

# Principal Curves as Skeletons of Tubular Objects

## Locally Characterizing the Structures of Axons

Erhan Bas · Deniz Erdogmus

Received: date / Accepted: date

**Abstract** Developments in image acquisition technology make high volumes of neuron images available to neuroscientists for analysis. However, manual processing of these images is not practical and is infeasible for larger and larger scale studies. Reliable interpretation and analysis of high volume data requires accurate quantitative measures. This requires analysis algorithms to use as realistic as possible realistic mathematical models that inherit the underlying geometry of biological structures in order to extract topological information. In this paper, we first introduce principal curves as a model for the underlying skeleton of axons and branches, then describe a recursive principal curve tracing (RPCT) method to extract this topology information from 3D microscopy imagery. RPCT first finds samples on the one dimensional principal set of the intensity function in space. Then, given an initial direction and location, the algorithm iteratively traces the principal curve in space using our principal curve tracing (PCT) method. Recursive implementation of PCT provides a compact solution for extracting complex tubular structures that exhibit bifurcations.

**Keywords** Axon tracing · Principal curve

### 1 Introduction

Functional and structural neuroimaging techniques are common tools to analyze the connectivity and functionality of neuronal structures which can be used to understand how neuronal networks behave and can be used for treatment of

This work was partially supported by NSF grants ECCS0929576, ECCS0934506, IIS0934509, and IIS0914808.

E. Bas and D. Erdogmus  
Cognitive Systems Laboratory  
Northeastern University, 360 Huntington Ave.  
Tel.: +1-617-3733021  
E-mail: [bas,erdogmus]@ece.neu.edu

neuro-degenerative diseases such as Alzheimer or Parkinson. Different techniques have been implemented to highlight the structural and/or functional properties of certain biological networks. Structural and functional MR(s/f MRI) and diffusion tensor/spectral imaging (DTI/DSI) for human neuroimaging have been used to identify important neuronal fiber tracks in the brain that helps to identify brain tumors and increase the accuracy of treatment. Similarly, multiphoton/confocal images from mice help researchers to develop reliable wiring diagrams, which enable a better understanding of how the central nervous system operates, and how the brain works. With recent developments in imaging technology, high volumes of neuron images are available to neuroscientists to analyze, but manual processing of this data is becoming impractical due to high data volume. Moreover, reliable interpretation of such massive data requires accurate quantitative measures that can be obtained without human intervention. To this end, realistic mathematical models that explain the underlying geometry of neuronal structures can be used to extract topological information, which in turn can be used in obtaining functional models and conclusions using simulations.

Structures with branching tree forms are common in three-dimensional (3D) biomedical images; such structures include the bronchial tree, vascular and neuronal networks in the brain and elsewhere in the body of an organism. In order to obtain the topology or wiring information from 3D imagery, different methods have been proposed in various contexts. Most methods seek some form of optimal path/tree solution by minimizing an objective function locally or globally. A commonly sought optimality condition is the shortest Euclidean length between end points while constrained to the high (or low) intensity regions of the image where the object of interest resides. Optimal wiring (trace) connecting two voxels can be identified by searching all possible trace candidates (global search) or by constraining this optimal trace

search procedure to local neighborhoods (local search), and assigning the path that has the minimum length as the solution.

Global methods such as wave propagation or vector scooping Deschamps and Cohen (2001); Schmitt et al (2004); Vasilkoski and Stepanyants (2009); Rodriguez et al (2009) return optimal traces (also called the centerline) in exchange for speed. Since search space increases with the dimension and size of the data and a greedy search can be employed to find a reasonably good suboptimal trace; these methods are generally applied to already segmented or well defined sparsely distributed (neuronal) structures. Since searching all possible scenarios is not computationally feasible for increased data size (i.e. number of voxels), one common approach is to use slice based approximations to process 3D data assuming curvilinear structures mainly propagate in one direction, thus remaining in a planar field most of the time. This way, the trace optimization problem in 3D can be approximated by a 1D tracking problem of a 2D-object. Tracking can be achieved through simple predictive linear filters Cai et al (2006, 2008); Wang et al (2007); Jurrus et al (2009); Aykac et al (2003). Another commonly used global method is the 3D thinning operation. This type of method can be used to obtain the skeleton of the data, but this operation is not robust to noise and needs detailed supervision to prevent unnecessary splits Palágyi and Kuba (1998); Zhou et al (2008).

Local methods employ local decisions to trace the centerline without searching the whole space and testing all possible scenarios. In general, local methods start from a seed point and iterate through space by optimizing an objective function defined in the vicinity of the current iteration. Although, these *exploratory* algorithms estimate the optimum paths locally, and they provide fast and feasible solutions for the problem Bas and Erdogmus (2010a); Wink et al (2004); Frangi et al (1999); Al-Kofahi et al (2004); Bas and Erdogmus (2010b). In this setup, a common feature is the local curvature. Curvature of the curvilinear structure can be estimated by the eigenanalysis of the Hessian matrix of the image intensity in various biomedical images. Moreover, multiscale representation of the data enables the highlighting of curvilinear structures by searching in the scale space as well as in spatial feature space. The multiscale representation can also be used for curvilinear radius extraction or image denoising Chen et al (2002). In addition, a more abstract connectivity analysis can be achieved by using atlases. Although most methods for connectivity analysis assumes tree-like structures which imposes local connectivity through graph representation of the data, it is also possible to define regional connectivity Huang et al (2009) by using functional neuroimages instead of voxel-based connectivity measures.

