# Graph Run-Length Matrices for Histopathological Image Segmentation

Akif Burak Tosun, *Student Member, IEEE*, and Cigdem Gunduz-Demir*, *Member, IEEE*

*Abstract*—The histopathological examination of tissue specimens is essential for cancer diagnosis and grading. However, this examination is subject to a considerable amount of observer variability as it mainly relies on visual interpretation of pathologists. To alleviate this problem, it is very important to develop computational quantitative tools, for which image segmentation constitutes the core step. In this paper, we introduce an effective and robust algorithm for the segmentation of histopathological tissue images. This algorithm incorporates the background knowledge of the tissue organization into segmentation. For this purpose, it quantifies spatial relations of cytological tissue components by constructing a graph and uses this graph to define new texture features for image segmentation. This new texture definition makes use of the idea of gray-level run-length matrices. However, it considers the runs of cytological components on a graph to form a matrix, instead of considering the runs of pixel intensities. Working with colon tissue images, our experiments demonstrate that the texture features extracted from "graph run-length matrices" lead to high segmentation accuracies, also providing a reasonable number of segmented regions. Compared with four other segmentation algorithms, the results show that the proposed algorithm is more effective in histopathological image segmentation.

*Index Terms*—Cancer, graphs, histopathological image analysis, image segmentation, image texture analysis, perceptual image segmentation.

## I. INTRODUCTION

IN the current practice of medicine, there has been an increasing use of imaging systems in making decisions for several medical phenomena. Today, the imaging systems are primarily used to acquire the digital images of different parts of a human body, but they are not typically used to automatically analyze the images or make decisions. Instead, human experts make these analyses/decisions by visually examining the images [1]. Although there are huge efforts for the standardization of this process by defining quantitative measures [2], [3], there may still exist a considerable amount of observer variability in the analyses/decisions. Histopathological tissue examination, which constitutes the gold standard for today's cancer diagnosis and grading, is one of the most important medical

practices that experience this observer variability problem [4]. The main reason behind this is the fact that the quantitative measures are calculated with the visual assessment of the experts. For instance, in cancer grading, each grade is characterized by the degree of distortions and irregularities observed in a biopsy tissue [5]. Although these degrees are defined quantitatively, the quantification should be done by the eyes of a pathologist. To alleviate this problem, many studies have focused on developing computational quantification tools that provide mathematical and objective analyses and assessments [6]–[11]. Most of these studies make their analyses assuming that images are homogeneous. Therefore, image segmentation is at the heart of these computational tools.

Histopathological image segmentation could be considered in the context of color-texture image segmentation problem, for which many approaches have been proposed. Pixel-based approaches divide the image pixels into groups, usually based on their color histograms, by using different techniques, such as k-means clustering [12], fuzzy clustering [13], watershed transformation [14], and thresholding [15], [16]. There are also studies that apply clustering or thresholding to different color spaces or to different pairwise combinations of color channels to calculate multiple segmentation maps of the same image. They then fuse these intermediary maps to obtain the final segmentation [17], [18]. Region-based approaches group the image pixels into clusters, maintaining connectivity among the pixels of the same cluster. Examples include region growing algorithms [19]–[21], split and merge procedures [22], and watershed transformations [23]. These algorithms use color, color gradient, and/or texture to define the region homogeneity. They usually perform a region merge step at the end as they typically result in oversegmentation. Graph-based approaches consider the image as a weighted graph, where nodes represent pixels and the weight of each edge connecting two nodes represents the similarity between them. They then formulate image segmentation as a problem of partitioning this graph into components, minimizing a cost function. It has been proposed to solve this problem using different similarity measures, different cost functions, and different optimization methods [24]–[27]. Statistical approaches consider image segmentation as a probabilistic optimization problem. They model the image probability distributions directly, using parametric and nonparametric estimation [28], [29], or by using graphical models such as Markov random fields and Bayesian networks [30], [31].

Although all these approaches lead to promising results, the color-texture image segmentation problem is not completely solved yet and there still remain different challenges to overcome for different applications. The main challenge lies in the nature of this problem. Image segmentation is closely related

A. B. Tosun is with the Department of Computer Engineering, Bilkent University, Ankara TR-06800, Turkey (e-mail: tosun@cs.bilkent.edu.tr).

*C. Gunduz-Demir are with the Department of Computer Engineering, Bilkent University, Ankara TR-06800, Turkey (e-mail: gunduz@cs.bilkent.edu.tr).
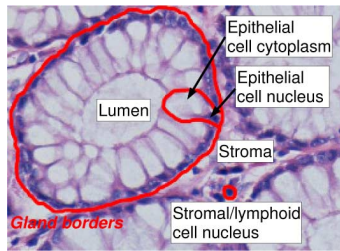
Fig. 1. Cytological components of a colon tissue.



Fig. 2. Histopathological images of colon tissues stained with hematoxylin-and-eosin, which is routinely used to stain biopsies in hospitals. These tissues include both normal regions (marked with 1) and cancerous (adenocarcinoma-tous) regions of different grades (marked with 2). Particularly, (a) and (b) contain Grade 1 colon cancer whereas (c) and (d) contain Grade 2 and Grade 3 colon cancer, respectively. In (d), the region marked with 3 can be included in either region without affecting the medical assessment in the context of colon adeno-carcinoma diagnosis.
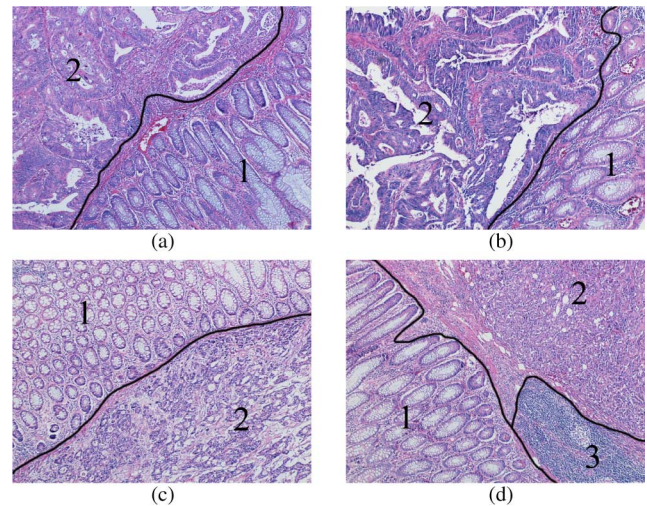
with human perception. Humans typically combine their background knowledge with image data to segment the image into its semantically uniform regions. To incorporate the human perception into segmentation, adaptive clustering algorithms have been proposed [32], [33]. These algorithms adaptively define color-texture descriptors that show spatial variations with respect to the image content. It has also been proposed to define descriptors at different scales for mimicking a human observer looking at the same image from different distances and to combine them in segmentation [34].

In order to improve the success of segmentation, background knowledge, which is specific to the image content and the intent of segmentation, should also be incorporated into the segmentation algorithm. In histopathological image segmentation, this background knowledge includes the normal appearance of a tissue, which could be expressed in terms of the organization of the cytological tissue components. Cancer causes changes in the organization of these components, leading to tissues deviating from their normal appearances. For example, in colon tissues, epithelial cells are lined up around a lumen to form glandular structures and non-epithelial cells take place in stroma found in between these glands (Fig. 1). Colon adenocarcinomas, which account for 90%–95% of all colorectal cancers, cause organizational changes in colon tissues. Pathologists differentiate normal and cancerous regions by looking at these changes (Fig. 2).

The segmentation problem in histopathological images can be considered in two different types of scope. The first type is to locate biological objects such as cells and glands on an image [35]–[38]. The second type, which is also the focus of this paper, is to locate homogeneous regions in a heterogeneous image. In literature, there are a few studies focusing on the latter problem. Most of them perform grid analysis, in which segmentation is achieved dividing an image into fixed grids and classifying them in a supervised way [39], [40]. These studies use pixel intensities and/or pixel textures to quantify the grids. However, they do not consider the background knowledge of tissue organization to define these features. Indeed, it is quite difficult to express this background knowledge in terms of pixels (i.e., using pixel-based feature descriptors). This is mainly due to large variations observed in biopsy samples and noise occurs in preparing these samples and taking their images. The variations and noise typically cause local changes in pixel values. However, they do not change the semantics in the distribution of tissue components on a large scale. For example, in Fig. 2, one could capture the normal appearance of a colon tissue in spite of variations and noise observed in its normal regions. This is our main motivation behind defining a set of new low-level feature descriptors that better correlate with high-level image semantics.

