Journal of Neuroscience Methods 214 (2013) 210–222

Contents lists available at SciVerse ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Computational Neuroscience

Neurient: An algorithm for automatic tracing of confluent neuronal images to determine alignment

Jennifer A. Mitchel[a,b], Ian S. Martin[c], Diane Hoffman-Kim[a,b,*]

[a] Department of Molecular Pharmacology, Physiology, and Biotechnology, Brown University, Providence, RI 02912, USA
[b] Center for Biomedical Engineering, Brown University, Providence, RI 02912, USA
[c] Redwire Consulting, LLC 8 Thorndike St. Everett, MA 02149, USA

HIGHLIGHTS

- We present Neurient, an algorithm developed to trace dense fluorescent neurite images.
- Neurient allows unsupervised image processing, with optional tunable parameters.
- Method: find seed points, index into a directed lookup table, trace neurite segments.
- Trace coordinates and orientations can be used to evaluate neurite alignment.
- The Neurient code is open source and freely available for use and customization.

ARTICLE INFO

Article history:
Received 26 September 2012
Received in revised form 25 January 2013
Accepted 25 January 2013

Keywords:
Neurite outgrowth
Fluorescence microscopy
Neurite tracing
Computer-assisted image analysis
High throughput
Open source

ABSTRACT

A goal of neural tissue engineering is the development and evaluation of materials that guide neuronal growth and alignment. However, the methods available to quantitatively evaluate the response of neurons to guidance materials are limited and/or expensive, and may require manual tracing to be performed by the researcher. We have developed an open source, automated MATLAB-based algorithm, building on previously published methods, to trace and quantify alignment of fluorescent images of neurons in culture. The algorithm is divided into three phases, including computation of a lookup table which contains directional information for each image, location of a set of seed points which may lie along neurite centerlines, and tracing neurites starting with each seed point and indexing into the lookup table. This method was used to obtain quantitative alignment data for complex images of densely cultured neurons. Complete automation of tracing allows for unsupervised processing of large numbers of images. Following image processing with our algorithm, available metrics to quantify neurite alignment include angular histograms, percent of neurite segments in a given direction, and mean neurite angle. The alignment information obtained from traced images can be used to compare the response of neurons to a range of conditions. This tracing algorithm is freely available to the scientific community under the name Neurient, and its implementation in MATLAB allows a wide range of researchers to use a standardized, open source method to quantitatively evaluate the alignment of dense neuronal cultures.

1. Introduction

Injuries to the peripheral and central nervous systems are the focus of a large body of clinical research (Dobkin and Havton, 2004; Chen et al., 2007). Following nervous system injuries, the extracellular environment greatly impacts the degree of regeneration possible (Busch and Silver, 2007). Strategies for directing nerve growth have been investigated both in vivo and in vitro (Geller and Fawcett, 2002; Schmidt and Leach, 2003; Schlosshauer et al., 2006; Li and Hoffman-Kim, 2008; Ichihara et al., 2008; Hoffman-Kim et al., 2010). Methods to evaluate neuronal response to environmental cues vary from study to study, and response metrics have included the number of cells forming neurites or axons (Foley et al., 2005; Gomez et al., 2007), the average number of neurites per cell (Mahoney et al., 2005), the average neurite length per cell (Fournier et al., 2003; Haq et al., 2007), the length of the longest neurite or axon (Gomez et al., 2007), the total neurite extension length (Houchin-Ray et al., 2007; Koh et al., 2008), and the degree of neurite branching (Mahoney et al., 2005).

In the context of informing repair strategies for nerve injuries, neuronal alignment is an important indicator of the ability of neurons to respond to a material. A number of studies have presented a directive cue to navigating neurons, which aligned preferentially

* Corresponding author at: Department of Molecular Pharmacology, Physiology, and Biotechnology, Brown University, Box G-B393, 171 Meeting Street, Providence, RI 02912, USA. Tel.: +1 401 863 9395.

0165-0270/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jneumeth.2013.01.023

to the cue (Kim et al., 2008; Kofron et al., 2010). Alignment can be measured for the cell soma, the cellular extensions, or both. Somal or nuclear alignment can be used to indicate alignment relative to a directive cue, and this method is especially useful for cell types which may undergo a whole-cell response to cues, such as many glial cell types (Kofron and Hoffman-Kim, 2009). Neurite alignment has been evaluated by a number of approaches, including subjective categorization (Miller et al., 2001, 2002), evaluation of outgrowth direction relative to a cue (Dowell-Mesfin et al., 2004; Walsh et al., 2005; Hanson et al., 2009), and overall alignment over an entire field of view (Johansson et al., 2006; Corey et al., 2007).

Automated image analysis has recently gained attention as a method to produce high throughput, unbiased evaluation of complex image-based data (Rittscher, 2010). Methods to automate the evaluation of neuronal response to extracellular cues have included soma segmentation, nuclear analysis, and neuronal tree segmentation (reviewed in Meijering, 2010). Approaches for segmentation of the neuronal tree can be roughly categorized into "global processing" or "local exploration" (Meijering, 2010). Most global processing algorithms implement the following sequence of operations: binarization, skeletonization, rectification, and graph representation (Leandro et al., 2009). After binarization, a common step is to extract the centerlines of the segmented areas (Rodriguez et al., 2009). Global processing algorithms typically work best on uniformly high quality images, may be computationally wasteful, and identify critical points only in the last stages of processing (Meijering, 2010). The most common global processing segmentation methods are line pixel algorithms, which use a model to examine each pixel and find the local geometric properties of the lines, followed by a linking process to connect the detected centerline points into connected centerlines (Huang et al., 2010).

In contrast to global processing methods, "tracing" tends to refer to local exploration methods. Local exploration starts with the detection of topologically relevant points, and iteratively predicts the next point along a neurite (Can et al., 1999; Al-Kofahi et al., 2002). In particular, vector tracing or exploratory algorithms detect starting, or seed, points and trace the centerlines from these initial points until pre-defined stop conditions are met (Zhang et al., 2007a,b).

Despite the publication of a range of algorithms to detect, trace, or analyze neurites and neuronal trees, many studies are still evaluated qualitatively, or by either manual or semi-manual tracing (Ascoli, 2008; Gertz et al., 2010; Meijering, 2010). Commercially available software packages for automated measurement of neurite outgrowth may be prohibitively expensive, while algorithms in the literature may not be accessible to researchers with limited programming experience and are often published without the source code (Pool et al., 2008). Limitations of current tracing methods have been discussed elsewhere and include sensitivity to noise, inability to detect neurites with a faint fluorescent signal, and requirements for manual neurite selection, noise elimination, and/or cell body demarcation (Wu et al., 2010). In addition, some methods require that neurites be well separated, and work best when few non-neuronal cells are present (Pool et al., 2008), which is not typical of many neurite outgrowth model culture systems.

Here we present a novel tracing algorithm that uses aspects of both global and local methods and is implemented for MATLAB. An earlier version of this algorithm has been used successfully in a previous study to acquire alignment data from neurite images (Kofron et al., 2010). Our code, under the name Neurient, is freely available under the GNU General Public License at http://www.neurient.org. Using the default parameters, no manual input from the user is required to use Neurient, and in theory, unlimited volumes of data can be processed. The source code is freely available, allowing members of the scientific community to experiment with it and tailor it to their own specific needs. Further, because Neurient is written in a high-level scripting language and does not require compiling, it is accessible to users with minimal programming expertise. In this paper we describe the details of the tracing algorithm and its implementation. We compare our method with previously published open source tracing methods and semi-manual neurite tracing and demonstrate its efficacy at tracing complex neurite images and extracting relevant neurite orientation information.

2. Materials, methods and tracing algorithm

Quantitative evaluation of images of neurites from primary neuronal cultures was the principal application motivating the design and implementation of Neurient. An overview of the image tracing algorithm is shown in Fig. 1, where it is depicted as three phases. Neurient first computes a large data set for each image, which is referred to as the kernel response lookup table. In parallel, Neurient determines a set of points on the image, referred to as seed points, that fit the criteria to be putative points along neurites. Taking these two data sets together, Neurient's final step is iterative tracing in which each seed point is used as a starting location for a trace path, where the directions associated with each path are determined by the information stored in the kernel response lookup table. Generation of a kernel response lookup table (Phase I), finding the seed points (Phase II), and exploratory tracing (Phase III) are each discussed in the following sections, as is a brief experimental overview describing the cells and culture conditions used to generate our test images. Neurient was developed and tested using both MATLAB (R2008b) and GNU Octave (v 3.2).

2.1. Phase I: generation of the kernel response lookup table

The kernel response lookup table (KRLT) is a computed data structure that contains orientation information for the input image. The kernel, discussed in detail in Section 2.1.3, is a matrix that is cross-correlated with the input image to produce a response image in which areas of the input image parallel to the kernel direction are highlighted. The kernel responses are computed globally for all N_θ angles of interest over every part of the input image, as MATLAB is not optimized for computing kernel responses piecemeal during the exploration phase. These responses are stored in a lookup table for later retrieval during the exploration phase.

The process used to generate the KRLT is summarized in Fig. 2 for a kernel of length k. First, the $L \times W$ input image is band-limited to eliminate high-frequency components. It is then padded with zeros to produce a larger square image. This padded image can be rotated by any angle without loss of image information due to clipping. For each angle θ_i, the padded image is first rotated clockwise by θ_i, translated to the right by $\lfloor k/2 \rfloor$, cross-correlated with the double-edge detection kernel, then rotated back by $-\theta_i$ to produce a kernel response image for that particular angle θ_i. The kernel response images are cropped back to the original image dimensions $L \times W$ and compiled into a three-dimensional lookup table of dimension $L \times W \times N_\theta$. These steps are described in greater detail in the following sections.

2.1.1. Band-limiting of the input image

Discrete image rotation is generally an irreversible process in which some image information is lost (Parker et al., 1983). Except in cases of rotation through the cardinal angles (0°, 90°, 180°, 270°), pixel coordinates in a rotated image rarely correspond directly to pixels in the original image. Instead, the rotated coordinates often correspond to a point somewhere in between neighboring pixels in the original. For this reason, some form of interpolation is typically used to estimate the intensity of these interstitial coordinates. It is this interpolation that results in a loss of some high-frequency image information.

