

A Multi-Model Approach to Simultaneous Segmentation and Classification of Heterogeneous Populations of Cell Nuclei in 3D Confocal Microscope Images

Gang Lin,¹ Monica K. Chawla,² Kathy Olson,² Carol A. Barnes,² John F. Guzowski,³ Christopher Bjornsson,⁴ William Shain,⁴ Badrinath Roysam^{1*}

¹ECSE Department and Center for Subsurface Sensing and Imaging Systems, Rensselaer Polytechnic Institute, Troy, New York

²Evelyn F. McKnight Brain Institute, University of Arizona, Tucson, Arizona

³Center for the Neurobiology of Learning and Memory, University of California, Irvine, California

⁴Laboratory of Nervous System Disorders, NYS-DOH, Wadsworth Center, Albany, New York

Received 19 July 2006; Revision Received 18 April 2007; Accepted 14 May 2007

Grant sponsor: National Institutes of Health; Grant number: AG023309; Grant sponsor: National Science Foundation; Grant number: EEC-9986821; Grant sponsor: NIH; Grant number: NSR01-044287.

*Correspondence to: Badrinath Roysam, Professor, JEC 7010, Rensselaer Polytechnic Institute, 110, 8th St., Troy, NY 12180, USA

Email: roysam@ecse.rpi.edu

Published online 25 July 2007 in Wiley InterScience (www.interscience.wiley.com)

DOI: 10.1002/cyto.a.20430

© 2007 International Society for Analytical Cytology

• Abstract

Automated segmentation and morphometry of fluorescently labeled cell nuclei in batches of 3D confocal stacks is essential for quantitative studies. Model-based segmentation algorithms are attractive due to their robustness. Previous methods incorporated a single nuclear model. This is a limitation for tissues containing multiple cell types with different nuclear features. Improved segmentation for such tissues requires algorithms that permit multiple models to be used simultaneously. This requires a tight integration of classification and segmentation algorithms. Two or more nuclear models are constructed semiautomatically from user-provided training examples. Starting with an initial over-segmentation produced by a gradient-weighted watershed algorithm, a hierarchical fragment merging tree rooted at each object is built. Linear discriminant analysis is used to classify each candidate using multiple object models. On the basis of the selected class, a Bayesian score is computed. Fragment merging decisions are made by comparing the score with that of other candidates, and the scores of constituent fragments of each candidate. The overall segmentation accuracy was 93.7% and classification accuracy was 93.5%, respectively, on a diverse collection of images drawn from five different regions of the rat brain. The multi-model method was found to achieve high accuracy on nuclear segmentation and classification by correctly resolving ambiguities in clustered regions containing heterogeneous cell populations. © 2007 International Society for Analytical Cytology

• Key terms

cell nuclei; segmentation; classification; watershed algorithm; region merging; model-based; Bayesian estimator; parzen window; batch processing; 3D confocal microscopy

THREE-dimensional (3D) segmentation of fluorescently labeled cell nuclei in confocal image stacks is an essential image analysis task required in numerous quantitative studies (1–3). The results of nuclear segmentation can be used for counting, morphometry, classification, and associative measurement of secondary fluorescent markers (3,4). As an example, Figure 1 shows fluorescently labeled neuronal and glial cell nuclei from various rat brain regions, drawn from a study involving compartmental and temporal analysis of FISH (fluorescence in situ hybridization) signals (3D-cat-FISH). Nuclear segmentation is an essential first step to quantitating FISH data (4,5).

The seemingly straightforward task of segmenting blob-like nuclei continues to present challenges, especially in high-throughput applications requiring high levels of accuracy, automation, reliability, and speed. Most of the challenges are rooted in the complexity and variability of nuclear appearance across images, staining and imaging protocols, and ambiguities associated with tight clustering of objects (6,7). Other sources of error include nonuniform staining, imaging artifacts such as depth-

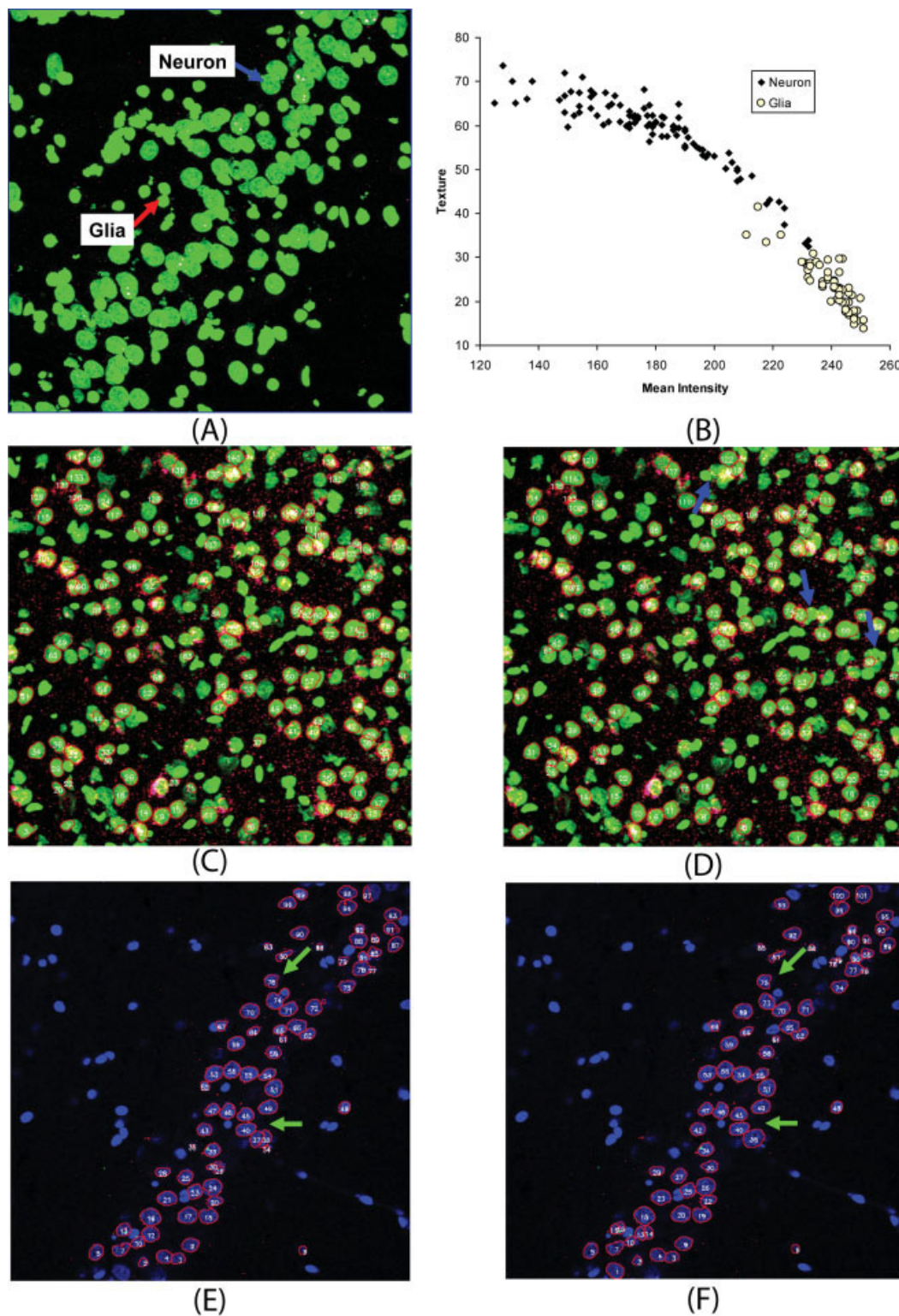


Figure 1. Illustrates the proposed multi-model based merging and classification method. (A) The maximum-intensity projection of a 3D confocal image stack of fluorescently labeled cell nuclei in the CA3 region of the rat Hippocampus. Two different cell types (neuron and glia) are indicated by arrows. (B) Scatter plot of distinctive features (mean intensity and texture) for two object types. (C) and (E) Segmentation examples of our previous single model based merging, where errors at touching objects between different types are indicated. (D) and (F) The proposed multi-model based merging algorithm resolves these cases correctly.

dependent attenuation, the inherent anisotropy of confocal images, and artifacts from tissue sectioning resulting in nuclei that are cutoff or otherwise damaged.