In this paper, we propose a new algorithm for the effective and robust segmentation of histopathological tissue images. In the proposed algorithm, our main contributions are the introduction of a new texture measure that models the spatial distribution of cytological tissue components and the use of this texture measure in histopathological image segmentation. In particular, the algorithm defines the texture of cytological components on a graph using the idea of gray-level run length matrices [41]. However, it considers the runs of cytological components on the graph to form a run-length matrix, instead of considering the runs of pixel intensities. In other words, the algorithm constructs "a graph run-length matrix" by counting the number of "graph-edge runs" instead of constructing a gray-level run-length matrix by counting the number of gray runs. Working with colon tissue images, our experiments demonstrate that the proposed algorithm that uses this new texture definition improves the segmentation performance for histopathological images.

The proposed segmentation algorithm differs from the previous grid-based segmentation algorithms in several aspects. First, it uses the texture of components to incorporate background knowledge in segmentation. Second, it does not define its measure directly on pixel values; thus, it is expected to be less vulnerable to variations and noise in the pixel values. Third, it is an unsupervised algorithm and does not require any training samples. In our previous work, we proposed another texture descriptor that is also defined on tissue components [42]. In particular, it defines two texture measures for each component type. The first one is the standard deviation of the component sizes; it is to quantify how uniform the corresponding components are in terms of their sizes. The second one is the sum of the position vectors for every corresponding component with reference to the centroid; it is to quantify how uniform the components are distributed in space. Nevertheless, this previous work does not employ any graph to quantify the relations of components
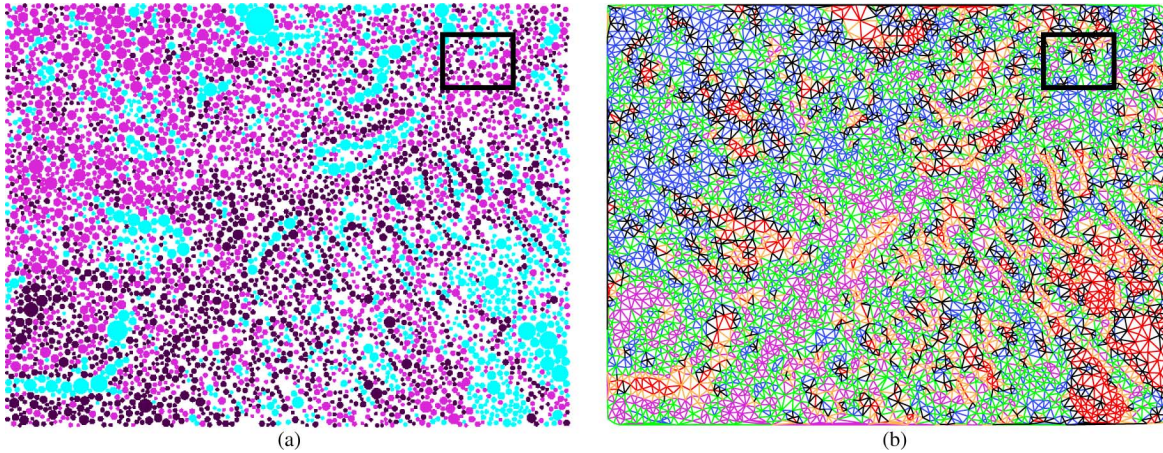
Fig. 3. A graph generated for representing the spatial distribution of cytological components within a tissue: (a) circular primitives representing the tissue components and (b) labeled edges defined in between these primitives.
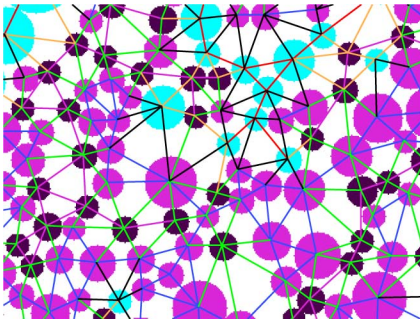


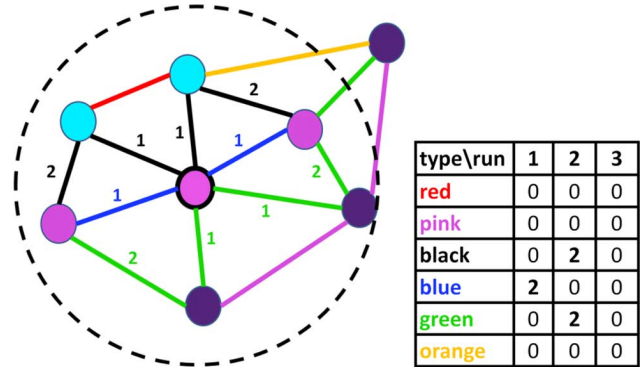Fig. 4. The graph of the subimage confined within a rectangle in Fig. 3.



Fig. 5. Illustration of calculating a graph run-length matrix for a single initial node, which is shown as a thick bordered pink circle.

and does not use any graph-related feature in its segmentation. Different than our previous work, this paper introduces a new texture descriptor, in which graphs are used to quantify the relations of cytological tissue components. This new texture definition improves the segmentation performance of our previous work, also leading to a reasonable number of segmented regions. Moreover, it can determine a common parameter set that is used for all images in obtaining these results, which is one of the problems that we encountered in our previous work.

## II. GRAPH RUN-LENGTH MATRICES

The proposed algorithm relies on modeling the spatial distribution of cytological components within a tissue. For this purpose, it introduces a new texture measure to quantify the spatial relations of these components. This texture definition first generates a graph on the cytological components, then defines a run-length matrix using the edges of the generated graph, and finally extracts a set of texture features from the graph run-length matrix. The details of these steps are given in the following subsections. The segmentation algorithm that uses this new texture definition is explained in Section III.

### A. Graph Generation

In this work, we represent the spatial relations between cytological tissue components using the color graphs that we define

for the classification of histopathological images [43]. For the construction of these color graphs, the cytological tissue components are approximately represented with three different types of primitives, each of which is defined on the pixels of one of the three prominent colors observed in a tissue image. These colors are white, pink, and purple and their corresponding pixels are obtained by k-means clustering.[1] In [43], it is proposed to approximately represent tissue components with circular primitives since their exact localization gives a much more difficult segmentation problem. Specifically, lumina and epithelial cell cytoplasms are represented with white primitives, stroma are represented with pink primitives, and cell nuclei are represented with purple primitives. For the image given in Fig. 2(a), primitives are shown in Fig. 3(a). Note that in Figs. 3–5, white, pink, and purple primitives are shown as cyan, pink, and purple circles, respectively.

After the primitives are identified, a color graph is generated by constructing a Delaunay triangulation on the centroids of these primitives and then labeling each triangle edge according

[1]In hospitals, biopsies are routinely stained with the hematoxylin-and-eosin technique, which mainly leads to white-like, pink-like, and purple-like pixels. Thus, $k$ is selected as three in k-means clustering. For the initialization of the cluster centers, the principal component of the data is calculated, its range is divided into $k$ equal intervals, and initial centers are defined as the averages of the data points falling in these intervals [44].

to the primitive types of its end points. As there are three primitive types in an image, its graph could consist of six different edge types. In Fig. 3(b), the edges assigned in between the primitives are shown; here, edges of different types are illustrated with different colors. For better illustration, Fig. 4 shows an enlarged picture of graph nodes and edges for the subimage that is confined within a rectangle in Fig. 3.

In our work, although we use a similar graph generation algorithm with [43], the way and aim of using these graphs are completely different. In [43], graph theoretical features (such as average degree and diameter) are used to classify histopathological images that are completely homogeneous. On the other hand, in this current work, we propose to employ the graph edges to define a texture measure that is used for the segmentation of heterogeneous histopathological images.

