This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2680444, IEEE Transactions on Pattern Analysis and Machine Intelligence

Reconstructing Evolving Tree Structures in Time Lapse Sequences by Enforcing Time-Consistency

Przemysław Głowacki, Miguel Amável Pinheiro, Agata Mosinska, Engin Türetken, Daniel Lebrecht, Raphael Sznitman, Anthony Holtmaat, Jan Kybic, and Pascal Fua, *Fellow, IEEE*

Abstract—We propose a novel approach to reconstructing curvilinear tree structures evolving over time, such as road networks in 2D aerial images or neural structures in 3D microscopy stacks acquired *in vivo*. To enforce temporal consistency, we simultaneously process all images in a sequence, as opposed to reconstructing structures of interest in each image independently. We formulate the problem as a Quadratic Mixed Integer Program and demonstrate the additional robustness that comes from using all available visual clues at once, instead of working frame by frame. Furthermore, when the linear structures undergo local changes over time, our approach automatically detects them.

Index Terms—Curvilinear networks, tubular structures, curvilinear structures, automated reconstruction, temporal consistency, integer programming.

1 Introduction

Reliably reconstructing networks of curvilinear structures from images remains an open Computer Vision problem, even though it has now been studied for more than 30 years [1], [2]. Furthermore, it has mostly been addressed in terms of modeling structures that have been captured at a specific moment in time. However, these networks, be they made of axons and dendrites seen *in vivo* in optical microscopy image stacks [3], blood vessels in retinal-scans [4], or roads in aerial images taken at long intervals, evolve over time. In this paper, we therefore propose an approach to reconstructing such evolving tree structures while enforcing temporal consistency.

To this end, we first process individual images to find pixels or voxels that are very likely to be on the centerlines of linear structures. Finding tree structures in individual images could then be achieved by minimizing an appropriate objective function [5] but would not leverage the fact that portions of the linear structures are often stable over time. To exploit this, we find centerline points that correspond to identical features across time instances and connect these temporal correspondences by *temporal edges*. Combining both types of edges yields a *spatio-temporal* graph that lets us incorporate into our objective function terms that enforce temporal consistency.

Fig. 1 illustrates our approach, which we first introduced in a conference paper [6] using relatively local consistency constraints. The ones we use here are much longer range. We will show that this allows us both to enforce temporal consistency over the stable parts of the structure and to reliably detect changes elsewhere.

Even though imposing temporal consistency is well-known to increase the robustness of video-based object tracking [7] or 3D body pose estimation [8], [9], we do not know of any other automated delineation algorithm besides our own that exploits it early on when working with time-lapse imagery. A more typical approach is that of [10] in which dendritic spines are traced individually in time-lapse images. Only then is spatial and structural information between two dendritic structures used to establish correspondences.

Our contribution is a novel approach to modeling tree structures over several images simultaneously while enforcing temporal consistency. We will demonstrate its power on sequences of *in vivo* two-photon micrographs of neuronal networks and on road networks that change over time.

2 Approach

For many tree structures evolving over time, frame-to-frame changes tend to be local while the global topology remains relatively stable. For example, consider the neural network of Fig. 1, which was imaged *in vivo* over several weeks. The tips of some branches undergo modifications but the rest

- *Przemysław Głowacki, Agata Mosinska, and Pascal Fua are with the Computer Vision Laboratory, EPFL, Lausanne, Switzerland.*
  E-mail: przemyslaw.glowacki@epfl.ch, pascal.fua@epfl.ch
- *Miguel Amável Pinheiro and Jan Kybic are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.*
  E-mail: amavemig@cmp.felk.cvut.cz, kybic@fel.cvut.cz
- *Engin Türetken is with CSEM, Neuchâtel, Switzerland.*
  E-mail: engin.tueretken@alumni.epfl.ch
- *Daniel Lebrecht and Anthony Holtmaat are with the Department of Basic Neurosciences, Université de Genève, Switzerland.*
  E-mail: daniel.lebrecht@unige.ch, anthony.holtmaat@unige.ch
- *Raphael Sznitman is with ARTORG Center for Biomedical Engineering Research, Universität Bern, Switzerland.*
  E-mail: raphael.sznitman@artorg.unibe.ch

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2680444, IEEE Transactions on Pattern Analysis and Machine Intelligence

