Bayesian 3D tracking from monocular video

Ernesto Brau†
ernesto@cs.arizona.edu
Jinyan Guan†
jguan1@email.arizona.edu
Kyle Simek†
ksimek@email.arizona.edu
Luca Del Pero∗
ldelper@inf.ed.ac.uk
Colin Reimer Dawson‡
ccdawson@email.arizona.edu
Kobus Barnard‡
kobus@sista.arizona.edu

†Computer Science
University of Arizona
‡School of Information
University of Arizona
∗School of Informatics
University of Edinburgh

Abstract

We develop a Bayesian modeling approach for tracking people in 3D from monocular video with unknown cameras. Modeling in 3D provides natural explanations for occlusions and smoothness discontinuities that result from projection, and allows priors on velocity and smoothness to be grounded in physical quantities: meters and seconds vs. pixels and frames. We pose the problem in the context of data association, in which observations are assigned to tracks. A correct application of Bayesian inference to multi-target tracking must address the fact that the model’s dimension changes as tracks are added or removed, and thus, posterior densities of different hypotheses are not comparable. We address this by marginalizing out the trajectory parameters so the resulting posterior over data associations has constant dimension. This is made tractable by using (a) Gaussian process priors for smooth trajectories and (b) approximately Gaussian likelihood functions. Our approach provides a principled method for incorporating multiple sources of evidence; we present results using both optical flow and object detector outputs. Results are comparable to recent work on 3D tracking and, unlike others, our method requires no pre-calibrated cameras.

1. Introduction

Tracking remains difficult when there are multiple targets interacting and occluding each other. These difficulties are common in many applications such as surveillance, mining video data, and video retrieval, motivating much recent work in multi-object tracking [39, 4, 5, 23, 24, 6, 41]. In these contexts, it often makes sense to analyze extended frame sequences (“off-line” tracking), and the camera parameters are often unknown.

In this paper we develop a fully 3D Bayesian approach for tracking an unknown and changing number of people in a scene using video taken from a single, fixed viewpoint. We propose a generative statistical model that provides the distribution of data (evidence) given an association, where we extend the well-known formulation of Oh et al. [31]. We model people as elliptical right-angled cylinders moving on a relatively horizontal ground plane. We infer camera parameters and people’s sizes as part of the tracking process. Further, with a reasonable value for the mean height of people, we can establish location with respect to the camera in absolute units (i.e., meters).

This formulation enables inference in the constant dimension data-association space, provided that we integrate out the continuous model parameters such as those associated with trajectories. In other words, we estimate the marginal likelihoods during inference, which deals with potential dimensionality issues due to an unknown number of tracks. This principled approach is very amenable to extensions, such the incorporation of new model elements (e.g., pose estimation and gaze direction) or new sources of evidence (e.g., color and texture).

Given a model hypothesis, we project each person cylinder into each frame using the current camera, computing their visibility as a consequence of any existing occlusion. We then evaluate the hypothesis using evidence from the output of person detectors and optical flow. Our method thus integrates tracking as detection (e.g., [32, 23, 1]) and classical approaches like tracking as following evidence locally in time as is common in filtering methods (e.g., [20, 22]). We use a Gaussian process in world coordinates to provide a smoothness prior on motion with respect to absolute measures. Given a reasonable kernel, observations that are far apart in time do not influence each other much, and we exploit this for efficiency.

To track multiple people in videos we infer an association between persons and detections, collateral-
ing a likely set of 3D trajectories for the people in the scene. We use MCMC sampling \((8; 3)\) to sample over associations, and, for a given association, we then sample trajectories to search for a probable one, conditioned on the association. We use this to estimate the integral over all trajectories, again conditioned on the association. During inference we also sample the global parameters for the video which includes the camera and the false detection rate, which we consider to be a function of the scene background.

**Closely related work.** Our data association approach extends that of Oh et al. \([31]\). We further follow Brau et al. \([8]\) who used Gaussian processes for trajectory smoothness while searching over associations by sampling. Others \([40, 7]\) use a similar data association model, but propose an effective non-sampling approach for inference. All these efforts are focused on association of points alone; neither appearance or geometry are considered.

