Branching Gaussian Processes with Applications to Spatiotemporal Reconstruction of 3D Trees

Kyle Simek[1(✉)], Ravishankar Palanivelu[2], and Kobus Barnard[3]

[1] Matterport, Inc., Sunnyvale, USA
kylesimek@gmail.com
[2] School of Plant Sciences, University of Arizona, Tucson, USA
rpalaniv@email.arizona.edu
[3] Computer Science, University of Arizona, Tucson, USA
kobus@cs.arizona.edu

Abstract. We propose a robust method for estimating dynamic 3D curvilinear branching structure from monocular images. While 3D reconstruction from images has been widely studied, estimating thin structure has received less attention. This problem becomes more challenging in the presence of camera error, scene motion, and a constraint that curves are attached in a branching structure. We propose a new general-purpose prior, a *branching Gaussian processes* (BGP), that models spatial smoothness and temporal dynamics of curves while enforcing attachment between them. We apply this prior to fit 3D trees directly to image data, using an efficient scheme for approximate inference based on expectation propagation. The BGP prior's Gaussian form allows us to approximately marginalize over 3D trees with a given model structure, enabling principled comparison between tree models with varying complexity. We test our approach on a novel multi-view dataset depicting plants with known 3D structures and topologies undergoing small nonrigid motion. Our method outperforms a state-of-the-art 3D reconstruction method designed for non-moving thin structure. We evaluate under several common measures, and we propose a new measure for reconstructions of branching multi-part 3D scenes under motion.

Keywords: Multiview stereo · Nonrigid models · Expectation propagation

1 Introduction

Curvilinear 3D tree structure is of central importance to several areas of science including the study of plants [10,34], vascular systems [8,23,28], and branching neurons [6,7]. But estimating such 3D structure from images poses unique challenges, because the 3D shapes are thin, self-occluding, and often textureless. In

addition, traditional reconstruction methods like visual hull or multi-view stereo require precise knowledge of camera poses and intrinsic parameters, and they assume the scene remains stationary between images. When either of these are not true, epipolar constraints are violated and 3D reconstruction fails. This can be especially problematic for multi-view 3D reconstruction of plants, because they can move a surprising amount in just a few minutes.

We propose a model-based approach to recover 3D structure in these scenarios by explicitly modeling nonrigid motion and by defining a noise model that is robust to camera errors. We introduce a new representation called a *curve tree*, which defines both the curves' geometry and their topological structure. The curve tree representation provides more than just *geometric* structure; the *logical* structure is represented as well so that distinct parts may be identified, counted, and measured, and topology of the multi-part structure may be understood. We then introduce a novel family of priors over curve trees called *branching Gaussian processes* (BGPs) that model curves as smooth Gaussian processes and enforce attachment between curves in a tree topology. We also extend the BGP to the temporal dimension to model sequences of curve trees over time. This prior aids inference by limiting the space of plausible models, and in the case of a moving scene, it is necessary to make the problem well-posed.

Using the temporal BGP prior, we pose 3D reconstruction from images as Bayesian inference. Our image-based likelihood function is non-convex and highly multimodal, making exact inference intractable. We develop an approximate inference scheme based on expectation propagation that exploits the BGP prior's Gaussian form to approximate the posterior as multivariate Gaussian. We also approximately solve the intractable Bayesian model selection problem, allowing us to compare curve trees of varying complexity in a principled way. Our method is fully trained from a single exemplar sequence and is effective at recovering 3D branching structure even in scenarios with ambiguous data and self-occlusion. When ambiguity cannot be resolved, our method provides a rich representation of uncertainty in the form of a posterior distribution. We evaluate our results using the DIADEM metric for branching structure and a new score designed to assess the logical structure of multi-part tree-structured 3D models.

2   Related Work

Gaussian processes (GP's) are an increasingly common tool for machine learning and computer vision [24], especially as smoothing priors for deformable geometry. Zhu et al. used GP's for recovery of nonrigid 3D surfaces [42], and Serradell et al. used GP priors to match graphs embedded in 2D or 3D [25]. Like us, Sugiyama et al. proposed constructing a GP over an embedded graph [29], but their covariance function is based on geodesic distance, which is not positive definite in general. Our proposed covariance function is a positive-definite alternative but still has the property that correlations are inversely monotonic with geodesic distance.

Shape estimation from images has been widely studied, but less attention has been paid to the challenges posed by thin geometry. Segmentation

of thin structure in images has gained some attention in recent years [14, 20, 35, 39], but the problem of 3D reconstruction remains largely unexplored. Lopez et al. used space carving to recover thin branching structure as a voxel volume, but their approach relies on near-exact camera calibrations [19]. Amy Tabb achieved robustness to camera errors using a probabilistic variation of visual hull, allowing the position of thin structure to be inconsistent in some cameras and still be reconstructed [30]. Both of these approaches assume the scene is static between images, and they produce only geometry, not part labels or topology. Kahl and August propose a generative model for 3D space curves that follow a Markovian random process through space [16]. They suggest their model could be made robust to both camera errors and scene motion by modeling small perturbations to the 3D curves in each image. We achieve robustness in a similar way, but our model (a) permits a large class of Gaussian processes beyond just Markov processes, (b) introduces branching attachment between curves, (c) enforces temporal consistency between per-view perturbations, and (d) addresses the model selection problem when fitting an unknown number of curves.
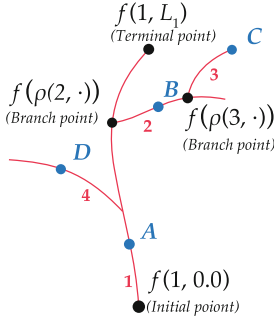
Our approach seeks not just 3D shape (i.e., geometry), but 3D *structure* (e.g., topology, medial axis, distinct geometric parts). Branching topology is of particular importance since it enables measurement of morphological traits such as branch angles, branching depth, and average curvature of individual branches. Approaches exist for estimating 2D trees from images [36,37] and 3D trees from 3D data modalities like point clouds [27] and voxels (e.g., the Diadem challenge, [2,4,37,38,40,41]). Methods estimating 3D tree structure from 2D projective images are less common. Some image-based methods recover a plausible branching 3D plant model without attempting to recover the exact branching architecture [3,9,18,26,31–33]. In addition, several semi-automatic tools use human input to help reconstruct vascular systems [1], root system architectures [5,17,43], and neuron arbors [22] from images.

