Active Mask Segmentation of Fluorescence Microscope Images

Gowri Srinivasa, *Member, IEEE*, Matthew C. Fickus, *Member, IEEE*, Yusong Guo, Adam D. Linstedt, and Jelena Kovačević, *Fellow, IEEE*

***Abstract*—We propose a new active mask algorithm for the segmentation of fluorescence microscope images of punctate patterns. It combines the (a) flexibility offered by active-contour methods, (b) speed offered by multiresolution methods, (c) smoothing offered by multiscale methods, and (d) statistical modeling offered by region-growing methods into a fast and accurate segmentation tool. The framework moves from the idea of the "contour" to that of "inside and outside," or masks, allowing for easy multidimensional segmentation. It adapts to the topology of the image through the use of multiple masks. The algorithm is almost invariant under initialization, allowing for random initialization, and uses a few easily tunable parameters. Experiments show that the active mask algorithm matches the ground truth well and outperforms the algorithm widely used in fluorescence microscopy, seeded watershed, both qualitatively, as well as quantitatively.**

***Index Terms*—Active contours, active masks, cellular automata, fluorescence microscope, multiresolution, multiscale, segmentation.**

I. Introduction

BIOLOGY has undergone a revolution in the past decade in the way it can examine and analyze processes in live cells, due largely to advances in fluorescence microscopy [2], [3]. Biologists can now collect images across a span of resolutions and modalities, multiple time points, as well as multiple channels marking different structures, leading to enormous quantities of image data. As visual processing of such high-dimensional and large data sets is time consuming and cumbersome, automated processing is becoming a necessity. Segmentation is often the first step after acquisition, as it enables biologists to access specific structures of interest (for example, individual cells) within an image.

A. Goal

The acquisition of fluorescence microscope images is done by introducing (into the cells) nontoxic fluorescent probes capable of tagging proteins or molecules of interest. A fluorescent microscope images these cells by illuminating the specimen with light of a specific wavelength, exciting the fluorescent probes to emit light of a longer wavelength; a CCD camera records photon emissions resulting in a digital image. As only some parts of the sample are tagged and the tagging is not uniform, the resulting image looks like a distribution of bright dots on a dark background, a *punctate* pattern, as in Fig. 1. We focus on images in which such patterns represent individual cells in a multicell specimen. Thus, we aim to:

Develop an algorithm for the segmentation of fluorescence microscope images of punctate patterns.

B. Segmentation in Fluorescence Microscopy

As different imaging modalities present highly specific challenges, a solution developed for one problem often cannot be applied to another without compromising segmentation accuracy [4]. This specificity has resulted in a vast body of literature; we briefly mention only those works which directly relate to fluorescence microscopy.

In the fluorescence microscope image segmentation community, two algorithms are in wide use: the Voronoi [5], [6] and the seeded watershed algorithms [7], [8]. The watershed algorithm was designed as a "universal segmenter," and, as it often results in splits or merges, various modifications have been proposed over the years [9], [10]. Its performance is critically dependent on the initial seeds, and, unless the background is seeded, it does not produce tight contours around the cells, thus including a significant portion of the background with the cell (similarly for the Voronoi algorithm).

Region-based segmentation methods could be suited to segment fluorescence microscope images due to the relative homogeneity of statistical properties of the foreground and background in such images. Although there exists a number of region-growing segmentation methods designed by the image processing and computer vision communities [11]–[15], these methods have not been applied directly to fluorescence microscope cell images.

Manuscript received May 27, 2008; revised March 19, 2009. First published April 17, 2009; current version published July 10, 2009. This work was supported in part by the National Science Foundation through awards DMS-0405376, ITR-EF-0331657, in part by the National Institutes of Health through awards R03-EB008870, R01-GM-56779, and in part by the PA State Tobacco Settlement, Kamlet-Smith Bioinformatics Grant. A preliminary version of this paper was presented at the IEEE International Symposium on Biomedical Imaging, Paris, France, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitri Van De Ville.

G. Srinivasa is with the Department of Information Science and Engineering and the Center for Pattern Recognition, PES School of Engineering, Bangalore, India.

M. C. Fickus is with the Department of Mathematics and Statistics, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH 45433 USA.

Y. Guo and A. D. Linstedt are with the Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

J. Kovačević is with the Departments of Biomedical Engineering, Electrical and Computer Engineering, and the Center for Bioimage Informatics, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2021081

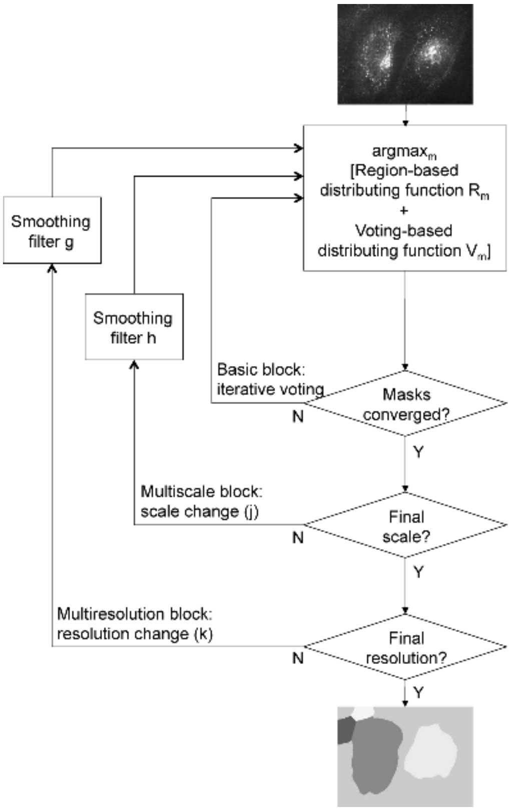

argmax$_m$
[Region-based
distributing function R$_m$
+
Voting-based
distributing function V$_m$]
Smoothing filter g
Smoothing filter h
Basic block: iterative voting
Masks converged?
N
Y
Multiscale block: scale change (j)
Final scale?
N
Y
Multiresolution block: resolution change (k)
Final resolution?
N
Y

Fig. 1. Active mask algorithm. In the basic block, the region-based and voting-based distributing functions work in concert to first separate foreground from background and then iteratively separate masks in the foreground. In the multiscale block, we look at local areas at multiple scales to allow the algorithm to discern cell boundaries and escape local optima. In the multiresolution block, we start at a coarse scale to obtain a fast first segmentation, and then change the scale to refine the result.

Active contour algorithms are considered state-of-the-art in medical image segmentation and have recently been used for the segmentation of biological images [16]–[18]. Inspired by the segmentation accuracy produced by the active-contour algorithms, we modified an existing image statistics-based active contour algorithm, STACS [19], to fluorescence microscope images. To be precise, we kept STACS's region-based and curvature-based forces, and imposed topology preservation to keep the number of cells in the image constant. The resulting algorithm, TPSTACS [18], performed extremely well, albeit slowly. Aiming to keep the flexibility of the active-contour based algorithms, which allow for the choice and/or design of forces suited to a given application, we followed it by its *multiresolution* version, whereby we achieved an order-of-magnitude increase in speed by segmenting the coarse version first, followed by refinements at finer levels. We finally added a *multiscale* flavor, to allow for both the use of convolution-based processing to increase the speed, as well as for "smoothing to connect the dots" to discern boundaries of cells [20].

C. Idea

In our previous attempts to segment fluorescence microscope images, several issues and questions repeatedly arose: (a) What is the "contour" in a digital image? (b) Updating the level set function in active-contour algorithms is inefficient and slow. (c) How do we reconstruct the level set function in the multiresolution version? (d) Updating in large increments in the multiscale version does not preserve topology. To address these issues, we seek:

An algorithm that combines the:
(a) flexibility offered by active-contour methods;
(b) speed offered by multiresolution methods;
(c) smoothing offered by multiscale methods;
(d) statistical modeling offered by region-growing methods.

We work on fluorescence microscope images of punctate patterns, and assume that: (a) the statistical properties of the foreground (cell) and background are distinct and relatively uniform; (b) the foreground is bright, while the background is dark. The first assumption is crucial, the second not at all; one can easily modify the algorithm should the situation be reversed in another modality (such as brightfield microscopy). Thus, in this paper, we are essentially looking for two different statistical models in the image (foreground and background). We note, however, that the techniques presented here may be generalized to the case of more models (by adding additional region-based distributing functions to the foreground voting, discussed later in the paper).

D. Related Work

Literature in recent years have shown that a combination of techniques (such as edge-based, region-based, shape-based, etc.) is more useful to solve problems that are intractable using any one method alone. In particular, the active contour framework is one that is amenable to the combination of multiple techniques. This, together with the encouraging success from our previous efforts, has inspired the present work [20].

A seminal work showed that active contour algorithms can be used to segment images that lack edges [21]. As level-sets adapt gracefully to the image topology, multiphase level-sets were designed to distinguish multiple regions of interest sans tedious topology preservation [22], [23]. Likewise, the combination of parametric active contours and region-based methods has been used effectively for multiband segmentation [24]. Further, methods which use the level-set embedding to keep track of the regions using a discrete grid have demonstrated the efficiency of this class of algorithms [25]–[27]. Multiscale and multiresolution methods have also been used with segmentation algorithms, and active contours in particular, to improve their accuracy and speed [28]–[31]. While the idea of using numbered masks is prevalent in region-based methods, the PDE literature has also made use of this [32]. Related PDE literature also includes algorithms that evolve regions based on mean curvature [32]–[35].

