

J Neurosci Methods. Author manuscript; available in PMC 2011 October 2.

Published in final edited form as:

J Neurosci Methods. 2010 July 15; 190(2): 299–309. doi:10.1016/j.jneumeth.2010.05.012.

A computational framework for studying neuron morphology from *in vitro* high content neuron-based screening

Yue Huang^{a,b}, Xiaobo Zhou^{a,*}, Benchun Miao^c, Marta Lipinski^d, Yong Zhang^a, Fuhai Li^a, Alexei Degterev^c, Junying Yuan^d, Guangshu Hu^b, and Stephen T.C. Wong^a

^aMethodist Hospital Research Institute, Radiology Department, 6565 Fannin Street, B5-022, Houston, TX 77030, United States

^bBiomedical Engineering Department, Medical School, Tsinghua University, Beijing 100084, People's Republic of China

^cDepartment of Biochemistry, Tufts University School of Medicine, Boston, MA 02111, United States

^dDepartment of Cell Biology, Harvard Medical School, Boston, MA 02115, United States

Abstract

High content neuron image processing is considered as an important method for quantitative neurobiological studies. The main goal of analysis in this paper is to provide automatic image processing approaches to process neuron images for studying neuron mechanism in high content screening. In the nuclei channel, all nuclei are segmented and detected by applying the gradient vector field based watershed. Then the neuronal nuclei are selected based on the soma region detected in neurite channel. In neurite images, we propose a novel neurite centerline extraction approach using the improved line-pixel detection technique. The proposed neurite tracing method can detect the curvilinear structure more accurately compared with the current existing methods. An interface called NeuriteIQ based on the proposed algorithms is developed finally for better application in high content screening.

Keywords

High content screening; Microscopy image; Nuclei segmentation; Neurite outgrowth; Line-pixel detection; Branch area

1. Introduction

With the use of advanced high resolution fluorescence microscopy imaging techniques, high content screening (HCS) has made it possible to study the intricate nervous system, and discover candidates for drug targets (Zhou and Wong, 2006). However, HCS generates volumes of microscopy images, each of which contains millions of pixels. Thus, instead of manual analysis which is tedious and time consuming, fully automated methods are required to extract and analyze phenotypic change information in large amounts of microscopy image data. The main goal of this paper is to propose an image processing system to analyze neuron images for studying neuron mechanism. This comprehensive system is developed to analyze two types of neuron images viewed from in vitro microscopy in high content

^{© 2010} Elsevier B.V. All rights reserved.

^{*}Corresponding author. Tel.: +1 617 525 6242., XZhou@tmhs.org, zhou@crystal.harvard.edu (X. Zhou).

screening experiments. This system includes (1) segmenting and counting neuronal nuclei in nuclei image; (2) labeling neurite in neurite image and (3) calculating some neuron morphology features.

There have been a number of nuclei segmentation methods reported in the literature, including thresholding, edge based methods, watershed algorithm, active contours, and other pattern analysis algorithms (Chen et al., 2007; Ge and Parvin, 1999; Li et al., 2005; Nandy et al., 2007). While these methods are popular and effective, they suffer from certain limitations. Thresholding cannot effectively deal with clustering nuclei (Li et al., 2007b). Edge based methods always leads to misidentification of noisy edges and discontinuous boundaries (Lin et al., 2003). Active contours require the contours' initialization, which can be a challenging work (Li et al., 2007a). The performance of morphological approaches is not good in nuclei detection if the nuclei vary significantly in size or shape. Watershedbased methods are widely used in nuclei segmentation, especially when clustered nuclei are present (Malpica et al., 1997). However, classical watershed often results in oversegmentation. In this paper, a modified watershed algorithm is proposed to address nuclei detection and segmentation.

Neurites can be considered as bright elongated line-like structures surrounded by a dark background. Therefore, the problem of labeling neurites is equivalent to that of detecting curvilinear structures in microscopy images. Many efforts have been devoted to detect curvilinear structures in images. Some researchers proposed a tracing technique called exploratory algorithm or vector tracking, which starts with detecting a set of seed points followed by tracing the centerlines from these initial points recursively until certain predefined stop conditions are satisfied (Al-Kofahi et al., 2002; Zhang et al., 2007a). This method is computationally efficient since it only processes the pixels near the centerline. Zhang et al. (Sofka and Stewart, 2006; Zhang et al., 2007b) used this tracing technique to find the starting points and end points first in each neurite, then a dynamic programming approaches is applied to link each pair of points. Ali et al. (Ali et al., 1999) proposed a multi-scale matched filter to define the tracing direction and calculate 'vesselness' of each traced pixel using a 6-dimensional measurement.

A line-pixel detection algorithm has been proposed in several works (Steger, 1998; Xiong et al., 2006; Zhang et al., 2007b). This method uses a model to find the local geometric properties of the lines, and examines each pixel followed by a linking process which connects the detected centerline points into connected center-lines. Meijering (Meijering et al., 2004) proposed a semi-automated method that links consecutive ridge pixels derived from the line-pixel detection algorithm by selecting starting and ending points manually and calculating the global minimum cumulative cost function. Xiong et al. (Xiong et al., 2006) also detected the branch points and end points of neurites based on line-pixel method. The performance of these methods is also determined by user specified parameters selection such as the maximum of neurite width and a threshold for the strength of line.

However, the aforementioned algorithms discussed above cannot directly extract the centerlines in branched areas. In most tracing methods, each pixel has only one direction because the direction of each point in the centerline is defined as the same as its boundary whose response is largest with the template across directions (Al-Kofahi et al., 2002). Most line-pixel detection algorithms use the Hessian matrix to determine the normal direction of each pixel. However in bifurcation areas, the orientation is unclear. Xiong et al. in (Xiong et al., 2006) used a circle with constant radius to find the centerline at each end point, and then branch points could be easily detected. However, some branch points will be missed because it is hard to select an appropriate radius for all conditions. Al-Kofahi et al. presented a method which is formulated as a generalized likelihood for branch detection by checking

each ending point after the tracing phase (Al-Kofahi et al., 2007). Tsai et al. (2004) described an approach a so-called exclusion region and position refinement to improve the accuracy of estimating the location of branch structure at the end of tracing process. Most of these methods cannot detect the branch points and the single line simultaneously, but only consider it as a post-processing step. In this paper, we shall propose a modified line-pixel method to detect neurite branch and single line simultaneously.