With respect to representation, several others share our preference for 3D Bayesian models for humans (e.g., \([36, 11, 37, 9]\)). In particular, Isard and MacCormick \([21]\) use a 3D cylinder model for multi-person tracking using a single, known camera. However, this approach does not deal with data association, since it is not detection-based. Similarly, there is other work in tracking objects on the 3D ground plane \([16, 13, 28]\) without considering data association. Other approaches estimate data association as well as model parameters \([39, 19, 10]\). However, we model data association explicitly in a generative way, as opposed to estimating it as a by-product of inference. In addition, none of these approaches model humans as 3D objects.

Andriyenko and Schindler \([3]\) pose data association as an integer linear program. In subsequent work \([4]\), they formulate an energy approach for multi-target tracking in 3D that includes terms for image evidence, physics based priors, and a simplicity term that pushes towards fewer trajectories. Later, Andriyenko et al. \([5]\) attempt to solve both data association and trajectory estimation problems using similar modeling ideas as in their previous work. In contrast to our work, they simultaneously optimize both association and trajectory energy functions, which results in a space of varying dimensionality.

**Technical contributions** include: (1) A full Bayesian formulation that incorporates both data association and the 3D geometry of the scene; (2) Robust inference of camera parameters while tracking; (3) A Gaussian process prior on trajectory smoothness applied in absolute 3D coordinates; (4) Inferring people’s heights and widths simultaneously while tracking to improve performance; (5) Explicitly handling occlusion as a natural consequence of perspective projection while tracking; (6) Extending data association tracking to use multiple detections from multiple detectors, and associated proposal strategies; (7) A new model for the prior on the number of tracks, and associated births and deaths; and (8) Integrating optical flow and detection information into probabilistic evidence for 3D tracking.

### 2. Model, priors, and likelihood

In the data-association treatment of the multi-target tracking problem \([30, 8]\), an unknown number of objects (targets) move in a volume, producing observations (detections) at discrete times. The objective is to determine the association, \(\omega\), which specifies which detections were produced by which target, as well as which were generated spuriously. Here, the targets are the people moving around the ground plane, and the observations \((B)\) are detection boxes obtained by running a person detector \([14]\) on each frame of a video.

Our goal is to find \(\omega\) which maximizes the posterior distribution \(p(\omega \mid B) \propto p(B \mid \omega)p(\omega)\), where \(p(\omega)\) is the prior distribution and \(p(B \mid \omega)\) is the likelihood function. The prior over associations contains priors over quantities like the number of tracks and the number of detections per track. The likelihood arises from modeling the underlying 3D scene captured by the video.

In our model, each person in the scene has a 3D configuration \(z_r\), which is composed of their trajectory (a sequence of points on the ground plane) and their size, which consists of height, width, and girth. We also model evidence from optical flow features \([26]\), \(I\). Using all this, we can compute the likelihood function of an association by integrating over all possible 3D configurations; that is \(p(B, I \mid \omega) = \int p(B \mid z, \omega)p(I \mid z, \omega)p(z) \, dz\) where the factors in the integrand are, respectively, the two likelihoods of the 3D scene given the two sources of data and the prior over the scene (with \(z = (z_1, \ldots, z_m)\)). The overall graphical model is shown in Figure 1.

#### 2.1. Association

Formally, an association \(\omega = \{\tau_r \subset B\}_{r=0}^m\) is a partition of the set of detections \(B\), where \(\tau_1, \ldots, \tau_m\) are called *tracks*, and represent across-time chains of observations of the objects being tracked, and \(\tau_0\) is the set of false alarms. An example association is shown in Figure 2(a). The association entity is based on well-known work by Oh et al. \([31]\), but we extend that work by (1) allowing tracks to produce multiple measurements at any given frame and (2) employing a prior on associations which allows parameters governing track dynamics and detector behavior to adapt to the environment of a particular video.

We assume an association is the result of the following generative process. When the video starts, there are \(e_1\) people in the scene. At each subsequent frame \(t\), \(e_t\) people enter the scene, resulting in \(m = \sum_{t=1}^T e_t\) tracks, whose lengths are \(l_r, r = 1, \ldots, m\). In addition, \(d_t\) people exit the scene. At frame \(t\) we also observe \(a_{rt}\) detections due to person \(r\) and \(n_t\) detections due to noise. We define \(a_t = \sum_{r=1}^m a_{rt}\) as
the number of true detections at frame $t$, and $N_t = n_t + a_t$ as the total number of detections at $t$. Finally, a fully-specified assignment in frame $t$ is a permutation of its $N_t$ detections, with the first $n_t$ associated to noise, the next $a_{1t}$ associated to the first track in the frame, etc. (see Figure 1(a))