E. Outline

In Section II, we introduce the basic framework behind the active mask algorithm. We argue for moving from the idea of the "contour" to that of "inside and outside," that is, to that of *masks*. We then show how we adapt to the topology of the image by allowing for *multiple masks*. While active-contour forces act

(a) Original image (b) Collection of masks ψ (c) Background mask μ_1 (d) Foreground mask μ_2 (e) Foreground mask μ_4

Fig. 2. Image with nine cells, its corresponding collection of masks ψ, and representative individual binary masks μ_1 (background), μ_2, and μ_4.

on the contour, we introduce *distributing functions*, which act on the masks as a whole. We first describe the *region-growing distributing functions*, which depend on the statistical properties of the image and act to roughly segment the foreground from the background. The *voting-based distributing functions* that follow aim to separate the foreground into multiple masks. We then explain how these functions work in concert to form the basic block of the active mask algorithm (see basic block in Fig. 1).

In Section III, we add the multiscale block, which iterates on the basic block by increasing/decreasing the "magnification" of the "lens" we use; essentially, we look at local areas at multiple scales to allow the algorithm to discern the cell boundaries and escape local optima (see the multiscale block in Fig. 1). We follow with the multiresolution block; it starts at a coarse version of the image and quickly achieves the initial segmentation by going through the basic and multiscale blocks. It then "lifts" that segmentation result to a finer resolution and repeats the process (see the multiresolution block in Fig. 1). We then illustrate the entire multiresolution-multiscale active mask algorithm.

Finally, in Section IV, we describe our biological testbed on which we demonstrate the performance of the algorithm. We compare the performance of the active mask algorithm with that of the seeded watershed algorithm as well as the ground truth (images hand-segmented by experts), and show that the active mask algorithm outperforms seeded watershed both qualitatively and quantitatively.

II. Active Mask Segmentation: Basic Block

Our segmentation task is to both separate foreground (cells) from the background, as well as to separate cells from each other. In Fig. 1, we give a pictorial overview of how we accomplish this task; in this section, we concentrate on the basic block of the active mask algorithm.

A. Framework

We are given a d-dimensional image f of size $N_1 \times \cdots \times N_d$ with $(M-1)$ cells present. A multidimensional pixel in that image is denoted as $n = (n_1, \dots, n_d)$. We choose to depart from the active-contour framework based on contours. Instead, given an image f, our goal is to find an M-valued image ψ of identical size, where $\psi(n) = m$ implies that the pixel n lies inside the mask m attempting to approximate cell m. In short, ψ will be the truth table for whether or not a given pixel lies within the corresponding segmented region, dubbed a *mask*.

Ideally, we seek an algorithm that iteratively adjusts multiple masks and ultimately results in each cell being perfectly covered by a single mask, with one mask for the background. We now introduce some notation to simplify the expressions in the algorithm that follows. Let $\Omega = \prod_{k=1}^{d}[0, N_k)$ be a rectangular subset of the d-dimensional integer lattice $\mathbb{Z}^d$. We regard our image f as a member of $\ell^2(\Omega)$, that is, a member of $\ell^2(\mathbb{Z}^d)$ that satisfies $f(n) = 0$ whenever $n \notin \Omega$. Similarly to [32], we regard a collection of M masks as a function ψ that assigns each pixel $n \in \Omega$ a value $\psi(n) \in \{1, \dots, M\}$. Here, n is an element of the mth mask if $\psi(n) = m$. That is, a *collection of masks* is defined to be an element of

$$\mathcal{S}_M(\Omega) := \left\{\psi : \mathbb{Z}^d \to \{0, \dots, M\} \middle| \psi(n) = 0 \Leftrightarrow n \notin \Omega\right\}.$$

The segmented regions are $\psi^{-1}\{m\} = \{n \in \mathbb{Z}^d | \psi(n) = m\}$, $m = 1, \dots, M$, each having a corresponding *mask* μ_m, which is a binary-valued characteristic function

$$\boxed{\mu_m(n) := \begin{cases} 1, & \psi_i(n) = m; \\ 0, & \text{otherwise.} \end{cases}} \qquad (1)$$

Fig. 2 illustrates both the collection of all masks ψ as well as representative individual masks μ_m derived from ψ, for $M = 10$.

Based on the statistics of the image, it is relatively easy to roughly separate foreground/background pixels; this will be the task of the region-based distributing functions R_m described in Section II-B. Given those foreground pixels, the main challenge is to delineate the cells, especially those that touch. As cells almost always have smooth boundaries, we need to partition a homogeneous region (the foreground) into multiple regions with smooth boundaries between them; this will be the task of the voting-based distributing functions V_m described in Section II-C.

B. Region-Based Distributing Functions

We now describe the action of the region-based distributing functions, which depend on the statistical properties of the image, and which are tasked with separating background from foreground. For the purpose of segmenting fluorescence microscope images, we arbitrarily decide to always let the first mask represent the background, with the remaining $(M-1)$ masks representing individual cells. In our active mask algorithm below, the mask to which a given pixel belongs is iteratively determined by letting it and its neighboring pixels vote. The role of the region-based distributing function R_1 is to skew the voting of points of lower average intensity towards the first mask, and those of higher average intensity away from it. In practice, we construct R_1 via a soft-thresholding of a smoothed version of the original image f (see Fig. 3).

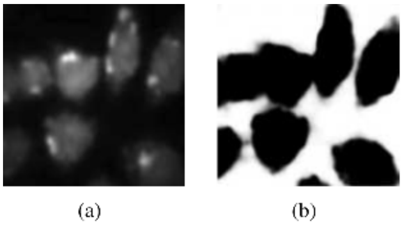


Fig. 3. Illustration of the effect of the region-based distributing function R_1 applied to image f from Fig. 2(a) (detail). (a) Smoothed image $(f \star h)(n)$; (b) soft-thresholded image as in (2).

The *region-based distributing function* R_1 is defined to be

$$\boxed{R_1(n) := \alpha \operatorname{sig}\left(\beta\left((f \star h)(n) - \gamma\right)\right)} \tag{2}$$

for some *region-based lowpass filter* h. In (2): $\operatorname{sig} : \mathbb{R} \to \mathbb{R}$ is any sigmoid-type function which asymptotically achieves the values ± 1 at $\pm\infty$, respectively; $(f \star h)$ denotes convolution adjusted for boundary issues as in (4); the *skewing factor* $\alpha \in (-1, 0)$ should be close to -1; β determines the *harshness of the threshold*; *average border intensity* γ should be taken as the average intensity of those pixels which lie on the boundary between being inside every cell and being outside them all.

In this work, we choose the following for the region-based lowpass filter h and the sigmoid function

$$\begin{aligned} h(n) &= e^{-|n|^2/a^2} \\ \operatorname{sig}(x) &= \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2/2} dt \end{aligned} \tag{3}$$

where $a > 0$ is called the *scale of the region-based lowpass filter*. Such an R_1 has the properties that:

- For a typical pixel n outside a cell, $(f \star h)(n) < \gamma$, and so $R_1(n) \approx -\alpha \approx 1$. During the voting described in Section II-D, $R_1(n)$ then skews the voting so that the background is chosen.
- For a typical pixel n inside a cell, $(f \star h)(n) > \gamma$, and so $R_1(n) \approx \alpha \approx -1$. During the voting described in Section II-D, $R_1(n)$ then skews the voting so that any mask but the background one is chosen.

Since the geometry of the cells (smoothness) alone is sufficient to distinguish cells from each other, we took $R_m(n) = 0$ for all masks but the first one, as we explain in Section II-D.

We designed R_1 to depend purely on the density of the underlying fluorescence microscope images; in general, the region-based distributing functions may take into account other statistical properties, edges, textures, and morphological features present in the image. Because of (6), one should scale the R_m's so as not to exceed 1 in magnitude, as then, their value will completely override the smoothing action of the majority voting introduced later in (9).

C. Voting-Based Distributing Functions

We now describe the action of the voting-based distributing functions, which depend on the geometric properties of the image only, and whose task is to separate individual foreground masks from each other. We draw on the idea of majority voting based on local averages. We do that by computing a convolution $(f * g)(n) = \sum_{\ell \in \mathbb{Z}^d} f(n - \ell) g(\ell)$ with some nonnegative *voting-based lowpass filter* g. Near the boundary of the image, such averages take into account many values of the image that were artificially defined to be zero, and as such, are disproportionately small. We, thus, compute these averages using the nonstandard, noncommutative convolution

$$(f \star g)(n) := \begin{cases} \frac{(f*g)(n)}{(\chi_\Omega * g)(n)}, & n \in \Omega \\ 0, & n \notin \Omega \end{cases} \tag{4}$$

where χ_Ω is the characteristic function of Ω. If $f \in \ell^2(\Omega)$, then $(f \star g) \in \ell^2(\Omega)$ is well-defined for any nonnegative, nonzero $g \in \ell^1(\mathbb{Z}^d)$. In practice, we compute both the numerator and denominator of (4) by FFT-implemented circular convolutions of zero-padded versions of f, g and χ_Ω. In so doing, the segmentation of even large images may be computed with relative efficiency.

For example, in this work, we take g to be

$$g(n) = 1 - \operatorname{erf}\left(|n|^2/b^2 - 1\right) \tag{5}$$

where b is the *scale of the voting-based lowpass filter*.