The rest of the paper is organized as follows. Section 2 describes the animal model in material part and illustrates our approach step by step. Validation is presented in Section 3. Section 4 presents the development and application of interface NeuriteIQ and we conclude the paper in Section 5.

2. Materials and methods

In this section, we describe neuron images acquisition and the proposed method for neuron image processing and analysis.

2.1. Neuron images acquisition

Normal E15 C57BL/6 mouse in the cortical hemisphere, including the hippocampus, are obtained from the Department of Biochemistry at Tufts University. Neurons are derived from C57BI/6 mouse using a standard protocol: E15 C57BI/6 pregnant females are euthanized by overdose of isoflurane, cortical hemispheres are dissected from embryos in HBSS- (Ca²⁺ and Mg²⁺ free) and treated with 0.1% trypsin for 10° at RT. After washing with HBSS-, they are triturated with fire-polished Pasteur pipettes in 0.025% DNAse I, 12 mM MgSO4, spun down 5 min at 900 rpm and washed in HBSS-. Another 5 min spin at 900 rpm the neuronal pellet will be resuspend in DMEM supplemented with 10% fetal calf serum, penicillin/streptomycin and glutamine to 0.2-1X106/mL and plated in poly-D-lysine coated plates. The cells will be incubated at 37 °C for 24 h, where the media will be changed to Neurobasal with B27 supplement, penicillin/streptomycin and glutamine, and the neurons will be incubated at 37 °C for additional 4 days to allow full neuronal differentiation and neurite extension. Then, neurons were blocked in 0.4% Triton X-100 buffer with 10% normal donkey serum for 30 min at RT, followed by incubation with neurite stain TUJ1 (Type III neuronal specific tubulin beta) antibody at a 1:250 dilution in 0.1% Triton solution overnight at 4 °C. After two more washes with PBS, neurons were incubated with nuclear stain Sytox Green (Molecular Probes, 1:5000 dilution) for 30 min at 37 C. Laser confocal fluorescence microscopy is used for capturing neuron images. To acquire different subcellular structures, e.g. neurite and nuclei, green (excitation 480 nm/emission 535 nm) and red (560 nm/630 nm) fluorescent filters were used to control contrast in the final image captured with a CCD (charge coupled device) digital camera system. The quality of these standard wide field microscopy images is good if the images are not captured out of focus. Finally, two types of *in vitro* data, nuclei images and neurite images, were generated, as shown in Fig. 1. In Figs. 1, 3, 5, 6 and 7, the number in x, y axes shows the size of the image with the unit 'pixel'. The image resolution is 1.23 μm/pixel. In the following, we search for an efficient method to process and analyze these images.

2.2. Processing in nuclei images

In the nuclei channel, our goal is to detect and segment all the nuclei. Nuclei are round, bright blots of different intensities and sizes. Nuclei detection is the basis of segmentation, because it provides the 'seeds' information, e.g. the 'seeds' in seeded-watershed. Fig. 2 shows the flowchart of the proposed nuclei detection and segmentation method.

In a preprocessing stage, a data-driven background correction algorithm (Lindeberg, 1998) is applied to correct the degeneration of the images, which uses the cubic B-Spline method to estimate the background iteratively.

The traditional watershed method does not have good performance although it is very popular in cell detection and segmentation. The challenge is that a regional minimum is required to mark one nucleus before applying watershed methods. Here, we employ the nuclei detection algorithm based on gradient vector field and watershed (Li et al., 2007b,c). First, a Gaussian filter is employed to remove the noise, smooth out the nuclei intensity, and generate a unique local intensity maximum, which represent the position of the nuclei, inside each nucleus. The performance of the Gaussian filter depends on the standard deviation σ , which is obtained experimentally using a set of training images. Then, an algorithm based on a gradient vector field is applied to detect the positions of these nuclei on the filtered image. The mathematical definition of the gradient vector field in each pixel is as follows:

$$\overrightarrow{F}(x,y) = \frac{\partial I(x,y)}{\partial x} i + \frac{\partial I(x,y)}{\partial y} j, \tag{1}$$

where I(x, y) is the filtered image. The gradient vector $\vec{F}(x, y)$ of each pixel points to the local maximum of the filtered image. If we put a small particle on each nuclei pixel and let it move along the pixel's gradient vector lines, the particle will eventually reach the local maximum. As a result, the points covered by a significant number of small particles can be considered as candidates for central points. Noise and redundant central points are removed if the number of their convergent pixels is less than a threshold thr_c . Finally, the watershed process is applied to the processed image. The representative nuclei segmentation results are shown in Fig. 3.

2.3. Processing in neurite images

In the neurite channel, we extracted the centerline of neurite and their branches simultaneously. The purpose is to evaluate the total average length and intensity of the neurites per neuronal cell. The flowchart of our proposed neurite tracing method is shown in Fig. 4. Since this is done on the population basis it is not possible to assign specific neurites to specific cell bodies. Furthermore, our system is intended for use in high-throughput applications, such approach is preferable as it is faster and more robust to cope with well to well variability. At this time, we are not going to distinguish between axons and dendrites, but rather assess the state of all neuronal projections. However, it should be possible to apply our method to neuronal cultures stained with antibodies for either class of projections in order to specifically evaluate them.

2.3.1. Preprocessing—To enhance the image contrast, morphology operation such as 'Bottom-Hat' and 'Top-Hat' transforms are employed.

2.3.2. Soma region detection and neuron nuclei selection—Soma regions, which are bodies of neuron cells, are a cluster of pixels with bright intensities. Since the soma regions, the neuritis regions, and the background have different intensities, fuzzy c-means clustering method is utilized to classify the image pixels into three classes, namely, soma, neurite, and background. Morphological opening operators are also used to smooth the boundary of soma regions (Zhang et al., 2007a).