One of the biggest challenges in tracing of curvilinear

structures is the detection and the analysis of branching locations. Greedy methods such as wave propagation or scooping provide a direct solution to splitting or merging branches by checking all of the image/feature space. Unfortunately, as mentioned earlier, as the data size increases these methods are not suitable for tracing without prior segmentation information where boundary knowledge of the curvilinear structure is given. Previously mentioned 2D tracking versions of these methods employ occlusion detection algorithms to decide if a bifurcation exists. A more robust tracking method is discussed in Jurrus et al (2009), where authors used an adaptive predictive filtering scheme to improve the trajectory. However, note that these decisions are based on an arbitrary 2D projection, i.e. sagittal or coronal slice, which does not have to be aligned with the curvilinear structure. Such 2D approaches introduce an artificial occlusion problem that is not present in 3D, then try to resolve this self-created issue. Methods that operate in 3D will capture physical conditions and constraints better, and avoid such artificially created occlusions automatically. For that reason, 3D exploratory algorithms are commonly used for local decisions of bifurcation points. Moreover, they can be combined with other decision mechanisms easily. Probabilistic methods such as stochastic tractography or hypothesis testing techniques can be used to detect and handle bifurcations Al-Kofahi et al (2004); Cetingul et al (2009), where most of these methods treat bifurcation points as special cases, and detection and processing of these cases are handled separately.

In this paper we propose a geometrically motivated strategy to detect and handle branches. The method described here relies on the concept of principal curves that underly functions and uses a principal curve tracing method to trace the curvilinear structures, while creating the tree in a recursive fashion. Section 2 briefly explains the principal curve concept and an associated tracing method. In section 3, recursive branching and tree reconstruction strategy is discussed. Algorithm results are presented in section 4, followed by a brief conclusion in section 5.

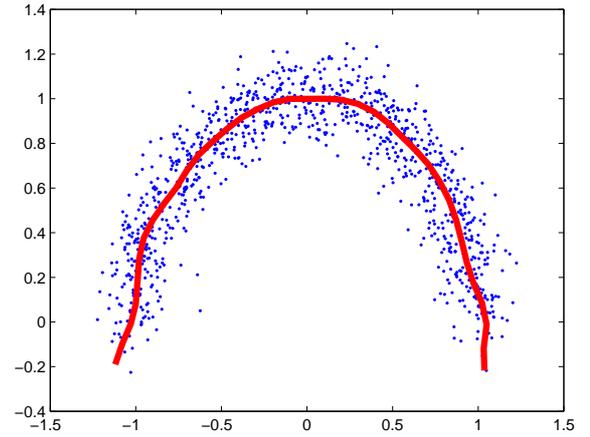
## 2 Principal Curves

In statistical machine learning, principal curves are the curves that pass through the data or data cloud. In differential geometry, the principal directions at a point on a manifold are directions along which local curvature is stationary (i.e. minimum, maximum, or saddle curvature directions). Principal curves of functions (a particular choice of the manifold/surface) provide a compact summary of the one dimensional underlying structure and can be interpreted as the nonlinear skeleton of a function in some high dimensional space; such as intensity or probability distribution over space. Hastie and Stultz Hastie and Stuetzle (2003) defined self-consistent principal curves as consisting of points which are

the conditional mean of data that orthogonally project to this point given the curve. Following this, most semi-global principal curve algorithms seek to reduce the least mean square error of fitting a model to the data through regression Hastie and Stuetzle (2003); Kegl et al (2000); Chang and Ghosh (2002); Meinicke et al (2005). Local principal curve extraction methods are also present in the literature to define principal curves. In general these methods use first and second order statistics and derivatives to approximate the principal curves Einbeck et al (2005); Erdogmus and Ozertem (2007).

Here, we briefly discuss the principal surface definition described in Erdogmus and Ozertem (2007). Let  $\mathbf{x} \in \mathbb{R}^n$  be a random vector with samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ , having a given pdf estimate of  $p(\mathbf{x})$ . Let  $\mathbf{g}(\mathbf{x})$  be the transpose of its local gradient, and let  $\mathbf{H}(\mathbf{x})$  be the local Hessian of this pdf. Using the gradient and Hessian, define the local covariance-inverse as:  $\mathbf{C}^{-1}(\mathbf{x}) = -p^{-1}(\mathbf{x})\mathbf{H}(\mathbf{x}) + p^{-2}\mathbf{g}(\mathbf{x})\mathbf{g}^T(\mathbf{x})$ . The local covariance is defined in this manner using the second order term in the Taylor series expansion of  $\log p(\mathbf{x})$  in order for principal curve projections to be consistent with linear principal component analysis (PCA) projections in the case of a Gaussian density. Selection of the monotonic function  $\log$  above is not unique and it be replaced by any monotonically increasing, twice continuously differentiable function without influencing the principal curves, but changing the projection trajectories for moving points on to the principal curve from its vicinity. Let  $\{(\lambda_1(\mathbf{x}), \mathbf{q}_1(\mathbf{x})), \dots, (\lambda_n(\mathbf{x}), \mathbf{q}_n(\mathbf{x}))\}$  be the eigenvalue-eigenvector pairs of  $\mathbf{C}^{-1}(\mathbf{x})$ , sorted in ascending order:  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ . A point  $\mathbf{x}$  is on the  $0$ -dimensional principal set (local maxima), iff  $\mathbf{g}(\mathbf{x})$  is orthogonal to  $n$  eigenvectors ( $\mathbf{g}(\mathbf{x}) = 0$ ) and all Hessian eigenvalues are negative. Similarly,  $\mathbf{x}$  is on the  $d$ -dimensional principal set, iff  $\mathbf{g}(\mathbf{x})$  is orthogonal to at least  $(n-d)$  eigenvectors. For instance, without loss of generality, let  $S_{\perp}(\mathbf{x}) = \text{span}\{\mathbf{q}_2(\mathbf{x}), \mathbf{q}_3(\mathbf{x}), \dots, \mathbf{q}_n(\mathbf{x})\}$  be the normal space spanned by the  $n-1$  orthogonal eigenvectors and  $S_{\parallel}(\mathbf{x}) = \text{span}\{\mathbf{q}_1(\mathbf{x})\}$  be the tangent vector at  $\mathbf{x}$ . If a point is on the principal curve, then  $\mathbf{g}(\mathbf{x})$  is orthogonal to  $S_{\perp}(\mathbf{x})$ . Updates constrained to the  $S_{\perp}(\mathbf{x})$  plane will converge to or diverge from the principal curves depending on the update direction, whereas propagating along the tangent space (vector) ( $S_{\parallel}(\mathbf{x})$ ) will trace the locally defined principal curve at  $\mathbf{x}$ . So an iterative tracing algorithm using correction-update scheme is possible by incorporating the iterations on the normal plane (correction step) and the tangential vector (update step) with proper directions. Fig. 1 illustrates the tracing of a perturbed semicircular data given an initial direction (black arrow) and seed location (green circle). Consider a general weighted variable-width kernel density estimate (KDE)<sup>1</sup> obtained from samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  and initial tracing direction  $\gamma_0$ . KDE is

<sup>1</sup> KDE is used as an example since it encompasses parametric mixture models as a special case; the method is general for any pdf model.