The basic step of the voting-based iteration is as follows: Given a collection of multiple masks ψ_i at iteration i, we begin by forming the masks μ_m as in (1). Next, we allow each mask to spread its influence by convolving it with g. Because $\{\psi_i^{-1}\{m\}\}_{m=1}^M$ partitions Ω, $\chi_\Omega = \sum_{m=1}^M \mu_m$, and so the values of these convolutions sum to 1

$$\sum_{m=1}^M (\mu_m \star g)(n) = \frac{1}{(\chi_\Omega * g)(n)} \sum_{m=1}^M (\mu_m * g)(n) = 1. \tag{6}$$

The *voting-based distributing function* R_1 is defined to be

$$\boxed{V_{m,i}(n) := (\mu_m \star g)(n)} \tag{7}$$

for $n \in \Omega$ and mask m at iteration i. The outcome of the voting is then determined to be

$$\psi_{i+1}(n) = \underset{m=1,\ldots,M}{\arg\max} \left[V_{m,i}(n)\right] \tag{8}$$

for any $n \in \Omega$, and $\psi_{i+1}(n) = 0$ otherwise. Here, $V_{m,i}(n)$ represents the degree to which pixel n wants to belong to mask m, based on the influence of its neighboring pixels at iteration i. In other words, the mask m producing the largest value of $V_{m,i}(n)$ is the new mask to which pixel n will belong. We have essentially pitted the masks against each other; each mask tries to conquer any neighboring pixels, and may only be stopped by another mask attempting to do the same.

Similar ideas have appeared in the PDE literature in the context of level sets. In particular, analog versions of our voting-based distributing functions have been used to evolve regions according to mean curvature [32]–[35]. Even in this continuous-domain setting, this method for evolving a contour is “difficult…to analyze theoretically or investigate numerically” [34]. We focus on the discrete version of this idea, which is related to median filtering and the threshold growth of cellular automata [36], a setting in which rigorous proofs of convergence remain elusive.

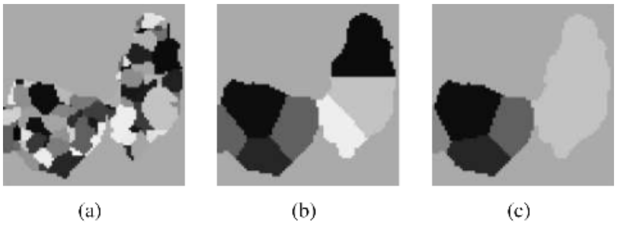


Fig. 4. Illustration of the effect of the voting-based distributing functions; (a)–(c) are details from Fig. 5(c)–(e), meant to demonstrate the separation of the foreground region into masks with smooth boundaries.

We illustrate (8) in Fig. 4 as well as with the following simple example.

Example 1: Consider the ith iteration of a 2-D image f with the voting-based lowpass filter g being 1 on a 3×3 square centered at the origin, and 0 otherwise. Further, suppose that we have $M = 3$ masks and that the values of ψ_i in a 3×3 neighborhood of some pixel $n = (1,1)$ are

$$\begin{array}{|c|c|c|}\hline \psi_i(0,0) & \psi_i(0,1) & \psi_i(0,2) \\ \hline \psi_i(1,0) & \psi_i(1,1) & \psi_i(1,2) \\ \hline \psi_i(2,0) & \psi_i(2,1) & \psi_i(2,2) \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 2 & 2 & 2 \\ \hline 2 & 1 & 3 \\ \hline 2 & 1 & 1 \\ \hline \end{array}.$$

Computing $V_{m,i}(1,1)$ for $m =$ 1, 2, 3, yields 3/9, 5/9 and 1/9, respectively; thus, in iteration $i + 1$, the pixel (1, 1) will change its membership from mask 1 to mask 2. □

The evolution of the algorithm based on the voting-based distributing functions is given in a pseudocode form in Algorithm 1. The skewing factor in the algorithm is needed for later; to examine just the effect of the voting-based functions, we set $R_m = 0$ for all m. Experiments reveal that Algorithm 1 will eventually result in masks at equilibrium (that is, further iteration of the voting procedure causes no further change and the algorithm has converged). The minimum thickness of the masks at equilibrium seems to be mostly dependent on the size of the support of the voting-based lowpass filter g. Similarly, the smoothness of the boundaries between distinct masks at equilibrium seems to depend greatly on the smoothness of the voting-based lowpass filter g. If we are looking for smooth boundaries, we should avoid the blocky g described in the example above, and instead choose a g weighted in a more isotropic manner as in (5). In this sense, Algorithm 1 is the active mask version of the smoothing forces traditionally associated with active contour algorithms.

Algorithm 1 [Active mask voting]
Input: Number of masks M, iteration number i, collection of masks ψ_i, voting-based lowpass filter g, skewing factors R_m.
Output: Collection of masks ψ and final iteration number i.

```
AMVoting($M, i, \psi_i, g, R_m$)
    while $\psi_{i+1} \neq \psi_i$ do
        Compute the voting-based distributing functions $V_m$ (7)
        Active mask voting (9)
        $i = i + 1$
    end while
    return $(\psi_i, i)$
```

D. Active Mask Basic Block Evolution

Iteratively applying Algorithm 1 will not successfully segment the image in question, as it does not take the image itself into account. As with active contours, we need to combine the purely geometric smoothness forces with image-based forces to push the segmentation in the right direction. In the active mask framework, the region-based distributing functions R_m, given by (2) for $m = 1$ and taken to be $R_m = 0$, for $m = 2, \ldots, M$, will skew the voting of the purely geometric voting-based distributing functions V_m (7).

The *active mask basic block* is defined to be

$$\boxed{\psi_{i+1}(n) := \operatorname*{argmax}_{m=1,\ldots,M} \left[V_{m,i}(n) + R_m(n)\right]} \qquad (9)$$

for $n \in \Omega$, and $\psi_{i+1}(n) = 0$ otherwise. Thus, as we explained in Section II-B, with $\alpha \approx -1$, for a typical pixel n inside a cell, R_1 skews the voting so that any mask but the background one is chosen. Similarly, for a typical pixel n outside a cell, R_1 skews the voting so that the background is chosen.

To start the iterative process, we initialize the masks ψ randomly. That is, for any given $n \in \Omega$, we let $\psi_0(n)$ be a number randomly chosen from $\{1, \ldots, M\}$, denoted $\psi_0(n) = \text{rand}(M)$. The strong preference that R_1 exhibits for the background will quickly distinguish it from the foreground. With the foreground versus background question settled, the remaining masks compete for supremacy, in a manner similar to the unbiased voting (8). Inevitably, larger masks will consume smaller ones. If at some point $\psi_i(n) \neq m_0$ for all $n \in \Omega$, then a given mask is no longer present at a given iteration and is eliminated forever, reducing the number of masks M, and consequently, the number of convolutions one must compute in (7). Ultimately, the iterations will converge when the remaining masks that correspond to the cells achieve an equilibrium. This will typically happen when the inherent geometry of the cells causes a stalemate. For example, when two round cells touch, a separate mask grows to dominate each cell. The boundary between the two masks will coincide with the boundary between the two cells, as the narrowing of the cells near their boundary creates a narrow pass which neither side is able to conquer. Thus, the repeated application of Algorithm 1 will often result in a first mask which contains the background, while the remaining nonzero masks each describe a nonzero cell. The steps of the basic block of the active mask algorithm are given in a pseudocode form in Algorithm 2.

Algorithm 2 [Active mask basic block]
Input: Image f, initial number of masks M, initial collection of masks ψ_0, scale of the region-based lowpass filter a, scale of the voting-based lowpass filter b, skewing factor α, harshness of the threshold β, average border intensity γ.
Output: Collection of masks ψ and final iteration number i.

```
ActiveMasks($f, M, \psi_0, a, b, \alpha, \beta, \gamma$)
    Compute the region-based lowpass filter $h$ (3)
    Compute the region-based distributing functions $R_m$ (2)
    Compute the voting-based lowpass filter $g$ (5)
    Active mask voting as in Algorithm 1
    $(\psi, i) =$ AMVoting($M, 0, \psi_0, g, R_m$)
    return $(\psi, i)$
```

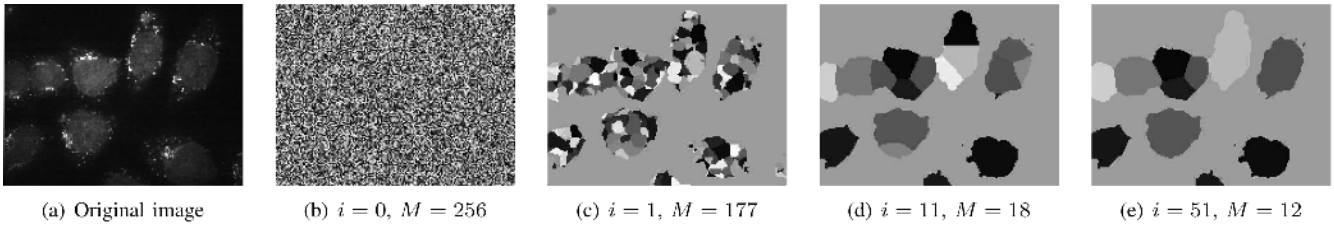


Fig. 5. Illustration of the evolution of the basic block of the active mask algorithm with random initialization. (a) A COPII image (enhanced for visibility). (b) Random initialization with $M = 256$ initial masks. (c) In iteration $i = 1$, the number of masks reduces to $M = 177$. (d)–(e) Further iterations resulting in convergence at iteration $i = 51$, with a final number of masks $M = 12$.

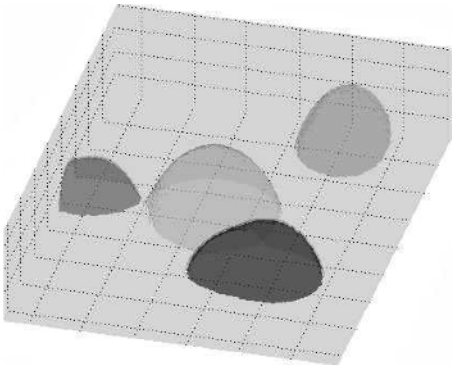


Fig. 6. Three-dimensional active mask segmentation of a z-stack. The blobs are segmented cells.