We assume that $e_1 \sim \text{Pois}(\kappa)$, and that $l_r \sim \text{Exp}(\theta)$, $r = 1, \ldots, m$. Assuming the distribution of the number of tracks is stationary, this implies that $e_t \sim \text{Pois}(\kappa \theta), t > 1$. The number of detections per target per frame, as well as the number of noisy detections, are also Poisson distributed, with parameters $\lambda_A$ and $\lambda_N$, respectively. Under these conditions, it can be shown that the prior depends only on the total tracks $m$, entrances $e$, exits $d$, true detections $n$, and track lengths $l$, as well as the number of ways to permute track labels within frames, and detections within tracks and frames. The resulting expression for $p(\omega | \kappa, \theta, \lambda_N)$ is

$$
\frac{(\kappa e^{-\lambda_A})^m \theta^d \lambda_N^e}{\prod_{i=1}^P (N_i! e_i! n_i! \prod_{i=1}^m a_i!)}
$$

Finally, we consider $\kappa$, $\theta$, and $\lambda_N$ to depend on the video and must infer their values. Consequently, we place vague Gamma priors on them; e.g., $\kappa \sim \mathcal{G}(\alpha_\kappa, \beta_\kappa)$.

### 2.2. Scene and Camera

Each track $\tau_r \in \omega$, has a corresponding trajectory on the ground plane. The trajectory corresponding to track $\tau_r$ is $x_r = (x_{r1}, \ldots, x_{rl_r})^T$, $x_{rj} \in \mathbb{R}^2$. The length $l_r$ of trajectory $x_r$ is determined by the first and last detections of track $\tau_r$. Note that, while $\tau_r$ contains no elements for frames where the person was not detected, $x_{rj}$ is specified for every $j$ between the track’s initial and final frame. Each person has three size dimensions: width, height and girth, denoted by $d_r = (w_r, h_r, g_r)$. We will denote the 3D configuration of track $\tau_r$ by $z_r = (x_r, d_r)$.

We model motion as a realization of a multi-output Gaussian process (GP) [33, 35]. Specifically, trajectory $x_r$ is the curve generated by a sample from a GP with inputs $s_r = \{1, \ldots, l_r\}$, with the zero mean function and the squared-exponential covariance function. That is, $x_r | \tau_r \sim \mathcal{N}(0, K_r)$, where $K_r$ is the covariance matrix, whose element $(s, s')$ is given by $k(s, s') = \sigma_w^2 \exp -\frac{1}{2\sigma_2^2} (s - s')^2$, for all pairs in $s_r \times s_r$. The smoothness and scale parameters $\lambda_x$ and $\sigma_w$ are set using calibration data. Person size is a priori normally distributed, e.g., $h_r \sim \mathcal{N}(\mu_h, \sigma_h)$, following actual human size [27].

Combining these elements and assuming trajectories and sizes to be independent of one another, we get the following
prior for a scene:

\[ p(z | \omega) = \prod_{r=1}^{m} p(x_r | \tau_r, \phi) p(d_r | \phi_d), \]

where \( \phi = (l, \sigma_l) \) and \( \phi_d = (\mu_d, \sigma_d, \mu_h, \sigma_h, \mu_g, \sigma_g) \).

**Camera.** We assume a standard perspective camera \cite{18} with simplifying assumptions \cite{12}. We set the origin of the world to be on the ground plane, for which we use the \( xz \)-plane. We assume the camera center to be at \((0, \eta, 0)\) (\( \eta \) is the camera height), a pitch angle of \( \psi \), and a focal length of \( f \) (see Figure 3 (top)). Further, we assume the camera has unit aspect ratio, and that the roll, yaw, axis skew, and principal point offset are all zero. We let \( \eta, \psi \), and \( f \) have vague normal priors whose parameters we set manually. Specifically, we have \( \eta \sim \mathcal{N}(\mu_\eta, \sigma_\eta), \psi \sim \mathcal{N}(\mu_\psi, \sigma_\psi), \) and \( f \sim \mathcal{N}(\mu_f, \sigma_f) \). Assuming independence between parameters, the camera prior is

\[ p(C) = p(\eta | \mu_\eta, \sigma_\eta)p(\psi | \mu_\psi, \sigma_\psi)p(f | \mu_f, \sigma_f) \]

where \( C = (\eta, \psi, f) \).

