Seeing Tree Structure from Vibration

Tianfan Xue[1](✉), Jiajun Wu[2], Zhoutong Zhang[2], Chengkai Zhang[2], Joshua B. Tenenbaum[2], and William T. Freeman[2,3]

[1] Google Research, Mountain View, USA
tianfan.xue@gmail.com
[2] MIT CSAIL, Cambridge, USA
[3] Google Research, Cambridge, USA

Abstract. Humans recognize object structure from both their appearance and motion; often, motion helps to resolve ambiguities in object structure that arise when we observe object appearance only. There are particular scenarios, however, where neither appearance nor spatial-temporal motion signals are informative: occluding twigs may look connected and have almost identical movements, though they belong to different, possibly disconnected branches. We propose to tackle this problem through spectrum analysis of motion signals, because vibrations of disconnected branches, though visually similar, often have distinctive natural frequencies. We propose a novel formulation of tree structure based on a physics-based link model, and validate its effectiveness by theoretical analysis, numerical simulation, and empirical experiments. With this formulation, we use nonparametric Bayesian inference to reconstruct tree structure from both spectral vibration signals and appearance cues. Our model performs well in recognizing hierarchical tree structure from real-world videos of trees and vessels.

Keywords: Vibration · Tree structure · Hierarchical Bayesian model

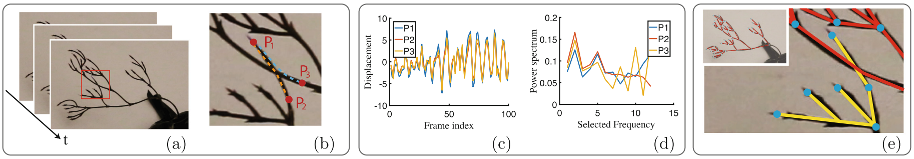
1 Introduction

In visual perception, motion information often helps to resolve appearance ambiguities. Animals may conceal themselves with camouflaged clothing, but they are unlikely to match their motion with that in the background, such as foliage waving in the breeze [6]. In medical imaging, it might be hard to separate blood vessels (or fibers) purely from their appearance, but the distinction becomes clear once the vessels start to vibrate. Extensive studies in cognitive science also suggest that humans, including young children, recognize objects from both appearance and motion cues [37].

T. Xue and J. Wu—Contributed equally to this work.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01240-3_46) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11213, pp. 762–779, 2018.
https://doi.org/10.1007/978-3-030-01240-3_46



Input Extracted vibration Estimated tree structure

Fig. 1. We want to infer the hierarchical structure of the tree in video (a). Inference based on a single frame has inherent ambiguities: figure (b) shows an example, where it is hard to tell from appearance whether point P_1 is connected to P_2 (orange curve) or to P_3 (blue curve). Time domain motion signals do not help much, as these branches have almost identical movements (c). We observe that the difference is significant in the frequency domain (d), from which we can see P_1 is more likely to connect to P_2 due to their similar spectra. We therefore develop an algorithm that infers tree structure based on both vibration spectra and appearance cues. The results are shown in (e). (Color figure online)

Computer vision researchers have combined motion and appearance information to solve a range of tasks [1,34]. Bouman *et al.* proposed to estimate physical object properties based on their appearance and vibration [3]. Wang *et al.* proposed a layered motion representation [42], which has been widely employed in object segmentation and structural prediction [23,38].

In this paper, we focus on tree structure estimation. This problem is even more challenging, as both motion and appearance cues can fail to discriminate pixels of disjoint branches. We show an example in Fig. 1. The three points $\{P_i\}$ in Fig. 1 are on two occluding branches. There are two plausible explanations: either P_1 and P_2, or P_1 and P_3 may be on the same branch. Due to self-occlusion, it is hard to infer the underlying connection just from their appearance. It is also challenging to resolve this ambiguity using only temporal motion information: the movement of these three nodes are dominated by the vibration of the root branch, so they share almost the same trajectories (Fig. 1c).

We propose to incorporate spectral analysis to deal with this problem. This is inspired by our observation that pixels of different branches often have distinctive modes in their spectra of frequency responses, despite their similar spatial trajectories. As shown in Fig. 1d, P_3 has distinct amplitude at certain frequencies compared with P_1 and P_2; intuitively and theoretically (discussed in Sect. 3), P_3 is more likely to be on a separate branch.

Our formulation of tree vibration builds upon and extends a physics-based link model from the field of botany [33]. Here, we deduce a key property of tree structure: each branch is a linear time-invariant (LTI) system with respect to the vibration of root. With this property, we can infer the natural frequencies of each sub-branch in a tree from its frequency response, and group nodes based on the inferred natural frequencies. We also provide justifications of this property through theoretical analysis, numerical simulation, and empirical experiments.

Based on our tree formulation, we develop a hierarchical grouping algorithm to infer tree structure, using both spectral motion signals and appearance cues.

As each node in a tree may connect to an indefinite number of children, our inference algorithm employs nonparametric Bayesian methods.

For evaluation, we collect videos of both artificial and real-world tree-structured objects. We demonstrate that our algorithm works well in recognizing tree structure, using both appearance cues and spectra of vibration. We compare our algorithm with baselines that use spatial motion signals; we also conduct ablation studies to reveal how each component contributes to the algorithm's final performance. Our model has wide applications, as tree structure exists extensively in real life. Here we show two of them: seeing shape from shadow, and connecting blood vessels from retinal videos.

Our contributions are three-fold. Our main contribution is to show that tiny, barely visible object motion can reveal object structure. Our model can resolve the ambiguity in tree structure estimation using spectral information. Second, we propose a novel, physics-based tree formulation, with which we may estimate the natural frequencies of each sub-branch. Third, we design a hierarchical inference algorithm, using nonparametric Bayesian methods to predict tree structure. Our algorithm achieves good performance on real-world videos.

2 Related Work

Motion for Structured Prediction. Researchers in computer vision have been using motion signals for various tasks [1,34,39,47]. For structured prediction in particular, the layered motion representations [42] have been studied and applied extensively [23,38]. These papers model motion signals in the temporal domain; they are not for scenarios where objects may only have subtle motion differences.