The evolution of the basic block of the active mask algorithm and the successive reduction in the number of masks as the iterations converge are illustrated in Fig. 5. Fig. 5(a) shows a fluorescence microscope image of HeLa cells with six whole cells visible and three partially visible. Fig. 5(b) shows the initial random state of ψ with $M = 256$ masks (we call this iteration $i = 0$). In the first iteration, R_1 coarsely separates the background from the foreground, discarding empty masks and, thus, drastically reducing the number of masks to $M = 177$ as shown in Fig. 5(c). Subsequently, the masks in the foreground region compete with each other, with the geometry of the cells prevailing. Thus, in Fig. 5(d), at $i = 11$, we see the masks showing an increased correspondence to the cells while the number of masks is steadily decreasing, with convergence achieved in iteration $i = 51$.

The active mask algorithm works easily with higher-dimensional images; in fact, the cells we seek to segment are 3-D; we give an example of 3-D segmentation in Fig. 6. The blobs in Fig. 6 are segmented cells; a 2-D slice along the z-axis of this volume represents the segmentation outcome for the corresponding 2-D image.

No algorithm is perfect, and occasionally splits and merges of cells will occur, particularly during 2-D segmentation [for instance, a split is visible in Fig. 5(e) in what should be mask μ_4; see Fig. 2(e)]. Generally, these anomalies depend on the design of the voting-based and region-based distributing functions, and how sensitive the filters are to the features that distinguish the objects of interest. In particular, for the segmentation of fluorescence microscope cell images with the functions V and R described above, if the region-based function has not been smoothed adequately, the masks during the local majority voting may split a single cell into two or more regions based on the kinks on the surface of a single cell. Alternatively, if the scale of the voting-based lowpass filter is too small, the function may be too sensitive to small indentations in the foreground, causing spurious splits. Likewise, if the scale parameters of the filters are too large, cusps between touching cells may be smoothed out, resulting in spurious merges. We address these problems in the following section.

III. Active Mask Segmentation: Multiresolution and Multiscale Blocks

In the previous section, we described the basic block of the active mask segmentation algorithm. We also noted instances when splits occur [see Fig. 5(e)]. We now address the issue of splits by allowing the scale parameters to change; we address the issue of speed by allowing resolution to change.

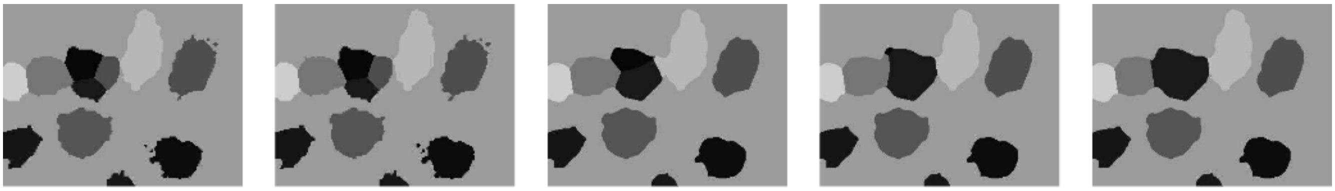
A. Multiscale Block

To overcome the problem of spurious splits noted in Section II-D, we could slowly change the scale $a > 0$ of the region-based lowpass filter h from which the region-based distributing function (2) is computed. For example, with h as in (3), choosing a to be large results in a quickly converging algorithm but one whose ultimate masks overestimate the size of the cells in addition to oversmoothing their boundaries, while choosing a to be small yields better masks but slower convergence.

For the best of both worlds, a is initially taken to be large to rapidly obtain a coarse segmentation. Then, after (9) has converged for this specific a, these resulting masks form the initial guess for the rerunning of (9) with a slightly smaller a. When the change in a is slight, the second iteration will converge in a few steps. This process is then repeated, gradually pulling a down to a point where experimentation reveals the ultimate segmentation as being a good match to the ground truth.

B. Multiresolution Block

In addition to simplifying one's perspective over discrete domains, a mask-based formulation of segmentation permits an easy implementation of multiresolution schemes to speed the algorithm's performance, by obtaining a fast and coarse segmentation of the image at a low resolution and then successively refining the result at higher resolutions [37]. While one could use any lowpass filter to obtain a coarse version of the original image, our choice, Haar, was guided by two principles: simplicity and the ability to model the signal readout behavior of the CCD camera of a fluorescence microscope [38].



(a) $k = 3$, $a = 3.5$, $i = 57$, $M = 12$ (b) $k = 3$, $a = 3$, $i = 65$, $M = 12$ (c) $k = 2$, $a = 5$, $i = 134$, $M = 11$ (d) $k = 2$, $a = 5$, $i = 149$, $M = 10$ (e) $k = 2$, $a = 4$, $i = 179$, $M = 10$

Fig. 7. Illustration of the evolution of the multiresolution multiscale active mask algorithm starting from Fig. 5(e) which used resolution level $k = 3$ and scale of the region-based lowpass filter $a = 4$ to Fig. 7 at resolution level $k = 2$ and scale $a = 4$, with the final number of masks $M = 10$. This final result is also shown in pseudo color in Fig. 11(c). (Images at different resolutions have been scaled to the same size for display purposes).

1) From Fine to Coarse: Let H be the Haar smoothed and downsampled (coarse) version of the original image f

$$(Hf)(n) := 2^{-d} \sum_{\ell \in [0,1]^d} f(2n + \ell). \tag{10}$$

For example, for $d = 1$, the output Hf is half the size of f and is given by $Hf(n) = [f(2n) + f(2n+1)]/2$.

Thus, we consider a multiresolution segmentation algorithm in which we first segment a coarse version of f, $H^K f$ (H applied to f K times), which is smaller than f by a factor of 2^K in each dimension. Once Algorithm 2 has iteratively produced segmentation masks $\psi^{(K)}$ for $H^K f$, we use this information as a starting point for the segmentation of the slightly more detailed image $H^{K-1} f$.

2) From Coarse to Fine: The multiresolution block at iteration $k - 1$ will begin with the *lifted* version of the coarser mask $\psi^{(k)}$. The lifted version $\psi^{(k-1)}$ is obtained by copying each binary value of $\psi^{(k)}$ into 2^d values of ψ^{k-1}

$$\psi^{(k-1)}(n) := \psi^{(k)}(\lfloor 2^{-1} n \rfloor) \tag{11}$$

where the flooring operation is performed coordinate-wise (equivalent to upsampling and repetition interpolation). For example, for $d = 1$ and if $\psi^{(k)} = 1101$, $\psi^{(k-1)} = 11110011$. Here, a contour-based implementation of this same idea would be unnecessarily complicated: one would need to first determine the inside of a given contour to find a mask, then apply (11) to this mask to obtain a new one of twice its size, and finally take the boundary of this resulting mask to find the new contour. In short, multiresolution segmentation lends itself more naturally to mask-based, rather than contour-based, segmentation algorithms.

Algorithm 3 [Multiresolution multiscale active masks]

Input: Image f, initial number of masks M, initial resolution level K, scale of the region-based lowpass filter a, scale of the voting-based lowpass filter b, skewing factor α, harshness of the threshold β, average border intensity γ.

Output: Collection of masks ψ and final iteration number i.

```
**MRMSActiveMasks**$(f, M, K, a, b, \alpha, \beta, \gamma)$

  *Initialization*
  $i = 0, k = K, j = 1, \psi_0^{(k,j)}(n) = \mathrm{rand}(M), I = f$
  *Multiresolution block*
  **while** $k \geq 0$ **do**
    *Compute coarse version of the image*
    $f(n) = H^k I(n)$
    *Multiscale block*
    **while** $j \leq J_k$ **do**
      *Basic block*
      $\left(\psi_p^{(k,j)}, p\right) = \mathbf{ActiveMasks}\left(f, M, \psi_0^{(k,j)}, a_{k,j}, b, \alpha, \beta, \gamma\right)$
      *Change of scale parameter at resolution k*
      $\psi_0^{(k,j+1)}(n) = \psi_p^{(k,j)}(n)$
      $j = j + 1, i = i + p$
    **end while**
    *Lift mask $\psi$ to the next higher resolution $k - 1$*
    $\psi_1^{(k-1,1)}(n) = \psi_p^{(k,j-1)}(\lfloor 2^{-1} n \rfloor)$
    $k = k - 1$
  **end while**
  **return** $(\psi, i)$
```

C. Multiresolution Multiscale Active Mask Evolution

The complete multiresolution multiscale algorithm, beginning with some randomly chosen $\psi_0^{(K,1)} \in \mathcal{S}_M(\Omega)$, is given in a pseudocode form in Algorithm 3 and shown in Fig. 1, with index k referring to the multiresolution block, and index j to the multiscale block. We illustrate the evolutionary behavior of the algorithm in Fig. 7. For any fixed k and j, we iteratively apply (9) until the masks are in equilibrium. We resume the process shown in Fig. 5(e), at $k = 3$ and $a_j = 4$, and use those masks as the starting point for the next step in the scale change. When the change of scales for a given resolution is completed [see Fig. 7(a)–(b)], we lift the resulting masks to a higher resolution, $k = 2$, and begin the process again [see Fig. 7(c)–(e)]. The final masks are given by the function $\psi_i^{(K_0, J_{K_0})}$, where $K_0 = 2$ in the example. Note how the splits from Fig. 5(e) are no longer present in Fig. 7(e). We could change the scale parameter and pull the resolution back to the original resolution; however, as satisfactory segmentation has already been achieved at this stage, we choose $k = 2$ to stop. When $k = 0$, the masks given by the function $\psi_i^{(0,J_0)}$ provide a full-resolution, multiscale active mask segmentation of the original image. The scale b of the voting-based lowpass filter g used in the voting-based distributing function, chosen to be a fixed constant here, can also be changed. The smaller the scale b, the longer it takes for $\psi^{(k,j_k)}$ to converge, but the boundaries between the foreground masks are more accurate. The parameters α, β and γ are fixed constants which are experimentally determined for optimal performance. The choice of all the parameters as well as the performance of the algorithm, both in terms of speed and accuracy, are given in Section IV.