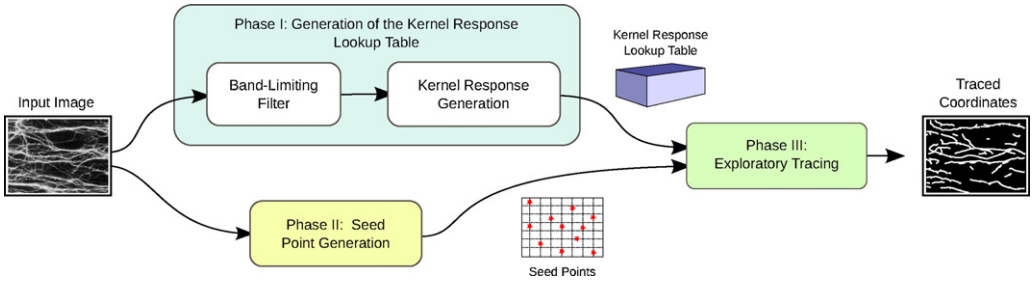

Fig. 1. Tracing algorithm information flow.

Such loss can also be illustrated in the frequency domain. Since the two-dimensional Fourier transform (FT) is rotationally invariant, rotating an image by a given angle is equivalent to rotating its FT by the same angle in the frequency domain (Casasent and Psaltis, 1976). When the FT is rotated by an oblique angle, the four corners, as well as the high-frequency information they contain, are clipped because they lie outside the image boundaries following rotation (Owen and Makedon, 1996). This phenomenon is illustrated in Fig. 3.

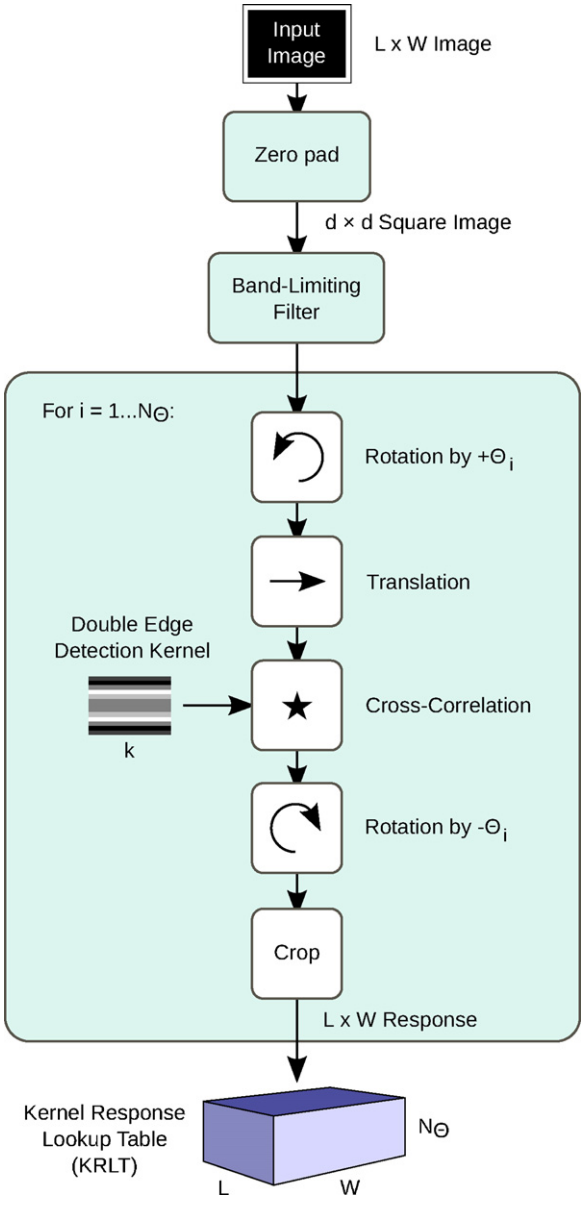

Fig. 2. Phase I overview: generation of the kernel response lookup table. The input image is pre-processed by the band-limiting and padding steps. The pre-processed image is then rotated by θ_i, translated, cross-correlated with the kernel, and rotated by $-\theta_i$. The zero-padding is removed, and the kernel response image for θ_i is stored in the ith layer of the kernel response lookup table.

In order to mitigate any bias due to this non-uniform loss of high-frequency information, we intentionally band-limit the input image by applying the equivalent of an ideal low-pass filter as a preprocessing step. In this step, the input image is first padded with zeros to produce a square image of size $d \times d$, where $d = \left\lceil \sqrt{L^2 + W^2} \right\rceil$ is the diagonal length of the input image, and then its discrete Fourier transform (FT) is computed. Then a disc-shaped mask of radius $d/2$ is applied, thereby eliminating high frequency components beyond a certain radius. This masked FT can be rotated by any arbitrary angle without being clipped by the image boundaries, and similarly its spatial counterpart could be rotated with minimal loss of high-frequency information. Finally, an inverse Fourier transform is applied to the masked FT to obtain the band-limited image in the spatial domain. In this section, we have described a process to generate a band-limited image which will, unlike the original input image, result in minimal information loss when it is rotated during subsequent steps of generating the KRLT.

2.1.2. Comparison of image rotation methods

During the Phase I generation of the KRLT, the input image is rotated by N_θ discrete angles. There are numerous image rotation methods available, but the most commonly used methods are: nearest-neighbor, bilinear, bicubic, and Fourier. This section explains the process used to select the best rotation method for Neurient.

Nearest-neighbor is the simplest form of interpolation, in which the intensity of the pixel closest to the rotated coordinates is used. Bilinear and bicubic interpolation fit separate second- or third-order polynomials, respectively, to the row and column of neighboring pixels and use them to estimate the intensity of the rotated coordinate. The Fourier method accomplishes rotation by performing a sequence of three shears using the Fourier transform (Owen and Makedon, 1996; Larkin, 1997).

In order to select a rotation/interpolation method that introduces minimal directional bias into the tracing algorithm, an experiment was performed using 100 random noise images with uniformly distributed intensity values for each pixel. These images were rotated through a range of angles between 0° and 180°. For each angle, a cross-correlation with a typical tracing kernel ($r = 0$, $k = 15$; see Section 2.1.3) was performed and a root mean square (RMS) kernel response was generated. The results are illustrated in Fig. 4A. In Fig. 4B this process was repeated for these same images after being processed by the band-limiting filter described in Section 2.1.1. The flatter responses in Fig. 4B indicate that the band-limiting filter is successful at reducing angular bias caused by the image rotation.

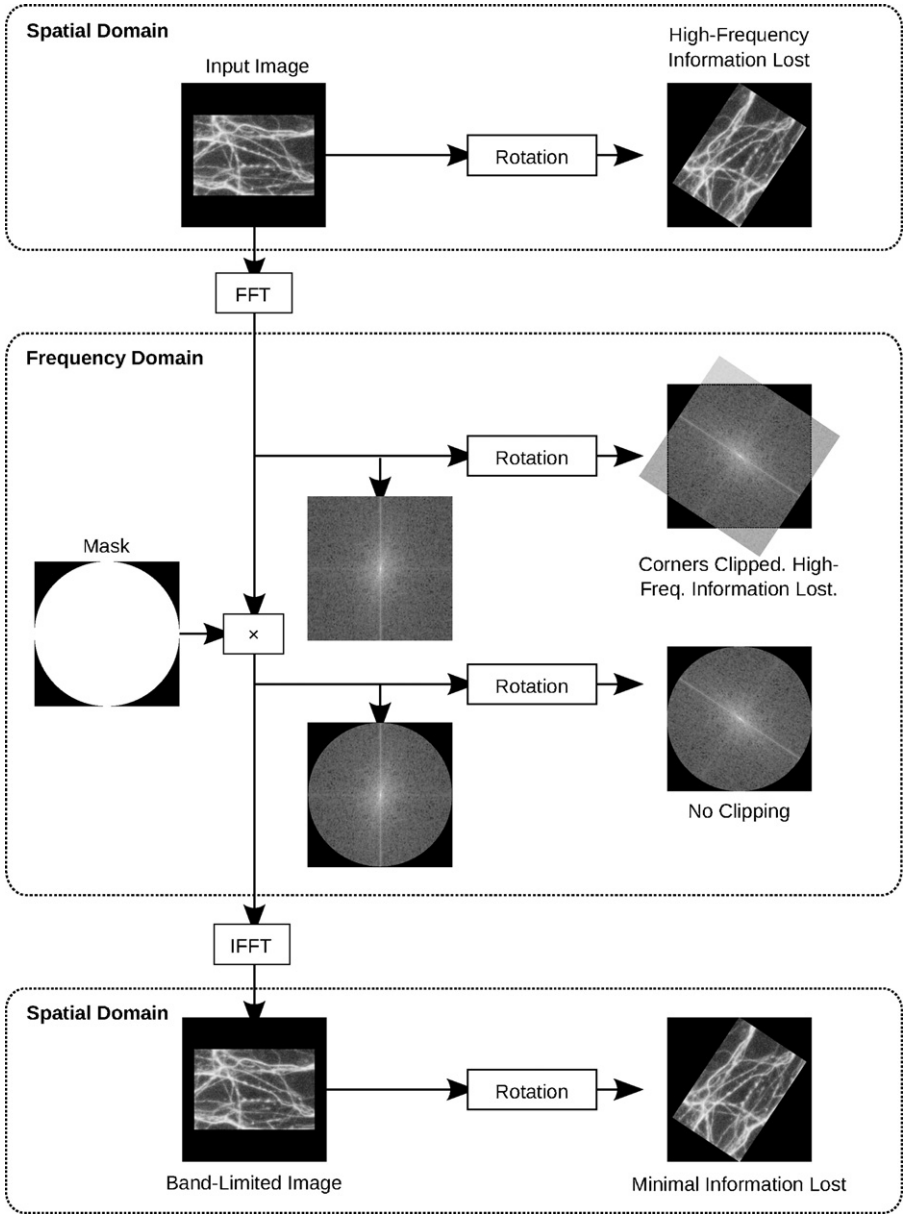

Fig. 3. Image band-limiting procedure. A disc-shaped mask is applied to the Fourier transform of the padded input image, uniformly eliminating high-frequency components that could be lost during rotation. An inverse Fourier transform is then applied to produce a new band-limited image which is superficially similar to the original, but contains less high-frequency information.