Regarding spectral analysis of motion, the pioneer work of Fleet and Jepson [10] discussed how phase signals could help to estimate object velocity. Gautama and Van Hulle [14] extended the work, proposing a phase-based approach for optical flow estimation. Zhou *et al.* [48] also discussed how phase information helps recognizing object motion. Recently, there have also been a number of works on visualizing and magnifying subtle motion signals from video [7,46], and Rubinstein *et al.* did a thorough review in [35].

The problem of tree structure estimation has been widely studied in computer vision, especially in medical imaging [11,40,41,43], mostly from a static image. In this paper, we explore how motion signals in a video could help in structured prediction, in addition to appearance cues. Though we currently employ a simple and intuitive appearance model, it is straightforward to incorporate more sophisticated appearance models into our approach.

Modeling Tree Vibration. Tree vibration is an important research area in the field of botany [20,31]. Moore and Maguire [31] reviewed the concepts and dynamic studies by examining the natural frequencies and damping ratios of trees in winds. Recently, James *et al.* [20] reviewed tree bio-mechanics studies using dynamic methods of analysis.

Our formulation of tree vibration is based on the lumped-mass procedure. Related literature include spring-mass-damper models for trees as a single mass point [30], or as a complex system of coupled masses that represent the trunk and branches [21,33]. Our formulation also considers a tree as a system of coupled masses, but different from Murphy *et al.* [33] which studied only one-layer structure, we explore hierarchical tree structure of multiple layers.

Bayesian Theory of Perception. Researchers have developed Bayesian theories for human visual perception in general [24,26,32], and for object motion perception in particular [4,44]. Our inference algorithm draws inspirations from the recent hierarchical Bayesian model for object motion from Gershman *et al.* [16], which employs the nested Chinese restaurant process (nCRP) [2] as a prior of object structure.

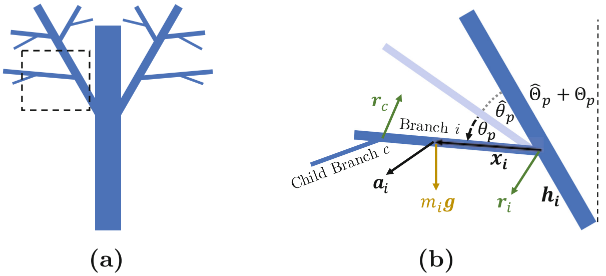


Fig. 2. (a) Hierarchical beam structure. (b) Force analysis for one of the branches (the one marked by dashed rectangle in (a)).

3 Formulation

We here present our formulation that recovers tree structure from the temporal complex spectra of vertices. We start by introducing a physics-based, hierarchical link model, representing a tree as a set of beams with certain mass and stiffness (Fig. 2a). Using this model, we derive a set of ordinary differential equations (ODEs) of node vibrations (Sect. 3.2) and prove an important property (Sect. 3.3): each sub-branch of a tree is a linear time-invariant system under certain assumptions. A Bayesian inference algorithm exploits the property for structure estimation (Sects. 4.1 and 4.2).

3.1 A Physics-Based Link Model

We use a rigid link model to describe the vibration of a tree, as shown in Fig. 2a. In this model, each branch i of the tree is modeled as a rigid beam with a certain mass m_i and length l_i. Under the uniform mass assumption, the center of mass of a branch is at $\frac{l_i}{2}$. Each branch connects to its parent through a torsional spring with stiffness k_i. Our model relates to the simpler, one-layer physical model from

Murphy *et al.* [33], where they attempted to compute the mass and stiffness of all the beams. We observe this to be impractical in real data given the presence of noise and occlusion. Instead, we derive a set of non-linear ordinary derivative equations (ODEs) that describe the relationship between the vibration of a tree and its structure and physical properties.

We describe the vibration of a tree by the deviation angles $\{\theta_i\}$ of branches. As shown in Fig. 2b, let $\hat{\theta}_i$ be the directional angle from vertical line to a branch when the tree is static (no external forces except gravity), and let θ_i be the deviation angle from its static location when the tree is vibrating (θ_i changes over time). To derive the governing equations for θ_i, we start by applying the Newton's law to each branch i, which gives[1]

$$m\boldsymbol{a}_i = -\boldsymbol{r}_i + \sum_{c \in C_i} \boldsymbol{r}_c + m\boldsymbol{g}, \tag{1}$$

where $\boldsymbol{r}_c \in \mathbb{R}^2$ is the force exerted by branch c on its parent, C_i is the set of children of branch i, and $\boldsymbol{g}$ is the acceleration due to gravity. The negative sign before $\boldsymbol{r}_i$ is due to our definition and Newton's third law. Branch i's acceleration $\boldsymbol{a}_i \in \mathbb{R}^2$ is defined as the acceleration of the branch's center of mass.

In addition, we have the rotation equation,

$$I_i \dot{\omega}_i = -k_i\theta_i + \sum_{c \in C_i} k_c\theta_c + \boldsymbol{r}_i \times \boldsymbol{x}_i + \sum_{c \in C_i} \boldsymbol{r}_c \times \boldsymbol{x}_i, \tag{2}$$

where I_i is branch i's moment of inertia when it rotates around its center, $\dot{\omega}_i$ is its angular acceleration, θ_c is branch c's deviation angle, $\boldsymbol{x}_i$ is its movement, and k_i is the stiffness of the torsional spring it connects to. Also, the branch acceleration $\boldsymbol{a}_i$ relates to the acceleration of its endpoint $\boldsymbol{a}_{i_o}$ via

$$\boldsymbol{a}_i = \boldsymbol{a}_{i_o} + \dot{\omega}_i \times \boldsymbol{x}_i + \omega_i \times (\omega_i \times \boldsymbol{x}_i), \tag{3}$$

where $\boldsymbol{a}_{i_o} \in \mathbb{R}^2$ is the acceleration of the junction point.

Therefore, the angular velocity and angular acceleration of branch i are

$$\omega_i = \dot{\theta}_i + \sum_{p \in P_i} \dot{\theta}_p \quad \text{and} \quad \dot{\omega}_i = \ddot{\theta}_i + \sum_{p \in P_i} \ddot{\theta}_p, \tag{4}$$

where P_i is the set of ancestors of branch i. These equations do not include fictitious forces. All quantities are global values under the reference frame.

