A Generative Statistical Model for Tracking Multiple Smooth Trajectories

Ernesto Brau[†]
ernesto@cs.arizona.edu

Kobus Barnard[†]
kobus@cs.arizona.edu

Ravi Palanivelu[‡]
rpalaniv@ag.arizona.edu

Damayanthi Dunatunga[‡]
damayanthidunatunga@yahoo.com

Tatsuya Tsukamoto[‡]
tsukamot@email.arizona.edu

Philip Lee[†]
phlee@email.arizona.edu

[†]Computer Science
University of Arizona

[‡]Plant Sciences
University of Arizona

Abstract

We present a general model for tracking smooth trajectories of multiple targets in complex data sets, where tracks potentially cross each other many times. As the number of overlapping trajectories grows, exploiting smoothness becomes increasingly important to disambiguate the association of successive points. However, in many important problems an effective parametric model for the trajectories does not exist. Hence we propose modeling trajectories as independent realizations of Gaussian processes with kernel functions which allow for arbitrary smooth motion. Our generative statistical model accounts for the data as coming from an unknown number of such processes, together with expectations for noise points and the probability that points are missing.

For inference we compare two methods: A modified version of the Markov chain Monte Carlo data association (MCMCDA) method, and a Gibbs sampling method which is much simpler and faster, and gives better results by being able to search the solution space more efficiently. In both cases, we compare our results against the smoothing provided by linear dynamical systems (LDS).

We test our approach on videos of birds and fish, and on 82 image sequences of pollen tubes growing in a petri dish, each with up to 60 tubes with multiple crossings. We achieve 93% accuracy on image sequences with up to ten trajectories (35 sequences) and 88% accuracy when there are more than ten (42 sequences). This performance surpasses that of using an LDS motion model, and far exceeds a simple heuristic tracker.

1. Introduction

Tracking multiple targets is very important in a number of domains, including surveillance, virtual reality, and monitoring biological growth and behavior. We are particularly interested in tracking biological tip growth as observed in tubes growing from pollen grains (see Figure 1(a)). Other relevant examples which have drawn the attention of the machine vision community include seedling root growth [14], hypocotyl growth [24], and neuron growth [3].

In all these examples, there is a pressing need to quantify the trajectories automatically from image data. Typically all that can be assumed about the trajectories are that they are relatively smooth. This makes tracking very challenging when there are many trajectories that cross many times in the captured images, further compounded by the presence of noise (false points) and missing data.

Multiple-target tracking is a well-studied problem with many approaches. Classical approaches include the multiple hypothesis tracker (MHT) [7, 19] and the joint probabilistic data association (JPDA) filter [5]. More recently, MCMC approaches have been proposed such as Markov chain Monte Carlo data association (MCMCDA) [16] and the MCMC-based particle filter [12]. In MCMCDA, the tracking problem is solved by sampling over the space of associations of points to tracks using the Metropolis Hastings algorithm (MH). The MCMC-based particle filter also uses the MH algorithm to generate samples (particles) for the posterior at each time step. Finally, a data association approach is also used in [11], where they use Fourier inference over permutations to determine the associations. Notice, however, that all these approaches rely on motion models that are application-specific. This implies considering different dynamic models for different applications. Our goal is to provide a more general model that can be used in any application as long as the motion is smooth.

The traditional model for smooth motion is the linear-Gaussian model – also known as the linear dynamical system (LDS). While LDS models linear dynamics with Gaussian noise very well, it is not well suited for motion which

exhibits significantly more erratic smooth behavior. Furthermore, in order to attain maximum flexibility, one is required to set many parameters. On the other hand, Gaussian processes only require a single scale parameter, assuming a chosen kernel function. Another problem with LDS models is that the state at a given time is conditionally independent of previous states given the immediately preceding state. Our model does not make this assumption, making it more flexible and able to fit a wider range of motions. We note that Gaussian processes have been used previously in tracking [10, 22, 23], but differently than in this work.

We consider the following set up. An unknown number of indistinguishable objects move independently according to some movement model with their positions being measured at known intervals. The measured positions have error, and there is noise in the background which leads to false detections. In addition, some measurements will typically be missed. Crossing occurs because the trajectories occur in 3D space, but images only provide 2D data. For our main application (pollen tubes), there is only a small amount of depth variation – the arena is close to 2D – but there is sufficient depth that trajectories often cross. In fact, we have a modest amount of 3D information available because we have multiple images taken at roughly the same time but at different focal planes. However, the amount of ambiguity that this resolves is minimal because the depth of focus is of the order of the depth of the object being tracked. Nonetheless, to make use of the cue when it is helpful, we set up the tracking problem in 3D space. Exploiting image stack data in this way simply requires adding a third coordinate to the problem, and this work applies equally well to tracking with and without estimating depth.

As mentioned earlier, a key component of this work is to address smooth motion using a Gaussian process where the correlation of object position with preceding and subsequent positions drops of exponentially with time. We propose that this is an excellent way to translate the notion of smooth, but otherwise arbitrary, motion into a generative statistical model for the trajectories. Trajectories are also characterized by weak priors on velocity, as well as their likelihood of starting (birth) and ending (death). In addition, to effectively determine the number of tracks, we also need estimates for the noise density and the probability of missing data. We find that these parameters are sufficient to effectively model the observations associated with multiple, complex, and overlapping trajectories.

