

# Images as Occlusions of Textures: A Framework for Segmentation

Michael T. McCann\*, *Student Member, IEEE*, Dustin G. Mixon, Matthew C. Fickus, *Member, IEEE*,  
Carlos A. Castro, John A. Ozolek, Jelena Kovačević, *Fellow, IEEE*

**Abstract**—We propose a new mathematical and algorithmic framework for unsupervised image segmentation, which is a critical step in a wide variety of image processing applications. We have found that most existing segmentation methods are not successful on histopathology images, which prompted us to investigate segmentation of a broader class of images, namely those without clear edges between the regions to be segmented. We model these images as occlusions of random images, which we call textures, and show that local histograms are a useful tool for segmenting them. Based on our theoretical results, we describe a flexible segmentation framework that draws on existing work on non-negative matrix factorization and image deconvolution. Results on synthetic texture mosaics and real histology images show the promise of the method.

**Index Terms**—image segmentation, occlusion models, texture, local histograms, deconvolution, non-negative matrix factorization

## I. INTRODUCTION

Image segmentation is the process of locating the boundaries between visually distinct regions in an image, thereby partitioning the pixels. Applications of image segmentation are numerous: from remote sensing [1] and video processing [2] to non-destructive testing [3]. Within the biomedical field, segmentation is used with diverse imaging modalities such as MRI [4], both light microscopy [5]–[7] and electron microscopy [8], ultrasound [9], and many others to identify regions at all scales from organelles to organisms.

Copyright © 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org).

The authors gratefully acknowledge support from the NSF through awards 0946825 and 1017278, the Achievement Rewards for College Scientists Foundation Scholarship, the John and Claire Bertucci Graduate Fellowship, the Philip and Marsha Dowd Teaching Fellowship, and the CMU Carnegie Institute of Technology Infrastructure Award.

Michael McCann is with the Center for Bioimage Informatics and the Dept. of Biomedical Engineering at Carnegie Mellon University, Pittsburgh, PA, [mtmccann@cmu.edu](mailto:mtmccann@cmu.edu).

Dustin Mixon and Matthew Fickus are with the Dept. of Mathematics and Statistics at the Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, [Matthew.Fickus, Dustin.Mixon}@afit.edu](mailto:{Matthew.Fickus, Dustin.Mixon}@afit.edu). The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Carlos Castro is with the Dept. of Obstetrics and Gynecology at Magee-Womens Research Institute and Foundation of the University of Pittsburgh, Pittsburgh, PA, [ccastro@mwri.magee.edu](mailto:ccastro@mwri.magee.edu).

John Ozolek is with the Dept. of Pathology at Children’s Hospital of Pittsburgh of the University of Pittsburgh, Pittsburgh, PA, [ozolja@upmc.edu](mailto:ozolja@upmc.edu).

Jelena Kovačević is with the the Dept. of Electrical and Computer Engineering, Center for Bioimage Informatics, and the Dept. of Biomedical Engineering at Carnegie Mellon University, Pittsburgh, PA, [jelenak@cmu.edu](mailto:jelenak@cmu.edu).

## A. Previous Work

There exists a wide variety of classic approaches to generic image segmentation, including graph cuts [10], active contours [11], level sets [12], Gabor filtering and clustering [13], random fields [14], watersheds [15], region growing [16], and mean shift [17]. In addition, there also exist engineered segmentation systems, such as BlobWorld [18], JSEG [19], EDISON [20], and CTex [21]. Broadly, these methods vary in how the segmentation regions are parametrized and whether edge or region information is used.

Even with this plethora of methods to try, working on a new segmentation problem is not trivial. Each existing method makes assumptions about the images it aims to segment. When these assumptions are met, the method works, and when they are not, it fails. Because many of these assumptions are implicit, selecting a method to use on a new segmentation problem involves educated guessing. When no suitable method can be found, a new method is designed. This is especially true for biomedical imaging, where the diversity of imaging modalities and analysis objectives means that algorithms are often designed for each specific application. For example, the active shape models in [22] aim to segment the complex shapes in CT scans, the active masks in [6] work for the punctate appearance of fluorescence microscopy images, and the multiple model approach in [9] handles the problems of speckle and motion in ultrasound images of the heart.

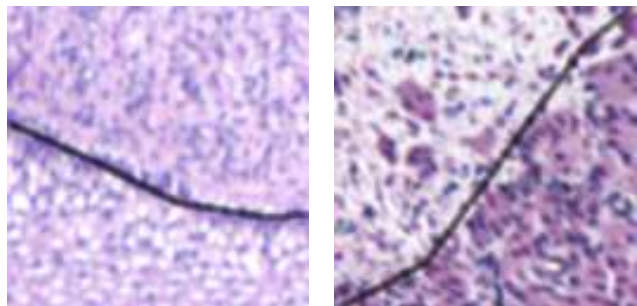


Fig. 1. Examples of difficult tissue boundaries in teratoma images (black lines denote tissue boundaries as drawn by a pathologist). The lack of distinct edges between regions makes automated segmentation difficult, and even experts cannot reliably localize the boundaries at pixel level.

In this work, we were inspired by the problem of segmenting tissues in bright field microscopy images of hematoxylin and eosin (H&E)-stained slices of teratoma tumors (see Figure 1 for examples; more details on this dataset can be found

in [23]). Other works that address segmentation of histology images include [24], where the authors use intensity neighborhoods as features to segment bone, cartilage, and fat, and [25], where the authors use the cooccurrence of different tissue components to segment cancerous glands specifically in colon biopsies. Our working hypothesis is that this task is difficult because tissues are complicated, while tissue boundaries are sometimes subtle and not marked by edges. Our previous work [26] began exploring models for these complicated tissues and proposed a simple supervised segmentation method for them.

In the current work, we propose a new model for images formed from multiple tissues. Working from this model, we design a new unsupervised segmentation method based on local histograms that is well-suited to a broad class of images with poorly defined boundaries. We expect this class to include images such as aerial photos of grass and trees (“Where is the edge of the forest?”), photographs of crowds (“Where is the edge of the crowd?”), and, as is our focus in this paper, histology images (“Where is the edge of the adipose tissue?”).

## B. Contributions

The main contributions of this paper are as follows: 1) We propose a *mathematical framework* for image segmentation which models images as occlusions of textures. Given an image formed according to this model, we prove that its local value histograms will approximately be convex combinations of the value distributions of its component textures. 2) Based on this result, we present a new *algorithmic framework* for image segmentation based on histogram factorization and deconvolution.

## C. Outline

The outline of the paper is as follows: In Section II, we introduce local histograms, occlusions, and necessary notation. We then describe our proposed model and prove the main theorem. In Section III, we propose an algorithmic framework for image segmentation based on this theorem and describe a possible implementation of one instance of such an algorithm. We present experimental results of a comparison of our algorithm to other segmentation algorithms in Section IV and conclude in Section V.

## II. MATHEMATICAL FRAMEWORK

In this section, we introduce necessary notation, then build the texture model as follows: We first define dependence decay, which describes how well a texture can be characterized by its local histograms, show how to construct textures with a known dependence decay, and explore dependence decay in real textures. Based on this, we propose modeling images as occlusions of textures. Under this model, we present and prove our main result, which describes how the local histograms of this class of images are mixtures of the value distributions of their component textures.

## A. Notation and Background

Let  $f$  be an image,  $f : X \rightarrow V$  where  $X$  is a discrete set of pixel locations and  $V$  is a discrete set of pixel values. For example, for a  $512 \times 512$  8-bit grayscale image,  $X = \mathbb{Z}_{512} \times \mathbb{Z}_{512}$  (we use  $\mathbb{Z}_K$  to denote the set of integers modulo  $K$ ) and  $V = \mathbb{Z}_{256}$ , while for a stack of ten  $1600 \times 1200$  8-bit RGB color images,  $X = \mathbb{Z}_{10} \times \mathbb{Z}_{1600} \times \mathbb{Z}_{1200}$  and  $V = \mathbb{Z}_{256} \times \mathbb{Z}_{256} \times \mathbb{Z}_{256}$ . Clearly, the elements of  $V$  can be vectors or scalars; since the set is always discrete, however, without loss of generality we treat it as a set of integers  $0, 1, \dots, |V| - 1$ .

We now define the local histogram and occlusion operators; our definitions are a slight adjustment of those presented in [26]. Define first the indicator function,

$$1_A(x) = \begin{cases} 1, & \text{for } x \in A; \\ 0, & \text{for } x \notin A. \end{cases}$$

Then, the *local histogram* transform of an image  $f$  is

$$L_w f(x, v) = \sum_{x' \in X} 1_{\{v\}}(f(x')) w(x - x'), \quad (1)$$

for  $x \in X$  and  $v \in V$ , and where  $w$  is an averaging filter that sums to one. For example, with a constant averaging filter  $w$  of size  $3 \times 3$ , the local histogram transform  $L_w f(x, v)$  gives the fraction of the pixels in the  $3 \times 3$  neighborhood around  $x$  that have value  $v$ .

We call a function  $\sigma : X \rightarrow \{0, 1, \dots, N - 1\}$  a *labeling function*. We define the *occlusion* of a set of images  $\{f_n\}_{n=0}^{N-1}$  with respect to a labeling function  $\sigma$  as

$$O_\sigma \{f_n\}_{i=0}^{N-1}(x) = \sum_{n=0}^{N-1} 1_{\{\sigma(x)\}}(f_n(x)) \quad (2)$$

for  $x \in X$ . In other words, an occlusion of a set of images is itself an image that matches one of its component images at each pixel; which image shows through at each pixel is governed by  $\sigma$ . Note that we use the term *occlusion* because we view the set of images as a stack with one image blocking, or occluding, the others at each pixel.

## B. Flat Textures

We now define a texture as we use it in our work; note that it is simply a random-valued image and that many other texture definitions exist.

**Definition 1** (Texture). A texture  $F = \{F(x)\}_{x \in X}$  is a  $V$ -valued random field indexed by pixel locations,  $X$ .