Model-based segmentation algorithms have demonstrated the highest levels of segmentation accuracies to date (8–10). They rely on a mathematical model of the expected nuclear morphology and intensity profiles, and variations thereof. Various object modeling methods have been proposed in the literature—for example, shape based modeling (11–13) and deformable modeling (14,15). Deformable object modeling methods based on active contours are widely used for medical image segmentation (16–18). The level sets method is an implicit representation of deformable surface models (19) that computes the motion of a moving front by solving a partial differential equation on a volume, usually combining a data-fitting term with a smoothing term. This method allows for geometric surface deformation and is topology-free. Although its potential has been demonstrated for 3D medical image segmentation, level-set based methods are impractical for high-throughput or large-scale nuclear segmentation. First, they are computationally expensive, so do not scale easily when the number of nuclei in the confocal stacks is large. Second, they lack the ability to split clusters of objects in a general manner. Finally, they require effective initialization and are subject to drastic failure modes, such as “leakage” of a contour. Graph-cuts based segmentation algorithms have been proposed in the literature (20,21). In this methodology, the cost of a cut is computed using an energy function incorporating low-level information such as boundary and regional constraints. Although this method has the ability to compute globally optimal solutions, it has been difficult to include high-level information, and correctly split clusters of objects such as touching nuclei.

The prior literature on model-based nuclear segmentation has focused on homogeneous populations of cell nuclei in a field for which a single model is sufficient. A homogeneous population can be derived by preprocessing the images to eliminate one sufficiently distinctive subpopulation of nuclei (10). Such an approach is effective for a specific application, but not for a broader class of applications. Generally, one or more subpopulations may not be sufficiently distinctive to be sequestered reliably by preprocessing. In addition, the criteria for sequestering a subpopulation may vary from one application to another. Even when sequestration is carried out, errors in preprocessing lead to errors in model-based segmentation since the underlying modeling assumptions are violated. Finally, the greatest need for model-based methods arises when nuclei of different types are clustered together. Overall, there is a need to develop broadly applicable algorithms that explicitly incorporate multiple nuclear models, each corresponding to an identifiable subpopulation of cells.

In this article, we extend our prior work (10,22) to handle multiple types of objects. This produces two benefits: a higher level of merging accuracy is achieved, and a morphology-based classification of nuclei is obtained at the same time. The construction of multiple object models is straightforward, but their application to actual segmentation is non-

trivial. Models can be estimated (learned) from a set of examples delineated by the user using an interactive image annotation tool (the “training set”). To apply these models to nuclear segmentation, the algorithm must correctly guess the correct model to apply in each case, even as the segmentations are being computed. In other words, an object classification (model selection) algorithm must be tightly integrated into the processing steps of the model-based segmentation algorithm.

The need for such an integrated algorithm arises in the analysis of nonhomogeneous clusters of nuclei. Indeed, automatic model selection would be trivial if the nuclei appeared in isolation, i.e., were not clustered together. At the next level of complexity, homogeneous clusters of nuclei are segmented well using widely available watershed algorithm based techniques. Handling nonhomogeneous clusters containing nuclei of different types is the primary subject of this article.

As in our prior work (22), our methodology is based on deliberate initial over-fragmentation of the image data using a 3D watershed algorithm, and subsequent model-based merging of fragments. Our new methodology combines several ideas from the prior literature on region-based segmentation (22–31), model-based object merging (22), model selection, and classification (32,33). Our algorithm generates a set of candidate objects formed by merging two or more fragments using an efficient hierarchical strategy. For each candidate, a confidence score is computed. This score is used as a quantitative guide to making fragment merging decisions. Upon the completion of the procedure, an object classification is also produced.

MATERIALS AND METHODS

Object Modeling

The proposed method requires mathematical modeling of each type of object present in the image data. Building an accurate and universally applicable model to aid the segmentation of cell nuclei remains a challenge for several reasons. The morphological variability among nuclei is significant, both within an anatomic region, and across regions. Additional variability arises from staining variations, instrument variations, and settings (e.g., laser power, pinhole size, contrast and gain offsets, etc.), imaging artifacts such as depth-dependent attenuation, photo bleaching, imaging noise, and various distortions.

In this work, we adopt a statistical modeling approach that is broadly applicable yet computationally efficient. It is based on a selected set of morphological and intensity-based features of objects. The choice of these features is important. They must be chosen to fully characterize the objects in question, and be capable of discriminating the intact objects from objects of other classes and artifacts such as multinuclear clusters or subnuclear fragments. In addition, they must be general enough to account for object diversity and image variability. Earlier articles used descriptive features such as volume, intensity, texture, convexity, and shape factors (10). Features like volume and intensity are simple yet versatile. Others are

less so—for example, it remains a challenge to obtain a good shape descriptor for an arbitrary 3D object. Since we are considering multiple object types simultaneously during the merging procedure, features that can distinguish different object types are useful, in addition to the features that distinguish the intact objects and fragments within the same class. In a nutshell, we continue to use all of the features that have proven successful in our prior work (10,22). In addition, we have incorporated several new features in order to help distinguish multiple cell types. Specifically, we have added two new intensity-based features (intensity, texture) to permit object classification. These features were not needed in our earlier single-model work. We have also added six new morphology based features (surface intensity gradient, eccentricity, variance of object radius, surface area, axial depth, and boundary sharing) to help distinguish intact objects from fragments of different types of objects. In addition, we have developed improved estimators (described in the latter section). The newly added features are summarized below.

Intensity I : This is the average intensity of the voxels in the object.

Texture T : This is an indicator of object smoothness. It is measured by the average gradient of the normalized intensity, and can be expressed mathematically by $T = \frac{k}{(V \times I)} \sum \text{gradient}(X)$, where k is a constant scale factor that normalizes the average intensity of all objects to a common value of 128 (middle of the grayscale range of 256 for 8-bit dynamic range images), V is the volume of the object, and X is the voxel intensity.

Surface intensity gradient: This is the average un-normalized intensity gradient of voxels that lie on the object's surface.

Eccentricity: ratio of maximum to minimum distance of the centroid from the object's surface (34).

Variance of object radius: standard deviation of distance from surface voxels to the centroid.

Surface area: number of voxels on the object's surface.

Axial depth: number of optical slices where the object is present.

Boundary sharing: percentage of surface voxels that are shared with neighbors.

The intensity I and texture T are features that mainly help distinguish the glia from neurons (10). Other features distinguish intact objects from fragments, an essential capability for making fragment merging decisions. For the experiments presented here, the design and selection of features were based on our observations of the images and human judgment. While the above features were adequate for the work presented here, we recognize that some other features, as well as more general statistically based feature selection techniques may prove necessary in other applications (34). The core methodology described here still remains valid.

Semi-Automated Model Training

To build the object models, we need a representative collection of manually labeled samples for each object class. From a statistical standpoint, it is desirable to maximize the number of training samples in order to build more accurate models.

However, manual entry of large numbers of samples is tedious and time consuming. Keeping these tradeoffs in mind, we propose an efficient semiautomated approach based on the observation that the vast majority of cells, especially the isolated ones, are indeed segmented correctly by earlier algorithms. The bulk of the errors occur when an algorithm based on a single model faces ambiguities that cannot be resolved with one model alone.

Our methodology is conceptually straightforward—the training images are first subjected to automated processing using a segmentation method that does not require manual training, i.e., the training samples are drawn automatically (4). The results of this preliminary segmentation are edited by the user to correct segmentation and classification errors. Typically, the effort required to edit an automatically generated result is much less compared to manual entry of training samples *de novo*. The editing effort is further reduced by a thoughtfully designed graphical user-interface (GUI). The GUI shown in Figure 6 allows rapid 3D object inspection and editing. The edit window focuses only on a small local area surrounding the object being examined, but with a global reference indicating its location relative to the entire image, which enables rapid 3D visualization independent of the original image size. In addition, the user has the option of sorting the objects in increasing order of confidence in model fitting, so the editing process mostly focuses on low-confidence regions of the image. As the editing process reaches the high-confidence regions, and errors become sufficiently rare, it can be terminated. Integration of the above methods results in a large training sample that allows accurate model construction, although the manual effort expended in generating this sample is very modest.