### B. Run-Length Matrix Calculation

After obtaining a graph, we calculate the run length matrix of this graph to quantify the spatial relations of cytological tissue components (i.e., the texture of components). In defining graph run-length matrices, we make use of the idea of calculating gray-level run-length matrices. On a gray-level image, the run-length matrix $I$ quantifies the coarseness of a texture in a specific direction. Given a direction, $I(i,j)$ is the number of runs of pixels with a gray-level $i$ and a run length $j$. A gray-level run is defined as a set of consecutive pixels with the same gray value in the given direction [41].

Our proposed approach uses graph-edge runs instead of using gray-level runs. It defines a graph-edge run as a path that starts from an initial node and contains nodes, all of which are reachable with a set of edges of the same type. Given the initial node, the graph run-length matrix entry $R(c,l)$ is the number of graph-edge runs with an edge type $c$ and a path length $l$. As the graphs are undirected and unweighted, the length of a path is defined as the number of hops required to reach from the initial node to the furthermost node in the path.

In calculating the graph run-length matrix of a single initial node, the algorithm first locates a circular window at the center of this initial node. Then, for each particular edge type, it extracts paths by employing the breadth-first search algorithm on edges that are of this particular type and that are located within the circular window. Fig. 5 illustrates the calculation of a graph run-length matrix for a single node that is shown as a thick bordered pink circle. To calculate the graph run-length matrix of an entire region, the algorithm accumulates the run-length matrices of the nodes located in this region.

### C. Feature Extraction

From the original definition of a gray-level run-length matrix, Galloway [41] proposes to define five texture features whose definitions are given in Table I. In this table, $n_r$ is the total number of runs in the run-length matrix $(\sum_i \sum_j I(i,j))$ and $p$ is the number of pixels in the image. Our proposed approach takes the original definitions of the first four features and modifies them for the graph run-length matrices to define its texture features: short path emphasis, long path emphasis, edge type nonuniformity, and path length nonuniformity.

TABLE I
TEXTURE FEATURES FOR GRAY-LEVEL RUN-LENGTH MATRICES

$$\text{Short run emphasis} = \frac{1}{n_r} \sum_i \sum_j I(i,j)/j^2$$

$$\text{Long run emphasis} = \frac{1}{n_r} \sum_i \sum_j I(i,j) \times j^2$$

$$\text{Gray level nonuniformity} = \frac{1}{n_r} \sum_i \left( \sum_j I(i,j) \right)^2$$

$$\text{Run-length nonuniformity} = \frac{1}{n_r} \sum_j \left( \sum_i I(i,j) \right)^2$$

$$\text{Run percentage} = \frac{\sum_i \sum_j I(i,j)}{p}$$

The *short path emphasis* gives more importance to shorter graph-edge runs than the longer ones, dividing the number of runs by the square of their lengths. First, this feature, SPE, is calculated regardless of the edge types (similar to Table I). Then, $\text{SPE}(c)$ is calculated for each of the six edge types separately, considering only the runs of the corresponding type. The equations of these descriptors are given as follows. In these equations, $n_r$ is the total number of runs in the graph run-length matrix $(\sum_c \sum_l R(c,l))$ and $n_r(c)$ is the total number of runs corresponding to edge type $c$

$$\text{SPE} = \frac{1}{n_r} \sum_c \sum_l R(c,l)/l^2 \tag{1}$$

$$\text{SPE(c)} = \frac{1}{n_r(c)} \sum_l R(c,l)/l^2. \tag{2}$$

The *long path emphasis* gives higher weight to longer graph-edge runs than the shorter ones, multiplying the number of runs by the square of their lengths. Likewise, this feature, LPE, is calculated regardless of the edge types as well as $\text{LPE}(c)$ is calculated for each of the six edge types separately. The equations of these descriptors are given as follows:

$$\text{LPE} = \frac{1}{n_r} \sum_c \sum_l R(c,l) \times l^2 \tag{3}$$

$$\text{LPE(c)} = \frac{1}{n_r(c)} \sum_l R(c,l) \times l^2. \tag{4}$$

The *edge type nonuniformity* determines how the distribution of edge types affects the texture. It takes its lowest value when the runs are evenly distributed over all edge types. Similarly, the *path length nonuniformity* determines how the distribution of path lengths affects the texture. It takes its lowest values when the runs are evenly distributed over all path lengths. In the following equations, the edge type nonuniformity, ETN, and the path length nonuniformity, PLN, are defined as follows:

$$\text{ETN} = \frac{1}{n_r} \sum_c \left( \sum_l R(c,l) \right)^2 \tag{5}$$

$$\text{PLN} = \frac{1}{n_r} \sum_l \left( \sum_c R(c,l) \right)^2. \tag{6}$$
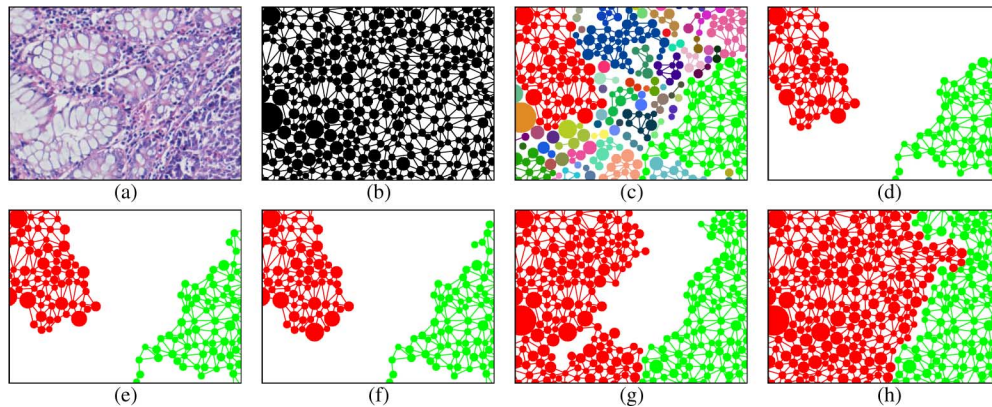
Fig. 6. The illustration of the segmentation algorithm: (a) an original subimage, (b) its constructed graph [no color information is presented for better visualizing the subsequent steps], (c) graph connected components obtained after disconnecting dissimilar primitives [primitives and edges of the same component are shown with the same color], (d) initial seeds obtained after eliminating small-sized components, (e) grown seeds after one iteration, (f) grown seeds after two iterations, (g) grown seeds after 15 iterations, and (h) final grown seeds.

## III. SEGMENTATION ALGORITHM

The proposed approach employs a region growing algorithm that uses the graph run-length features for segmentation. In this algorithm, region growing is achieved on the primitives, not on the pixels as in the case of previous studies [19], [42]. For each primitive, a window is centered at the centroid of the primitive and a run-length matrix is accumulated over the matrices of the primitives that are located in this window, as explained in Section II-B. The graph run-length features calculated on this accumulated matrix are used as the descriptors of the primitive, which is located at the center of the window. In the subsequent steps of the algorithm, these descriptors are used in (dis)similarity calculation. In our algorithm, Euclidean distance is used as a dissimilarity measure.

In the seed determination step, seed regions are found using the neighborhood relations defined by the constructed graph. To this end, the distance between every pair of adjacent primitives is computed and a pair is disconnected if the distance between them is over *a distance threshold*. Then, the small-sized connected components that include less primitives than *a component size threshold* are eliminated and the remaining components are considered as the initial seeds. Fig. 6(b)–(d) illustrate the steps of seed determination for a small subimage shown in Fig. 6(a); here primitives and edges of the same component are shown with the same color.

In the region growing step, remaining primitives are iteratively assigned to the initial seed regions. In each iteration, primitives that are adjacent to at least one of the seed connected components are considered. A primitive is assigned to its closest seed if the distance between them is less than a grow threshold. Here, we start the grow threshold with the distance threshold, which is used in the seed determination step, and increase it by 10% in each iteration. Region growing continues until there are no unassigned primitives left. For a seed component, the run-length features are obtained averaging them over all of its primitives. Fig. 6(e)–(h) show the grown seeds obtained at the end of different iterations.

In the region merge step, adjacent regions are merged if the distance between them is less than *a merge threshold*. At the end of this step, the regions contain all of the primitives but not all of the pixels; primitives do not cover all of the pixels. Thus, to obtain the final regions, the Voronoi diagram of the primitives, which is the dual of their Delaunay triangulation, is found and the Voronoi polygon of each primitive is included to the region that the primitive belongs to.