At last, replacing the branch acceleration ($\boldsymbol{a}_i$ and $\boldsymbol{a}_{i_o}$) and angular acceleration $\dot{\omega}_i$ in Eqs. 1 and 2 using Eqs. 3 and 4, and eliminating forces between branches r_i, we get the ODE with respect to all deviation angles $\{\theta_i\}$,

$$I_i f_i(\ddot{\theta}) = -k_i\theta_i + \sum_{c \in C_i} k_c\theta_c + \boldsymbol{r}_i(\theta, \dot{\theta}, \ddot{\theta}) \times \boldsymbol{x}_i + \sum_{c \in C_i} \boldsymbol{r}_c(\theta, \dot{\theta}, \ddot{\theta}) \times \boldsymbol{x}_i, \tag{5}$$

where $\boldsymbol{r}_i(\theta, \dot{\theta}, \ddot{\theta})$ is a vector functions of θ, $\dot{\theta}$, and $\ddot{\theta}$. Please see our supplementary material for its definition in detail.

[1] In this chapter, we use a lower-case letter a to denote a scalar, a bold lower-case letter $\boldsymbol{a}$ to denote a vector, and a capital letter A to denote a matrix. We denote the matrix product as $A\boldsymbol{b}$, where $A \in \mathbb{R}^{n \times m}$ and $\boldsymbol{b} \in \mathbb{R}^m$.

3.2 ODE of Node Vibration

The ODE (Eq. 5) is highly nonlinear due to sinusoidal and quadratic terms. To solve it, we first linearize the equation around its stable solution. We assume that the deviation angle θ_i of each branch i is small and ignore all $O(\theta_i^2)$ terms. Under this assumption, the quadratic term of angular velocity $O(\dot{\theta}^2)$ can also be ignored, because according to the conservation of energy, the potential energy $\frac{1}{2}k\theta^2$ of a branch is on the same scale of its kinetic energy $\frac{1}{2}I_i\dot{\theta}^2$.

We can now derive a fully linear system under the above assumption as

$$M\ddot{\boldsymbol{\theta}} + K\boldsymbol{\theta} = \mathbf{0}, \tag{6}$$

where M and K are two matrices depending on the structure of a tree and its physical properties, including the moment of inertia (I), mass (m), and stiffness (k) of all branches.

In practice, from an input video, it is easier to measure the 2D shift of each node, rather than the rotation of each branch. To derive the ODE of 2D shifts of all nodes from Eq. 6, we denote node i's 2D location in a stable tree as $\hat{\boldsymbol{y}}_i$, and the 2D shifts from its stable location as $\boldsymbol{y}_i$. We have

$$\boldsymbol{y}_i + \hat{\boldsymbol{y}}_i = \sum_{j \in P_i} l_j \boldsymbol{n}(\theta_j + \hat{\theta}_j), \tag{7}$$

where $\boldsymbol{n}(\theta) = (\cos\theta, \sin\theta)$ and l_j is the length of branch j (recall that P_i is the set of ancestors of branch i). Let $\boldsymbol{y}$ be the concatenation of 2D shifts of all the nodes. Plugging Eqs. 7 to 6, we have

$$N\ddot{\boldsymbol{y}} + L\boldsymbol{y} = \mathbf{0}, \tag{8}$$

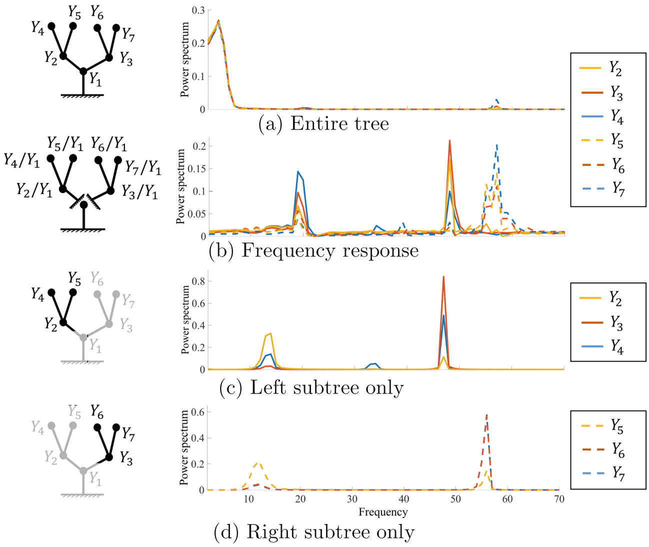
where N and L are matrices depending on M, K, l_j, and θ_j. The constant term must be zero, as $\boldsymbol{y} = \ddot{\boldsymbol{y}} = \mathbf{0}$ when the tree is stable. Please see our supplementary material for a detailed derivation.

3.3 Inferring Modes of Each Sub-branch

Based on the second order ODE, we can infer the modes of each sub-branch and use them to group nodes into branches using the following property.

Property 1 (Each sub-branch is a LTI-system). *Imagine a branch undergoes a forced vibration. Let $y^i_{root}(t)$ and $y^i_{leaf}(t)$ be the displacements of the root and one of its leaf node respectively at time t ($i = 1, 2$). Then, if the displacement of the root is $\alpha_1 \cdot y^1_{root}(t) + \alpha_2 \cdot y^2_{root}(t)$, where $\alpha_1, \alpha_2 \in \mathbb{R}$, the vibration of the leaf is $\alpha_1 \cdot y^1_{leaf}(t) + \alpha_2 \cdot y^2_{leaf}(t)$.*

This is a corollary of Eq. 8, which shows that the displacement of a node satisfies a linear, second order ODE. The system is also time-invariant, as all matrices in Eq. 8 do not change in time.



(a) Entire tree

(b) Frequency response

(c) Left subtree only

(d) Right subtree only

Fig. 3. Spectrum analysis on a synthetic tree. Directly calculating the power spectrum of the vibration of each nodes does not help to infer the tree structure, as all the nodes have similar power spectrum (a). By dividing the spectrum of each node by the spectrum of the root node, we obtain the frequency response of each node. We now clearly see the difference between the two subtrees (b). The modes of each frequency response also match the modes of the free vibration of each subtree (c) and (d).

The key observation of our work is that we can infer the mode of free vibration of each sub-branch as if that sub-branch is disconnected from the rest of the tree, as suggested by Property 1. Let S be a set of nodes in a sub-branch; let $Y_i(\eta)$ be the temporal spectrum of the displacement of the i-th node in that branch ($i \in S$), where η is the frequency index; let Y_{root} be the temporal spectrum of the root displacement. Because each sub-branch is a LTI-system, the frequency response of the sub-branch is

$$\overline{Y}_i(\eta) = \frac{Y_i(\eta)}{Y_{root}(\eta)}, \quad \forall \eta. \tag{9}$$

It is well known that when there is no damping, the natural frequencies of an oscillating system coincide with its resonance frequency [12, Chap. 4]. In our case, this suggests that the natural frequencies of a sub-branch are the same as the modes of the frequency response of that branch[2].

