Multilevel Segmentation of Histopathological Images Using Cooccurrence of Tissue Objects

Ahmet Cagri Simsek, Akif Burak Tosun, *Student Member, IEEE*, Cevdet Aykanat, *Member, IEEE*, Cenk Sokmensuer, and Cigdem Gunduz-Demir*, *Member, IEEE*

***Abstract*—This paper presents a new approach for unsupervised segmentation of histopathological tissue images. This approach has two main contributions. First, it introduces a new set of high-level texture features to represent the prior knowledge of spatial organization of the tissue components. These texture features are defined on the tissue components, which are approximately represented by tissue objects, and quantify the frequency of two component types being cooccurred in a particular spatial relationship. As they are defined on components, rather than on image pixels, these *object cooccurrence features* are expected to be less vulnerable to noise and variations that are typically observed at the pixel level of tissue images. Second, it proposes to obtain multiple segmentations by multilevel partitioning of a graph constructed on the tissue objects and combine them by an ensemble function. This *multilevel graph partitioning* algorithm introduces randomization in graph construction and refinements in its multilevel scheme to increase diversity of individual segmentations, and thus, improve the final result. The experiments on 200 colon tissue images reveal that the proposed approach—the object cooccurrence features together with the multilevel segmentation algorithm—is effective to obtain high-quality results. The experiments also show that it improves the segmentation results compared to the previous approaches.**

***Index Terms*—Histopathological image analysis, image segmentation, multilevel segmentation, segmentation ensemble, texture.**

I. Introduction

DIGITAL pathology is becoming an increasingly important tool for automated biopsy analysis. Automated analysis of histopathological tissue images not only increases throughput but also improves reproducibility. Digital pathology systems have been implemented for different purposes including classification [1]–[3], retrieval [4], [5], and segmentation, which can further be categorized into two in terms of its objective. Cell/gland segmentation [6], [7] aim to locate cells and glands on a tissue image, whereas tissue image segmentation, which is the focus of this paper, aims to divide a heterogeneous tissue image into its homogeneous regions.

There exist only few studies on tissue image segmentation. The majority of them carry out grid analysis for segmentation [8]–[11]. To this end, they divide a tissue image into equal-sized grids, extract their features, and assign them to one of the predefined labels using a classifier. The connected grids of the same label are then considered to be in the same segmented region. Another group of studies consider pixels instead of grids [12], [13]. These studies characterize the pixels using features extracted within their neighborhood and process them to construct a segmented region. In characterizing a grid or a pixel, these studies use low-level color/texture descriptors that they define on image pixels. On the other hand, it is usually harder to express prior medical knowledge using only the pixel level descriptors. This may become even harder for tissue images, which typically consist of a considerable amount of variation and noise at their pixel level.

Our recent studies have introduced high-level descriptors for tissue image representation. These descriptors are designed to incorporate prior knowledge into segmentation. For that, we find approximate locations of tissue components, define texture features on the components instead of directly defining them on pixels, and use the features in a segmentation algorithm [14], [15]. These studies focus on defining effective high-level descriptors whereas they achieve segmentation using a standard region growing algorithm, for which there is a risk of trapping into local optima especially when initial seeds are not correctly selected [16]. In region growing algorithms, seed selection is commonly controlled by parameters and it is usually hard to find common parameter values that work well for different images especially when these images show large variation, as in the case of tissue image segmentation. Parameter values that yield good seeds for a set of images may lead to under- or over-segmentation for others; this raises the risk of obtaining local optimal results. To increase the likelihood of attaining a global optimum, it has been proposed to obtain multiple results and combine them by an ensemble [17]. Two important factors affecting the final result of the ensemble are the quality and diversity of individual results [18].

In this paper, we propose a new framework for unsupervised segmentation of tissue images. In this framework, our main contributions are twofold. First, we define a new set of high-level texture descriptors to quantify the organization of tissue components in an image. This definition relies on modeling how frequently a tissue component of a particular type coccurs with the other in given distances. Second, we implement a

Manuscript received December 16, 2011; revised March 8, 2012; accepted March 17, 2012. Date of publication March 23, 2012; date of current version May 18, 2012. *Asterisk indicates corresponding author.*

A. C. Simsek, A. B. Tosun, and C. Aykanat are with the Department of Computer Engineering, Bilkent University, Ankara TR-06800, Turkey (e-mail: asimsek@cs.bilkent.edu.tr; tosun@cs.bilkent.edu.tr; aykanat@cs.bilkent.edu.tr).

C. Sokmensuer is with the Department of Pathology, Hacettepe University Medical School, Ankara TR-06100, Turkey (e-mail: csokmens@hacettepe.edu.tr).

*C. Gunduz-Demir is with the Department of Computer Engineering, Bilkent University, Ankara TR-06800, Turkey (e-mail: gunduz@cs.bilkent.edu.tr).