## IV. EXPERIMENTS

### A. Dataset

We conduct our experiments on 150 images of colon biopsy samples that are randomly taken from the Pathology Department archives of Hacettepe School of Medicine. The samples consist of 5–6-$\mu$m-thick tissue sections that are stained with hematoxylin-and-eosin, which is routinely used to stain biopsies in hospitals. The images of these samples are taken with a Nikon Coolscope Digital Microscope using 5 × microscope objective lens and 1920 × 2560 image resolution. The tissue images are divided into training and test sets. The training set consists of 50 images that are used to estimate the model parameters. The test set consists of the remaining 100 images that are not used in parameter estimation at all.

Each image is heterogeneous and contains a mixture of normal regions and adenocarcinomatous (cancerous) regions of different grades. The first column of Fig. 8 shows the manual segmentation (gold standard) of these regions provided by a pathologist, who is specialized in colorectal carcinomas. In this figure, normal and adenocarcinomatous regions are labeled as N and AC, respectively.[2] In a tissue image, there may also exist some regions that can be included into either a cancerous or a normal region without affecting the medical assessment in the

---

[2]Colon adenocarcinoma originates from epithelial cells and causes organizational changes of these cells, leading to distortions in glands, which are formed of the epithelial cells. To locate adenocarcinomatous regions in a tissue image, regions containing cancerous epithelial cells (and cancerous glands) should be separated from those containing normal epithelial cells (and normal glands). In evaluating segmentation results, it is important how homogeneous segmented regions are in terms of their epithelial cells (and glands).
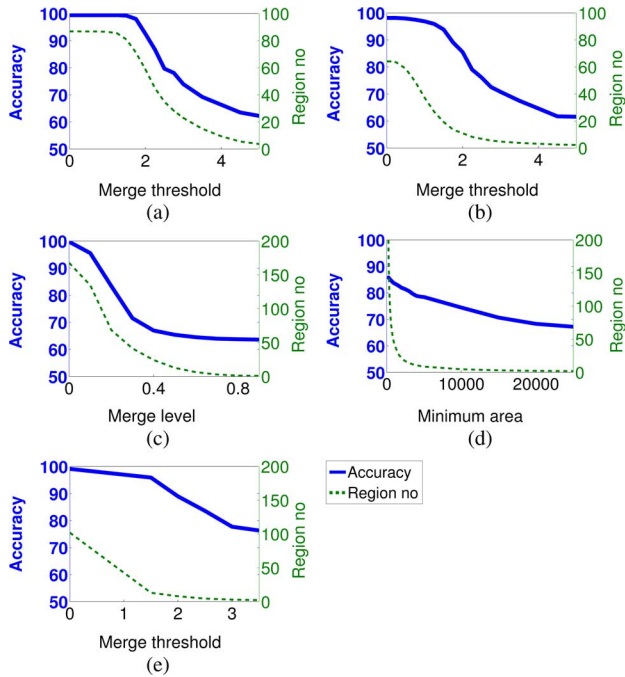
Fig. 7.  The segmentation accuracy and the number of segmented regions as a function of the merge threshold (minimum area) parameter. These results are obtained on the training samples for (a) graphRLM, (b) grayRLM, (c) JSEG, (d) GBS, and (e) objectSEG algorithms.

context of colon adenocarcinoma diagnosis[3]. Such regions are shown with gray shades in Fig. 8.

### B. Evaluation

In our experiments, we provide visual results obtained by the algorithms. Additionally, we quantitatively assess the results using two different criteria: the segmentation accuracy and the number of segmented regions. For computing the accuracy, true positive, false positive, true negative, and false negative pixels are calculated, comparing the segmentation results with the gold standard. Using these pixels, sensitivity and specificity are also computed.

The proposed algorithm and those that we use for comparison are unsupervised, and hence, they do not label their segmented regions. In order to compute the accuracy, we compare each segmented region with the gold standard and label it with the class of the region in the gold standard that mostly overlaps the segmented region (i.e., with the class of the dominant region in the gold standard). Therefore, the pixels located in the nonoverlapping parts of the segmented region are considered as either false positive or false negative, depending on the class of its dominant region. Note that, in our evaluations, we do not consider the pixels of regions that could be included in either a normal or a cancerous region.

### C. Comparisons

To investigate the effectiveness of graph run-length matrices (*graphRLM*), we compare the results of our proposed algorithm with those of four other approaches. In the first

[3]These regions do not contain any epithelial cells (and glands). Thus, they do not affect the assessment in the context of colon adenocarcinoma diagnosis.

approach, we implement the pixel-based counterpart of the proposed algorithm to examine the differences between the use of graph and gray-level run-length matrices. In this approach, the features used in segmentation are extracted from gray-level run-length matrices (*grayRLM*). For that, pixel intensities are quantized into three and seven texture features [41], [45] are defined on the run-length matrices computed at four different angles $(0, \pi/4, 2\pi/4, 3\pi/4)$. The remaining segmentation steps are exactly the same as those of our algorithm except that the grayRLM algorithm uses an area threshold to eliminate small-sized seeds instead of using a component size threshold.

The second approach is our previous work (*objectSEG*), which we specifically implement for the segmentation of histopathological images [42]. This algorithm also employs cytological tissue components to define its texture descriptors; however, it does not use a graph algorithm either in its texture definition or in its segmentation. Therefore, we include this algorithm in our comparisons to investigate the effectiveness of the use of graphs in texture definition.

The last two approaches are the JSEG algorithm, in which segmentation is achieved by defining a texture descriptor on the quantized pixels [19], and the graph-based algorithm (GBS), in which segmentation is achieved by employing a graph constructed over the pixels of an image [26]. We include them in our comparisons since they have been shown to be effective in many unsupervised segmentation problems although they are not specifically designed for histopathological images.

## V. RESULTS

### A. Parameter Selection

All approaches have different model parameters. We estimate the values of these parameters on the training samples. To this end, we determine a candidate set for each parameter, try all possible combinations of these candidate sets, and select the one that leads to the best performance on the training images. The parameters of each algorithm and their candidate values are summarized in Table II.

For each algorithm, we first select the best parameter set that leads to the best average accuracy without considering the number of segmented regions. Table III reports the average segmentation results obtained with such kind of parameter selection. Note that all results reported in this subsection are obtained on the training samples. As seen in this table, this selection leads to very high segmentation accuracies, but at the same time, very high number of segmented regions. The main reason of having such high number of regions is that as the accuracy before a merge step cannot be lower than the accuracy after it, the merge parameters are always selected as 0 (i.e., no oversegmented regions are merged). Therefore, we decide to explicitly investigate the effects of a region merge step by calling it with different merge threshold parameters (minimum area parameter of the GBS algorithm, which controls its merge step) right after obtaining the regions. For algorithms, Fig. 7 shows the average accuracy and the average number of segmented regions as a function of their merge parameters. This figure shows that a reasonable number of segmented regions can only be obtained with lower accuracy values. When we