To determine the trajectories we adopt the data association formulation of tracking [16] and use MCMC sampling to explore the solution space. The generative statistical model is constructed to allow principled comparisons between hypothesis involving differing numbers of tracks. To effectively explore the solution space we adopt and extend the sampling moves from Oh *et al*. [16]. In particular, we consider moves for track birth, death, merge, split, extension, reduction, switch and update. We also present a Gibbs sampling approach to compute the associations. This method has the advantage of being simpler and faster when compared to MCMCDA, while giving similar results.

We apply our method to several kinds of smooth motion, notably tracking linear structures arising from biological tip growth, and focusing on pollen tubes growing on a petri dish. As in many other possible applications of our method, the goal is to automatically quantify trajectories in images acquired in high-throughput experiments for identifying mutant behavior and/or testing the effect of multiple environmental conditions.

Scientific motivation for tracking pollen tubes. In order for seed plants to reproduce, the male sperm cells contained in pollen grains must be transferred to the female egg cell. Upon arrival at the surface of a flower pistil, pollen grains absorb water from the stigma and germinate to produce pollen tubes, which transport their cellular contents, along with two sperm cells, to the ovule of the plant [17]. Pollen tubes grow exclusively at their tip [21], traveling through the pistil (female organ within the plant) until they reach the ovary and enter the ovule through an opening, called the micropyle [13]. A pollen tube's journey terminates at one of two synergid cells inside the ovule, where it bursts to release sperm cells. Pollen tubes are able to reach ovules due to the guidance provided by diffusible signals from the ovules. Characterizing and quantifying the interaction between pollen tubes and ovules is crucial to achieve a better understanding of the seed formation process, and could have a very broad impact. For instance, it could lead to a better understanding of how plants regulate fertilization and avoid spurious fertilization events, which could be used to prevent pollen spread from genetically modified crops into native species.

One current difficulty in characterizing this interaction is that a human must watch time-lapse images of pollen tubes growing *in vitro* (such as those in Figure 1(a)), identify each tube as it grows and recognize its behavior; *e.g*., which pollen tubes were targeting which ovules, which successfully entered an ovule, and which were repelled. This process is slow and time-consuming. The first step in automating this operation is to robustly find the paths of the pollen tubes in the images. Taking the image difference of consecutive time-lapse images provides the noisy tips of the pollen tubes at all time points, as well as plenty of background noise. Since pollen tubes exhibit smooth growth, we can apply our tracking algorithm, thereby taking a step towards modeling pollen tube growth behavior and interactions with ovules that affect tube trajectories through complex and poorly understood signaling.

Reproducibility. An implementation of our tracker [2] and associated data [1] are available on-line.

2. Gaussian Processes as Smooth Functions

Our goal is to model smooth movement using Gaussian processes. To do this, we shall formulate a dual interpretation of (realizations of) Gaussian processes as parametric functions evaluated at indices. We shall then interpret the motion of objects as generated by a Gaussian process.

A Gaussian process is a stochastic process whose finite-dimensional distributions are normally distributed [20]. We will denote a Gaussian process by $\{f_x\}_{x\in\mathbb{R}^d}$, with $f_x \in \mathbb{R}^{d'}$, for all $x \in \mathbb{R}^d$. Here $\mathbb{R}^d$ is called the index set and the $x \in \mathbb{R}^d$ are called indices. A Gaussian process is completely determined by a mean and covariance function, $m : \mathbb{R}^d \to \mathbb{R}^{d'}$ and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d'\times d'}$, given by

$$m(x) = \mathbb{E}[f_x], \tag{1}$$

$$k(x, x') = \mathrm{cov}(f_x, f_{x'}), \tag{2}$$

for all indices $x, x' \in \mathbb{R}^d$. This means that for any finite subset of the Gaussian process $\mathbf{f} = (f_1, \ldots, f_T)^\mathsf{T}$, where we use the short-hand notation $f_t = f_{x_t}$, with $x_1, \ldots, x_T \in \mathbb{R}^d$, we have that

$$\mathbf{f} \sim \mathcal{N}(m(X), K(X, X)), \tag{3}$$

with $X = (x_1, \ldots, x_T)^\mathsf{T}$, $m(X) = (m(x_1), \ldots, m(x_T))^\mathsf{T}$, and $K(X, X)$ is the covariance matrix and is given by $(k(x_i, x_j))_{ij}, 1 \le i, j \le T$.