Once the preliminary segmentation has been edited by the user, an automated classification is carried out using generic object features such as intensity and texture (4). The results are presented to the user to inspect and edit wherever necessary. Although the human user is, in effect, the gold standard for cell classification, some subjectivity inevitably remains. Specifically, we cannot guarantee that all training objects are valid even after the manual editing. To guard against this possibility, we first identify and eliminate outliers in the data using the simple and computationally inexpensive method of α -thresholding (35). Specifically, for each component x_i of our feature vector \mathbf{x} , its mean and standard deviation are calculated, denoted as μ_i and σ_i . We deem any feature values y that satisfies the condition: $y \geq \mu + k\sigma$ or $y \leq \mu - k\sigma$ where k is a weighting coefficient, as outliers. If any one component of the object feature vector satisfies the above condition, the object is deleted from the training set. The remaining objects (inliers) are used to construct the object models.

Core 3D Segmentation Algorithm

The core nuclear segmentation algorithm uses a two-step approach consisting of (i) initial segmentation using a gradient-weighted 3D watershed algorithm (10); and (ii) multiple model-based merging of fragments. Each of these steps is detailed below.

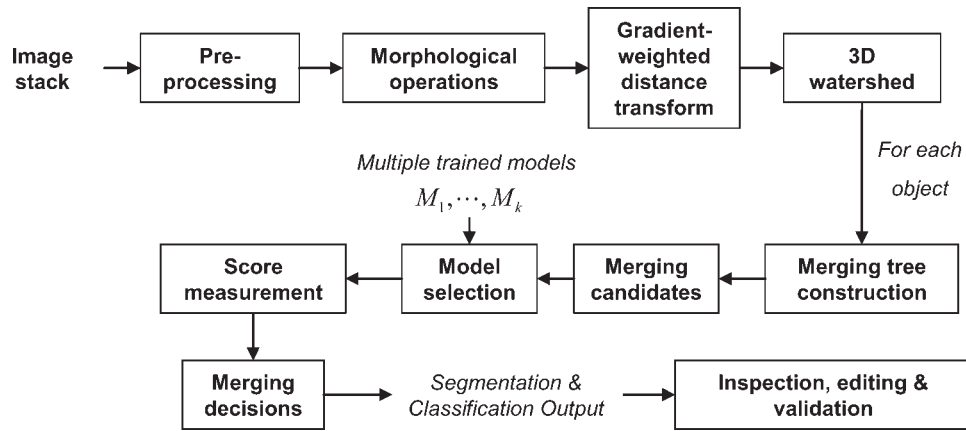


Figure 2. Flow chart illustrating the major steps in the proposed segmentation method.

Initial 3D gradient-weighted watershed segmentation. The watershed algorithm has been widely used for cell nucleus segmentation (36–49). It exploits the properties of chamfer distance transforms computed over binarized images, so is computationally efficient. It is also effective for the vast majority of simple conditions (36). However, it has several known limitations—it typically over-segments the image data and does not take into account image-based cues such as intensity gradients.

In prior work, we proposed a watershed algorithm based on a gradient-weighted distance transform to address the above limitations to a significant extent (10,22). The same method for initial segmentation is adopted in the present work, which is briefly summarized here. First, a pre-processing is carried out including intensity restoration, denoising by median filtering, adaptive thresholding, and morphological cleanup (50,51). The next step fuses the geometric segmentation cues captured by the distance transform (52) and intensity-based cues captured by the intensity gradient (37) into a single “gradient-weighted distance transform.” A 3D watershed segmentation is conducted on the smoothed gradient-weighted distance transform to identify many of the nuclei and fragments of others. This process is summarized as a flow-chart in Figure 2.

Fragment merging. The initial watershed segmentation described above correctly splits many of the multinuclear clusters, but some over-segmentation persists, necessitating an algorithm for merging fragments. Several methods have been proposed in the literature to merge the over-segmented objects using cues such as the intensity gradient at the touching border (37), the size of the objects (45), or a combination of other morphological features (10). Since a large number of merging possibilities exist, the need arises for an efficient algorithm for managing the merging process. In our prior article (22), we described a hierarchical algorithm that searches for optimal combinations of nuclear fragments (22,31,53) guided by a model of the nuclei. In this article, we extend this methodology to accommodate multiple models. The merging procedure (Fig. 2) searches over all the objects generated by initial seg-

mentation. For each object, it builds a merging tree, forms the merging candidates, selects the most likely model, fits the model to the image data, and computes confidence scores. The final step selects the optimal subset of merging decisions.

Merging candidate generation. To efficiently carry out the merging procedure, we build a hierarchical merging tree based on the region adjacency graph (RAG) data structure (54). The details of this procedure are described in our earlier article (22). We provide a brief summary here. As illustrated in Figure 3A, two or more objects are neighbors if they share voxels along a shared boundary. We define the notion of a Root Path Set (RPS) for each node object v at depth d on a tree, denoted $\text{RPS}(d, v)$. For the root node, the RPS is denoted $\text{RPS}(1, 1)$, and is trivial, consisting of only one object, $\{v\}$. For any other node, the RPS consists of all nodes along the path from the root node, e.g., $\text{RPS}(2, 4)$, and $\text{RPS}(3, 3) = \{1, 2, 3\}$. An object u is a neighbor of $\text{RPS}(d, v)$ if u is a neighbor of any one object in $\text{RPS}(d, v)$, and $u \notin \text{RPS}(d, v)$. On the basis of the initial segmentation, we construct the RAG, in which each node is an object, and any two neighboring objects are connected by a link. For an object r that is being considered for merging (e.g., object no. 1 in Fig. 3A), we build a merging tree denoted T_r to obtain all the merging candidates. Initially, T_r only contains the object r as root, and the tree depth $d = 1$. Then, T_r is grown as follows: for each node v at depth d of T_r , find all the neighbors of $\text{RPS}(d, v)$ from the RAG. For each neighbor u , add it to the tree as v ’s children with two exceptions: $u \in \text{RPS}(d, v)$; or $\text{RPS}(d, u) = \text{RPS}(d, v')$, where v' is another node that appeared at depth d . We then increment the depth d by 1, and repeat the above procedure until no more objects can be added to T_r . For example, at depth $d = 2$ of the tree illustrated in Figure 3A, the neighbors of $\text{RPS}(2, 2)$ are objects no. 3, no. 4, and no. 5, so we add them as the children of object no. 2 at $d = 3$. Similarly, the neighbors of $\text{RPS}(2, 3)$ are objects no. 2, no. 4 and no. 5, so we add them to the tree except for object no. 2, since $\text{RPS}\{1, 2, 3\}$ already has appeared by the previous operation, i.e., adding object no. 3 to $\text{RPS}(2, 2)$.