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

Digital Object Identifier 10.1109/TBME.2012.2191784

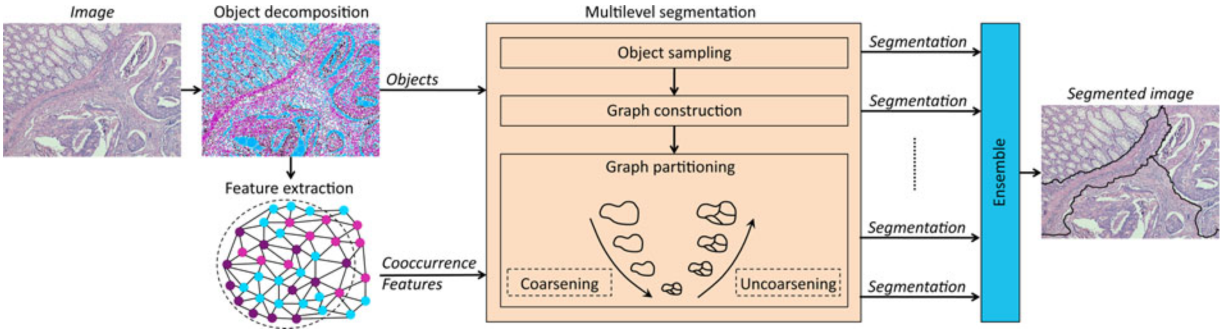


Fig. 1. Schematic overview of the proposed approach.

new tissue image segmentation algorithm that uses a multilevel scheme to obtain multiple segmentations and combines them by an ensemble. In this scheme, we construct a graph on a selected subset of the tissue components and achieve segmentation by multilevel partitioning of the constructed graph. We redesign the uncoarsening phase of this multilevel scheme to increase the diversity, and hence, the performance of the ensemble. Moreover, this multilevel scheme is invoked many times with selecting different component subsets in the sake of increasing the diversity further. Working with 200 colon tissue images, our experiments show that the proposed approach—the new descriptors together with the multilevel segmentation algorithm—is effective in obtaining high-quality results. The experiments also show that our approach leads to better results compared against its counterparts.

The proposed approach differs from the previous grid- and pixel-based studies in the following aspects. First, it uses high-level descriptors, which are defined to incorporate prior medical knowledge into segmentation, whereas these previous studies employ low-level descriptors directly defined on image pixels. Second, our approach does not require a labeled training set since it is an unsupervised algorithm. On the other hand, the previous studies necessitate obtaining training samples along with their labels since they train classifiers to classify the grids or pixels with one of the predefined labels.

Our current approach is also different than our previous algorithms, namely the *objectSEG* [14] and *graphRLM* [15] algorithms, in terms of its descriptor definition and its segmentation algorithm. The algorithm proposed in this study defines texture features based on second-order statistics among the tissue components. To this end, it measures the frequency of two component types being cooccurred in a particular spatial relation. On the other hand, the *objectSEG* algorithm defines its texture considering the spatial arrangement of the tissue components with respect to a reference point and the *graphRLM* algorithm uses a texture definition based on higher order statistics among the components. Moreover, to get closer to global optimal results, our current approach obtains multiple segmentations with a multilevel graph partitioning algorithm and combines them in an ensemble scheme. However, both of our previous algorithms use region growing for segmentation, and hence, have a higher risk of obtaining local optimal results.

II. Methodology

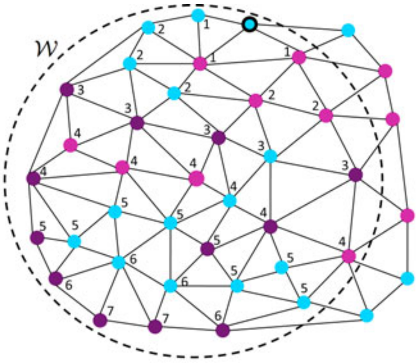
The proposed approach relies on characterizing a tissue image with high-level texture features and using them in an efficient segmentation algorithm. To this end, we introduce *object cooccurrence features* that quantify the spatial organization of tissue components. These features are extracted by decomposing a tissue image into a set of objects of different types, which approximately represent the tissue components, and calculating the frequency of the cooccurrence of two object types with respect to different distances.

Image segmentation is then achieved by partitioning the objects according to their cooccurrence features. In this approach, we propose to obtain multiple object partitions (segmentations) and combine them with an ensemble function. For an object partitioning step, a weighted graph is constructed on a selected subset of the objects, with an edge weight being defined as the similarity of its end points with respect to their features, and *a multilevel graph partitioning algorithm* is used to obtain a partition. In order to increase the diversity of the partitions, we propose to make modifications in the partitioning algorithm by eliminating initial partitioning, randomizing boundary refinements, and removing the balance constraint of the partitions. An overview of the proposed approach is given in Fig. 1.

A. Feature Extraction

The proposed approach introduces a texture measure to quantify the spatial organization of components in a tissue. To this end, it transforms a tissue image from the pixel domain to the component domain and defines texture on the tissue components instead of defining it on pixel values. In this new domain, the components are approximately represented by three types of circular objects[1]. These are white objects that correspond to luminal regions and epithelial cell cytoplasms, pink objects that correspond to stromal regions, and purple objects that correspond to cell nuclei. To define these objects, image pixels

[1] The exact localization of the components is quite difficult in our problem domain, which focuses on segmentation of low-magnification tissue images. Thus, we approximately locate the components making use of the domain-specific knowledge. In tissues stained with the routinely used hematoxylin-and-eosin technique, the components appear in three main colors, namely, white, pink, and purple, and they usually form round structures.



Type \ Distance	1	2	3	4	5	6	7	8	...
White-white	1	3	1	1	6	2	0	0	0
White-pink	2	2	0	4	0	0	0	0	0
White-purple	0	0	4	2	2	2	2	0	0
Pink-pink	0	0	0	0	0	0	0	0	0
Pink-purple	0	0	0	0	0	0	0	0	0
Purple-purple	0	0	0	0	0	0	0	0	0

Fig. 2. Illustration of object cooccurrence matrix definition for a selected white object shown as a thick-bordered cyan circle. The window $\mathcal{W}$ is located at the center object but the steps illustrated here are repeated for every object in $\mathcal{W}$. Here, we select a side object for illustration since we want to show the definition for larger distances (for this side object, the maximum distance is 7, which would be smaller if, for example, the center object was selected).

are clustered into three by the *k*-means algorithm and a set of nonoverlapping circles are iteratively located on the pixels of each cluster using the circle-fit algorithm [15]. Note that in the *k*-means algorithm, the principal component of data is divided into *k* equal intervals and an initial center is computed as the average of data in each interval [19].

