithm. Note that Zamir *et al* [23] makes use of appearance information, so better performance is expected.

References

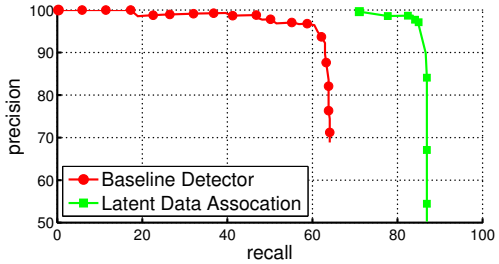
[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

[2] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In *European Conference on Computer Vision (ECCV)*, volume 6311 of *Lecture Notes in Computer Science*, pages 466–479. Springer Berlin Heidelberg, 2010.

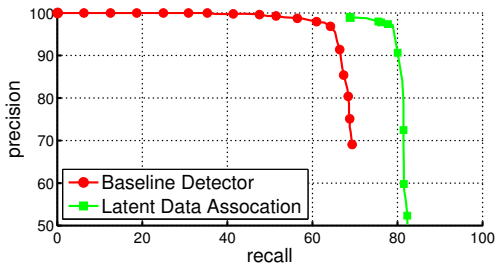
[3] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1265–1272, 2011.

[4] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1926–1933, 2012.

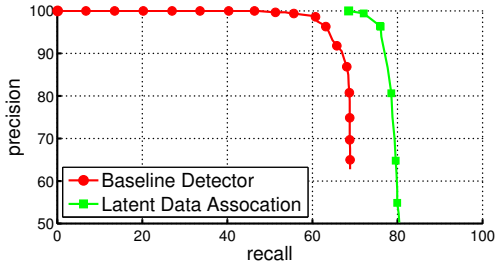
[5] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451 – 460, 1975.



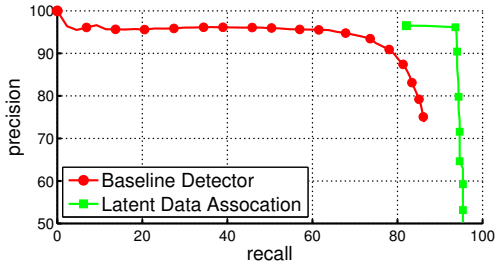
(a) TUD-Campus



(b) TUD-Crossing



(c) TUD-Stadmitte



(d) PETS'09 S2L1 View #1

Figure 7. Precision-Recall curves for all datasets plotted alongside the baseline detector.

[6] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 3457–3464, 2011.

[7] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple Object Tracking Using K-Shortest Paths Optimization. volume 33, pages 1806–1819, 2011.

[8] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):246309, 2008.

[9] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820–1833, 2011.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 1, pages 886–893 vol. 1, 2005.

[11] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.

[12] J. Ferryman, A. Shahrokni, et al. An overview of the PETS 2009 challenge. 2009.

[13] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, 1983.

[14] H. Jiang, S. Fels, and J. Little. A Linear Programming Approach for Multiple Object Tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2007.

[15] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision (ECCV)*, volume 3024 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin Heidelberg, 2004.

[16] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1683–1698, 2008.

[17] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, volume 1, pages 878–885 vol. 1, 2005.

[18] E. Maggio, M. Taj, and A. Cavallaro. Efficient multitarget visual tracking using random finite sets. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1016–1027, 2008.

[19] K. Murphy. Switching kalman filters. Technical report, Citeseer, 1998.

[20] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 735–742 Vol.1, 2004.

[21] V. Prisacariu and I. Reid. fastHOG - a real-time GPU implementation of HOG. Technical Report 09, 2009.

[22] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[23] A. Roshan Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 343–356. Springer Berlin Heidelberg, 2012.

[24] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Robotics and Automation, IEEE International Conference on*, volume 2, pages 1665–1670 vol.2, 2001.

[25] J. Vermaak, S. Godsill, and P. Perez. Monte carlo filtering for multi target tracking and data association. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(1):309–332, 2005.

[26] B.-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(4):1224–1245, 2005.

[27] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008.