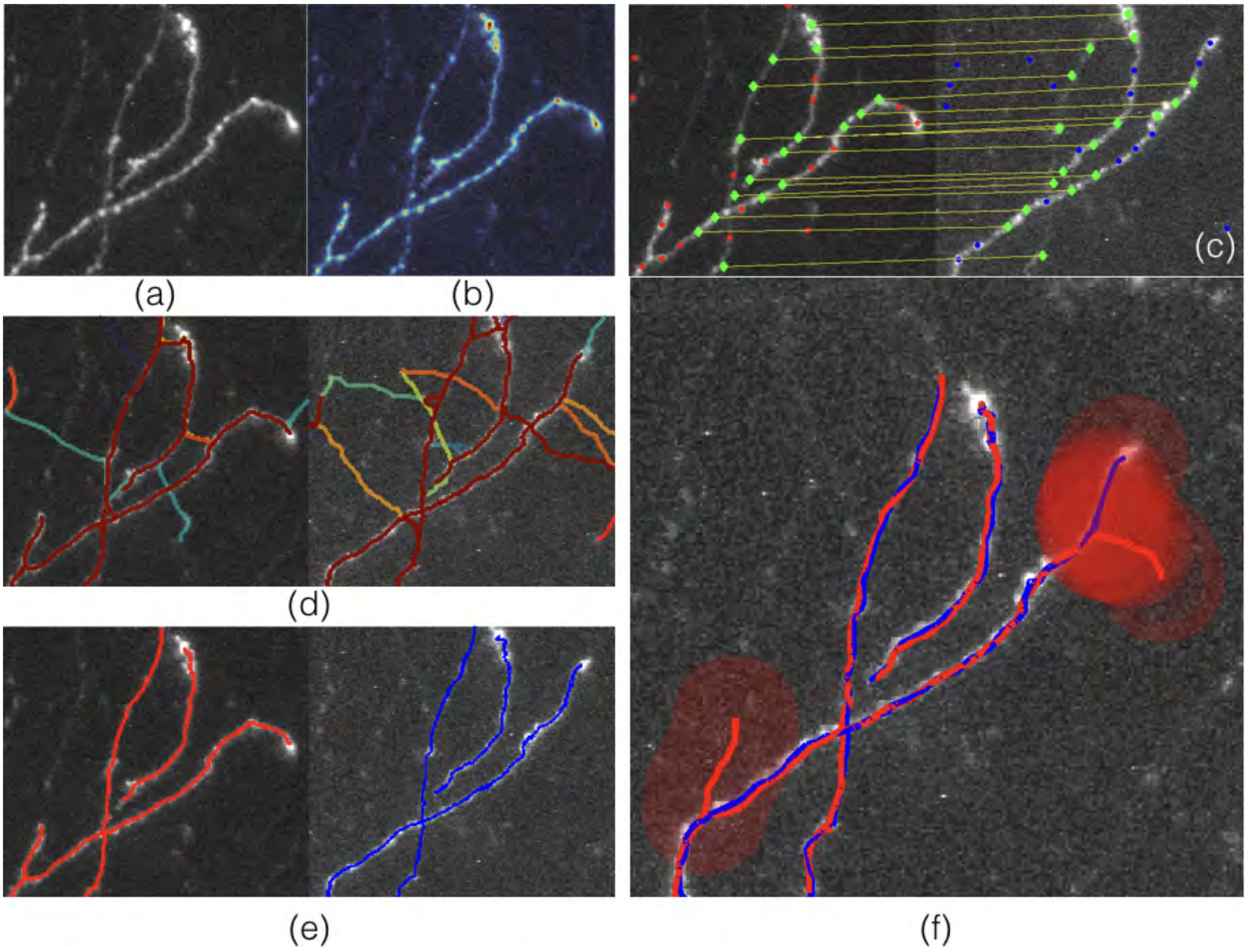

Fig. 1. Key algorithmic steps, best viewed in color. (a) Maximum intensity projection of one of three *in vivo* image-stacks of a neural network taken at one week intervals. (b) Corresponding tubularity image. (c) Maxima of tubularity selected as graph nodes in two different stacks. Those shown in green have been determined to correspond to the same location in both, while those in red or blue appear in only one. (d) Connecting neighboring nodes by high-tubularity paths produces a spatial graph in each image. High-quality paths are shown as red while low quality ones appear as blue. (e) Connecting the corresponding vertices across images turns the spatial graphs into a single spatio-temporal one and solving the resulting QMIP problem yields two temporally consistent trees. (f) The red tree from the first image can be deformed and superposed on the blue tree in the second one, making the changes highlighted in red easy to detect.

stays mostly the same. To exploit this overall consistency while allowing some degree of change, we propose the following approach.

Given N consecutive D-dimensional images $\mathcal{I} = \{I^n\}_{n=1}^N$ featuring evolving tree structures, our goal is to reconstruct a set of trees in each individual image such that they collectively form a temporally consistent sequence. By this, we mean that branches do not appear or disappear randomly and the topology is preserved from one time frame to the next. As a starting point, we find corresponding points across images and use them as nodes of a graph whose edges connect to nodes within the same image or to other images. As in [5], the final set of trees can then be reconstructed by solving a Quadratic Mixed Integer Program (QMIP) problem.

We now summarize the method of [5] and extend it to take into account temporal-consistency constraints.

2.1 Reconstruction without Time Consistency

The method of [5] was designed to handle single images. For a given image I, a local scale-space tubularity measure is computed for every pixel, in the case of 2D data, or voxel, in the case of 3D data, using the oriented flux cross-section trace measure [11]. It expresses how likely it is that a given spacial position lies on a centerline of a tubular structure of a specific radius. A set of evenly distributed sample points $\mathcal{X} = \{\mathbf{x}_i\}$ is selected by first thresholding the tubularity image and then iteratively choosing the highest tubularity point and suppressing its neighborhood until no non-zero tubularity points are left. A number of tree roots are also manually annotated by a human operator. A *spatial graph*, $\mathcal{G} = (\mathcal{X}, \mathcal{E}_s)$, is then built taking the manually anotated roots and automatically selected sample points as vertices. Every two vertices that are close to each other are connected by two oppositely directed edges. For every pair of consecutive edges $\mathbf{e}_{ij}, \mathbf{e}_{jk} \in \mathcal{E}_s$ in the graph a probability score p_{ijk} is computed to assess how likely it is that the underlying tubular path is indeed part of the solution.

The final reconstruction is then obtained by choosing a subset of edges from the graph $\mathcal{G}$ that forms the most likely set of trees. Formally, the problem is formulated as a QMIP with binary variables y_{ij} indicating whether the edge $\mathbf{e}_{ij}$

is part of the solution. A set of constraints adapted from [12] ensures that the result truly is a set of trees emanating from the manually annotated root vertices.

2.2 Simultaneous Reconstruction in all Images

When dealing with image sequences depicting the same region of interest at different times one can process them image by image using the approach described above. This, however, ignores constraints arising from temporal consistency. To account for them, we perform the initial sampling of tubularity images in all images simultaneously while trying to match corresponding sample points between consecutive images. We then create a *spatio-temporal* graph comprising all the sample points at all times. For convenience we connect all of them to an imaginary root vertex. In addition to the *spatial edges* connecting neighboring vertices in specific images, we create *temporal edges* between matching vertices in consecutive images. Finally, we reconstruct the trees in all images simultaneously and enforce time consistency by favoring those whose topology is similar. Our approach goes through four steps:

1) Find graph nodes in individual images as tubularity maxima and corresponding nodes in other images, as in Fig. 1(c).
2) Build a spatio-temporal graph such as the one depicted in Fig. 1(d) by linking nodes both within images when they are close enough and across images when they match.
3) Minimize a quadratic objective function to find a set of trees whose local topology is temporally consistent, such as those of Fig. 1(e).
4) Align these trees spatially to identify places where substantial changes have occurred, as can be seen in Fig. 1(f).

In the following two sections, we first discuss how we build our spatio-temporal graphs and then define the corresponding objective function to be minimized.

3 Building Spatio-Temporal Graphs

The first step in building our spatio-temporal graph is to find corresponding nodes across images, such as those shown in Fig. 1(c). We assume that there may be some non-linear deformation from one image to the next but that it is smooth.

Finding an Initial Set of Correspondences. We first use the Scale-Space Distance Transform method of [13] to compute a tubularity measure in each image independently.