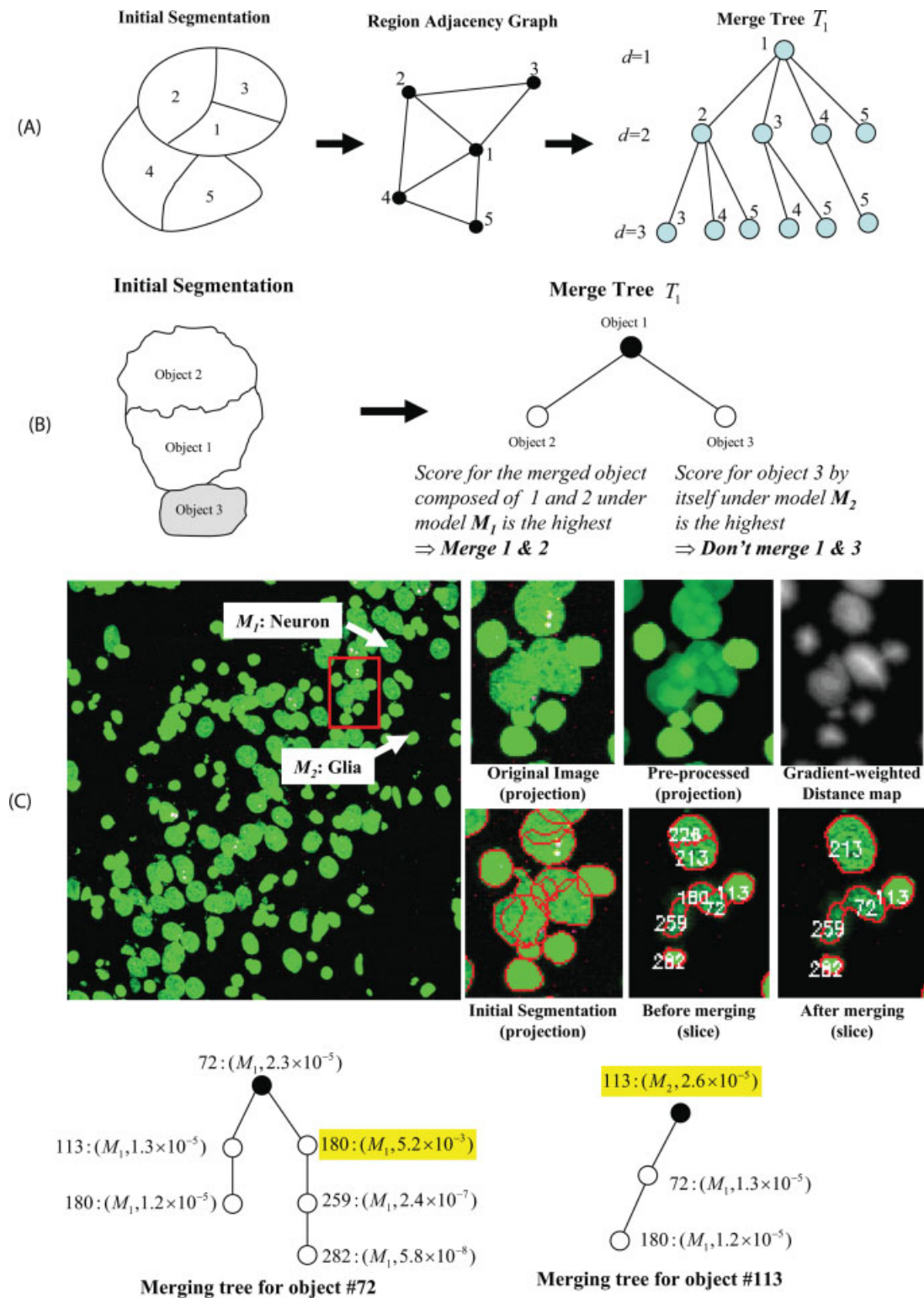


Figure 3. Illustrating multiple-model based object merging. (A) From the initial watershed segmentation, a hierarchical merging tree is constructed using the region adjacency graph. The Root Path Set (RPS) associated with each node (except for the root) consisting of all the objects on the path from the node to the root represents a merging candidate. (B) M_1 and M_2 are object models corresponding to neuronal and glial nuclei, respectively. A confidence score is calculated for each merging candidate. The merging decision is made by choosing the candidate with the highest score. (C) Illustrating the major processing steps for a sample image. The lower panels show sample merging trees for an image containing two object types (Neuron and Glia), where the nodes with highest confidence scores are highlighted. The merging tree for object 72 indicates the need to merge it with object 180 under model M_1 . Object 113 on the other hand, is not merged with any other object, since it has the maximum score by itself under M_2 .

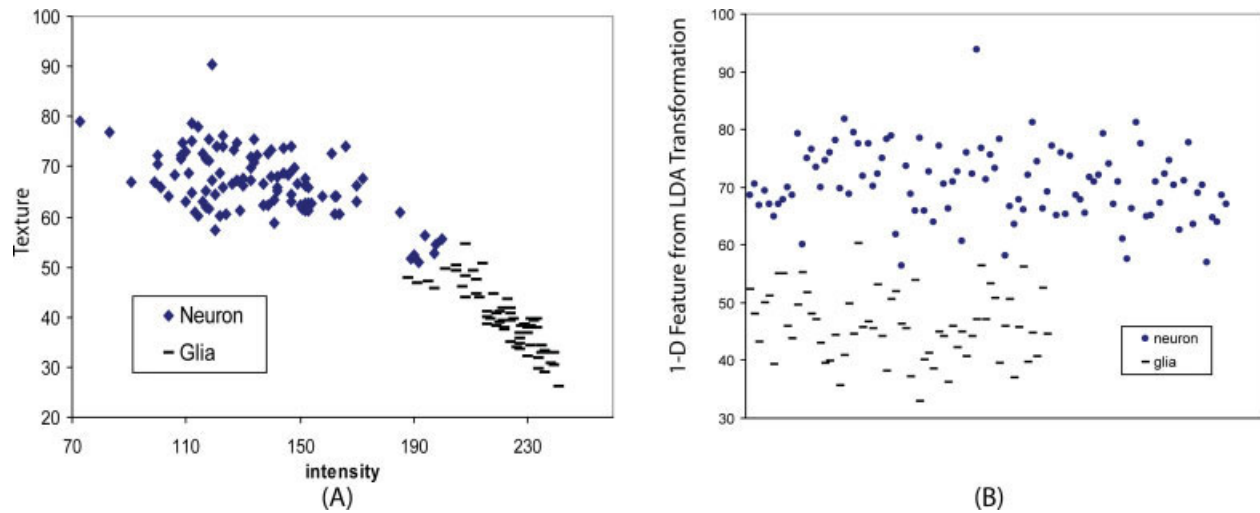


Figure 4. Illustrating Fisher's Linear Discriminant Analysis (LDA) for one image—the image contains two cell types, i.e., neurons and glia. The panel on the left is a scatter plot of the original 2D feature set (intensity and texture measure). The right-hand panel shows the transformed 1D feature (y-axis) generated by LDA. The data points are spread horizontally to permit visualization of individual entries. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

To reduce computation, we limit the combinatorial tree growth by setting an upper bound to the size of the RPS. That is, the total number of voxels contained in all objects in a RPS should not exceed a prespecified threshold. One simple idea is to set the threshold to the maximum number of voxels that an object can possibly contain. Therefore, this size bound can be fixed for any input image, independent of the initial segmentation used.

As a refinement to our previous work (22), we no longer impose a limit on the maximum depth that a merging tree can grow. The reason for the change is that the degree of over-segmentation generally varies widely depending on the initial segmentation, so a universal upper bound on the tree depth is ineffective. It has to be set to the maximum number of fragments that a single object can possibly contain in the initial segmentation, and any upper bound smaller than that will result in missed merging candidates. By removing the tree depth constraint and using the size bound instead, the depth of the merging tree is set adaptively, i.e., a more fragmented object will have a merging tree of greater depth. We have found this to be a better tradeoff that does not miss merging candidates.

Merging criteria. Once the merging candidates are computed, merging decisions can be made. To achieve this end, we need a statistical measure of confidence in a merging decision, i.e., a score measuring the likelihood of an object formed by merging several regions/fragments as being an intact object, such as nucleus. Since we are dealing with multiple types of objects concurrently, an automatic model selection must also be performed concurrently.

Object Model Selection: To classify the objects using the trained models, we adopt Fisher's Linear Discriminant Analysis (LDA) (33). We denote the object models $\mathbf{M} = \{M_1, M_2, \dots, M_K\}$, where K is the total number of object classes in the given image. The feature vector of an object is denoted

$\mathbf{x} = \{x_1, x_2, \dots, x_m\}$, where m is the feature dimension. The basic idea of LDA is to transform the object features into a new space, usually with a lower dimension $d < m$, so that the transformed data among these K classes are as well separated as possible. Specifically, we want to find a matrix \mathbf{W} of size $m \times d$, such that the transformed features $\mathbf{z} = \mathbf{W}^T \mathbf{x}$ are well separated among K classes, but are scattered in a small region within each of these classes. Mathematically, the objective function for finding \mathbf{W} can be expressed as:

$$J(\mathbf{W}) = \frac{|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_W \mathbf{W}|}, \quad (1)$$

where $\mathbf{S}_B = \sum_{i=1}^K N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$ is the between-class scatter matrix, \mathbf{m}_i is the average feature vector for class i , and $\mathbf{m} = \frac{1}{K} \sum_{i=1}^K \mathbf{m}_i$ is the overall mean. The denominator term $\mathbf{S}_W = \sum_{i=1}^K \sum_{j=1}^{N_i} (\mathbf{x}_i^j - \mathbf{m}_i)(\mathbf{x}_i^j - \mathbf{m}_i)^T$ is the within-class scatter matrix, in which \mathbf{x}_i^j is the j th feature vector in class i , and N_i is the total number of sample objects in class i . By maximizing the criterion in Eq. (1), the solution \mathbf{W} is composed of the largest eigenvectors of the matrix $\mathbf{S}_W^{-1} \mathbf{S}_B$, and the new dimension is d . For example, in the case of two classes, Fisher's LDA projects the original object features into a new one-dimensional space. Having computed \mathbf{W} using training samples as above, we can transform any objects into a lower-dimensional space, and classify them using a standard method, such as the Bayesian or k -nearest neighbor classifier (55). Figure 4 shows an example of LDA on one image containing two distinct object types—neurons and glia. The selected 2D features are intensity and texture. The transformed 1D feature provides class separation that is comparable to the 2D case.