As an illustration, Fig. 3a shows a tree with two sub-branches (Y_{2-4} and Y_{5-7}). All nodes have similar power spectra as their vibrations are dominated by the vibration of the root (Y_1). To distinguish the spectra of the two sub-branches, we calculate the frequency response of each node, *i.e.*, the ratio between the

[2] In the presence of small damping, the difference between the modes of frequency response and the modes of free vibration is also small.

spectrum of the root and the spectrum of each branch. As shown in Fig. 3b, there is a clear difference between the frequency responses of two branches. The modes of each frequency response also match the modes of free vibrations of each sub-branch, as if they are detached from the root (see Fig. 3c and d).

We can then group nodes into different sub-branches based on their spectrum response, because the natural frequencies of each sub-branch depend on its inherent physical properties like mass and stiffness. In practice, the modes of frequency responses are not a robust measure in the presence of noise and damping. Therefore, we group nodes based on their the normalized power spectra and phases instead, with the help of the appearance information described in Sect. 4.1.

4 Algorithm

We now introduce our structure estimation algorithm based on the tree formulation. Our algorithm has two major components: a recognition module that extracts motion and appearance cues from visual input, and an inference module that predicts tree structure.

4.1 Extracting Motion and Appearance Cues

We use an bottom-up recognition algorithm to obtain motion and appearance cues from input videos (Fig. 4a) with a given set of interest points (Fig. 4b).

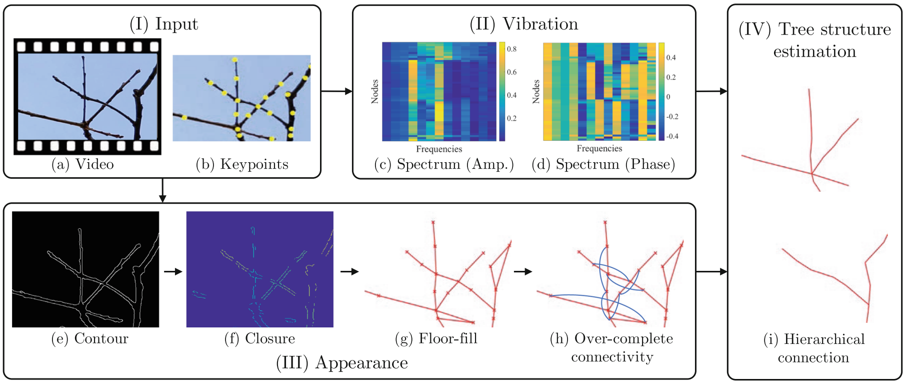


Fig. 4. Overview of our framework. We take a video (a) and a set of keypoints (b) as input (I). We use normalized amplitudes (c) and phases (d) of keypoints as our vibration signals (II); we also obtain appearance cues (III) through several intermediate steps (Sect. 4.1). Finally, we apply our inference algorithm (Sect. 4.2) for tree structure estimation.

Motion. Given an input video, we first manually label all nodes in the first frame and then track them over time using optical flow. There are many tracking algorithms that can extract trajectories of sparse keypoints [18,19,28,36], but we choose to calculate the dense motion field for two reasons. First, most of vibrations are small, and optical flow is known to perform well on capturing the small motion with subpixel accuracy. Second, sparse tracking algorithms, like the KLT tracker [28], might suffer the aperture problem, as most of branches only contain one-dimensional local structure. On the other hand, dense optical flow algorithms aggregate the information from other locations, so it would be more robust to the aperture problem.

Specifically, we first compute a dense flow field from the first frame to one of the frame t in the sequence [27]. We then get trajectory of each node in the sequences from dense motion fields through interpolation. We further apply Fourier transform to the trajectory of each node independently to get its complex spectrum Y (Fig. 4-II), and extract its modes from the fifth order spectral envelope [13]. We use the normalized amplitude (Fig. 4c) and phase (Fig. 4d) of these modes for inference, as discussed in Sect. 4.2.

Appearance. We use an over-complete connectivity matrix as our appearance cues. As shown in Fig. 4-III, we compute the matrix via the following steps: obtaining a contour map, computing the closure of each interest point, flood-filling the contour map from all closures, and adding edges to junctions.

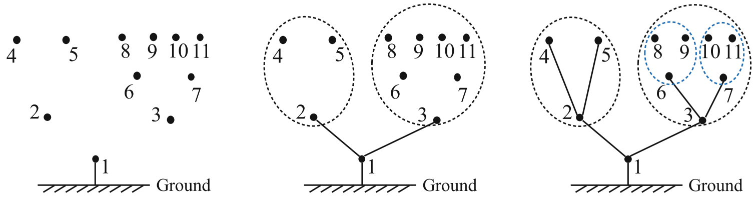
Given the first frame from an input video, we first use Canny edge detector [5] with threshold 0.5 to obtain an initial contour map (Fig. 4e). Then, for each interest point i, we consider all contour pixels S_i whose distance to i is no larger than r_i. We search for the minimum r_i, such that if we connect i to all pixels in S_i, the angle between each two adjacent lines is no larger than $30°$. We call S_i the closure for point i (Fig. 4f).

We then apply a shortest-path algorithm to obtain the connectivity map of all interest points. Our algorithm is a variant of the Dijkstra's algorithm [8], where there is a hypothetical starting point connecting to pixels in the union of all closures with cost 0. The cost between two 8-way adjacent pixels is 0, if they are both on the contour map, or 1 otherwise. The algorithm is then in essence expanding all closures simultaneously. When it finishes, we connect two keypoints if their corresponding closures are adjacent after expansion (Fig. 4g). To balance the expansion rate of each closure, we use a tuple (c_i, d_i) as the entry for any pixel i in the priority queue, where the primary key c_i is the traditional term for the distance on the graph from i to the origin, and the secondary key d_i is the Chebyshev (L_∞) distance between i to the center of its closure.