Then, at each successive iteration m, we find the point $\mathbf{x}_m^n$ that maximizes the tubularity across all images, where n refers to the image in which it was found. Since we expect only relatively small displacements from one image to the next, we do not model rotations. Instead, for each one of the remaining images $I^{\bar{n}} \in \mathcal{I} \backslash I^n$, we compute the Normalized Cross Correlation (NCC) between a square or cubic patch centered around $\mathbf{x}_m^n$ in I^n and similar patches in $I^{\bar{n}}$ centered in the neighborhood of that location. Finally, we retain the location $\mathbf{x}_m^{\bar{n}}$ that yields the largest NCC value. We keep such correspondences between consecutive pairs of points $\{\mathbf{x}_m^{n'} \leftrightarrow \mathbf{x}_m^{n'+1}\}_{1 \leq n' \leq N-1}$, as illustrated by Fig. 1(c). Once computed, the tubularity is set to zero in both the neighborhood of $\mathbf{x}_m^n$ and that of the corresponding points. The procedure is then iterated until the tubularity of the selected point $\mathbf{x}_m^n$ falls below a threshold.

Enforcing Geometric Consistency. The procedure described above only relies on NCC scores computed locally and does not guarantee that the displacements of neighboring points are spatially consistent with each other. To enforce this and remove potential mismatches, we use a Gaussian Processes Regression (GPR) [14] to remove correspondences inconsistent with a locally smooth deformation model.

Hence, to find a geometrically consistent set of correspondences $\mathcal{S}_n$ between images I^n and I^{n+1}, we first select from our correspondences a set $\mathcal{S}_n^0 = \{\mathbf{x}_l^n \leftrightarrow \mathbf{x}_l^{n+1}\}_{1 \leq l \leq L}$ of the L points with the highest average local tubularity. In the example of Fig. 2 (Iteration #1), the selected $\mathbf{x}_l^n$ points are shown in green. We treat $\mathcal{S}_n^0$ as being a reliable set and use the GPR to estimate the mean and covariance of the location of a point $\mathbf{x}^n$ in I^{n+1}. This can be computed as

$$
\begin{aligned}
m_{\mathcal{S}_n^0}(\mathbf{x}^n) &= \mathbf{k}' \boldsymbol{\Gamma}_{\mathcal{S}_n^0}^{-1} \mathbf{X}_{\mathcal{S}_n^0}^{n+1} \quad , \qquad (1) \\
\sigma_{\mathcal{S}_n^0}^2(\mathbf{x}^n) &= k(\mathbf{x}^n, \mathbf{x}^n) + \beta^{-1} \; - \mathbf{k}' \boldsymbol{\Gamma}_{\mathcal{S}_n^0}^{-1} \mathbf{k} \quad ,
\end{aligned}
$$

where k is a kernel function that implicitly defines a mapping composed of an affine and a non-linear transformation as in [15], [16], β^{-1} is a measurement noise variance, $\boldsymbol{\Gamma}_{\mathcal{S}_n^0}$ is the $L \times L$ symmetric matrix with elements $\Gamma_{i,j} = k(\mathbf{x}_i^n, \mathbf{x}_j^n) + \beta^{-1}\delta_{i,j}$, $\mathbf{k}$ is the vector $[k(\mathbf{x}_1^n, \mathbf{x}^n), \ldots, k(\mathbf{x}_L^n, \mathbf{x}^n)]^T$ and $\mathbf{X}_{\mathcal{S}_n^0}^{n+1}$ is the $L \times D$ matrix $[\mathbf{x}_1^{n+1}, \ldots, \mathbf{x}_L^{n+1}]^T$.

We then add all correspondences that are consistent with this GPR to $\mathcal{S}_n^0$. A correspondence is considered to be valid if the Mahalanobis distance between corresponding points $\mathbf{x}^{n+1}$ and $m_{\mathcal{S}_n^0}(\mathbf{x}^n)$ is small enough. This gives us an augmented correspondence set $\mathcal{S}_n^1$, such as the one depicted by Fig. 2 (Iteration #2). We then repeat the process using $\mathcal{S}_n^1$ to compute the regression of Eq. 1 and iterate until the set stabilizes, typically after 4 to 5 iterations, as shown in Fig. 2 (Iteration #3). We do this for each consecutive image pair. It yields sets of image points $\mathcal{X}^n = \{\mathbf{x}_i^n\}$ and of geometrically consistent correspondences $\mathcal{S}_n$.

This procedure is greedy in the sense that it starts with a set of high-confidence correspondences and progressively adds new lower-confidence ones. However, recomputing the GPR at every iteration to eliminate outliers gives it the necessary robustness.

Building the Graph. We treat points in all the $\mathcal{X}^n$ as nodes of our graph and create two kinds of edges. As in the single-image case of Section 2.1, the *spatial edges* $\mathcal{E}_s^n = \{\mathbf{e}_{ij}^n = (\mathbf{x}_i^n, \mathbf{x}_j^n)\}$ correspond to edges connecting points within I^n and consecutive pairs of such edges are assigned an image-based probability of being part of the final curvilinear structure. To these, we add *temporal edges*