Nuclei detected in the nuclei channel include nuclei of neurons and other cells. Neuronal nuclei, which are the focus of our study in drug screening, are supposed to be fully overlapped by soma regions. Therefore, neuron nuclei selection can be addressed based on

the information in the soma region of the neurite channel. Here, each nucleus in the nuclei channel is examined: only the nucleus which is fully overlapped with the soma region can be considered as neuron nuclei. Fig. 5 shows an example of soma regions and neuron nuclei selection.

2.3.3. Centerline extraction—As discussed in the introduction, both line-pixel detection and tracing methods address complicated areas as a post-processing step. In this paper, an improved method is presented to detect all the centerlines including branched areas and curvilinear structures simultaneously.

As described in (Steger, 1998), an ideal 'bright and dark' line profile of width 2ω and height h in 1D can be modeled as:

$$f(x) = \begin{cases} h, |x| \le \omega \\ 0, |x| > \omega \end{cases} , \tag{2}$$

The first and second derivative of f(x) can be obtained by convolving f(x) with derivatives of the Gaussian kernel and multiplying by the normalizing coefficient. Given σ_{\min} which makes $-f''(x, \sigma)$ maximum across the scale, t_x is introduced and defined as:

$$t_{x} = \frac{-\sigma_{\min}^{1/2} f'(x, \sigma_{\min})}{f''(x, \sigma_{\min})}$$
(3)

It has been approved that point x can be considered as a centerline point if $t_x \in [-0.5, 0.5]$ and $-f''(x, \sigma_{\min})$ is larger than a user specified threshold (Steger, 1998).

In the 2D case, the curvilinear structure shows characteristics of the 1D line profile in the direction perpendicular to the line. So far, most reports use a 2×2 Hessian matrix to estimate the normal direction of each pixel and calculate t_x based on the direction (Malpica et al., 1997; Steger, 1998; Xiong et al., 2006). Therefore, each pixel has only one direction in this method. However, for branched areas, most pixels have more than one direction, and the eigenvector does not point to the normal direction of the centerline, and t_x always becomes too large to detect the centerline.

To remedy these shortcomings, we propose an improved method, which is based on the same model, to detect all the centerline pixels including single lines and branched areas simultaneously. The basic idea is to apply the formulation in the 1D case with different angles to the image. Before discussing the details, we first define the directions, and introduce the steerable filter theory.

To reduce computational cost, we quantize all directions $(0, 2\pi)$ into 16 directions. Each direction is indicated by an index number from the set $\{0, 1, 2 \dots 15\}$ and has a 22.5° angle to its two neighbor quantized directions.

The steerable filter is composed of a class of basic filters and the response of the steerable filter at an arbitrary orientation is synthesized as a linear combination of basic filters:

$$f^{\theta}(x,y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x,y)$$
(4)

where M is the number of basic filters, $f^{\theta j}(x, y)$ is jth basic filter, θ_j is the jth basic angle, $j \in \{1, 2, ..., M\}$, and $k_j(\theta)$ is the interpolation functions (Freeman and Adelson, 1991).

In this study, two kinds of steerable filters, $G'(x, y, \theta, \sigma)$ and $G''(x, y, \theta, \sigma)$, are designed based on the first and second order derivatives of the circularly symmetric Gaussian function

 $G(x,y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{x^2+y^2}{2\sigma^2})$, where σ represents neurite width (Chen et al., 2007). It was proven in (Freeman and Adelson, 1991) that $G'(x,y,\theta,\sigma)$ can be represented by synthesizing two basic filters with basic angles $\theta_1 = 0$ and $\theta_1 = \pi/2$ linearly, while $G''(x,y,\theta,\sigma)$ can be represented by synthesizing three basic filters with basic angles $\theta_1 = 0$, $\theta_2 = \pi/4$ and $\theta_1 = \pi/2$ linearly. The definition of these two steerable filters is described as follow:

$$G'(x, y, \theta, \sigma) = k_1(\theta) \times G_x(x, y, \sigma) + k_2(\theta) \times G_y(x, y, \sigma),$$
(5)

and

$$G''(x, y, \theta, \sigma) = k_{11}(\theta) \times G_{xx}(x, y, \sigma) + k_{22}(\theta) \times G_{yy}(x, y, \sigma) + k_{12}(\theta) \times G_{yy}(x, y, \sigma)$$

$$\tag{6}$$

with

$$\begin{aligned} k_1(\theta) &= \cos(\theta), & k_2(\theta) &= \sin(\theta), \\ k_{11}(\theta) &= \cos^2(\theta), k_{22}(\theta) &= \sin^2(\theta), k_{12}(\theta) &= \sin(2\theta), \\ G_x(x, y, \sigma) &= -\frac{x}{\sqrt{2\pi\sigma^3}} \exp(-\frac{x^2 + y^2}{2\sigma^2}), \\ G_y(x, y, \sigma) &= -\frac{y}{\sqrt{2\pi\sigma^3}} \exp(-\frac{x^2 + y^2}{2\sigma^2}), \\ G_{xx}(x, y, \sigma) &= \frac{1}{\sqrt{2\pi\sigma^3}} \frac{x^2}{\sigma^2} - 1) \exp(-\frac{x^2 + y^2}{2\sigma^2}), \\ G_{xy}(x, y, \sigma) &= \frac{1}{\sqrt{2\pi\sigma^3}} \frac{xy}{\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}), \\ G_{yy}(x, y, \sigma) &= \frac{1}{\sqrt{2\pi\sigma^3}} \frac{y^2}{\sigma^2} - 1) \exp(-\frac{x^2 + y^2}{2\sigma^2}). \end{aligned}$$

As in the 1D case, we define t_x for each pixel in the image as:

$$t_x(x, y, \theta, \sigma) = -\sigma^{1/2} \frac{f'(x, y, \sigma, \theta)}{f''(x, y, \sigma, \theta)},$$
(7)

where $f'(x, y, \sigma, \theta)$ and $f''(x, y, \sigma, \theta)$ are the first and second normalized derivatives at (x, y) with different directions. For each θ , we find a σ which satisfies the following criterion (Steger, 1998):

$$\sigma = \underset{\sigma}{\arg\min(-f''(x, y, \sigma))} \tag{8}$$

Therefore, formulation can be re-defined as:

$$t_x(x, y, \theta, \sigma) = -\sigma_{\theta}^{1/2} \frac{f'(x, y, \theta, \sigma_{\theta})}{f''(x, y, \theta, \sigma_{\theta})},$$
(9)

where $f'(x, y, \theta, \sigma_{\theta})$ and $f''(x, y, \theta, \sigma_{\theta})$ can be estimated by convolving the image with steerable filters described in Eqs. (7) and (8) and multiplying by the normalized coefficients (Sofka and Stewart, 2006; Xiong et al., 2006).