In other words, for each  $x \in X$ ,  $F(x)$  is a  $V$ -valued random variable; one realization of a texture is an image. Associated with each of these random variables is a probability mass function,  $p_{F(x)} : V \rightarrow [0, 1]$ ; the probability that  $F(x)$  is equal to  $v$  is  $p_{F(x)}(v)$ . While each  $F(x)$  may have its own unique probability mass function, we find it useful to consider only those textures for which these functions are the same, which we call *flat textures*.

**Definition 2** (Flat texture). A texture  $F$  is flat when  $p_{F(x_i)}(v) = p_{F(x_j)}(v) = p_F(v)$ , for all  $x_i, x_j \in X$  and  $v \in V$ .

In other words, a flat texture is one where the distribution of values is the same at every pixel. Note that this does not imply that the random variables  $F(x_i)$  are independent, only that they have the same distribution. Flat textures may have complex and spatially varying patterns of dependence; we will further explore this dependence in the following sections.

Flatness is not a highly restrictive condition. In fact, it is a property of most real-world textures. For example, take  $F$  to be the texture for  $128 \times 128$  color images of grass. Intuitively, we would assign one color distribution to  $F$ , with peaks for brown, yellow, and green. We would not say that grass images are more likely to be brown in the bottom right corner or green at the top. The fact that colors do not appear more or less frequently in specific areas of the image means that this texture is flat.

### C. Histograms of Flat Textures

Many textures can be distinguished based solely on their value distribution,  $p_F(v)$ . For example, consider a grass texture and a sky texture;  $p_{\text{grass}}(v)$  would have a peak around green, while  $p_{\text{sky}}(v)$  would have peaks around blue and white. If our goal is to classify whether an image is of grass or sky, we could build a value histogram from the pixels in the image and compare it to our model of  $p_{\text{grass}}(v)$  and  $p_{\text{sky}}(v)$ .

We now consider more closely the relationship between the histograms of a realization of a flat texture and that texture's value distribution,  $p_F(v)$ . Let

$$\hat{p}_f^{\{x_i\}_{i=0}^{k-1}}(v) = \frac{1}{k} \sum_{i=0}^{k-1} 1_{\{v\}}(f(x_i)) \quad (3)$$

be a histogram built from  $k$  pixels at arbitrary unique locations  $x_0, x_1, \dots, x_{k-1}$  of the image  $f$ . If  $k$  is large enough, we expect that  $\hat{p}_f^{\{x_i\}}(v)$  will estimate  $p_F(v)$ ; intuitively, sampling  $f$  over a small region will give a worse estimate than sampling  $f$  over a large one. The quality of the estimate will be determined by the number and position of the pixels and the way the statistical dependence between pixels of  $F$  changes over space. Moreover, for some textures, such an estimate may never be good. We quantify how quickly these estimates improve by introducing the notion of *dependence decay*.

**Definition 3** (Dependence decay). *A flat texture  $F$  exhibits dependence decay with complexity  $a$  if, with probability  $1 - \delta$  and for any  $v \in V$  and  $k \leq |X|$ ,*

$$|\hat{p}_f^{\{x_i\}_{i=0}^{k-1}}(v) - p_F(v)| \leq ak^{-1/2},$$

with  $0 < a$  and  $0 < \delta \ll 1$ .

The value  $\delta$  is a small positive number so that  $1 - \delta$  is our standard of high probability. If the complexity,  $a$ , is small, then  $F$  is simple in the sense that even histograms formed from a small number of pixels approximate  $p_F$  well.

### D. Constructing Dependence-Decay Textures

Though any texture can exhibit dependence decay with a very loose bound (for example, select  $a$  such that  $a|X|^{-1/2} > 1$ ), we focus on the class of textures for which the dependence

between pixels decays sufficiently so that this bound is useful, that is,  $ak^{-1/2} \ll 1$  even when  $k \ll |X|$ ; informally, we will refer to these as *dependence-decay textures*. At this point, it is natural to wonder whether there exist any dependence-decay textures. To find the answer, we explore a property of textures we call *persistence length* and show that when a texture's persistence length is short enough, it is dependence-decay.

**Definition 4** (Persistence length). *A texture  $F$  has a persistence length  $\epsilon$  if, for any two pixel locations,  $x_i$  and  $x_j$ , the random variables  $F(x_i)$  and  $F(x_j)$  are independent whenever  $\|x_i - x_j\|_1 > \epsilon$ .*

Persistence length is another way of thinking about the scale of a texture. For example, consider a texture  $F$  that generates images of colored pebbles. If we know that the value of a sample of this texture at pixel location  $x_i$  is blue, we can expect the value at pixel location  $x_j$  to be blue as well when  $x_i$  and  $x_j$  are close; when these locations are far enough apart, however,  $f(x_i)$  no longer gives any predictive information about  $f(x_j)$ . The distance at which  $f(x_i)$  stops being helpful in predicting  $f(x_j)$  is  $F$ 's persistence length. In this example, it is the diameter of the pebbles.

Persistence length is related to dependence decay, since statistically independent samples are the best ones with which to estimate a texture's value distribution. We make this relationship explicit with a theorem; the proof is given in Appendix A.

**Theorem 1** (Persistence length and dependence decay). *A flat texture with persistence length  $\epsilon$  exhibits dependence decay with complexity  $a = (-2A_\epsilon \log \frac{1}{2}(1 - \sqrt[4]{1 - \delta}))^{1/2}$ , with  $A_\epsilon = 2\epsilon^2 + 2\epsilon + 1$ .*

If we could create textures with a given persistence length, this theorem would allow us to establish their dependence decay properties as well. One straightforward way of doing this is to use a dead leaves model; that is, generate images by layering templates of random size, shape, position, and value on top of each other (see [27] for a more thorough description). Textures defined in this way have a persistence length equal to the  $\ell^1$ -diameter of their largest template, since pixels separated by more than this distance cannot belong to the same template and are therefore independent.

### E. Dependence Decay in Real Images

We now discuss how the above theory applies to real images. As our example textures, we take grayscale images from the Prague Texture Segmentation Benchmark [28].

For each image in the dataset, we define a corresponding  $F$  that assigns equal probability to this image and all its circular translates. This means  $p_F(v)$  is equal to the global histogram of the image. To show that  $F$  is a dependence-decay texture we must set a probability  $\delta$ , then for each  $k = 1, 2, \dots$ , find the arrangement of points,  $x_0, x_1, \dots, x_{k-1}$ , that maximizes  $|\hat{p}_f^{\{x_i\}}(v) - p_F(v)|$  in at least a fraction  $\delta$  of the realizations of  $F$ . If this maximum error,

$$E(k) = \max_{x_0, x_1, \dots, x_{k-1}} |\hat{p}_f^{\{x_i\}_{i=0}^{k-1}}(v) - p_F(v)| \quad (4)$$

can be bounded usefully by a function  $ak^{-1/2}$ , then  $F$  is a dependence decay texture.

In practice, we cannot check every possible arrangement of  $k$  pixels even for small  $k$ , so we make the simplifying assumption that, in the worst case, the  $x_i$  will be 8-connected and perform a greedy search for worst-case arrangements. This assumption is reasonable for natural images because adjacent pixels are nearly always more correlated than those that are far apart. To further speed up the computation, we crop the image to a size of  $100 \times 100$  pixels and quantize it to 8 values.

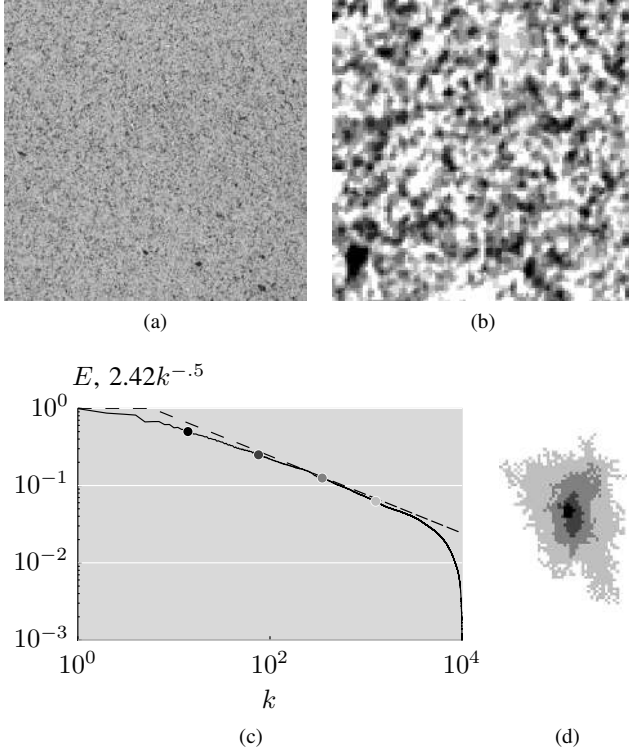


Fig. 2. An example of dependence decay in one texture from the Prague set [28]. The original texture image (a) is cropped and quantized (b) before the analysis. The plot in (c) shows how the approximation error (4) (solid line) is bounded by the function  $2.42k^{-1/2}$  (dashed line). The regions shown in (d) are four worst-case arrangements (described in Section II-E). When pixels sampled in these patterns are used to estimate the color distribution of the texture, they lead to errors of  $2^{-1}$ ,  $2^{-2}$ ,  $2^{-3}$ , and,  $2^{-4}$  (decreasing error from black to light gray). The corresponding points are marked with circles in (c).

Our experiments show that many of these images do exhibit dependence decay, see Figure 2 for an example. This supports the claim that many real-world textures exhibit dependence decay. For the periodic textures in the dataset, the complexity  $a$  is large because pixels are highly correlated, see Figure 3 for an example. If the search were not constrained to contiguous regions of pixels, the complexity would be even higher. This is because when the offset between samples matches the period, the samples are completely dependent on each other. We will see in the next section that the high  $a$  value associated with periodic textures means that they are not usefully governed by our main result, Theorem 2. This essentially says that periodic textures can be combined into pathological images that are impossible to segment. We will stress here, however,

that periodic textures can still be segmented by the algorithm we present in Section III, because the worst case error only happens with very specific corner cases that are unlikely to occur in practice.