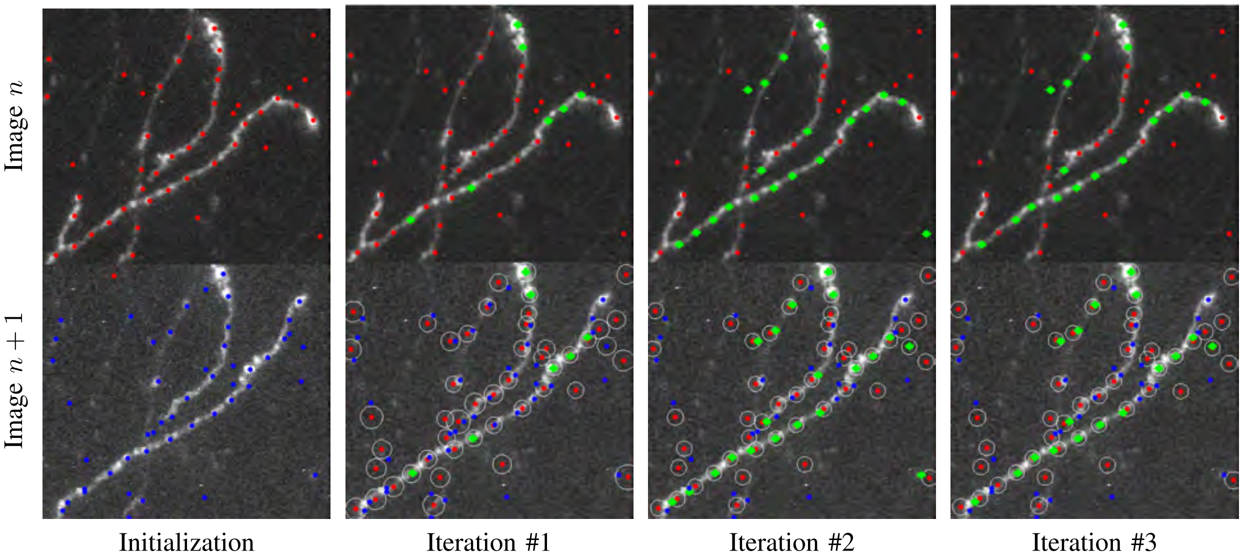

Fig. 2. Establishing correspondences. **Initialization.** A set of corresponding points with possible inconsistencies is found in each image using high-tubularity locations and NCC. **Iteration #1** A set of corresponding points (shown in green) with the highest tubularity likelihoods has been selected, which are then used to instantiate a GPR that maps the remaining red points in image I^n (top) to the red locations in image I^{n+1} (bottom). The white circles denote the covariances associated to these locations. The blue points in image I^{n+1} that are close enough to them and correlate well with the original red points in image I^n are taken to form new correspondences. **Iterations #2 and #3.** They are added to the set of correspondences, shown in green. The process is then repeated.

$\mathcal{E}_t^n = \{\mathbf{e}_{ij}^{n,n+1} = (\mathbf{x}_i^n, \mathbf{x}_j^{n+1}) \mid (\mathbf{x}_i^n \leftrightarrow \mathbf{x}_j^{n+1}) \in \mathcal{S}_n\}$ that connect nodes in I^n and I^{n+1} that belong to the set $\mathcal{S}_n$ of geometrically consistent correspondences.

The spatial and temporal edges play very different roles and we instantiate fewer of the latter than the former. The spatial ones represent actual paths that are the building blocks of our final delineations. Missing one may result in a part of the structures not being modeled. We therefore chose to make the graph overcomplete to minimize the chances of this. By contrast, the temporal edges will only be used to ensure that similar spatial edges in different images can support each other. Missing one might result in evidence not being exploited fully but is less likely to have adverse consequences, It is therefore acceptable to instantiate fewer to keep computational complexity under control.

The whole procedure is controlled by three parameters: The size of the neighborhoods used to compute the NCC and the two thresholds on the minimum acceptable value of the tubularity value and maximum acceptable Mahalanobis distance for corresponding vertices. In practice, we take a constant neighborhood size equal to half the minimum distance between graph nodes, that is the size of the suppression neighborhood introduced in Section 2.1. which is always between 20 and 30 pixels. We take the minimum acceptable value of the tubularity value in image to be 0.15 times the maximum value in that image and the Mahalanobis threshold to be 4. We justify these choices in Section B.2.1 of the appendix.

4 Enforcing Temporal Consistency

Given the spatio-temporal graph $\mathcal{G}$ introduced in the previous section, our goal now is to find a subgraph forming a set of trees that evolve consistently over time. For every image in the sequence, the locations of the tree roots are provided by an operator and are added to the set of graph nodes. An additional imaginary root $\mathbf{x}_r$ is created and connected to all these root nodes for all time instants. This way, reconstructing the trees in all images can be achieved by finding the most likely arborescence rooted at $\mathbf{x}_r$.

4.1 Solving without Time Consistency

Reconstructing the trees of interest means deciding which edges of the graph $\mathcal{G}$ should be part of the final solution. To this end, we use Bayes theorem and derive an energy model from probability estimates, as in [5]. Let $Y_{ij}^n \in \{0,1\}$ be a binary random variable denoting the presence or absence of the edge $\mathbf{e}_{ij}^n$ in the final solution and Y be the set of all Y_{ij}^n variables. Our goal is to infer the most likely Y.

We could ignore the existence of the *temporal edges* and directly use the algorithm of [5]. In this case, computing the optimal tree would simply amount to solving

$$\begin{aligned} y^* &= \arg\max_{y \in \mathcal{Y}} P\left(\mathcal{I}, \mathcal{X}, \mathcal{E}_s | Y = y\right) \,, \\ &= \arg\min_{y \in \mathcal{Y}} \sum_{\mathbf{e}_{ij}^n, \mathbf{e}_{jk}^n \in \mathcal{E}_s} w_{ijk} y_{ij}^n y_{jk}^n, \end{aligned} \tag{2}$$

where $w_{ijk} = -\log \frac{p_{ijk}}{1-p_{ijk}}$, p_{ijk} is the probability of edge pair $(\mathbf{e}_{ij}^n, \mathbf{e}_{jk}^n)$ being part of a tubular structure introduced in Section 2.1, and $\mathcal{Y}$ is the set of all Y defining feasible trees rooted at $\mathbf{x}_r$. As time consistency is not enforced, this amounts to independent reconstructions in all images, as described in Section 2.1. However, this formulation can also be exploited for our purposes first by introducing auxiliary floating point *flow variables* and imposing *flow constraints* that enforce both spatial connectivity [12], [5] and temporal consistency, as described below.

4.2 Flow Variables and Spatial Connectivity

Given the spatio-temporal graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ introduced above, solving the minimization problem of Eq. 2 amounts to finding a subset $\mathcal{E}_s' \subseteq \mathcal{E}_s$ of edges that form a tree rooted at node $\mathbf{x}_r$ that minimizes the objective function. This implies that there must be exactly one directed path from $\mathbf{x}_r$ to every vertex in that solution tree.

To ensure that this is the case, we introduce a set $F = \{f_{ij}^m\}$ of variables called *flow variables*. Each of those corresponds to one vertex-edge pair $(\mathbf{x}_m, \mathbf{e}_{ij}) \in \mathcal{X} \times \mathcal{E}_s$ in the graph. If vertex $\mathbf{x}_m$ is not part of the solution tree, all the *flow variables* f_{ij}^m are set to 0. If it is part of the solution the value of f_{ij}^m indicates whether the unique path from the root vertex $\mathbf{x}_r$ to the target vertex $\mathbf{x}_m$ traverses the edge $\mathbf{e}_{ij}$ or not. If yes it is set to 1 and otherwise to 0. This way, if the solution is a tree, there is a unit flow from the root to every target vertex that is part of it.