The motion of 3D trees has been modeled in computer graphics for generating animated synthetic trees (e.g., [9,18]). Estimating the motion of real trees is less common. Glowacki et al. recover sequences of 2D trees in a two-step process, using bottom-up methods to find a graph of tree points and integer programming to find the optimal tree-structured subgraph [13]. However, their bottom-up approach does not address self-occlusions, and they rely on the graph deformation model of Serradell et al. [25] based on Euclidean embeddings, which cannot model 2D curves that pass-by each other due to parallax.

While previous work addresses parts of our problem, we are unaware of other systems that attempt accurate estimation of 3D trees under nonrigid motion.

3   Curve Trees

We define a *curve tree* as a finite set of curves attached in a tree topology. The cth curve in a d-dimensional curve tree is defined by the *curve function* $f_c : [0, L_c] \to \mathbb{R}^d$. In the expression $f_c(t)$, the value $c \in \{1, \ldots, N\}$ is the *curve*

Point properties

Point	Index, (c, t)
A	$(1, 10.0)$
B	$(2, 7.0)$
C	$(3, 11.0)$
D	$(4, 12.0)$

Curve properties

c	$\rho(c, \cdot)$	L_c
1	$(0, 0)$	45.0
2	$(1, 30.0)$	20.0
3	$(2, 13.0)$	11.0
4	$(1, 15.0)$	20.0

Fig. 1. Anatomy of a curve tree (best viewed in color). Curve indices are indicated in red. Initial, terminal, and branch points are noted. (Color figure online)

index, and $t \in [0, L_c]$ is the real-valued *spatial index*. The point $f_c(0)$ is the curve's *initial point*, and $f_c(L_c)$ is its *terminal point*. The Cartesian product of the curve index and spatial index is called a *point index* (c, t), and it uniquely identifies each point in a curve tree. The set of all point indices is the *index space* $\mathcal{I} = \cup_{c=1}^{N}(\{c\} \times [0, L_c]) \cup (0, 0)$, where $(0, 0)$ represents "no point," which is included for notationally convenience. The collection of curve functions are combined into a *curve-tree function* $f : \mathcal{I} \to \mathbb{R}^d$, where $f(c, t) = f_c(t)$ is the point on the cth curve with spatial index t.

If a curve's initial point lies on another curve, we say the curves are attached; the former curve is called a *child curve*, and the point where it attaches on the *parent curve* is called its *branch point*. Curves may have at most one parent curve, and curves with no parent are *root curves*. Attachment between curves is represented by a *branch function*. The function $\rho_{\mathcal{C}} : \{0, \ldots, N\} \to \{0, \ldots, N\}$ maps a curve index to the index of its parent curve or zero for root curves. Similarly, $\rho_{\tau} : \{0, \ldots, N\} \to \mathbb{R}$ maps the curve to the spatial index of its branch point on the parent curve, or zero for root curves. The *branch function* combines $\rho_{\mathcal{C}}$ and ρ_{τ}:

$$\rho(c, t) = \begin{cases} (\rho_{\mathcal{C}}(c), \rho_{\tau}(c)) & \text{if } c \neq 0 \text{ and } \rho_{\mathcal{C}}(c) \neq 0 \\ (0, 0) & \text{otherwise.} \end{cases} \quad (1)$$

The input t is unused, but including it simplifies the recursive definition later. The parent/child attachment is stated formally by the attachment constraint:

$$f(c, 0) = f(\rho(c, t)), \quad (2)$$

for all non-root curves $\{c : \rho(x) \neq (0, 0)\}$ and spatial indices $t \in [0, L_c]$.

A curve tree is fully defined by the number of curves N, the spatial-input upper-bounds $\boldsymbol{L} = \{L_1, \ldots, L_N\}$, the curve-tree function $f(c, t)$, and the branch function $\rho(c, t)$. Figure 1 illustrates these concepts.

4   Branching Gaussian Processes

A Gaussian process is a generalization of the Gaussian distribution to parameterized collections of random variables. These collections may be uncountably infinite, allowing a Gaussian process to define a probability measure over the space of continuous functions. Gaussian processes are defined by the property that any finite subset of random variables has a multivariate Gaussian distribution [24]. A Gaussian process $f : \Theta \rightarrow \mathbb{R}$ is denoted by

$$f(\theta) \sim \mathcal{GP}(\mu(\theta), k(\theta, \theta')), \tag{3}$$

where $\mu(\theta)$ is the mean function, and $k(\theta, \theta')$ is the covariance function. The choice of covariance function dictates the properties of the Gaussian process such as smoothness, periodicity, or stationarity. Without loss of generality, we assume that $\mu(\theta)$ is zero, so $k(\theta, \theta')$ fully defines the Gaussian process.

We propose a novel Gaussian process over curve trees called a *branching Gaussian process* (BGP) by defining a covariance function over the curve tree input space $\mathcal{I}$. A BGP is defined by three properties: (a) individual curves are continuous; (b) the curve tree satisfies the attachment constraint Eq. (2); and (c) each subtree is conditionally independent of the rest of the tree given the subtree's branch point. Below we define a family of branching Gaussian process kernels called a *rooted recursive kernel*, in which (a) a point's marginal variance is determined by its distance from a branch point, and (b) a child curve inherits the variance of its branch point. This is a good model for a plant, where the root point is fixed and leaf points vary more than branch points.

It is also possible to define a family of BGPs in which curves are modeled by stationary covariance functions (e.g., the squared exponential) and the marginal variance of all points are equal (see Supplementary Material).

