An adjustable-threshold algorithm for the identification of objects in three-dimensional images

Artem L. Ponomarev1,∗ and Ronald L. Davis1, 2

1Department of Molecular and Cellular Biology, USA and 2Psychiatry and Behavioral Sciences, Baylor College of Medicine, Houston, Texas 77030, USA

Received on October 1, 2002; revised on February 3, 2003; accepted on February 6, 2003

ABSTRACT

Motivation: To develop a highly accurate, practical and fast automated segmentation algorithm for three-dimensional images containing biological objects. To test the algorithm on images of the Drosophila brain, and identify, count and determine the locations of neurons in the images.

Results: A new adjustable-threshold algorithm was developed to efficiently segment fluorescently labeled objects contained within three-dimensional images obtained from laser scanning confocal microscopy, or two-photon microscopy. The result of the test segmentation with Drosophila brain images showed that the algorithm is extremely accurate and provided detailed information about the locations of neurons in the Drosophila brain. Centroids of each object (nucleus of each neuron) were also recorded into an algebraic matrix that describes the locations of the neurons.

Availability: Interested parties should send their request for the NeuronMapperTM program with the segmentation algorithm to artemp@bcm.tmc.edu.

Contact: artemp@bcm.tmc.edu

INTRODUCTION

The rapid progress in microscopy and three-dimensional imaging has produced many new challenges in image analysis. For instance, many of the images now obtained from confocal or two-photon microscopy are rich with information, and may contain hundreds or thousands of objects of variable size and location. The amount of information contained within these images frequently prohibits a clear and complete analysis by simple inspection. Computational tools are required to sort through the images in order to obtain a better comprehension of the images, or to extract the salient features or those that are important to the individual investigator.

One type of such analysis is called segmentation.

Segmentation is broadly defined as the computational steps required for identifying discrete objects or image areas that are relatively homogeneous. Several strategies have been devised to accomplish this. Several segmentation approaches are based on thresholding. In the simplest of images, objects of interest are represented by pixels of high intensity relative to the background. Thus islands of contiguous high pixel intensity can be defined by simply establishing a threshold. This can be effective with objects that are well separated, and when they are represented by pixels with high and uniform pixel intensity relative to the background. Another way is to make the threshold variable, either locally or through an iterative scheme, and base the image analysis on a mathematical construct that works with light intensity distributions and/or the geometric properties of the objects to be segmented. There is also a strategy that centers on using model-based schemes, such as neural networks or oscillator networks (as in the LEGION algorithm, Chen and Wang, 2002) that can be made to produce the desired result.

The need for a new segmentation algorithm was motivated by our desire to quickly and efficiently identify the nuclei of neurons in high-resolution image stacks collected from Drosophila brains. This represents one step in our overall goal to map all neurons of the Drosophila brain and to determine the extent of stereotypy that exists in brain organization between individual organisms. For example, it is unknown whether there is substantial stereotypy between individuals or whether neuronal location is highly plastic. Even the simple question of the number of neurons in the adult fly brain has not been answered with any appreciable accuracy.

We set out to determine the number and location of neuronal nuclei from the noisy and sometimes ambiguous image information obtained from two-photon microscopy. The nuclei of neurons in Drosophila brains were first labeled with a fluorescent antibody against an antigen found in all neuronal nuclei. We will describe the details of the specimen preparation and imaging elsewhere; here

∗To whom correspondence should be addressed.
we describe a new segmentation algorithm that provides the automatic identification of nuclei within a typical image stack. Each Z-stack typically represents a volume of 158.7 × 158.7 × 100 μm and each voxel represents a volume of 0.31 × 0.31 × 0.3 μm.

METHODS

Several Z-sections of a typical Z-stack are illustrated in Figure 1a–c. The challenge was to extract precise information from the noisy optical information, and to segment the nuclei from the background. There are two clear facts about the nuclei: they are on average brighter than the background, and are generally round in shape. However, there are fluorescence intensity fluctuations both in the background, and are generally round in shape. Therefore we needed to develop a practical approach that would handle this noisy input, and would define the nuclei with precision. Another goal was to replace the large and complex images with a point-list matrix of the nuclei in space.