As shown in [12], the tree connectivity constraints can therefore be enforced by minimizing the criterion of Eq. 2 subject to

$$
\begin{aligned}
&\sum_{\mathbf{x}_j \in \mathcal{X} \setminus \{\mathbf{x}_r\}} f_{rj}^m \leq 1, && \forall \mathbf{x}_m \in \mathcal{X} \setminus \{\mathbf{x}_r\}, \\
&\sum_{\mathbf{x}_j \in \mathcal{X} \setminus \{\mathbf{x}_k\}} f_{jk}^m \leq 1, && \forall \mathbf{x}_m \in \mathcal{X} \setminus \{\mathbf{x}_r\}, \\
&\sum_{\mathbf{x}_j \in \mathcal{X} \setminus \{\mathbf{x}_i, \mathbf{x}_r\}} f_{ij}^m - \sum_{\mathbf{x}_j \in \mathcal{X} \setminus \{\mathbf{x}_i, \mathbf{x}_m\}} f_{ji}^m = 0, && \begin{matrix} \forall \mathbf{x}_m \in \mathcal{X} \setminus \{\mathbf{x}_r\}, \\ \forall \mathbf{x}_i \in \mathcal{X} \setminus \{\mathbf{x}_r, \mathbf{x}_m\}, \end{matrix} \\
&f_{ij}^m \leq y_{ij}^n, && \forall e_{ij} \in \mathcal{E},\ \mathbf{x}_m \in \mathcal{X} \setminus \{\mathbf{x}_r, \mathbf{x}_i, \mathbf{x}_j\}, \\
&f_{im}^m = y_{im}^n, && \forall e_{im} \in \mathcal{E}, \\
&f_{ij}^m \geq 0, && \forall e_{ij} \in \mathcal{E},\ \mathbf{x}_m \in \mathcal{X} \setminus \{\mathbf{x}_r, \mathbf{x}_i\}, \\
&y_{ij}^n \in \{0, 1\}, && \forall e_{ij} \in \mathcal{E}
\end{aligned}
\tag{3}
$$

where $\mathbf{x}_r$ is the imaginary root vertex. During the optimization the edge variables are treated as integers and the flow variables as real numbers. However, in the end the flows must be equal to zero or one. As shown in [12], explicitly constraining the edge variables y_{ij}^n to be either zero or one achieves this goal and turns our initial integer program into a QMIP. In practice, the path from the imaginary root to any vertex $\mathbf{x}_r^n$ of the spatio-temporal graph cannot pass through spatial edges corresponding to different times and it is unnecessary to introduce the corresponding flow variables, which simplifies the computation.

4.3 Flow Variables and Temporal Consistency

We now turn to enforcing temporal consistency by using the flow variables introduced in Section 4.2.

In [6], temporal consistency was promoted by penalizing situations in which an edge connecting two vertices was included at one time instant but not the next. Here, we propose a more global approach to achieving this goal by imposing temporal consistency on the solution's topology rather than only on local connections.

More specifically, if two corresponding vertices in consecutive images both belong to the solution, they must both be linked to the imaginary root by a path along which the flow is always positive and we want to ensure that these positive flows are assigned to corresponding edges in the two images. In other words, given a pair of corresponding vertices $x_m^n, x_{m'}^{n+1}$ and a pair of corresponding edges $e_{ij}^n, e_{i'j'}^{n+1}$, we should have either $f_{ij}^m = f_{i'j'}^{m'} = 1$ or $f_{ij}^m = f_{i'j'}^{m'} = 0$. In practice, for a given pair of corresponding vertices, it is neither beneficial nor computationally efficient to include the flow consistency constraints for every pair of corresponding edges. Imposing it for edges very distant from the vertex in question might put too much weight on the consistency term as compared to the image term. It would also introduce many quadratic terms in the cost function, which would slow down the computation considerably. Instead, for a given pair of vertices x_m^n and $x_{m'}^{n+1}$ we only take into consideration those edges whose distance from the vertex in question is smaller than a predefined *temporal consistency range parameter* r.

We could make this a hard constraint but, since we want to allow for occasional topology changes, we introduce a *temporal consistency probability parameter* q that represents the probability of this constraint being satisfied. In Section A of the appendix, we will show that finding the most probable delineation given this added constraint amounts to computing

$$
\begin{aligned}
y^* &= \arg\max_{y \in \mathcal{Y}} P\left(\mathcal{I}, \mathcal{X}, \mathcal{E}_s | Y = y\right) P\left(Y = y | \mathcal{E}_t\right), \qquad (4) \\
&= \arg\min_{y \in \mathcal{Y}} \sum_{\mathbf{e}_{ij}^n, \mathbf{e}_{jk}^n \in \mathcal{E}_s} w_{ijk} y_{ij}^n y_{jk}^n \\
&- \sum_{(\mathbf{x}_m^n, \mathbf{x}_{m'}^{n+1}) \in \mathcal{E}_t} \sum_{(\mathbf{e}_{ij}^n, \mathbf{e}_{i'j'}^{n+1}) \in \bar{\mathcal{E}}_t} w_p \left(2 f_{ij}^m f_{i'j'}^{m'} - f_{ij}^m - f_{i'j'}^{m'}\right).
\end{aligned}
$$

where $w_p = \log \frac{q}{1-q}$ and $\bar{\mathcal{E}}_t$ is the set of spatial edge pairs in two consecutive images whose endpoints are both connected by temporal edges in $\mathcal{E}_t$. Note that the connectivity constraints are the same as before and can therefore be imposed by performing the minimization under the linear constraints of Eqs. 3, which are also expressed in terms of the the y_{ij}^n and f_{ij}^m variables. The problem therefore remains a QMIP. Furthermore, this approach unlike our earlier one [6], remains effective even when the sampling step yields sample points that cannot be assigned corresponding ones in subsequent time steps.

4.4 Fine alignment

Given the final delineations, the iterative GPR method of Section 3 can be applied once again for fine alignment of the delineations. In every pair of consecutive images I^n and I^{n+1}, we take the retained matching nodes to be the reliable set of initial matches S_n^0. We also sample additional uniformly distributed points from the paths of the solution trees and treat them as candidate points for matching. We then iterate the GPR estimation and matching until convergence. Points without correspondences are then detected as potential differences between time instants.