4.1   Rooted Recursive Kernels

We define a random function $f : \mathbb{R} \rightarrow \mathbb{R}^n$ as *rooted* if and only if $f(0) = \mathbf{0}$ almost surely (i.e., it is "rooted" to the origin). A Gaussian process is rooted if and only if its covariance function has the property $k(0, t) = k(t, 0) = 0$ for any $t \in \mathbb{R}$. Two examples of rooted covariance functions are the dot product covariance function, $f(t, t') = t\,t'$, and the Weiner process covariance function, $f(t, t') = \min(t, t')$. An arbitrary covariance function $k : \mathbb{R}^2 \rightarrow \mathbb{R}$ can be converted to a rooted covariance function k' by conditioning on the point at $t = 0$:

$$k'(t, t') = k(t, t') - k(t, 0)\, k(0, 0)^{-1} k(0, t'). \tag{4}$$

Let (c, t) and (c', t') be the indices of two points in a curve tree, and without loss of generality, assume the tree depth of (c, t) is greater than or equal to (c', t'). The *rooted recursive* BGP kernel $k_r : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}$ is defined as

$$k_r(c, t, c', t') = \begin{cases} 0 & \text{if } c = 0 \text{ or } c' = 0 \\ \delta_{c\,c'} k_c(t, t') + k_r(\rho(c, t), c', t') & \text{otherwise} \end{cases} \tag{5}$$

where $\delta_{c\,c'}$ is the Kronecker delta. The *curve covariance* $k_c : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is any rooted covariance function, and it defines smoothness properties of individual curves. The Kronecker delta ensures the first term contributes zero covariance between different curves. Inter-curve correlation comes from the second term, which is a recursion that replaces the first input with its branch index; if the resulting index is shallower than the second input, inputs are swapped so the first input is always at least as deep as the second. This recursive term contributes a constant offset equal to the variance of the nearest common ancestor point. Because k_c is rooted, it contributes no variance to initial curve points, so $k(c, 0, c, 0) = k(\rho(c), c, 0) = k(\rho(c), \rho(c))$, which implies the attachment property, $f(c, 0) = f(\rho(c, 0))$.

4.2   Temporal Branching Gaussian Processes

We may extend the curve tree model by introducing a temporal input τ to the point index, $(c, t, \tau) \in \mathcal{I} \times \mathbb{R}$. We construct a temporal BGP covariance function from the tensor product of a standard BGP covariance function and a temporal covariance function:

$$k_{\mathrm{tBGP}}(c, t, \tau, c', t', \tau') = k_\tau(\tau, \tau') k_{\mathrm{BGP}}(c, t, c', t'). \tag{6}$$

Here, k_τ is a covariance function over temporal inputs $\tau, \tau' \in \mathbb{R}$, and k_{BGP} is any BGP covariance function.

5   Application: Temporal 3D Tree Structure from Images

For the reconstruction task, we consider the first 18 images of an *Arabidopsis* plant on a turnable separated by 10 degrees of yaw (Fig. 2a) (the rear 18 images do not provide significant additional structural information). Images are captured at fixed time intervals, so time is represented by view-index. A camera model is estimated for each view using a crudely built calibration target, resulting in significant error (Fig. 2c). Capture takes roughly three minutes, during which time the plant moves in response to changing direction of light and temperature gradients (Fig. 2b). Our goal is to recover a 3D curve tree for each of the 18 views. This problem is underconstrained, since variation in appearance between views may be explained by either parallax, nonrigid plant motion, or both. Our temporal BGP provides a prior over motion and geometry, making the problem well-posed. We use a curve tree to represent the medial axis of a plant's stems. Stems are mostly uniform in width, so we define the plant's shape as the union of spheres with 1 mm diameter and center $f(c, t)$ for all $(c, t) \in \mathcal{I} \backslash (0, 0)$.

5.1   Prior

The motion of plants tends to arise as random perturbations around a mean shape, and we define a BGP covariance function that reflects this:

$$k(c, t, \tau, c', t', \tau') = k_{\mathrm{BGP}}(c, t, c', t') + k_{\mathrm{tBGP}}(c, t, \tau, c', t'\tau') + \sigma_0^2. \tag{7}$$

(a)   (b)   (c)

Fig. 2. *(a)* First three images in an 18-view sequence, with ten degrees of rotation between views. *(b)* A stationary 3D point on a turntable should trace an elliptical arc in 2D; the non-elliptical path here indicates nonrigid motion. *(c)* Turntable cameras should follow a circular path and have symmetric viewing directions. This top-down illustration shows significant camera calibration error, which complicates inference.

The first term in Eq. (7) is a branching GP covariance that represents a non-moving tree that is the center of perturbations. We implement it using the rooted-recursive covariance function from Eq. (5) with curve covariance

$$k_c(t, t') = \sigma_L^2 t\, t' + \sigma_s^2 \left(|t - t'| \frac{\min(t, t')^2}{2} + \frac{\min(t, t')^3}{3} \right). \tag{8}$$

This is the sum of a dot-product kernel plus a cubic-spline covariance function. The cubic-spline process penalizes the curve's second derivative, representing a bending energy that encourages straight lines. The cubic-spline covariance can only generate curves having zero initial slope, so we add a dot-product kernel to model initial slopes.

The second term in Eq. (7) is the temporal BGP covariance from Eq. (6) and it models random perturbations. For the k_τ term in Eq. (6) we use the Ornstein Uhlenbeck covariance function,

$$k_\tau(\tau, \tau') = \exp\left(|\tau - \tau'|/\ell_\tau\right), \tag{9}$$

which is a mean-reverting random walk that ensures random perturbations never drift too far from the mean tree structure. For the k_{BGP} term in Eq. (6) we again use a rooted recursive BGP covariance function with the same form as Eq. (8) but with different parameters $(\sigma_L')^2$ and $(\sigma_s')^2$.

The final term in Eq. (7) is a constant offset variance. Because the first two terms are rooted recursive BGP covariance functions, the tree's root point has zero marginal variance under them. Adding a constant offset variance allows trees to initiate from anywhere in $\mathbb{R}^3$, not just at the origin. We model all three (x, y, z) dimensions as i.i.d. under this prior.