Merging Confidence Calculation: using the classification procedure described above, we can classify an object—the class is denoted c . To measure the confidence score of a merging candidate \mathbf{x} , we compute the probability that \mathbf{x} belongs to the

object model M_c using a Bayesian formula. Note that the feature vector \mathbf{x} used here can be different from the feature vector used for classification, since the emphasis at this stage is on distinguishing fragments from intact nuclei. To simplify our discussion, we use the same notation throughout the paper. Since the variance of our selected object features varies considerably, we first normalize them such that every dimension has zero mean and unit standard deviation using the formula $\mathbf{x} = (\mathbf{x} - \mathbf{u})/\boldsymbol{\sigma}$, where \mathbf{u} and $\boldsymbol{\sigma}$ denote the mean and standard deviation, respectively (for convenience, we use the same notation before and after feature normalization). At this stage, Principal Component Analysis (PCA) can also be applied to remove the dependences among these features, and reduce the dimensionality. Let $\mathbf{x} = (x_1, x_2, \dots, x_r)$ ($r < m$) denote the new transformed feature vector, and c is the object class. Based on Bayes' rule (33), the merging confidence score can be expressed as:

$$p(M_c|\mathbf{x}) \propto p(\mathbf{x}|M_c)p(M_c), \quad (2)$$

where $p(M_c)$ is the a priori probability of model M_c and $p(\mathbf{x}|M_c)$ is the class-conditional probability. The prior $p(\mathbf{x}|M_c)$ is preset based on the known relative abundance of the object classes.

To calculate $p(\mathbf{x}|M_c)$, we previously used a parametric estimate assuming a multivariate Gaussian distribution (10,22,56), and estimated the distribution parameters from the training samples. In the present work, we eliminate the Gaussian assumption to permit greater generality in the modeling. Specifically, we adopt a non-parametric Parzen window method for estimating the probability density (57), as suggested by some authors (58,59). For feature vector \mathbf{x} and a sample size N , the estimated density function is given by:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^N \phi(\mathbf{x} - \mathbf{x}_j, h)$$

where \mathbf{x}_j is j th sample, $\phi(\cdot)$ is the Parzen window function, and h is the window width. It has been shown (57) that $p(\mathbf{x})$ converges to the true density function as the sample size grows, i.e., $N \rightarrow \infty$, if the window functions $\phi(\cdot)$ and h are properly chosen. We use the smooth Gaussian window in this work, given by:

$$\phi(z, h) = \frac{1}{\sqrt{(2\pi h^2)^m |\Sigma|}} \exp\left(-\frac{z^T \Sigma^{-1} z}{2h^2}\right)$$

where Σ denotes the covariance matrix of m -dimensional random variable z . The window size h plays an important role in the estimate. When h is small, the influence of each training sample is limited to a small region. When h is larger, there is more overlap of the windows and the estimate is smoother. In this work, we set h to the distance from \mathbf{x} to the k th nearest neighbor among all the sample points (32). Let $d_1, d_2, d_k, \dots, d_N$ denote the distances between \mathbf{x} and the training samples in the increasing order, then $h = d_k$. To reduce computation, we ignore samples whose distance $d > 3h$ during density calculation. In summary, the overall posterior probability (2) can be written as follows:

$$p(M_c|\mathbf{x}) = \frac{1}{N_c \sqrt{(2\pi h^2)^r |\Sigma_c|}} \times \sum_{j=1}^{N_c} \exp\left(-\frac{(\mathbf{x} - \mathbf{x}_c^j)^T \Sigma_c^{-1} (\mathbf{x} - \mathbf{x}_c^j)}{2h^2}\right) p(M_c), \quad (3)$$

where N_c is the sample size of class c , \mathbf{x}_c^j is the j th feature vector in class c , and Σ_c is the covariance matrix of class c . The above probability estimate reflects the confidence of the object \mathbf{x} being intact in its class, and it will be used as a score for region/fragment merging. Figure 3 shows a detailed illustration of the above based on actual data.

Merging Decision: Each node of the merging tree T_r in which each root path set $RPS(d, v)$ is a merging candidate for root node r . The complete set of J merging candidates for the tree is $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_j\}$ where $\mathbf{y}_j = (v_1^j, \dots, v_d^j)$ is a candidate formed by merging d fragments. For each candidate \mathbf{y}_j , we compute a confidence score $p(M_c|\mathbf{y}_j)$ using Eq. (3), assuming the class c_j . We decide to accept this proposed merging if the score satisfies two conditions: (i) it is the highest among all candidate objects, i.e., $p(M_c|\mathbf{y}_j) = \max_{i=\{1, \dots, j\}} p(M_c|\mathbf{y}_i)$; and (ii) is greater than the score of all its constituent fragments, i.e., $p(M_c|\mathbf{y}_j) > \max_{i=\{1, \dots, d\}} p(M_c|\mathbf{v}_i^j)$.

The above merging procedure terminates after applying the merging step to each object generated by the initial segmentation. A flowchart description of this procedure is shown in Figure 2. Upon completion, the object classification data is also available as a valuable addendum to the output. This is often an important problem to solve in its own right. In our experiments, both the segmentation and classification results are presented in the GUI for human observers to inspect and validate.

Batch Processing

In recent years, biological studies increasingly require the analysis of large sets of image data, e.g., hundreds of images drawn from a single experiment. These images are often recorded in batch mode, i.e., similar specimen, staining and imaging settings. All the procedures described above can be conveniently combined for automated batch processing. Once the object models have been trained from a set of typical examples drawn across a batch of images, analysis of all images in the batch can be carried out automatically without human intervention. This not only speeds up the image analysis and increases the throughput, but also facilitates postsegmentation statistical analysis, since the results of all the images in one batch can be compiled and stored together, and readily imported as a whole into a spreadsheet tool such as Microsoft Excel. The experimental results presented in the next section were generated by this batch processing method.

RESULTS

The above algorithms and graphical user interface (GUI) were implemented using the IDL software platform (Interactive Data Language, RSI Boulder, CO) and C++. All the results presented here were generated fully automatically in

batch processing mode after object models were trained once for the entire batch. For images prepared using *in situ* hybridization, ~20 μm thick tissues from the following regions of the rat brain: CA1, CA3, barrel cortex, gustatory cortex, and dentate gyrus. Five batches of 3D confocal microscope image stacks were used for these tests. Details of tissue preparation and *in situ* hybridizations are described in our earlier paper (4). Images were acquired using a Zeiss LSM 510 Meta NLO confocal microscope equipped with the following lasers: Argon multi-line 458/477/488/514, 200 mW gas laser (LASOS Lasertechnik GmbH, Jena, Germany), Helium-Neon 543, 5 mW diode laser (LASOS), Helium-Neon 633, 15 mW diode laser (LASOS) and a MIRA 900 Titanium-Sapphire pulsed infrared laser, 700–1,000 nm tunable (Coherent, Santa Clara, CA). A 40 \times Plan NeoFluar objective with a 1.3 NA and working distance of 0.12 mm was used. Typical voxel sizes along the *x*-, *y*- and *z*- directions were 0.64, 0.64, and 0.75 μm , respectively. At the imaging wavelengths (488, 568, and 647 nm) and for the lens used the chromatic aberrations are not substantial enough to warrant corrections (60, 61).

There are two major types of objects in these images: neuronal cell nuclei and glial cell nuclei. For each batch, one typical image stack was picked for training object models. The training was carried out in a semi-automated fashion, i.e., automated generation of object samples aided by human inspection and editing where necessary. Once the object models are obtained, they are used for all other images in the entire batch. The entire segmentation procedure is performed in a fully automated fashion. As described in Fragment Merging section, the object features were chosen differently for classification (or model selection) and confidence score calculation for merging, depending on the images in question. The following are two feature sets we selected: three features (intensity, texture and volume) for classifying neuron and glia, and the scatter plot shown in Figure 1(B) justifies the use of these features; 10 features (volume, convexity, shape, eccentricity, bending energy, variance of object radius, surface intensity gradient, surface area, axial depth, and boundary sharing) for object merging.