		Single [5]	Short-range [6]	Long-range
BR1	Image #1	0.6303	0.3654	**0.6345**
	Image #2	0.4653	0.3433	**0.5178**
	Image #3	0.5929	0.3374	**0.6325**
	Average BR1	0.5628	0.3487	**0.5949**
BR2	Image #1	0.4964	**0.5444**	0.4242
	Image #2	0.2437	0.5327	**0.5884**
	Image #3	0.5200	0.5771	**0.6766**
	Average BR2	0.4201	0.5514	**0.5631**
BR3	Image #1	0.7722	**0.7903**	**0.7903**
	Image #2	**0.6890**	**0.6890**	0.6746
	Image #3	0.4761	0.4395	**0.9185**
	Average BR3	0.6458	0.6396	**0.7945**
BR4	Image #1	0.3846	**0.6000**	0.5940
	Image #2	0.5088	0.5272	**0.5677**
	Image #3	0.5910	0.6118	**0.6340**
	Average BR4	0.4948	0.5797	**0.5986**
BR5	Image #1	0.3713	**0.3956**	0.3952
	Image #2	0.2439	0.2915	**0.3687**
	Image #3	0.2060	0.2761	**0.3134**
	Average BR5	0.2737	0.3211	**0.3591**
RD1	Image #1	0.2650	0.4630	**0.4680**
	Image #2	0.3880	0.4000	**0.4150**
	Image #3	0.3240	0.3330	**0.4670**
	Average RD1	0.3257	0.3987	**0.4500**
RD2	Image #1	0.4116	0.5000	**0.5380**
	Image #2	0.7160	0.6880	**0.7210**
	Image #3	0.2920	0.6110	**0.7140**
	Average RD2	0.4732	0.6000	**0.6577**

TABLE 1

DIADEM scores [17] for the brain images datasets (denoted BRi) and road images datasets (denoted RDi). The scores in the first column were obtained without time-consistency, that is, by using the method of [5]. The scores in the second and third columns were obtained by imposing short-range time consistency as in our earlier method [6] and long-range time consistency as described in this paper.

5 Results

We first present the two very different image datasets we used to test our approach. We then show that enforcing time-consistency allows us to improve overall performance. We discuss a third dataset in the appendix.

5.1 Image Datasets

We evaluated our method on 2-photon microscope image stacks of axons in the brain of a mouse and on aerial images of the same area taken in different years and seasons.

Brain Structures. We collaborate with neuroscientists who aim at mapping structural circuit changes in the mouse brain during the learning processes. To this end they acquire large-scale 2-photon laser scanning microscopy images of a sparse set of fluorescently labeled neurons in the neocortex. Images are taken through a permanently implanted cranial window, which lets them track specific structures over months during which the mouse learns new tasks.

We used 3 large image stacks of the same brain area at three different times. To train the path classifier, we selected a region from one of the stacks, asked an expert to manually delineate the structures, and sampled 20000 positive and 20000 negative paths. Five sequences of smaller volumes that did not overlap the training ones were then selected from the three image stacks for testing. A single test sequence consists of three volumes representing roughly the same brain area, each one taken from a different stack. We will refer to them as BR1, BR2, BR3, BR4, and BR5.

Urban Roads. We also tested our approach on a road network delineation task. We used two different sequences of three aerial images of the same area taken in different years at different seasons, one in Switzerland and the other in the USA. As can be seen in the top rows of both Fig. 4 and 5, the appearance varies substantially from one to the next due to changing illuminations, different level of occlusion from trees captured in different seasons, and new constructions in the case of Fig. 5. For each area, we picked several regions spanning those varying appearances to train a single classifier as we did for the brain images. We then cropped the test images so that they do not overlap the training ones. We will refer to them as RD1 and RD2 and we provide ground truth delineations in the appendix.

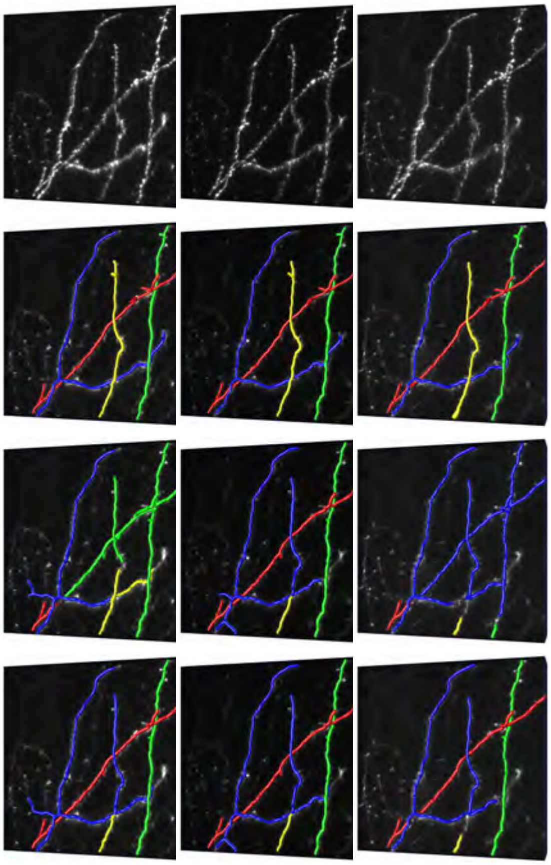

Fig. 3. Axon Delineation. **First row.** Input images. **Second row.** Ground truth. **Third row.** Delineations without temporal consistency. **Fourth row.** Delineations with temporal consistency enforced. The red, green, and blue branches are much better modeled. Half of the yellow branch is mistakenly attached to the blue one because, where the two branches almost cross, the image is extremely blurry. However, this error would be easier to correct during a post-processsing state because the topology is at least consistent in all frames.

5.2 Overall Performance

In Table 1, we compare the results of reconstructing the linear structures independently in each test image of our