At the cardinal angles, all four rotation methods produce the same RMS intensity response. This is expected because at these angles every rotated coordinate corresponds directly to a pixel in the original image such that no interpolation is necessary. Nearest-neighbor interpolation introduces a bias in favor of oblique angles, while the bilinear, bicubic and Fourier methods introduce a bias against the oblique angles. For images pre-processed with the band-limiting filter described in Section 2.1.1, the Fourier method indicates the least variation in RMS kernel response over all angles and therefore the least potential directional bias. Therefore, the Fourier rotation method was chosen for use in generating the KRLT.

2.1.3. Cross-correlation kernel

During KRLT generation, the band-limited input image is rotated to each possible angle θ_i and cross-correlated with a kernel designed to detect local alignment of the image features. We adopted a zero-sum, double edge-detection kernel similar to kernels used previously (Al-Kofahi et al., 2002, 2003), as illustrated in Fig. 5. The kernel is a two-dimensional finite impulse response filter which consists of a one-dimensional double edge detection kernel $[-1\ -2\ 0\ 2\ 1\ \ldots\ 0\ \ldots\ 1\ 2\ 0\ -2\ -1]^T$ tiled k times, where k is a parameter that simultaneously determines the ability of the trace to follow curved structures, and its sensitivity to continuous structures. This kernel favors pixel intensity near the edges of the neurite and penalizes intensities just outside of the neurite. The width and length of the kernel can be adjusted by varying parameters r and k, respectively. The kernel width can be modified by adjusting r, as there are $2r+1$ zeros at the center of the kernel, and should be tuned to match the width of the features in the input image as closely as possible. Though it is also capable of detecting features of other widths, the kernel works best at detecting neurites of the ideal thickness t, where $t = 6+2r$, with a minimum width of 6 pixels. In cases where the features of interest are smaller than 6 pixels wide, it is possible to enlarge the input

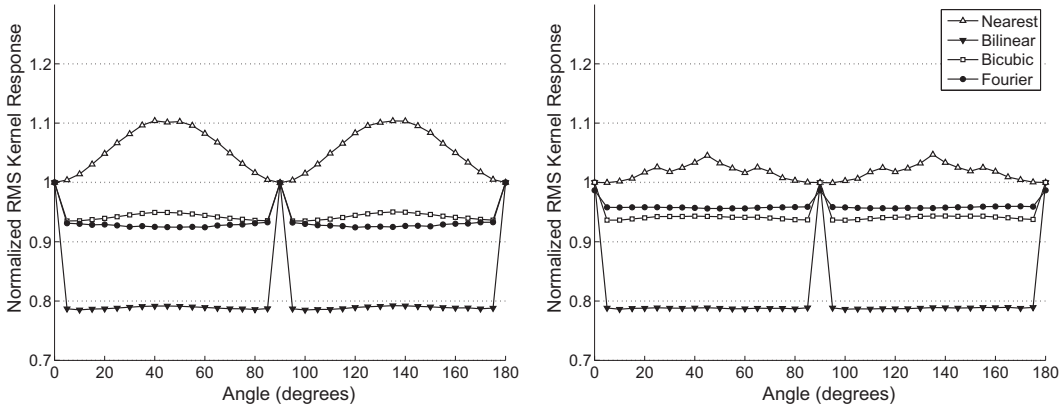

Fig. 4. Comparison of rotation algorithms. (A) 100 images of uniformly distributed random noise were rotated using nearest neighbor, bilinear, bicubic, and Fourier rotation methods. The normalized average RMS kernel response is plotted for each angle of rotation. (B) The same process was used as in A, with the added step that the random noise image was processed using the band-limiting step described in Section 2.1.1. Standard deviations are omitted for clarity, but consistently ranged between 0.01 and 0.016 for all methods.

image such that the features are, on average, at least 6 pixels wide, though in our studies this step did not appear to be necessary.

2.1.4. The kernel response lookup table

Generation of the KRLT, which is used during Phase III, required first optimizing the code for band-limiting the input image (Section 2.1.1), determining the ideal image rotation method (Section 2.1.2) and specifying the double edge detection kernel (Section 2.1.3). In the following section, we describe the KRLT itself in more detail.

We are interested in how the input image responds to the double edge-detection kernel at specific angles. In some cases, previous studies have used a finite set of different kernels, sometimes hand-drawn, corresponding to a set of evenly spaced angles (Can et al., 1999). Using a different kernel for each angle not only places an upper bound on the angular resolution of the traces, but also risks introducing a bias toward certain angles. In our algorithm, we use a novel approach in which the image is rotated while the kernel remains fixed. This permits arbitrarily high angular resolution while avoiding any bias that could be caused by using different kernels for each angle.

As indicated in Fig. 2, for each angle, the padded and band-limited input image is first rotated counterclockwise, then shifted to the right by $\lfloor k/2 \rfloor$. This shift is necessary to ensure that the kernel responses correspond to features in the vicinity of the point of interest, not directly over the point of interest, which allows distinction of angles in the full range from 0 to 360°. We then compute the cross-correlation of the entire rotated image with the fixed kernel, yielding a rotated kernel response image for a particular angle. The kernel response image is rotated back to its original orientation and cropped to remove the zero-padding. This step reduces the memory used and extra computations that would be required if the kernel response image were used directly in the exploration step without modification. The process is repeated for the rest of the angles of interest and the resulting kernel response images are appended to the KRLT.

The KRLT contains high intensity areas wherever an image feature is aligned with a particular angle (Fig. 6). Determining the angle of alignment for a particular point in the original image is therefore as simple as extracting a column of N_θ values and finding the index that corresponds to its maximum value. This technique is used during the exploration phase to determine the direction of the next step in the traced path.

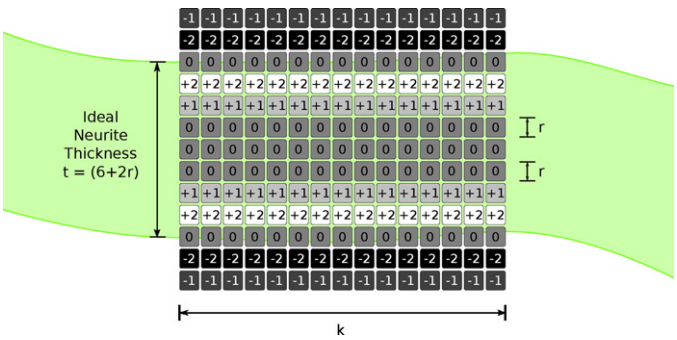

Fig. 5. The cross-correlation kernel, shown for parameters $r = 1$ and $k = 15$. The kernel detects edges that transition from bright on the inside to dark on the outside, favoring pixel intensity near the inside edge of the neurite and penalizing pixel intensity just outside of the neurite.

2.2. Phase II: determination of seed points

In order to identify putative neurite structures in the input images, a set of initial seed points is generated. Though many tracing algorithms require this step to be performed by hand, here we use a previously described method to obtain the seed points automatically (Al-Kofahi et al., 2002; Zhang et al., 2007a,b). A seed point is defined as a point which can be used as a starting point for exploratory tracing, and which sits on or near the centerline of a neurite segment. Many seed points may be detected along the same neurite, so criteria for preventing re-tracing segments are implemented during the exploratory tracing phase (Section 2.3).

The basic seed point detection method finds local maxima along a set of grid lines. We consider M vertical and N horizontal lines, spaced g pixels apart, and only look for seed points along these $M+N$ lines, rather than every pixel in the image. In order to remove high-frequency noise components, pixel intensities along the grid-lines are low-pass filtered using a one-dimensional Gaussian kernel approximation: $[0.25 \quad 0.5 \quad 0.25]^T$.

An intensity threshold is determined based on the median brightness of the image, defined by the formula $\Gamma = \gamma + \sigma/f$, where γ is the median pixel intensity, σ is the standard deviation around the median, and f is a user-defined parameter that helps tailor the trace results to a particular set of images (Al-Kofahi et al., 2003). Local maxima along the $M+N$ grid lines are determined, with at most one candidate seed point allowed per grid line segment, which is g pixels in length. Any candidate seed points with intensity less than Γ are discarded, as well as any seed points that are within a