Although the curve tree curves are continuous, in practice we perform inference on a discrete subset of points at indices $X = \{(c_1, t_1), \ldots, (c_M, t_M)\}$. Let $z_{ij} = f(c_j, t_j, \tau_i)$ be the jth 3D point in the curve tree at time τ_i, and let the vector $\boldsymbol{z}_i = (z_{i1}^\top, \ldots, z_{iM}^\top)^\top$ represent the discretely sampled curve tree at that time. Let the vector $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T\}$ represent the set of curve trees over all time indices.

Our choice of k_τ makes our model Markovian in the temporal dimension, so we can decompose the joint prior into

$$p(\boldsymbol{Z}) = p(\boldsymbol{z}_1) \prod_{i=2}^{T} p(\boldsymbol{z}_i | \boldsymbol{z}_{i-1}). \tag{10}$$

Since all factors are linear Gaussian, this represents a linear dynamical system (LDS),

$$\boldsymbol{z}_i = F_i \boldsymbol{z}_{i-1} + \epsilon_i, \tag{11}$$

with transition matrix F_i and system noise $\epsilon_i \sim \mathcal{N}(0, Q_i)$ defined as

$$F_i = \ddot{K}_{*i}^\top \ddot{K}_i^{-1}, \tag{12}$$

$$Q_i = \ddot{K}_i - \ddot{K}_{*i}^\top \ddot{K}_i^{-1} \ddot{K}_{*i}. \tag{13}$$

Here, $\ddot{K}$ denotes the Kronecker product $K \odot I_3$, and

$$(K_i)_{jk} = k(c_j, t_j, \tau_i, c_k, t_k, \tau_i), \tag{14}$$

$$(K_{*i})_{jk} = k(c_j, t_j, \tau_{i-1}, c_k, t_k, \tau_i). \tag{15}$$

The Kronecker product lifts the covariance matrix from modeling one output dimension to modeling three i.i.d. dimensions.

We train the prior using maximum marginal likelihood [24], using a gradient-free local optimizer (MATLAB's `patternsearch` function) and multiple initializations to avoid local maxima. (See Supplementary Material for trained values.)

5.2   Likelihood

The likelihood function projects the plant's 3D shape into each view, rasterizes it into a binary foreground/background image, and evaluates each pixel against the input image. We trained a random forest classifier to identify foreground (i.e., stem) pixels using a single training image [15]. The classifier generates a map D whose ith pixel has value $d_i \in [0, 255]$ representing the posterior probability of the class (foreground/background) given the evidence. However, using posterior values in a likelihood function would introduce bias in favor of the more common class (i.e., background), so we next re-calibrate the classifier outputs to represent *likelihood* values, i.e., the probability of the *evidence*, given the *class*.

Let $\gamma_i^* \in \{0, 1\}$ be value of the ith binary pixel in the training foreground mask. We construct a histogram over the set $\{d_i | \gamma_i^* = 1\}$ that represents the likelihood of classifier output given a foreground label, $p(d_i | \text{fg})$. We construct a background likelihood $p(d_i | \text{bg})$ similarly. To avoid overfitting, we smooth the histograms with a Gaussian filter with $\sigma_{\text{lik}} = 1$. The likelihood of the image at time τ is then

$$p(D_\tau | \boldsymbol{z}_\tau) = \prod_i p(d_i | \text{fg})^{\gamma_{\tau i}} p(d_i | \text{bg})^{1 - \gamma_{\tau i}}. \tag{16}$$

where $\gamma_{\tau i}$ is the value of the ith binary pixel in the rasterization of $\boldsymbol{z}_\tau$ into view τ. The supplementary materials describe an efficient method evaluating this likelihood. The full likelihood over all images $\boldsymbol{D} = \{D_1, \dots, D_T\}$ is the product of per-view likelihoods, $p(\boldsymbol{D}|\boldsymbol{Z}) = \prod_{\tau=1}^{M} p(D_\tau|\boldsymbol{z}_\tau)$.

6 Inference

We begin by describing inference of a curve tree's 3D shape when its topological structure (i.e., the number of curves, point indices, and branch function) is known. Later, we present methods for bootstrapping and refining the topology using Bayesian model selection. The goal is to estimate a posterior distribution over points in the curve tree over all time frames. Closed-form inference is not possible because the likelihood function is non-convex and highly multimodal. However, the posterior distribution is likely to be highly peaked due to strong evidence from multiple views, making it well modeled by a multivariate Gaussian distribution. Thus, we seek a Gaussian approximation of the true posterior that minimizes their Kullback-Leibler divergence.

6.1 Estimating 3D Structure Given Topology

Our inference approach assumes 3D points are initially observed by a single reference image, making them well-localized in two dimensions, but their depth is uncertain; Sects. 6.2 and 6.3 describe methods for achieving such initializations. Under this assumption, inference is akin to an epipolar search in several views. Conditioned on the reference view, a point's probability density is concentrated along a 3D ridge near the reference image's backprojection line. This ridge's width is a function of camera calibration error and scene motion, and its length is loosely constrained by the prior to be near the center of the scene. We call the projection of this ridge the *epipolar region* of a point in a particular view. The prior defines strong spatiotemporal correlations between points, so ambiguous evidence in a particular image can be resolved by consulting nearby points in space and time that are unambiguous. This contextual information can shrink the ambiguous point's epipolar region, allowing some local optima to be ruled out, and making the point suitable for informing other ambiguous points.

We implement this logic using the expectation propagation algorithm (EP) where a Gaussian approximation of the posterior is iteratively updated, starting as noninformative and becoming more peaked as more evidence is considered [21]. When the likelihood in an epipolar region has multiple modes, it is approximated by a Gaussian with high variance so all modes are well supported. This effectively defers a decision on the point's location until later iterations when local context is more informative.