IV. Experimental Results

There are many biological problems for which punctate markers of the cell periphery are used. We focus on one such problem where manual segmentation presents a critical obstacle.

A. *Biological Testbed: Influence of Golgi Protein Expression on the Size of the Golgi Body and Cell*

The Golgi body is a membrane-bound organelle in eukaryotic cells whose cytoplasmic surface is a site for a number of important signaling pathways [39]. Further, the Golgi body mediates the processing and sorting of proteins and lipids in the final stages of their biosynthesis. Defects in the biosynthetic pathway are responsible for many human diseases [40]. An understanding of the molecular basis of these defects is paving the way to future effective therapeutics [41].

The Linstedt lab seeks to understand how the Golgi body's underlying structures are established and maintained, how they are regulated in stress as well as the purpose of each structural feature, that is, they seek a structure/function analysis from the underlying components of the organelle. The current hypothesis being tested is that the affinity of interaction between vesicle coat complexes and SNARE molecules (and other Golgi proteins) establishes and controls the size of Golgi compartments [42]. These tests depend on a quantitative ratiometric assay comparing Golgi size to cell size. The assay is fluorescence microscopy-based and accurate segmentation is a critical step in determining cell volume. To date, the experimental tests of this hypothesis have resulted in the discovery that direct interactions between the cytoplasmic domains of Golgi proteins exiting the endoplasmic reticulum (another membrane-bound organelle in eukaryotic cells) and the COPII (a vesicle that transports proteins from the endoplasmic reticulum to the Golgi body) component Sar1p regulate COPII assembly, providing a variable exit rate mechanism that influences Golgi size [43].

1) Need for Automated Segmentation: Hand segmentation of cell boundaries was performed based on the diffuse hazy background staining of the COPII subunit Sec13 in each image. Hand segmentation limits the analysis to cells with a flat morphology (fewer slices in a z-stack) and to a small number of cells. To extend the findings, the Linstedt lab needs to analyze a developmental time course over a large number of cells and, as the cell types involved are not flat, it will be necessary to have a larger number of slices per cell.

The segmentation challenge posed by these images is that the COPII subunit Sec13 exhibits not only a bright punctate pattern in the cytoplasm, but also a hazy cytoplasmic pattern [see Fig. 9(a) for an example]. Consequently, the signal is not uniform and not always strong, which makes it hard to visually determine the boundary of a cell. Further, as these experiments do not require the nuclear protein to be imaged, there is no parallel channel from which to obtain accurate seeds as initialization for the seeded watershed or Voronoi algorithms. As the computation of cell area/volume is critical in this application, algorithms that do not yield tight contours are not suitable. Using rule-based methods to obtain tight contours or approaches, such as active contours that require involved tuning for each image, make the segmentation semi-automated at best, and would require investing a considerable effort to obtain good segmentation results. Thus, we look to the active mask algorithm to segment these images.

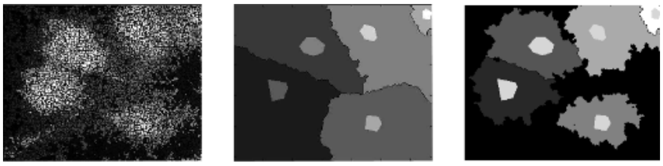

(a) Automatic seeds (b) Hand-drawn foreground seeds (c) Automatic background seeds added

Fig. 8. Seeded watershed algorithm results: (a) With automatic seeds, the algorithm results in 5179 regions that need to be merged based on rules [7]. (b) With hand-drawn foreground seeds (light gray) but no background seeds, the algorithm results in regions that include a significant portion of the background with each cell. (c) Adding background seeds, the algorithm results in reasonable segmentation (background seeds are automatically chosen as every minimum in the smoothed cell image).

B. *Data Sets*

The data was generated in two experiments.

1) DS1: The first data set consists of 15 z-stacks containing 3–8 cells each. Each stack contains 40 slices of size 1024 × 1344. The HeLa cells were double labeled with the COPII subunit Sec13 (a cytosolic protein peripherally associated with the membrane) and the Golgi marker protein, giantin, in two parallel channels [see Fig. 11(a)–(b)]. Sec13 staining has a diffuse cytoplasmic background used to mark the boundary of the cell. The pixel size was 0.05 μm in x/y directions and 0.3 μm in z direction. The findings based on hand segmentation of these images were reported in [43]. We developed the cell-volume computation and Golgi-body segmentation algorithm based on two of the stacks and tested it on others.

2) DS2: The second data set consists of 5 z-stacks containing 2–4 cells each. Each stack contains 20 slices of size 1024 × 1344. The HeLa cells were labeled with Sec13. While the cells imaged in both DS1 and DS2 belong to the same cell line (HeLa), the ones in DS2 are thinner, and thus we have fewer slices per stack. The pixel size was the same as for DS1.

3) Hand-Segmented DS2: To assess the performance of the algorithm, we used the hand-segmented version of the DS2 set as ground truth. The images were hand segmented by two experts in three separate sessions. This was done to minimize the effect of an individual's bias on a given day as well as ensuring sufficient respite to segment as accurately and precisely as possible [see Fig. 9(a)–(b)].

C. *Parameter Selection*

We now discuss the selection of parameters as input to Algorithm 3; for convenience, we summarize them in Table I.

1) Initial Number of Masks M: We initialized the algorithm with $M = 256$ random masks. The larger the M, the lower the possibility of unwanted merges.

2) Initial Resolution Level K: In this work, we decomposed the original images to $K = 3$ levels (one-eighth the original resolution), lifting the result as per (11), to $k = 2$ (one-fourth the original resolution). We did not refine any further, as we obtained a satisfactory result at $k = 2$.

TABLE I
PARAMETERS USED IN EXPERIMENTS

Param.	Description	Ref.	Value	Res. k
M	Initial number of masks		256	
K/K_0	Initial/final resolution levels		$3/2$	$3/2$
a	Scale of the	(3)	$(4, 3.5, 3)$	3
	region-based lowpass filter		5	2
b	Scale of the	(5)	8	3
	voting-based lowpass filter		16	2
α	Skewing factor	(2)	-0.9	
β	Harshness of the threshold	(2), (12)	$1.5 - 2.0$	
γ	Average border intensity	(2), (12)	$3.0 - 4.2$	

3) Scale Factor a of the Region-Based Lowpass Filter: For the multiscale evolution phase, we did not overly tune the parameters. We determined the scale parameter $a_{K,1}$ for the region-based lowpass filter h in Algorithm 3 based on the resolution k as well as the approximate size of a cell at the coarsest resolution, $k = K$. We began with $a_{3,1} = 4$ at one-eighth the original resolution and pulled back to $a_{3,3} = 3$ in decrements of 0.5 (the numbers in parentheses in Table I). At $k = K - 1 = 2$, that is, one-fourth the original resolution, we used just one value, $a_{2,1} = 5$. In practice, the more gradual the scale change, the faster the convergence for different scales j at a fixed k. Further, while it is not necessary to decrement the values during the scale change process (they can also be incremented), larger a allows us to obtain a fast coarse segmentation, whereas smaller a takes longer to compute but provides a finer segmentation. We used the same values of a on all the images, without tuning them per image or stack.

4) Scale Factor b of the Voting-Based Lowpass Filter: We changed the scale for the voting-based lowpass filter g in (5) based on k; for $k = 3, b = 8$, for $k = 2, b = 16$. In selecting the values, we were guided by the twin goals of expediting convergence while not compromising on the quality of the boundary.

5) Skewing Factor α: We chose $\alpha = -0.9$ as the weight of R_1 given by (2).

6) Harshness of the Threshold β and Average Border Intensity γ: To facilitate and broaden the use by biologists, we have made an effort to use those parameters deemed intuitive; thus, to determine β and γ, we use the average foreground and background value, denoted μ_{fore} and μ_{back}, respectively. We then set

$$\beta = \frac{4}{\mu_{\text{fore}} - \mu_{\text{back}}}, \quad \gamma = \frac{\mu_{\text{fore}} + \mu_{\text{back}}}{2}. \tag{12}$$

While the tuning was minimized to a large extent through scale change, it was not entirely eliminated. We determined β and γ based on one image from each stack and used the same numbers for all the other images from that stack. These coarse adjustments are possible because the performance of the algorithm is robust within a small range of the parameter values. While it may be possible to obtain better results than we report with a more involved tuning procedure, it is not practical to expect an involved tuning of the parameters in high-throughput applications, and, thus, we did not resort to such a tuning. The range of parameter values is given in Table I, while values used for each stack as well as the scripts used to obtain the results in this section are provided with the reproducible research compendium at [44].

D. Reference Algorithm: Seeded Watershed

Our choice of the reference algorithm was entirely guided by our general goal, that of segmenting fluorescence microscope images of punctate patterns. As the seeded watershed algorithm is widely used for such a task, it was one of our candidates. Following reviewers' comments, we also considered algorithms from [13]–[15] as well as general region-growing from a single seed point (available in Matlab). Of these, sufficient information/source code with instructions was available only for the seeded watershed algorithm and the general region-growing algorithm available in Matlab. Since the gray levels inside and outside the cells are not uniform for the class of images we consider, the general region-growing algorithm did not perform well. We, thus, chose to compare the performance of our active mask algorithm to that of the seeded watershed algorithm, as well as the hand-segmented masks.