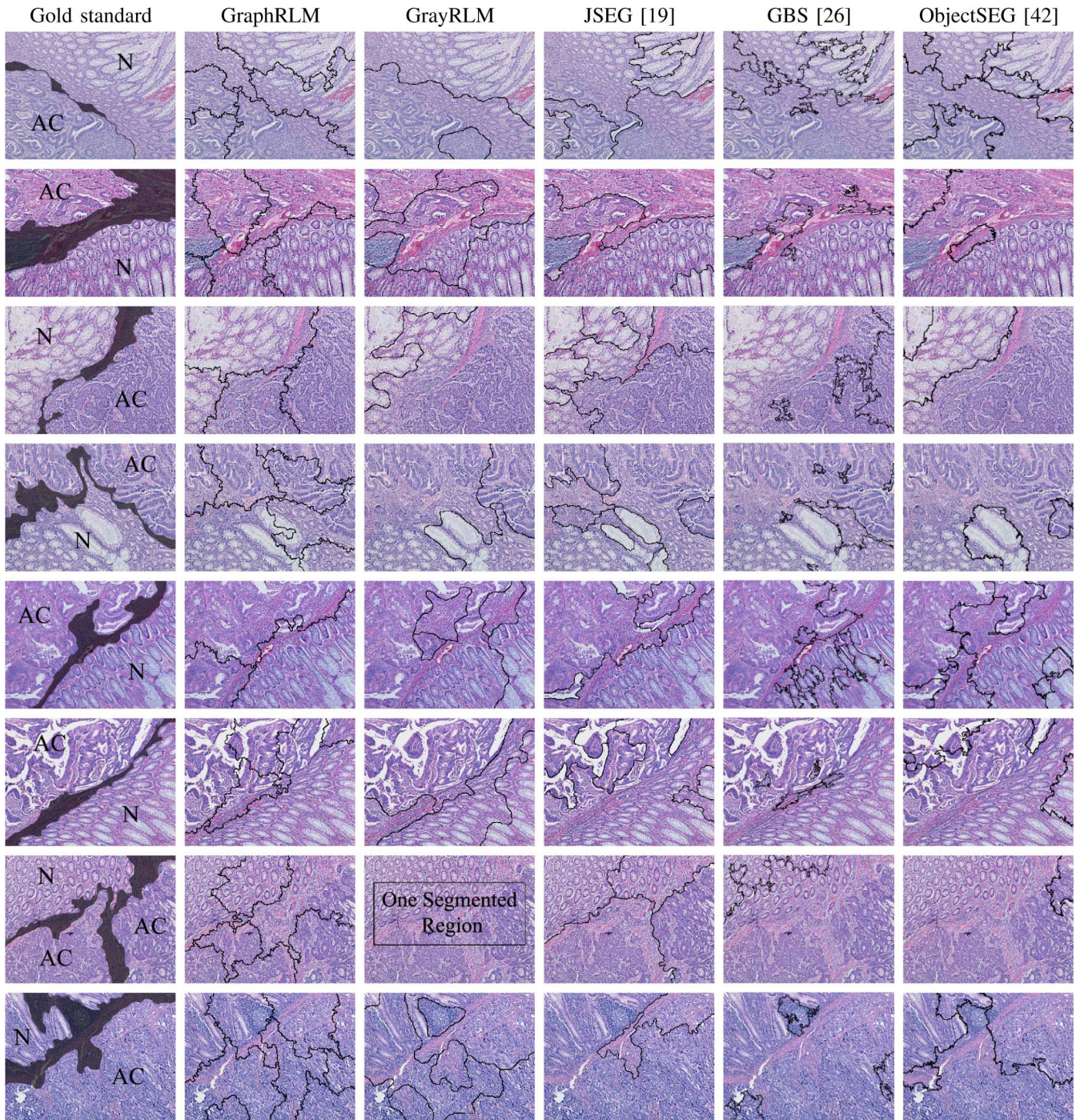| Gold standard | GraphRLM | GrayRLM | JSEG [19] | GBS [26] | ObjectSEG [42] |



Fig. 8. The visual results on example images. These results are obtained when only the parameter combinations that give at most 10 regions are considered.

examine the visual results to understand its reason, we observe that it is not possible to find a common merge parameter that works for all images and it is necessary to select different merge parameters for different images. This is indeed what we observed in our previous work [42], in which we had to optimize this parameter for each image separately for both the objectSEG and JSEG algorithms.

Thus, we include the number of segmented regions into the parameter selection criteria, setting an upperbound $N$ on the number and considering only the parameter sets that yield at most $N$ number of segmented regions. As most of the images

contain 2–3 regions in the gold standard, we select the value of $N$ as 5 and 10. The upperbound $N$ is used to express the trade-off between the accuracy and the number of segmented regions. Allowing upperbounds that are greater than the expected number of regions increases the accuracy at the cost of obtaining oversegmented results. Tables IV(a) and (b) report the quantitative results when $N = 5$ and $N = 10$, respectively.[4] Table IV(a) shows that the restriction of having at most 5 regions causes

---

[4]These results are different than those reported in [42] because the merge parameters were selected for each image separately for both the objectSEG and JSEG algorithms in [42]. Thus, higher accuracies could be obtained.

TABLE II
PARAMETERS OF THE ALGORITHMS AND THEIR VALUES THAT ARE
CONSIDERED IN THE ESTIMATION OF THE BEST PARAMETER SETS

| **graphRLM** |
|---|
| Window size $= \{32, 64, \mathbf{96}, 128\}$ |
| Distance threshold $= \{0.25, 0.50, ..., \mathbf{1.25}, ..., 3.00, 3.50, 4.00\}$ |
| Component size threshold $= \{10, 25, 50, ..., \mathbf{100}, 150, 200, 250\}$ |
| Merge threshold $= \{\mathbf{0.00}, 0.10, 0.25, 0.50, ..., 3.00, 3.50, 4.00\}$ |
| **grayRLM** |
| Window size $= \{32, 64, 96, 128\}$ |
| Distance threshold $= \{0.25, 0.50, ..., 3.00, 3.50, 4.0\}$ |
| Area threshold $= \{5000, 10000, ..., 60000\}$ |
| Merge threshold $= \{0.00, 0.10, 0.25, 0.50, ..., 3.00, 3.50, 4.00\}$ |
| **JSEG** |
| Quantization threshold $= \{10, 25, 50, 75, 100, 150, ..., 450\}$ |
| Scale level $= \{1, 2, 3\}$ |
| Merge level $= \{0.0, 0.1, ..., 0.9\}$ |
| **GBS** |
| Gaussian sigma $= \{0.1, 0.2, ..., 1.0\}$ |
| Scale $= \{50, 100, ..., 1000, 1500, ..., 2500\}$ |
| Min area $= \{100, 200, ..., 1000, 1500, ..., 5000, 10000, ..., 25000\}$ |
| **objectSEG** |
| Large window size $= \{96, 128, 160, 192\}$ |
| Small window size $= \{32, 64\}$ |
| Merge threshold $= \{0.00, 1.00, 1.25, ..., 3.50\}$ |

TABLE III
AVERAGE AND STANDARD DEVIATION OF SEGMENTATION RESULTS OBTAINED
ON THE TRAINING SAMPLES. PARAMETER SETS ARE SELECTED ON THE
TRAINING SAMPLES WITHOUT ANY RESTRICTION ON THE NUMBER
OF THE SEGMENTED REGIONS

|  | Accuracy | Sensitivity | Specificity | Region no |
|---|---|---|---|---|
| graphRLM | $99.0 \pm 1.3$ | $99.2 \pm 1.2$ | $98.7 \pm 2.2$ | $76.2 \pm 8.8$ |
| grayRLM | $98.2 \pm 1.9$ | $98.5 \pm 2.0$ | $97.4 \pm 3.5$ | $50.1 \pm 15.8$ |
| JSEG | $97.0 \pm 1.5$ | $95.2 \pm 3.7$ | $97.8 \pm 1.5$ | $167.7 \pm 21.3$ |
| GBS | $86.0 \pm 5.1$ | $83.8 \pm 12.0$ | $85.5 \pm 12.0$ | $220.6 \pm 13.1$ |
| objectSEG | $99.0 \pm 1.2$ | $98.9 \pm 1.7$ | $98.9 \pm 2.4$ | $141.1 \pm 11.7$ |

TABLE IV
AVERAGE AND STANDARD DEVIATION OF SEGMENTATION RESULTS OBTAINED
ON THE TRAINING SAMPLES. PARAMETER SETS ARE SELECTED ON THE
TRAINING SAMPLES CONSIDERING ONLY THE PARAMETER COMBINATIONS
THAT GIVE AT MOST (A) 5 REGIONS AND (B) 10 REGIONS

|  | Accuracy | Sensitivity | Specificity | Region no |
|---|---|---|---|---|
| graphRLM | $87.1 \pm 13.6$ | $90.7 \pm 18.7$ | $79.2 \pm 33.7$ | $2.9 \pm 1.1$ |
| grayRLM | $77.2 \pm 14.3$ | $74.4 \pm 35.8$ | $71.1 \pm 38.5$ | $2.9 \pm 1.1$ |
| JSEG | $69.0 \pm 12.3$ | $46.8 \pm 48.0$ | $72.6 \pm 40.2$ | $2.4 \pm 1.3$ |
| GBS | $73.8 \pm 8.9$ | $58.6 \pm 34.7$ | $77.2 \pm 25.1$ | $3.7 \pm 0.9$ |
| objectSEG | $81.4 \pm 14.3$ | $80.0 \pm 30.5$ | $76.4 \pm 32.8$ | $3.2 \pm 0.9$ |