Now suppose we have a collection of T points $\{y_1, \ldots, y_T\} \subset \mathbb{R}^{d'}$, through which we wish to fit a curve. We assume that these data points were generated by a continuous function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ evaluated at every point in a set $X = \{x_1, \ldots, x_T\}$, so that

$$y_t = f(x_t) + \varepsilon, \tag{4}$$

for $t = 1 \ldots T$, where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ and σ^2 is the noise variance. We can model this data using Gaussian processes [18] by assuming that the function values $f(x_1), \ldots, f(x_T)$ that generated the data $\mathbf{y} = (y_1, \ldots, y_T)^\mathsf{T}$ are a finite subset of the realization of a Gaussian process $\{f_x\}_{x\in\mathbb{R}^d}$, with $f_{x_t} = f(x_t)$. This is particularly useful for us because Gaussian processes with a certain covariance function model smooth functions very well. This covariance function is the *squared exponential function*, given by

$$k(x, x') = \exp\Big(-\frac{1}{2l^2}||x - x'||^2\Big) I, \tag{5}$$

where l is a scale parameter and I is a $d' \times d'$ identity matrix. Loosely speaking, this covariance function provides a distribution over all functions from $\mathbb{R}^d \to \mathbb{R}^{d'}$ [18], with smooth functions being more likely than others. We also use the zero mean function $m(x) = \mathbf{0}$. Using this model, we can compute many quantities of interest, such as the MAP estimate of $\mathbf{f}$ and the marginal likelihood of the data $p(\mathbf{y})$.

2.1. The Posterior

As mentioned above, we model these smooth tracks using Gaussian processes. Naturally, knowing the measurement-track correspondences isn't enough; we wish to estimate the actual positions of the objects. One way to do this is by maximizing the posterior distribution of the actual locations given the data. Under the general regression setup described above, we we are looking for the $\mathbf{f}$ that maximizes $p(\mathbf{f} \,|\, \mathbf{y}, X)$, which, by Bayes' rule, is given by

$$p(\mathbf{f} \,|\, \mathbf{y}, X) = \frac{p(\mathbf{y} \,|\, \mathbf{f}, X) p(\mathbf{f} \,|\, X)}{p(\mathbf{y} \,|\, X)}. \tag{6}$$

Since all the densities in the right side of this equation are normal (by equations (3), (4) and (9)), we can complete the square to compute the posterior, yielding

$$\mathbf{f} \,|\, \mathbf{y}, X \sim \mathcal{N}(\sigma^{-2}\Sigma\mathbf{y}, \Sigma), \tag{7}$$

with $\Sigma = (K(X, X)^{-1} + \sigma^{-2} I)^{-1}$.

2.2. The Marginal Likelihood

The marginal likelihood of the data $p(\mathbf{y} \,|\, X)$, is given by

$$p(\mathbf{y} \,|\, X) = \int p(\mathbf{y} \,|\, \mathbf{f}, X) p(\mathbf{f} \,|\, X)\, d\mathbf{f}. \tag{8}$$

Here, the prior distribution $p(\mathbf{f} \,|\, X)$ is given by (3), and $\mathbf{y} \,|\, \mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$ by (4). After a simple calculation, we conclude that

$$\mathbf{y} \,|\, X \sim \mathcal{N}(\mathbf{0}, K(X, X) + \sigma^2 I). \tag{9}$$

As we shall see below, this quantity will turn out to play a very important role when using this model in conjunction with our tracking algorithms.

3. Tracking Multiple Targets

The multiple-target tracking problem is posed as follows [16]. Assume that we have K objects moving in a region of interest (normalized to unit volume) for a finite period of time. Let $f_{k\,t}$ be the location of object k at discrete time t, $k = 1, \ldots, K$ and $t = 1, \ldots, T$. At each of these discrete time-steps t, the following happens. (1) A Poisson-distributed number of objects arises (i.e., moves for the first time), with parameter λ_b. (2) Each existing object from time $t-1$ disappears with probability p_z and persists with probability $1 - p_z$. (3) Each persisting object k move from position $f_{k\,t-1}$ to $f_{k\,t}$, and is detected with probability p_d as $f_{k\,t} + \varepsilon$, where ε is additive noise. (4) A Poisson-distributed number of noise detections (called "false alarms"), with parameter λ_f, are detected. The multiple-target problem consists of estimating K and $\{f_{k\,t_{ki}}, \ldots, f_{k\,t_{kf}}\}$, for $k = 1, \ldots, K$, where $f_{k\,t_{ki}}$

and $f_{k\,t_{kf}}$ are the initial and final positions of target k, respectively.

As already mentioned, targets are detected as they move about the region of interest. Let $Y_t = \{y_{t\,j} : j = 1, \ldots, n_t\}$ be all the detections at time t, and $Y = \bigcup_{t=1}^{T} Y_t$ be the set of all measurements. An association [16] $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$ is a partition of Y, such that (1) $|\tau_k \cap Y_t| \leq 1$ for $k = 1, \ldots, K$ and $t = 1, \ldots, T$ and (2) $|\tau_k| \geq 2$ for $k = 1, \ldots, K$. Under this representation, $\tau_1, \ldots, \tau_K$ represent tracks, i.e., trajectories of objects, and τ_0 is the set of false alarms.