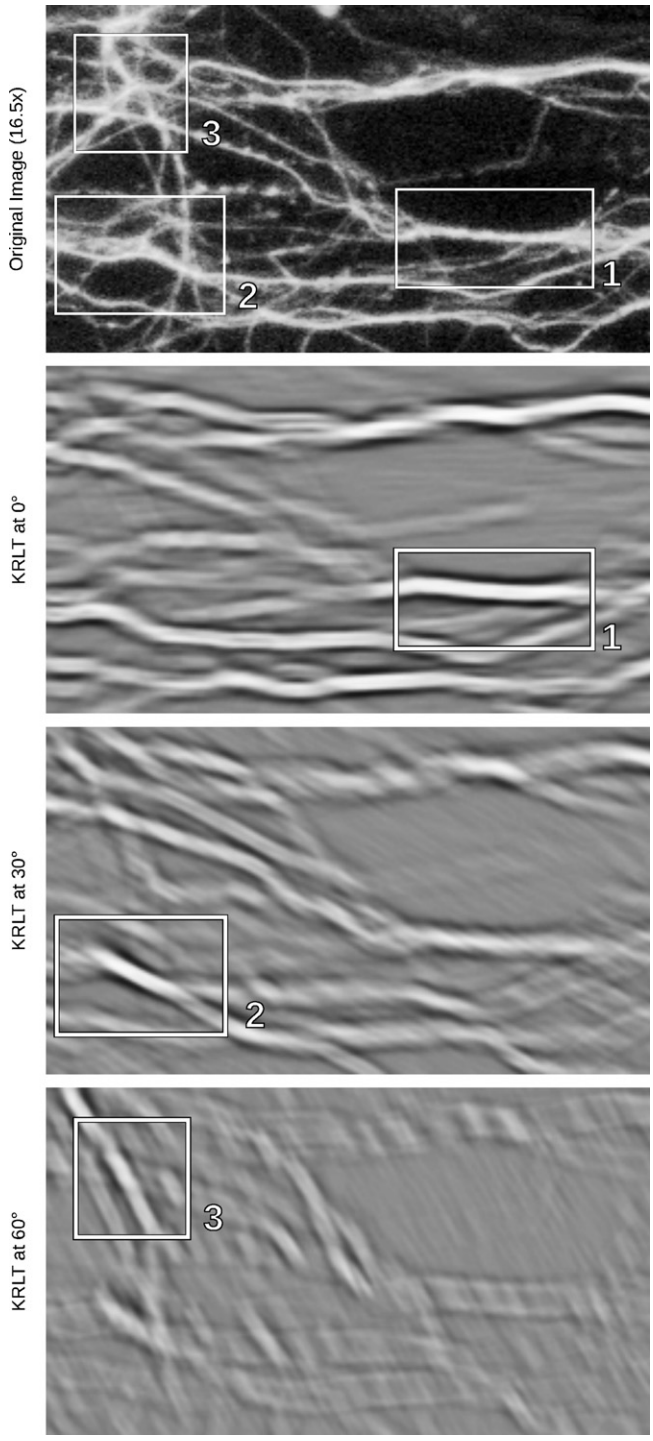

Fig. 6. Example slices of the kernel response lookup table. Features present in the original image (boxes) appear as bright areas with dark borders in the KRLT at the corresponding angle. A horizontal neurite in the original image (box 1) is apparent in the KRLT at 0°; similarly, lines at 30° and 60° in the original image (boxes 2 and 3, respectively) appear in the KRLT slices that correspond to those angles.

threshold distance b of the edge of the image, as determined by the following equation:

$$b = \left\lceil \sqrt{k^2 + (r+5)^2} \right\rceil \quad (1)$$

where k is the length of the kernel and r is the parameter that adjusts the kernel's width (see Fig. 5 and Section 2.1.3). b is the minimum distance required to keep the kernel within the image boundaries

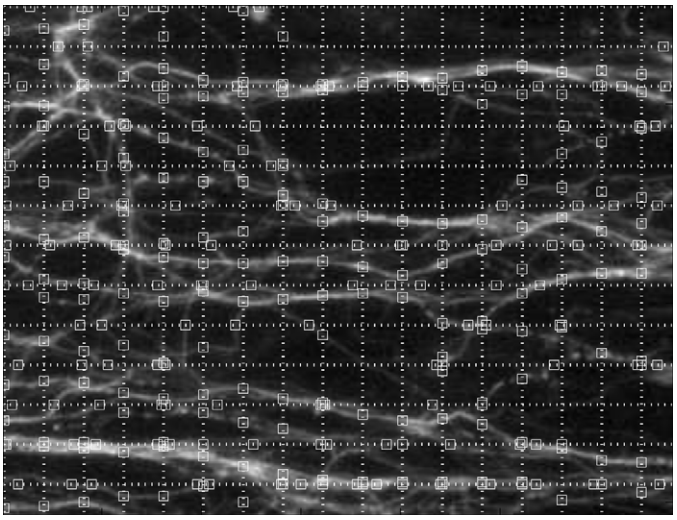

Fig. 7. Seed points. Grid lines are shown as dotted white lines, and seed points are shown as white squares. The majority of neurites have at least one seed point along their length.

for all angles θ_i. Fig. 7 illustrates valid seed points for a subregion taken from the central area of a larger image.

2.3. Phase III: exploratory tracing

Following the generation of the KRLT and the seed points, images are traced using the exploratory vector method described here and illustrated in Fig. 8. For each seed point, a trace path is generated, consisting of a series of adjacent segments that trace a neurite. A trace path contains a series of points, starting with a seed point, and associated directions corresponding to the direction of the underlying neurite segment. For each seed point, an initial direction is chosen by indexing into the kernel response lookup table at the point's coordinates and determining the direction corresponding to the maximum kernel response. The trace path is stepped forward along a line segment of length s in this direction and the location of the tip of this line segment is set as the new point. The new point is checked for violations as described below.

Violation criteria were determined so that trace paths terminate at the proper location. Reasons for termination of a trace path include if the trace path: is no longer following a neurite, is too close to the edge of the image, or is within a part of the image that has already been traced. Whether or not a trace path is still following a neurite is determined based on the intensity of the new point. If the intensity of the new point is less than the intensity threshold used to find the seed points, Γ, this is counted as a violation. If three consecutive violations of this type are found, the trace path for the current seed point terminates. The trace path is also terminated at the first instance of an edge violation, or at the first instance of retracing coordinates that have already been visited.

In the case that the violation criteria are not met, i.e. the new point has been determined to be a valid point along a neurite, the new point is appended to the trace path, and is considered to be the current point. From here, the current point is used to index into the KRLT, this time with the additional restriction that the only available directions are within $\pm 90^\circ$ of the previous direction. This criteria ensures that the trace path continues along a given neurite rather than double back on itself and get stuck in a local maximum. We note that this could potentially result in missing curves that are sharper than 90° along a length scale of size k, but such features are rare. Again the trace is stepped forward along a line segment of length s and checked for violations. This process continues along a given path corresponding to the initial seed point

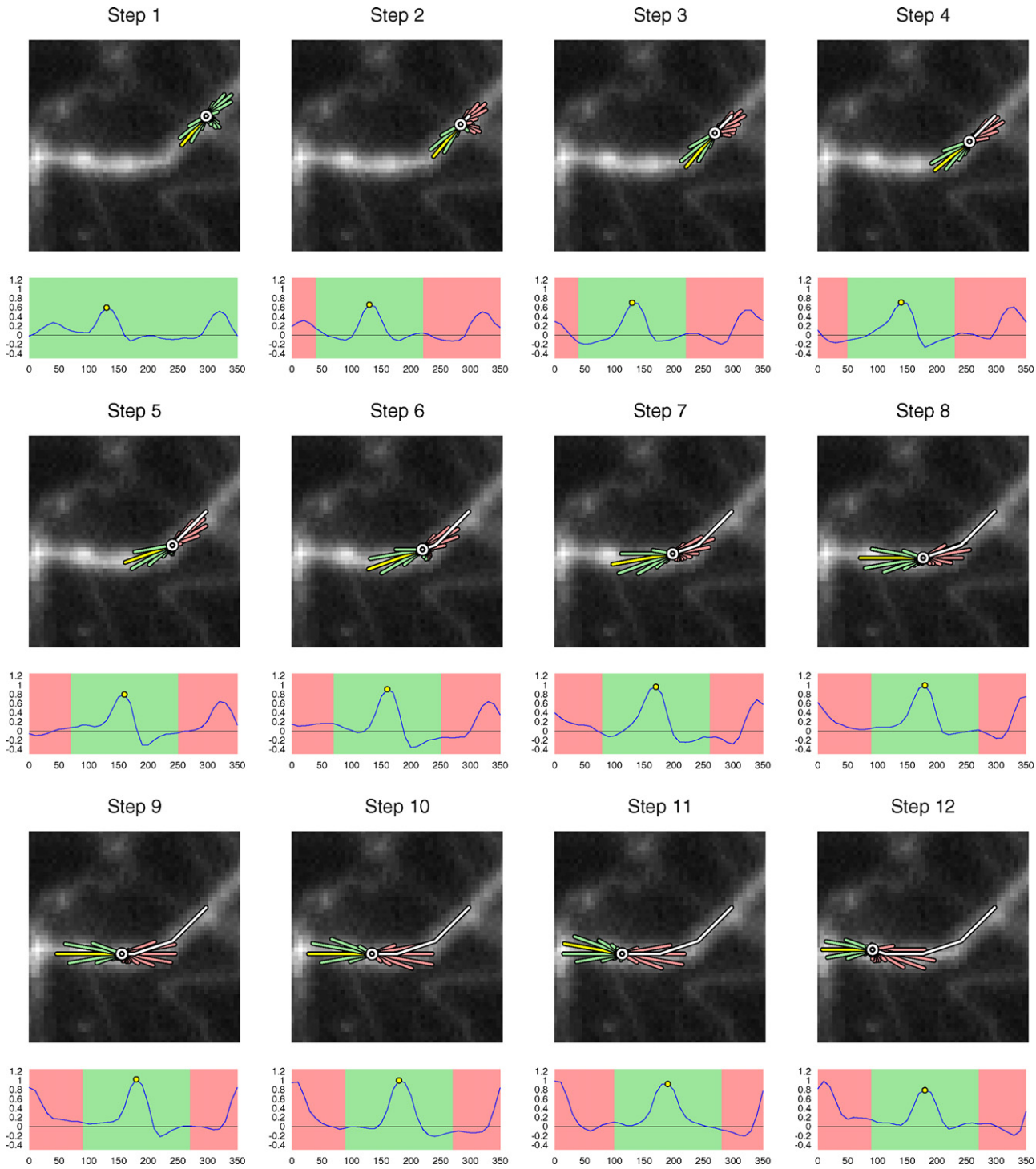

Fig. 8. Tracing via the local exploration method. The first twelve steps are shown for a particular seed point. In each step, the magnitude of the kernel response is indicated both by the length of the radial vectors emanating from the current point in the image and by the plot of the kernel response versus angle below the image. In the images and corresponding plots, available directions and their kernel responses are shown in green, and the chosen direction, corresponding to the largest kernel response, is shown in yellow. Directions that are restricted are indicated in red. For each step, these restricted directions represent a turn of greater than 90° relative to the previous direction. The growing trace path is shown in white, as it follows the bright neurite. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

until a pre-determined number of consecutive intensity violations is reached, or until the neurite reaches an edge or an area that has already been traced. At this point the trace is terminated and the algorithm moves on to the next seed point to repeat the process until all seed points have been investigated.