At the end, an image I is decomposed into a set of objects, $O(I) = \{o_i\}$, each of which is represented by its coordinates (x_i, y_i) and its type $t_i \in \{white, pink, purple.\}$ Next, each object o_i is characterized by the spatial distribution of objects within its specified neighborhood. For that, a window $\mathcal{W}$ is located at the center of this object and an object cooccurrence matrix C is defined over this window, considering the cooccurrence type between a pair of every object within the window. In this matrix, an entry $C(t_{jk}, d)$ keeps the number of times objects of type t_j cooccur with objects of type t_k at a given distance d. In this paper, we construct a Delaunay triangulation on all objects and make use of breadth first traversal on this triangulation to calculate the distance between the objects.

Particularly, to define the object cooccurrence matrix of a window, we consider each object within this window, calculate the distance from this to every other object within the same window using breadth first traversal, and update the matrix accordingly. Fig. 2 illustrates the matrix definition for a selected white object that is shown as a thick-bordered cyan circle; here, we use cyan circles to represent the white objects. In this figure, the distance from this object to the others, which is computed using breadth first traversal, is indicated next to the corresponding object. The matrix for the selected object is calculated using these distances and the object types. For instance, when $d = 4$, we can reach one white (cyan), four pink, and two purple objects from the selected white object. Thus, $C(white - white, 4) = 1$, $C(white - pink, 4) = 4$, and $C(white - purple, 4) = 2$. To obtain the matrix of the entire window $\mathcal{W}$, this calculation is repeated for every object in $\mathcal{W}$ and their corresponding matrices are accumulated.

After its calculation, we extract 24 features from the object cooccurrence matrix. For that, we define four features from each cooccurrence type by accumulating the cooccurrence values for different distances. For cooccurrence type t_{jk}, these features are defined as follows, with M_d being the maximum distance from one object to another within a window $\mathcal{W}$

$$\begin{aligned}
\Phi_1 &= C(t_{jk}, 1) \\
\Phi_2 &= C(t_{jk}, 2) + C(t_{jk}, 3) \\
\Phi_3 &= C(t_{jk}, 4) + C(t_{jk}, 5) \\
\Phi_4 &= \sum_{d=6}^{M_d} C(t_{jk}, d).
\end{aligned}$$

The motivation behind defining the object cooccurrence features is that the relative spatial distribution of the tissue components differs in normal and cancerous regions. For example, in a normal colon tissue, there are single-layer epithelial cells around a lumen. In our domain, cell nuclei are represented as purple objects and lumina as white objects. As they are rich in mucin, the cytoplasms of the epithelial cells appear in almost white and are represented as white objects too. Thus, white objects cooccur in a close proximity and are surrounded with a single layer of purple objects. In low-grade cancer, single layers of epithelial cells typically turn into multiple layers. They become poor in mucin, which makes their cytoplasms appear in pink. Thus, purple and pink objects are found close to each other. In high-grade cancer, the organization among the tissue components is degenerated. The components, and their corresponding objects, seem to randomly distribute all over a tissue. This makes the cooccurrences of any two object types similar to each other.

The definition of an object cooccurrence matrix is similar to that of a gray-level cooccurrence matrix, which is defined over gray-level pixels to quantify their distribution with respect to a distance and a direction [20]. In contrast with this previous definition, we define our cooccurrence matrices over tissue objects and make use of graph traversals to measure the distance between the objects for quantifying the distribution of tissue components. Since the domain of tissue objects is expected to be less sensitive to small variations and errors at the pixel level, this new definition of the cooccurrence matrix becomes more effective in tissue image representation.

B. Multilevel Segmentation

The previous step transforms an image into the object domain, in which objects $O = \{o_i\}$ are characterized by extracting their object cooccurrence features $\Phi = \{\Phi(o_i)\}$. Our algorithm achieves segmentation on the object domain as well, and transforms the segmentation results back to the pixel domain. In this algorithm, we propose to obtain multiple object segmentations and combine them in an ensemble function. The main steps of

Algorithm 1 SEGMENTATION

Input: object set O, feature set Φ, number of segmented objects K, number of vertices in the coarsest graph K', sampling ratio $ratio$, number of segmentations M

Output: segmented objects S

```
$\Psi = \emptyset$
for $i = 1 \rightarrow M$ do
    $G^0 \leftarrow$ CONSTRUCTSAMPLEGRAPH$(O, \Phi, ratio)$
    $\Pi'_i \leftarrow$MULTILEVELGRAPHPARTITIONING$(G^0, K')$
    $\Pi_i \leftarrow$ INDUCEPARTITION$(\Pi'_i, V^0, O, \Phi)$
    $\Psi = \Psi \cup \{\Pi_i\}$
end for
$S \leftarrow$ SEGMENTATIONENSEMBLE$(\Psi, O, K)$
```

the algorithm are provided in Algorithm 1. For the pseudocodes of the subroutines called by this algorithm, the reader is referred to the technical report given in [21].

This algorithm considers the object segmentation as a graph partitioning problem. It first constructs a graph considering the objects as vertices and defining edges between the vertices by Delaunay triangulation. The weight $w(u, v)$ is then assigned to the edge $e(u, v)$ using (1), according to the similarity between the objects corresponding to the vertices u and v. Here, $dist(\Phi(u), \Phi(v))$ is the Euclidean distance between the object cooccurrence features of the vertices u and v

$$w(u, v) = 1 - \frac{dist(\Phi(u), \Phi(v))}{\max_{ij} dist(\Phi(i), \Phi(j))}. \tag{1}$$

In this study, instead of constructing a graph on the entire set of the objects, we select a random subset of them and construct the graph on these selected objects. By doing that, we can obtain a different graph every time we run the algorithm. This helps increase the diversity of individual segmentations.