We now find the association $\hat{\omega}$ that maximizes the posterior distribution which, by Bayes' rule, is

$$p(\hat{\omega} \,|\, Y) \propto p(Y \,|\, \hat{\omega}) p(\hat{\omega}), \tag{10}$$

where $p(Y \,|\, \omega)$ is the likelihood function, discussed below. The prior distribution is

$$p(\omega) = \prod_{t=1}^{T} p_z^{z_t} (1 - p_z)^{c_t} p_d^{d_t} (1 - p_d)^{g_t} \frac{\lambda_b^{a_t} \lambda_f^{f_t} e^{-\lambda_b - \lambda_f}}{a_t! f_t!}, \tag{11}$$

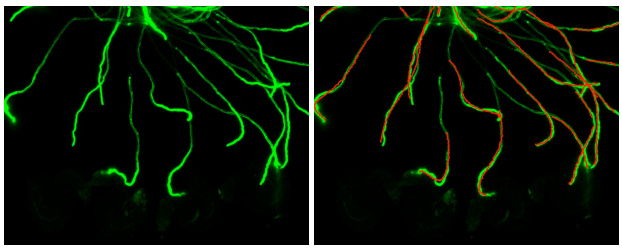
where z_t is the number of objects disappearing at time t, c_t the number of targets that persisted from time $t-1$, d_t the number of detected targets at time t, a_t the number of new targets, g_t those that were undetected and, finally, f_t the number of false alarms. Notice that all these quantities can be easily derived from ω.

4. Estimating the Tracks

As discussed in section 3, our objective is to find an association ω that maximizes the posterior distribution (10). To do this, we assume that the data consists of noisy measurements of the values of the realized random variables of a Gaussian process in $\mathbb{R}^2$ or $\mathbb{R}^3$ – depending on the dimensionality of the data – with indices in $\mathbb{R}$ (i.e., $d' = 2$ or 3 and $d = 1$ using the notation from section 2). Given an association $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$, every track $\tau_k = (\tau_k(1), \ldots, \tau_k(|\tau_k|))^{\mathsf{T}}$ (where $\tau_k(i)$ is the measurement corresponding to the ith observation of track τ_k), $k \neq 0$, has a corresponding vector of function values $\mathbf{f}_k = (f_{k1}, \ldots f_{k|\tau_k|})^{\mathsf{T}}$. A track also has a corresponding set of indices $X_k = \{t_1, \ldots, t_{|\tau_k|}\}$, such that for all $t_m \in X_k$, $\tau_k(m) = y_{t_m\,j} = f_{km} + \varepsilon$, for some $j \in \{1, \ldots, n_{t_m}\}$, where ε is additive Gaussian noise, as before.

We can now compute the likelihood $p(Y \,|\, \omega)$. Since all the tracks are independent of each other, we can decompose the likelihood into factors:

$$p(Y \,|\, \omega) = p(\tau_0 \,|\, \omega) \prod_{k=1}^{K} p(\tau_k \,|\, \omega). \tag{12}$$



(a) Data image (b) Superimposed tracks

Figure 1: (a) Frame 16 (of 33) of one of the pollen tube image sequences. (b) The tracks found by MCMCDA with the Gaussian process model superimposed on the same image.

We must now compute the individual likelihoods $p(\tau_k \,|\, \omega)$ for each track. Clearly, τ_k only depends on the part of ω that dictates which measurements it is associated to; this is simply X_k. This means that $p(\tau_k \,|\, \omega) = p(\tau_k \,|\, X_k)$, which is the marginal likelihood of τ_k. In the case of the Gaussian process model, this is simply a Gaussian distribution, i.e., $\tau_k \,|\, X_k \sim \mathcal{N}(\mathbf{0}, K(X_k, X_k) + \sigma^2 I)$. For an LDS, the marginal likelihood can be computed iteratively using the Kalman Filter [6]. Finally, for the distribution of false alarms, $y_f \in \tau_0$, we use a uniform distribution, i.e., $p(y_f) = r$. Then, $p(\tau_0 \,|\, \omega) = r^{|\tau_0|}$.

4.1. Data Association

Given any association $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$, we can now easily compute the posterior probability $p(\omega \,|\, Y)$ which, for convenience, we will call $\pi(\omega)$. We are still left with the problem of how to find a good estimate of the optimal – in the MAP sense – association. To solve this problem, we have resorted to Markov chain Monte Carlo (MCMC) sampling [15]. Specifically, we use the Metropolis-Hastings algorithm to sample from the posterior distribution $\pi(\omega)$ to search for an association that is close to the maximum. This algorithm works by building a reversible, ergodic Markov chain with invariant distribution $\pi(\omega)$ [9]. At each iteration, it proposes a new state of the chain ω' from the current state ω according to a proposal distribution $q(\omega' \,|\, \omega)$. The proposed state ω' is then accepted with probability

$$A(\omega, \omega') = \min\left\{1, \frac{\pi(\omega') q(\omega \,|\, \omega')}{\pi(\omega) q(\omega' \,|\, \omega)}\right\}. \tag{13}$$

If ω' is accepted, it becomes the current state in the chain; if it is rejected, ω is the new state. It can be shown that, if the kernel of this Markov chain (given by $q(\omega' \,|\, \omega) A(\omega, \omega')$) satisfies the detailed balance condition [15], then $\pi(\omega)$ is the stationary distribution of the chain and that the chain will converge to it.