Finally, an observed junction in a 2D image may be an actual tree fork, or may be just two disconnected, overlapping branches. To deal with the case, for all points that have 4 or more neighbors, we add an edge between each pair of its neighbors whose angle is no smaller than $135°$. This leads to an over-complete connectivity matrix E (Fig. 4h), which we use as our appearance cues.

```
Algorithm cluster($\boldsymbol{Y}, r$)
Data: Nodes with complex spectra $\boldsymbol{Y} = \{Y_i\}$ and the root's index $r$
Calculate the free vibration of each node in this tree
for each node $i$ do
    $Y_i \leftarrow Y_i$ ./ $Y_r$
end
Cluster nodes based on their appearance and frequency
Let $\{S_j\}_{j=1,\cdots,k}$ be all $k$ clusters
for $j = 1, \cdots, k$ do
    Select subroot $r_j$
    Call cluster($\boldsymbol{Y}_{S_j}, r_j$) recursively
end
```

Algorithm 1. Our hierarchical clustering algorithm



(a) Depth 0 **(b)** Depth 1 **(c)** Depth 2

Fig. 5. Illustration of our hierarchical clustering algorithm. See Sect. 4.2 for details.

4.2 Inference

Overview with a Toy Example. We start with a high-level overview of our hierarchical inference algorithm along with a toy tree with three levels of hierarchy (Fig. 5). As shown in Algorithm 1, given the root, our algorithm first computes the free vibration of the rest of nodes (Step I), groups them into several clusters (Step II), and then recursively finds tree structure for each cluster (Step III).

In this toy tree with v_1 as the root, the algorithm groups the other nodes into two clusters: (v_2, v_4, v_5) and $(v_3, v_6, v_7, \ldots, v_{11})$, as shown in Fig. 5b. For each subtree, the algorithm recursively applies itself for finer-level tree structure. Here in the right branch, we get two level-2 subtrees (v_6, v_8, v_9) and (v_7, v_{10}, v_{11}).

Step I: Computing Free Vibration. We first compute the vibration of each node given the root. Based on Eq. 9, we divide the complex spectrum of each leaf node by the complex spectrum of the root. Note that under certain frequency, the complex spectrum of the root might be close to zero. Therefore, a direct division might magnify the noise. To deal with this, we calculate the spectrum of each node i after removing the root r via $Y_i \cdot Y_r^* / \left(|Y_r|^2 + \epsilon^2\right)$, where Y_r^* is the complex conjugate of Y_r, and ϵ controls the noise level. This is similar to the Weinner filter [45]. When $\epsilon = 0$, We have the normal division as

$$\frac{Y_i \cdot Y_r^*}{|Y_r|^2} = \frac{Y_i \cdot Y_r^*}{Y_r \cdot Y_r^*} = \frac{Y_i}{Y_r}. \tag{10}$$

Step II: Grouping Nodes. We group nodes into clusters $\{S_j\}$ under the assumption that nodes in each cluster share similar vibration patterns (complex frequencies) and appearance cues. Each node has an unknown number of children, we use a Chinese Restaurant Process (CRP) prior [2] over the tree structure. Let z_i be the index of cluster that node i is assigned to, and let $Z = \{z_i\}$ be the assignment of all nodes. The joint probability of assignment is

$$P(Z|E, Y) \propto \; P_{\text{CRP}}(Z) \cdot P_m(Y|Z) \cdot P_a(E|Z), \tag{11}$$

where $P_{\text{CRP}}(\cdot)$ is the CRP prior, $P_m(\cdot)$ is the likelihood based on motion, and $P_a(\cdot)$ is the likelihood based on appearance.

Motion Term: we use two statistics of the spectrum: the normalized amplitude $Y_i^n = |Y_i|/\|Y_i\|_2$ and the phase $Y_i^p = \text{angle}(Y_i)$. Our motion term is

$$\log P_m(Y|Z) = \sum_i -\sigma_n^{-2} \|Y_i^n - C_{z_i}^n\|_2^2 - \sigma_p^{-2} \|Y_i^p - C_{z_i}^p\|_2^2. \tag{12}$$

C_k^n and C_k^p are the mean normalized amplitudes and phases of nodes in cluster k.

Appearance Term: nodes in the same sub-branch are expected to be connected to each other and to the root. To this end, we define the appearance term as

$$\log P_a(E|Z) = \sum_{z_i = z_j} \alpha \cdot \mathbf{1}(i, j|Z, E) + \sum_i \beta \cdot \mathbf{1}(i, r|Z, E), \tag{13}$$

where $\mathbf{1}(i, j)$ is the indicator function of whether there exists a path between nodes i and j given the current assignment Z and the estimated connectivity matrix E (see Sect. 4.1). Given the joint probability in Eq. 11, we run Gibbs sampling [15] for 20 iterations over each assignment z_i.

Step III: Recursion. As shown in the toy example (Fig. 5), for each cluster S_j, our algorithm selects the node closest to the root r in the Euclidean space as the subroot r_j. It then infers subtree structure for S_j recursively. The whole inference algorithm takes 3–5 s for a tree of 50 vertices on a Desktop CPU.

5 Evaluations

We now present how we use simulation to verify our formulation (Sect. 3), and show qualitative and quantitative results on videos of artificial and real trees.

5.1 Simulation

Based on formulation described in Sect. 3.1, we implemented a tree simulator by solving Eq. 5 using the Euler Method [9]. As shown in Sect. 4.1, the analytic form of ODE is very complicated. Therefore, we do not eliminate all the redundant variables, including the acceleration of the branch ($\boldsymbol{a}_i$ and $\boldsymbol{a}_{i_o}$), forces between branches ($\mathbf{r}_i$), and angular velocity of each branch (ω_i). Instead, we directly solve Eqs. 1 and 2 numerically. Also, to increase the stability of Euler method in presence of numerical error, we force the system to have constant total energy for every time-stepping update. If the system's energy increases during an update, we rescale the kinetic and potential energy of each branch to ensure that the total energy of the system is constant. This makes our simulation robust and stable. See the supplementary material for the detailed derivation.

Figure 6 shows the vibration modes of a simulated tree (left) with three mode shapes (right). Here we manually specify the structure of the tree and physical property of each branch, including mass, stiffness, and length, and numerically solve for the rotation angle of each branch. The mode of power spectra (the natural frequencies) of the trunk and two branches matches the three mode shapes of the tree, which is consistent with the theory in Sect. 3.