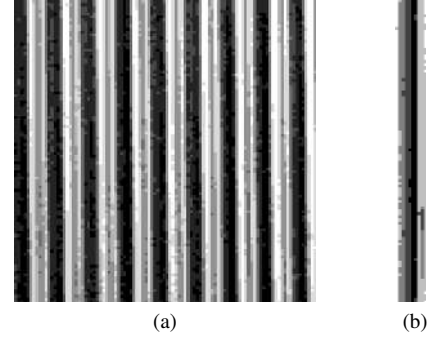


Fig. 3. An example of dependence decay in a periodic texture from the Prague set [28], shown cropped and quantized in (a). The regions shown in (b) are four worst-case arrangements (described in Section II-E), increasing in size from black to light gray. Sampling pixels in these vertical stripes leads to very poor estimates of the true color distribution of this texture because the sample may e.g. contain only black pixels, omitting the lighter pixels in the image.

#### F. Oclusions of Dependence-Decay Textures

We now come to the main result of the paper. We propose modeling images as oclusions of realizations of textures,  $O_\sigma\{f_n\}_{n=0}^{N-1}$ , where the component textures,  $F_0, F_1, \dots, F_{N-1}$ , and the labeling function,  $\sigma$ , are not known. Under this model, segmentation is equivalent to recovering  $\sigma$ .

Figure 4 illustrates modeling a natural image as such an occlusion. Starting with an original image as in Figure 4a, imagine there exist three textures,  $F_{\text{sky}}$ ,  $F_{\text{window}}$ , and  $F_{\text{brick}}$ , each generating images of the corresponding texture, as in Figure 4b. We can then use a labeling function  $\sigma$  (Figure 4c) to create a new image  $f_{\text{lighthouse}} = O_\sigma\{f_{\text{sky}}, f_{\text{window}}, f_{\text{brick}}\}$  (Figure 4d).

If we use oclusions of textures as our model for images, then we can define segmentation as the problem of taking an image and recovering the labeling function that was used to generate it. If the component textures of the image are dependence-decay, then a reasonable approach is to examine the local histograms of  $f$ , since these histograms well approximate the true distribution of values in each texture. For example, in Figure 4a, local histograms taken from the upper right corner will approximate  $p_{\text{sky}}$ , while those from the lower left corner will approximate  $p_{\text{brick}}$ . Local histograms taken around the window will be mixtures of  $p_{\text{brick}}$  and  $p_{\text{window}}$ . The following theorem formalizes this process; the proof is given in Appendix B.

**Theorem 2** (Local histograms of oclusions are mixtures of component value distributions). *Given a set of dependence-decay textures  $\{F_0, F_1, \dots, F_{N-1}\}$  with complexities  $a_0, a_1, \dots, a_{N-1}$ , a constant averaging window  $w$ , and a labeling function  $\sigma$ , then with probability  $(1 - \delta)^N$ , and for any  $x \in X$*

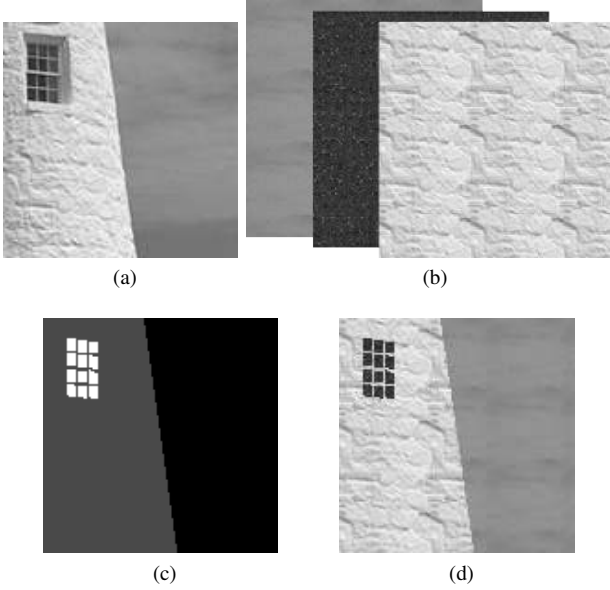


Fig. 4. Modeling of an image (a) as an occlusion of realizations of textures (b) according to a labeling function (c), resulting in (d).

and  $v \in V$ ,

$$\left| L_w O_\sigma \{f_n\}_{n=0}^{N-1}(x, v) - \sum_{n=0}^{N-1} [w * 1_{\{n\}}(\sigma)](x) p_{F_n}(v) \right| \leq a \left( \frac{|w|}{N} \right)^{-1/2}, \quad (5)$$

with  $|w| = \sum_{x \in X} 1_{(0,1]}(w(x))$ ,  $a = \max\{a_0, a_1, \dots, a_{N-1}\}$ , and  $\delta$  the threshold for high probability selected in Definition 3.

Here we use the notation  $(0, 1]$  to indicate the interval  $\{x \mid 0 < x \leq 1\}$ .

In other words, if an image is formed from the occlusion of realizations of several dependence-decay textures, then each of its local histograms will be approximately a convex combination of the true value distributions of those textures. The amount that each texture contributes to a histogram is proportional to how much of that texture exists in the neighborhood of the local histogram. This theorem offers a key insight about how to do segmentation because it relates the local histograms of  $f$ , which we can calculate, to the labeling function  $\sigma$ , which we want to discover.

### III. PROPOSED ALGORITHM

Based on our discussion, the task of segmentation can now be seen as finding the labeling function  $\sigma$  using local histograms of  $f$ , which we can calculate. Thus, based on Theorem 2, the task of segmenting an image modeled as occlusions of realizations of textures,  $f = O_\sigma \{f_n\}_{n=0}^{N-1}$ , into  $N$  regions can be approached as an optimization

$$\arg \min_{\sigma} \left\| L_w f - \sum_{n=0}^{N-1} [w * 1_{\{n\}}(\sigma)] \hat{p}_f^{X_n} \right\|, \quad (6)$$

where  $X_n = \{x \mid \sigma(x) = n\}$  is the set of pixels belonging to the  $n$ th region and  $\hat{p}_f^{X_n}(v)$  is a histogram formed from the pixels in  $X_n$  as defined in (3). This optimization finds a labeling function,  $\sigma$ , that splits the image into  $N$  regions. These regions are such that for each  $x \in X$ , the local histogram,  $L_w f(x, v)$ , is approximated by a convex combination of the value histograms of those regions,  $\hat{p}_f^{X_n}(v)$ , with the weights of this convex combination determined by the amount of each label in the neighborhood of the histogram,  $[w * 1_{\{n\}}(\sigma)](x)$ .

Our approach is philosophically similar to the method in [29], which uses an active contour to separate the image into a foreground and a background region that have maximally distinct color distributions. This method differs from ours in two key ways: First, our method does not find maximally distinct distributions, rather it finds distributions that are consistent with the local histograms taken inside their corresponding region. Second, our method does not parametrize the boundary between regions as a level set, allowing it to more naturally handle segmentation of more than two regions. In short, though both methods focus on color distributions rather than edges, they actually optimize different quantities in different ways.

Rather than solve (6) directly, we approximate it via variable splitting [30], [31] and relaxation [32] as

$$\arg \min_{\substack{\sigma, p_0, p_1, \dots, p_{N-1} \\ \alpha_0, \alpha_1, \dots, \alpha_{N-1}}} \left\| L_w f - \sum_{n=0}^{N-1} \alpha_n p_n \right\| + \sum_{n=0}^{N-1} \lambda_n \|w * 1_{\{n\}}(\sigma) - \alpha_n\| + \sum_{n=0}^{N-1} \mu_n \left\| \hat{p}_f^{X_n} - p_n \right\|, \quad (7)$$

subject to

$$\sum_{v \in V} p_n(v) = 1, \quad \sum_{n=0}^{N-1} \alpha_n(x) = 1, \quad p_n(v), \alpha_n(x) \geq 0,$$

where  $\{\lambda_n\}$  and  $\{\mu_n\}$  are Lagrange multipliers. The first term of the optimization finds histograms  $p_0, p_1, \dots, p_{N-1}$  that can be combined with weights  $\alpha_0, \alpha_1, \dots, \alpha_{N-1}$  to create the local histograms of  $f$ . The second term finds an indicator function for each region in the image  $1_{\{n\}}(\sigma(x))$ , such that the blurred version of this indicator is similar to the corresponding weight image  $\alpha_n$ . The third term keeps the histograms  $p_0, p_1, \dots, p_{N-1}$  close to the empirical histograms of the regions  $\hat{p}_f^{X_n}$ .

The benefit of working with (7) is that the third term is straightforward and the first two terms are well-studied problems, known as non-negative matrix factorization (NMF) [33], [34] and image deconvolution (deblurring) [35], [36], respectively. To find a local minimum of (7), we draw on the wide variety of existing approaches for each of these subproblems to iteratively minimize each of the three terms.

Figure 5 shows the block diagram of our proposed algorithm and the following pseudocode gives more details:

- 1: **procedure**  $[\sigma] = \text{SEGMENT}(f)$
- 2:   compute the local histogram transform of  $f$
- 3:   factor the histograms to initialize  $\{\alpha_n\}$  and  $\{p_n\}$

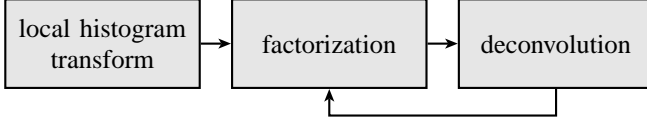


Fig. 5. Block diagram of the proposed algorithm.

```

4:   while not converged do
5:     deconvolve  $\{\alpha_n\}$  to estimate  $\sigma$ 
6:     use  $\sigma$  to compute  $\hat{p}_f^{X_n}$ 
7:     set  $p_n = \hat{p}_f^{X_n}$ 
8:     use  $p_n$  and the histograms to calculate  $\{\alpha_n\}$ 
9:   end while
10: end procedure

```

The following subsections present the details of one possible implementation of each of these steps; others are possible.