Our proposal distribution $q(\omega' \,|\, \omega)$ is a modification of the one introduced in existing work on MCMC data association [16]. It consists of five different types of moves:

birth/death moves, merge/split moves, extension/reduction moves, update moves and switch moves. It is important to note that these moves are such that they always create valid tracks; in other words, the proposed association always meets the conditions discussed in section 3. In what follows, $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$ is the current state of the Markov chain, and ω' is the proposed state.

Birth/death moves. In a birth move, a new association ω' is proposed by adding a new track to the ω. The resulting association takes the form $\omega' = \{\tau_0 \backslash \tau_{K+1}, \tau_1, \ldots, \tau_K, \tau_{K+1}\}$. A death move consists of uniformly at random selecting a track τ_k from ω and converting its observations into noise, so that $\omega' = \{\tau_0 \cup \{\tau_k\}, \tau_1, \ldots, \tau_{k-1}, \tau_{k+1}, \ldots, \tau_K\}$.

Merge/split moves. In a merge move, two tracks τ_{k_1} and τ_{k_2} are randomly selected from ω and are merged, so that $\omega' = (\omega \backslash \{\tau_{k_1}, \tau_{k_2}\}) \cup \tau_{k'}$, where $\tau_{k'} = \{\tau_{k_1}(1), \ldots, \tau_{k_1}(|\tau_{k_1}|), \tau_{k_2}(1), \ldots, \tau_{k_2}(|\tau_k|)\}$. In a split move, a track τ_k is chosen at random and split (at a random point t) into two different tracks τ_{k_1} and τk_2, yielding $\omega' = (\omega \backslash \{\tau_k\}) \cup \{\tau_{k_1}, \tau_{k_2}\}$, with $\tau_{k_1} = \{\tau_k(1), \ldots, \tau_k(t)\}$ and $\tau_{k_2} = \{\tau_k(t+1) \ \ldots, \tau_k(|\tau_k|)\}$.

Extension/reduction moves. In an extension move, a track τ_k is chosen at random and is assigned observations before its appearance time or after its disappearance time (a decision taken uniformly at random). In the former case, if n observations are added, the result is another track $\tau_{k'} = \{y_1^*, \ldots, y_n^*, \tau_k(1), \ldots, \tau_k(|\tau_k|)\}$. In a reduction move, a track τ_k is selected and shortened by removing observations before or after a randomly selected point t, yielding $\tau_{k'} = \{\tau_k(t), \ldots, \tau_k(|\tau_k|)\}$ or $\tau_{k'} = \{\tau_k(1), \ldots, \tau_k(t)\}$, respectively.

Update move. We select a random track τ_k and a time point $1 \leq t \leq |\tau_k|$, and assign observations after t, producing $\tau_{k'}$. The resulting association is $\omega' = (\omega \backslash \{\tau_k\}) \cup \{\tau_{k'}\}$.

Switch move. A pair of observations $\tau_{k_1}(t_1)$ and $\tau_{k_2}(t_2)$ are selected from different tracks, which become $\tau_{k_1} = \{\tau_{k_1}(1), \ldots, \tau_{k_1}(t_1), \tau_{k_2}(t_2{+}1), \ldots, \tau_{k_2}(|\tau_{k_2}|)\}$ and $\tau_{k_2} = \{\tau_{k_2}(1), \ldots, \tau_{k_2}(t_2), \tau_{k_1}(t_1+1), \ldots, \tau_{k_1}(|\tau_{k_1}|)\}$.

The main modification we made to the moves described above is allowing for growth backward in time. For example, in the extension move, we allow the possibility of adding observations to a track before its start point, rather than only after its end point. A proof of the reversibility of the original proposal distribution is given in [16], which shows that the moves presented always generate valid tracks. Our modifications also maintain this invariant, and therefore the modified Markov chain is reversible.

4.2. Gibbs Sampler

Although MCMCDA gives good results, it has one major disadvantage in that it has a very high rejection rate, especially in the switch and update moves. For these two particular moves, we observed rejection rates of up to 90%. Motivated by this, we developed a Gibbs sampling approach [4] to maximizing the posterior $P(\omega \,|\, Y)$. Since Gibbs samplers never reject, we were able to find comparable solutions, and often better ones, in a shorter running time.

The Gibbs sampling algorithm is a special case of the Metropolis-Hastings where the proposal distribution is equal to the full-conditional posterior distribution. That is, in general, if our goal is to sample from target distribution $\pi(x)$, our proposal is $q(x' \,|\, x) = \pi(x'_k \,|\, x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)$, where the x_k are the elements of the random vector x. In this way, the acceptance probability (13) becomes 1 and, thus, we accept all generated samples [4, 6, 15].