2.4. Comparison with other tracing methods

We compared the automated tracing method presented in this paper with two other free neurite tracing algorithms, NeuronJ (Meijering et al., 2004) and NeuriteTracer (Pool et al., 2008). These

algorithms were downloaded, installed, and used with the free image analysis program ImageJ (Abramoff et al., 2004).

For comparisons with hand-traced images, we used a data set previously generated and published by our lab (Richardson et al., 2011). Images of neurite outgrowth from explants on materials containing Schwann cell replica topography (Bruder et al., 2007) were manually traced using the straight line segment tool of Openlab v.4.0.4 software. The hand-traced data consisted of neurite segments of variable length associated with an angle measurement. The data was summed and grouped into 10° bins from 0° to 90°. The total length in each bin was used to generate data with $n = 100$ points with the distribution of the binned angle data weighted by the total neurite length at each angle. This data set was compared to the distribution produced by the Neurient code using a chi-square goodness-of-fit test.

2.5. Neuronal culture and image acquisition

In order to develop Neurient, we used images generated in our lab as test images. The procedures used to isolate, culture, stain, and image primary neurons are described below.

2.5.1. Substrate preparation and cell culture

All coating and cell culture reagents were obtained from Invitrogen Life Technologies unless otherwise indicated. Substrates for cell culture were made by replicating silicon templates, casting Sylgard 184 polydimethyl siloxane (PDMS, Dow Corning) with curing agent at a 10:1 wt/wt ratio. Fabrication of materials with anisotropic directive cues is discussed in previous papers from our group (Bruder et al., 2006; Mitchel and Hoffman-Kim, 2011). Substrates used here were flat or contained topography, where the topography consisted of ellipses connected by rectangles (C. López-Fagundo, unpublished results). Topographical features were 1 μm in height. Ellipse major and minor axis dimensions were 36.1 μm and 13.5 μm, respectively, and the rectangles were 105 μm × 1.8 μm. PDMS substrates were plasma activated at 10.5 W for 60 s with a plasma cleaner/sterilizer (PDC - 32 G, Med RF level, Harrick), sterilized by immersion in 70% ethanol, and rinsed with sterile dH_2O. Substrates were coated with 100 mg/mL poly-L-lysine (PLL, 70–150 kDa, Sigma) for one hour, rinsed with dH_2O, coated with 50 μg/mL mouse laminin (LN) in Hanks' Balanced Salt Solution (HBSS) without calcium or magnesium for one hour, and rinsed with dH_2O before plating.

Dissociated dorsal root ganglia (DRG) neurons dissected from the spinal column of postnatal day 0–4 rat pups were used in these experiments. DRG media consisted of Dulbecco's Modified Eagle's Medium (DMEM) supplemented with 10% fetal bovine serum (FBS), 100 U/mL penicillin, 100 μg/mL streptomycin and 50 ng/mL nerve growth factor. DRG explants were cleaned of axons and blood vessels, incubated in 0.25% trypsin–EDTA in HBSS for 45 min at 37 °C, and triturated in DRG media. Dissociated DRG neurons were plated at 57,700 cells/cm^2 and cultured in 12-well plates in DRG media for 5 days.

Cortical neurons were isolated from postnatal day 0–3 rat pups. Cortical media consisted of Neurobasal media supplemented with 2% B27, 100 U/mL penicillin, and 100 μg/mL streptomycin. Cortical regions were incubated in 0.25% trypsin–EDTA in HBSS for 30 min at 37 °C with agitation every 10 min. Trypsin was removed, and cells were resuspended in Neurobasal media supplemented with 10% FBS. Cells were centrifuged at 1200 rpm for 5 min and resuspended in cortical media. Cells were plated on PLL-coated glass coverslips.

2.5.2. Visualization of neurites

Cells were fixed with 2% paraformaldehyde in 0.1 M phosphate buffered saline (PBS) for 20 min. Following fixation and rinsing in PBS, cultures were blocked for nonspecific staining for 1 h at room temperature with 10% goat serum (Jackson Immuno Research Laboratories) and 1% bovine serum albumin (Sigma), in PBS (working solution) with the addition of 0.1% Triton X-100 (VWR) for permeabilization. To visualize the neurites, samples were incubated overnight at 4 °C with monoclonal anti-neuronal class III beta tubulin (β-III-tubulin) primary antibody (Covance Research Products, Inc.) diluted 1:500 in working solution. Samples were rinsed in PBS, incubated for 1 h at room temperature with Cy3-conjugated goat anti-mouse secondary antibody (Jackson Immuno Research Laboratories) diluted 1:200 in working solution and rinsed again in PBS.

Neurite images were obtained at 100× magnification with a 200-ms exposure on a Nikon Eclipse TE2000-S microscope equipped with phase contrast and epifluorescence optics and a software-controlled motorized stage. Corresponding phase and epifluorescence images were captured using a Hamamatsu Orca-ER camera, an Orbit shutter controller, and a Ludl stage controller, and collected with OpenLab v4.0.4 Software (Improvision). Image locations were chosen at random within a user-defined area.

3. Results and validation

3.1. Neurient traces dense neurite images

We have used our algorithm to trace large quantities of fluorescently labeled neurite images, with the end goal of determining the directional response of neurons to materials presenting directive extracellular cues (Kofron et al., 2010). Examples of traced neurite images are shown in Fig. 9. Neurite images traced include confluent primary co-cultures (Fig. 9, top panel). In these experiments it is not typically the goal to correlate specific neurites with a single neuronal soma, and in these images, it is often not possible, and so our algorithm does not distinguish the source of a given neurite segment.

As the traces indicate, brightly stained somata are essentially ignored, and the trace is highly representative of neurites present in the original image (Fig. 9, second panel). The input images are low contrast and complex, and the algorithm was designed to pick out as many of the small features as possible, while also tracing the large features. A portion of each image is magnified and each trace path is shown with a green circle and a red square to indicate the beginning and end, respectively, of that set of traced coordinates (Fig. 9, third panel). Here, a trace path is illustrated as a continuous line colored turquoise, and may be made up of many segments. Each segment is the same length ($s = 3$), while each path is variable in length.

In Fig. 9, the neurite image on the left was taken of DRG neurons cultured on a polymeric substrate with directive topography, a material which produced a highly aligned response (C. López-Fagundo, unpublished results). In contrast, the image on the right was taken of neurons cultured on a flat, uniform surface. The difference in orientation between the neurites on the two materials is reflected in histograms of the angular distribution of the segments of the trace (Fig. 9, bottom panel). For the image of aligned neurons, the corresponding histogram shows a strong directional orientation, with most of the traced segments near the angles of 0°, 180° and 360°, which reflects the neurites growing parallel to the horizontal directive cue presented by the culture platform. In contrast, the histogram corresponding to the trace of neurons with no directional cue is essentially uniform, where no particular orientation is favored.

To further quantify the neurite orientation, we calculated the percent of neurite segments within a given window around a particular angle. For example, 68% of neurite segments are within ±20° of the x-axis in the aligned image (Fig. 9, left), while 26% are oriented in this region in the random image (Fig. 9, right). This tracing