#### A. Local Histogram Transform

Implementation of the local histogram transform is straightforward based on its definition: each level of the local histogram can be computed as a single convolution. The size and shape of the window  $w$  over which local histograms are computed is an important parameter of the method. In general, when  $w$  has small support, the resulting segmentation will have better-localized boundaries, but may be noisy. Making  $w$  larger smooths the boundaries but also makes the segmentation less susceptible to noise.

The main challenge in computing the local histogram transform is memory, since an image can have 24 bits of color and 2 million pixels, meaning that the full local histogram of the image would have  $2^{24} \times 2 \times 10^6 \approx 3.4 \times 10^{13}$  values in it. Luckily, real-world images rarely make meaningful use of this color depth. We can therefore safely quantize the image down to tens of values. The result is a new image,  $f' : X \rightarrow V'$  with  $|V'| \ll |V|$ . The specific quantization method can be chosen to fit the problem; we used k-means clustering.

The quantization step can also be understood as a way to control the value bandwidth of the local histograms. Using empirical histograms rather than, for example, kernel density estimation implicitly assumes that  $p_F(v_i)$  and  $p_F(v_j)$  are unrelated, no matter how similar the values  $v_i$  and  $v_j$  are. This assumption is more accurate for a well-quantized image than one with 24 bits of color.

#### B. Factorization

To compute the initial factorization of the local histograms of  $f'$ , we arrange them into a matrix,  $A$ , of size  $|V'| \times |X|$ . Our goal is to express  $A$  as  $A = HW$ , where  $H$  is a  $|V'| \times N$  matrix of histograms and  $W$  is an  $N \times |X|$  matrix of weights. We frame this as the optimization

$$\begin{aligned}
 & \arg \min_{H, W} \|A - HW\| \\
 & \text{subject to } H_{ij} \geq 0, W_{ij} \geq 0, \quad \text{for all } i, j, \\
 & \quad \sum_i H_{ij} = 1, \sum_j W_{ij} = 1,
 \end{aligned}$$

```

1: procedure  $[H, W] = \text{FACTOR\_ALS}(A)$ 
2:   initialize  $H$  with random values
3:   while not converged do
4:      $W = (HH^T)^{-1}HA$ 
5:     set negative values in  $W$  to zero
6:      $H = (W^TW)^{-1}W^T A$ 
7:     set negative values in  $H$  to zero
8:     normalize  $H$  and  $W$ 
9:   end while
10: end procedure

```

Fig. 6. Alternating least-squares method for non-negative matrix factorization.

which we solve with a variant of the alternating least-squares method [37], described in Figure 6.

To increase the robustness of the algorithm, we repeat it multiple times with different random initializations of  $H$  and keep the result with the lowest error. To reduce the computational cost, we factor only a random subset of the histograms in each iteration.

We used alternating least-squares for its simplicity, but repeat that our framework is flexible: any NMF method can be used. Other approaches in the literature may be faster or allow for the inclusion of priors. For example, encouraging  $W$  to be sparse would correspond to the assumption that each point in the image is only near a small number (one or two) texture regions. For more discussion of approaches to the NMF problem, see [34].

#### C. Deconvolution

Each row of the matrix  $W$  from the factorization step can be reshaped into an image of weights  $\alpha_n$  that represents a blurred version of one level of the labeling function,  $w * 1_{\{n\}}(\sigma)$ . To recover  $\sigma$  from these images, we must deconvolve them, that is, solve

$$\arg \min_{\sigma} \sum_{n=0}^{N-1} \|w * 1_{\{n\}}(\sigma) - \alpha_n\|. \quad (8)$$

The problem in (8) is a sum of  $N$  deconvolution problems with the added complexity that the image to be recovered in each of the problems,  $1_{\{n\}}(\sigma)$ , is  $\{0, 1\}$ -valued. In general, existing deconvolution methods can be adapted to this problem by relaxing  $\sigma$  to be a set of real-valued images,  $\{\hat{\sigma}_n\}$  with the constraint that  $\sum_{n=0}^{N-1} \hat{\sigma}_n(x) = 1$ . Then the joint deconvolution can be solved by working on each  $\hat{\sigma}_n$  in turn and renormalizing every few iterations. Then,  $\sigma$  can be recovered from  $\{\hat{\sigma}_n\}$  via  $\sigma(x) = \arg \max_n \hat{\sigma}_n(x)$ .

The design freedom our framework offers is especially valuable in the deconvolution step, as there exist numerous deconvolution methods that allow the use of prior knowledge about the image to be recovered. If we can describe the shape of the regions with a few parameters, parametric deconvolution can achieve excellent results. In less constrained cases, many looser priors may be used. For example, total variation regularization [38] promotes solutions with a blocky appearance.

In this work, we explore two deconvolution methods: The first method is to assume that  $\hat{\sigma}_n = \alpha_n$  for all  $n$ . This

means that the label at a given point corresponds to the histogram with the largest weight at that point, so  $\sigma(x) = \arg \max_n \alpha_n(x)$ . This method, while basic, is useful because it is fast and requires no particular knowledge about the properties of the labels we want to recover. The downside is that the resulting labelings tend to be smooth.

The second method is parametric deconvolution. We assume that the regions to be labeled,  $\{1_{\{n\}}(\sigma)\}$ , are the Voronoi cells of three seed pixel locations. We then deconvolve using gradient descent on the seed pixel locations. This method is useful only on images that fit this model; we include it as a proof of concept for parametric deconvolution.

#### D. Selecting the Number of Textures

In what we have presented so far, the number of textures  $N$  must be known beforehand. While this may be sufficient in certain applications, we may also want a method for selecting an appropriate  $N$  automatically from the input image. Selecting the number of textures is essentially the same problem as selecting the number clusters for any unsupervised clustering algorithm. There are numerous approaches to this problem; for an overview see Section 3.3 in [39]. In general these methods replace the parameter  $N$  with one or more new parameters that control the tradeoff between having tight clusters and having a small number of clusters.

In our work, we have experimented with a simple method in which we set a parameter  $r < 1$  that controls how much we expect the model error to decrease when the number of textures increases by one. We begin by setting  $N = 1$ , running the whole algorithm, and measuring the model error as the value of the norm in (6). We then increase  $N$  by one and repeat this process. As soon as the error at  $N_{i+1}$  fails to be smaller than  $r$  times the error at  $N_i$ , we select  $N_i$  as the final number of textures.

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

We compared our segmentation method to four other methods from the literature on synthetic test images and on real histology images; results are given in Table I and example segmentations in Figures 8, 7, 9, and 10. We now describe the test data, the specifics of these experiments, and each of the comparison methods and discuss the results of the comparison. The test data and MATLAB code for our proposed algorithm is available in the reproducible research compendium for this article, [40].

### A. Datasets

The three datasets we used for testing are as follows:

The *random texture* dataset is a synthetic dataset designed to illustrate the difficulty of segmenting edgeless images. To generate these images, we choose three seed pixel locations at random and let the ground truth regions be the Voronoi cells formed by these seeds. For each region, we select a color distribution at random and select each pixel independently from this distribution (since each pixel is independent of its neighbors, these images have a persistence length of 1). The

TABLE I  
COMPARISON RESULTS IN TERMS OF THE RAND INDEX MEAN AND STANDARD DEVIATION. THE VALUE 1 INDICATES PERFECT AGREEMENT WITH THE GROUND TRUTH.

Method	Dataset		
	random texture	histology	Prague
Normalized	0.785 ± 0.122	0.736 ± 0.176	0.812 ± 0.064
JSEG	0.943 ± 0.067	0.751 ± 0.141	<b>0.839</b> ± 0.056
EDISON	0.697 ± 0.084	0.798 ± 0.148	0.784 ± 0.077
Efficient	0.566 ± 0.189	0.619 ± 0.129	0.823 ± 0.070
gPb-owt-ucm	0.707 ± 0.206	0.787 ± 0.130	0.835 ± 0.062
ORTSEG	0.989 ± 0.002	<b>0.830</b> ± 0.144	0.766 ± 0.073
ORTSEG-D	<b>0.992</b> ± 0.004		

resulting images can be segmented by eye, but lack distinct edges between regions. At the same time, the high frequency content within regions can be mistaken for edges. We created 25 such images. The number of unique colors in each image is 8 and the size of each image is  $128 \times 128$ .

The *histology* dataset consists of color images of H&E-stained tissue (for more details on these images, see [23]). The ground truth was drawn by an expert pathologist via visual inspection. From a set of 36 images comprising more than 20 tissue types, we manually selected 36  $128 \times 128$  subimages such that each contained only two tissues. The aim of this process was to include only subimages that had very accurate ground truth labeling. This dataset is relatively small because only trained pathologists can reliably label histology images and creating pixel-wise labels is very time-consuming; we hope to expand the dataset in the future. In the histology dataset, some regions have clearly defined boundaries while many do not, we expect that our edgeless approach should work well.

The *Prague* dataset contains the grayscale mosaics from the Prague Texture Segmentation Benchmark [28]. These images were designed to be similar to the natural images in datasets like the Berkeley Segmentation Dataset and Benchmark [41], while avoiding the ambiguity in the ground truth that comes with human labeling. We take this dataset to be indicative of performance on natural image segmentation; it is *not* representative of the class of edgeless images for which our algorithm is designed. The dataset contains 20 images of size  $512 \times 512$  with the number of textures per image varying from 3 to 12.