**Projecting the scene.** We convert a 3D scene to a 2D representation by transforming every cylinder at every frame into a 2D box in the image via the camera. Given a trajectory element \( x_{rj} \), we take uniformly-spaced (3D) points on the surface of the cylinder, project them onto the image plane using the camera \( C \) and find the minimum bounding box \( h_{rj} \) around the resulting 2D points. We call \( h_{rj} \) a model box (see Figure 3 (top)).

For each model box \( h_{rj} \), we also compute the region \( \hat{h}_{rj} \) that is not occluded from the camera, as follows. First, we discretize \( h_{rj} \) into a grid of small cells. We then shoot a ray from the center of each grid cell to the center of the camera, and declare it visible if the ray does not intersect any other box. Then, \( \hat{h}_{rj} \) is simply the union of these visible cells.

**2.3. Likelihood.**

We use two sources of evidence: person detectors and optical flow. First, we run multiple person detectors on the video frames to get bounding boxes \( B_t = \{b_{t1}, \ldots, b_{tN_t}\} \), \( t = 1, \ldots, T \), where \( N_t \) is the number of detections in frame \( t \). We parametrize each box \( b_{tj} \) by \( (b_{tj}^x, b_{tj}^y, b_{tj}^w, b_{tj}^h) \), representing the \( x \)-coordinate of the center, and the \( y \)-coordinates of the top, and bottom, respectively. We also run a dense optical flow estimator on the video, which outputs a set of velocity vectors \( I_t = \{v_{t1}, \ldots, v_{tN_t}\} \) for each frame \( t = 1, \ldots, T, \) where \( N_t \) is the number of pixels in the frame. Finally, we use \( B = \bigcup_{t=1}^{T} B_t \) and \( J = \{I_1, \ldots, I_{T-1}\} \), and we denote the complete set of data by \( D = (B, J) \).

**Box likelihood.** We model data boxes as having i.i.d. Laplace-distributed errors in the \( x, \) \( y, \) \( w, \) \( h \)-directions, \( r \neq 0 \), and the corresponding model box (for simplicity, assume track \( \tau_r \) starts at \( t = 1 \)) \( h_{rt} = C(x_{rt}, d_r) \), we have that \( b_{tj}^r - h_{tj}^r \sim \mathcal{Laplace}(\mu^r, \sigma^r) \) (see Figure 3 bottom-left) which implies that \( b_{tj}^r \mid h_{tj}^r \sim \mathcal{Laplace}(h_{tj}^r + \mu^r, \sigma^r) \), and analogously for \( h_{rt}^{top} \) and \( h_{rt}^{bot} \). At each frame we also observe \( r_t \) spurious detections, which we model as uniformly distributed across the image, e.g., \( p(b_{tj}^r) = \frac{1}{w_t h_t} \) and \( p(b_{tj}^{top}) = \frac{1}{w_t} \), for all false alarms \( b_{tj} \in \tau_0 \), where \( w_t \) and \( h_t \) are the width and height of the image. Combining all these factors, and considering conditional independence, we get a box likelihood \( p(B | z, \omega, C) \) given by

\[ \prod_{b \in \tau_0} p(b | w_t, h_t) \prod_{b \notin B \setminus \tau_0} p(b | h(b), C, \phi_B), \]

where \( h(b) \) is the model box of the cylinder for the target and frame corresponding to box \( b \), and \( \phi_B = (\mu^x, \sigma^x, \mu^{top}, \sigma^{top}, \mu^{bot}, \sigma^{bot}) \).

**Image likelihood.** We aggregate optical flow vector into averages as follows. Let \( I_B \) be the set of boxes of all sizes and locations that fit within the image, and \( v(b) \) be the average of the optical flow vectors from frame \( t \) contained in box \( b \). We define \( I_t = \{v(b) \mid b \in I_B \} \), and let \( I = \{I_1, \ldots, I_{T-1}\} \) as before. Now, consider a pair of consecutive model boxes \( h_{rt} \) and \( h_{rt+1} \), and let \( u_{rt} = (u_{rt}^x, u_{rt}^y) \) be the difference of their centers (called model direction) and \( v = (v^x, v^y) \in I_t \) be the average flow vector that cor-