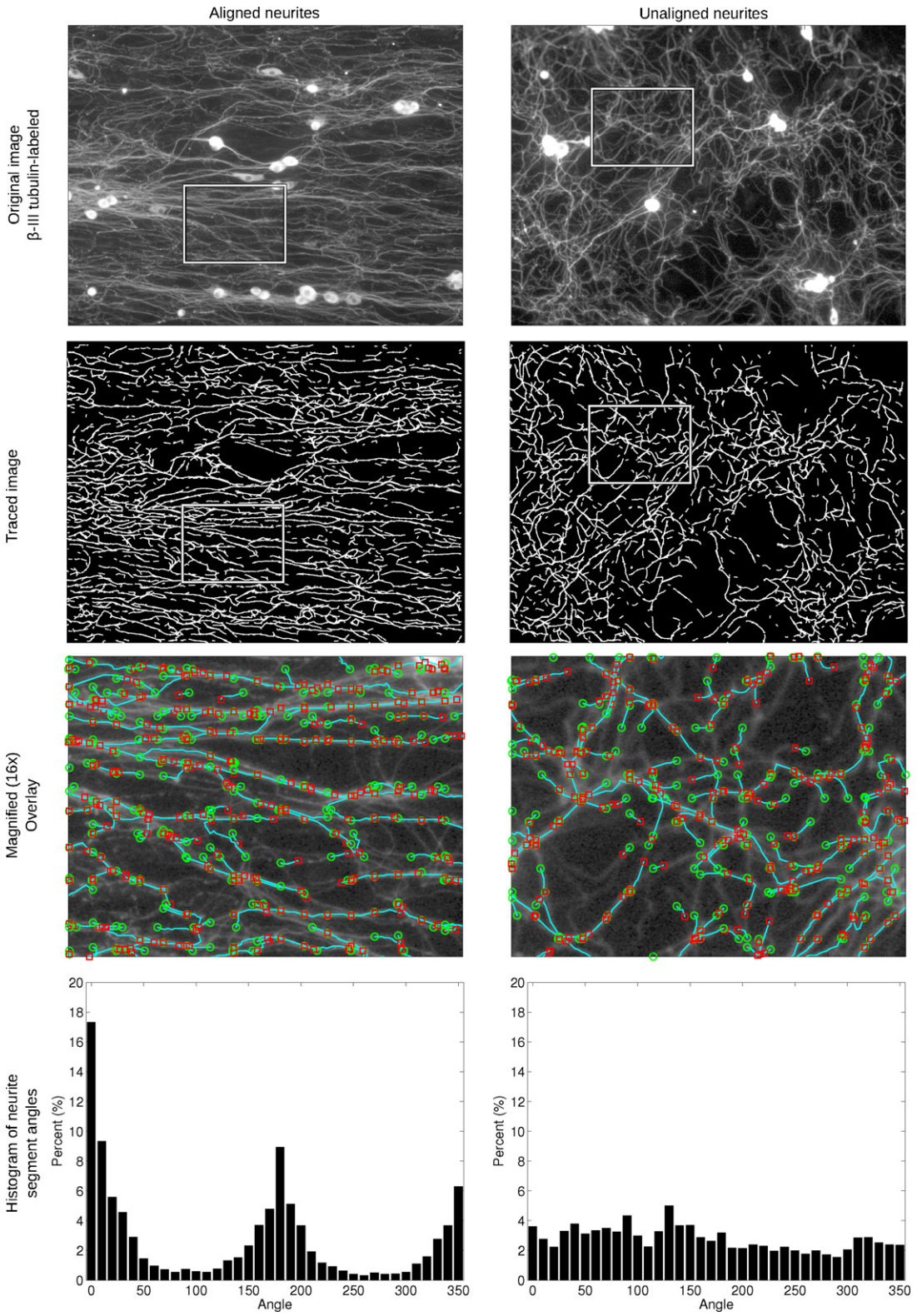

Fig. 9. Example traces of aligned and random images. Aligned (left column) and random (right column) neurites were traced with parameters ($r = 0$, $k = 15$, $f = 4$). The traced images (second row) match up closely with the original images (top row). A magnified view of the boxed areas of each image is shown with the trace paths overlaid, with a green circle marking the seed point and a red square marking the end of each path (third row). Histograms of the angular distribution of trace segments show alignment to the horizontal axis for the aligned neurite image, and a random distribution for the unaligned neurite image (bottom row). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Table 1
Parameters used in the Neurient code. These parameters were either obtained directly from the input image, specified by the user to optimize the tracing results, or derived in the code from other parameters.

Symbol	Typical value(s)	Used in	Interpretation
Input parameters			
L, W	512–1344	Phase I, II, III	Image length and width, in pixels
γ	–	Phase II	Median pixel brightness across entire image
σ	–	Phase II	Standard deviation of pixel brightness around the median
User-specified parameters			
N_θ	32, 36	Phase I	Number of angles used to generate the KRLT
k	6–15	Phase I	Kernel length; the number of times the 1D kernel $[-2\ \ -1\ \ 0\ \ 1\ \ 2\ \ \ldots\ \ 2\ \ 1\ \ 0\ \ -1\ \ -2]^T$ is repeated to generate the 2D kernel
r	0–2	Phase I	Related to the thickness of a typical neurite, where the optimal neurite thickness t is $t = 6 + 2r$
g	20	Phase II	Grid spacing used to determine locations of potential seed points
f	2–4	Phase II	Used in determining the neurite threshold Γ
s	3	Phase II	The distance the trace steps forward at each step during exploratory tracing
Derived parameters			
M	–	Phase II	Number of horizontal grid lines: $M = \lfloor W/g \rfloor$
N	–	Phase II	Number of vertical grid lines: $N = \lfloor L/g \rfloor$
Γ	–	Phase II, III	Neurite intensity threshold: $\Gamma = \gamma + \sigma/f$

method therefore allows us to quantify and compare the alignment of neurons cultured on anisotropic and isotropic materials (Kofron et al., 2010).

3.2. Parameter selection

There are a number of parameters used in the proposed algorithm, which are derived from the input image, specified by the user, or calculated from other parameters. The numerical values of these parameters can affect the performance of the tracing algorithm, and can be altered for different types of images. A summary of the parameters used in this project, including typical values used in our analysis, is presented in Table 1. Parameters determined from the input image include the image dimensions, L and W, the median pixel intensity across the entire image, γ, and the standard deviation of pixel intensity around the median, σ.

User specified parameters include those used when generating the KRLT, finding the initial seed points, and performing the exploratory tracing step. Parameters N_θ, r, and k are used to generate the KRLT. Because our algorithm cross-correlates a horizontal kernel with rotated images, rather than using a pre-determined set of rotated kernels on the original image as other researchers have done (Al-Kofahi et al., 2003), we can specify an arbitrary number of angles, N_θ. For the results shown here, we have used $N_\theta = 36$, which yields a difference between adjacent angles of $\Delta\theta = 10^\circ$.

As described in Section 2.1.3 and Fig. 5, the parameter k affects the kernel's maneuverability. The kernel length k determines the ability of the tracing step to successfully turn corners and reject imperfections in the image. A larger value of k results in a trace path more accurately choosing the orientation of the next segment in the case where there is a locally straight structure of a length similar to k, even if the structure has a low signal to noise ratio or is discontinuous. On the other hand, a smaller value of k will allow the trace to follow a tightly curved structure, but will also result in more sensitivity to discontinuities and noise. For the images presented here, we set $k = 15$. In previously published algorithms, k has been set at 6 (Zhang et al., 2007b), or allowed to vary from 8 to 50 (Al-Kofahi et al., 2002).

The last parameter used to generate the kernel response lookup table is r, a parameter based on neurite thickness. As illustrated in Fig. 5, there are $2r + 1$ zeros in the center of each column of the kernel; these zeros will correspond to the center of the neurite, and therefore this part of the neurite does not contribute to the kernel response. Neurites of thickness t, where t is related to r through the formula $r = (t - 6)/2$, are the optimal width to be detected by a kernel with a given r. In our images, we had many thin neurites, at times only 1–2 pixels in width, that were successfully traced with $r = 0$, which corresponds to an optimal thickness of $t = 6$ pixels. This result indicates the flexibility of the kernel, and its ability to trace neurites of varying thickness within a single image.

The user-specified parameter g defines the spacing between the grid lines used during the search for seed points. The numbers of horizontal and vertical grid lines, M and N, respectively, are determined based on g and the image dimensions using the formulas $M = \lfloor W/g \rfloor$ and $N = \lfloor L/g \rfloor$. Here we set $g = 20$ as described previously (Al-Kofahi et al., 2002). The derived parameter Γ is used both to filter the seed points and as a stopping criterion during the tracing phase. Γ is the neurite threshold, and is dependent on image intensity, according to the formula $\Gamma = \gamma + \sigma/f$, where f is a user-specified parameter, usually set to 2 or 4 (Al-Kofahi et al., 2003).

Last, the parameter s sets the length of each segment in the tracing phase. This is the distance a trace is stepped forward after the direction of the highest kernel response is determined for a given point, and therefore should never be larger than k, the length of the kernel.

3.3. Trace results for cortical neurons

Though our code was initially developed for the application of high density cultures of dissociated DRG, it can be used to trace images and obtain the orientation data for any cell type exhibiting neurite-like structures. In Fig. 10, we show the traces of cortical neurons cultured on flat materials. Relative to high density, neurite-rich DRG cultures, these cortical neuron images exhibit shorter and sparser neurites. Despite these morphological differences between cortical and DRG neurons, the Neurient code accurately traces images of cortical neurons, demonstrating the algorithm's flexibility.

3.4. Comparison with other tracing methods

Different trace algorithms are optimal for different types of images, where factors affecting trace performance include image quality, cell type, and the presence of multiple cell types. Here we compare our results with two previously published methods to validate our approach: NeuronJ (Meijering et al., 2004) and NeuriteTracer (Pool et al., 2008). For both algorithms, the code to process images was readily available online and easy to use. Both algorithms have been implemented and released for ImageJ. We used a number of other neurite tracing methods in addition, but they

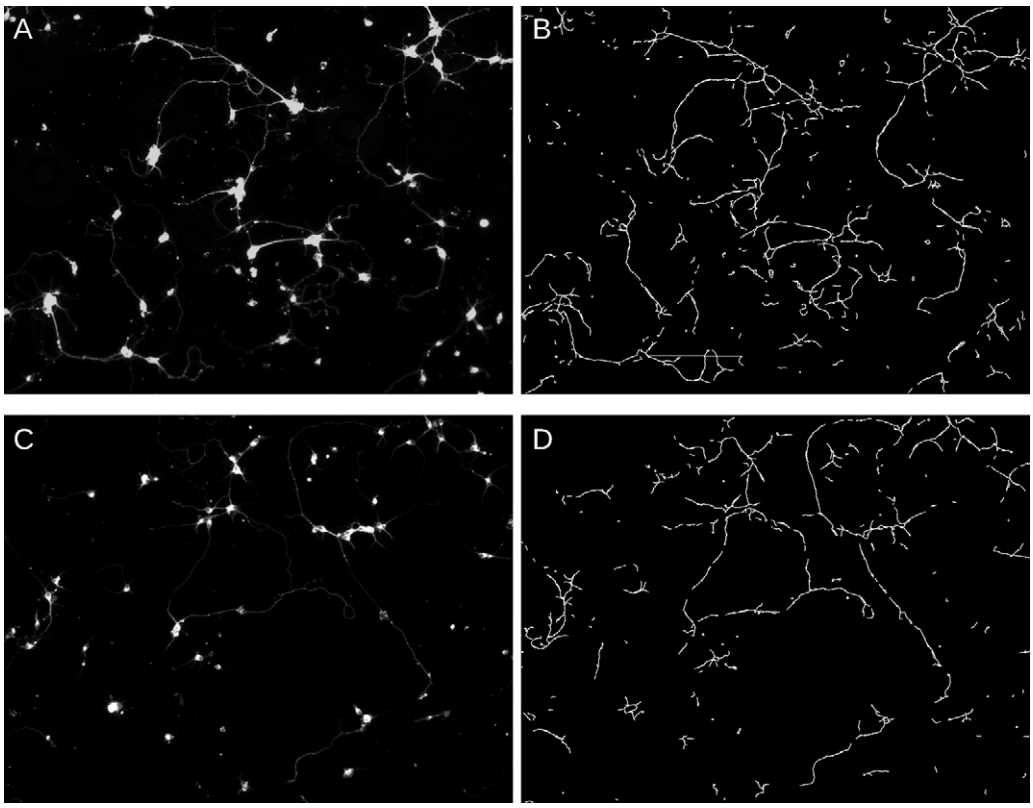

Fig. 10. Example traces of cortical neurons. (A) and (C) Cortical neurons labeled for β-III tubulin. (B) and (D) Corresponding traced cortical neuron images.