Let $G = (V, F, E, W)$ be a sample graph constructed on the selected object set V where F is the feature set for the selected objects (vertices) and E and W are the sets of graph edges and their weights, respectively. The aim is to partition the vertices V into N disjoint parts (segments)[2], $V = \{V_1, \ldots, V_N \mid V_i \cap V_j = \emptyset \text{ for } i \neq j\}$, such that the sum of weights of the edges between the vertices in different sets is minimized. In this study, we partition the input graph using a multilevel scheme, on which we make modifications to obtain high quality but at the same time diverse results.

The proposed multilevel scheme is composed of two main phases: coarsening and uncoarsening. This scheme coarsens the original graph by merging its vertices level-by-level until K' segments (vertices) remain and uncoarsens them by refining the segments at each level. At the end, it outputs a segmentation vector Π containing the segment each vertex belongs to. As opposed to the traditional ones, this multilevel scheme omits an initial partitioning phase, which follows the coarsening, in the sake of obtaining more diverse results.

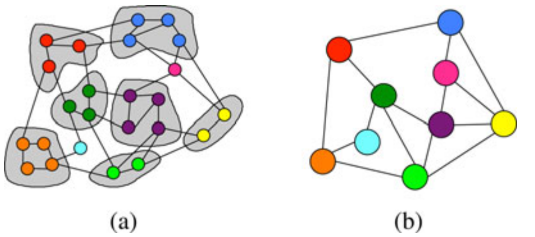


Fig. 3. Illustration of coarsening a graph at level t. (a) Graph G^t and the clusters of its vertices. (b) Graph G^{t+1} obtained by coarsening G^t.

1) Coarsening Phase: It coarsens the original input graph G^0 into a sequence of smaller graphs, $G = < G^1, \ldots, G^T >$, such that the graphs have less vertices at a successive level, $|V^0| > |V^1| > \cdots > |V^T|$. This coarsening is achieved by coalescing disjoint vertex subsets of G^t into multinodes such that each multinode of G^t forms a single supervertex of G^{t+1}, for $t = 0, 1, \ldots, T - 1$. This level-wise coarsening continues until K' vertices remain in the coarsest (smallest) graph.

In this study, we use a randomized agglomerative clustering for vertex coalescing. In this process, each vertex is assumed to constitute a singleton cluster at the beginning of each coarsening level. Then, vertices are visited in a random order. If a vertex has already been clustered, it is not considered for being a source of a new clustering. However, an unclustered vertex can choose to join a singleton or a multinode cluster. All neighbor vertices of an unclustered vertex are considered for selection and a neighbor singleton or a neighbor multinode vertex that has the maximum feature similarity with the source vertex is selected for coalescing. During this process, the feature vector of a multinode cluster is incrementally computed by averaging those of its constituent vertices.

At the end of the clustering process in a level, multiple edges between two clusters are collapsed into a single edge between the two respective supervertices of the coarser level. The weight of an edge between two supervertices is calculated from scratch considering the feature similarity of the respective supervertices. Fig. 3(a) shows a sample coarsening of a graph with 24 vertices into nine clusters and Fig. 3(b) shows the coarser level graph (with nine vertices) induced by this clustering.

2) Uncoarsening Phase: At each level t of the uncoarsening phase, for $t = T, T-1, \ldots, 1$, the segmentation found on G^t is projected back to an initial segmentation on the finer level graph G^{t-1}. In this projection, the constituent vertices of each multinode in G^{t-1} are assigned to the segment of the respective supervertex in G^t. Then, this projected initial segmentation is refined through iterative vertex moves between segments. The motivation behind this multilevel refinement is as follows. The segmentation found at level t may be a local optimal segmentation of G^t. However, as the finer graph G^{t-1} has more degrees of freedom, the projected segmentation may not be locally optimal with respect to G^{t-1}, and hence, G^{t-1} can be used to further improve the segmentation.

The proposed approach does not use an additional initial partitioning phase after coarsening, as most of the traditional

[2] In the forthcoming multilevel graph partitioning discussion, to keep the presentation simpler and easier to read, we will use *segment* and *segmentation* instead of *part* and *partition*, respectively.

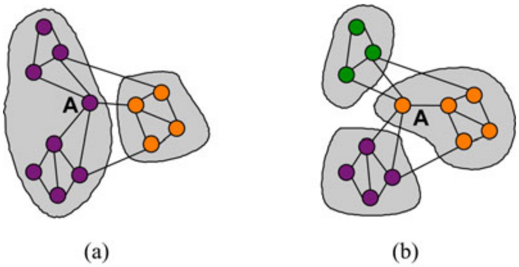


Fig. 4. Illustration of a vertex move: (a) before and (b) after. After moving vertex A from its source segment to the other, the vertices of its source segment get disconnected and new segments are created.

multilevel partitioning algorithms do, but uses the vertex set of the coarsest graph to define the initial partition/segmentation.

We will briefly describe the proposed iterative improvement heuristic used to refine the initial segmentation at each level of uncoarsening. The proposed scheme consists of a number of passes over vertices, in each of which a vertex v is considered to be moved from its current (source) segment to one of the destination segments that contain at least one neighbor vertex of v. These vertex moves try reducing the similarity between different segments, which in turn corresponds to maximizing the similarity within the segments.