### B. Algorithms Tested

We implemented two versions of the algorithm described in Section III:

*ORTSEG* (Occlusion of Random Texture SEGmenter) uses the maximum weight deconvolution method and therefore includes no strong prior information about the region boundaries. The parameters are the number of textures  $N$  (or the automatic selection parameter  $r$ ), the size of the window  $|w|$ , and the number of colors to quantize to  $|V'|$ .



*ORTSEG-D* uses parametric deconvolution. Since it makes strong assumptions on the region boundaries, it is only useful on the random texture dataset. The parameters are the same as those for *ORTSEG*.

We compare our two algorithms to five others:

*Normalized Cut* [10] forms a graph from an image by using pixels as nodes and assigning edge weights based on distance and color similarity. It segments by recursively finding graph cuts that maximize the similarity within and dissimilarity between regions. The only parameter is the number of regions.

*JSEG* [19] quantizes the colors in an image and then searches for segmentations that produce regions of uniform color distribution via region growing. *JSEG* does not rely on edge information, and is therefore the most similar to *ORTSEG* of these comparison methods. The parameters are the region merging threshold, the quantization threshold, and the number of scales.

*EDISON* [20] uses mean shift to cluster pixels in terms of their color and location. Edge information is included as a weight during the mean shift computation. The parameters are the spatial bandwidth, the range bandwidth, and the minimum region size.

*Efficient Graph-Based Segmentation* [42] is another graph-based algorithm. Pixels are used as nodes and edges are assigned between adjacent pixels based on difference in intensity. It groups pixels into regions when there is no evidence of a boundary between them. The parameters are the smoothing  $\sigma$ , scale  $k$ , and the minimum region size.

*gPb-owt-ucm* [41] detects local edges with a combination of color and texture features. Global information is combined with the local edges via spectral clustering to detect contours in the image. Finally, a watershed transform finds regions from the contours. As of its publication in 2010, this method was the top performer on the Berkeley Segmentation Dataset [43], which is a large and well-annotated dataset for benchmarking natural image segmentation methods. The only parameter is the edge strength required to create a segment,  $k$ .

Note that we do not compare to our previous work [26] because it presented a supervised segmentation method, while *ORTSEG* and all the comparison methods are unsupervised.

### C. Experimental Setup

The comparison was performed as a leave-one-out cross validation. for each method and dataset, each image was sequestered in turn while the others were used to perform a parameter sweep over the parameters mentioned in Section IV-B with the goal of allowing each method to achieve its maximum performance. Specifically, we set up the sweep according the principles: 1) When the authors of the method specified an operating range for a parameter, we swept between its minimum and maximum value. When they did not, we swept between values that produced opposite kinds of bad results (e.g. too coarse and too fine segmentations) to ensure that the optimal value was between our end points. 2) When the correct value for a parameter (e.g. number of regions) was known for an entire dataset, this value was the only one used. 3) When methods had comparable parameters (e.g. a window

size), comparable values were swept. 4) The total number of parameter settings to sweep did not exceed 150 for any method.

The performance of the method was then evaluated on the sequestered image using the parameters that resulting in the best average performance on the training set. The results in Table I represent the average performance when this process was repeated over all images.

As our performance metric, we chose the Rand index [44], [45]. The Rand index gives the fraction of pairs of pixels that are either grouped into a single region in both the ground truth and the test image or are in different regions in both the ground truth and the test image. It therefore ranges from 0 to 1, with 1 being perfect agreement between the test image and the ground truth. We selected it because it naturally handles multi-region unsupervised segmentation and avoids degenerating when a method greatly over- or under-segments an image. There are a wide variety of other, similarly good performance metrics for segmentation algorithms; see Section 3 of [46] for a good overview. In addition to the Rand index, we provide results in terms of the variation of information [47] on the reproducible research page for this paper; the ranking of methods is largely unchanged between metrics and the best-performing method for each dataset does not change.

### D. Discussion

On the random texture dataset (Figure 8), *ORTSEG* is clearly superior to the comparison methods, which shows that it excels when segmenting truly edgeless images. *ORTSEG-D* performs slightly better, giving essentially perfect segmentations (Figure 7), because it leverages prior knowledge about how the region boundaries were formed. *JSEG* has the next best performance, which makes sense because it does not rely on edge information. We believe that *ORTSEG* outperforms *JSEG* in this case because *ORTSEG* optimizes its labels over the whole image rather than in a region merging scheme. The other methods have a difficult time with the random texture dataset, likely due to their reliance on edge information.

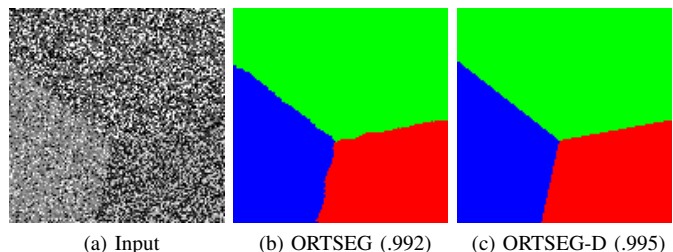
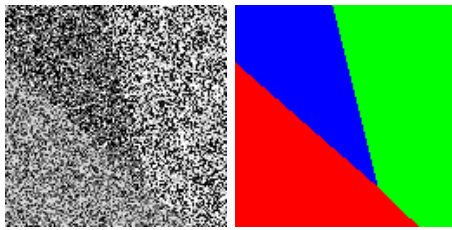


Fig. 7. Example segmentation results from *ORTSEG* and *ORTSEG-D* on a synthetic texture; Rand indexes are given in parentheses. The parametric deconvolution in *ORTSEG-D* forces region boundaries to be linear, improving the segmentation quality.

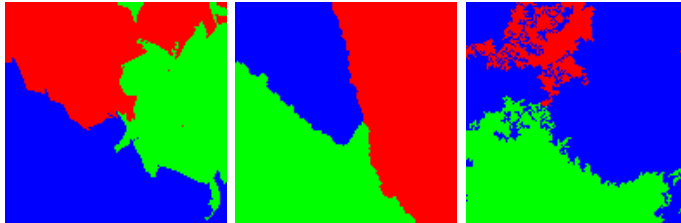
On the histology dataset (Figure 9), *ORTSEG* also outperforms the comparison methods, though the next best method is within a standard deviation. On this dataset, *ORTSEG* selects a relatively large window  $w$  ( $25 \times 25$ , as opposed to  $9 \times 9$  on the random textures), resulting in smoother boundaries compared





(a) Input

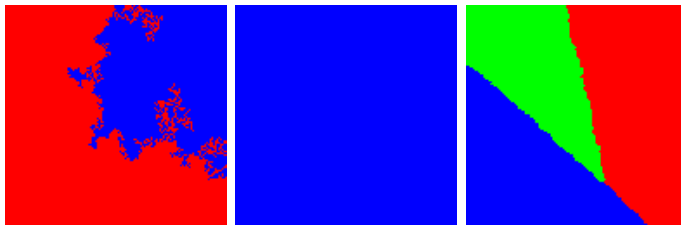
(b) Ground truth



(c) Normalized (.778)

(d) JSEG (.962)

(e) EDISON (.692)



(f) Efficient (.602)

(g) gPb-owt-ucm (.347)

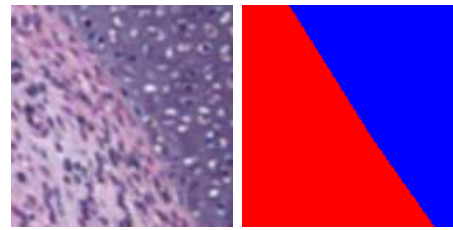
(h) ORTSEG (.987)

Fig. 8. Example segmentation results for the comparison methods (c)-(g) and our method (h) on the random texture dataset; Rand indexes are given in parentheses. This dataset is challenging due its lack of meaningful edges.

to e.g. EDISON. This is an asset because the tissues generally have smooth boundaries. We believe that ORTSEG performs worse on the histology images than on the random textures because the histology images are much more complex. The relatively stronger performance of the comparison methods makes sense because at least some of the images in this dataset do have sharp edges.

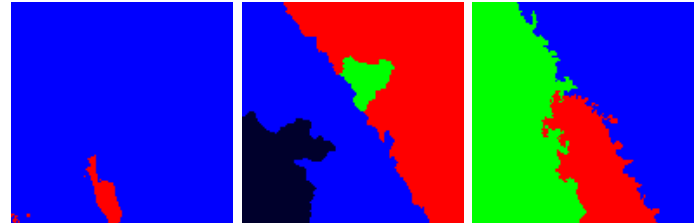
On the Prague dataset (Figure 10), ORTSEG does not outperform the comparison methods. We believe that ORTSEG’s relatively poor performance is because, by design, it does not detect the sharp edges between regions, while other methods do (e.g. note the sharp edges that Normalized Cut finds). Surprising, JSEG has the best performance on this dataset. We have a few ideas why. First, this may be because the seeded region growing approach is good at finding the correct number of regions in the image. Second, while JSEG does not explicitly look for edges, it uses high  $J$  values as evidence of boundaries, which may let it leverage the edges in these images to some degree. Finally, JSEG uses information at a variety of scales while our implementation of ORTSEG does not, which may help in the Prague dataset where the scale of the textures is large relative to the complexity of the boundaries.