(a)

|  | Accuracy | Sensitivity | Specificity | Region no |
|---|---|---|---|---|
| graphRLM | $93.3 \pm 7.3$ | $92.8 \pm 12.4$ | $91.5 \pm 17.9$ | $6.1 \pm 1.8$ |
| grayRLM | $84.5 \pm 14.7$ | $83.7 \pm 30.6$ | $78.9 \pm 33.7$ | $4.7 \pm 2.3$ |
| JSEG | $87.9 \pm 7.9$ | $82.4 \pm 22.9$ | $89.7 \pm 14.3$ | $6.5 \pm 2.1$ |
| GBS | $77.0 \pm 8.4$ | $65.0 \pm 26.2$ | $81.5 \pm 18.7$ | $6.8 \pm 1.7$ |
| objectSEG | $87.6 \pm 11.8$ | $89.0 \pm 19.9$ | $83.8 \pm 25.1$ | $5.3 \pm 1.6$ |

(b)

TABLE V
AVERAGE AND STANDARD DEVIATION OF SEGMENTATION RESULTS OBTAINED
ON THE TEST SAMPLES. PARAMETER SETS ARE SELECTED ON THE TRAINING
SAMPLES CONSIDERING ONLY THE PARAMETER COMBINATIONS THAT GIVE
AT MOST 10 REGIONS

|  | Accuracy | Sensitivity | Specificity | Region no |
|---|---|---|---|---|
| graphRLM | $94.8 \pm 4.9$ | $95.6 \pm 6.5$ | $92.6 \pm 13.1$ | $5.9 \pm 1.4$ |
| grayRLM | $80.9 \pm 15.9$ | $82.7 \pm 30.9$ | $70.1 \pm 39.2$ | $4.3 \pm 2.1$ |
| JSEG | $90.4 \pm 6.7$ | $89.3 \pm 17.8$ | $88.6 \pm 14.9$ | $7.8 \pm 2.7$ |
| GBS | $73.7 \pm 9.7$ | $62.3 \pm 32.2$ | $76.3 \pm 27.5$ | $4.9 \pm 1.6$ |
| objectSEG | $92.6 \pm 9.1$ | $93.1 \pm 17.4$ | $89.3 \pm 17.3$ | $6.0 \pm 1.9$ |

However, sometimes, small regions, on which it is hard to define distinctive features, are also obtained in spite of an upperbound; for instance, the results of the GBS algorithm shown in Fig. 8 consist of such small regions although the number of segmented regions is smaller than 10. When we examine the results of our algorithm, we do not observe such regions for the training images and observe only two small regions for the test images.

*B. Test Results*

After selecting the parameters on the training samples, we test our algorithm on the 100 test images. Table V reports the results. It shows that similar results are obtained for the test samples. Although there is a slight increase in the accuracy of the proposed algorithm for the test samples, the t-test shows that this increase is not statistically significant. This table also shows that our algorithm improves the accuracy of the other algorithms; this improvement is statistically significant with a significance level of 0.05. To understand the reasons of this improvement, we examine the visual results and observe that the other algorithms cause larger variations in their segmentation results; although they are good for some images, they are bad for others. This can be seen on the example images shown in Fig. 8. The variations are indeed due to the difficulty for these algorithms to select a common parameter set that works for all images. On the other hand, the proposed algorithm gives better segmentation results by selecting a better common parameter set that works for more images.

The experiments demonstrate that the proposed algorithm improves the results of the others. The grayRLM algorithm is im-

lower accuracies. This is attributed to the following two behaviors of the algorithms. They either eliminate some important initial seed regions to start with less seeds in the seed determination step. Or they tend to merge heterogeneous segmented regions in the region merge step to keep the number of regions smaller than or equal to 5. Selecting $N = 10$ alleviates the effects of these behaviors, and hence, increases the accuracy of all algorithms. As seen in Table IV(b), the number of segmented regions reported for $N = 10$ is much less than the one given in Table III.

As mentioned in the introduction, most of the studies that focus on histopathological image analysis develop classification tools. These tools assume that a given image is homogeneous, and hence, extract features from the entire image and use them for its classification. The main purpose of histopathological image segmentation is to provide homogeneous regions to the classification tools. Thus, oversegmentation up to a point is usually acceptable in terms of medical interpretation as long as the regions are homogeneous and they are large enough such that the classification tools could define distinctive features on them. Considering the magnification of the microscope objective lens used in our experiments and the tissue area covered by a single snapshot, segmentation results with an upperbound 10 usually give large enough regions to extract useful information. This can be seen in the segmentation of example images (Fig. 8).

plemented as the pixel-based counterpart of our algorithm to investigate the difference between using graph and gray-level run-length matrices. The comparisons show that texture descriptors defined on cytological tissue components using graph run-length matrices is more effective than those defined on pixel intensities using gray-level run-length matrices. The component-based textures are more successful in incorporating the background knowledge into segmentation. The objectSEG algorithm defines its texture descriptors on tissue components in a different way; it does not employ graphs in this definition. However, the comparisons show that the graphs are more efficient in terms of defining component-based textures. The JSEG and GBS algorithms are the examples of effective algorithms for color-texture image segmentation. The comparisons point to the ill-posedness of the problem. Although algorithms may give good results in general, there is a need of algorithms that are particularly designed for special types of images. Incorporating the background knowledge specific to such images, the segmentation algorithms have the potential of improving their results. For example, the $J$ criterion used by the JSEG algorithm could be defined on cytological components of a tissue. In our previous work [42], we defined such kind of a criterion and used it as one of the texture measures in region growing. Similarly, the GBS algorithm could be adapted to work on tissue components rather than pixels; for example, it could construct its graph on components and define edge weights based on component similarities. Such adaptations could be considered as the future aspects of our study.

### C. Parameter Analysis

The effects of each parameter on the segmentation results are also investigated. For that, three of the four parameters are fixed and the accuracy and the number of segmented regions are observed as a function of the other parameter. In Table II, the selected parameter values are indicated in bold. Fig. 9 shows the parameter analysis performed on the test images.

The window size determines the size of a region, on which texture descriptors are defined for a single component. Larger values give too generic descriptors, which make adjacent components more similar. This results in more components being grouped in the same seed, and hence, larger but a smaller number of seed regions. In general, a smaller number of initial seeds leads to less segmented regions, which usually causes higher segmentation errors. On the other hand, too smaller values give too specific descriptors, which make adjacent components less similar. This gives small-sized initial seeds that are mostly eliminated. This decreases the number of segmented regions, and hence, lowers the accuracy.

The distance threshold determines at what similarity level the components form a single seed. Larger values lead to less seeds that are larger in size and contain more dissimilar components. This decreases the accuracy and the number of segmented regions. If it is too small, the components cannot form large enough seeds that remain uneliminated at the end of the seed determination step. This also decreases the accuracy and the number of segmented regions.
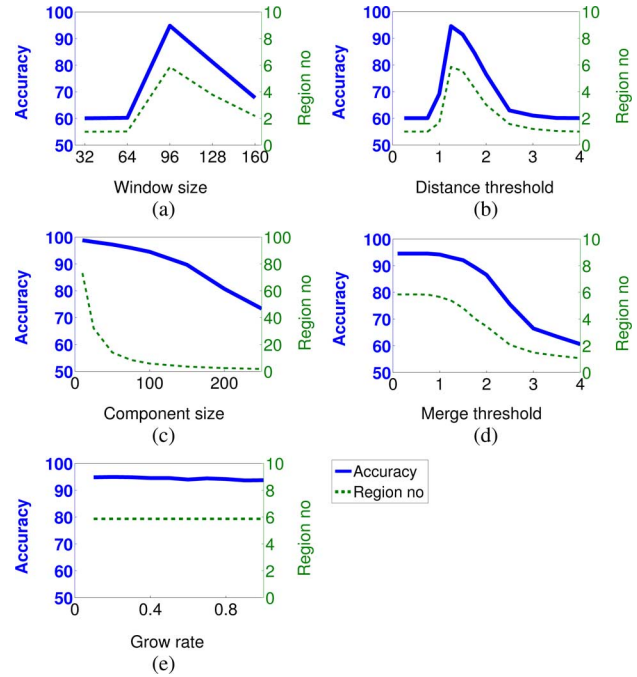


Fig. 9. The segmentation accuracy and the number of segmented regions as a function of the model parameters: (a) window size, (b) distance threshold, (c) component size threshold, (d) merge threshold, and (e) grow rate percentage.