When we use the watershed algorithm described in [7], the algorithm bins regions in the image based on their grayscale values. We obtain a large number of regions (spurious splits) despite smoothing the fluorescence microscope images to reduce the effect of shot noise [see Fig. 8(a)]. This result requires a postprocessing step defining rules to merge the highly fragmented regions into a reasonable number of regions. We also used hand-drawn seeds to initialize the algorithm and used the version described in [8] to obtain as many regions as the initial seeds. Even this proved insufficient as the entire image was assigned to one region or another [see Fig. 8(b)]. The resulting cell masks were not tight and included a significant portion of the background. We then further tuned the seeding procedure to seed regions in the background to prevent the foreground regions from expanding to the image borders. This result produced tighter masks [see Fig. 9(c)] which we used for comparison with results produced by the active mask algorithm.

E. Performance Evaluation

We now discuss the performance of the algorithm both qualitatively (visual inspection) as well as quantitatively. We also report the runtime.

1) Qualitative Evaluation: We compared the results of the active mask algorithm with the hand-segmented masks, shown in Fig. 9. As we noted before, the seeded watershed with automatic seeds performs poorly without extensive postprocessing. Both the seeded watershed algorithm with hand-drawn foreground and automatic background seeds as well as the active mask algorithm with either random or hand-drawn seeds, seem to include a slightly larger area than that marked by hand segmentation. The results of active mask algorithm seem to match the area marked by the experts better than the masks produced by seeded watershed algorithm [see Fig. 9(c)–(e)].

Further, it appears that the active mask algorithm initialized with random seeds performs almost as well as when initialized with hand-drawn seeds. Although this is the case for a majority of the images, due to a random initial configuration, there are some anomalous results as well, such as the one shown in

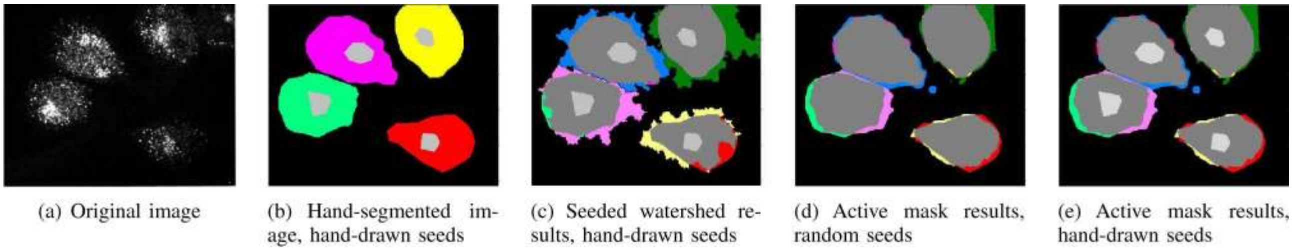

(a) Original image (b) Hand-segmented image, hand-drawn seeds (c) Seeded watershed results, hand-drawn seeds (d) Active mask results, random seeds (e) Active mask results, hand-drawn seeds

Fig. 9. (a) Image of HeLa cells marked with COPII at one-fourth the original resolution. (b) Hand-segmented masks (green, pink, yellow and red), together with hand-drawn seeds (light gray). (c) Seeded watershed algorithm initialized with hand-drawn seeds and overlaid on the hand-segmented masks for comparison. (The excess areas marked by seeded watershed algorithm are indicated in light pink, blue, green and yellow.) Area similarity is $\mathrm{AS} = 82.48\%$. (d) Active mask algorithm initialized with random seeds overlaid on the hand-segmented masks. Area similarity is $\mathrm{AS} = 91.80\%$. (e) Active mask algorithm initialized with hand-drawn seeds overlaid on hand-segmented masks. Area similarity is $\mathrm{AS} = 91.80\%$.

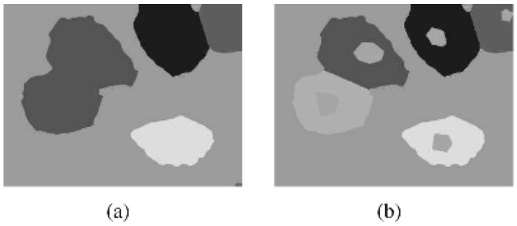

(a) (b)

Fig. 10. Effect of initialization on algorithm 3. (a) A merge occurs, two cells are covered by the same mask. (b) Given reasonable seeds, the anomaly from (a) disappears.

Fig. 10(a). If indeed the problem was only that of an unfavorable initial configuration, often just rerunning the algorithm without changing parameters produces a better result. If the problem resulted from insufficient information or a choice of parameters that induced the masks to split a cell at a particular scale, changing the scale parameters of the filter(s) in the distributing function(s) eliminates spurious splits during the course of the evolution itself [see Fig. 5(b)–(e)]. Introducing details available in a higher-resolution version of the image can also help recover from such spurious splits (see Fig. 7). These techniques are already built into the mask evolution phase. However, they do not help in recovering from a spurious merge during the course of the evolution. As it is not always possible to rerun the algorithm, if seeds are available, for example, in the form of reliable information from a parallel channel, they do help in limiting anomalies resulting from the initial randomness [see Fig. 10(b)].

2) Quantitative Evaluation: To quantify the performance of the algorithm, we used the standard performance measure, area similarity (AS), for each cell in each of 80 different 2-D slices. AS normalizes twice the area that is common to the masks by the sum of the areas of hand segmentation and the algorithm and, thus, penalizes the algorithm that produces larger regions. Performance of AS $\geq 70\%$ generally implies a good agreement of the algorithm's result with the ground truth [45]. The area similarity between the mask j in the active mask algorithm (or seeded watershed algorithm) and the mask i in the hand-segmented data set (where $n(\mathrm{AM}_j)$ is the number of nonzero values in the mask AM_j) and is given by

$$\mathrm{AS}(i,j) = \frac{2n(\mathrm{HS}_i \wedge \mathrm{AM}_j)}{n(\mathrm{HS}_i) + n(\mathrm{AM}_j)}.$$

TABLE II

QUANTITATIVE ASSESSMENT. AREA SIMILARITY MEASURE COMPUTED FOR THE ACTIVE MASK ALGORITHM AND THE REFERENCE FOR EVALUATION, SEEDED WATERSHED ALGORITHM

	Area similarity [%]	
	Automatic seeds	Hand-drawn seeds
Seeded watershed	0.14	72.94
Active masks	84.73	86.06

For partial cells (cells on the periphery), hand-segmented masks were not provided. While the active mask algorithm segmented every cell in the image, whether it was at the periphery or at the center, the seeded watershed algorithm tended to include peripheral cells in the same region as a neighboring central cell. For the seeded watershed algorithm, this required seeding the incomplete cells to avoid leakage of watershed regions of valid whole cells in the image. To compute the numbers, we considered only those cells that were chosen in the hand-segmented data set. As the density of cells varies per image, and to account for the influence of neighboring cells on the performance of the algorithm for a cell, we averaged the performance measures per image before computing the overall average for the data set.

A summary of the results is presented in Table II. For the seeded watershed algorithm, $\mathrm{AS} = 72.94\%$ with $\sigma = 13.90$ for hand-drawn seeding, while for the active mask algorithm, $\mathrm{AS} = 84.73\%$ with $\sigma = 7.90$ and $\mathrm{AS} = 86.06\%$ with $\sigma = 7.15$ for random seeding and hand-drawn seeding, respectively.[1] AS for the seeded watershed with automatic seeding is very low. Whereas in practice the seeded watershed is not initialized with automatic seeding as such, we have provided this measure for completeness. As mentioned in Section IV-D, there are various flavors of the algorithm; perhaps a more sophisticated version, such as in combination with active contours, would perform better [46]. We do not comment further on the seeded watershed with automatic seeding. In what follows, we use only the seeded watershed algorithm with perfect seeding for comparison.[2] The active mask algorithm outperforms the seeded watershed algorithm in both cases. Moreover, the active mask algorithm has a

[1] This is with all the preprocessing described in Section IV-D to make the comparison as fair as possible to the seeded watershed algorithm.

[2] Perfect seeding means hand-drawn seeds for the foreground and automatic for the background, as explained earlier in Section IV-D.

lesser chance of detecting false positives in the image. As we mentioned earlier, the performance of active mask algorithm with random seeding is not significantly different from that with hand-drawn seeds. These numbers underscore that the algorithm almost always performs just as well with random seeds as it does with hand-drawn seeds.

3) Runtime: We implemented the active mask algorithm in MATLAB (version 2007a) for flexible design and prototyping [47]. Due to the random initial configuration, runtime ranged between 30 s to 2 min on an Intel Pentium M 1.6-GHz processor with 1.5-GB memory for the results above. We have also created an ImageJ plugin [48], to facilitate wide distribution among biologists. Apart from algorithmic optimization (such as not waiting for absolute convergence at each scale before changing the scale parameter, and using the segmentation of one image to initialize the segmentation of the succeeding image in a stack) to expedite convergence, optimizing the code in compiled languages such as C may yield anywhere between a 10x to a 1000x speedup, if the problem is not memory bound (that is, a lot of data and very few computations) [49].

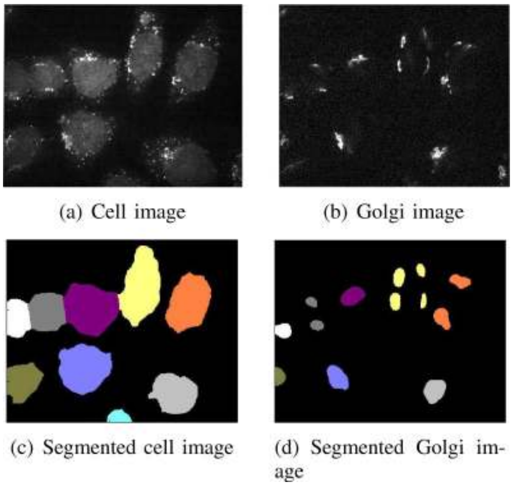
F. Application-Specific Processing

In Section IV-A, we introduced our biological testbed, the application that motivated the design of an automated segmentation algorithm for fluorescence microscope images of punctate patterns. As examples of application-specific postprocessing modules, we show how cell-volume computation and Golgi-body segmentation were performed, given the segmentation outcome of the active mask algorithm.