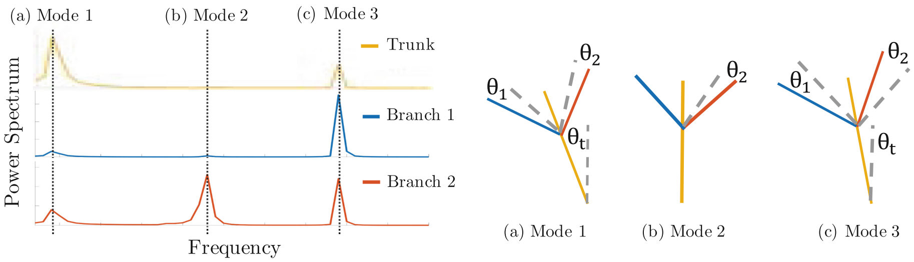


Fig. 6. Mode shapes. The left three curves show the power spectra of the trunk and the two branches. The three mode shapes extracted from vibration are shown on the right.

5.2 Real, Normal Speed Videos

Data. We record videos of both artificial and real trees. For artificial trees, we take 3 videos in an indoor lab environment, where wind is generated by a fan. We take 8 videos of outdoor real trees. All videos are taken at 24 frames per second by a Canon EOS 6D DSLR camera, with a resolution of 1920 × 1080.

Methods. We compare our full model, which makes use of appearance and vibration cues jointly (appearance + motion), with a simplified variant, which uses only appearance information, but ignores all motion signals during inference. We also compare with three alternative approaches for hierarchical structure recovery from spatial-temporal motion signals.

- **Appearance + Flow/Tracking:** We replace the spatial-temporal feature in our algorithm by motion recovered by either optical flow or a KLT tracker.
- **Hierarchical motion segmentation:** We use the popular hierarchical video segmentation algorithm [17] to obtain image segments and their structure. We then derive the tree structure from the segment hierarchy.

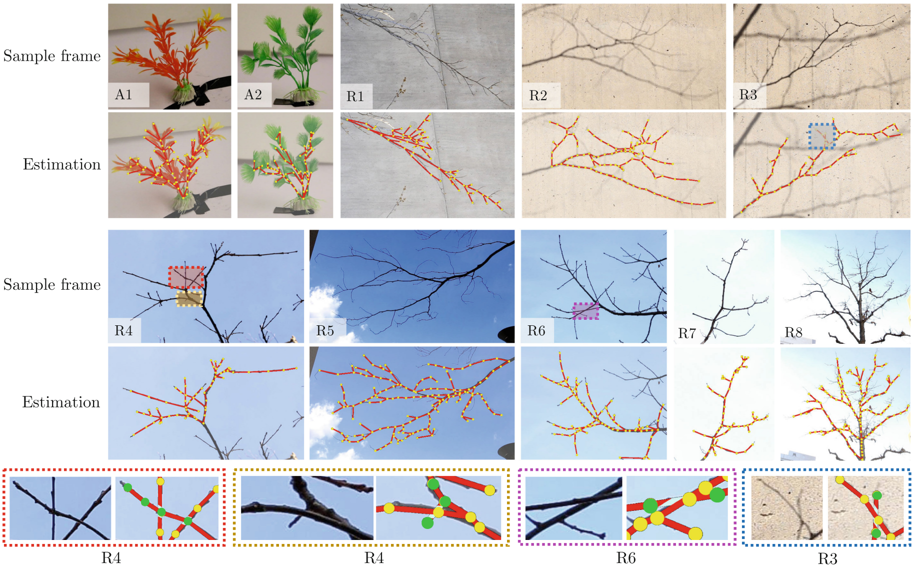


Fig. 7. Estimated tree structure on real videos. A1–A2: on artificial trees; R1–R8: on real trees. At bottom, we show cases where appearance is insufficient for inferring the correct structure. Using vibration signals, our algorithm works well in these cases.

Results. Figure 7 shows that our algorithm works well on real videos. Results in the bottom row suggest that our algorithm can deal with challenging cases. Using motion signals, it correctly recovers the structure of occluded twigs, which is indistinguishable from pure visual appearance.

For quantitative evaluations, we manually label the parents of each node and use it as ground truth. We use two metrics. In Table 1, we evaluate different methods in (a) the percentage of nodes whose parents are correctly recovered and (b) minimum edit distance—the minimum edges that need to be displaced to make the predicted tree and the ground truth identical. Our algorithm achieves good performance in general. Including motion cues consistently improves the accuracy of the inference on videos of all types, and spatial feature significantly out-performs the raw motion signal.

Table 1. Results evaluated by the percentage of nodes whose parents are correctly recovered (top) and the edit distance between reconstruction and ground truth (bottom). Our method outperforms the alternatives in most cases.

Metrics	Methods	Artificial			Real trees								High-speed videos						Avg.
		A1	A2	A3	R1	R2	R3	R4	R5	R6	R7	R8	H1	H2	H3	H4	H5	H6	
Acc. (%)	MoSeg	33	37	73	50	65	84	56	56	68	70	74	47	57	46	47	43	51	56.3
	Appear.	40	31	90	67	59	83	70	66	71	89	85	55	56	62	66	61	69	65.7
	A+Flow	43	32	92	79	69	83	85	75	84	95	**94**	64	58	69	69	72	**76**	73.5
	A+Track	38	**46**	88	79	63	83	84	**83**	**88**	89	93	67	64	66	76	71	**76**	73.8
	Ours	**54**	45	**100**	**81**	**76**	**94**	**95**	**83**	**88**	**97**	**94**	**69**	**69**	**72**	**77**	**74**	70	**79.3**
Edit Dis.	MoSeg	26	16	7	25	22	8	16	20	13	8	15	20	21	24	22	17	15	17.4
	Appear.	20	21	3	12	19	5	5	30	9	4	8	16	16	16	16	16	16	13.7
	A+Flow	19	13	2	**7**	13	5	3	12	11	**1**	**6**	13	18	11	12	**8**	**8**	9.5
	A+Track	24	**10**	2	10	16	6	4	12	8	4	7	13	15	12	10	10	**8**	10.1
	Ours	**14**	12	**0**	8	**12**	**2**	**0**	**6**	**4**	**1**	**6**	**10**	**12**	**9**	**6**	9	**8**	**7.0**

5.3 Real, High-Speed Videos

Experimental Setup. To understand and analyze motion, we take high-speed videos of trees using an Edgertronic high-speed camera. We captured 1 normal-speed video (30 FPS) and 5 high-speed videos with a frame rate varying from 60 to 500 FPS, each of which contains 1,000 frames. For each video, we manually label around 100 interest points and their connections. Intuitively, the root branches should have higher stiffness and lower natural frequencies. Therefore, low-frame-rate videos should provide more information about the tree's main structure, whose natural frequency is low, and high-frame-rate videos should provide more information of fast vibrating thin structure.