**Fig. 1** Tracing the principal curve of a data cloud starting from a seed point - an illustration of the concept.

given as

$$p(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x}_i) G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \quad (1)$$

where  $w(\mathbf{x}_i)$  is the weight and  $\Sigma_i$  is the variable kernel covariance<sup>2</sup> of the Gaussian kernel  $G(\mathbf{x}_i) = C_{\Sigma_i} e^{-\frac{1}{2}\mathbf{x}^T \Sigma_i^{-1} \mathbf{x}}$  for the  $i^{\text{th}}$  data sample  $\mathbf{x}_i$ .

The gradient and the Hessian of the KDE are:

$$\mathbf{g}(\mathbf{x}) = - \sum_{i=1}^N w(\mathbf{x}_i) G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \Sigma_i^{-1} (\mathbf{x} - \mathbf{x}_i) \quad (2)$$

$$\mathbf{H}(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x}_i) G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) (\Sigma_i^{-1} (\mathbf{x} - \mathbf{x}_i) (\mathbf{x} - \mathbf{x}_i)^T \Sigma_i^{-1} - \Sigma_i^{-1}) \quad (3)$$

For  $p(\mathbf{x})$  mean-shift (MS) updates are in the form  $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{ms}(\mathbf{x})$ , where

$$\mathbf{ms}(\mathbf{x}) = \left( \sum_{i=1}^N G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \Sigma_i^{-1} \right)^{-1} \sum_{i=1}^N G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \Sigma_i^{-1} \mathbf{x}_i \quad (4)$$

and  $\mathbf{ms}(\mathbf{x})$  can be decomposed as  $\mathbf{ms}(\mathbf{x}) = \mathbf{ms}_{\parallel}(\mathbf{x}) + \mathbf{ms}_{\perp}(\mathbf{x})$ . Here  $\mathbf{ms}_{\perp}(\mathbf{x})$  is the normal component given as  $\mathbf{V}_{\perp} \mathbf{V}_{\perp}^T \mathbf{ms}(\mathbf{x})$ , where  $\mathbf{V}_{\perp} = [\mathbf{q}_2(\mathbf{x}), \dots, \mathbf{q}_n(\mathbf{x})]$ . Similarly,  $\mathbf{ms}_{\parallel}(\mathbf{x})$  is the tangential component given by  $\mathbf{q}_1(\mathbf{x}) \mathbf{q}_1(\mathbf{x})^T \mathbf{ms}(\mathbf{x})$ . Constrained MS iterations on  $S_{\perp}(\mathbf{x})$  force  $\mathbf{x}$  to converge to the principal curve through fixed-point iterations whose convergence is guaranteed. At each iteration, sign of the  $\mathbf{ms}_{\parallel}(\mathbf{x})$  must be corrected with the current tracing direction defined by the previous iteration ( $\gamma_{t-1}$ ) and normalized to the step length:  $\gamma(\mathbf{x})_t = \text{sign}(\gamma(\mathbf{x})_{t-1}^T \mathbf{ms}_{\parallel}(\mathbf{x})) \mathbf{ms}_{\parallel}(\mathbf{x})$ . Summary of the algorithm is presented in Table 1.

<sup>2</sup> Assuming Gaussian kernels here for illustration.

**Table 1** Summary of Principal Curve Tracing Algorithm

1. At iteration  $t=0$  initialize  $\mathbf{x}$ , the step size  $\mu$ , and the direction of the curve  $\gamma_0$ .
2. At iteration  $t$  evaluate the mean shift update  $\mathbf{ms}(\mathbf{x}(t))$  as in (9).
3. Evaluate the gradient, the Hessian, and perform the eigendecomposition of  $\mathbf{C}^{-1}(\mathbf{x})=\mathbf{V}\Gamma\mathbf{V}$ , where  $\mathbf{V}_{1\dots n}$  are the eigenvectors with corresponding eigenvalues  $\Gamma = \text{diag}\{\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n\}$ .
4. Let  $\mathbf{V}_1$ , and  $\mathbf{V}_{2\dots n}$  be the eigenvectors that consequently span  $S_{\parallel}(\mathbf{x})$ , and  $S_{\perp}(\mathbf{x})$ , such that  $\mathbf{ms}_{\parallel}(\mathbf{x}) = \mathbf{q}_1(\mathbf{x})\mathbf{q}_1(\mathbf{x})^T \mathbf{ms}(\mathbf{x})$ , and  $\mathbf{ms}_{\perp}(\mathbf{x}) = \mathbf{V}_{2\dots n}\mathbf{V}_{2\dots n}^T \mathbf{ms}(\mathbf{x})$ .
5. Evaluate the new curve direction vector  $\gamma_t = \text{sign}(\gamma_{t-1}^T \mathbf{ms}_{\parallel}(\mathbf{x}))\mathbf{ms}_{\parallel}(\mathbf{x})$
6. If  $p(\mathbf{x}) < thr$  then stop, else  $\mathbf{x}(t+1) \leftarrow \mathbf{x}(t) + \mathbf{ms}_{\perp}(\mathbf{x}) + \mu \frac{\gamma_t}{\|\gamma_t\|}$

### 3 Volumetric Axon Tracing

Following the concept outlined in the previous section, we formulated a 3D curve tracing algorithm that is suitable for extraction of curvilinear structures from 3D biomedical images. In general, in 3D biomedical images, intensity or color values ( $\mathbf{c}$ ) and pixel locations ( $\mathbf{p}$ ) are two common features. Due to problems that occur in high dimensional spaces while estimating density using kernel methods, i.e. curse of dimensionality, instead of estimating the density over  $\mathbf{x} = [\mathbf{p}; \mathbf{c}]$ , we assumed that color values lie on the manifold described by the spatial features ( $\mathbf{x} \leftarrow \mathbf{p}$ ), and incorporate the intensity  $I(\mathbf{p})$  as kernel weights. KDE is given as

$$p(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x}_i) G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \quad (5)$$