For validating the automated results, the segmentation and classification was carried out manually by a group of three to five trained observers independently. These observers then arrived at a consensus to resolve interobserver differences. The consensus results were compared with the fully automated results. Table 1 summarizes the segmentation and classification performance data, and provides a detailed breakdown of the types of errors that were found. The overall segmentation accuracy was $(93.7 \pm 3.3)\%$, and the classification accuracy was $(93.5 \pm 5.7)\%$. The most common segmentation error was the presence of multi-object clusters (6.3%). The most common classification error was missed glia (5.9%). Very few objects were fragmented (1.2%) or misclassified (0.3%).

To evaluate our method for more than two cell types, nuclear labeling was performed on thicker (~100 μm) tissue slices from the CA1 region using CyQuant (Invitrogen, Carlsbad, CA). This provided a broader sampling of cell types including neurons, glia, and vascular cells. Figure 5A shows an

example confocal image as a maximum-intensity projection. The specimen preparation and imaging protocols are described elsewhere (55). Panel B shows the results of automatic classification (model selection). Panel C is a segmentation that was based on assuming a single model for nuclei. The panel on the right was based on three models—examples of neuronal, glial, and vascular cells were provided as training examples. The arrows indicate specific objects for which the results on the right are clearly superior. For this dataset with 142 nuclei, the single model method had 10 segmentation errors (three under-segmentation cases & seven over-segmentation cases). For the same dataset, the multiple-model method had seven errors (two under-segmentations, five over-segmentations). Further studies to validate our methodology on a large scale with three or more models are currently underway, and will be reported separately.

For segmentation, the improvement mainly results from these factors: (1) use of multiple models; (2) more accurate nuclear classification leading to more specific object modeling; (3) more effective feature extraction compared to our prior work; (4) more reliable confidence scoring is employed for merging, i.e., adaptive Parzen window probability estimate is used instead of parametric probability estimator based on assuming a multivariate Gaussian distribution on the object features.

DISCUSSION

The proposed multi-model approach offers several benefits. It improves the segmentation by using a more specifically tuned model for each object type, and achieves an object classification at the same time. This methodology is also more elegant and general compared to our prior work (22). Our algorithm does not have an innate upper bound on the number of models that can be used. The only requirement is that the user must specify examples of each object class, and that sufficient numbers of examples of objects of each class are available in the image data for computing reliable models. Finally, the models must be distinct enough to be classified with an acceptably low error rate.

The performance of the approach depends on the effectiveness and accuracy of the object modeling, which is ordinarily a labor-intensive task. The semiautomated model generation process that we propose requires a minimum of manual supervision and editing. The number of models must be selected meaningfully—they must reflect the actual cellular morphological diversity in the image data. Having more than the necessary number of models will only raise the dimensionality of the classification problem and the computational cost, without providing any benefits from a classification standpoint. On the other hand, the segmentation performance is largely unaffected. In other words, our methodology is graceful and does not have a drastic failure mode from a segmentation standpoint.

The effectiveness of models depends on the choice and design of image-based features. The selected features must not only be capable of distinguishing the fragments and intact objects (needed for calculating the merging confidence scores),

Table 1. Performance of the multimodel segmentation and classification algorithm on 17 confocal image stacks drawn from three sub-regions [CA1, CA3 and Dentate Gyrus (DG)] of the rat hippocampus and two different cortical regions [Barrel Cortex (BC) and Gustatory Cortex (GC)]

IMAGE NO.	BRAIN REGION	NUMBER OF OBJECTS FROM INITIAL SEGMENTATION	NUMBER OF MERGING OPERATIONS	FINAL NUMBER OF OBJECTS	NUMBER OF MULTIOBJECT CLUSTERS	NUMBER OF FRAGMENTED OBJECTS	OVERALL SEGMENTATION ACCURACY (%)	NUMBER OF DETECTED GLIA	NUMBER OF MISSED GLIA	NUMBER OF MIS-CLASSIFICATIONS	OVERALL CLASSIFICATION ACCURACY (%)
1	CA1	403	106	95	3	2	94.7	101	1	0	99.0
2	CA1	352	76	82	2	0	97.6	88	2	1	96.6
3	CA1	322	57	75	4	2	92.0	95	1	0	98.9
4	CA3	316	73	67	0	0	100.0	128	1	0	99.2
5	CA3	273	73	56	1	1	96.4	100	3	0	97.0
6	CA3	302	85	59	0	1	98.3	117	3	1	96.6
7	BC	382	49	79	3	0	96.2	155	10	0	93.5
8	BC	370	40	80	3	0	96.3	129	4	1	96.1
9	BC	451	84	102	10	1	89.2	125	16	0	87.2
10	GC	292	55	70	5	0	92.9	85	8	0	90.6
11	GC	333	36	72	6	0	91.7	101	12	0	88.1
12	GC	489	86	125	10	1	91.2	122	16	0	86.9
13	GC	417	46	86	9	0	89.5	147	8	0	94.6
14	GC	408	73	108	6	1	93.5	97	11	1	87.6
15	DG	494	71	191	11	2	93.2	72	0	0	100.0
16	DG	467	99	199	14	5	90.5	42	1	0	97.6
17	DG	550	123	249	20	5	90.0	25	4	1	80.0
Average		389.5	72.5	105.6	6.3	1.2	93.7 ± 3.3	101.7	5.9	0.3	93.5 ± 5.7