Figure 3. Likelihood computation. Top: the cylinder from target \( z \), in frame \( j \) gets projected via camera onto the image plane, and model box \( h_{rj} \) is computed around it. Bottom-left: The likelihood for the \( x \)-component of \( h_{rj} \) (blue) given one of its corresponding data boxes \( b \in B \) (dark red), i.e., \( b \mid h_{rj} \sim \mathcal{Laplace}(h_{rj}^x, \sigma^x) \). Bottom-right: \( h_{rj} \) along with its model direction \( u_{rj} \) (thick blue arrow) and the flow vectors it contains (dotted red arrows). The thick red arrow is the average of the flow vectors which lie in \( h_{rj} \); i.e., those not occluded by the red box.
responds to the box of location and size \( h_{r'} \). We model the error between each of their coordinates as having a Laplace distribution, so that \( v_x^\tau \mid u_x^{\tau'} \sim \text{Laplace}(u_x^{\tau'}, \sigma_x^\tau) \), and analogously for \( v_y^\tau \) (see Figure 3 (bottom-right)). Finally, any \( v \in I \) which does not have a corresponding model box has coordinates which have vague Laplace distributions, e.g., \( v_x^\tau \sim \text{Laplace}(0, \sigma_x^\tau) \).

The full image likelihood \( p(I \mid z, \omega, C) \) is

\[
\prod_{t=1}^{T-1} \left( \prod_{v \in I_t^*} p(v \mid u(v), C, \phi_t) \prod_{v \in I_t \setminus I_t^*} p(v \mid \phi_t) \right),
\]

where \( I_t^* \) is the set of foreground boxes at time \( t \), \( u(v) \) is the model direction corresponding to \( v \), and \( \phi_t \) are the Laplace distribution parameters. We can simplify this by taking advantage of the sparsity of the trajectory boxes and dividing by the constant \( \prod_{v \in I} p(v \mid \phi_t) \) to get

\[
p(I \mid z, \omega, C) \propto \prod_{t=1}^{T-1} \prod_{v \in I_t^*} p(v \mid u(v), C, \phi_t).
\]

Finally, since detection boxes and optical flow are conditionally independent, we have that \( p(D \mid z, \omega, C) = p(B \mid z, \omega, C)p(I \mid z, \omega, C) \).

**Occlusion.** Having a 3D model provides valuable information about occlusion, which we exploit in two ways. In the box likelihood computation, we replace the first factor in eq. 3 with the mixture \( [\hat{h}(b)] p(b \mid h(b), C) + (1 - [\hat{h}(b)]) p(b) \) where \( [\hat{h}(b)] \) is the area of \( \hat{h}(b) \), i.e., the fraction of \( h(b) \) which is visible. In addition, we only average the flow vectors which are contained in the visible cells of the model box which corresponds to \( u(v) \) (see figure 3, bottom-right).

### 3. Inference

We wish to find the MAP estimate of \( \omega \) as a good solution to the data association problem. In addition, we need to infer the camera parameters \( C \), and the association prior parameters \( \gamma = (\kappa, \theta, \lambda_N) \), which we consider functions of the video. Hence, we seek a value \( (\omega, C, \gamma) \) that maximizes the posterior distribution

\[
p(\omega, C, \gamma \mid D) \propto p(\omega \mid \gamma)p(\gamma)p(C)p(D \mid \omega, C)
= p(\omega \mid \gamma)p(\kappa)p(\theta)p(\lambda_N)p(C)
\times \int p(D \mid z, \omega, C)p(z \mid \omega) \, dz,
\]

where the factors in the expression are given by equations 1, 2, 3, and 5. To search the space of associations and associated parameters we use Markov chain Monte Carlo (MCMC) sampling techniques. At each iteration, we use different moves to sample over each of three variable blocks, stopping when the posterior stops changing.

**Sampling association parameters.** Sampling \( \gamma \) is straightforward. The full conditional distributions of its components are easy to compute (and sample from), given the conditional independence properties of our model, e.g., \( p(\kappa \mid \theta, \lambda_N, \omega, C, D) = p(\kappa \mid \theta, \omega) \), with analogous equalities holding for the full conditionals of \( \theta \) and \( \lambda_N \). From this and the conjugate hyper-priors (see Section 2.1), we have that \( \kappa \mid \theta, \omega \sim \text{Gamma}(m + \alpha_\kappa, 1 + (T - 1)\theta + \beta_\kappa) \), \( \theta \mid \kappa, \omega \sim \text{Gamma}(e + d + \alpha_\theta, l + (T - 1)\kappa + \beta_\theta) \), and \( \lambda_N \mid \omega \sim \text{Gamma}(n + \alpha_\lambda, T + \beta_\lambda) \), where the Gamma distribution is parametrized by shape and rate in all cases.