Moreover, as the data size increases, regression methods become infeasible due to the computational load. To reduce the computations, bounded-support Gaussian kernels are employed  $G_{\Sigma_i}(\mathbf{x} - \mathbf{x}_i) \leftarrow \alpha B_{2,\varepsilon}(\mathbf{p} - \mathbf{p}_i) G_{\Sigma_i^p}(\mathbf{p} - \mathbf{p}_i)$  constraining computations to neighboring pixels. Here,  $\mathbf{p}_i$  is the position of a voxel in the neighborhood, where K-Nearest Neighbors (KNN) in space denoted by  $N_p^{KNN}$ .  $B_{2,\varepsilon}(\mathbf{p} - \mathbf{p}_i)$  is the support ball with  $L_2$  norm radius  $\varepsilon$ , and  $\alpha$  is the normalization constant of the kernel. KDE at voxel location  $\mathbf{p}$  is given as

$$\begin{aligned} p(\mathbf{x}) &= \sum_{i=1}^N I(\mathbf{p}_i) K_{\Sigma_i^c}(\mathbf{c} - \mathbf{c}_i) \alpha B_{2,\varepsilon}(\mathbf{p} - \mathbf{p}_i) K_{\Sigma_i^p}(\mathbf{p} - \mathbf{p}_i) \\ &= \sum_{x_i \in B_{2,\varepsilon}(\mathbf{p})} w_i \alpha K_{\Sigma_i^p}(\mathbf{p} - \mathbf{p}_i) \end{aligned} \quad (6)$$

where  $w_i = I(\mathbf{p}_i) K_{\Sigma_i^c}(\mathbf{c} - \mathbf{c}_i)$  is the weight of the  $i^{\text{th}}$  kernel. Letting  $\beta_i(\mathbf{x}) = \alpha w_i K_{\Sigma_i^p}(\mathbf{p} - \mathbf{p}_i)$ , the gradient and Hessian are

$$\mathbf{g}(\mathbf{x}) = - \sum_{x_i \in B_{2,\varepsilon}(\mathbf{p})} \beta_i(\mathbf{x}) \Sigma_{ip}^{-1}(\mathbf{p} - \mathbf{p}_i) \quad (7)$$

$$\mathbf{H}(\mathbf{x}) = \sum_{x_i \in B_{2,\varepsilon}(\mathbf{p})} \beta_i(\mathbf{x}) (\Sigma_{ip}^{-1}(\mathbf{p} - \mathbf{p}_i)(\mathbf{p} - \mathbf{p}_i)^T \Sigma_{ip}^{-1} - \Sigma_{ip}^{-1}) \quad (8)$$

**Table 2** Summary of Principal Curve Tracing Algorithm for Image Datasets

1. At iteration  $t=0$  initialize  $\mathbf{x}$ , and the direction of the curve  $\gamma_0$ .
2. At iteration  $t$  evaluate the wKDE mean shift update  $\mathbf{ms}(\mathbf{x}(t))$  as in (9) using the samples that are in the vicinity.
3. Evaluate the gradient, the Hessian, and perform the eigendecomposition of  $\mathbf{C}^{-1}(\mathbf{x})=\mathbf{V}\Gamma\mathbf{V}$  where  $\mathbf{V}_{1\dots n}$  are the eigenvectors with corresponding eigenvalues  $\Gamma = \text{diag}\{\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n\}$ .
4. Let  $\mathbf{V}_1$ , and  $\mathbf{V}_{2\dots n}$  be the eigenvectors that consequently span  $S_{\parallel}(\mathbf{x})$ , and  $S_{\perp}(\mathbf{x})$ , such that  $\mathbf{ms}_{\parallel}(\mathbf{x}) = \mathbf{q}_1(\mathbf{x})\mathbf{q}_1(\mathbf{x})^T \mathbf{ms}(\mathbf{x})$ , and  $\mathbf{ms}_{\perp}(\mathbf{x}) = \mathbf{V}_{2\dots n}\mathbf{V}_{2\dots n}^T \mathbf{ms}(\mathbf{x})$ .
5. Evaluate the new curve direction vector  $\gamma_t = \text{sign}(\gamma_{t-1}^T \mathbf{ms}_{\parallel}(\mathbf{x}))\mathbf{ms}_{\parallel}(\mathbf{x})$
6. If  $\mathbf{x}$  is outside the image boundary or  $p(\mathbf{x}) < thr$  or  $\mathbf{x}$  is previously traced then terminate tracing, else  $\mathbf{x}(t+1) = \arg \min_{x_i \in \mathbf{T}_x} (\gamma_t^T (\mathbf{x}_i - \mathbf{x}))$ .

Here  $\mathbf{T}_x$  is the connected neighborhood of the  $\mathbf{x}$  composed of 26 voxels in 3D.

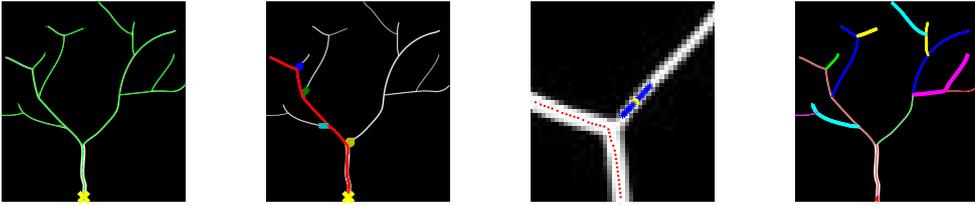
Fixed point  $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{ms}(\mathbf{x})$  mean-shift (MS) iterations can be achieved by equating (7) to 0 such that

$$\mathbf{ms}(\mathbf{x}) = \left( \sum_{x_i \in B_{2,\varepsilon}(\mathbf{p})} \beta_i(\mathbf{x}) \Sigma_{ip}^{-1} \right)^{-1} \sum_{x_i \in B_{2,\varepsilon}(\mathbf{p})} \beta_i(\mathbf{x}) \Sigma_{ip}^{-1} \mathbf{p}_i \quad (9)$$