We define the optical center of mass as the centroid for each nucleus. This provides for a simple and intuitive way to assign a point in space to that object. Another idea that we introduced was to define the object (the nucleus in this case) as a set of image voxels that approximates the visible nucleus (that is, a nucleus as it appears to a trained researcher) in size, shape, and volume. The algorithm therefore partitions the whole space of the image into the background and ‘defined nuclei’ subsets according to a defined set of rules, and then tests the partition to determine if it is acceptable and if the created spatial pattern of nuclei matches the visible pattern with spatial precision. In such a scheme, the human operator is the final judge of the efficiency of the algorithm and his/her responsibility is to adjust the parameters that control the partition to maximize the probability of proper segmentation. Because of imaging artifacts and the limit of resolution from microscope optics, such segmentation will never be 100% precise except with the simplest of images. Nevertheless, the algorithm that we developed allows for segmentation of brain nuclei with a precision of 98% as judged by an experienced biologist through visual inspection. The program is fast enough for our practical needs, processing a 512 × 512 × 512 Z-stack in about 1 hour on a PC with a 1.5 GHz microprocessor.

ALGORITHM

In what follows we give the formal definition of our adjustable threshold algorithm.

DEFINITION 1. We denote \( v_{xyz} \) a voxel located at \((x, y, z)\) location in a stack; \( I_{xyz} \) is the gray scale intensity of \( v_{xyz} \).

A voxel is a three-dimensional pixel that has four associated values: \((x, y, z)\) and \(I\). Before inputting an array of voxels into the program, the pre-processing steps of de-convolution, deblurring and smoothing improve the final result.

DEFINITION 2. An interconnected set (IS) of voxels in space is a set of a certain size \( V \), such that there exists a path defined on voxels within the set that connects one voxel to another within the set. A path is a line in space drawn continuously from one voxel to another in \(±x, ±y, ±z\) directions only (no diagonal lines are allowed in a path).

One can visualize a IS as an island in space that has an arbitrary shape or holes inside, but is one object in a topological sense.

DEFINITION 3. A defining set (DS) of a voxel \( v_{xyz} \) is a IS that contains \( v_{xyz} \) with all other voxels \( v_{uvw} \) within IS having \( I_{uvw} \geqslant I_{xyz} \).

There is one and only one DS for each voxel in an image. This construct is based on the very simple idea that if a voxel were to belong to a nucleus, all of its interconnected neighbors with the same or higher intensity also belong to that nucleus. (This is contrary to some more general algorithms where a voxel has only a certain probability of belonging to an object based on some measure of voxel similarity defined by the vicinity of this voxel—see the Discussion).

DEFINITION 4. A valid DS (or VDS) is a set that satisfies the restricting conditions (RC’s). A valid voxel \( v_{xyz} \) is a voxel whose DS is a VDS.

In the program Definition 4 effectively discards voxels that cannot possibly belong to any nucleus, and keeps the valid voxels for further processing and nucleus definition without applying a global threshold. The RC’s are purely geometric contextual conditions that impose parameter restrictions on a DS to approximate the size and shape of a visible nucleus. These are described in what follows.

DEFINITION 5. If a group of valid voxels forms a IS, the unity of their DS’s (which are also VDS’s) is called a segmented nucleus.

Using Definition 5 the program does the final sorting of the valid voxels and combines them into segmented nuclei, which are very close in size and shape to the corresponding nuclei in an image. (A graphics interface is attached to the algorithm to visualize the output of the program and to do visual checks, error counting, and allows manual correction if necessary.)
An adjustable-threshold algorithm

Fig. 1. Shown are three sections from a typical image of a *Drosophila* brain at 12 µm deep (a), at 17 µm deep (b) and at 23 µm deep (c). In panels (d–f) the result of smoothing and segmentation is shown for the same three slices as in (a–c). The centroids of nuclei are shown as square marks (3×3×1 voxels, the center of the mark is the actual centroid). Since only isolated slices from the same Z-stack are shown, not all nuclei are marked: more centroids are located in other planes. The grid was introduced to help with visual checks.

**COROLLARY.** From Definitions 1–5 it also follow that a segmented nucleus is equivalent to its dimmest valid voxel’s VDS.

**DEFINITION 6.** A centroid is a point in space given by a sum of (x, y, z) locations of the voxels belonging to a segmented nucleus divided by V (the volume of the segmented nucleus).