Here we make a few observations about the parameters selected for ORTSEG during the cross validation. For the random texture dataset, the typical window size selected was  $9 \times 9$ , while the quantization was set to 8 colors and the number of textures was set to 3. This relatively small window size



(a) Input

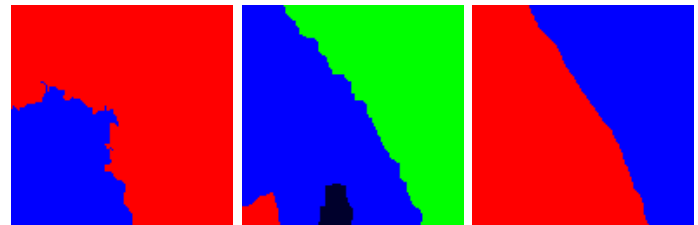
(b) Ground truth



(c) Normalized (.500)

(d) JSEG (.821)

(e) EDISON (.851)



(f) Efficient (.614)

(g) gPb-owt-ucm (.903)

(h) ORTSEG (.924)

Fig. 9. Example segmentation results for the comparison methods (c)-(g) and our method (h) on the histology dataset; Rand indexes are given in parentheses. This is a real-world example of a dataset in which the lack of edges is challenging for many segmentation methods.

makes sense: the textures in images are small scale (neighboring pixels are statistically independent) so only a small window is needed to recognize them and has the advantage of localizing the boundaries better than a larger window. The performance on this dataset is not very sensitive to scale, however, as window sizes from  $5 \times 5$  to  $25 \times 25$  generally give Rand indexes that vary less than .01. In the histology dataset, the typical window size selected was  $25 \times 25$  and the typical quantization selected was 50 colors. The number of textures was set to 2. Though the histology images are the same size as the random images, the increased scale of the textures means that a larger window is necessary, reducing the ability of the algorithm to accurately localize boundaries. Again, results were not very sensitive to changes in window size. Changes to the number of quantized colors did not make much difference once the number of colors was greater than 10. The performance of ORTSEG on the Prague dataset was not strong enough to draw any conclusions about parameters.

The computation complexity of ORTSEG is dependent on the particular implementation of the individual steps. The local histogram transform is a filtering operation and will therefore scale well with input image size; this is also true of many deconvolution algorithms. The factorization step as we described it involves inverting a matrix that grows in size with the square of the number of pixels in the image, however if the number of textures remains low and no texture region is

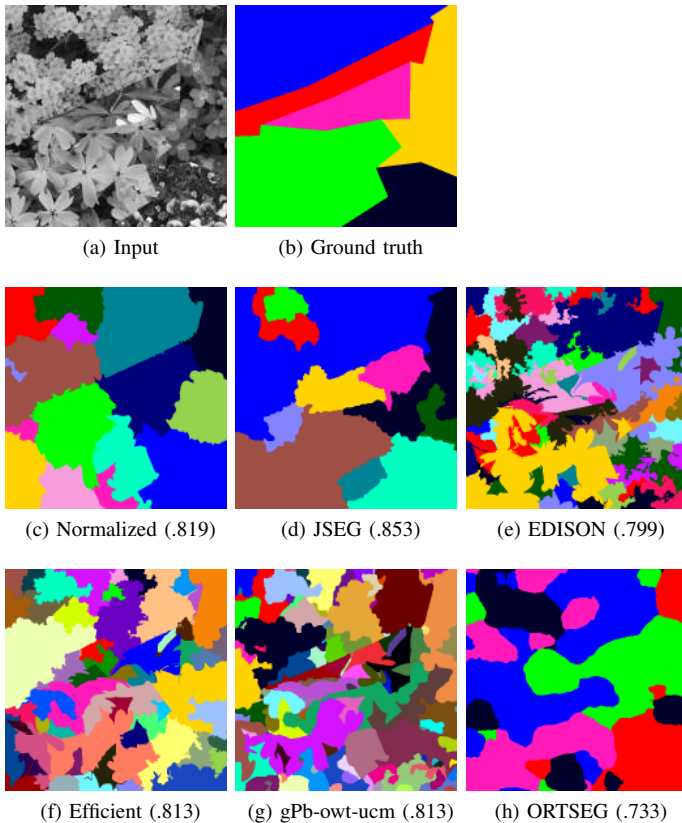


Fig. 10. Example segmentation results for the comparison methods (c)-(g) and our method (h) on the Prague dataset; Rand indexes are given in parentheses. This dataset simulates segmentation of natural images. Our method is not well-suited to such images because it by design does not make use of edges as evidence of boundaries.

very small, then in practice a random subset of pixels can be used for factorization, reducing the complexity of this step.

For our non-optimized MATLAB implementation, run times per image on a Windows 7 laptop with an Intel Core i7-620M processor for the random texture, histology, and Prague datasets are about 1.5 seconds, 2.5 seconds, and 22 seconds, respectively. On the same computer, segmenting a color  $1600 \times 1200$  image into five regions (result not shown) takes about 30 seconds. Selecting the number of textures automatically as described in Section III-D increases these times multiplicatively because it simply runs the algorithm repeatedly with different values of  $N$ .

## V. CONCLUSIONS AND FUTURE DIRECTIONS

Inspired by the difficult task of segmenting histology images, we proposed a new mathematical and algorithmic framework for image segmentation. We began with the idea of textures as random fields and explored how the dependence between pixels in these textures varies spatially by introducing the notions of dependence decay and persistence length. We then modeled images as occlusions of textures and showed that the local histograms of these images are convex combinations of the value distributions of their component textures. Based on this theorem, we proposed a segmentation framework that first discovers the value distributions of the component

textures of an image, finds the contribution of each texture to every local histogram of the image, then deconvolves the contributions to recover a segmentation.

We presented one implementation of this framework, ORTSEG, and compared it to four segmentation methods from the literature on three datasets. ORTSEG outperformed the other methods on images of random textures and real histology images, indicating that it handles the difficult class of edgeless images and has real applicability to histology image segmentation.

In the future, we aim to expand our mathematical model by studying histograms of filter responses rather than simply colors. This would allow for a richer characterization of textures and has interesting algorithmic ramifications. We will also explore other implementations of the presented algorithmic framework, including calculating histograms over multiple scales and testing a wider variety of deconvolution approaches.

## APPENDIX A PROOF OF THEOREM 1

*Proof:* Let  $F$  be a flat texture with value distribution  $p_F$  and persistence length  $\epsilon$ . Given any set of  $N_0$  unique pixel locations,  $\Theta = x_0, x_1, \dots, x_{N_0-1}$ , we aim to partition that set into subsets such that the minimum distance between two locations in a subset is larger than  $\epsilon$ . We call the locations within  $\epsilon$  of a location its neighbors. We place the locations in order of decreasing number of neighbors. Let the largest number of neighbors of any location be  $A_0$ . We build a subset by first selecting the location with the largest number of neighbors and then selecting the location with the next largest number of neighbors among all those that are not neighbors of the first location. We continue this process with each new location selected such that it is not a neighbor of any previously selected location. We end this process after forming a subset of size  $K_0 = \lfloor \frac{N_0}{A_0} \rfloor$ . This is possible because each addition to the subset can only preclude a maximum of  $A_0$  locations from further consideration. Additionally, each location with  $A_0$  neighbors must either be in this set or be a neighbor of a location in this set. If this is not the case, then there are  $A_0$  locations not in the set, which leads to  $N_0 \geq K_0 + A_0$ , which is a contradiction.

Removing the locations in the first subset from consideration, let  $N_1$  be the number of remaining locations and  $A_1$  be the maximum number of neighbors among them. We know that  $A_1 \leq A_0 - 1$  because, similarly to the reasoning above, each point that had  $A_0$  neighbors belongs to the first subset or lost at least one neighbor to the first subset. We now follow the same procedure as above, creating a subset of size  $K_1 = \lfloor \frac{N_1}{A_1} \rfloor$ . We continue the process until all points have been placed in subsets. The same argument as above gives that  $A_{i+1} \leq A_i - 1$  and therefore,

$$\begin{aligned}
K_{i+1} &= \left\lfloor \frac{N_{i+1}}{A_{i+1}} \right\rfloor = \left\lfloor \frac{N_i - K_i}{A_{i+1}} \right\rfloor = \left\lfloor \frac{N_i - \left\lfloor \frac{N_i}{A_i} \right\rfloor}{A_{i+1}} \right\rfloor \\
&\geq \left\lfloor \frac{N_i - \frac{N_i}{A_i}}{A_{i+1}} \right\rfloor \geq \left\lfloor \frac{N_i - \frac{N_i}{A_i}}{A_i - 1} \right\rfloor \\
&= \left\lfloor \frac{A_i N_i - N_i}{A_i(A_i - 1)} \right\rfloor = \left\lfloor \frac{N_i}{A_i} \right\rfloor = K_i.
\end{aligned}$$

The above implies that  $K_i \geq K_0 = \left\lfloor \frac{N_0}{A_0} \right\rfloor$ . Finally, since the largest possible value for  $A_0$  is  $A_\epsilon = 2\epsilon^2 + 2\epsilon + 1$ , which is the number of locations in the  $\ell^1$ -ball with radius  $\epsilon$ , we can define  $K$  where  $K_i \geq K = \left\lfloor \frac{N_0}{A_\epsilon} \right\rfloor$ .

Let  $Y_i = \{y_0, y_1, \dots, y_{K_i-1}\}$  be the  $i$ th subset. By the definition of persistence length and flatness, the random variables  $F(y_0), F(y_1), \dots, F(y_{K_i-1})$  are i.i.d. (persistence length gives independence, flatness gives identical distribution) with distribution  $p_F$ . Let the empirical distribution of  $F(y_0), F(y_1), \dots, F(y_{K_i-1})$  be

$$\hat{p}_F^{Y_i}(v) = \frac{1}{K_i} \sum_{k=0}^{K_i-1} 1_{\{v\}}(F(y_k)).$$

We can select an arbitrary ordering on the set of values and define the cumulative value distribution and the empirical cumulative value distribution of  $F$  as

$$P_F(v) = \sum_{w=0}^v p_F(w)$$

and

$$\hat{P}_F^{Y_i}(v) = \frac{1}{K_i} \sum_{k=0}^{K_i-1} 1_{[0,v]}(F(y_k)),$$

respectively.