This scheme considers only boundary vertices, which have at least one neighbor on a segment different than their own, for movement. At each pass, a FIFO queue B is maintained to keep the boundary vertices. Each vertex v in B is considered for movement in turn. The similarity of the vertex v to the source and destination segments is computed as the negative Euclidean distance $dist(\Phi(v), \Phi(S))$ between the feature vectors of the vertex v and the segment S. Then, the vertex is moved to the segment, to which it is the most similar. Note that if the vertex v is the most similar to its source segment, it is not moved at all. Since a move can cause some nonboundary vertices to become boundary or vice versa, B is updated after each vertex move. Moreover, a vertex move may cause some other vertices to lose their connectivity with the remaining vertices in the source segment, even though these vertices are not moved. Such vertices are identified at the end of each pass and new segments are created from them (see Fig. 4). The vertex moves are continued until either B is empty or after $|B_{\text{init}}|$ trials, where $|B_{\text{init}}|$ denotes the number of boundary vertices at the beginning of the pass. Each vertex is locked after its move to avoid $thrashing$, where the same vertices are moved repeatedly from one segment to another in the same pass.

At the end of uncoarsening, we obtain a segmentation only on the selected vertices (objects) since our algorithm takes a random subset of the objects, but not all of them, every time it runs. Then, we use this segmentation to induce a complete segmentation on the entire object set: we first determine the selected neighbor objects of each unselected object o_i using Delaunay triangulation and then assign o_i to the segment that contains the most similar neighbor object of o_i.

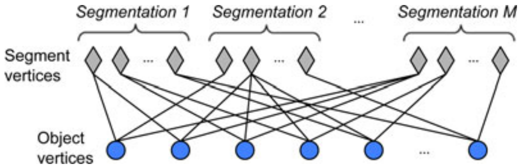


Fig. 5. Bipartite graph constructed on the object and segment vertices.

C. Ensemble Framework

In the last step, we combine multiple segmentation results obtained by the multilevel partitioning algorithm. To this end, we use the cluster ensemble framework given in [24]. In this framework, multiple segmentation results are represented as an unweighted bipartite graph $G_b = (U \cup V, E)$, where the set of objects and the set of segments obtained in the segmentations constitute the two vertex sets of the bipartite graph. That is, an object o_i is represented as an object vertex $v_i \in V$, and kth segment of the jth segmentation is represented as a segment vertex $u_j^k \in U$. An edge is defined between an object and a segment vertex if the object vertex is assigned to this segment in the corresponding segmentation. Fig. 5 depicts an example bipartite graph construction.

After constructing the bipartite graph G_b, we partition G_b using the normalized-cut criterion [22] through a recursive bipartitioning framework. In this bipartite graph partitioning, the unnormalized cutsize is equal to the number of the cut edges where an edge is said to be cut if it connects two vertices in different parts. The normalization is introduced by dividing the cutsize to the total number of edges that connect the vertices of each part of a partition. Note that normalized cuts can produce isolated objects. Although not frequent, such cases are handled by selecting the largest component of each partition and merging the remaining small components (isolated objects) with their most similar adjacent partitions.

The final vertex partition produced by the bipartitioning framework is decoded as inducing a final object segmentation. In this segmentation, the object vertices that are assigned to the same segment of the vertex partition constitute the objects of a segmented region. In our ensemble framework, minimizing the unnormalized cutsize corresponds to minimizing the number of object-to-segment assignments that are not respected in the final segmentation, with regard to the individual segmentation results in the ensemble. Since the segmentations are obtained on objects, but not on image pixels, we map the pixels to the segmented objects, by assigning each pixel to its closest object, and obtain the segmented image regions.

III. Experiments

A. Dataset

In our experiments, we use 200 microscopic images of colon biopsy samples stained with hematoxylin-and-eosin. These biopsies are randomly selected from the Pathology Department Archives of Hacettepe Medical School, Ankara, Turkey. The images are taken using a Nikon Coolscope Digital Microscope; the microscope objective lens is 5× and the image resolution is 1920 × 2560. The images are so selected as to contain both normal regions and adenocarcinomatous (cancerous) regions of different grades. The selected images are randomly divided into training and test sets. The training set includes 50 tissue images and is used to estimate the parameters of the algorithms. The test set includes the remaining 150 tissue images[3].

B. Evaluation

We quantitatively evaluate all the algorithms by comparing their results with the gold standard provided by our medical collaborator. As the algorithms are unsupervised, we obtain the quantitative results as follows. First, we assign each segmented region S to the label of its most overlapping region R in the gold standard. Then, we consider the overlapping pixels of S and R as either true positive (TP) or true negative and the nonoverlapping pixels of S as either false positive (FP) or false negative, according to the label of R (e.g., if R is cancerous, the overlapping pixels are considered as TP and the nonoverlapping pixels as FP). Subsequently, we compute the $F\text{-}score$ as the evaluation criterion for comparison and parameter selection. We select the parameter set that maximizes the average $F\text{-}score$ on training images. For our algorithm, the parameter sets considered in parameter selection are all combinations of the following values: window size $= \{32, 64, 96, 128\}$, coarsest graph size $= \{2, 3, 4, 5, 10, \ldots, 50\}$, sampling ratio $= \{0.01, 0.02, 0.05, 0.10, \ldots, 0.30, 0.50, 0.70, 0.90, 1.00\}$, and number of segmentations $= \{50, 100, \ldots, 300, 400, 500\}$. The parameter sets of the other algorithms can be found in [15]. Note that we select the parameters separately for each number K of the segmented regions.