A centroid appears to be roughly at the center of a visible nucleus. This is a standard technique for assigning a single point to an object. The matrix of centroids is the resulting three-dimensional map of the objects in the image that can be studied further.

This algorithm can be generalized for other applications, even outside of biological imaging since a DS, or several types of DS’s, can be defined using conditions other than those used in Definition 3. For instance, color can be introduced along with intensity for aerial mapping to distinguish various features of a landscape. Non-spherical
objects can be identified by introducing asymmetries in the RC’s, or the RC’s can be made probabilistic. (That is, characteristic statistical properties of objects can be given as distributions and utilized to determine a probability that a particular RC is satisfied.) The use of various sets of RC’s in Definition 4 can generalize the algorithm for biological objects other than nuclei.

For fly brain nuclei, we chose the following set of RC’s:

**DEFINITION 7.** \( V_{\text{max}} \) is the maximum volume of a DS.

**DEFINITION 8.** \( V_{\text{min}} \) is the minimum volume of a DS.

**DEFINITION 9.** \( L_{\text{max}}^{xy} \) is the maximum extent of a DS in the \( x \) and \( y \) directions.

**DEFINITION 10.** \( L_{\text{max}}^{z} \) is the maximum extent of a DS in the \( z \) direction.

**DEFINITION 11.** \( G_{\text{max}} \) is the maximum sphericity:

\[
G_{\text{max}} = \frac{R_{g}^{2}}{N^{2/3}} \tag{1}
\]

where \( R_{g} \) is the gyroradius of a nucleus, a standard measure of sphericity in various areas of science. It is the mean-square distance between the voxels comprising the nucleus and its center of mass:

\[
R_{g}^{2} = \frac{1}{N} \sum_{i=0}^{N} (r_{i} - R_{CM})^{2} \tag{2}
\]

where \( r_{i} \) is the location of a voxel, \( R_{CM} \) is the center of mass of a nucleus, and \( N \) is the number of voxels in the object, or the volume.

The gyroradius assumes the minimum value for a perfect sphere with no holes. It depends, however, on the volume of a sphere. \( G_{\text{max}} \) is normalized by the volume of the nucleus so that it is volume-independent [as in Equation (1)] and depends only on the shape of the nucleus. In the continuous limit a spherical object minimizes this parameter \((G_{\text{max}})_{\text{min}} \rightarrow 0.23\). Since real nuclei are not perfectly round, the restricting value of \( G_{\text{max}} = 0.3 \) was chosen for optimal segmentation. The check for sphericity is performed by a subalgorithm, termed the splitting algorithm, that checks suspected fused nuclei (if such are left) after segmentation based on the other RC’s. If such is the case the local threshold is further raised to split an object, and to make the resulting parts satisfy all the RC’s.

The RC’s are user-defined and can be optimized by trial and error. If one is interested in the number of nuclei only, but not in their locations, the precision is even higher, since different errors compensate for each other: split nuclei and improperly identified nuclei add to the count, but missed and fused nuclei reduce the count. The accuracy has reached 98.1% so far for automatic counting. This is the result for a cleanly imaged stack. The algorithm is not designed to correct imaging problems such as bleaching or optical aberrations.

It may appear that the concept of the threshold does not appear anywhere in the algorithm. In reality the intensity of any given voxel is chosen as the threshold for the DS that corresponds to that voxel. Since some voxels are discarded as not belonging to any nuclei and the resulting VDS’s correspond to the DS’s of their dimmest voxels, the intensities of those dimmest voxels of the defined nuclei are the final thresholds, which may vary from nucleus to nucleus. Figure 2 illustrates a situation when two nearby nuclei are segmented by allowing for sufficiently high and different threshold values.

**RESULTS AND IMPLEMENTATION**

For our specific problem and for the chosen set of RC’s we found it easiest to implement the code as three sweeps through the stack. In the first pass voxels whose associated DS satisfies the maximum conditions for volume and extent are selected. In the second, the selected voxels are united in topologically isolated islands and tested for the minimum volume condition and for sphericity. The program then makes another ‘sweep’ and attempts to split all voxel sets that fail the sphericity condition. In any particular implementation of the code the CPU time required for segmentation is only \( \text{const} \times V \), where \( V \) is the total size of the input dataset and \( \text{const} \) only depends on the nuclei density, the computer speed and the optimization level. The proportionality only to the first power of the input data volume \( V \) makes the code fast.