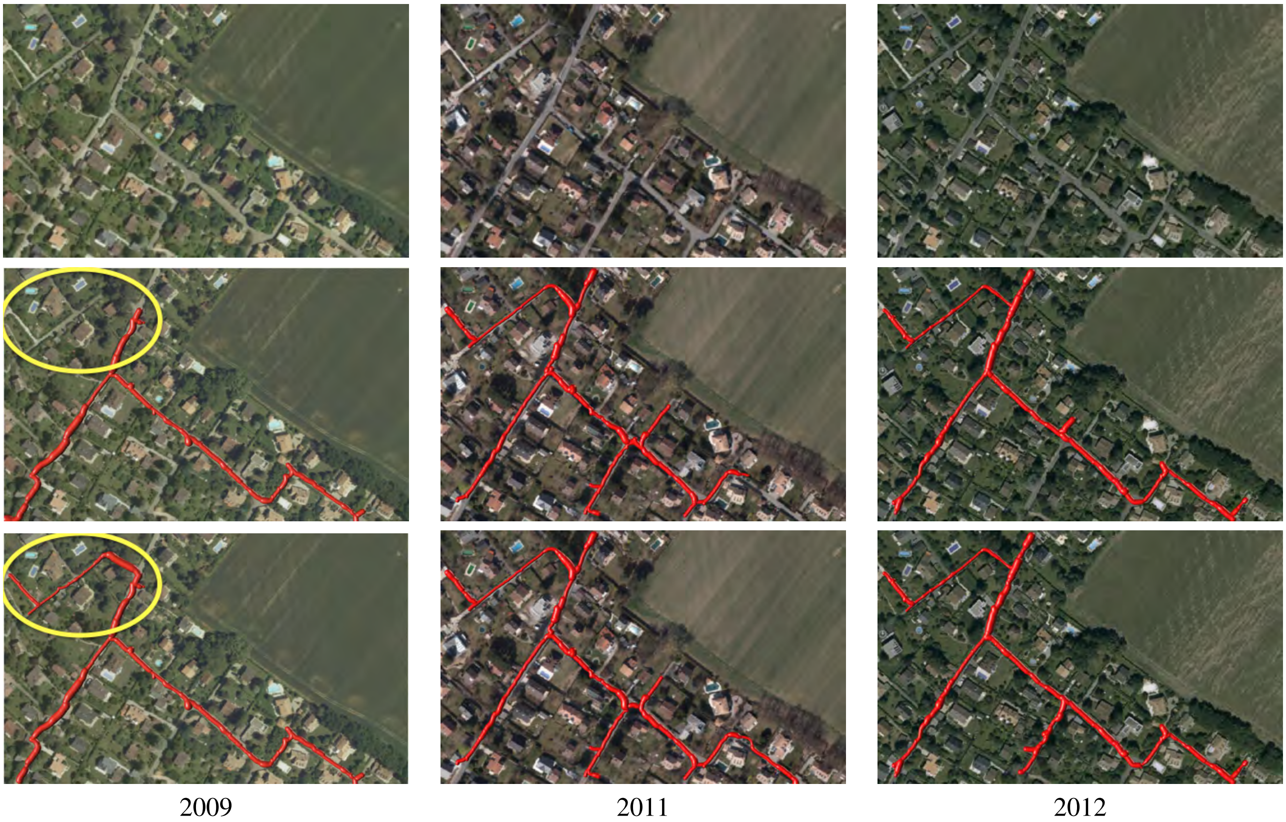

Fig. 4. Road delineation. **First row.** Images acquired in 2009, 2011, and 2012. **Second row.** Delineations without temporal consistency. **Third row.** Delineations with temporal consistency enforced. Note the added road fragments within the overlaid yellow ellipse.

datasets as in [5], of imposing short-range temporal constraints as in [6], and of imposing long-range constraints temporal constraints as discussed in this paper. We quantify the reconstruction accuracy in terms of the DIADEM metric [17]. It measures the similarity with the ground truth in a way that is appropriate for trees and ranges from 0.0 to 1.0, with 1.0 being best.

We set the temporal consistency probability and range parameters q and r introduced in Section 4.3 to 0.75 and 4 in all experiments. When $q \approx 0.5$, the consistency term is dwarfed by the image term and temporal consistency is not enforced. $q \approx 1.0$ prevents any differences between the delineations at different times and can cause some correctly delineated branches to be removed entirely, which results in a very time-consistent but incomplete solution. Similarly, taking $r = 4$ proved sufficient to improve robustness without unduly increasing computational complexity. Higher values tend to overemphasize consistency with the same results as when using too large a value for q,

We present sample results in Figs. 1, 3, 4, and 5. As can be seen, imposing temporal consistency results in higher quality results on average. Furthermore, this allows change detection as shown in lower left corner of Fig. 1(f) where a branch seems to have retracted and in the upper right corner where another seems to have changed orientation.

As shown in Table 1, enforcing local temporal consistency improves the overall quality of the results as we did in our previous approach [6] and imposing long-range consistency as we do here improves them even more. Our algorithm propagates the scores assigned to paths in one image to similar paths in another one. As a result, the road segments within the yellow ellipses overlaid in the bottom rows of Fig. 4 and 5 are found in all images when temporal consistency is imposed even though some were ignored when processing the images individually. This is usually beneficial but can occasionally backfire, as can be seen within the yellow box overlaid on the bottom rows of Fig. 5. The road fragment is occluded by overhanging trees in both the 2002 and 2015 images and only clearly visible in the 2016 one and imposing temporal consistency results in its disappearance from the final delineation. This explains why the DIADEM scores of Table 1 are mostly higher when enforcing temporal consistency but with a few exceptions. In other words, our algorithm favors paths that are assigned good scores in the majority of images and penalizes the others.

6 Conclusion

We have proposed a novel framework for modeling complex curvilinear structures over time. The heart of our approach lies in finding local and stable structures that are consistent over time, and which can be used to disambiguate cases where individual time-instance reconstructions would fail. These additional time constraints are combined with spatial ones and enforced by a Quadratic Mixed Integer Program. This makes it possible to reconstruct trees that remain consistent over time.