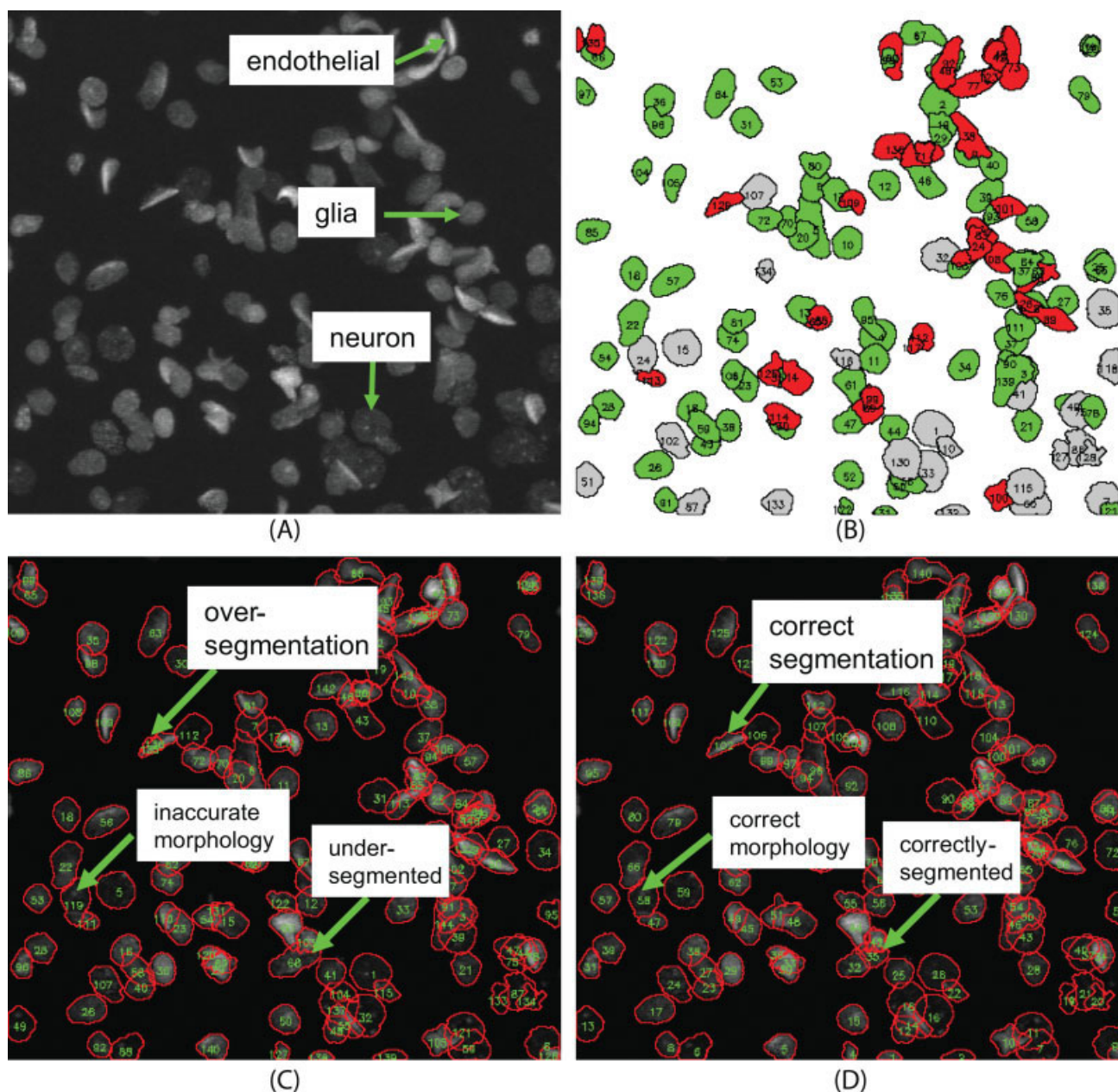


Figure 5. Example illustrating improved segmentation of a hippocampal dataset (panel A) when three models are employed (panel D) compared to only one model (panel C). The three models correspond to neurons, glia, and vascular cell nuclei, shown in gray, green and red respectively in Panel B.

but also distinguishing multiple object types (needed for object classification). As noted earlier, it is advantageous to use different sets of features for these two aspects, instead of one universal set of features.

The proposed method does have known weaknesses that will be addressed in future work. For example, if the fragment of one type of object is indistinguishable from another type of intact object in the image, the approach will fail since the fragment will be identified as an intact object of the misclassified type, and won't be merged. If this turns out to be the case, we need to exploit and extract additional discriminative features,

and use a separate feature set for object classification if necessary. As noted earlier, in this work, we used a different feature set consisting of volume and intensity-based features to classify neurons and glia, which is independent of the morphology-based features used for computing merging confidence score. Another known weakness of our method is that it assumes that only one fluorescent channel shows the nuclei. If the confocal stack contains additional channels that distinguish the classes of objects in question, such as cell-type specific fluorescent markers that are increasingly available and often employed in many studies, they can be exploited to provide

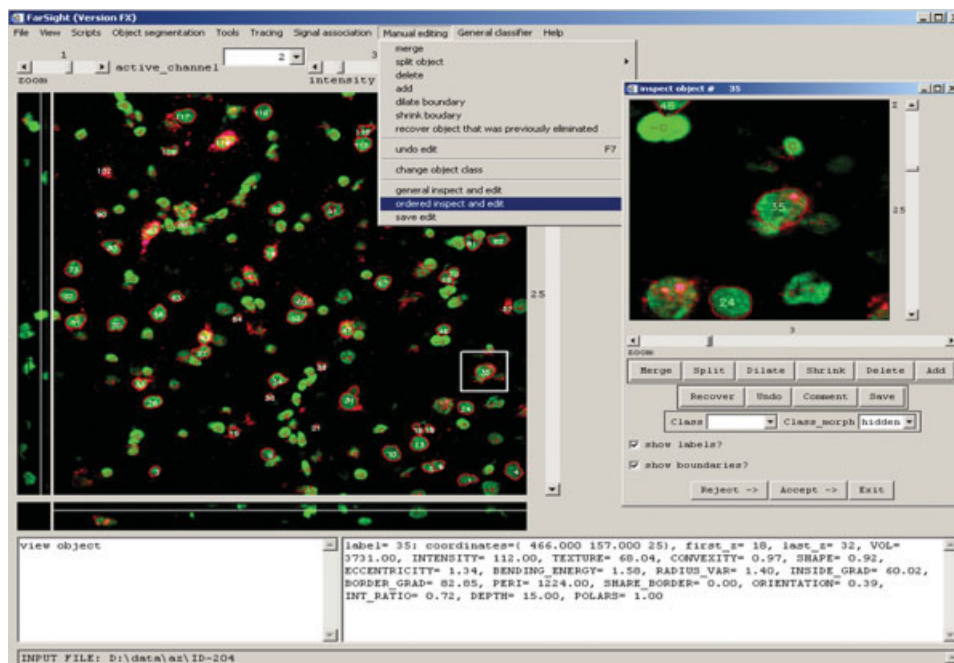


Figure 6. The graphical user interface (GUI) for efficiently inspecting and editing segmentation and classification. The window on the right is used for specifying training samples, and for validating the automated segmentation and classification results. It displays segmentations as outlines, object ID's, and codes to indicate automated actions ("G" implies a glial cell classification, "X" indicates a deleted cell). The features of the selected cell are displayed on the lower right. The user can accept, reject, or edit the automated results. By visiting the results in increasing order of segmentation confidence, the user can choose to exit the system once an acceptable confidence is reached. The manual effort is proportional to the number of errors rather than the total number of objects.

additional discrimination. Happily, our overall methodology is still applicable in this case. The model then contains additional features that capture the cell type information from the additional channels.

Handling partially imaged nuclei, i.e., those on the border of the image volume is a challenge, and needs special care. Any partially presented objects in the image are generally not suitable for model fitting when models were derived from intact objects. Practically, partial objects are fragments. In this work, we excluded them from consideration, so they do not contribute to the final error rate reported here. In other words, all the partial objects were removed by an automatic procedure before prompting for user for inspection and validation. In principle, the problem of partial objects can be potentially overcome by deliberately modeling the partial objects as a separate class, and treating them as a separate object class in addition to those intact object classes. Unfortunately, the intraclass variation for partial objects is high, so was not considered further by us.

One assumption we made about initial segmentation is that there are no under-segmentations (multinuclear clusters) remaining. Although rare, a few clusters may exist. In applications for which such errors become less rare, a procedure for selective object splitting based on our calculated score can be employed. Although the idea of split and merge is not novel, we avoided the splitting operation in our work since it was not necessary, and would only add to the computational load.

During the batch processing, whenever the analysis of an image is completed, its segmentation and classification results can be utilized to augment the training data set. Consequently, the object models can be updated and refined adaptively using the greater number of available samples. This will yield further improvement on the accuracy, but require additional computation for the model refinement.

LITERATURE CITED

1. Roysam B, Lin G, Abdul-Karim M-A, Al-Kofahi O, Al-Kofahi K, Shain W, Szarowski DH, Turner JN. Automated 3D Image Analysis Methods for Confocal Microscopy. In: Pawley J, editor. Handbook of Confocal Microscopy, 3rd ed. New York: Springer; 2006. pp 316–337.
2. Russ JC. Computer-assisted microscopy: The Measurement and Analysis of Images. New York: Plenum; 1990. xii, 453 pp.
3. Lin G, Bjornsson CS, Smith KL, Abdul-Karim MA, Turner JN, Shain W, Roysam B. Automated image analysis methods for 3D quantification of the neurovascular unit from multi-channel confocal microscope images. *Cytometry Part A* 2005;66A: 9–23.
4. Chawla MK, Lin G, Olson K, Vazdarjanova A, Burke SN, McNaughton BL, Worley PF, Guzowski JF, Roysam B, Barnes CA. 3D-catFISH: A system for automated quantitative three-dimensional compartmental analysis of temporal gene transcription activity imaged by fluorescence in situ hybridization. *J Neurosci Methods* 2004;139:13–24.
5. Ortiz de Solorzano C, Santos A, Vallcorba I, Garcia-Sagredo J, del Pozo F. Automated FISH spot counting in interphase nuclei: Statistical validation and data correction. *Cytometry A* 1998;31:93–99.
6. Nilsson B, Heyden A. Segmentation of complex cell clusters in microscopic images: Application to bone marrow samples. *Cytometry Part A* 2005;66A:24–31.
7. Baggett D, Nakaya M, McAuliffe M, Yamaguchi TP, Lockett S. Whole cell segmentation in solid tissue sections. *Cytometry Part A* 2005;67A:137–143.
8. Cong G, Parvin B. Model-based segmentation of nuclei. *Pattern Recogn* 2000;33: 1383–1393.
9. Lee K-M, Street WN. Model-based detection, segmentation, and classification for image analysis using on-line shape learning. *Mach Vis Appl* 2003;13:222–233.