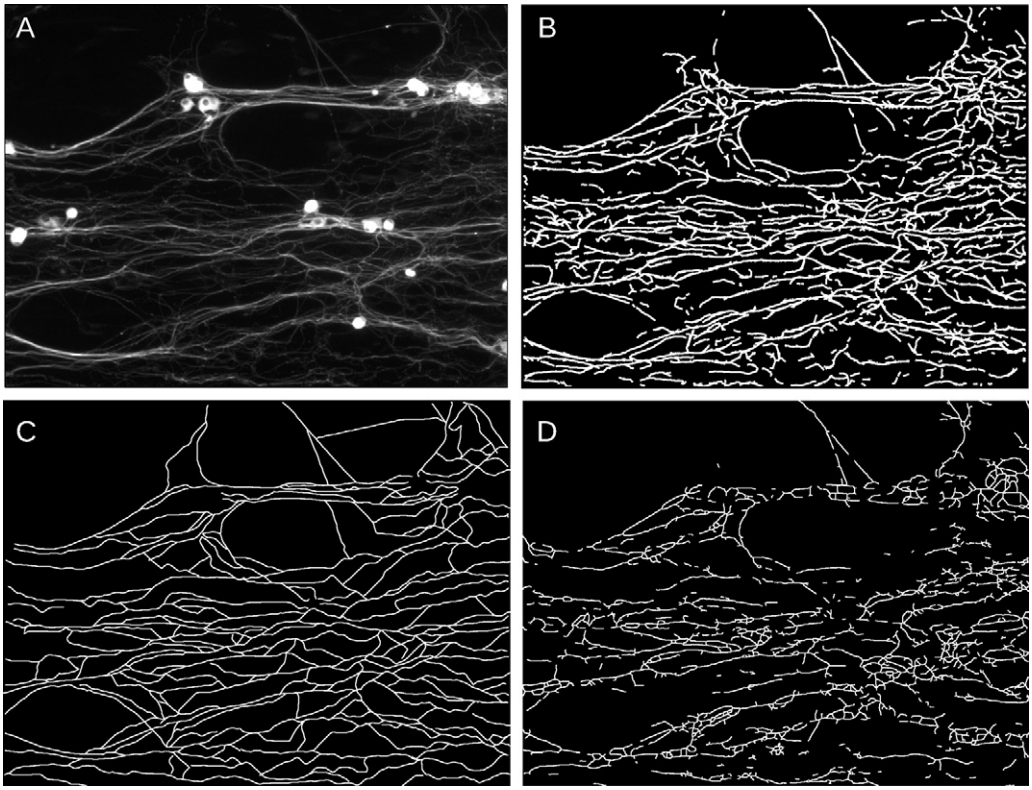

Fig. 11. Comparison of neurite tracing methods. (A) Original β-III tubulin stained DRG neurite image. (B) Trace results from Neurient with standard parameters ($k = 15$, $r = 0$). (C) Trace results using NeuronJ, a semi-manual plugin for ImageJ. (D) Trace results using NeuriteTracer, an automated plugin for ImageJ.

were not viable options for our images. An aligned, high density DRG image is shown in Fig. 11A, and traces from Neurient, NeuronJ, and NeuriteTracer are shown in Fig. 11B–D, respectively.

NeuronJ is a semi-manual tracing tool which requires the user to specify initial and final points along chosen neurites, with the option of including immediate points to anchor the trace to the neurite. The plugin is easy to use and allows the user to have a large degree of control over the trace. However, this tracing software was not ideal for the high density images used in this study. In particular, many intermediate anchor points were required to keep a given trace along a neurite, with the end result that the semi-manual tracing was nearly as time consuming as fully manual tracing. The result of using NeuronJ to trace the image in Fig. 11A is shown in Fig. 11C.

NeuriteTracer was able to trace many of the structures from the original image, as shown in Fig. 11D. This algorithm was developed for low density neuronal cultures and works best when few non-neuronal cells are present (Pool et al., 2008), and is therefore not ideal to our application. In particular, NeuriteTracer requires DAPI-stained images of cell nuclei, thresholded areas of which are then removed from the neuronal image. For high density mixed-cell cultures, DAPI-stained nuclei corresponding to non-neuronal cells may occupy a relatively large portion of the image, resulting in a correspondingly large reduction in the image area which is available for tracing. Further, NeuriteTracer uses a global method including binarization and skeletonization steps, and therefore excludes many low-signal neurites and appears to not follow some of the long, continuous neurites. This may result from the lack of an edge detecting kernel, which can help an exploratory tracing algorithm to follow continuous structures even with intersections and breaks (Al-Kofahi et al., 2003).

In addition to a comparison with available open source tracing methods from the literature, we compared the orientation data from a previously published work (Richardson et al., 2011) with the data generated by the Neurient code. We found that the angular distributions produced by these two methods were not statistically different for 24 images corresponding to four distinct experimental conditions (chi-square, $p > 0.49$).

4. Discussion and conclusion

The neurite tracing algorithm described here addresses a specific unmet need in the neuroscience and biomedical engineering communities, which is to extract quantifiable neurite orientation data from images of neurons cultured under isotropic or directive experimental conditions. A more general need in these fields is for researchers to have access to analysis tools which are user-friendly and open source (von Hippel, 2001; Erickson et al., 2005; Chesler and Baker, 2010). Though a range of neurite analysis and tracing programs have been proposed in the literature, many of these are not available for use by the community, require extensive programming knowledge to use or replicate, or have closed-source code, which means that researchers cannot tailor the code to their own specific needs. Of the open source neurite tracing algorithms available (tabulated in Ho et al., 2011) none were developed to generate orientation data for complex neurite images.

In addition to addressing the needs described above, the method presented here has a number of advantages for our application over previously published methods. In particular, our focus on obtaining orientation data from high-density images is unique within the literature. Many published neuron tracing algorithms seek to evaluate neuronal morphology, and therefore may track neurites emanating from a single cell, focusing on characterizing parameters such as neurite length and branching morphology (Ho et al., 2011; Helmstaedter et al., 2011; Leach et al., 2011). For high density, longer term cultures, this sort of morphological delineation is impossible for both human users and computer programs to implement. Despite the high degree of complexity of these types of images, a large amount of quantitative information can be obtained from them, as demonstrated here. Previous studies have indicated the importance of neuronal density in affecting outgrowth characteristics (Bruder et al., 2007; Serra et al., 2010; Cullen et al., 2010). As many tracing algorithms are developed with other applications in mind, notably morphological mapping (Meijering, 2010), extraction of orientation data is rarely an option in neuronal tracing. This was our main motivation in developing Neurient.

Another advantage of our method is that it is entirely automatic, and no manual pre-processing is required. Though users can tune their results by changing parameters such as g, f, and k based on neurite density, image quality, and neurite continuity, respectively, this is not required to obtain initial quantitative data. MATLAB is widely used at academic institutions, and many science and engineering students learn it during their coursework. We believe that writing our code in the MATLAB scripting language will make it accessible to a range of researchers. For those without access to MATLAB, our code can be used with Octave, a free and open source language which is compatible with MATLAB. Lastly, the automated nature of our code removes potential for user-bias, which may be present with any manual or semi-manual tracing method.

Lastly, our method places no restrictions on the angles of interest or the angular resolution. Many previous methods have relied on finite sets of predetermined rotated kernels. This may prove problematic when analysis could benefit from using a different set of angles, for example if the region of interest is a narrow, high-resolution window around a given axis. With our method, the kernel response lookup table is arbitrary in size, and researchers can tailor their angular bins to their specific applications and needs.

The tracing algorithm described here does have some limitations. It is computationally slow, requiring approximately 1.5–3 min, depending on image complexity, for a 1024×1344 image on a Dell Precision T1500 machine with a quad-core 2.8 GHz CPU and 8 GB RAM, running Windows 7. Our method has a global component (the generation of the kernel response lookup table), and like all global methods requires processing across the entire image rather than concentrating solely on areas of interest. However, given the speed of modern computers, the rate of increase of computational power, and the automated nature of the method, we believe that this limitation is relatively minor. Lastly, while Neurient was not designed to track morphological information, it could be modified to do so, or used in conjunction with other software designed to extract such information from the traced paths that it generates.

Our implementation of this tracing algorithm is open source and freely available to the scientific community to use and modify. We have demonstrated here that the Neurient code produces accurately traced images from complex and high density neurite images, and that the traced coordinates correspond to quantifiable neurite orientation data. The orientation data has been successfully used to quantify the difference in alignment between neurons on unpatterned and patterned materials. Though we have used Neurient primarily with dissociated DRG neuron cultures, we have also demonstrated that it can be used with other cell types exhibiting extensions. Neurient compares favorably with other open source, freely available methods for automated neurite tracing. To our knowledge, this is the only tool available at the present time that was designed to analyze such complex, high density images, and that also produces usable orientation data from the input images.

Acknowledgements

The authors would like to thank Liane Livi for cortical neuron images. Funding provided by GAANN Award P200A090076,

administered by the Brown University Institute for Molecular and Nanoscale Innovation, to J.M.; NSF Career 0547060, NSF CBET 1134166, NIH R01 EB005622-01 to D.H.K.

References

Abramoff M, Magalhaes P, Ram S. Image processing with imagej. Biophotonics Int 2004];11:36–42.

Al-Kofahi KA, Can A, Lasek S, Szarowski DH, Dowell-Mesfin N, Shain W, Turner JN, Roysam B. Median-based robust algorithms for tracing neurons from noisy confocal microscope images. IEEE Trans Inf Technol Biomed 2003];7(4):302–17.

Al-Kofahi KA, Lasek S, Szarowski DH, Pace CJ, Nagy G, Turner JN, Roysam B. Rapid automated three-dimensional tracing of neurons from confocal image stacks. IEEE Trans Inf Technol Biomed 2002];6(2):171–87.