The component size threshold is used to eliminate small-sized components in the seed determination step. If it is too large, most of the components are eliminated. This decreases the number of segmented regions, and hence, the accuracy. If it is too small, small-sized groups are also selected as seeds. This results in very large number of segmented regions at the end, which increases the accuracy.

The merge threshold determines at what similarity level the seeds are merged after the region growing step. Larger values result in more and more seeds being merged into a single region. This decreases the number of segmented regions and the accuracy. On the other hand, smaller values lead to less seeds being merged. Hence, the number of segmented regions and the accuracy tend to be higher. In our experiments, this parameter is selected as 0.00, which means that no merge operation is performed.[5] However, the number of segmented regions is comparable with those of the other algorithms, which perform region merge operation.

In addition to these parameters, the algorithm contains two implicit choices: the grow threshold percentage (grow rate) in region growing and the selection of a dissimilarity measure. In the algorithm, the grow rate is fixed to 0.1. To investigate the effects of its selection, we fix all parameters and change the grow rate from 0.1 to 1.0 in the increments of 0.1. The test results given in Fig. 9(e) shows that the grow rate only slightly affects

---

[5]A smaller number of segmented regions (less oversegmented results) could be obtained in two different ways: starting with a smaller number of initial seeds at the beginning and/or merging oversegmented regions at the end. In the proposed algorithm, the former one is controlled by the window size, distance threshold, and component size threshold parameters whereas the latter one is controlled by the merge threshold parameter. In our experiments, although the merge threshold is selected to be 0.0, the other parameters are selected such that the algorithm generates at most 10 regions (when $N = 10$).

TABLE VI
DISSIMILARITY MEASURES ANALYZED IN THE EXPERIMENTS

| Euclidean distance | $D(u,v) = \sqrt{\sum_i (u_i - v_i)^2}$ |
| Bhattacharya distance | $D(u,v) = \ln \sum_i \sqrt{(u_i \cdot v_i)}$ |
| Clark distance | $D(u,v) = \sqrt{\sum_i \dfrac{(u_i - v_i)^2}{(u_i + v_i)^2}}$ |
| K-divergence | $D(u,v) = \sum_i u_i \ln \dfrac{2\,u_i}{(u_i + v_i)}$ |

TABLE VII
EFFECTS OF A DISSIMILARITY MEASURE ON THE SEGMENTATION
ACCURACY AND THE NUMBER OF SEGMENTED REGIONS

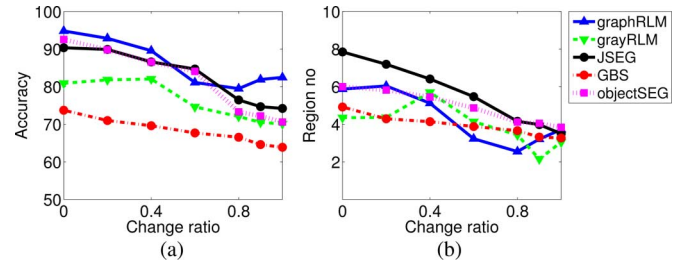|  | Accuracy | Region no |
| --- | --- | --- |
| Euclidean distance | $94.8 \pm 4.9$ | $5.9 \pm 1.4$ |
| Clark distance | $92.0 \pm 9.1$ | $6.2 \pm 1.7$ |
| Bhattacharya distance | $84.5 \pm 12.1$ | $4.0 \pm 1.4$ |
| K-divergence | $77.9 \pm 13.8$ | $3.9 \pm 1.6$ |



Fig. 10. The effects of the contrast change ratio on (a) the segmentation accuracy and (b) the number of segmented regions.

TABLE VIII
COMPUTATIONAL TIMES OF THE ALGORITHMS

| graphRLM (grow rate = 0.1) | $52.9 \pm 3.2$ |
| --- | --- |
| graphRLM (grow rate = 0.5) | $30.8 \pm 2.2$ |
| graphRLM (grow rate = 1.0) | $26.4 \pm 2.8$ |
| grayRLM | $172.5 \pm 103.4$ |
| JSEG | $19.1 \pm 4.2$ |
| GBS | $11.7 \pm 2.5$ |
| objectSEG | $7242.3 \pm 265.4$ |

the accuracy and the number of segmented regions. On the other hand, the grow rate affects the speed of segmentation. When it is selected as 0.5, the average running time decreases from 52.9 s to 30.8 s.

To analyze the effects of using different dissimilarity measures, we select four different measures from four different dissimilarity families: 1) Euclidean distance from the Lp Minkowski family, 2) Bhattacharya distance from the Fidelity family, 3) K-divergence from the Shannon's entropy family, and 4) Clark distance from the $\chi^2$ family. The definition of these measures are given in Table VI with $D(u,v)$ being the dissimilarity between feature vectors $u$ and $v$. In the analysis, we fix all parameters other than the ones that are used to measure dissimilarity and select the others on the training samples. Table VII reports the results obtained on the test samples. It shows that Euclidean and Clark distances give better results. This indicates the importance of using the correct dissimilarity measure. It also shows that one could use different measures to obtain good segmentations.

### D. Robustness Analysis

To understand the robustness of our algorithm with respect to local distortions, we analyze the effects of changes in image contrast on segmentation results. For that, we increase the contrast of a test image by saturating the image at its low and high pixel values and mapping its remaining pixels to the original interval. The contrast change ratio $R$ determines the low and high saturation points such that $1 - R$ of the original interval falls in between these points. In our experiments, we estimate the parameters on the undistorted training images and observe the test results as a function of the contrast change ratio (Fig. 10). The results of all algorithms show that the accuracy decreases with the increasing ratios. However, our algorithm still yields high ($\geq 90\%$) accuracies when the ratio $\leq 0.4$. When the ratio becomes 0.6, it gives lower accuracies for some images. We an-

alyze these images and observe that their pink primitives are largely affected by the contrast change such that the number of pink primitives decreases and the remaining ones look like noisy components, which decreases the accuracy. For these images, pink primitives almost disappear when the ratio reaches to 1.0. This disappearance alleviates the look of noisy pink components, and hence, slightly increases the accuracy. Fig. 10 also shows that our algorithm gives the best accuracies except the case when the ratio is 0.6.

### E. Computational Time Analysis

The proposed approach first transforms image pixels into a primitive domain and then uses this domain throughout the remaining steps of the segmentation algorithm. Thus, after this transformation, its computational complexity depends on the number of primitives in an image, which is much less than the number of image pixels. The computational time required for processing a single image is 52.9 s on the average. This result is obtained on a computer with a Core2Duo 2.8 GHz processor and 3 GB of RAM. As mentioned before, the grow rate affects the speed of segmentation; the computational times for different grow rates are given in Table VIII. This table also reports the computational times of the other algorithms.

## VI. CONCLUSION

This paper presents a new algorithm for the unsupervised segmentation of histopathological images. It proposes to incorporate the background knowledge that is specific to histopathological images into segmentation. For this purpose, it introduces a new set of texture descriptors that quantify the spatial distribution of cytological tissue components with the help of a graph constructed on these components.

The proposed algorithm is tested on 150 images of colon tissues that contain normal and cancerous regions. The experiments show that the proposed algorithm gives accurate segmentation results, providing a reasonable number of segmented re-

gions. Moreover, they also show that the algorithm enables to select a common parameter that leads to good segmentation results. Compared with four other algorithms, the results show that the proposed algorithm is more effective in the segmentation of histopathological images.

One future research direction is to use the proposed texture descriptors for supervised classification. The descriptors extracted from the segmented regions could be used for cancer diagnosis and grading. Although the algorithm is particularly designed for histopathological images, the new texture definition has a potential to be used in different applications such as remote sensing image analysis. For that, one could define primitives on the dominant colors in a similar way and construct a graph to define the texture of these primitives. This would be another future research direction of the paper.