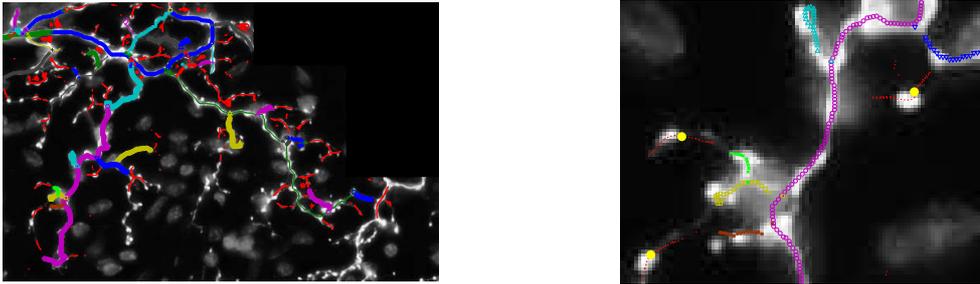
and  $\mathbf{ms}(\mathbf{x})$  can be decomposed as  $\mathbf{ms}(\mathbf{x}) = \mathbf{ms}_{\parallel}(\mathbf{x}) + \mathbf{ms}_{\perp}(\mathbf{x})$ . Here  $\mathbf{ms}_{\perp}(\mathbf{x})$  is the normal component given as  $\mathbf{V}_{\perp} \mathbf{V}_{\perp}^T \mathbf{ms}(\mathbf{x})$ , where  $\mathbf{V}_{\perp} = [\mathbf{q}_2(\mathbf{x}), \mathbf{q}_3(\mathbf{x}), \dots, \mathbf{q}_n(\mathbf{x})]$ . Similarly,  $\mathbf{ms}_{\parallel}(\mathbf{x})$  is the tangential component given by  $\mathbf{q}_1(\mathbf{x})\mathbf{q}_1(\mathbf{x})^T \mathbf{ms}(\mathbf{x})$ . Constrained MS iterations on  $S_{\perp}(\mathbf{x})$  force  $\mathbf{x}$  to converge to the principal curve through fix-point iterations whose convergence is guaranteed. At each iteration, sign of the  $\mathbf{ms}_{\parallel}(\mathbf{x})$  must be corrected with the current tracing direction defined by the previous iteration ( $\gamma_{t-1}$ ) and normalized to the step length:  $\gamma(\mathbf{x})_t = \text{sign}(\gamma(\mathbf{x})_{t-1}^T \mathbf{ms}_{\parallel}(\mathbf{x}))\mathbf{ms}_{\parallel}(\mathbf{x})$ . Although these updates are at subpixel level, instead of calculating the neighborhood of the current iteration, we used the neighborhood of the voxel encapsulating the current iteration to estimate the probability density due to the computational issues. Furthermore, in order to obtain continuous tracing over voxels, we restricted the tracing iterations to immediate spatially neighboring voxels until the termination of the tracing iterations. Termination of the tracing procedure is controlled by the estimated probability value  $p(\mathbf{x})$  calculated as in (6) or the convergence of iterations to a previously traced branch. Summary of the algorithm is presented in Table 2.

### 4 Recursive Implementation for Tree Reconstruction

Principal sets are combinations of 1D manifolds that form a tree structure which spans the whole dataset/object. An intuitive approach to process complex structures having bifurcations is to recursively trace each branch. For that purpose, first, we calculate samples from the 1D principal set of the whole structure in the preprocessing step. Then, for a given seed location and an initial direction, we trace the corresponding branch until it terminates. After each termination



**Fig. 2** a) Synthetic tree data with its projected principal subset. Green dots represent the projections of voxels that are close to the ridge only. b) Red dots display the first traced branch, other colors represent detected clusters after the first iteration. c) Zoomed region around a branching point. d) Final result.



**Fig. 3** Detection and evaluation of principal subset. a) Principal subset. b) Detected clusters.

of branch tracing, the 1D principal set samples are used to check if there are any nearby branches bifurcating from the currently traced branch. For those detected branches, a new PCT algorithm is initialized until all branch candidates are visited. Noticing that points that are close to the ridge of the density function will typically have smaller  $\|\mathbf{ms}_\perp(\mathbf{x})\|$ , samples from the 1D principal set of a KDE constructed from a 3D image as described above can be obtained as described next. For every voxel, we calculate the subspace constrained mean-shift update,  $\mathbf{ms}_\perp(\mathbf{x})$ , and check if the projection of a voxel ends up in the same (starting) voxel or not.

Let  $\mathbb{Y}$  denote the principal subset of the data samples chosen from the projected voxel locations  $\mathbb{Y} = \{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_N\} \in \mathbb{R}^n$  such that  $\bar{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{ms}_\perp(\mathbf{x}_i)$  satisfying  $\mathbf{ms}_\perp(\mathbf{x}_i) < \frac{1}{2}\mathbf{1}$ , where  $N$  is the number of voxels in the dataset and  $\mathbf{1} \in \mathbb{R}^n$  is unit column vector. Fig. 2-a shows the projected principal subset of the synthetic tree data. Note that  $\mathbb{Y}$  is a set of projected samples (not all the data points) that are close to the centerline of the curvilinear structure. Ideally, the PCT method described before will trace these samples of  $\mathbb{Y}$  during iterations, resulting in a complete and continuous tree like structure. In a recursive implementation scheme, to reduce computations and allocated space, a residual strategy can be adopted. In the following sections, assume that  $\mathbb{Y}_k$  represents the residual of the principal projected samples that have not been visited by the previous  $k-1$  PCT iterations, such that after each termination of PCT algorithm, the traced