10. 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 Part A* 2003;56A:23–36.
11. Bernard R, Kanduser M, Pernus F. Model-based automated detection of mammalian cell colonies. *Phys Med Biol* 2001;46:3061–3072.
12. Ezquerro N, Mullick R. Knowledge-guided segmentation of 3D imagery. *Graph Models Image Process* 1996;58:510–523.
13. Chassery JM, Garbay C. An iterative segmentation method based on a contextual color and shape criterion. *IEEE Trans Pattern Anal Mach Intell* 1984;6:794–800.
14. Mitchell SC, Bosch JG, Lelieveldt BPF, van der Geest RJ, Reiber JHC, Sonka M. 3D active appearance models: Segmentation of cardiac MR and ultrasound images. *IEEE Trans Med Imaging* 2002;21:1167–1178.
15. Liu L, Sclaroff S. Region segmentation via deformable model-guided split and merge. Boston University Computer Science Technical Report No. (2000-24). Boston, MA: Boston University; 2000.
16. Ghanei A, Soltanian-Zadeh H. A discrete curvature-based deformable surface model with application to segmentation of volumetric images. *IEEE Trans Inf Technol Biomed* 2002;6:285–295.
17. McInerney T, Terzopoulos D. Deformable models in medical image analysis: A survey. *Med Image Anal* 1996;1:91–108.
18. Staib LH, Duncan JS. Model-based deformable surface finding for medical images. *IEEE Trans Med Imaging* 1996;15:720–731.
19. Osher S, Sethian J. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation. *J Comput Phys* 1988;79:12–49.
20. Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell* 2000;22:888–905.
21. Wang S, Siskind JM. Image segmentation with ratio cut. *IEEE Trans Pattern Anal Mach Intell* 2003;25:675–690.
22. Lin G, Chawla MK, Olson K, Guzowski JF, Barnes CA, Roysam B. Hierarchical, model-based merging of multiple fragments for 3D segmentation of nuclei. *Cytometry Part A* 2005;63A:20–33.
23. Guzowski JF, Timlin JA, Roysam B, McNaughton BL, Worley PF, Barnes CA. Mapping behaviorally relevant neural circuits with immediate-early gene expression. *Curr Opin Neurobiol* 2005;15:599–606.
24. Zucker SW. Region growing: Childhood and adolescence. *Comput Graph Image Process* 1976;5:382–399.
25. Adams R, Bischof L. Seeded region growing. *IEEE Trans Pattern Anal Mach Intell* 1994;16:641–647.
26. Chang YL, Li X. Adaptive image region-growing. *IEEE Trans Image Process* 1994;3:868–872.
27. Hojjatoleslami SA, Kittler J. Region growing: A new approach. *IEEE Trans Image Process* 1998;7:1079–1084.
28. Horowitz SL, Pavlidis T. Picture Segmentation by a Directed Split and Merge Procedure. *Proceedings of the International Joint Conference on Pattern Recognition (ICPR)*. 1974. pp 424–433.
29. Haralick RM, Shapiro LG. Image segmentation techniques. *Comput Vis, Graph Image Process* 1985;29:100–132.
30. Roysam B, Ancin H, Bhattacharjya AK, Chisti MA, Seegal R, Turner JN. Algorithms for automated cell population analysis in thick specimens from 3D confocal fluorescence microscopy data. *J Microsc* 1994;173:115–126.
31. Wu X. Adaptive split-and-merge segmentation based on piecewise least-square approximation. *IEEE Trans Pattern Anal Mach Intell* 1993;15:808–815.
32. Alpaydin E. *Introduction to Machine Learning*. Cambridge, MA: MIT; 2004.
33. Duda RO, Hart PE, Stork DG. *Pattern Classification*. New York: Wiley; 2001. xx, 654 pp.
34. Theodoridis S, Koutroumbas K. *Pattern Recognition*. San Diego: Academic Press; 1999. 625 pp.
35. Sahoo P, Soltani S, Wong A, Chen Y. A survey of thresholding techniques. *Comput Vis Graph Image Process* 1988;41:233–260.
36. Ancin H, Roysam B, Dufresne TE, Chestnut MM, Ridder GM, Szarowski DH, Turner JN. Advances in automated 3D image analyses of cell populations imaged by confocal microscopy. *Cytometry* 1996;25:221–234.
37. Wählby C, Sintorn I-M, Erlandsson F, Borgefors G, Bengtsson E. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *J Microsc* 2004;215:67–76.
38. Shafarenko L, Petrou M, Kittler J. Automatic watershed segmentation of randomly textured color images. *IEEE Trans Image Process* 1997;6:1530–1544.
39. Vincent L. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Trans Image Process* 1993;2:176–201.
40. Beucher S. Watersheds of functions and picture segmentation. *IEEE Int Conf Acoust Speech Signal Process* 1982:1928–1931.
41. Beucher S. The watershed transformation applied to image segmentation. *Scan Microsc* 1992;6:299–314.
42. Beucher S, Meyer F. *The Morphological Approach to Segmentation: The Watershed Transformation, Mathematical Morphology and Image Processing*. New York: Marcel Dekker; 1993.
43. Najman L, Schmitt M. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Trans Pattern Anal Mach Intell* 1996;18:1163–1173.
44. Wählby C, Bengtsson E. Segmentation of cell nuclei in tissue by combining seeded watersheds with gradient information. In: Bigun J, Gustavsson T, editors. *Lecture Notes in Computer Science*, Vol. 2749. Berlin: Springer-Verlag; 2003. pp 408–414.
45. Adiga PSU, Chaudhuri BB. Efficient cell segmentation tool for confocal microscopy tissue images and quantitative evaluation of FISH signals. *Microsc Res Tech* 1999;44:49–68.
46. Pavlidis T, Liow Y-T. Integrating region growing and edge detection. *IEEE Trans Pattern Anal Mach Intell* 1990;12:225–233.
47. Vincent L, Soille P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans Pattern Anal Mach Intell* 1991;13:583–598.
48. Malpica N, Ortiz de Solorzano C, Vaguero J, Santos A, Vallcorba I, Garcia-Sagredo J, del Pozo F. Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry A* 1997;28:289–297.
49. Ortiz de Solorzano C, Garcia Rodriguez E, Jones A, Pinkel D, Gray J, Sudar D, Lockett S. Segmentation of confocal microscope images of cell nuclei in thick tissue sections. *J Microsc* 1999;193:212–226.
50. Serra JP, Soille P. *Mathematical Morphology and its Applications to Image Processing*. Dordrecht: Kluwer Academic Publishers; 1994. ix, 383 pp.
51. Bovik AC, Aggarwal SJ, Merchant F, Kim NH, Diller KR. Automatic area and volume measurements from digital biomedical images. In: Häder D-P, editor. *Image Analysis: Methods and Applications*. Boca Raton, FL: CRC; 2001. pp 23–64.
52. Borgefors G. Distance transformations in digital images. *Comput Vis Graph Image Process* 1986;34:344–371.
53. Ballard DH, Brown CM. *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall; 1982. 523 pp.
54. Sonka M, Hlavac V, Boyle R. *Image Processing, Analysis, and Machine Vision*. London: Chapman & Hall Computing; 1993. xix, 555 pp.
55. Lin G, Al-Kofahi Y, Tyrrell JA, Björnsson C, Shain W, Roysam B. Automated 3-D quantification of brain tissue at the cellular scale from multi-parameter confocal microscopy images. In: *Proceedings of 2007 International Symposium on Biomedical Imaging: From Nano to Macro*, Washington, DC, April 2007.
56. Hair JF, Tatham RL, Anderson RE, Black W. *Multivariate Data Analysis*. Upper Saddle River, NJ: Pearson Prentice Hall; 2005.
57. Parzen E. On the estimation of a probability density function and mode. *Ann Math Stat* 1962;33:1065–1076.
58. Kwak N, Choi C-H. Input feature selection by mutual information based on Parzen window. *IEEE Trans Pattern Anal Mach Intell* 2002;24:1667–1671.
59. Peng H, Long F, Ding C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 2005;27:1226–1238.
60. Hibbs A, MacDonald G, Garsha K. *Practical Confocal Microscopy*. In: Pawley JB, editor. *Handbook of Biological Confocal Microscopy*, 3rd ed. New York: Springer, 2006. pp 650–670.
61. Zucker RM. Quality assessment of confocal microscopy slide based systems: Performance. *Cytometry Part A* 2006;69A:659–676.