For each pixel (x, y), we calculate a vector t(x, y) whose length is 16, and (x, y) can be considered as a centerline candidate if we can find a set of θ satisfying the following:

$$\theta = \{\theta_i \mid |t(x, y, \theta_i, \sigma_{\theta_i})| < \frac{1}{2}\}, \quad i = 0, 1, 2, \dots, 15$$
(10)

and $-l''(x, y, \theta_i, \sigma_{\theta_i}) > thr_s$, where thr_s is a user specified threshold and -l'' can be treated as the strength of the line (Zhang et al., 2007c).

Now we are going to find the direction of each centerline candidate. Since each neurite has a local minimum width along its normal direction and it has been proven that σ is proportional to the width(Zhang et al., 2007c), the normal directions of each candidate are obtained by the finding local minimum of $\sigma_{\theta'}$ (Wu et al., 2006):

$$\sigma_{\theta}(x, y) = \min_{\theta} \sigma_{\theta}(x, y), \theta \in [\theta - \Delta\theta, \theta + \Delta\theta], \tag{11}$$

where $\Delta\theta$ is defined as $\pi/8$. This angular resolution $\pi/8$ is selected based on the definition that all the possible directions in a 2D digital image are quantized into 16 directions. There is a tradeoff between resolution and computational cost since higher resolution achieves precise extraction results but the running time and memory requirement will be increased dramatically.

For pixels in single lines, there are two values of θ , while for pixels in branched areas, the number of θ corresponds to the number of intersecting neurites in the branch structure.

2.3.4. Results—To deal with one-pixel thin neurites, we employ non-maximum suppression followed by a hysteresis linking process, which connects the neurite centerlines (Lindblad et al., 2004). Figs. 6 and 7 show the results of neurite labeling. It is clear that the proposed method can extract the centerline of all neurites simultaneously.

This proposed method can also be applied to extract skeletons of other objects with similar structure as neurite, such as vessels, which are bright elongated line-like structures, and whose profile along the vertical orientation can be modeled as Eq. (3). It is expected that our method can extend the curvilinear structures and branch point detection methods to 3D image stacks. As in 2D case, if the object in 3D stacks has the same profile as described in Eq. (3) with certain direction, it can be addressed by the proposed methods.

3. Validation

To test the performance of the proposed methods, two experiments were designed for two channels. In the nuclei channel, the validation was conducted under two conditions: images with touching cells and images without touching cells. In the non-touching cases, we tracked the difference of area and perimeter for each cell between observers and algorithm; while in the touching cases, cell number was calculated to test the accuracy of the algorithm. In the neurite channel, area difference and length difference are used to justify neurite tracing accuracy. In addition, we also perform the branch detection accuracy validation in neurite channel. To reduce the burden of manual labeling, we selected 20 crops among 82 nuclei

images for each case and asked two independent individuals to segment or detect all the nuclei in each crop using the Adobe Photoshop 7.0 software.

3.1. Validation on nuclei image

3.1.1. Validation on the non-touching case—Two variables, area difference and perimeter difference, are involved in validating the results. Let S_A , S_{O_i} , P_A and P_{O_i} denote the nuclei area and perimeter segmented by the algorithm and the *i*th observer. We defined the area and perimeter differences as:

$$dS_{AO_i} = \left| \frac{S_A - S_{O_i}}{S_O} \right|, i = 1, 2$$
(12)

$$dP_{AO_i} = \left| \frac{P_A - P_{O_i}}{P_{O_i}} \right|, i = 1, 2 \tag{13}$$

Next, a set of methods are used to evaluate the results

- **a.** Calculate the mean and standard deviation of two sets of dS_{AO_i} , dP_{AO_i} from two observers, as well as the *p*-value between them using a two-sided paired *t*-test.
- **b.** Calculate the Pearson linear correlation coefficients of S_A and S_{O_i} , PA and P_{O_i} .

Table 1 lists the mean and standard deviation of dP_{AO} and dS_{AO} for different observers as well as p-value between them, which indicates that the results generated from observer 1 and observer 2 are similar. Table 2 lists the Pearson linear coefficients of S_A and S_{O_i} , P_A and P_{O_i} between the algorithm and two observers. These results prove the strong correlation between the algorithm and observer1, the algorithm and observer2, as well as observer 1 and observer 2.

3.1.2. Validation on the touching case—In this case, the number difference between observers and the algorithm is achieved as follows:

$$dN_{AO_i} = |\frac{N_A - N_{O_i}}{N_O}|, i = 1, 2,$$
(14)

where N_{O_i} and N_A are the cell number determined by the *i*th observer and the algorithm.

As in non-touching case, we determined the correlation between the observers and the algorithm according to these methods:

- **a.** Calculate the mean and standard deviation of two sets of dN_{AO_i} from two observers, as well as the p-value between them using a two-sided paired t-test.
- **b.** Calculate the Pearson linear correlation coefficients of N_{O_i} and N_A .

Table 3 lists the mean and standard deviation of dN_{AO_i} for two observers, as well as the p-value between them. Table 4 lists the Pearson linear coefficients of N_{O_i} and NA. These findings imply that the results from observer 1 and observer 2 are similar, and there also exists a strong correlation between the algorithm and observer1, the algorithm and observer2 as well as observer1 and observer2.