branch and its neighborhood is deleted from the principal subset  $\mathbb{Y}_{k+1} \leftarrow \mathbb{Y}_k \setminus \mathbb{B}_k$ , where  $\mathbb{B}_k$  represents the  $k^{\text{th}}$  traced branch (curve) using the PCT algorithm. We assumed that points in the principal subset that are  $thr_{Low}$  voxels away from the traced branch are considered as neighborhood and deleted from  $\mathbb{Y}_k$ . Fig. 2-a shows the initial principal subset  $\mathbb{Y}_1$ . Red dots in Fig. 2-b represents the first traced branch  $\mathbb{B}_1$ . Samples in the subset  $\mathbb{Y}_k$  that are close to the traced branch can be found in terms of their distance to the traced curve.  $\mathbb{S}_k = \{\dots \mathbf{x}_j \dots\} \in Y$  represents the set of data points in the principal subset  $\mathbb{Y}_k$  such that  $thr_{Low} < d(\mathbf{x}_j - \mathbb{B}_k) < thr_{High}$ . Here,  $d(\mathbf{x}_j - \mathbb{B}_k)$  is the Euclidean distance from the point  $(\mathbf{x}_j)$  to the closest sample point on the curve. In general, samples in  $\mathbb{S}_k$  form tight cluster lying on 1D manifolds since they are samples from the close branches that have already been projected to the principal manifolds. Detection and evaluation of these clusters in principal subsets fulfill the recursive cycle of the proposed algorithm. *Branching principal subset clusters:* Mean shift method is an unsupervised clustering algorithm that associates datapoints converging to the same density (or cluster) mode. Datapoints converge through fixed-point iterations as described in (9). Notice that, subspace constrained algorithm is a projection of the MS update on the current estimate of the normal subspace  $S_\perp(\mathbf{x})$ . Instead of projecting the data points to subspaces, if one uses the whole space, iterations will result in convergence to the local density mode instead of the ridge.

Once all the points converge to their corresponding modes, the association of the converged data points and the detection of clusters are straightforward, and it can be achieved by applying connected component analysis. In order to detect the bifurcating branches and to obtain corresponding mode locations constrained to the data samples, we used the method described in Sheikh et al (2007) forcing mode seeking iterations to the datapoints. In Fig. 2-a, for a given seed location (yellow cross) and direction (assumed bottom to top), red dots represent the traced branch, Fig. 2-b shows the detected clusters after the first termination of the PCT algorithm. Different colors represent detected nearby principal subsets. Fig. 2-c shows a close-up view of one of the detected clusters with its mode (yellow dot). Red dots represent the tracing iterations in subpixel accuracy of the PCT algorithm.

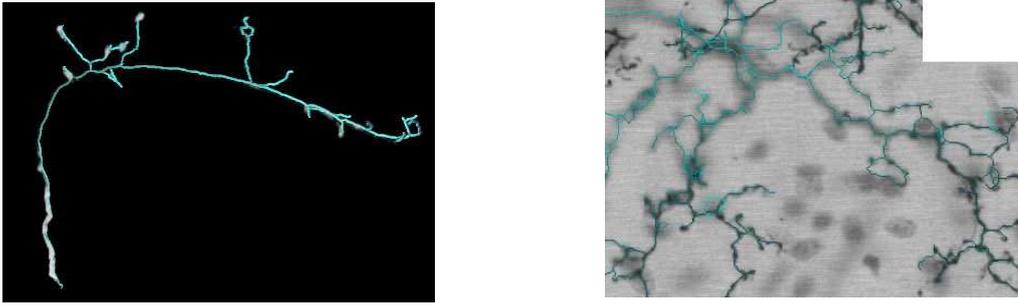
Every detected cluster is tested to find whether it is connected to the traced branch,  $\mathbb{B}_k$  or not. For the  $k^{th}$  traced branch, evaluation of detected clusters is achieved by initializing two PCT algorithms for every detected cluster. Obtained cluster modes are used as the seed locations. The eigenvector,  $\mathbf{V}_1$ , which corresponds to the smallest eigenvalue  $\lambda_1$  evaluated at the seed location, is selected as the initial direction of the curve  $\gamma_{0,k}$ . Similarly, the second PCT is initialized for the opposite direction of the  $\gamma_{0,k}$  for the same seed point. If any of these two PCT trajectories of the tested clusters converges to the  $\mathbb{B}_k$ , the center of the cluster is attached to the  $\mathbb{B}_k$  and a new PCT is initialized for the detected branch. Fig. 2-d displays the output of the final recursion ( $k = 13^{th}$  iteration) of the algorithm. Each recursion traces a fiber segment ( $\mathbb{B}_k : k = 1 \dots 13$ ) that is labeled with a different color-shape marker. Notice that each PCT iteration traces a fiber segment, not a piecewise dendrite segment and the number of the recursion is same as the number of branches. Figure 3-a displays a single slice from a 3D cerebellar climbing image stack illustrating an intermediate recursion step. To give a better idea about the process, instead of displaying nearby residual principal subset samples that are close to a single branch, we created a figure showing all principal subset samples that are nearby to all traced branches. Here, red dots represent residual principal subset samples. Traced branches are labeled with different colors and shapes (red is reserved for the principal subset samples). Fig. 3-b displays a zoomed-in version, and clustered mode locations of the principal subsets are shown with yellow circles. Be aware that 2D projection of the tracing results are directly displayed on an image slice which might result in possible apparent (but not real) mismatches.