**Sampling associations.** We use the Metropolis-Hastings (MH) algorithm to sample from \( p(\omega \mid \gamma, C, D) \), using an extension of the MCMCDA proposal mechanism [31, 8]. Let \( \omega \) be the current sample. We draw an association \( \omega' \) from the proposal distribution \( q(\cdot \mid \omega) \), which we accept or reject based on the MH acceptance probability

\[
\min \left( 1, \frac{p(\omega' \mid \gamma, C, D)q(\omega' \mid \omega)}{p(\omega \mid \gamma, C, D)q(\omega' \mid \omega)} \right).
\]

We use seven sampling moves to efficiently explore the space of associations, which are loosely based on the standard MCMCDA moves. At each MH iteration, we perform move \( j \) with probability \( q_m(j) \), where \( j \in \{1, \ldots, 7\} \). (birth is 1, death 2, etc.) In what follows, let \( \omega = \{\tau_0, \ldots, \tau_m\} \) be the current sample, and \( \omega' \) be the proposed association.

**Birth/death moves.** A frame, \( t_i \), is sampled uniformly, and the first detection \( \tau_{m+1} \) in the new-born track \( \tau_{m'} \) is sampled uniformly from the set of false alarms at time \( t_i \). We then decide whether to grow forward or backward in time with probability \( 1/2 \). Assuming forward growth: to grow to time \( t = t_i + 1 \), we fit a line through the bottom of the previous \( s \) boxes, extrapolate the position of the next box, and independently choose to append candidates at time \( t \) based on their squared distance from the predicted point (see Figure 5). If none of the detections from time \( t \) is assigned, we stop growing \( \tau_{m'} \) with probability \( c \); otherwise, we continue with \( t + 1 \). The new association is then set to
be \( \omega' = \omega \cup \{ \tau_m' \} \). To kill a track, we choose \( r \) uniformly from \( \{1, \ldots, m\} \), and let \( \omega' = \omega \setminus \{ \tau_r \} \).

**Extension/reduction moves.** For extension, we choose a track \( \tau_r \) uniformly. We then grow it forward or backward to produce \( \tau_{r'} \) using the procedure described for the birth move. For reduction, we pick a detection \( \tau_{r,j} \) uniformly from \( \{ \tau_{r,2}, \ldots, \tau_{r,t_r-1} \} \), choose a direction, and remove all detections from the track after (or before) \( \tau_{r,j} \). In both, the resulting association is \( \omega' = (\omega \setminus \{ \tau_r \}) \cup \{ \tau_{r'} \} \).

**Merge/split moves.** We replace the standard MCMCDCA merge and split moves with alternatives that exploit the fact that we allow tracks to contain multiple detections from a single frame. In the merge move, we assign a weight to each pair of tracks \( (\tau_{r'}, \tau_{r''}) \) proportional to the probability of birthing track \( \tau_{r'} \cup \tau_{r''} \), as described in the birth move above. We then choose a pair based on those probabilities, and the resulting track becomes \( \tau_r = \tau_{r'} \cup \tau_{r''} \). The proposed association then becomes \( \omega' = (\omega \setminus \{ \tau_{r'} \}) \cup \{ \tau_{r''} \} \). To split track \( \tau_r \), we first choose two frames \( t \) and \( t' \) uniformly, \( t < t' \). All detections before \( t \) go to \( \tau_{r,t} \), and all detections after \( t' \) go to \( \tau_{r',t} \). Each detection between \( t \) and \( t' \) go to either track with probability \( \frac{1}{2} \). The resulting association is \( \omega' = (\omega \setminus \{ \tau_r \}) \cup \{ \tau_{r,t}, \tau_{r',t} \} \).

**Switch move.** First select tracks \( r_1 \) and \( r_2 \) uniformly, and choose one detection from each track (with indices \( j \) and \( k \)) such that their locations are within a distance \( \tau \) times their temporal offset. Then, the detections after \( j \) in track \( r_1 \) and those before \( k \) in track \( r_2 \) are swapped. The proposed association is \( \omega' = (\omega \setminus \{ \tau_{r_1,1}, \tau_{r_2,1} \}) \cup \{ \tau_{r_1,2}, \tau_{r_2,2} \} \).