## ACKNOWLEDGMENT

## REFERENCES

[1] I. D. Nagtegaal and J. H. J. M. van Krieken, "The role of pathologists in the quality control of diagnosis and treatment of rectal cancer—An overview," *Eur. J. Cancer*, vol. 38, no. 7, pp. 964–972, 2002.

[2] Referral guidelines for suspected cancer—NICE Guideline Nat. Inst. Health Clinical Excellence, London, U.K., 2005.

[3] J. Barret, M. Jiwa, P. Rose, and W. Hamilton, "Pathways to the diagnosis of colorectal cancer: An observational study in three UK cities," *Fam. Pr.*, vol. 23, no. 1, pp. 15–19, 2006.

[4] A. Andrion, C. Magnani, P. G. Betta, A. Donna, F. Mollo, M. Scelsi, P. Bernardi, M. Botta, and B. Terracini, "Malignant mesothelioma of the pleura: Inter-observer variability," *J. Clin. Pathol.*, vol. 48, no. 9, pp. 856–860, 1995.

[5] C. Wittekind and L. H. Sobin, *TNM Classification of Malignant Tumours*. New York: Wiley, 2002.

[6] A. N. Esgiar, R. N. G. Naguib, B. S. Sharif, M. K. Bennett, and A. Murray, "Microscopic image analysis for quantitative measurement and feature identification of normal and cancerous colonic mucosa," *IEEE Trans. Inf. Technol. Biomed.*, vol. 2, no. 3, pp. 197–203, Sep. 1998.

[7] B. Weyn, G. Van De Wouwer, M. Koprowski, A. Van Daele, K. Dhaene, and P. Scheunders, "Value of morphometry, texture analysis, densitometry, and histometry in the differential diagnosis and prognosis of malignant mesothelioma," *J. Pathol.*, vol. 189, no. 4, pp. 581–589, 1999.

[8] M. Wiltgen, A. Gerger, and J. Smolle, "Tissue counter analysis of benign common nevi and malignant melanoma," *Int. J. Med. Inform.*, vol. 69, no. 1, pp. 17–28, 2003.

[9] C. Demir, S. H. Gultekin, and B. Yener, "Learning the topological properties of brain tumors," *IEEE-ACM Trans. Comput. Biol. Bioinform.*, vol. 2, no. 3, pp. 262–270, 2005.

[10] O. Sertel, J. Kong, U. V. Catalyurek, G. Lozanski, J. H. Saltz, and M. N. Gurcan, "Histopathological image analysis using model-based intermediate representations and color texture: Follicular lymphoma grading," *J. Signal Process. Syst.*, vol. 55, pp. 169–183, 2009.

[11] O. Sertel, J. Kong, H. Shimada, U. V. Catalyurek, J. H. Saltz, and M. N. Gurcan, "Computer-aided prognosis of neuroblastoma on whole slide images: Classification of stromal development," *Pattern Recognit.*, vol. 42, no. 6, pp. 1093–1103, 2009.

[12] P. Scheunders, "A genetic c-means clustering algorithm applied to color image quantization," *Pattern Recognit.*, vol. 30, no. 6, pp. 859–866, 1997.

[13] T. Q. Chen and Y. Lu, "Color image segmentation: An innovative approach," *Pattern Recognit.*, vol. 35, no. 2, pp. 395–405, 2002.

[14] L. Shafarenko, M. Petrou, and J. Kittler, "Histogram based segmentation in a perceptually uniform color space," *IEEE Trans. Image Process.*, vol. 7, no. 9, pp. 1354–1358, Sep. 1998.

[15] E. Littmann and H. Ritter, "Adaptive color segmentation—A comparison of neural and statistical methods," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 175–185, 1997.

[16] H. D. Cheng, X. H. Jiang, and J. Wang, "Color image segmentation based on homogram thresholding and region merging," *Pattern Recognit.*, vol. 35, no. 2, pp. 373–393, 2002.

[17] F. Kurugollu, B. Sankur, and A. Harmanci, "Color image segmentation using histogram multithresholding and fusion," *Image Vis. Comput.*, vol. 19, no. 13, pp. 915–928, 2001.

[18] M. Mignotte, "Segmentation by fusion of histogram-based k-means clusters in different color spaces," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 780–787, May 2008.

[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] F. Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," *Image Vis. Comput.*, vol. 23, no. 10, pp. 877–886, 2005.

[21] M. Krinidis and I. Pitas, "Color texture segmentation based on the modal energy of deformable surfaces," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1613–1622, Jul. 2009.

[22] D. Panjwani and G. Healey, "Markov random field models for unsupervised segmentation of textured color images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 939–954, Oct. 1995.

[23] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. Image Process.*, vol. 6, no. 11, pp. 1530–1544, Nov. 1997.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[25] S. Wang and J. M. Siskind, "Image segmentation with ratio cut," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 6, pp. 675–690, Jun. 2003.

[26] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[27] J.-S. Kim and K.-S. Hong, "Color-texture segmentation using unsupervised graph cuts," *Pattern Recognit.*, vol. 42, no. 5, pp. 735–750, 2009.

[28] Y.-W. Tai, J. Jia, and C.-K. Tang, "Soft color segmentation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1520–1537, Sep. 2007.

[29] Z. Tu, C. Narr, P. Dollar, I. Dinov, P. Thompson, and A. Toga, "Brain anatomical structure segmentation by hybrid discriminative/generative models," *IEEE Trans. Med. Imag.*, vol. 27, no. 4, pp. 495–508, Apr. 2008.

[30] Z. Kato and T. C. Pong, "A Markov random field image segmentation model for color textured images," *Image Vis. Comput.*, vol. 24, no. 10, pp. 1103–1114, 2006.

[31] L. Zhang and Q. Ji, "Image segmentation with a unified graphical model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1406–1425, 2010.

[32] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, "Adaptive perceptual color-texture image segmentation," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1524–1536, Oct. 2005.

[33] 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.

[34] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 142–159, Feb. 2000.

[35] R. Farjam, H. Soltanian-Zadeh, K. Jafari-Khouzani, and R. A. Zoroofi, "An image analysis approach for automatic malignancy determination of prostate pathological images," *Cytom. Part B—Clin. Cytom.*, vol. 72B, no. 4, pp. 227–249, 2007.

[36] C. Wittke, J. Mayer, and F. Schweiggert, "On the classification of prostate carcinoma with methods from spatial statistics," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 4, pp. 406–414, Jul. 2007.

[37] S. Naik, S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology," in *Proc. 5th IEEE Int. Symp. Biomed. Imag.: From Nano to Macro*, 2008, pp. 284–287.

[38] C. Gunduz-Demir, M. Kandemir, A. B. Tosun, and C. Sokmensuer, "Automatic segmentation of colon glands using object-graphs," *Med. Image Anal.*, vol. 14, no. 1, pp. 1–12, 2010.

[39] J. Smolle, "Computer recognition of skin structures using discriminant and cluster analysis," *Skin Res. Technol.*, vol. 6, no. 2, pp. 58–63, 2000.

[40] Y. Wang, D. Crookes, O. S. Eldin, S. Wang, P. Hamilton, and J. Diamond, "Assisted diagnosis of cervical intraepithelial neoplasia (CIN)," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 1, pp. 112–121, 2009.

[41] M. M. Galloway, "Texture analysis using gray level run lengths," *Comput. Vis. Graph.*, vol. 4, pp. 172–179, 1975.

[42] A. B. Tosun, M. Kandemir, C. Sokmensuer, and C. Gunduz-Demir, "Object-oriented texture analysis for the unsupervised segmentation of biopsy images for cancer detection," *Pattern Recognit.*, vol. 42, no. 6, pp. 1104–1112, 2009.

[43] D. Altunbay, C. Cigir, C. Sokmensuer, and C. Gunduz-Demir, "Color graphs for automated cancer diagnosis and grading," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 3, pp. 665–674, Mar. 2010.

[44] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, U.K.: MIT Press, 2004.

[45] A. Chu, C. M. Sehgal, and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognit. Lett.*, vol. 11, no. 6, pp. 415–420, 1990.