We obtain the gold standards considering colon adenocarcinoma, which accounts for 90–95% of all colorectal cancers. As this cancer type originates from epithelial cells, which form colon glands, it causes deformations in the glandular architecture. Thus, to obtain its gold standard, an image is segmented into regions containing normal and cancerous glands. In addition to these, there may exist inflammatory and stromal regions that do not contain any glands (as shown with green shades in Fig. 6). In our experiments, we do not consider such regions in F-score calculations.

C. Results

The multilevel segmentation (*MLSeg*) algorithm segments an image into K regions. This is opposed to the case, in which the number of segmented regions is dynamically selected by an algorithm; the ones that we use in our comparisons are the examples of such algorithms. The dynamic selection of this number usually depends on an external parameter (e.g., a merge threshold). However, it is hard to select a common value of this parameter such that it gives good results for multiple images in terms of both the accuracy and the number of segmented regions. The values that give accurate results for multiple images typically lead to oversegmentations. Thus, in order to prevent such oversegmentations, we externally set the number K of regions in our algorithm.

In the experiments, we select K as 2 and 3 since the tissue images we use have 2 or 3 regions to be segmented. Table I reports the average test results for different K values. It shows that the *MLSeg* algorithm can achieve accurate results, $> 90\%$ accuracy and $> 90\%$ F-score on the average, even when $K =$2 and 3. Larger values of K slightly improve the results.

D. Comparisons

We compare the proposed *MLSeg* algorithm with two sets of algorithms. The first set includes our earlier studies, the *graphRLM* [15] and *objectSEG* [14] algorithms, that are also implemented for histopathological image segmentation. These studies quantify an image defining different sets of features on the tissue objects and achieve segmentation using a region growing algorithm. We make these comparisons to understand the effectiveness of the newly introduced object cooccurrence features as well as our multilevel segmentation algorithm.

The second set includes two algorithms: the graph-based segmentation (*GBS*) [23] and J value segmentation (*JSEG*) [25] algorithms that are not specifically implemented for histopathological images but are known as effective segmentation algorithms for images in general. We make these comparisons to understand the importance of using domain-specific knowledge in segmentation.

None of these algorithms take the number of segmented regions externally; instead, they select this number dynamically for each image according to their parameters. Each algorithm selects the parameter set that gives the highest F-score on the training images. This selection maximizes this measure at the expense of obtaining more segmented regions. Thus, to prevent oversegmentations, we enforce the algorithms not to consider the parameter sets that give more regions than an upperbound N. In our previous study [15], we selected $N =$ 5 and 10. Table II provides the test results for these upperbounds. The results show that the selection of N greatly affects the segmentation quality. When it is set to a smaller value to obtain less oversegmented results, the performance significantly drops. When it is set to a larger value, such as $N =$ 10, the performance increases but the results tend to be oversegmented. Comparing the results of the *MLSeg* algorithm with those given in this table, we observe that the *MLSeg* algorithm achieves higher quality even when $K =$ 2 or 3.

To make a fairer comparison, we also modify the algorithms so that they segment an image into exactly K regions. For that, we enforce the *graphRLM* algorithm to select the largest K initial seeds in its seed determination step. We enforce the other algorithms to dynamically select their merge parameters for each image so that their merging step merges the regions to each other until K regions are left. Table III reports the test results when $K =$ 2 and 3. Here, we do not report the results of the *JSEG* algorithm since it gives very inaccurate results; some of its results are depicted in the last column in Fig. 6. As possible future work, one could consider to modify the other steps of the

[3] In this study, we increase the number of test images from 100 to 150. Therefore, the results of the algorithms that we use in our comparisons are different from those reported in our previous work [15].