Once we sample \( \omega' \), we must evaluate its posterior (eq. 7), which contains an integral over \( z \) that corresponds to the marginal likelihood of \( \omega' \). Due to the camera projection, this likelihood cannot be performed analytically, nor can it be computed numerically, due to the high dimensionality of \( z \). Instead, we estimate the value of the integral using the Laplace-Metropolis approximation [17], which uses the fact that

\[
p(\mathcal{D} | \omega, C) = p(\mathcal{D} | z^*, \omega, C)p(z^* | \omega) / p(z^* | \mathcal{D}, \omega, C),
\]

where \( z^* = \arg \max p(\mathcal{D} | z, \omega, C)p(z | \omega) \). If we approximate the denominator with the Gaussian pdf, we get

\[
p(\mathcal{D} | \omega, C) \approx (2\pi)^{-\frac{D}{2}} |\mathbf{H}|^{-\frac{1}{2}} p(\mathcal{D} | z^*, \omega, C)p(z^* | \omega), \tag{9}
\]

where \( \mathbf{H} \) is the Hessian of \(-\log(p(\mathcal{D} | z, \omega, C)p(z | \omega))\) evaluated at \( z^* \), and \( D \) is the dimension of \( z \).

We estimate \( z^* \) using the Hybrid Monte Carlo (HMC) algorithm [29], using central finite differences to approximate the gradient of the posterior \( p(z | \mathcal{D}, \omega, C) \). We also use finite differences to approximate \( \mathbf{H} \) at \( z^* \). Unfortunately, the finite differences approximation requires too many evaluations of the posterior, an expensive calculation. To address this, we exploit the conditional independence that exists between frames in the likelihood, e.g., \( p(b, b' | z, \omega, C) = p(b | z, \omega, C)p(b' | z, \omega, C) \), in two ways. In the gradient computation, for example, updating a single dimension of \( z \) only affects a small number of boxes, whose likelihoods we can update independently of the rest. Conditional independence also means that most off-diagonal elements of \( \mathbf{H} \) are very close to 0, a fact which we exploit by only computing the finite differences on the diagonal.

**Sampling cameras.** We use HMC to sample from the camera posterior \( p(C | \gamma, \omega, B, I) \propto p(B, I | C, \omega)p(C) \), as this has proved effective in the task of camera estimation under a similar parametrization [12]. We use the same HMC implementation as that used to approximate \( z^* \) for eq. 9.

### 4. Data preparation and calibration

**Data.** For person detections, we used the readily available MATLAB implementation of the object detector developed by Felzenszwalb et al. [14], pre-trained for humans. We found that the detector missed well-defined smaller figures, mitigated by using double-sized images. For image data, we precomputed the dense optical flow of each frame using an existing software [26]. To speed up the computation of the average flow (\$2.3\$), we precompute the integral flow of each frame using integral images.

**Parameter calibration.** We manually annotated boxes for 47 videos from the DARPA Mind’s Eye Year One (MEdge) data set, by drawing tight bounding boxes around each target throughout the video. To calibrate relevant parameters of the generative model, we match each detection box to the ground truth box with which it has maximum overlapping area, provided it is greater than 50\%, otherwise it is counted as a false detection. Using this matching, we find reasonable values for \( \lambda_A \) and for the parameters of the likelihoods \( \phi_B \) and \( \phi_I \). For the former, we simply average number of detections associated to each ground truth box; we estimate the latter using a maximum likelihood approach.

---

1. [http://www.visint.org/datasets](http://www.visint.org/datasets)
(using the ground truth boxes). The remaining parameters are set manually.

**Initialization.** The sampler is initialized with an empty association \( \omega = \{ \} \), and a camera \( C \) which is fit to the data \( B \) under the box likelihood (eq. 3) using RANSAC [15].

5. Experiments and results

We tested our tracker on two widely-used data sets: the PETS 2009 data set\(^2\), and the TUD data set\(^3\). For PETS we tested on the S2L1 video, which has over 795 frames, and contains 19 pedestrians walking freely about a very large area. The TUD data set contains three videos, called campus, crossing, and Stadtmitte, with 71, 201, and 179 frames, respectively, featuring between 8 and 13 people walking across the screen, and which were taken with a very low camera angle, causing targets to be frequently occluded for long periods of time.

**Performance measures.** We use the CLEAR metrics [38] which consists of two measurements, multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP). MOTA is a measure of false positives, missed targets and track switches, and ranges from \(-\infty \) to 1, with 1 being a perfect score. MOTP measures the average distance between true and inferred trajectories, and ranges from 0 to the threshold at which tracks are said to correspond which, as per convention, we set to 1 meter.