## 5 Application to confocal images

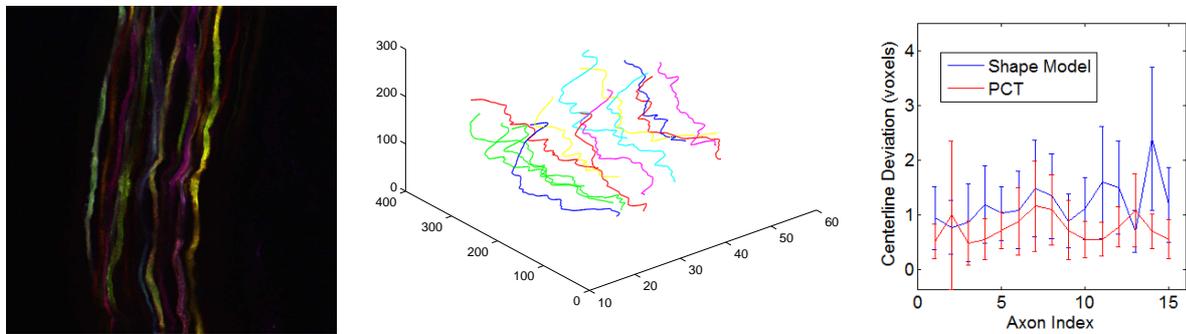
### 5.1 Datasets

We tested the recursive PCT algorithm on olfactory projection fiber, cerebellar climbing fiber, and neuromuscular projection fiber (Brainbow) datasets. We used an  $\epsilon$ -ball having a radius sufficient to cover  $N_x = 250$  neighbors (approximate radius of 7 voxels). In general, we used fixed bandwidth for both spatial and the intensity (color) kernels. We assumed that fiber radius and the intensity through the branches do not change dramatically, and used leave-one out cross validation to estimate the spatial and intensity kernels for the seed location, and used these values for the whole dataset for the first two datasets. At each step, direction of the principal curve is calculated ( $\gamma$ ) as described in Table 2. The immediate neighbor voxel center that is closest to the direction pointed by  $\gamma$  is selected as the next approximate curve sample.

Olfactory dataset has 34 images and each image has 512x512 pixels. Since voxel spacings are same in x, y and z directions, same kernel widths are used for all spatial kernels to simplify calculations. Fig. 4-a shows the 3D reconstruction obtained from the recursive PCT algorithm overlaid with the image stack. Similarly, cerebellar climbing fiber dataset has 28 slices, each having 580x352 pixels with isotropic voxel spacing. Here,  $thr_{Low}$  is selected as 4 voxels for both olfactory and cerebellar datasets. Fig. 4-b displays a 2D projection of the 3D trajectory overlaid with a region of single 2D slice. We also report qualitative and quantitative results using Brainbow images as shown in Fig. 5-a. This recent imaging technique labels individual fibers with different colors enabling easier interpretation by researchers. Although color distinction is not perfect, these images are highly informative for the analysis of neuronal connectivity. Brainbow dataset used in our experiments is composed of 62 slices each having 342x342 pixels without having any bifurcations. Similar to the previous datasets, voxel spacing is isotropic. Kernel covariances  $\Sigma_i^P$  and  $\Sigma_i^C$  are selected as  $\sigma_p^2 \mathbf{I}$  and  $\sigma_c^2 \mathbf{I}$  where  $\sigma_p = 5$  and  $\sigma_c = 0.3$ . Since fiber colors vary along the axis, we employed moving average smoothing for the colors to adapt to the spatial color change along each axon. To display the axon trajectories, another moving average filter is used to smooth trajectories. Fig. 5-b shows the trajectories of some selected axons. Since there are no bifurcations, we simply traced neuronal fibers separately using the PCT algorithm. We compared the PCT algorithm with a shape driven tracing algorithm proposed in Bas and Erdogmus (2010a) in terms of centerline deviation. For each slice, average displacement between the ground-truth and algorithms' outputs are displayed in Fig. 5-c. Error bars represent the standard deviation of error for each fiber trace.



**Fig. 4** a) 3D reconstruction with Olfactory image overlay. b) Cerebellar Images. Traced branches are represented with cyan color in both images.



**Fig. 5** a) Brainbow image slice. b) Tracing of selected axons using principal curve tracing. c) Calculated centerline deviation of algorithm results. At  $\alpha = 0.01$  significance level hypothesis stating means of PCT and method described in Bas and & Erdogmus (2010a) are equal, is rejected for all axons.

## 6 Conclusion

In order to achieve a complete wiring diagram, robust, reliable, and efficient algorithms are needed. In this study we developed a recursive tracing method to obtain connectivity of neuronal structures in various confocal microscopy images. Proposed method employs a principal curve tracing method which highlights the ridge underlying the intensity distribution. Estimation of the density function is achieved through the use of well studied KDE technique (which, in this case acts as a smooth interpolation technique), and the ridge of the density estimate is traced by PCT algorithm recursively.

Effectively, the whole process is an extension to the mean-shift (MS) clustering algorithm, seeking local conditions to extract the ridge of the intensity function, and utilizing subspace analysis to trace it. At the beginning, as a preprocessing step downsampled data points are projected to the closest principal curve. Tracing algorithm then lets the trajectory progress through a fiber in which the seed is given; the algorithm starts by using MS to project the seed to the nearest mode, followed by a step-correct cycle based on the local

tangent and orthogonal subspace eigenvectors of the Hessian at the current location. Recursive implementation of the algorithm is an extension to the original algorithm where detection and tracing of branch candidates are also handled by the same tracing framework.

In this study, we simply assumed that the fiber radius is fixed throughout the data space, and we did not mention problems related to KDE estimation, e.g. kernel bandwidth selection, selection of  $\epsilon$ -ball for the KNN on which the kernel support is bounded. However, one important concept is the selection of the kernel type. In our calculations, we used Gaussian kernels both for intensity (color) and spatial features due to its simplicity and compactness at its first and second derivative formulation. Although selected kernel bandwidth and type are sufficient for the tested datasets as displayed here, a better strategy would be to employ flat-top kernels that represent and highlight the volume of the axons better. In relation to that, the local covariance defined in the PCT algorithm uses the logarithm of the density function to better estimate the local geometry. Similarly, a flat-top kernel will be more compatible with the underlying inten-

sity distribution.

By defining voxel relations in terms of probabilities, we prevent the problems due to the variations in the image contrast. Since the termination of branches is given in terms of probabilities, the local contrast variations are also imposed to the estimated density, enabling us to define global termination probability that can be used throughout the datasets. However, estimation of density has its drawbacks: since kernel density estimation methods suffer from the curse of dimensionality, we follow a more simple but effective method interpolating intensity in the spatial domain only. Moreover, estimation of the density is limited to the bounded kernel that is defined by the KNN to make computations faster. As an implementation detail, a faster version of the algorithm can be implemented by replacing MS iterations with a medoidshift updates, since previously calculated pairwise distances during KNN calculations can also be used to calculate medoidshift updates.