Evaluation. For evaluation, we first pick two points (P_1 and P_2 in Fig. 8c) on two major branches of the tree and compare their power spectra as shown in

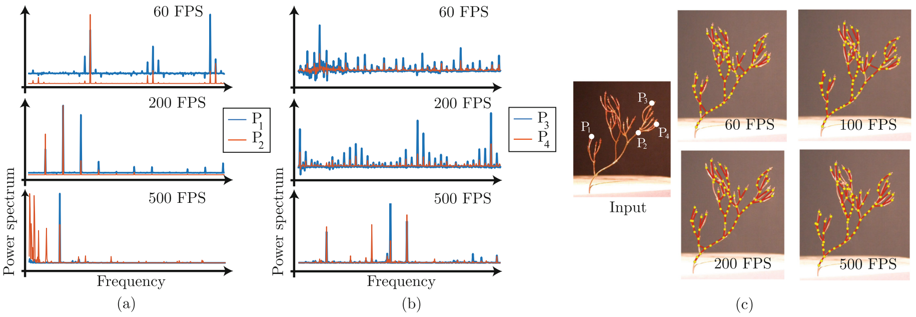


Fig. 8. Evaluation of the algorithm on videos with different frame rates. (a) and (b) shows the power spectra of selected nodes in the input videos captured at different frame rates, and (c) shows the estimated tree structures. See Sect. 5.2 for more details.

Fig. 8a. At 60 FPS, the power spectra of these two nodes are different for a wide range of frequencies; at 500 FPS, they are only different at lower frequencies, as the natural frequencies of the main branches are low. We then pick two points (P_3 and P_4 in Fig. 8c) on two small branches of the tree and compare their power spectra (see Fig. 8b). Now in both 60 FPS and 200 FPS videos, their spectra are similar, and the difference in modes only become significant at 500 FPS. Figure 8c shows that the estimation errors from low-frame videos (60 or 100 FPS) on the top-right corner no longer exist when the input is at 500 FPS, indicating high-speed videos are better for estimating fine structure. All these results are consistent with our theory. H1 to H6 in Tables 1 refer to videos captured at 30, 60, 100, 200, 400, 500 FPS, respectively.

6 Applications

Our model has wide applications in inferring tree-shaped structure in real-life scenarios. To demonstrate this, we show two applications: seeing object structure from shadows, and inferring blood vessels from retinal videos.

Shapes from Shadows. In circumstances like video surveillance, often the only available data is videos of projections of an object, but not the object itself. For example, we can see the shadows of trees in the video, but not the trees themselves. In these cases, it would be of strong interests to reconstruct the actual shape of the object. Our algorithm deals with these cases well. Among the eight real videos in Fig. 7, R2 and R3 are videos of tree shadows. Our algorithm successfully reconstructs the underlying tree structure, as shown in Fig. 7 and Table 1.

Vessels from Retinal Videos. Our model can contribute to biomedical research. We apply our model on a retinal video from OcuScience LLC. As shown in Fig. 9a–b, our algorithm performs well, reconstructing the connection among retinal vessels despite limited video quality. It achieves a smaller edit distance (4) compared with A+Flow (7) and A+Track (6).

Fully Automatic Recovery. While we choose to take keypoints as input to provide users with extra flexibility and to increase prediction accuracy, following

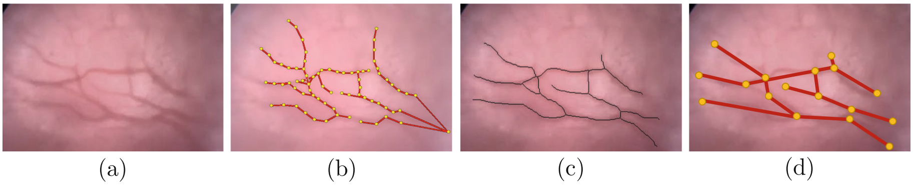


Fig. 9. Our result on a retinal video. (a) A frame from the input video. (b) Our model reconstructs the structure of blood vessels despite low video quality. (c–d) Results on fully automatic structure inference, where (c) shows the estimated object skeleton and (d) shows the object structure inferred by our model.

the convention in the literature [40], our system can be easily extended to become fully automatic. Here we provide an additional experiment on the retinal video. We first apply the segmentation method from Maninis *et al.* [29] on the first frame to obtain a segmentation of vessels. We then employ the classical skeletonization algorithm from Lee *et al.* [25] (Fig. 9c), and use the endpoints and junctions of the obtained skeleton as input keypoints to our model. As shown in Fig. 9d, our system works well without manual labels.

7 Discussion

In this paper, we have demonstrated that vibration signals in the spectral domain, in addition to appearance cues, can help to resolve the ambiguity in tree structure estimation. We designed a novel formulation of trees from physics-based link models, from which we distilled physical properties of vibration signals, and verified them both theoretically and experimentally. We also proposed a hierarchical inference algorithm, using nonparametric Bayesian methods to infer tree structure. The algorithm works well on real-world videos.

Our derivation makes four assumptions: passive motion, small vibration, no damping, and a known root. While real trees often satisfy the first two, they do not have zero damping (damping ratio ranging from 1.2% to 15.4% [22]). In these cases, our algorithm still successfully recovers their geometry from vibration. When the root is unknown, our method can discover multiple subtrees from a virtual root with a uniform motion spectrum. On the other hand, our model performs less well when assumptions are significantly violated (*e.g.*, large vibration or an incorrect root).

We see our work as an initial exploration on how spectral knowledge may help structured inference, and look forward to its potential applications in fields even outside computer science, *e.g.*, fiber structure estimation.

Acknowledgements. This work is supported by NSF #1231216, #1212849, and #1447476, ONR MURI N00014-16-1-2007, Toyota Research Institute, Shell Research, and Facebook. We thank Xiuming Zhang for helpful discussions.