We also use the evaluation proposed by Li et al. [25], of which we are using two metrics: mostly tracked (MT) and mostly lost (ML), We use a threshold of 80% for declaring a target mostly tracked.

**Experiments.** We report the results of running our tracker on PETS and TUD, as well as published results for other algorithms in Table 1. We also ran experiments designed to test the impact of the different parts of our model, in which we ran our tracker with certain aspects disabled. Here we used the relatively easy TUD-Campus video. The results for these experiments are in Table 2. Not surprisingly, the performance took the greatest blow when the tracker ignored optical flow features. These results also suggest that our handling of occlusion is also quite helpful, which supports our fully 3D approach.

![Figure 6. Visualization of some of our results: three frames of the PETS-S2L1 video with the 3D scene super-imposed.](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA</th>
<th>MOTP</th>
<th>MT</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PETS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td>0.83</td>
<td>0.8</td>
<td>0.67</td>
<td>0</td>
</tr>
<tr>
<td>Zamir [34]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Wu [41]</td>
<td>0.88</td>
<td>×</td>
<td>0.87</td>
<td>0.05</td>
</tr>
<tr>
<td>Andriyenko [5]</td>
<td>0.96</td>
<td>0.78</td>
<td>0.96</td>
<td>0</td>
</tr>
<tr>
<td>Andriyenko [5]</td>
<td>0.88</td>
<td>0.78</td>
<td>0.87</td>
<td>0.05</td>
</tr>
<tr>
<td><strong>TUD-X</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td>0.80</td>
<td>0.78</td>
<td>0.69</td>
<td>0.08</td>
</tr>
<tr>
<td>Zamir [34]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td><strong>TUD-S</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td>0.70</td>
<td>0.73</td>
<td>0.7</td>
<td>0</td>
</tr>
<tr>
<td>Zamir [34]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Andriyenko [5]</td>
<td>0.62</td>
<td>0.63</td>
<td>0.67</td>
<td>0</td>
</tr>
<tr>
<td>Andriyenko [5]</td>
<td>0.69</td>
<td>0.68</td>
<td>0.67</td>
<td>0</td>
</tr>
<tr>
<td>Andriyenko [5]</td>
<td>0.88</td>
<td>0.88</td>
<td>0.85</td>
<td>0</td>
</tr>
<tr>
<td><strong>TUD-C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td>0.84</td>
<td>0.81</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>Yan [42]</td>
<td>0.85</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 1. Comparison of performance of our approach and several state-of-the-art algorithms on the PETS and TUD (campus, crossing, and Stadtmitte, labeled TUD-C, TUD-X, TUD-S, resp.) data sets using the CLEAR metrics, as well as those proposed in [25]. We report MOTP as normalized distance, and use \( \times \) for values not reported, or reported in 2D.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA</th>
<th>MOTP</th>
<th>MT</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>0.84</td>
<td>0.81</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>NO-OF</td>
<td>0.59</td>
<td>0.79</td>
<td>0.38</td>
<td>0.25</td>
</tr>
<tr>
<td>NO-OCC</td>
<td>0.73</td>
<td>0.81</td>
<td>0.62</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 2. A summary of the effect of removing key features of our tracker. “Base” is our full algorithm, “NO-OF” ignores optical flow features, and NO-OCC does not reason about occlusion.

6. Discussion

We presented a tracker which incorporates representations for data association and 3D scene in a principled way. Across all data sets and all measures our method is comparable to the state-of-the-art. Since our approach is Bayesian and expandable, we expect performance will improve as it matures. In addition, our algorithm is easily parallelizable. We emphasize that we are learning more about the scene than other approaches typically do. In particular, we infer the camera and sizes of the tracked persons. We expect that further modeling improvements will similarly lead to better tracking and inferring more about the scene.

7. Acknowledgments

This material is based upon work supported in part by the DARPA Mind’s Eye program, and by the National Science Foundation under Grant No. IIS-0747511.

References


\(^2\)http://www.cvg.rdg.ac.uk/PETS2009/a.html

\(^3\)https://www.d2.mpi-inf.mpg.de/node/382


S. Kwak, W. Nam, B. Han, and J. H. Han. Learning occlusion with likelihoods for visual tracking. ECCV, 2011.


R. Mohedano and N. Garcia. Simultaneous 3d object tracking and camera parameter estimation by bayesian methods and transdimensional mcmc sampling. In ICIP, 2011.