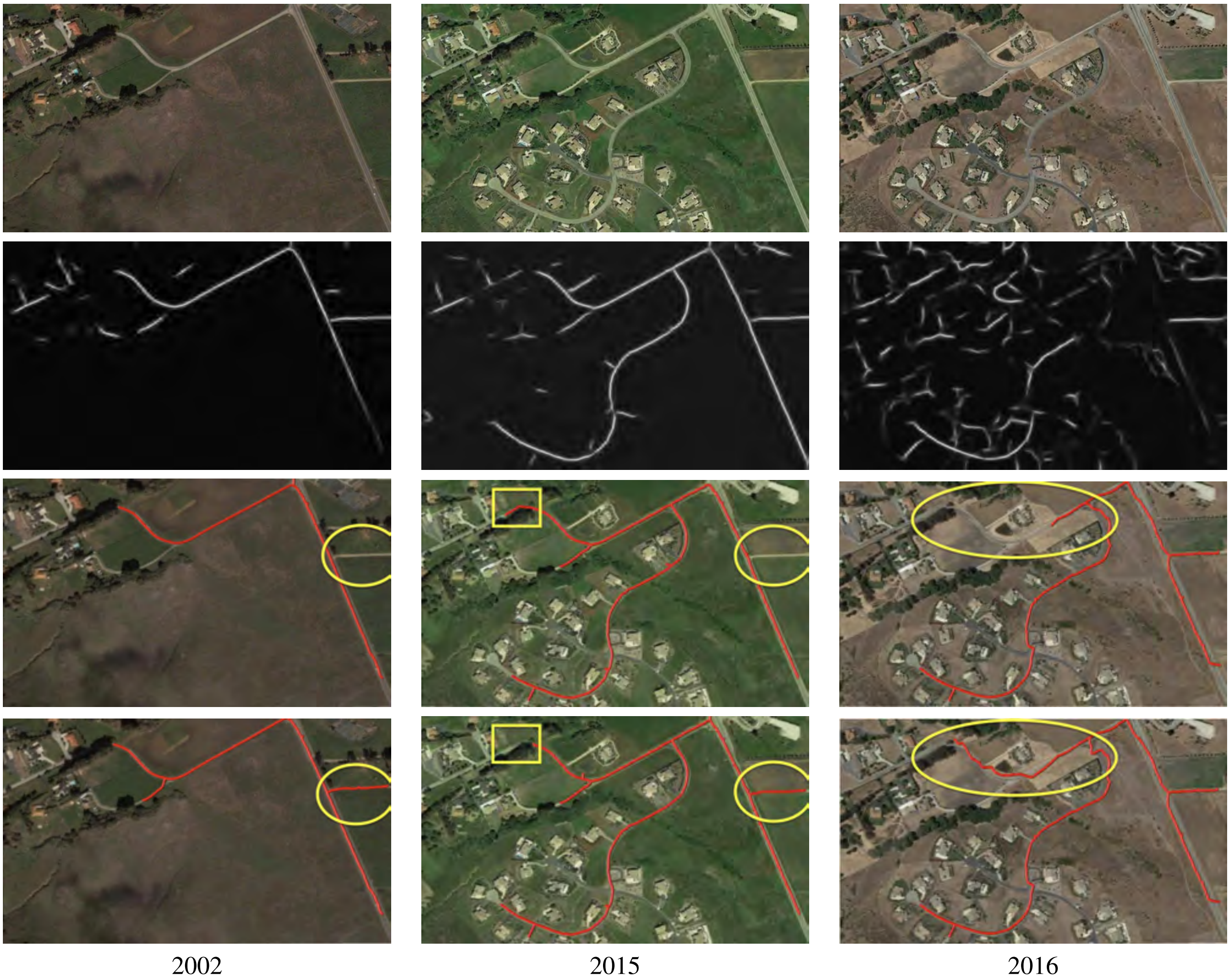

Fig. 5. Road delineation. **First row.** Images acquired in 2002, 2015, and 2016. New houses and roads have been built between 2002 and 2015. **Second row.** Tubularity images. Note that the 2016 one is noisier, making delineation more difficult. **Third row.** Delineations without temporal consistency. **Fourth row.** Delineations with temporal consistency enforced. Note the added road fragments within the overlaid yellow ellipses and the removed one within the overlaid yellow box.

We have shown that our approach successfully takes advantage of temporal information to produce more reliable and accurate reconstructions of tree structures. In addition, we showed that our approach has the added benefit of automatically detecting regions where significant topology change occur in tree structures.

References

[1] R. Nevatia and R. Babu, "Linear Feature Extraction and Description," *CGIP*, vol. 13, no. 3, pp. 257–269, 1980.

[2] M. Fischler and H. Wolf, "Linear Delineation," in *CVPR*, June 1983, pp. 351–356.

[3] A. Holtmaat, J. Randall, and M. Canea, "Optical Imaging of Structural and Functional Synaptic Plasticity in Vivo," *European Journal of Pharmacology*, 2013.

[4] Z. Li, S. Liu, R. Weinreb, J. Lindsey, M. Yu, L. Liu, C. Ye, Q. Cui, W. Yung, C. Pang, D. Lam, and C. Leung, "Tracking Dendritic Shrinkage of Retinal Ganglion Cells After Acute Elevation of Intraocular Pressure," *Invest Ophthalmology Vision Science*, vol. 52, no. 10, pp. 7205–12, 2011.

[5] E. Turetken, F. Benmansour, B. Andres, P. Glowacki, H. Pfister, and P. Fua, "Reconstructing Curvilinear Networks Using Path Classifiers and Integer Programming," *PAMI*, vol. 38, no. 12, pp. 2515–2530, 2016.

[6] P. Glowacki, M. Pinheiro, E. Turetken, R. Sznitman, D. Lebrecht, A. Holtmaat, J. Kybic, and P. Fua, "Reconstructing Evolving Tree Structures in Time Lapse Sequences," in *CVPR*, 2014.

[7] V. Lepetit and P. Fua, *Monocular Model-Based 3D Tracking of Rigid Objects: A Survey*. Now Publishers, September 2005.

[8] D. Ramanan, A. Forsyth, and A. Zisserman, "Strike a Pose: Tracking People by Finding Stylized Poses," in *CVPR*, 2005.

[9] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3D Pose Estimation and Tracking by Detection," in *CVPR*, 2010.

[10] Q. Li, Z. Deng, Y. Zhang, X. Zhou, U. Nagerl, and S. Wong, "A Global Spatial Similarity Optimization Scheme to Track Large Numbers of Dendritic Spines in Time-Lapse Confocal Microscopy," *TMI*, vol. 30, no. 3, pp. 632–641, 2011.

[11] M. Law and A. Chung, "Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux," in *ECCV*, 2008.

[12] C. Duhamel, L. Gouveia, P. Moura, and M. Souza, "Models and Heuristics for a Minimum Arborescence Problem," *Networks*, vol. 51, no. 1, pp. 34–47, 2008.

[13] A. Sironi, E. Turetken, V. Lepetit, and P. Fua, "Multiscale Centerline Detection," *PAMI*, 2016.

[14] C. E. Rasmussen and C. K. Williams, *Gaussian Process for Machine Learning*. MIT Press, 2006.

[15] E. Serradell, P. Glowacki, J. Kybic, F. Moreno, and P. Fua, "Robust Non-Rigid Registration of 2D and 3D Graphs," in *CVPR*, June 2012.

[16] X. Yu, J. Tian, and J. Liu, "Transformation Model Estimation for Point Matching via Gaussian Processes," in *World Congress of Engineering*, 2007.

[17] G. Ascoli, K. Svoboda, and Y. Liu, "Digital Reconstruction of Axonal and Dendritic Morphology DIADEM Challenge," 2010. [Online]. Available: http://diademchallenge.org/