Then, by the Dvoretzky-Kiefer-Wolfowitz inequality [48],

$$\Pr(\max |\hat{P}_F^{Y_i}(v) - P_F(v)| > \alpha) \leq 2e^{-2K_i\alpha^2}.$$

To return to the (non-cumulative) color distributions we note that

$$\begin{aligned}
|\hat{p}_F^{Y_i} - p_F|(v) &= |(\hat{P}_F^{Y_i}(v) - \hat{P}_F^{Y_i}(v-1)) - (P_F(v) - P_F(v-1))| \\
&\leq |(\hat{P}_F^{Y_i}(v) - P_F(v)) - (\hat{P}_F^{Y_i}(v-1) - P_F(v-1))| \\
&\leq 2 \max |\hat{P}_F^{Y_i}(v) - P_F(v)|,
\end{aligned}$$

so

$$\Pr(\max |\hat{p}_F^{Y_i}(v) - p_F(v)| > \alpha) \leq 2e^{-K_i\alpha^2/2}.$$

Finally, letting  $\alpha = \sqrt{\frac{-2\log \frac{\gamma}{2}}{K_i}}$ , gives

$$\max |\hat{p}_F^{Y_i}(v) - p_F(v)| \leq \sqrt{\frac{-2\log \frac{\gamma}{2}}{K_i}} \leq \sqrt{\frac{-2\log \frac{\gamma}{2}}{K}} \quad (9)$$

with probability at least  $1 - \gamma$ .

We now examine the histogram formed from all the locations in the original set  $\Theta$  rather than a single subset. If we have a total of  $I$  subsets, then

$$\begin{aligned}
\hat{p}_F^\Theta(v) &= \frac{1}{N_0} \sum_{n=0}^{N_0-1} 1_{\{v\}}(F(x_n)) \\
&= \frac{1}{N_0} \sum_{i=0}^{I-1} \sum_{y \in Y_i} 1_{\{v\}}(F(y)) = \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \hat{p}_F^{Y_i}(v)
\end{aligned}$$

and therefore

$$\begin{aligned}
\max |\hat{p}_F^\Theta(v) - p_F(v)| &= \max \left| \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \hat{p}_F^{Y_i}(v) - p_F(v) \right| \\
&= \max \left| \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \hat{p}_F^{Y_i}(v) - \frac{1}{N_0} \sum_{i=0}^{I-1} K_i p_F(v) \right| \\
&= \max \frac{1}{N_0} \sum_{i=0}^{I-1} K_i |\hat{p}_F^{Y_i}(v) - p_F(v)| \\
&\leq \frac{1}{N_0} \sum_{i=0}^{I-1} K_i \sqrt{\frac{-2\log \frac{\gamma}{2}}{K}} \\
&= \sqrt{\frac{-2\log \frac{\gamma}{2}}{K}} = \sqrt{\frac{-2\log \frac{\gamma}{2}}{\left\lfloor \frac{N_0}{A_\epsilon} \right\rfloor}} \leq \sqrt{\frac{-2A_\epsilon \log \frac{\gamma}{2}}{N_0}}
\end{aligned}$$

with probability at least  $(1 - \gamma)^I$ , because each of the  $I$  inequalities from (9) must hold simultaneously. Given that the minimum subset size is  $\left\lfloor \frac{N_0}{A_\epsilon} \right\rfloor$ ,  $I \leq A_\epsilon$ , so this probability is at least  $(1 - \gamma)^{A_\epsilon}$ . We finally set  $\gamma = 1 - \sqrt[A_\epsilon]{1 - \delta}$ . The result is

$$\max |\hat{p}_F^\Theta(v) - p_F(v)| \leq \sqrt{\frac{-2A_\epsilon \log \frac{1 - \sqrt[A_\epsilon]{1 - \delta}}{2}}{N_0}}$$

with probability at least  $(1 - \delta)$ . This means that  $F$  has dependence decay complexity  $a = (-2A_\epsilon \log \frac{1}{2}(1 - \sqrt[A_\epsilon]{1 - \delta}))^{1/2}$ , proving the theorem.  $\blacksquare$

## APPENDIX B PROOF OF THEOREM 2

*Proof:* Start with the first term of the left-hand side of (5),

$$\begin{aligned}
L_w \text{O}_\sigma \{f_n\}_{n=0}^{N-1}(x, v) &\stackrel{(a)}{=} L_w \sum_{n=0}^{N-1} 1_{\{n\}}(\sigma(x)) f_n(x) \\
&\stackrel{(b)}{=} \sum_{x' \in X} 1_{\{v\}} \left( \sum_{n=0}^{N-1} 1_{\{n\}}(\sigma(x')) f_n(x') \right) w(x - x') \\
&\stackrel{(c)}{=} \sum_{n=0}^{N-1} \sum_{x' \in X} 1_{\{n\}}(\sigma(x')) 1_{\{v\}}(f_n(x')) w(x - x'),
\end{aligned} \quad (10)$$

where (a) follows from the definition of occlusion (2), (b) from the definition of the local histogram transform (1), and (c) because, for a given  $c$  and  $n$ , the indicator functions are only nonzero when  $\sigma(x') = n$  and  $f_n(x') = v$ .

We now manipulate (10) so that it becomes, for each  $n$ , a histogram sampled from  $f_n$ . Using the fact that  $w$  is a constant filter (that is, it takes only the value  $1/|w|$  or 0), (10) becomes

$$\begin{aligned} & \sum_{n=0}^{N-1} \frac{\sum_{x' \in X} 1_{\{n\}}(\sigma(x')) 1_{\{v\}}(f_n(x')) 1_{(0,1]}(w(x-x'))}{|w|} \quad \underline{(a)} \\ & \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} \frac{\sum_{x' \in X} 1_{\{n\}}(\sigma(x')) 1_{\{v\}}(f_n(x')) 1_{(0,1]}(w(x-x'))}{|\Theta_n(x)|} \\ & \underline{(b)} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} \hat{p}_{f_n}^{\Theta_n(x)}(v), \end{aligned} \quad (11)$$

where  $\Theta_n(x) = \{y \mid \sigma(y) = n \text{ and } w(x-y) > 0\}$ . The equality (a) is a multiplication by 1 and (b) uses the definition of a histogram, (3).

Turning our attention to the second term of the left-hand side of (5), we have

$$\begin{aligned} & \sum_{n=0}^{N-1} [w * 1_{\{n\}}(\sigma)](x) p_{F_n}(v) \\ & \underline{(a)} \sum_{n=0}^{N-1} \sum_{x' \in X} w(x-x') 1_{\{n\}}(\sigma(x')) p_{F_n}(v) \\ & \underline{(b)} \sum_{n=0}^{N-1} \frac{\sum_{x' \in X} 1_{(0,1]}(w(x-x')) 1_{\{n\}}(\sigma(x'))}{|w|} p_{F_n}(v) \\ & \underline{(c)} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} p_{F_n}(v), \end{aligned} \quad (12)$$

where (a) follows from the definition of convolution, (b) from  $w$  being a constant filter, and (c) from defining  $\Theta_n(x)$  as above.

Subtracting (12) from (11), we have

$$\begin{aligned} & \left| \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} (\hat{p}_{f_n}^{\Theta_n(x)}(v) - p_{F_n}(v)) \right| \\ & \underline{(a)} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} \left| \hat{p}_{f_n}^{\Theta_n(x)}(v) - p_{F_n}(v) \right| \\ & \underline{(b)} \sum_{n=0}^{N-1} \frac{|\Theta_n(x)|}{|w|} a_n |\Theta_n(x)|^{-1/2}, \end{aligned} \quad (13)$$

where (a) follows from the triangle inequality and positivity of  $|\Theta_n(x)|$  and  $|w|$  and (b) from the dependence-decay property of  $\{F_0, F_1, \dots, F_{N-1}\}$ . The probability of (b) holding for each  $n$  is  $1 - \delta$ , thus the probability of it holding for all of them simultaneously is  $(1 - \delta)^N$ .

We aim to bound (13) by its maximum. To do this, we first replace  $a_n$  with  $a = \max\{a_0, a_1, \dots, a_{N-1}\}$ . We then maximize the quantity over  $|\Theta_0|, |\Theta_1|, \dots, |\Theta_{N-1}|$  with the constraint that  $|\Theta_0| + |\Theta_1| + \dots + |\Theta_{N-1}| = |w|$ . This maximum occurs at  $|\Theta_0| = |\Theta_1| = \dots = |\Theta_{N-1}| = |w|/N$ , meaning that (13) is bounded by

$$\sum_{n=0}^{N-1} \frac{1}{N} a \left( \frac{|w|}{N} \right)^{-1/2} = a \left( \frac{|w|}{N} \right)^{-1/2},$$

thus proving the theorem. ■



**Michael McCann** (S'10) received his B.S.E. in biomedical engineering in 2010 from the University of Michigan. He is currently a Ph.D. student in the Center for Bioimage Informatics and the Department of Biomedical Engineering at Carnegie Mellon University, where he received an NSF Graduate Research Fellowship. His work focuses on developing signal processing tools for histology images, including normalization, registration, and tissue segmentation/identification.



**Dustin Mixon** received the B.S. degree in mathematics from Central Washington University, Ellensburg, WA, in 2004 and the M.S. degree in applied mathematics from the Air Force Institute of Technology, Wright-Patterson AFB, OH, in 2006. From 2006 to 2009, he was an applied mathematical analyst with the Air Force Research Laboratory, Brooks City-Base, TX. He received the Ph.D. degree in applied and computational mathematics from Princeton University, Princeton, NJ, in 2012. He is currently an Assistant Professor of Mathematics in the Department of Mathematics and Statistics, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH. His research interests include applied harmonic analysis and mathematical signal processing.



**Matthew Fickus** (M08) received the Ph.D. degree in mathematics from the University of Maryland, College Park, in 2001. He spent the following three years at Cornell University, Ithaca, NY, as an NSF VIGRE postdoctoral researcher. In 2004, he joined the Department of Mathematics and Statistics, Air Force Institute of Technology, where he is currently an Associate Professor of Mathematics. His research focuses on frames and harmonic analysis, emphasizing their applications to signal and image processing.