Ascoli GA. Neuroinformatics grand challenges. Neuroinformatics 2008];6(1):1–3.

Bruder JM, Lee AP, Hoffman-Kim D. Biomimetic materials replicating Schwann cell topography enhance neuronal adhesion and neurite alignment in vitro. J Biomater Sci Polym Ed 2007];18(8):967–82.

Bruder JM, Monu NC, Harrison MW, Hoffman-Kim D. Fabrication of polymeric replicas of cell surfaces with nanoscale resolution. Langmuir 2006];22(20):8266–70.

Busch SA, Silver J. The role of extracellular matrix in CNS regeneration. Curr Opin Neurobiol 2007];17(1):120–7.

Can A, Shen H, Turner JN, Tanenbaum HL, Roysam B. Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. IEEE Trans Inf Technol Biomed 1999];3(2):125–38.

Casasent D, Psaltis D. Position, rotation, and scale invariant optical correlation. Appl Opt 1976];15:1795–9.

Chen ZL, Yu WM, Strickland S. Peripheral regeneration. Annu Rev Neurosci 2007];30:209–33.

Chesler EJ, Baker EJ. The importance of open-source integrative genomics to drug discovery. Curr Opin Drug Discov Devel 2010];13(3):310–6.

Corey JM, Lin DY, Mycek KB, Chen Q, Samuel S, Feldman EL, Martin DC. Aligned electrospun nanofibers specify the direction of dorsal root ganglia neurite growth. J Biomed Mater Res A 2007];83(3):636–45.

Cullen DK, Gilroy ME, Irons HR, Laplaca MC. Synapse-to-neuron ratio is inversely related to neuronal density in mature neuronal cultures. Brain Res 2010];1359:44–55.

Dobkin BH, Havton LA. Basic advances and new avenues in therapy of spinal cord injury. Annu Rev Med 2004];55:255–82.

Dowell-Mesfin NM, Abdul-Karim MA, Turner AM, Schanz S, Craighead HG, Roysam B, Turner JN, Shain W. Topographically modified surfaces affect orientation and growth of hippocampal neurons. J Neural Eng 2004];1(2):78–90.

Erickson BJ, Langer S, Nagy P. The role of open-source software in innovation and standardization in radiology. J Am Coll Radiol 2005];2(11):927–31.

Foley JD, Grunwald EW, Nealey PF, Murphy CJ. Cooperative modulation of neuritogenesis by PC12 cells by topography and nerve growth factor. Biomaterials 2005];26(17):3639–44.

Fournier AE, Takizawa BT, Strittmatter SM. Rho kinase inhibition enhances axonal regeneration in the injured CNS. J Neurosci 2003];23(4):1416–23.

Geller HM, Fawcett JW. Building a bridge: engineering spinal cord repair. Exp Neurol 2002];174(2):125–36.

Gertz CC, Leach MK, Birrell LK, Martin DC, Feldman EL, Corey JM. Accelerated neuritogenesis and maturation of primary spinal motor neurons in response to nanofibers. Dev Neurobiol 2010];70(8):589–603.

Gomez N, Lu Y, Chen S, Schmidt CE. Immobilized nerve growth factor and microtopography have distinct effects on polarization versus axon elongation in hippocampal cells in culture. Biomaterials 2007];28(2):271–84.

Hanson JN, Motala MJ, Heien ML, Gillette M, Sweedler J, Nuzzo RG. Textural guidance cues for controlling process outgrowth of mammalian neurons. Lab Chip 2009];9(1):122–31.

Haq F, Venkatramani A, Keith C, Zhang G. Neurite development in PC12 cells cultured on nanopillars and nanopores with sizes comparable with filopodia. Int J Nanomed 2007];2(1):107–15.

Helmstaedter M, Briggman KL, Denk W. High-accuracy neurite reconstruction for high-throughput neuroanatomy. Nat Neurosci 2011];14(8):1081–8.

von Hippel E. Innovation by user communities: Learning from open-source software. MIT Sloan Manage Rev 2001];42:82–6.

Ho SY, Chao CY, Huang HL, Chiu TW, Charoenkwan P, Hwang E. Neurphologyj: an automatic neuronal morphology quantification method and its application in pharmacological discovery. BMC Bioinformatics 2011];12:230.

Hoffman-Kim D, Mitchel JA, Bellamkonda RV. Topography, cell response, and nerve regeneration. Annu Rev Biomed Eng 2010];12:203–31.

Houchin-Ray T, Swift LA, Jang JH, Shea LD. Patterned PLG substrates for localized DNA delivery and directed neurite extension. Biomaterials 2007];28(16):2603–11.

Huang Y, Zhou X, Miao B, Lipinski M, Zhang Y, Li F, Degterev A, Yuan J, Hu G, Wong STC. A computational framework for studying neuron morphology from in vitro high content neuron-based screening. J Neurosci Methods 2010];190(2):299–309.

Ichihara S, Inada Y, Nakamura T. Artificial nerve tubes and their application for repair of peripheral nerve injury: an update of current concepts. Injury 2008];39(Suppl 4):29–39.

Johansson F, Carlberg P, Danielsen N, Montelius L, Kanje M. Axonal outgrowth on nano-imprinted patterns. Biomaterials 2006];27(8):1251–8.

Kim YT, Haftel VK, Kumar S, Bellamkonda RV. The role of aligned polymer fiber-based constructs in the bridging of long peripheral nerve gaps. Biomaterials 2008];29(21):3117–27.

Kofron CM, Hoffman-Kim D. Optimization by response surface methodology of confluent and aligned cellular monolayers for nerve guidance. Cell Mol Bioeng 2009];2(4):554–72.

Kofron CM, Liu YT, Lopez-Fagundo CY, Mitchel JA, Hoffman-Kim D. Neurite outgrowth at the biomimetic interface. Ann Biomed Eng 2010];38(6):2210–25.

Koh HS, Yong T, Chan CK, Ramakrishna S. Enhancement of neurite outgrowth using nano-structured scaffolds coupled with laminin. Biomaterials 2008];29(26):3574–82.

Larkin K. Fast fourier method for the accurate rotation of sampled images. Opt Commun 1997];139:99–106.

Leach MK, Naim YI, Feng ZQ, Gertz CC, Corey JM. Stages of neuronal morphological development in vitro – an automated assay. J Neurosci Methods 2011];199(2):192–8.

Leandro JJG, Cesar Jr RM, da F Costa L. Automatic contour extraction from 2d neuron images. J Neurosci Methods 2009];177(2):497–509.

Li GN, Hoffman-Kim D. Tissue-engineered platforms of axon guidance. Tissue Eng Part B: Rev 2008];14(1):33–51.

Mahoney MJ, Chen RR, Tan J, Saltzman WM. The influence of microchannels on neurite growth and architecture. Biomaterials 2005];26(7):771–8.

Meijering E. Neuron tracing in perspective. Cytometry A 2010];77(7):693–704.

Meijering E, Jacob M, Sarria JCF, Steiner P, Hirling H, Unser M. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. Cytometry A 2004];58(2):167–76.

Miller C, Jeftinija S, Mallapragada S. Micropatterned Schwann cell-seeded biodegradable polymer substrates significantly enhance neurite alignment and outgrowth. Tissue Eng 2001];7(6):705–15.

Miller C, Jeftinija S, Mallapragada S. Synergistic effects of physical and chemical guidance cues on neurite alignment and outgrowth on biodegradable polymer substrates. Tissue Eng 2002];8(3):367–78.

Mitchel JA, Hoffman-Kim D. Cellular scale anisotropic topography guides schwann cell motility. PLoS One 2011];6(9):e24316.

Owen CB, Makedon F. High quality alias free image rotation. Technical report PCS-TR96-301. Hanover, NH: Dartmouth College, Computer Science; 1996.

Parker JA, Kenyon RV, Troxel DE. Comparison of interpolating methods for image resampling. IEEE Trans Med Imaging 1983];2(1):31–9.

Pool M, Thiemann J, Bar-Or A, Fournier AE. Neuritetracer: a novel imagej plugin for automated quantification of neurite outgrowth. J Neurosci Methods 2008];168(1):134–9.

Richardson JA, Rementer CW, Bruder JM, Hoffman-Kim D. Guidance of dorsal root ganglion neurites and schwann cells by isolated schwann cell topography on poly(dimethyl siloxane) conduits and films. J Neural Eng 2011];8(4):046015.

Rittscher J. Characterization of biological processes through automated image analysis. Annu Rev Biomed Eng 2010];12:315–44.

Rodriguez A, Ehlenberger DB, Hof PR, Wearne SL. Three-dimensional neuron tracing by voxel scooping. J Neurosci Methods 2009];184(1):169–75.

Schlosshauer B, Dreesmann L, Schaller HE, Sinis N. Synthetic nerve guide implants in humans: a comprehensive survey. Neurosurgery 2006];59(4):740–7, discussion 747–8.

Schmidt CE, Leach JB. Neural tissue engineering: strategies for repair and regeneration. Annu Rev Biomed Eng 2003];5:293–347.

Serra M, Guaraldi M, Shea TB. Inhibitory neurons modulate spontaneous signaling in cultured cortical neurons: density-dependent regulation of excitatory neuronal signaling. Phys Biol 2010];7(2):026009.

Walsh JF, Manwaring ME, Tresco PA. Directional neurite outgrowth is enhanced by engineered meningeal cell-coated substrates. Tissue Eng 2005];11(7/8):1085–94.

Wu C, Schulte J, Sepp KJ, Littleton JT, Hong P. Automatic robust neurite detection and morphological analysis of neuronal cell cultures in high-content screening. Neuroinformatics 2010];8(2):83–100.

Zhang Y, Zhou X, Degterev A, Lipinski M, Adjeroh D, Yuan J, Wong STC. Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays. Neuroimage 2007a];35(4):1502–15.

Zhang Y, Zhou X, Degterev A, Lipinski M, Adjeroh D, Yuan J, Wong STC. A novel tracing algorithm for high throughput imaging screening of neuron-based assays. J Neurosci Methods 2007b];160(1):149–62.