Depending on the complexity of the neuronal structures, resulting connectivity trees might have unwanted loops. The main reason for this error is the lack of high resolution in images, especially in depth direction. To prevent these loops, we applied a minimum spanning tree Cormen et al (2001) algorithm to the obtained connectivity graph in order to disconnect unnecessary edges. Edges in the graph are weighted based on corresponding voxels' intensity value. In addition, another interesting observation is the path selective graph generation. Since our approach traces branches recursively, based on the iteration order, different tree representations can be obtained due to the loops in the graph. For example, after the tracing of the  $i^{th}$  branch,  $\mathbb{B}_i$ , assume that there are 2 detected principal subset clusters,  $C_{S_i,1}$  and  $C_{S_i,2}$  around  $\mathbb{B}_i$ , where evaluation of these clusters converge to  $\mathbb{B}_i$ . Moreover, assume that these clusters are connected to each other as well due to low resolution artifacts, and they form a closed loop with  $\mathbb{B}_i$ . Since the first PCT algorithm will also trace the second cluster, we will get an empty branch for the second one. It is because, the tracing iteration for the latter will terminate since the current tracing location will hit a previously traced structure as explained in Table 2 as soon as the tracing starts. This implies that, depending on the order of execution, parent and child nodes of a tree will switch. One method to eliminate this error (which is actually due to the resolution of images) is the repetitional tracing. For every recursive iteration, permutation of the tracing order can also be evaluated, and in the end ensemble of graphs can be used to get a final graph and therefore a final tree. However, this process will become infeasible as the number of branches increase in the images, therefore the neuronal structures get complicated. In fact, one can also approach this problem by using a perturbation analysis on the connectivity graphs as discussed in Carreira-Perpinán and Zemel (2005), there-

fore our future work includes the investigation of such methods.

## 7 Information Sharing Statement

The Matlab codes, all presented pictures and additional animations are publicly available under <http://ece.neu.edu/~bas/PCT>

**Acknowledgements** The authors would like to thank the organizers and data providers of the Diadem Challenge for putting together this challenge.

## References

- Al-Kofahi K, Can A, et al (2004) Median-based robust algorithms for tracing neurons from noisy confocal microscope images. *IEEE Transactions on Information Technology in Biomedicine* 7(4):302–317
- Aykac D, Hoffman E, et al (2003) Segmentation and analysis of the human airway tree from three-dimensional x-ray ct images. *IEEE Transactions on Medical Imaging* 22(8):940–950
- Bas E, & Erdogmus D (2010a) Piecewise linear cylinder models for 3-dimensional axon segmentation. In: 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro
- Bas E, & Erdogmus D (2010b) Principal curve tracing. In: European Symposium on Artificial Neural Networks
- Cai H, Xu X, et al (2006) Repulsive force based snake model to segment and track neuronal axons in 3d microscopy image stacks. *Neuroimage* 32:1608–1620
- Cai H, Xu X, et al (2008) Using nonlinear diffusion and mean shift to detect and connect cross-sections of axons in 3d optical microscopy images. *Medical Image Analysis* 12:666 – 675
- Carreira-Perpinán M, & Zemel R (2005) Proximity graphs for clustering and manifold learning. *Advances in neural information processing systems* 17:225–232
- Cetingul H, Plank G, et al (2009) Stochastic Tractography in 3-D Images via Nonlinear Filtering and Spherical Clustering. In: MIC-CAI Workshop on Probabilistic Models for Medical, PMMIA, pp 268–279
- Chang K, & Ghosh J (2002) A unified model for probabilistic principal surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23(1):22–41
- Chen J, Sato Y, et al (2002) Orientation space filtering for multiple orientation line segmentation. *IEEE Transactions in Pattern Analysis and Machine Intelligence* 22(5):417–429
- Cormen T, Leiserson C, et al (2001) Introduction to algorithms. The Algorithms of Kruskal and Prim. The MIT press and McGraw-Hill
- Deschamps T, & Cohen L (2001) Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis* 5(4):281–299
- Einbeck J, Tutz G, et al (2005) Local principal curves. *Statistics and Computing* 15(4):301–313
- Erdogmus D, & Ozertem U (2007) Self-consistent locally defined principal surfaces. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol 2, pp 549–552
- Frangi A, Niessen W, et al (1999) Model-based quantitation of 3-D magnetic resonance angiographic images. *IEEE Transactions on Medical Imaging* 18(10):946–956
- Hastie T, & Stuetzle W (2003) Principal curves. *Journal of the American Statistical Association* 84(406):502–516

- Huang S, Li J, et al (2009) Learning brain connectivity of Alzheimer's disease from neuroimaging data. *Advances in Neural Information Processing Systems* 22:808–816
- Jurrus E, Hardy M, et al (2009) Axon tracking in serial block-face scanning electron microscopy. *Medical Image Analysis* 13(1):180–188
- Kegl B, Krzyzak A, et al (2000) Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(3):281–297
- Meinicke P, Klanke S, et al (2005) Principal surfaces from unsupervised kernel regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp 1379–1391
- Palágyi K, & Kuba A (1998) A 3d 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Letters* 19(7):613–627
- Rodríguez A, Ehlenberger D, et al (2009) Three-dimensional neuron tracing by voxel scooping. *Journal of Neuroscience Methods* 184(1):169–175
- Schmitt S, Scholz M, et al (2004) New methods for the computer-assisted 3d reconstruction of neurons. *NeuroImage* 23(4):1283–1298
- Sheikh Y, Khan E, et al (2007) Mode-seeking by medoidshifts. In: *Proceedings of the IEEE International Conference on Computer Vision*, Citeseer, vol 141, pp 1–8
- Vasilkoski Z, & Stepanyants A (2009) Detection of the optimal neuron traces in confocal microscopy images. *Journal of Neuroscience Methods* 178(1):197–204
- Wang J, Zhou X, et al (2007) Dynamic local tracing for 3d axon curvilinear structure detection from microscopic image stack. In: *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp 81–84
- Wink O, Niessen W, et al (2004) Multiscale vessel tracking. *Medical Imaging, IEEE Transactions on* 23(1):130–133
- Zhou W, Li H, et al (2008) 3d dendrite reconstruction and spine identification. In: *Proceedings of MICCAI'08, Part II*, pp 18–26