Evidently, to use this technique we must be able to represent our target variable as a random vector and be able to sample from the full-conditional posterior. To do so, we must reinterpret our data-association problem in a slightly different but equivalent way. We shall now represent an association by a vector of assignment variables $\alpha_{t\,j} \in \{0, \ldots, K\}$, $t = 1, \ldots, T$, $j = 1, \ldots, n_t$, which represent the track (including the noise track) to which $y_{t\,j}$ is assigned. We shall denote the full random variable by $\alpha = (\alpha_{1\,1}, \ldots, \alpha_{1\,n_t}, \ldots, \alpha_{T\,n_T})$. This is clearly equivalent to the previous representation of an association.

We now show how to sample from $p(\alpha_{t\,j} \,|\, \alpha_{\backslash t\,j}, Y)$, where $\alpha_{\backslash t\,j} = \alpha \backslash \{\alpha_{t\,j}\}$. First, let us decompose $p(\alpha_{t\,j} \,|\, \alpha_{\backslash t\,j}, Y)$ into

$$p(\alpha_{t\,j} \,|\, \alpha_{\backslash t\,j}, Y) \propto p(Y \,|\, \alpha) p(\alpha_{t\,j} \,|\, \alpha_{\backslash t\,j}), \qquad (14)$$

where we have used the fact that $p(Y \,|\, \alpha_{\backslash t\,j})$ is a constant in the expression. Similarly we have that $p(\alpha_{\backslash t\,j}) \propto p(\alpha)$. This means that we can sample from $p(\alpha_{t\,j} \,|\, \alpha_{\backslash t\,j}, Y)$ by sampling from the conditional prior $p(\alpha_{t\,j} \,|\, \alpha_{\backslash t\,j})$ and scaling by the prior $p(\alpha)$ and the likelihood $p(Y \,|\, \alpha)$ and normalizing. Finally, sampling from the conditional prior is straight forward using the model described in section 3.

5. Experiments and Results

We compare the Gaussian-process-based motion model to LDS. Here we used a typical model [6, 8] where targets move according to

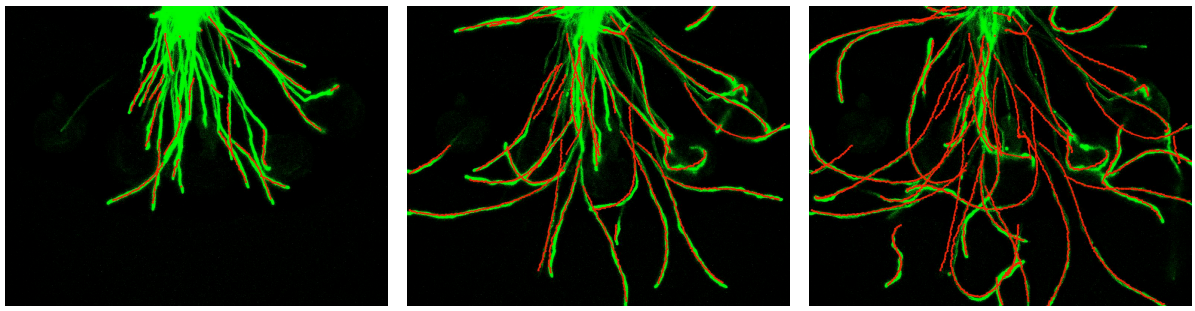
$$f_t = A f_{t-1} + w, \qquad (15)$$

and the measurements are obtained according to

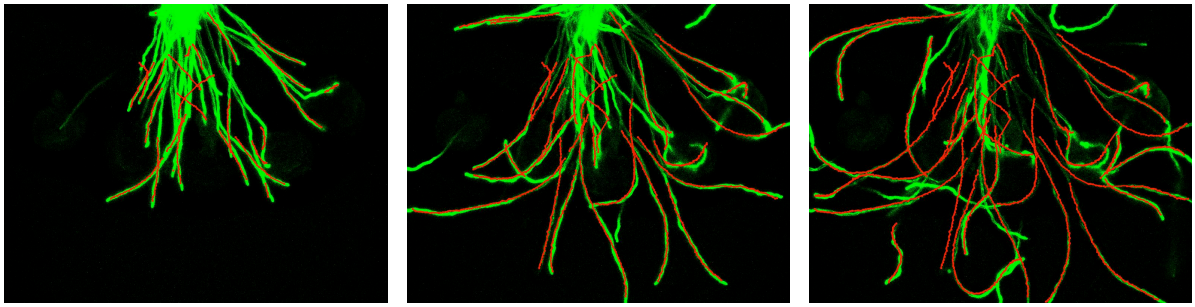
$$y_t = H f_t + v, \qquad (16)$$

where, in the case where the motion occurs in $\mathbb{R}^2$, A and H are given by

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$



(a) GDA-GP; $t = 5$ (b) GDA-GP; $t = 15$ (c) GDA-GP; $t = 30$



(d) GDA-LDS; $t = 5$ (e) GDA-LDS; $t = 15$ (f) GDA-LDS; $t = 30$

Figure 2: An example of a sequence from the difficult set, this association has 60 tracks, of which GDA-GP found 53, all almost completely correct. By contrast, GDA-LDS only found 35 most of which were not completely on target, and some of which were off by a wide margin.

where Δt is the time elapsed between x_t and x_{t-1} (recall that $f_t = f(x_t)$), and w and v are Gaussian random variables. Here, f_t is the augmented state that contains the position as well as the velocity.