References

1. Bascle, B., Blake, A., Zisserman, A.: Motion deblurring and super-resolution from an image sequence. In: Buxton, B., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 571–582. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61123-1_171
2. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. JACM **57**(2), 7 (2010)
3. Bouman, K.L., Xiao, B., Battaglia, P., Freeman, W.T.: Estimating the material properties of fabric from video. In: ICCV (2013)
4. Braddick, O.: Segmentation versus integration in visual motion processing. Trends Neurosci. **16**(7), 263–268 (1993)

5. Canny, J.: A computational approach to edge detection. IEEE TPAMI **8**(6), 679–698 (1986)
6. Davies, M.N., Green, P.R.: Perception and Motor Control in Birds: an Ecological Approach. Springer, Heidelberg (2012)
7. Davis, A., Bouman, K.L., Chen, J.G., Rubinstein, M., Durand, F., Freeman, W.T.: Visual vibrometry: estimating material properties from small motion in video. In: CVPR (2015)
8. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
9. Farlow, S.J.: Partial Differential Equations for Scientists and Engineers. Courier Corporation, North Chelmsford (1993)
10. Fleet, D.J., Jepson, A.D.: Computation of component image velocity from local phase information. IJCV **5**(1), 77–104 (1990)
11. Fraz, M.M., et al.: Blood vessel segmentation methodologies in retinal images-a survey. Comput. Methods Programs Biomed. **108**(1), 407–433 (2012)
12. French, A.: Vibrations and Waves. WW Norton, New York (1971)
13. Furoh, T., Fukumori, T., Nakayama, M., Nishiura, T.: Detection for lombard speech with second-order mel-frequency cepstral coefficient and spectral envelope in beginning of talking-speech. J. Acoust. Soc. Am. **133**(5), 3246 (2013)
14. Gautama, T., Van Hulle, M.: A phase-based approach to the estimation of the optical flow field using spatial filtering. IEEE TNN **13**(5), 1127–1136 (2002)
15. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE TPAMI **6**(6), 721–741 (1984)
16. Gershman, S.J., Tenenbaum, J.B., Jäkel, F.: Discovering hierarchical motion structure. Vis. Res. **126**, 232–241 (2016)
17. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR (2010)
18. Hare, S., et al.: Struck: structured output tracking with kernels. IEEE TPAMI **38**(10), 2096–2109 (2016)
19. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE TPAMI **37**(3), 583–596 (2015)
20. James, K.R., Dahle, G.A., Grabosky, J., Kane, B., Detter, A.: Tree biomechanics literature review: dynamics. J. Arboric. Urban For. **40**, 1–15 (2014)
21. James, K.R., Haritos, N., Ades, P.K.: Mechanical stability of trees under dynamic loads. Am. J. Bot. **93**(10), 1522–1530 (2006)
22. James, K., Haritos, N.: Branches and damping on trees in winds. In: Australasian Conference on the Mechanics of Structures and Materials (2014)
23. Jepson, A.D., Fleet, D.J., Black, M.J.: A layered motion representation with occlusion and compact spatial support. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 692–706. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47969-4_46
24. Knill, D.C., Richards, W.: Perception as Bayesian inference. Cambridge University Press, Cambridge (1996)
25. Lee, T.C.: Building skeleton models via 3-D medial surface axis thinning algorithms. CVGIP **56**(6), 462–478 (1994)
26. Lee, T.S., Mumford, D.: Hierarchical bayesian inference in the visual cortex. JOSA A **20**(7), 1434–1448 (2003)
27. Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Citeseer (2009)
28. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI (1981)

29. Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., Van Gool, L.: Deep retinal image understanding. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 140–148. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_17
30. Miller, L.A.: Structural dynamics and resonance in plants with nonlinear stiffness. J. Theor. Biol. **234**(4), 511–524 (2005)
31. Moore, J.R., Maguire, D.A.: Natural sway frequencies and damping ratios of trees: concepts, review and synthesis of previous studies. Trees **18**(2), 195–203 (2004)
32. Moreno-Bote, R., Knill, D.C., Pouget, A.: Bayesian sampling in visual perception. PNAS **108**(30), 12491–12496 (2011)
33. Murphy, K.D., Rudnicki, M.: A physics-based link model for tree vibrations. Am. J. Bot. **99**(12), 1918–1929 (2012)
34. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: CVPR (2017)
35. Rubinstein, M.: Analysis and visualization of temporal variations in video. Ph.D. thesis, MIT (2013)
36. Rubinstein, M., Liu, C., Freeman, W.T.: Towards longer long-range motion trajectories. In: BMVC (2012)
37. Spelke, E.S., Breinlinger, K., Macomber, J., Jacobson, K.: Origins of knowledge. Psychol. Rev. **99**(4), 605 (1992)
38. Sun, D., Liu, C., Pfister, H.: Local layering for joint motion estimation and occlusion detection. In: CVPR (2014)
39. Sun, D., Sudderth, E.B., Black, M.J.: Layered segmentation and optical flow estimation over time. In: CVPR (2012)
40. Türetken, E., Benmansour, F., Andres, B., Głowacki, P., et al.: Reconstructing curvilinear networks using path classifiers and integer programming. IEEE TPAMI **38**(12), 2515–2530 (2016)
41. 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)
42. Wang, J.Y., Adelson, E.H.: Layered representation for motion analysis. In: CVPR (1993)
43. Wang, Y., Narayanaswamy, A., Roysam, B.: Novel 4-D open-curve active contour and curve completion approach for automated tree structure extraction. In: CVPR (2011)
44. Weiss, Y., Adelson, E.H.: Slow and smooth: a Bayesian theory for the combination of local motion signals in human vision. Technical report, MIT (1998)
45. Wiener, N.: Extrapolation, Interpolation, and Smoothing of Stationary Time Series: with Engineering Applications. MIT Press, Cambridge (1949)
46. Wu, H.Y., Rubinstein, M., Shih, E., Guttag, J., Durand, F., Freeman, W.: Eulerian video magnification for revealing subtle changes in the world. ACM TOG **31**(4), 65 (2012)
47. Xue, T., Rubinstein, M., Liu, C., Freeman, W.T.: A computational approach for obstruction-free photography. ACM TOG **34**(4), 79 (2015)
48. Zhou, B., Hou, X., Zhang, L.: A phase discrepancy analysis of object motion. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6494, pp. 225–238. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19318-7_18