3.2. Validation in the neurite image

To validate the neurite tracing accuracy, we design the experiment as follows: 20 images were randomly selected from 81 images. For each image, one sub-region was randomly selected and all the neurites in that sub-region were chosen for the experiment. In addition, two independent observers were instructed to label the neurites manually using Adobe Photoshop 7.0 software. Since different users may produce different labeling results leading to large quantitative differences, manual tracing is guided by the automated results in neurite validation. First, neurite centerlines were extracted automatically, and then observers were asked to examine the results in the former step and modify the error results. Two variables, relative length difference and centerline deviation, were calculated between two observers and the algorithm. l_A , l_{O_i} representing the neurite length generated by the automatic labeling and the ith observer, $A(l_1, l_2)$ denotes the area between l_1 and l_2 , and these variables are defined as follow:

$$L_{AO_i} = \left| \frac{l_A - l_{O_i}}{l_{O_i}} \right|, i = 1, 2 \tag{15}$$

$$S_{AO_i} = \frac{A(l_A - l_{O_i})}{l_{O_i}}, i = 1, 2$$
 (16)

where L_{AO_i} and ζA_{O_i} are relative length difference and centerline deviation between the *i*th observers and the algorithm. Then we used the following methods to validate our results:

- 1. Calculate the mean and standard deviation of L_{AO_i} and ς_{AO_i} from two observers, and also the *p*-values between them using a two-sided paired *t*-test.
- **2.** Calculate the Pearson linear correlation coefficients of l_A and l_{O_i} , $A(l_A, l_{O_i})$ and $A(l_{O_1}, l_{O_2})$.

Table 5 presents the mean value and standard deviation of two sets of L_{AO_i} and ζA_{O_i} , as well as p-values. These indicate that the results produced by two observers are similar. Tables 6 and 7 show the Pearson linear correlation coefficients of L_{AO_i} and ζ_{AO_i} of the two observers and our algorithm. The results show a strong correlation of neurite extraction between the algorithm and the two observers (Table 8).

3.3. Validation in branch detection

In order to evaluate performance of our proposed method on branch points detection, two widely used performance criteria, true positive rate (TPR) and false positive rate (FPR), are introduced to verify the results. When an automatic detected pixel is also marked as bifurcate point by the observer, the detection is a true positive. Otherwise, the detection is a false positive. TPR is defined as the number of the true positive divided by the total number of manual labeling bifurcate pixels and FRP is defined as the number of false positive divided by the total number of manual labeling bifurcate pixels.

Two independent observers are invited to do the validation on 10 randomly selected testing images. Table 9 reports the numerical results for TPR and FPR, where the TPR values are ranged from 83.9% to 89.4% for observer 1 and observer 2. Some statistical results including the *p*-value from the two-sided paired *t*-test and linear correlation coefficients are shown in Table 10 and Table 11. We find that the results produced by the two observers are similar.

3.4. Comparison with other software packages

We have compared our method with other available software packages, including NeuronJ and Neurite Tracer. Basically, NeuronJ, Neurite Tracing and proposed approach are all developed mainly based on the idea of line-pixel detection as described in introduction and Section 2.3. Therefore, they have similar performance of centerline extraction in non-branch area, as shown in Table 12. Evaluation in terms of automation, tracing completeness, the ability of branch area detection, smoothness from NeuriteIQ, NeuronJ and NeuriteTracer is presented in Table 13.

NeuronJ provides very accurate centerline extraction, but it is a semi-automatic method and requires the users to specify the starting and ending points. NeuriteTracer is also a semi-automatic approach. Semi-automatic labeling algorithms do not need to address 'branch problem' since the users always consider branch point as an ending point or a starting point. However, in high content screening, fully automated approach is prerequisite, since hundreds of images are generated from one experiment and hundreds of neurons exist in a single image. So it is impossible for the user to label the starting and ending points for each neurite in every image. Proposed approach introduces steerable filter to improve the traditional algorithm in order to have a better accuracy in bifurcation area and detect all the centerline pixels, including in non-branch area and in bifurcation area simultaneously and fully automatically. Besides, our framework also integrates information from nuclei channel and neurite channel, which automatically offers statistical results for assessing neurite outgrowth from hundreds of neurons. Based on the discussion above, proposed system outperforms the existing software in high content neuron-based imaging.

4. NeuritelQ development and application

As shown in Fig. 8, for easy application, an image processing interface NeuriteIQ is developed to assess quantitatively neurite outgrowth imaged by high content screening based on the algorithms discussed in Section 2. Neuron images from a single well or whole plates are both available to be analyzed in the interface. Apart from processing neuron images, NeuriteIQ computes several key neuron morphology features based on the processed results, as shown in Table 8. Due to the limited memory, only an Excel form containing labeling and measurement results for all the neuron images will be created after batch image processing. NeuriteIQ also generates heatmap when neuron images come from all the wells in a plate. Using color to represent the number, heat-map can evaluate the quality of the data in the whole plate quickly.

As an example, NeuriteIQ is utilized to analyze neuron images from a high content screening. Most researchers agree that primary cause of Alzheimer's disease (AD) is the abnormal accumulation of amyloid beta peptide, the primary component of the amyloid plaques (Degterev et al., 2005; Hardy and Selkoe, 2002). Loss of synaptic connections and neuronal projections has been shown to be a common feature of many neurodegenerative diseases including AD. Based on the fact, a high content screening experiment is designed for studying neuron morphology when they are treated amyloid beta peptide (A-beta peptide).

Neuron preparation is the same as described in section 2.1. Lyophilized A-beta 1–40 was resuspended in sterile ddH2O to 300 μM . The solution was ultracentrifuged at $100,000 \times g$ for 1 h to precipitate any pre-aggregated A-beta fibrils. Following treatment with A-beta 1–40, neurons were washed once with ice cold PBS, followed by fixation in 4% formaldehyde for 30 min at RT. As the experiment plate-map shown in Fig. 9, three dosages of A-beta peptide (2.5 μM , 5.0 μM and 10.0 μM) and three treating periods (24 h, 48 h and 72 h) are employed in treating neurons.

After nuclei detection and neurite labeling, several parameters are obtained to describe neuron outgrowth in each well, including neuronal nuclei number, average area of nuclei, total length of neurite, average neurite length, average neurite intensity and so on. Fig. 10 shows heat-map of nuclei number and average neurite length in the whole plate.