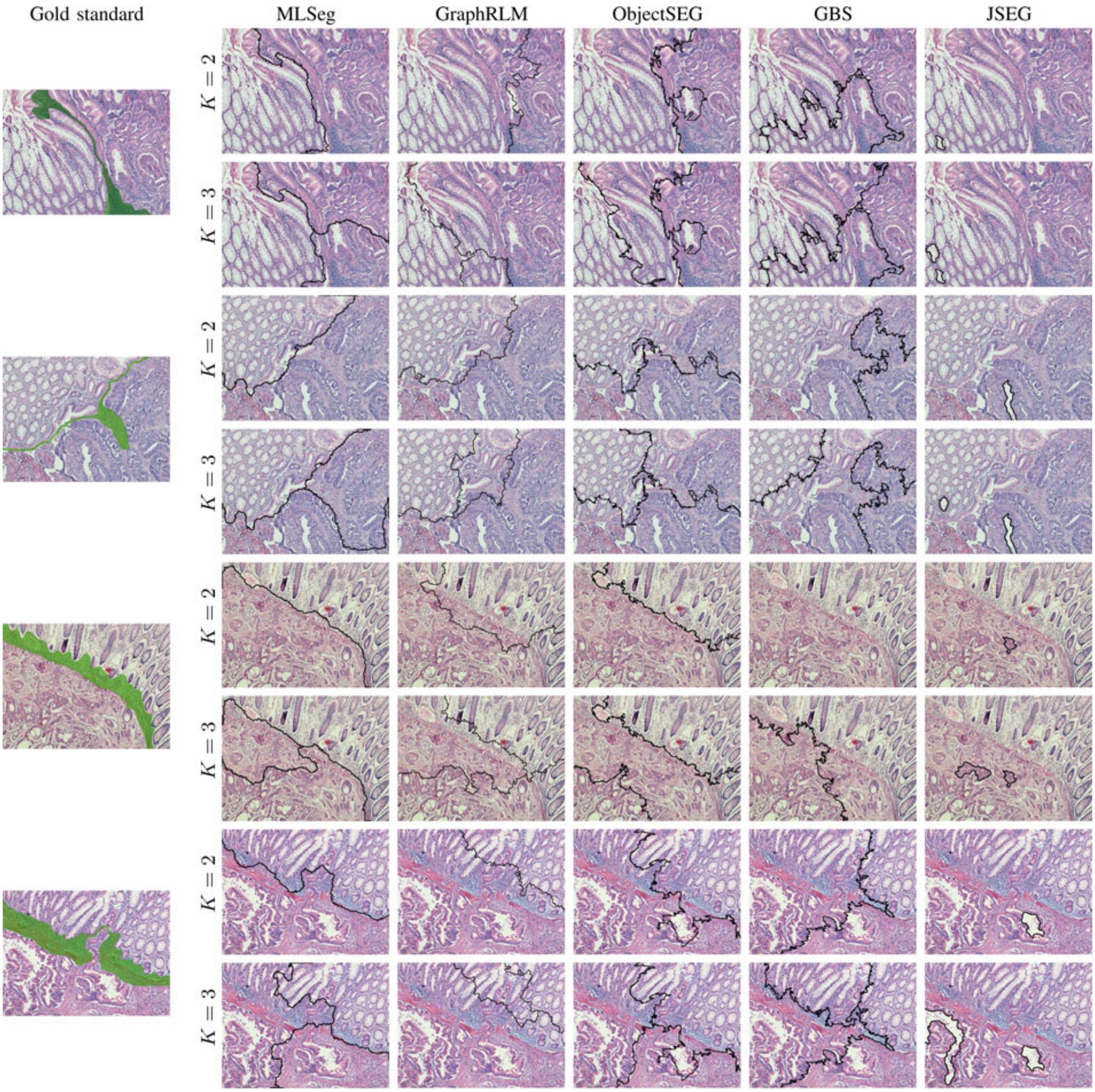


Fig. 6. Visual results obtained on example tissue images when the number K of segmented regions is set to 2 and 3.

TABLE I
AVERAGE TEST SET RESULTS OBTAINED BY THE PROPOSED *MLSeg* ALGORITHM AND THEIR STANDARD DEVIATIONS

K	Accuracy	Sensitivity	Specificity	F-score
2	92.0 ± 9.0	92.6 ± 15.9	87.8 ± 23.8	91.7 ± 14.0
3	92.9 ± 7.6	94.2 ± 12.4	90.3 ± 15.8	92.9 ± 10.7
4	94.2 ± 5.5	94.6 ± 8.6	92.9 ± 9.0	94.4 ± 6.0
5	94.9 ± 5.5	95.8 ± 7.1	92.9 ± 9.3	95.2 ± 5.6

JSEG algorithm to obtain exactly K segmented regions with better quality. Table III shows that the *MLSeg* algorithm greatly improves the results of the others. The *t*-test indicates that this improvement is statistically significant with $\alpha = 0.05$. Fig. 6 gives visual results on some example images.

E. Discussion

The *MLSeg* algorithm segments an image into a selected number K of regions. Of course, the selection of K is closely related to the application of interest. For applications in which such a value cannot be defined, dynamic selection should be incorporated. In the experiments, we also implement an extended version of our algorithm that selects K dynamically. To this end, starting with $K = 2$, we compute an invariant criterion for the segmentation result and find the first K for which the criterion falls below a certain threshold. We use the invariant criterion $J = |S_W|/|S_B|$, where S_W and S_B correspond to the within-segmentation (cluster) and between-segmentation scatter matrices [26]. Selecting a threshold value as 0.2, we obtain $94.1 \pm 6.4\%$ accuracy and $94.1 \pm 9.8\%$ F-score, on the average. The average number of regions is 3.3 ± 1.1. These results indicate the potential use of the algorithm for variable K values.

TABLE II
AVERAGE TEST SET RESULTS OBTAINED BY THE OTHER ALGORITHMS AND THEIR STANDARD DEVIATIONS

		Accuracy	Sensitivity	Specificity	F-score	Region no
$N \leq 5$	GraphRLM	84.8 ± 14.4	85.8 ± 26.2	76.2 ± 35.7	81.5 ± 26.6	2.8 ± 1.1
	ObjectSEG	86.9 ± 11.5	90.4 ± 21.8	77.2 ± 28.5	78.6 ± 27.9	4.1 ± 1.3
	GBS	73.4 ± 8.9	64.5 ± 33.8	72.2 ± 30.3	58.5 ± 37.7	3.7 ± 1.2
	JSEG	69.4 ± 12.1	62.7 ± 45.5	62.2 ± 39.8	46.2 ± 38.4	2.9 ± 1.3
$N \leq 10$	GraphRLM	91.5 ± 9.8	92.3 ± 15.6	87.3 ± 24.2	91.3 ± 13.8	5.6 ± 1.6
	ObjectSEG	89.9 ± 11.4	89.3 ± 23.5	85.8 ± 22.7	87.8 ± 19.1	5.8 ± 2.0
	GBS	74.0 ± 9.5	64.7 ± 31.9	74.5 ± 28.7	61.9 ± 32.0	5.0 ± 1.7
	JSEG	89.6 ± 7.2	89.1 ± 16.3	87.1 ± 18.2	87.6 ± 12.5	7.9 ± 2.7

The parameters of the algorithms are selected considering an upperbound N on the number of segmented regions.