Let $f(\theta)$ be an arbitrary distribution and $q(\theta)$ be a Gaussian approximation. EP seeks parameters of q that minimize the Kullback-Leibler divergence:

$$KL(f||q) = \int f(\theta) \log \frac{f(\theta)}{q(\theta)} d\theta. \tag{17}$$

Our posterior distribution has the form

$$p(\boldsymbol{Z}|D) \propto p(\boldsymbol{Z}) \prod_{i=1}^{n} L_i(\boldsymbol{Z}), \tag{18}$$

where $p(\boldsymbol{Z})$ is the temporal BGP prior in Eq. (10) and $L_i(\boldsymbol{Z}) = p(D_i|\boldsymbol{z}_i)$ are the single-view likelihood terms from Eq. (16). We seek a multivariate Gaussian approximation $q(\boldsymbol{Z}) \propto p(\boldsymbol{Z}) \prod_{i=1}^{T} \hat{L}_i(\boldsymbol{Z})$, where $\hat{L}_i(\boldsymbol{Z})$ are multivariate Gaussian approximations of $L_i(\boldsymbol{Z})$ with diagonal covariance.

Standard expectation propagation involves estimating the first two moments of the joint posterior at each iteration. In a curve-tree sequence with M points and T time frames, this would involve an intractable integral in $3TM$ dimensions. Instead, we adopt a variation of EP proposed by Gelman et al., where conditionally independent likelihood terms are updated independently in parallel in each iteration, rather than updating the entire joint posterior [11]. If we assume the likelihood of each point is independent conditioned on the 3D model, Gelman's method reduces moment-estimation to TM integrals in three dimensions. This independence assumption holds if we ignore self-occlusions which are minimal and take the limit as polyline point density approaches infinity.

Our EP implementation is shown in Algorithm 1. Here $q_{ij}(z_{ij})$ is the single-point marginal of the approximate Gaussian posterior distribution, $q(\boldsymbol{z}_i)$. We define the single-point likelihood $L_{ij}(z_{ij})$ as the likelihood of a sphere centered at z_{ij} with diameter 1 mm, evaluated under the single-view likelihood in Eq. (16). This single-point, single-view likelihood can be computed efficiently using a lookup table (see supplement). We approximate the moments of the three-dimensional marginal posterior of a point using importance sampling (again, see supplement). Gelman's algorithm requires a global update after each pass, causing the joint posterior to reflect the updated likelihood approximations. This would take $O(M^3T^3)$ time if implemented naïvely; instead, we exploit our model's LDS structure from Eq. (11) to implement the global update with a Kalman filter and smoother in $O(M^3T)$ time.

6.2   Bootstrapping

We bootstrap the inference process by using 2D image processing and backprojection to estimate an initial 3D tree whose depth is uncertain. We first construct a binary image of likely foreground pixels in the first view consisting of pixels whose foreground likelihoods are higher than their background likelihoods. The binary map is further refined by taking the median image over all views and removing any pixel whose intensity is not significantly different from the median pixel–i.e., basic background subtraction. Next, we eliminate all but the largest connected component from the foreground image. The result is a set of likely stem pixels with relatively few false positives.

We construct a 2D tree from this binary image by taking its morphological skeleton, constructing an adjacency graph of skeleton pixels, and extracting the

Algorithm 1. Our proposed variant of expectation propagation

1. Initialize $\{\hat{L}_{ij}(z_{ij})\}_{i=1}^{T}$ to improper Gaussians for $i \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, |\boldsymbol{z}|\}$.
2. Initialize $\{q_i(\boldsymbol{z}_i)\}_{i=1}^{T}$ to the marginal BGP priors, $\{p(\boldsymbol{z}_i)\}_{i=1}^{T}$.
3. For each $(i,j) \in \{1, \ldots, T\} \times \{1, \ldots, |\boldsymbol{z}|\}$:
   (a) Compute the "cavity" distribution $q_{ij}^{-(i,j)}(z_{ij}) = q_{ij}(z_{ij})/\hat{L}_{ij}(z_{ij})$.
   (b) Compute the "tilted" distribution $q_{ij}^{\backslash(i,j)}(z_{ij}) = q_{ij}^{-(i,j)}(z_{ij})L_{ij}(z_{ij})$.
   (c) Update $\hat{L}_{ij}(z_{ij})$ so $q_{ij}^{-(i,j)}(z_{ij})\hat{L}_{ij}(z_{ij})$ approximates $q_{ij}^{\backslash(i,j)}(z_{ij})$:
       i. Construct a 3×3 Gaussian $q_{ij}(z_{ij})$ using the first two moments of $q_{ij}^{\backslash(i,j)}(z_{ij})$.
       ii. Set $\hat{L}_{ij}(z_{ij}) = q_{ij}(z_{ij})/q_{ij}^{-(i,j)}(z_{ij})$.
4. Use Kalman filter and smoother to re-compute the per-view marginal posteriors $\{q_1, \ldots, q_M\}$ using the new likelihood approximations $\hat{L}_{ij}$.
5. Repeat step 3 until convergence.

graph's Euclidean minimum spanning tree. Each point in the 2D tree is back-projected to a depth closest to the prior mean (i.e., the origin). Chains of points with degree two or less are grouped into curves; points with higher degree become branch points. Each curve's points are assigned spatial indices according to their arc-length parameterization in 3D. In this way, we derive the topological structure of a 3D curve tree: the number of curves N, curve lengths L_i, point indices X, and the branch function $\rho(c,t)$. This bootstrapping heuristic can miss some curves and mis-estimate branch points, but these errors may be fixed during the topology refinement process described in the next section.

We construct an initial likelihood approximation for the first view from the 2D tree points by

$$\hat{L}_{1j}(z_{1j}) = \mathcal{N}(z_{1j}, \Lambda_{1j}^{-1}) \text{ where} \tag{19}$$

$$\Lambda_{1j} = H'(I_2/\sigma_L^2)H \tag{20}$$

Here, z_{1j} is the jth backprojected 3D point, H is the Jacobian of the perspective projection of z_{1j} into the first view, and σ_L^2 represents the uncertainty of the detected points. We use $\sigma^2 = 1$ pixel^2, which represents error due to pixel quantization and small localization errors. The precision matrix Λ is rank deficient, reflecting a lack of evidence for the point's depth.