In this application, the parameters included in NeuriteIQ are minimum neurite width, maximum neurite width and the specific threshold for $f''(x, \sigma)$ in Eq. (3). Maximum and minimum neurite widths correspond to the range of σ of Gaussian filters for defining the first and second derivatives of image. In most cases, σ_{max} and σ_{min} can be defined by observation. The selection of step length is a trade-off of calculation accuracy and computational cost. In high content imaging applications, thousands of neuronal images are generated from under the same imaging conditions. Therefore, thr_s is obtained experimentally and will be adjusted when the neuron images come from different biological sources. For example, thr_s for images in wells from A1 to F1 are the same since the neuron image quality from one condition is supposed to be the same.

5. Conclusions

The paper proposed a comprehensive framework to process and analyze *in vitro* neuron image from high content screening. First, a nuclei detection method based on modified watershed is employed to segment all the nuclei in nuclei channel, and then, neuronal nuclei are selected based on the soma regions detected in neurite channel. Next, an improved line-pixel detection method is utilized to automatically extract the curvilinear structures and branched areas simultaneously in the neurite channel. After that, an image processing system, called NeuriteIQ (Neurite Image Quantization) is developed based on the proposed algorithms. Its ability to trace and quantify morphological features in neuron assay based high content screening is crucial to understand biological processes of AD and for development of therapeutic drugs. It always takes NeuriteIQ 40–50 s to process a pair of 512 × 512 neuron images with a Pentium IV 2.8G Hz processor and 2G memory, which is acceptable by biologists.

So far, NeuriteIQ has been applied to assess quantitatively neurite loss imaged by high content screening in order to observe amyloid induced neurite damage and loss in Alzheimer's disease. For future work, we will start compound screening to identify the novel hits for treating Alzheimer's disease. NeuriteIQ will help the users to identify compounds to suppress A-beta induced the neurite degeneration and promote neurite outgrowth. The knowledge gained can be ultimately used for drug discovery for AD and other neurodegenerative diseases.

Acknowledgments

The authors would like to thank all research members in Center of Bioinformatics, Methodist Hospital Research Institute. The authors would also like to thank Miss Charlotte. H Chen from John Hopkins University and Choy Siu Kai from Hong Kong Baptist University for their advices in writing improvement. The work is in part funded by NIH R01 LM008696 and the National Basic Research and Development Program of China (973) under Grant No. 2006CB705700.

References

Al-Kofahi KA, Lasek S, Szarowski DH, Pace CJ, Nagy G, Turner JN, et al. Rapid automated threedimensional tracing of neurons from confocal image stacks. IEEE Transactions on Information Technology in Biomedicine. 2002; 6:171–87. [PubMed: 12075671]

Al-Kofahi Y, Dowell-Mesfin N, Pace C, Shain W, Turner J, Roysam B. Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images. Cytometry. 2007; 73:36–43. [PubMed: 18067123]

Ali C, Hong S, Turner JN, Tanenbaum HL, Roysam B. Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms. IEEE Transactions on Information Technology in Biomedicine. 1999; 3:125–38. [PubMed: 10719494]

- Chen, C.; Li, H.; Zhou, X.; Wong, S. Graph Cut Based Active Contour for Automated Cellular Image Segmentation in High Throughput Rna Interface (rnai) Screening. Washington, DC: ISBI; 2007. p. 69-72.
- Degterev A, Huang Z, Boyce M, Li Y, Jagtap P, Mizushima N, et al. Chemical inhibitor of nonapoptotic cell death with therapeutic potential for ischemic brain injury. Nature Chemistry & Biology. 2005; 1:112–9.
- Freeman WT, Adelson EH. The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1991; 13:891–906.
- Ge, C.; Parvin, B. Model based segmentation of nuclei. Computer Vision and Pattern Recognition; Computer Society Conference on IEEE; 1999; 1999. p. 261
- Hardy J, Selkoe DJ. The amyloid hypothesis of Alzheimer's disease: progress and problems on the road to therapeutics. Science. 2002; 297:353–6. [PubMed: 12130773]
- Li, C.; Xu, C.; Gui, C.; Fox, MD. Level set evolution without re-initialization: a new variational formulation. Computer Vision and Pattern Recognition, 2005. CVPR; Computer Society Conference on IEEE; 2005; 2005. p. 430-6.
- Li F, Zhou X, Ma J, Wong S. An automated feedfack system with hybrid model of scoring and classfication for solving over segmentation problems in RNAi high content screening. Journal of Microscopy. 2007a; 226:121–32. [PubMed: 17444941]
- Li F, Zhou X, Zhu J, Ma J, Huang X, Wong ST. High content image analysis for human H4 neuroglioma cells exposed to CuO nanoparticles. BMC Biotechnology. 2007b; 7:66. [PubMed: 17925027]
- Li G, Liu T, Tarokh A, Nie J, Guo L, Mara A, et al. 3D cell nuclei segmentation based on gradient flow tracking. BMC Cell Biology. 2007c; 8:40. [PubMed: 17784958]
- Lin G, Adiga U, Olson K, Guzowski JF, Barnes CA, Roysam B. A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. Cytometry A. 2003; 56:23–36. [PubMed: 14566936]
- Lindblad J, Wahlby CEB, Zaltsman A. Image analysis for automatic segmentation of cytoplasms and classification of Rac1 activation. Cytometry. 2004; 57:22–33. [PubMed: 14699602]
- Lindeberg T. Feature detection with automatic scale selection. International Journal of Computer Vision. 1998; 30:72–116.
- Malpica N, de Solorzano CO, Vaquero JJ, Santos A, Vallcorba I, Garcia-Sagredo JM, et al. Applying watershed algorithms to the segmentation of clustered nuclei. Cytometry. 1997; 28:289–97. [PubMed: 9266748]
- Meijering E, Jacob M, Sarria JC, Steiner P, Hirling H, Unser M. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. Cytometry A. 2004; 58:167–76. [PubMed: 15057970]
- Nandy, K.; Gudla, PR.; Lockett, SJ. Automatic segmentation of cell nuclei in 2D using dynamic programming. Proceedings of 2nd Workshop on Microscopic Image Analysis with Application in Biology; 2007.
- Sofka M, Stewart CV. Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures. IEEE Transactions on Medical Imaging. 2006; 25:1531–46. [PubMed: 17167990]
- Steger C. An unbiased detector of curvilinear structures. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998; 20:113–25.
- Tsai C-L, Stewart CV, Tanenbaum HL, Roysam B. Model-based method for improving the accuracy and repeatability of estimating vascular bifurcations and crossovers from retinal fundus images. IEEE Transactions on Information Technology in Biomedicine. 2004; 8:122–30. [PubMed: 15217257]
- Wu D, Ming Z, Jyh-Charn L, Bauman W. On the adaptive detection of blood vessels in retinal images. IEEE Transactions on Biomedical Engineering. 2006; 53:341–3. [PubMed: 16485764]