1) Cell-Volume Computation: Given the active mask segmentation outcome, each μ_m in (1) represents a distinct region. If we set $d = 2$ and segment the images of a stack in 2-D to obtain an approximation of the cell volume, we can simply sum the areas of the 2-D masks, as it is done when hand segmentation of the 2-D images is used to process the images. To ensure the cells along a stack are assigned the same mask number, we may initialize one of the slices in a stack where the cells are fairly discernible with an initial $M \gg L$ masks, where L is the expected number of cells. The segmentation outcome of this slice can be used to initialize the neighboring masks and so on. Postprocessing to match a region in one mask with the most overlapping region in a successive mask may also be used in computing the volume from a 2-D segmentation procedure [1]. This pseudo-3-D segmentation (as the 2-D segmentation benefits from the information in 3-D) also speeds up segmentation of a stack over segmenting each image in the stack independently.

We can also segment in 3-D directly. Such a segmentation affords an elegant way of visualizing the cells in the z-stack and the segmentation outcome is not hampered when a cell is occluded by another cell in a 2-D section (see Fig. 6). Further, the volume of each cell can be computed in a straightforward manner from such a result.

2) Golgi-Body Segmentation: The Golgi-body is a double-membraned organelle consisting of cisternae stacked to increase its surface area to facilitate secretion. Thus, though there is only one Golgi-body in each cell, in a 2-D section, a Golgi-body may appear as multiple fragments [see Fig. 11(b)]. If the cells are close, then by using the Golgi channel alone it is not easy to associate the different pieces of the organelle in 2-D with others that belong to the same cell. Given the active mask segmentation of cells for the COPII channel of an image, we may use the resulting masks to initialize the segmentation of the corresponding Golgi channel. The advantage is that multiple pieces of the Golgi body belonging to a cell are now marked by the same mask [see Fig. 11(d)]. Moreover, such a Golgi mask matches the cell to which the different Golgi fragments belong [see Fig. 11(c)]. This facilitates the computation of the Golgi-volume and, subsequently, the ratio of the Golgi volume to the cell volume as required by the application.


(a) Cell image (b) Golgi image

(c) Segmented cell image (d) Segmented Golgi image

Fig. 11. (a) Original HeLa cell image [COPII, also shown in Fig. 5(a)]. (b) Original HeLa Golgi image (giantin). (c) Active mask segmentation of the parallel cell image. (d) Active mask segmentation of the corresponding Golgi image. Using cell segmentation to initialize Golgi segmentation helps associate the different fragments of the Golgi-body in a 2-D slice of a cell to their corresponding cells.

V. Conclusion

We have proposed a new algorithm termed *active mask segmentation*, designed for segmentation of fluorescence microscope images of punctate patterns, a large class of data. The framework brings together flexibility offered by active-contour methods, speed offered by multiresolution methods, smoothing offered by multiscale methods, and statistical modeling offered by region-growing methods, to construct an accurate and fast segmentation tool. It departs from the idea of the contour and instead uses that of a mask, as well as multiple masks. The algorithm easily performs multidimensional segmentation, can be initialized with random seeds, and uses a few easily tunable parameters.

We compared the active mask algorithm to the seeded watershed one, and showed active masks to be highly competitive, both qualitatively and quantitatively. We have also compared the algorithm to the ground truth, hand-segmented masks provided by experts, and have found good agreement. We have shown how the segmentation results may be used with application-specific postprocessing modules using the example of cell-volume computation and Golgi-body segmentation for the study of the influence of Golgi protein expression on the cell size. Thus, we have demonstrated that, for segmentation of fluorescence microscope images of punctate patterns, active mask segmentation is a viable alternative to hand segmentation or any standard algorithm (such as seeded watershed or level-set based methods). We are currently expanding the repertoire of distributing functions that can be used to segment a wider range of images such as those of tissues as well as images that possess features such as edges. Moreover, though even simple, continuous-domain versions of our algorithm are difficult to analyze, we are currently seeking proofs of convergence of (9).

VI. Reproducible Research

To facilitate sharing the method with end users as well as developers, we provide the code and information necessary to reproduce the results in this paper at [44]. The code is given both as developed in Matlab, as well as the ImageJ plugin.

Acknowledgment

The authors would like to thank the four anonymous reviewers as well as the Associate Editor, Dr. D. Van De Ville, for their insightful and thorough reviews. They would also like to thank J. Newberg for adapting the seeded watershed algorithm to our data and W. Ruder for his helpful inputs on various markers for cells, as well as M. Gonzales-Rivero and S. Hsieh for implementing the volume computation and Golgi-body segmentation modules; Dr. A. Chebira and D.-C. Balcan for insightful discussions; Prof. R. Murphy and Prof. J. Moura for their valuable suggestions regarding performance evaluation; G. Kim, K. Qian, Dr. A. Chebira, and Dr. P. H. Yeomans for their work on the ImageJ plugin.

References

[1] G. Srinivasa, M. C. Fickus, M. N. Gonzales-Rivero, S. Y. Hsieh, Y. Guo, A. D. Linstedt, and J. Kovačević, "Active mask segmentation for the cell-volume computation and Golgi-body segmentation of HeLa cell images," in *Proc. IEEE Int. Symp. Biomed. Imaging*, Paris, France, May 2008, pp. 348–351.

[2] J. Zhang, R. E. Campbell, A. Y. Ting, and R. Y. Tsien, "Creating new fluorescent probes for cell biology," *Nat. Rev. Mol. Cell Biol.*, vol. 3, pp. 906–918, 2002.

[3] X. Michalet, A. N. Kapanidis, T. Laurence, F. Pinaud, S. Doose, M. Pflughoefft, and S. Weiss, "The power and prospects of fluorescence microscopies and spectroscopies," *Ann. Rev. Biophys. Biomol. Str.*, vol. 32, pp. 161–182, 2003.

[4] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Ann. Rev. Biomed. Eng.*, vol. 2, pp. 315–337, 2001.

[5] M. Morelock, E. Hunter, T. Moran, S. Heynen, C. Laris, M. Thieleking, M. Akong, I. Mikic, S. Callaway, R. DeLeon, A. Goodacre, D. Zacharias, and J. Price, "Statistics of assay validation in high throughput cell imaging of nuclear factor κB nuclear translocation," *ASSAY Drug Dev. Tech.*, vol. 3, pp. 483–499, 2005.

[6] T. R. Jones, A. E. Carpenter, and P. Golland, "Voronoi-based segmentation of cells on image manifolds," *Lecture Notes in Computer Science*, pp. 535–543, 2005.

[7] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[8] F. Meyer, "Topographic distance and watershed lines," *Signal Process.*, vol. 38, no. 1, pp. 113–125, 1994.

[9] A. Krtolica, C. O. Solorzano, S. Lockett, and J. Campisi, "Quantification of epithelial cells in coculture with fibroblasts by fluorescence image analysis," *Cytometry*, vol. 49, pp. 73–82, 2002.

[10] C. Wählby, "Algorithms for applied digital image cytometry," Ph.D. dissertation, Uppsala Univ, Sweden, 2003.

[11] T. R. Reed and J. M. H. du Buf, "A review of recent texture segmentation and feature extraction techniques," *CVGIP: Image Understand.*, vol. 57, no. 3, pp. 359–372, 1993.

[12] G. Koepfler, C. Lopez, and J. M. Morel, "A multiscale algorithm for image segmentation by variational method," *SIAM J. Numer. Anal.*, vol. 31, no. 1, pp. 282–299, 1994.

[13] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, Nov. 2004.

[14] C. Ballester, C. Caselles, and M. Gonzales, "Affine invariant segmentation by variational method," *SIAM J. Appl. Math*, vol. 56, no. 1, pp. 294–325, 1996.

[15] T. Pavlidis, "Segmentation of pictures and maps through functional approximation," *Comput. Graph. Image Process.*, vol. 1, pp. 360–372, 1972.

[16] A. Sarti, C. O. Solorzano, S. Lockett, and R. Malladi, "A geometric model for 3-D confocal image analysis," *IEEE Trans. Biomed. Eng.*, vol. 47, pp. 1600–1609, 2000.

[17] B. Appleton and H. Talbot, "Globally optimal geodesic active contours," *J. Math. Imag. Vis.*, vol. 23, no. 1, pp. 67–86, 2005.

[18] L. Coulot, H. E. Kirschner, A. Chebira, J. M. F. Moura, J. Kovačević, E. G. Osuna, and R. F. Murphy, "Topology preserving STACS segmentation of protein subcellular location images," in *Proc. IEEE Int. Symp. Biomed. Imaging*, Arlington, VA, Apr. 2006, pp. 566–569.

[19] C. Pluempitiwiriyawej, J. M. F. Moura, Y. Wu, and C. Ho, "STACS: A new active contour scheme for cardiac MR image segmentation," *IEEE Trans. Med. Imag.*, vol. 24, no. 5, pp. 593–603, May 2005.

[20] G. Srinivasa, M. C. Fickus, and J. Kovačević, "Multiscale active contour transformations for the segmentation of fluorescence microscope images," in *Proc. SPIE Conf. Wavelet Appl. Signal and Image Proc.*, San Diego, CA, Aug. 2007, vol. 6701:18, pp. 1–15.

[21] T. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[22] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, pp. 179–195, 1996.

[23] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 271–293, 2002.

[24] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, pp. 884–900, 1996.

[25] B. Song and T. Chan, Fast Algorithm for Level Set Based Optimization, Univ. California, Los Angeles, 2002 [Online]. Available: http://citeseer.ist.psu.edu/621225.html.