6.3   Topology Refinement Using Bayesian Model Selection

Trees with differing number of curves will have different dimensionality, so their posterior distributions are not comparable. This is a model selection problem, and the Bayesian solution is to integrate out the continuous parameters (i.e., the curve points) and compare the models' marginal likelihoods. The marginal likelihood does not over-fit and it includes a natural penalty for model complexity, making it an ideal model selection criterion. Computing it involves a

---

Algorithm 2. 3D Tree Structure Estimation

1. Initialize each $\hat{L}_i(\boldsymbol{z}_i), q_i(\boldsymbol{z}_i)$ to degenerate Gaussians for $i \in \{1, \ldots, T\}$.
2. Bootstrap: estimate $\hat{L}_1(\boldsymbol{z}_1), q_1(\boldsymbol{z}_1)$ using the procedure in Sect. 6.2.
3. Update $\{q_2(\boldsymbol{z}_2), \ldots, q_T(\boldsymbol{z}_T)\}$ with a Kalman filter pass.
4. Run three iterations of EP, keeping q_1 and $\hat{L}_1$ fixed on first pass.
5. Run pruning.
6. For each time index $i \in \{5, 9, 13, \ldots, T\}$:
    (a) Run birthing with reference image i.
    (b) Run three iterations of EP.
    (c) Run pruning.

---

high-dimensional integral that is intractable, but we estimate it using the Gaussian approximation we constructed during expectation propagation.

The marginal likelihood is $p(\boldsymbol{D}|\mathcal{M})$, where $\mathcal{M}$ represents curve tree's topological parameters, i.e., the number of curves N, the point indices X, and the branch function $\rho(c, t)$. In what follows, we omit the dependence on $\mathcal{M}$.

Using the chain rule we rewrite the marginal likelihood as

$$p(\boldsymbol{D}) = p(D_1) \prod_{t=2}^{T} p(D_t|D_{1:i-1}). \tag{21}$$

Under Bayes' rule (after rearranging),

$$p(D_i|D_{1:i-1}) = p(D_i|\boldsymbol{z}_i) \left(p(\boldsymbol{z}_i|D_{1:i-1})/p(\boldsymbol{z}_i|D_{1:i}) \right). \tag{22}$$

The first term in (22) is the true single-view likelihood, $L_i(\boldsymbol{z}_i)$. The second term is a ratio of the partial posteriors before and after considering data D_i. Gaussian approximations of these distributions are computed during Kalman filtering before and after the update step, respectively. In theory, any value of $\boldsymbol{z}_i$ can be used to evaluate Eq. (22), but our approximations are most accurate near the true posterior mode, so we use the current estimate of the mean of the approximate Gaussian posterior, $q(\boldsymbol{Z})$.

The space of tree topologies can be explored by proposing incremental changes to the current curve tree topology and keeping the model with the highest marginal likelihood. We propose two such methods: pruning and birthing.

Pruning. In the pruning step, a curve is deleted if doing so improves the marginal likelihood. Each curve is considered for pruning in random order. Pruning a parent curve causes its descendant subtrees to become independent trees. Pruning a Y-junction results in two curves attached end-to-end.

Birthing. The birthing step mimics bootstrapping, but when constructing the binary mask, we omit foreground pixels if they coincide with the projection of existing structure. The remaining pixels are skeletonized, backprojected, and converted to 3D curve trees as in bootstrapping, and the root of each new tree is attached to the nearest point on existing structure. Likelihood approximations

are created for the new points in the reference view using Eq. (19). We propagate this likelihood to the other views by a pass of Kalman filtering and smoothing. A subsequent pruning step removes spurious structure added during birthing.

Algorithm 2 shows how bootstrapping, geometry estimation, and topology refinement are applied in an end-to-end system. In principle, expectation propagation should be run after each model modification before evaluating the marginal likelihood. In practice, we omit EP after the pruning step, as we observed that it caused negligible change to the model and the approximated marginal likelihood.

7   Experiments and Results

We applied our approach to recover 3D structure of plants from images. Our dataset consists of image sequences of twelve *Arabidopsis* plants as described in Sect. 5. One specimen was held out for training, the rest were used for testing. Ground truth was manually collected on every fourth image by tracing each stem with a 2D poly-Bézier curve. Curve identities were kept consistent across views, and parent/child relationships between curves were noted. Each poly-Bézier curve was sampled at one pixel intervals and each point in each image was backprojected to a depth that minimized the sum of squared distances between its 2D projection and the nearest ground truth curve point in each image. This results in a sequence of five 3D curve trees per specimen that project exactly to their 2D counterparts.

We evaluate using four measures. We compute a one-to-one matching between points with distance at most $1\,\mathrm{cm}$ (see supplement). We report accuracy as the RMS error of matching points and we measure completeness and spurious structure using an intersection-over-union score that compares the number of

(a)   (b)   (c)   (d)
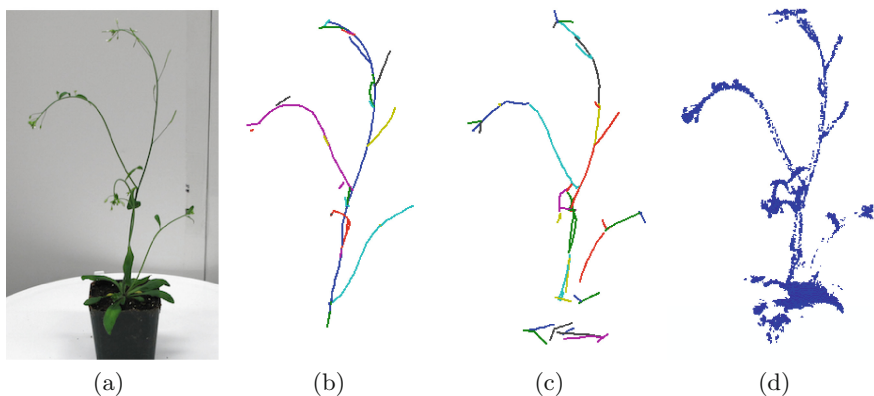
Fig. 3. Results of 3D reconstruction. *(a)* One of 18 input images. *(b)* Semi-automatic reconstruction result. Color indicates distinct parts. *(c)* Fully-automatic reconstruction result. *(d)* Voxel volume produced by [30]. (Color figure online)