Xiong G, Zhou X, Degterev A, Ji L, Wong ST. Automated neurite labeling and analysis in fluorescence microscopy images. Cytometry A. 2006; 69:494–505. [PubMed: 16680708]

- Zhang Y, Zhou X, Degterev A, Lipinski M, Adjeroh D, Yuan J, et al. A novel tracing algorithm for high throughput imaging screening of neuron-based assays. Journal of Neuroscience Methods. 2007a; 160:149–62. [PubMed: 16987551]
- Zhang Y, Zhou X, Degterev A, Lipinski M, Adjeroh D, Yuan J, et al. Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays. NeuroImage. 2007b; 35:1502–15. [PubMed: 17363284]
- Zhang Y, Zhou X, Witt RM, Sabatini BL, Adjeroh D, Wong STC. Dendritic spine detection using curvilinear structure detector and LDA classifier. NeuroImage. 2007c; 36:346–60. [PubMed: 17448688]
- Zhou X, Wong S. High content cellular imaging for drug development. IEEE Transactions on Signal Processing Magazine. 2006; 23:170–4.

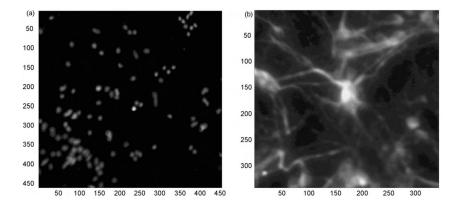


Fig. 1. Sample neuron images from (a) nuclei channel and (b) neuite channel. The number in x and y axes shows the size of the image. The units of x and y axes are both 'pixel'. The resolution of the image is 1.23 μ m/pixel. Figs. 3, 5, 6 and 7 have the same measurement.

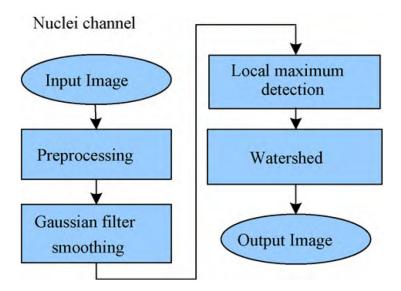


Fig. 2. Image processing pipeline in nuclei channel.

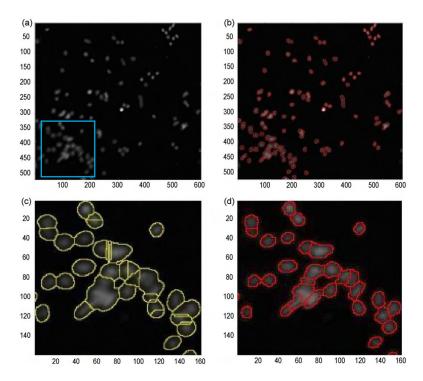


Fig. 3.
Segmentation results in the nuclei channel. (a) A raw image; (b) shows segmented image using proposed method in the paper; (c) and (d) are detailed segmented images of the blue square in (a) with traditional watershed and our proposed method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

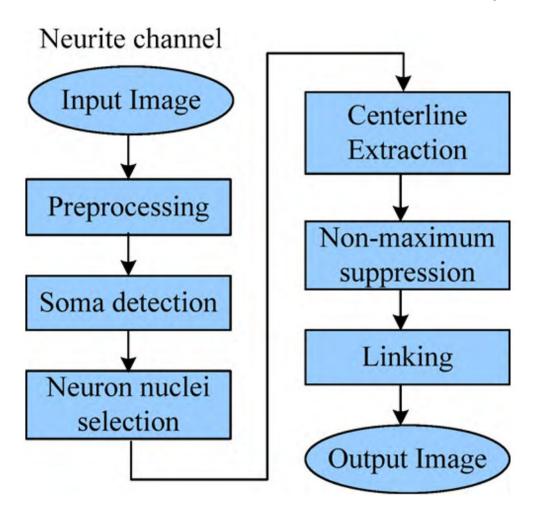


Fig. 4. Image processing pipeline in neurite channel.

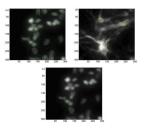


Fig. 5.

Soma region detection in neurite channel and neuron nuclei selection in nuclei channel. (a)

Shows detection results of all nuclei in nuclei channel; (b) shows detected soma region in neurite channel, which are circled by yellow lines; (c) selected neuron nuclei. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

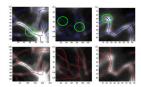


Fig. 6. Results based on Hessian matrix and the proposed method. (a)–(c) are results based on the Hessian matrix; (d)–(f) are derived from proposed method. Neurites in green circles are labeled with higher accuracy by the proposed method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

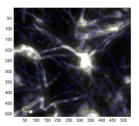


Fig. 7. Processed results in a neurite image. Connected and single centerlines are labeled in blue. Some regions are circled in yellow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

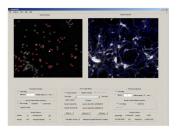


Fig. 8. Interface of NeuriteIQ, a useful software platform to process and analyze neuron images. In nuclei channel, only neuronal cell nuclei are labeled.

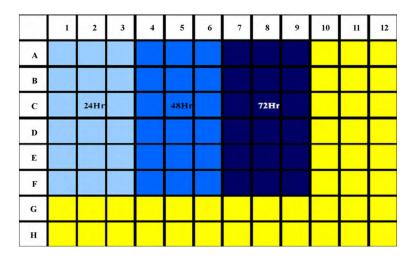


Fig. 9. Plate-map of the high content screening experiment. Wells in different colors represent neurons with different A-beta treating periods: wells in azury (columns 1–3, rows A–F) are neurons with treating period of 24 h; wells in blue (columns 4–6, rows A–F) are neurons with treating period of 48 h; wells in navy blue (columns 7–9) are neurons with a treating period of 72 h; wells in the yellow area are neurons without treated with any A-beta. Columns 1, 4 and 7, rows A–F are neurons receiving 2.5 μM of A-beta; Columns 2, 5 and 8, rows A–F are neurons receiving 5.0 μM of A-beta; Columns 3, 6 and 9, rows A–F are neurons receiving 10.0 μM of A-beta. Other wells, as in the yellow part, are considered as control case, where neurons are not treated with any A-beta or other drugs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

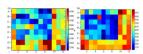


Fig. 10.