[26] F. Gibou and R. Fedkiw, "A fast hybrid k-means level set algorithm," presented at the Int. Conf. Stat., Math., Related Fields Nov. 2005.

[27] G. Papandreou and P. Maragos, "Multigrid geometric active contour models," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 229–240, Jan. 2007.

[28] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," *Proc. IEEE*, vol. 84, pp. 626–638, 1996.

[29] A. F. Laine, "Wavelets in temporal and spatial processing of biomedical images," *Ann. Rev. Biomed. Eng.*, vol. 2, pp. 511–550, 2000.

[30] P. Liò, "Wavelets in bioinformatics and computational biology: State of art and perspectives," *Bioinformatics*, vol. 19, no. 1, pp. 2–9, 2003.

[31] X. Bresson, P. Vandergheynst, and J.-P. Thiran, "Multiscale active contours," *Int. J. Comput. Vis.*, vol. 70, no. 3, pp. 197–211, 2006.

[32] B. Merriman, J. K. Bence, and S. J. Osher, "Motion of multiple junctions: A level set approach," *J. Comput. Phys.*, vol. 112, pp. 334–363, 1994.

[33] S. J. Ruuth, "Efficient algorithms for diffusion-generated motion by mean curvature," *J. Comput. Phys.*, vol. 144, pp. 603–625, 1998.

[34] S. J. Ruuth and B. Merriman, "Convolution-thresholding methods for interface motion," *J. Comput. Phys.*, vol. 169, pp. 678–707, 2001.

[35] F. Cao, *Geometric Curve Evolution and Image Processing*, ser. Lecture Notes on Math. New York: Springer, 2003, vol. 1805.

[36] J. Gravner and D. Griffeath, "Cellular automaton growth on $\mathbb{Z}^2$: Theorems, examples and problems," *Adv. Appl. Math.*, vol. 21, pp. 241–304, 1998.

[37] B. Leroy, I. L. Herlin, and L. D. Cohen, *Multi-Resolution Algorithms for Active Contour Models*, ser. Lecture Notes in Control and Inform. Sci., J. Sporring, M. Nielsen, L. M. J. Florack, and P. Johansen, Eds. Berlin, Germany: Springer, 1996, vol. 219.

[38] R. H. Webb and K. C. Dorey, *The Pixilated Image*, ser. Handbook of Biological Confocal Microscopy, J. B. Pawley, Ed., 2nd ed. New York: Springer, 1995.

[39] J. Donaldson and J. Lippincott-Schwartz, “Sorting and signaling at the Golgi complex,” *Cell*, vol. 101, no. 7, pp. 693–696, 2000.

[40] M. Aridor and L. A. Hannan, “Traffic jam: A compendium of human diseases that affect intracellular transport processes,” *Traffic*, vol. 1, no. 11, pp. 836–851, 2000.

[41] M. Aridor and L. A. Hannan, “Traffic jams II: An update of diseases of intracellular transport,” *Traffic*, vol. 3, no. 11, pp. 781–790, 2002.

[42] M. Puthenveedu and A. D. Linstedt, “Subcompartmentalizing the Golgi apparatus,” *Curr. Opin. Cell Biol.*, vol. 17, no. 4, pp. 369–375, 2005.

[43] Y. Guo and A. D. Linstedt, “COPII-Golgi protein interactions regulate COPII coat assembly and Golgi size,” *J. Cell Biol.*, vol. 174, no. 1, pp. 53–56, 2006.

[44] G. Srinivasa, M. C. Fickus, Y. Guo, A. D. Linstedt, and J. Kovačević, Active Mask Segmentation of Fluorescence Microscope Images [Online]. Available: http://www.andrew.cmu.edu/user/jelenak/Repository/08_SrinivasaFGLK/08_SrinivasaFGLK.html

[45] J. J. Bartko, “Measurement and reliability: Statistical thinking considerations,” *Schiz. Bull.*, vol. 17, no. 3, pp. 483–489, 1991.

[46] H. T. Nguyen and M. W. R. Boomgaard, “Watersnakes: Energy-driven watershed segmentation,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 3, pp. 330–342, 2003.

[47] K. Kennedy, C. Koelbel, and R. Schreiber, “Defining and measuring the productivity of programming languages,” *Int. J. High Perf. Comput. Appl.*, vol. 18, no. 4, pp. 441–448, 2004.

[48] ImageJ: Image Processing and Analysis in Java [Online]. Available: http://rsbweb.nih.gov/ij/.

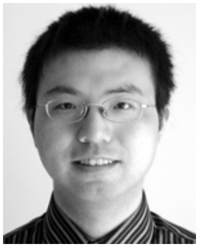
[49] L. DeRose and D. Padua, “Techniques for the translation of MATLAB programs into FORTRAN 90,” *ACM Trans. Progr. Lang. Syst.*, vol. 21, no. 2, pp. 286–323, 1999.

Gowri Srinivasa (S’05–M’09) received the B.E. degree in computer science and engineering from the P.E.S. Institute of Technology, Bangalore (affiliated with the Visvesvaraya Technological University, Karnataka), India, in 2004, and the Ph.D. degree in biomedical engineering from Carnegie Mellon University, Pittsburgh, PA, in 2008.

She is currently a Professor at the P.E.S. School of Engineering in Bangalore, India. Her current research interests include bioimaging, image processing, multiresolution techniques, and pattern recognition. She is an associate member of the Bio Imaging and Signal Processing Technical Committee of the IEEE Signal Processing Society.

Matthew C. Fickus (M’07) received the Ph.D. degree in mathematics from the University of Maryland, College Park, in 2001.

He spent three years at Cornell University, Ithaca, NY, in a postdoctoral position, supported by a VIGRE grant from the National Science Foundation. In 2004, he began his current position as an Assistant Professor at the Air Force Institute of Technology, Wright-Patterson Air Force Base, OH. His current research interests include Fourier series, wavelets, and frames, with an emphasis upon signal processing applications.



Yusong Guo received the B.S. degree in biology in June 2001 and the M.S. degree in botany in June 2003 from Nanjing University, China, and the Ph.D. degree in cell biology in May 2008 from Carnegie Mellon University, Pittsburgh, PA.

His research interests include analyzing the role of ßCOP-p115 interaction and Rab1b-p115 interaction in Golgi biogenesis and the role of COPII-Golgi protein interaction in regulating Golgi homeostasis. His projects are carried out on mammalian cells using cell-based assays, biochemical reconstitutions, molecular genetic experiments, and fluorescent imaging techniques.

Dr. Guo is a member of Sigma XI and a postdoctoral member of the American Society of Cell Biology. He is the first author of two papers published in *Molecular Biology of the Cell* and the *Journal of Cell Biology*, respectively.

Adam D. Linstedt received the Ph.D. degree in neuroscience at the University of California, San Francisco.

He is currently an Associate Professor in the Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, PA. His research interest is the molecular mechanisms that establish and maintain the membrane bounded subcellular compartments of the secretory and endocytic pathways.

Jelena Kovačević (S’88–M’91–SM’96–F’02) received the Dipl. Electr. Eng. degree from the EE Department, University of Belgrade, Yugoslavia, in 1986, and the M.S. and Ph.D. degrees from Columbia University, New York, in 1988 and 1991, respectively.

She is a Professor of biomedical engineering and electrical and computer engineering and the Director of the Center for Bioimage Informatics, Carnegie Mellon University, Pittsburgh, PA. Her research interests include bioimaging as well as multiresolution techniques such as wavelets and frames. From 1991–2002, she was with Bell Labs, Murray Hill, NJ. She was a co-founder and Technical VP of xWaveforms, based in New York City. She was also an Adjunct Professor at Columbia University. In 2003, she joined Carnegie Mellon University. She is a coauthor (with M. Vetterli) of the book *Wavelets and Subband Coding* (Prentice Hall, 1995).

Dr. Kovačević coauthored a top-10 cited paper in the *Journal of Applied and Computational Harmonic Analysis*, and the paper for which A. Mojsilovic received the Young Author Best Paper Award. Her paper on multidimensional filter banks and wavelets (with Martin Vetterli) was selected as one of the Fundamental Papers in Wavelet Theory. She received the Belgrade October Prize in 1986 and the E.I. Jury Award at Columbia University in 1991. She was the Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING. She served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, as a Guest Co-Editor (with I. Daubechies) of the Special Issue on Wavelets of the PROCEEDINGS OF THE IEEE, Guest Co-Editor (with M. Vetterli) of the Special Issue on Transform Coding of the *IEEE Signal Processing Magazine*, and Guest Co-Editor (with R. F. Murphy) of the Special Issue on Molecular and Cellular Bioimaging of the *IEEE Signal Processing Magazine*. She is/was on the Editorial Boards of the *Foundations and Trends in Signal Processing*, SIAM book series on Computational Science and Engineering, *Journal of Applied and Computational Harmonic Analysis*, *Journal of Fourier Analysis and Applications*, and the *IEEE Signal Processing Magazine*. She is a regular member of the NIH Microscopic Imaging Study Section. From 2000–2002, she served as a Member-at-Large of the IEEE Signal Processing Society Board of Governors. She is the Chair of the Bio Imaging and Signal Processing Technical Committee. She was the General Chair of ISBI 06, General Co-Chair (with V. Goyal) of the DIMACS Workshop on Source Coding and Harmonic Analysis and General Co-Chair (with J. Allebach) of the Ninth IMDSP Workshop. She is/was a plenary/keynote speaker at the “20 Years of Wavelets” 09, European Women in Mathematics 09, MIAAB Workshop 07, Statistical Signal Processing Workshop 07, Wavelet Workshop 06, NORSIG 06, ICIAR 05, Fields Workshop 05, DCC 98, as well as SPIE 98.