5.1. Image Processing and Ground Truth

We have posed our tracking problem as a data association problem, whose input is sets of points at different time steps. Here we briefly describe the way in which we have extracted points from images. For targets that exhibit tip-growth (such as pollen tubes), we first perform image subtraction from one frame to the next, a process which provides us with the tips of the targets at each time step. We then do a simple background subtraction by removing the average of all images from each frame. We subsequently threshold the images and find the 8-connected components (which we call blobs) in the resulting binary images. Finally, we find the blob centroids, which give us the points which are the input to our algorithms. Note that, any blob detector may be used for this step without any change to the tracking algorithm itself.

For the purposes of accurately evaluating our approach, we obtained the ground truth associations for all of our data sets. We did this by manually assigning the detected blobs to the correct tracks using a point-and-click tool we developed. Note that the ground truth is with respect to the detected blobs, and not to the actual targets themselves, which means that their quality is restricted by the quality of the blob detector. We discuss this further below.

5.2. Evaluation

We evaluated our approach in three ways. The first, which we call CTP (correct track percentage) is the percentage of track assignments that were correct. More precisely, if τ_k and τ'_k are respectively corresponding real and computed tracks, CTP is given by

$$\text{CTP} = \sum_{k=1}^{K} \frac{|\tau_k \cap \tau'_k|}{|\tau_k|}, \tag{17}$$

where we assign each estimated track to the ground truth track that it shares the most points with.

Due to the way we obtained the ground truth associations, this measure can be somewhat misleading. For example, an accuracy of 100% does not necessarily mean that all targets were tracked correctly, but that all of the detected blobs were assigned to the track that generated them. Nonetheless, since we have posed the problem as a data association problem, computing CTP is important as what is measures is very close to what by data association is trying to optimize.

A second minor issue with CTP is that in some cases a single target produces more than one measurement for a time point, and our algorithm may choose a different one

	Easy Set			Hard Set		
	CTP	CD	K_{err}	CTP	CD	K_{err}
MCMCDA-GP	0.87	0.05	0.31	0.79	0.05	1.05
MCMCDA-LDS	0.71	0.09	0.52	0.47	0.12	4.03
GDA-GP	0.93	0.01	0.22	0.88	0.05	2.01
GDA-LDS	0.74	0.07	0.61	0.59	0.09	3.00
Heuristic	0.51	0.28	1.83	0.13	0.27	9.03

Table 1: A comparison between the performance obtained using the Gaussian process model and that of the LDS model. In the table above, MCMCDA-GP is the MCMCDA algorithm using the GP model and MCMCDA-LDS uses the LDS model. The analogous terminology was used for GDA. We show values for CTP (correct track percentage, bigger is better), CD (curve distance, smaller is better), and K_{err} (smaller is better). All values in the tables are averages over the set of image sequences in each data set. See text for details on these measures. In general, the Gaussian process model outperforms LDS using both trackers, and GDA-GP tracker gives the best results overall.

than was chosen by the manual assignment. In other cases an estimated track misses a blob but subsequently continues along the correct path, making it almost indistinguishable from the real track. This has a negative impact on the evaluation since both choices can be considered correct.

These considerations suggest a second evaluation based on the similarity of two tracks when a curve is fit through its points (called CD for curve distance). Specifically, we fit a function to each track τ using the Gaussian process predictive distribution, which gives us $f_\tau(t)$, for any value of $t \in \mathbb{R}$. We discretize this function to get L points $f_\tau^{(1)}, \ldots, f_\tau^{(L)}$, and compare two corresponding tracks by

$$C_k = \sum_{l=1}^{L} \frac{(f_{\tau_k}^{(l)} - f_{\tau'_k}^{(l)})^2}{l}, \tag{18}$$

where l is the length of the shortest of the two tracks. The CD error is then calculated as $\text{CD} = \frac{1}{K}\sum_{k=1}^{K} C_k$, and is normalized to the size of the image.

For the last evaluation, we compute the average error committed in the estimate of the number of tracks in an image sequence. In other words, if ω and ω' are a ground truth and a computed association (respectively), the error for that image sequence is simply $K_{\text{err}} = \big||w| - |w'|\big|$.

	CTP	CD	K_{err}
GDA-GP	0.89	0.10	1.33
GDA-LDS	0.77	0.11	2.16

Table 2: Results of applying our Gibbs tracker to four videos of schools of fish and four videos of flocks of birds. GDA-GP outperforms GDA-LDS even though the motion of these targets can be well modeled in 3D by an LDS.

5.3. Pollen Tubes

As mentioned earlier, we have a set of 82 image sequences of pollen tubes growing in vitro. We tested our model against LDS on this data using two different tracking approaches: MCMCDA and the Gibbs sampler (both described in section 4), which we shall call GDA. We also implemented a very simple heuristic algorithm that greedily chooses assignments for tracks based on the likelihood function (12).