Heat-maps derived from NeuriteIQ after processing and calculating. (a) is neuronal nuclei number and (b) is average neurite length. Dark red and dark blue represent highest and lowest number separately. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Table 1

two observers, as well as p-value between them.

Huang et al.

ı ne mean	and sta	ndard de	evianon (or area a	na pern	i ne mean and standard devianon of area and perimeter difference for two o
	dS_{AO}			dP_{AO}		
	Mean	Std	Mean Std p-Value Mean Std p-Value	Mean	Std	p-Value
Observer1 0.0948 0.0645 0.2202 0.1291 0.0802 0.3350	0.0948	0.0645	0.2202	0.1291	0.0802	0.3350
Observer2 0.0780 0.0496	0.0780	0.0496		0.1047 0.0846	0.0846	

Page 24

Huang et al.

Table 2

Pearson linear coefficients of area and perimeter generated from the algorithm, observer1 and observer2.

	Area			Perimeter		
	Algorithm	Algorithm Observer1 Observer2 Algorithm Observer1 Observer2	Observer2	Algorithm	Observer1	Observer2
Algorithm	1	0.9795	0.9692	1	0.8536	0.8366
Observer1	0.9795	ı	0.9938	0.8536	ı	0.9353
Observer2 0.9692	0.9692	0.9938	1	0.8366	0.9353	1

Page 25

Table 3

The mean and standard deviation of the cell number difference for two observers, as well as the p-value between them.

	dN _{AO}		
	Mean	Std	<i>p</i> -value
Observer1	0.1368	0.0775	0.9816
Observer2	0.1360	0.1265	

Table 4

Pearson linear coefficients of cell number between the algorithm, observer1 and observer2.

	Cell number	r	
	Algorithm	Observer1	Observer2
Algorithm	-	0.9433	0.9178
Observer1	0.9433	-	0.9933
Observer2	0.9178	0.9933	-

Table 5

Huang et al.

The mean and standard deviation for difference in length, deviation in centerline, and the p-values for two observers.

	$L_{ m AO}$			ζvo		
	Mean	Std	p-Value	Mean	Std	p-Value
Observer1	0.0539	0.0417	0.8481	0.8230	0.4162	0.8260
Observer2	0.0592	0.0483		0.8886	0.8886 0.6180	

Page 28

Table 6

The Pearson linear correlation coefficients between the results (neurites length) generated from the algorithm, observer1 and observer2.

	Neurite leng	gth	
	Algorithm	Observer1	Observer2
Algorithm	=	0.9772	0.9819
Observer1	0.9772	-	0.9927
Observer2	0.9819	0.9927	-

Table 7

The Pearson linear correlation coefficients between the results (area between two neurites) generated from the algorithm, the observer1 and the observer2.

	Area betwe	en centerlin	e
	$A(l_{O_1}, l_{O2})$	$A(l_A, l_{\mathcal{O}_1})$	$A(l_A,l_{O2})$
$A(l_{O_1}, l_{O_2})$	_	0.8569	0.9032
$A(l_A,l_{O_1})$	0.8569	_	0.8320
$A(l_A, l_{O2})$	0.9032	0.8320	-

Table 8

Neuron morphology features in NeuriteIQ.

Nuclei channel	Neurite channel
Nuclei number	Total neurite length
Neuronal nuclei number	Average neurite length
Average area of neuronal nuclei	Average neurite intensity
Average intensity of neuron nuclei	Branch points number

Huang et al.

Table 9

Performance of proposed method on 10 testing images from two observers.

Testing image index		1	2	3	1 2 3 4 5 6 7 8 9	w	9	7	8	6	10
Observer1	True positive rate (%) 87.5 84.2 88.1 87.3 89.4 83.9 85.4 86.7 84.2 85.7	87.5	84.2	88.1	87.3	89.4	83.9	85.4	86.7	84.2	85.7
	False positive rate (%) 6.4 5.7 6.1 6.2 3.8 2.7 4.4 7.0 5.2 5.3	6.4	5.7	6.1	6.2	3.8	2.7	4.4	7.0	5.2	5.3
Observer2	True positive rate (%)	85.8	85.7	85.8 85.7 87.0	85.2	88.5	82.7	88.5 82.7 86.4	85.9	85.5	85.9
	False positive rate (%) 6.9 5.5 5.8 5.8 4.1 3.0 4.9 6.6 6.0 4.8	6.9	5.5	5.8	5.8	4.1	3.0	4.9	9.9	0.9	4.8

Page 32

Table 10 Statistical analysis of TPR between two observers.

	Mean	Deviation	p-Value	Pearson correlation coefficient
Observer 1	86.2	1.85	0.625	0.9286
Observer 2	85.9	1.45		

Table 11 Statistical analysis of FPR between two observers.

	Mean	Deviation	p-Value	Pearson correlation coefficient
Observer 1	5.28	1.31	0.932	0.9359
Observer 2	5.33	1.16		

Table 12

The mean and standard deviation for difference in length, deviation in centerline, and the p-values for NeuronJ, NeuronTracer and proposed method.

	L_{AO}		ζ_{AO}	
	Mean	Std	Mean	Std
NeuronJ	0.0332	0.0212	0.6544	0.3994
NeuriteIQ	0.0539	0.0417	0.8230	0.4162
NeuriteTracer	0.0556	0.0393	0.8199	0.4871

Table 13
Features comparison among different neuritis extraction approaches.

Algorithm	NeuronJ	NeuronTracer	NeuriteIQ
Automatic	Semi	Semi	Full
Completeness	Good	Good	Good
Branch detection	Fair	Fair	Good
Smoothness	Good	Good	Good