matched points against the total number of points in both models. We evaluate tree topology using the DIADEM metric with a threshold of 1 cm in the x, y, and z directions [12]. However, the DIADEM metric does not distinguish "parts," so it cannot distinguish between Y- and T-junctions, and it does not recognize part-level structural errors like over-segmentations of curves or misclassified T-junctions. Further, it has no concept of time, so identity switches are not penalized.

To address these issues, we propose a *tree-structure consistency* (TSC) score that measures structural consistency in three ways: (a) the geometric consistency of matching curves, (b) matching consistency between child and parent curves, and (c) consistency of curve identity between adjacent time frames. Evaluation begins by finding a correspondence between parts that maximizes their number of matching points. Then consistency is measured with respect to the corresponding parts. The TSC score measures the number of logical errors in the structure, normalized to the range $[0, 1]$, where 1 represents "no errors." For details, see the supplementary material.

Table 1. Structure estimation performance of semi- and fully-automatic reconstruction, for nine- and 18-image datatasets.

	Acc	IoU	TSC	DIADEM [12]
Semi (9)	4.7 ± 1.4	0.63 ± 0.14	0.49 ± 0.09	0.81 ± 0.09
Semi (18)	3.7 ± 1.0	0.70 ± 0.08	0.56 ± 0.06	0.84 ± 0.06
Auto (9)	7.1 ± 3.1	0.50 ± 0.11	0.41 ± 0.07	0.41 ± 0.18
Auto (18)	5.9 ± 2.2	0.59 ± 0.11	0.43 ± 0.08	0.44 ± 0.18

We ran our algorithm twice, once using all 18 images, and once using only the first nine. We also compared two variants of bootstrapping: a fully-automatic approach as described in Sect. 6.2, and a semi-automatic approach where bootstrapping is replaced with a manually traced 2D curve tree in the first image. Inference took an average of 33 min using Matlab code on a 16-core Xeon E5-2650 CPU running at 2.6 GHz. Results for all runs are shown in Table 1. We see that semiautomatic reconstruction performs significantly better on the TSC score than the fully automatic one, primarily due to the latter oversegmenting curves. This could be addressed with a "join curves" step during topology refinement. Figure 3 provides a visual comparison against Tabb's Shape-from-Silhouette-Probability-Maps (SfSPM) algorithm, which produces volumetric reconstructions of thin structure [30]. (For visualizations of all eleven reconstructions, see supplementary material.)

8   Conclusion

We have demonstrated a novel probabilistic model for branching multidimensional structure, and a Bayesian method for 3D reconstruction of moving trees

from a single moving camera. Our approach obtains good results with limited training data, showing that it generalizes well and avoids overfitting. Our model selection scheme naturally penalizes complexity with no additional tunable parameters. By taking advantage of the model structure, our expectation propagation scheme locates strong modes reasonably efficiently and should outperform black-box Bayesian inference like hybrid Monte Carlo. Our proposed BGP covariance function is a general-purpose kernel and can be applied in kernel methods such as SVM, PCA, GP classification, and kernel density estimation. It is also straight-forward to extend branching curves to branching surfaces; this could be used to model deformable leaves attached to the curve tree. We also contributed a new score measuring logical consistency of multi-part branching structure. Our dataset and evaluation code may be downloaded from ivilab.org.

References

1. The Vascular Modeling Toolkit. http://www.vmtk.org. Accessed 5 Mar 2015
2. Bas, E., Erdogmus, D.: Principal curves as skeletons of tubular objects. Neuroinformatics **9**(2), 181–191 (2011)
3. Chen, X., Neubert, B., Xu, Y.Q., Deussen, O., Kang, S.B.: Sketch-based tree modeling using Markov random field. In: ACM SIGGRAPH Asia 2008 Papers, pp. 109:1–109:9. ACM, New York (2008)
4. Chothani, P., Mehta, V., Stepanyants, A.: Automated tracing of neurites from light microscopy stacks of images. Neuroinformatics **9**(2–3), 263–278 (2011)
5. Clark, R.T., MacCurdy, R.B., Jung, J.K., Shaff, J.E., McCouch, S.R., Aneshansley, D.J., Kochian, L.V.: Three-dimensional root phenotyping with a novel imaging and software platform. Plant Physiol. **156**(2), 455–465 (2011)
6. Cubelos, B., Sebastián-Serrano, A., Beccari, L., Calcagnotto, M.E., Cisneros, E., Kim, S., Dopazo, A., Alvarez-Dolado, M., Redondo, J.M., Bovolenta, P., Walsh, C.A., Nieto, M.: Cux1 and Cux2 regulate dendritic branching, spine morphology, and synapses of the upper layer neurons of the cortex. Neuron **66**(4), 523–535 (2010)
7. Cuntz, H., Forstner, F., Borst, A., Häusser, M.: One rule to grow them all: a general theory of neuronal branching and its practical application. PLoS Comput. Biol. **6**(8), 1–14 (2010)
8. Den Buijs, O.J., Bajzer, Z., Ritman, L.E.: Branching morphology of the rat hepatic portal vein tree: a micro-CT study. Ann. Biomed. Eng. **34**(9), 1420–1428 (2006)
9. Diener, J., Reveret, L., Fiume, E.: Hierarchical retargetting of 2D motion fields to the animation of 3D plant models. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2006, Vienna, Austria (2006)
10. Evers, J.B., van der Krol, A.R., Vos, J., Struik, P.C.: Understanding shoot branching by modelling form and function. Trends Plant Sci. **16**(9), 464–467 (2011)

11. Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N., Cunningham, J.P.: Expectation propagation as a way of life, December 2014. arXiv.org
12. Gillette, T.A., Brown, K.M., Ascoli, G.A.: The DIADEM metric: comparing multiple reconstructions of the same neuron. Neuroinformatics **9**(2–3), 233–245 (2011)
13. Glowacki, P., Pinheiro, M.A., Türetken, E., Sznitman, R., Lebrecht, D., Kybic, J., Holtmaat, A., Fua, P.: Reconstructing evolving tree structures in time lapse sequences. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3035–3042. IEEE Computer Society (2014)
14. Gu, L., Cheng, L.: Learning to boost filamentary structure segmentation. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 639–647. IEEE (2015)
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
16. Kahl, F., August, J.: Multiview reconstruction of space curves. In: 2003 IEEE International Conference on Computer Vision (ICCV), pp. 1017–1024 (2003)
17. Le Bot, J., Serra, V., Fabre, J., Draye, X., Adamowicz, S., Pagès, L.: DART: a software to analyse root system architecture and development from captured images. Plant Soil **326**(1–2), 261–273 (2009)
18. Li, C., Deussen, O., Song, Y.Z., Willis, P., Hall, P.: Modeling and generating moving trees from video. In: Proceedings of the 2011 SIGGRAPH Asia Conference, pp. 127:1–127:12. ACM, New York (2011)
19. Lopez, L.D., Ding, Y., Yu, J.: Modeling complex unfoliaged trees from a sparse set of images. Comput. Graph. Forum **29**, 2075–2082 (2010)
20. Marin, D., Boykov, Y., Zhong, Y.: Thin structure estimation with curvature regularization, June 2015. arXiv.org
21. Minka, T.P.: Expectation propagation for approximate Bayesian inference, January 2013. arXiv.org
22. Narro, M.L., Yang, F., Kraft, R., Wenk, C., Efrat, A., Restifo, L.L.: NeuronMetrics: software for semi-automated processing of cultured neuron images. Brain Res. **1138**, 57–75 (2007)
23. Nordsletten, D.A., Blackett, S., Bentley, M.D., Ritman, E.L., Smith, N.P.: Structural morphology of renal vasculature. Am. J. Physiol. Heart Circ. Physiol. **291**(1), H296–H309 (2006)
24. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
25. Serradell, E., Glowacki, P., Kybic, J., Moreno-Noguer, F., Fua, P.: Robust nonrigid registration of 2D and 3D graphs. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 996–1003 (2012)
26. Shlyakhter, I., Rozenoer, M., Dorsey, J., Teller, S.: Reconstructing 3D tree models from instrumented photographs. IEEE Comput. Graph. Appl. **21**(1), 53–61 (2001)
27. Song, M., Huber, D.: Automatic recovery of networks of thin structures. In: 2015 International Conference on 3D Vision (3DV), pp. 37–45. IEEE (2015)
28. Strasser, G.A., Kaminker, J.S., Tessier-Lavigne, M.: Microarray analysis of retinal endothelial tip cells identifies CXCR4 as a mediator of tip cell morphology and branching. Blood **115**(24), 5102–5110 (2010)
29. Sugiyama, M., Hachiya, H., Towell, C., Vijayakumar, S.: Geodesic Gaussian kernels for value function approximation. Auton. Rob. **25**(3), 287–304 (2008)

30. Tabb, A.: Shape from silhouette probability maps: reconstruction of thin objects in the presence of silhouette extraction and calibration error. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 161–168. IEEE (2013)
31. Talton, J.O., Lou, Y., Lesser, S., Duke, J., Mech, R., Koltun, V.: Metropolis procedural modeling. ACM Trans. Graph. **30**(2), 11:1–11:14 (2011)
32. Tan, P., Fang, T., Xiao, J., Zhao, P., Quan, L.: Single image tree modeling. ACM Trans. Graph. **27**(5), 108:1–108:7 (2008)
33. Tan, P., Zeng, G., Wang, J., Kang, S.B., Quan, L.: Image-based tree modeling. ACM Trans. Graph. **26**(99), 87 (2007)
34. Trachsel, S., Kaeppler, S.M., Brown, K.M., Lynch, J.P.: Shovelomics: high throughput phenotyping of maize (Zea mays L.) root architecture in the field. Plant Soil **341**(1–2), 75–87 (2011)
35. Tu, Z., Zhu, S.C.: Parsing images into regions, curves, and curve groups. Int. J. Comput. Vis. **69**(2), 223–249 (2006)
36. Türetken, E., Benmansour, F., Andres, B., Pfister, H., Fua, P.: Reconstructing loopy curvilinear structures using integer programming. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829. IEEE Computer Society, June 2013
37. Türetken, E., Benmansour, F., Fua, P.: Automated reconstruction of tree structures using path classifiers and mixed integer programming. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 566–573. IEEE (2012)
38. Türetken, E., González, G., Blum, C., Fua, P.: Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. Neuroinformatics **9**(2–3), 279–302 (2011)
39. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8, June 2008
40. Wang, Y., Narayanaswamy, A., Tsai, C.L., Roysam, B.: A broadly applicable 3-D neuron tracing method based on open-curve snake. Neuroinformatics **9**(2–3), 193–217 (2011)
41. Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., Long, F., Myers, E.: Automated reconstruction of neuronal morphology based on local geometrical and global structural models. Neuroinformatics **9**(2–3), 247–261 (2011)
42. Zhu, J., Hoi, S.C., Lyu, M.R.: Nonrigid shape recovery by Gaussian process regression. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1319–1326 (2009)
43. Zhu, T., Fang, S., Li, Z., Liu, Y., Liao, H., Yan, X.: Quantitative analysis of 3-dimensional root architecture based on image reconstruction and its application to research on phosphorus uptake in soybean. Chin. Sci. Bull. **51**(19), 2351–2361 (2006)