**Carlos Castro** is of Colombian origin. He received the M.D. degree from Rosario University Medical School, Bogota, Colombia and the D.M.D. degree from Javeriana University Dental School in Bogota, Colombia. He went on to do his maxillofacial surgery fellowship at Rosario University Medical School and subsequently did oral, orthognathic surgery, and gnathology specialties at the same universities as well as the Military University School of Medicine. His path positions include Assistant Director of the Cumaral Hospital, Instructor of Oral and Maxillofacial Surgery and Pediatric Dentistry at Children's Hospital of Bogota, Colombia, General Dentist and Maxillofacial and Craniofacial Surgeon at Children's Hospital of Bogota as well as the Fundacion Santa Fe de Bogota.

In America, Dr Castro occupied several positions including Dental Laboratory Specialist, Microbiology Specialist, Organ Procurement and Transplantation Specialist, and Director of Operations of a medical laboratory. He worked in general pathology and subsequently surgical and perinatal pathology at the University of Pittsburgh and as a Pathologist Assistant at Magee-Womens Hospital in Pittsburgh, PA. He has also worked as a Senior Research Associate, Visiting Research Instructor, Senior Faculty Research Instructor, and Co-Director of the Magee-Womens Research Institute and Foundation Histology Core. He is currently Director of the same histology core.





**John Ozolek** graduated from Case Western Reserve University with a B.S. in Biology and matriculated to the University of Pittsburgh School of Medicine where he obtained his M.D. in 1989. He has completed postgraduate residency training in Pediatrics and Anatomic Pathology and Fellowships in Neonatal-Perinatal Medicine and Pediatric Pathology at the University of Pittsburgh. He is currently Assistant Professor of Pathology at the University of Pittsburgh and Staff Pathologist at the Children Hospital of Pittsburgh. Dr. Ozolek's scientific inter-

ests center on image analysis for the detection of specific tissue types and classification of challenging diagnostic dilemmas in surgical pathology. He has extensive collaborations with the Center for Bioimage Informatics and Departments of Biomedical Engineering and Mathematics at Carnegie Mellon University to develop algorithms and computational approaches for automating tissue recognition and correlation of histology and high-resolution magnetic resonance images in teratomas derived from human and non-human primate stem cells. This collaboration continues to strive to develop automated computational morphometric technologies for applications in cell biology and diagnostic pathology. His clinical and diagnostic research interests focus on the role of stem cells and neural crest cells particularly in the pathogenesis of branchial cleft cysts, developmental abnormalities, and tumors of the head and neck in children.



**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. 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 and an Adjunct Professor at Columbia University. In 2003, she joined Carnegie Mellon University, where she is a Professor and Head of Electrical

and Computer Engineering, Professor of Biomedical Engineering, and the Director of the Center for Bioimage Informatics. Her research interests include bioimaging as well as multiresolution techniques such as wavelets and frames. Dr. Kovačević coauthored the books *Wavelets and Subband Coding* (Prentice Hall, 1995) and *Foundations of Signal Processing* (Cambridge University Press, 2014), a top-10 cited paper in the *Journal of Applied and Computational Harmonic Analysis*, and the paper for which A. Mojsilović received the Young Author Best Paper Award. Her paper on multidimensional filter banks and wavelets 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 is a past Editor-in-Chief of the *IEEE Transactions on Image Processing*, served as a guest co-editor on a number of special issues and is/was on the editorial boards of several journals. She was a regular member of the NIH Microscopic Imaging Study Section and served as a Member-at-Large of the IEEE Signal Processing Society Board of Governors. She is a past Chair of the IEEE Signal Processing Society Bio Imaging and Signal Processing Technical Committee. She has been involved in organizing numerous conferences. She was a plenary/keynote speaker at a number of international conferences and meetings.

## REFERENCES

- [1] D. R. Li, G. Zhang, Z. Wu, and L. Yi, "An edge embedded marker-based watershed algorithm for high spatial resolution remote sensing image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2781–2787, Oct. 2010.
- [2] N. Jacobson, Y.-L. Lee, V. Mahadevan, N. M. Vasconcelos, and T. Q. Nguyen, "A novel approach to FRUC using discriminant saliency and frame segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2924–2934, Nov. 2010.
- [3] H. Ayasso and A. Mohammad-Djafari, "Joint NDT image restoration and segmentation using Gauss-Markov-Potts prior models and variational Bayesian computation," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2265–2277, Sep. 2010.
- [4] A. Gooya, H. Liao, K. Matsumiya, K. Masamune, Y. Masutani, and T. Dohi, "A variational method for geometric regularization of vascular segmentation in medical images," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1295–1312, Aug. 2008.
- [5] C. Zanella, M. Campana, B. Rizzi, C. Melani, G. Sanguinetti, P. Bourguin, K. Mikula, N. Peyri eras, and A. Sarti, "Cells segmentation from 3-D confocal images of early zebrafish embryogenesis," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 770–781, Mar. 2010.
- [6] G. Srinivasa, M. C. Fickus, Y. Guo, A. D. Linstedt, and J. Kovačević, "Active mask segmentation of fluorescence microscope images," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1817–1829, Aug. 2009.
- [7] K.-C. Chen, Y. Yu, R. Li, H.-C. Lee, G. Yang, and J. Kovačević, "Adaptive active-mask image segmentation for quantitative characterization of mitochondrial morphology," in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, Sep. 2012, pp. 2033–2036.
- [8] D. B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer, "Semi-automated reconstruction of neural circuits using electron microscopy," *Curr. Opin. Neurobiol.*, vol. 20, no. 5, pp. 667–675, Oct. 2010.
- [9] J. Nascimento and J. Marques, "Robust shape tracking with multiple models in ultrasound images," *IEEE Trans. Image Process.*, vol. 17, no. 3, pp. 392–406, 2008.
- [10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
- [11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.
- [12] R. Malladi, J. A. Sethian, and B. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995.
- [13] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recogn.*, vol. 24, pp. 1167–1186, 1991.
- [14] H. Derin and W. S. Cole, "Segmentation of textured images using Gibbs random fields," *Computer Vis. Graph. Image Process.*, vol. 35, no. 1, pp. 72–98, Jul. 1986.
- [15] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.
- [16] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.
- [17] D. Comanicu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [18] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using Expectation-Maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 1026–1038, 1999.
- [19] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
- [20] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. IEEE Int. Conf. Pattern Recogn.*, vol. 4, Quebec City, Aug. 2002, pp. 150–155.
- [21] D. E. Ilea and P. F. Whelan, "CTex—An adaptive unsupervised segmentation algorithm based on color-texture coherence," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1926–1939, Oct. 2008.
- [22] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "Use of active shape models for locating structures in medical images," *Image Vis. Comput.*, vol. 12, no. 6, pp. 355–365, Jul. 1994.
- [23] J. A. Ozolek and C. A. Castro, *Teratomas Derived from Embryonic Stem Cells as Models for Embryonic Development, Disease, and Tumorigenesis*. InTech, 2011, ch. 13.
- [24] C. Chen, J. A. Ozolek, W. Wang, and G. K. Rohde, "A general system for automatic biomedical image segmentation using intensity neighborhoods," *Int. J. Biomed. Imag.*, Jan. 2011.

- [25] A. Cagri, A. Burak, C. Aykanat, C. Sokmensuer, and C. Gunduz-Demir, "Multilevel segmentation of histopathological images using cooccurrence of tissue objects," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 6, pp. 1681–1690, Jun. 2012.
- [26] M. L. Massar, R. Bhagavatula, M. Fickus, and J. Kovačević, "Local histograms and image occlusion models," *Appl. Comput. Harmon. Anal.*, vol. 34, no. 3, pp. 469–487, May 2013.
- [27] A. B. Lee, D. Mumford, and J. Huang, "Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model," *Int. J. Comput. Vis.*, vol. 41, pp. 35–59, 2001.
- [28] M. Haindl and S. Mikes, "Texture segmentation benchmark," in *Proc. IEEE Int. Conf. Pattern Recogn.*, Dec. 2008, pp. 1–4.
- [29] O. Michailovich, Y. Rathii, and A. Tannenbaum, "Image segmentation using active contours driven by the bhattacharyya gradient flow," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2787–2801, Nov. 2007.
- [30] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," *Bull. Am. Math. Soc.*, vol. 49, no. 1, pp. 1–24, Jan. 1943.
- [31] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.
- [32] M. L. Fisher, "An applications oriented guide to lagrangian relaxation," *Interfaces*, vol. 15, no. 2, pp. 10–21, Mar. 1985.
- [33] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.
- [34] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 155–173, Sep. 2007.
- [35] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 2002.
- [36] P. Sarder and A. Nehorai, "Deconvolution methods for 3-D fluorescence microscopy images," *IEEE Signal Process. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, no. 3, pp. 32–45, May 2006.
- [37] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, Jun. 1994.
- [38] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992.
- [39] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recogn. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
- [40] M. T. McCann, D. G. Mixon, M. Fickus, C. A. Castro, J. A. Ozolek, and J. Kovačević. (2013) Images as occlusions of textures: A framework for segmentation. [Online]. Available: [http://www.jelena.ece.cmu.edu/repository/rr/13\\_McCannMFCOK/13\\_McCannMFCOK.html](http://www.jelena.ece.cmu.edu/repository/rr/13_McCannMFCOK/13_McCannMFCOK.html)
- [41] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [42] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.
- [43] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
- [44] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Am. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.
- [45] R. Unnikrishnan, C. Pantofaru, and M. H. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.
- [46] D. E. Ilea and P. F. Whelan, "Image segmentation based on the integration of colourtexture descriptors—A review," *Pattern Recogn.*, vol. 44, no. 10-11, pp. 2479–2501, Oct. 2011.
- [47] M. Meilä, "Comparing clusterings by the variation of information," in *Learning Theory and Kernel Machines*, ser. Lecture Notes in Computer Science, B. Schölkopf and M. Warmuth, Eds. Springer Berlin Heidelberg, 2003, vol. 2777, pp. 173–187.
- [48] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *Ann. Math. Stat.*, vol. 27, no. 3, pp. 642–669, Sep. 1956.