The results can be seen in Table 1. Note that we separated the data into two disjoint groups: one where the number of pollen tubes is ten or less, called the "easy" set, and the other where the number is larger than ten, called the "hard" set. The easy set has 35 test cases and the hard set has 47. An example of a successfully tracked set of tubes is shown in figures 1 and 2.

Note that there are two levels of comparison to be made. The main one is the difference in performance when using the LDS model of motion against the much more general GP model. In principle, any tracking algorithm can be used for this purpose. With this in mind, it is clear that our smooth model is far better suited to handle pollen tube data.

Turning our our attention to inference, the table shows that GDA is working better than MCMCDA. Since these methods have the same objective function, we conclude that the Gibbs sampler is more efficiently able to explore the solution space.

5.4. Birds and Fish

To further test our approach as a general method for tracking multiple smooth trajectories, we tested our Gibbs sampling method on on two other types of data sets, namely four videos of flocks of birds and four videos of schools of fish found on the Internet. Figure 3 shows two frames from a bird video with the tracks found, and Table 2 shows results using GDA-GP and GDA-LDS, averaged over the eight videos.

Fish and birds largely exhibit linear motion, which is exactly the type of motion that LDS is meant to model. In spite of this, the GP model gives better results than the LDS model. We propose that, while these types of targets do move according to an LDS, their images on the camera do not. There are two main reasons for this: (1) the targets are being projected on the image plane and (2) the movement

of the camera which took the images. Clearly, a projection from 3D to 2D completely changes the path that, say, a fish might take. Also, if the camera itself is moving as the video is shot, the relative motion of the targets is drastically altered. The motion is nevertheless still smooth, and thus the Gaussian process model works reasonably well.

(a) $t = 39$ (b) $t = 42$

Figure 3: Two frames of birds flying with the tracks found by GDA-GP superimposed on them. The extra curves are from targets that have already passed through the scene.

6. Conclusion

We presented a model for smooth motion for multiple target tracking. We have compared it with a more standard model (LDS) by integrating it into two different tracking approaches, and testing it on two diverse data sets. The results suggest that this is a very effective way to follow multiple, overlapping tracks, which are relatively smooth, but otherwise arbitrary. Further, the combination of data association and Gaussian processes seems particularly effective, partly by being able to exploit the full run of data if it is available. Even when the number of tracks was large (hard pollen tube data set), only one of them was missed on average, and the ones that were found were followed relatively accurately. We also introduced a Gibbs sampler for finding the associations which works significantly better than MCMCDA on our data sets, and runs faster and is easier to implement.

References

[1] Pollen tube data. `http://kobus.ca/research/data/CVPR_11_pollen`.

[2] Tracking software. `http://kobus.ca/research/programs/CVPR_11_pollen`.

[3] O. Al-Kofahi, R. J. Radke, B. Roysam, and G. Banker. Automated semantic analysis of changes in image sequences of neurons in culture. *IEEE Trans Biomed Eng*, 53(6):1109–23, 2006.

[4] C. Andrieu, N. D. Freitas, and M. J. Arnaud Doucet. An introduction to MCMC for machine learning, 2003.

[5] Y. Bar-Shalom, T. Fortmann, and M. Scheffe. Joint probabilistic data association for multiple targets in clutter. In *Conference on Information Sciences and Systems*, 1980.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, February 1996.

[8] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Toronto, February 1996.

[9] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*. Chapman and Hall/CRC, 1995.

[10] S. Hou, A. Galata, F. Caillette, N. Thacker, and P. Bromiley. Real-time body tracking using a gaussian process latent variable model. In *IEEE 11th International Conference on Computer Vision*, 2007.

[11] J. Huang, C. Guestrin, and L. Guibas. Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Research (JMLR)*, 10:997–1070, May 2009.

[12] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 27(11):1805–1819, November 2005.

[13] E. Lord and S. Russell. The mechanisms of pollination and fertilization in plants. *Annual Review of Cell and Developmental Biology*, 18:81–105, 2002.

[14] N. D. Miller, B. M. Parks, and E. P. Spalding. Computer-vision analysis of seedling responses to light and gravity. *The Plant Journal*, 52:374–381, 2007.

[15] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, University of Toronto, September 1993.

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

[17] R. Palanivelu and D. Preuss. Distinct short-range ovule signal attract or repel arabidopsis thaliana pollen tubes in vitro. *BMC Plant Biology*, 6:7, 2006.

[18] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes For Machine Learning*. MIT Press, 2006.

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

[20] M. Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(2):69–106, 2004.

[21] M. W. Steer and J. M. Steer. Pollen tube tip growth. *New Phytologist*, 111(3):323–358, 1989.

[22] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with gaussian process dynamical models. In *CVPR*, 2006.

[23] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *NIPS '05*, 2005.

[24] L. Wang, A. H. Assadi, and E. Spalding. Tracing branched curvilinear structures with a novel adaptive local pca algorithm. In *Proceedings of the 2008 International Conference on Image Processing, Computer Vision, Pattern Recognition (IPCV)*, 2008.