The results of segmentation of any typical Z-stack can be displayed in a custom-designed GUI that we named NeuronMapper. It is an object-oriented Multiple Document Interface written in Visual C++ with C++ wrapper classes for OpenGL and the Libtiff library. Figure 1d–f
illustrates the three Z-slices illustrated in Figure 1a–c after segmentation. The optical center of each segmented nucleus, or the centroid, is displayed as a colored mark superimposed on the nucleus. Many of the nuclei shown in Figure 1 remain unmarked because their centroid occurs on a nearby Z-slice of the stack. This graphical marking facilitates subsequent visual inspection of the stack in order to determine the precision of segmentation. We have also added the feature that the XYZ locations of all centroids in space are listed in tabular output relative to the lower left corner of the first Z-slice. In our case, the coordinates represent micron distances (typically, a unit distance in the matrix corresponds to 0.3 µm). The point-list matrix of the brain (the x, y, z coordinates of the nuclei) can be visualized with the NeuronMapper GUI.

The NeuronMapper graphical interface comprises several preprocessing modules (smoothing, for example), the segmentation algorithm, and multiple graphical displays in OpenGL. The program is able to manipulate TIFF files and display the result of segmentation graphically, as shown in Figure 1. As a stand-alone program the segmentation algorithm can easily be transferred to any platform in its C version (as long as it is linked to the LIBTIFF library). Several different TIFF files can be written as the graphical output for manual checks of accuracy. For instance, one of the files has defined nuclei displayed in colors chosen at random on the original image background in gray scale, so that the user can easily discern neighboring nuclei. Another file has the defined centroids depicted as small square marks superimposed onto the original image. The segmented and original images can be compressed and stored in DVD-R archives for future reference.

**DISCUSSION**

There are a number of already developed segmentation algorithms (Mardia and Hainsworth, 1988). The consensus has been that for segmentation problems, there is no general approach. Some of the algorithms are not suited for three-dimensional images or not designed for grayscale images. For instance, the segmentation algorithm in the Metamorph package is easily applicable to a two-dimensional representation of a flow cytometry device, in which cells can be counted after simple thresholding. It is error-prone because red cells may appear quite different at different angles of orientation with respect to the viewer. A three-dimensional segmentation task is not easy to implement in this package. The Amira package does not allow fast segmentation of a large number of small nuclei because it is not fully automatic. The LEGION algorithm (Shareef et al., 1997; Chen and Wang, 2002) seems to involve a model that mimics the workings of a human eye to recognize images. Even though we agree on the importance of such research, we believe that for a complex model it would be difficult to analyze the dependence of segmentation quality on model parameters. Our segmentation of images based on simple geometric ideas should give precise results, which are easy to analyze and refine.

Our algorithm was based on intuitive ideas, such as voxel similarity sorting and contour identification. Indeed, our algorithm combines voxels above a local threshold into a segmented nucleus, which is a similarity principle; and defines a contour with voxels of equal intensity, which is the surface of the segmented nucleus.

The most general approach prior to our segmentation algorithm is based on the likelihood estimation of a given voxel belonging to a given population in the image (Oh and Lindquist, 1999; Mardia and Hainsworth, 1988). A population can be either the background, or a nucleus, or any other object. A particular realization of the general scheme for two populations is the Kriging algorithm, which is based on the initial assignment of subsets of voxels to two separate populations with some voxels left unassigned (Oh and Lindquist, 1999). Then a special mathematical technique, the minimization of variance estimation (kriging), estimates whether each unassigned voxel belongs to the first or second population of voxels. In our algorithm the properties of a VDS are determined by fixed geometric conditions, not by the distributions of voxel intensities in the objects. Thus our approach is contextual and depends on the geometric properties of the object. In addition, many different types of objects with different geometric properties could, in principle, be segmented with the approach that we describe. We believe that our fast and practical segmentation algorithm can be utilized in many areas of biology where the count, location and identification of biological objects are needed.

**REFERENCES**