TABLE III
AVERAGE TEST SET RESULTS OBTAINED BY THE *MLSeg*, *GraphRLM*, *ObjectSeg*, AND *GBS* ALGORITHMS AND THEIR STANDARD DEVIATIONS

		Accuracy	Sensitivity	Specificity	F-score
$K = 2$	MLSeg	92.0 ± 9.0	92.6 ± 15.9	87.8 ± 23.8	91.7 ± 14.0
	GraphRLM	83.5 ± 13.1	81.1 ± 27.1	79.1 ± 32.6	80.8 ± 22.8
	ObjectSEG	82.9 ± 11.7	86.9 ± 22.9	69.8 ± 36.7	82.7 ± 18.9
	GBS	75.5 ± 11.0	71.5 ± 36.4	71.2 ± 28.8	56.0 ± 32.7
$K = 3$	MLSeg	92.9 ± 7.6	94.2 ± 12.4	90.3 ± 15.8	92.9 ± 10.7
	GraphRLM	87.6 ± 10.6	90.4 ± 15.7	80.1 ± 26.9	88.1 ± 12.9
	ObjectSEG	88.4 ± 8.3	90.6 ± 15.9	81.5 ± 23.7	88.5 ± 11.9
	GBS	76.3 ± 10.2	70.2 ± 32.9	74.5 ± 28.4	58.0 ± 33.3

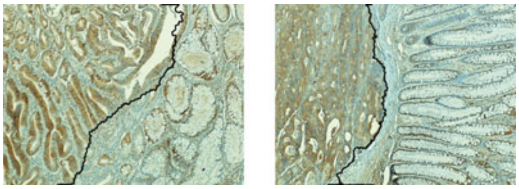


Fig. 7. Visual results obtained on the example images of tissues stained with immunohistochemistry.

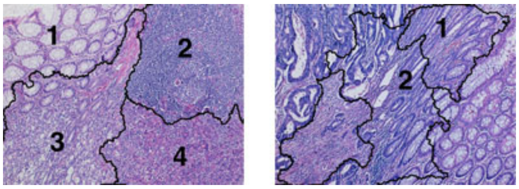


Fig. 8. Visual results obtained on the example images of tissues containing different types of regions.

TABLE IV
AVERAGE TEST SET RESULTS OBTAINED BY THE *MLSeg* ALGORITHM THAT USES AN ALTERNATIVE COOCCURRENCE MATRIX CALCULATION

K	Accuracy	Sensitivity	Specificity	F-score
2	88.9 ± 11.0	89.6 ± 18.0	82.4 ± 30.4	88.4 ± 17.2
3	91.6 ± 7.7	91.8 ± 13.8	87.7 ± 20.5	91.3 ± 12.0
4	92.9 ± 6.4	94.2 ± 7.3	90.1 ± 13.2	93.4 ± 6.7
5	93.9 ± 5.4	95.3 ± 7.0	90.7 ± 9.8	94.3 ± 5.8

The *MLSeg* algorithm makes use of Delaunay triangulation to calculate a cooccurrence matrix. It is also possible to use different approaches for this calculation. For instance, for an object, one may compute the Euclidean distance from this object to every other object in a window $\mathcal{W}$ and create a histogram by binning. The cooccurrence matrix is then calculated by accumulating the histograms of the objects that fall in $\mathcal{W}$ based on their object types and the same object cooccurrence features are extracted from this matrix. Table IV reports the test set results obtained by this approach. These results are slightly less than those obtained by the proposed cooccurrence matrix calculation. The results indicate the effectiveness of the use of Delaunay triangulation in extracting descriptive features.

Although the experiments are conducted on the images of hematoxylin-and-eosin stained tissues to locate normal and colon cancerous regions, the proposed algorithm has a potential to be used on different types of histology images as well as to locate regions of different characteristics. Fig. 7 shows the results when the algorithm is applied to two example images of tissues stained with immunohistochemistry. In these preliminary results, we observe that normal and cancerous regions are successfully segmented.

Moreover, we run our algorithm to locate different types of regions. As an example, the first column in Fig. 8 shows an image with four different regions: normal region (marked as 1), inflammatory region (marked as 2), and cancerous regions of different grades (marked as 3 and 4). The *MLSeg* algorithm is successful to segment these regions. As another example, the second column in Fig. 8 shows an image that contains a region of dysplastic glands (marked as 1 and 2). For this image, we observe that such regions can only be roughly segmented. To segment dysplastic glands accurately, one may consider to combine pixel-level textures to the proposed features. This would be an interesting future research direction.

We also conduct experiments to validate the robustness and stability of the segmentations found by our algorithm. To do so, for each image, we run our overall algorithm 30 times and calculate the standard deviation of the F-scores over these runs. We use the average of the standard deviations over the test set

TABLE V

AVERAGE TEST RESULTS OF THE *MLSeg* ALGORITHM, A MULTILEVEL SEGMENTATION ALGORITHM WITH *GraphRLM* FEATURES, A REGION GROWING ALGORITHM WITH OBJECT COOCCURRENCE FEATURES, AND A MULTILEVEL SEGMENTATION ALGORITHM THAT COMBINES THREE TYPES OF FEATURES

	Segmentation algorithm	Feature type	Accuracy	Sensitivity	Specificity	F-score
$K = 2$	Multilevel	Object cooccurrence	92.0 ± 9.0	92.6 ± 15.9	87.8 ± 23.8	91.7 ± 14.0
	Multilevel	GraphRLM	90.0 ± 9.8	92.1 ± 15.5	82.8 ± 26.5	90.1 ± 14.0
	Region growing	Object cooccurrence	74.6 ± 15.1	78.0 ± 35.5	56.5 ± 44.8	70.8 ± 29.2
	Multilevel	Object cooc., ObjectSeg, GraphRLM	92.6 ± 9.1	93.6 ± 15.3	87.5 ± 26.2	92.3 ± 13.7
$K = 3$	Multilevel	Object cooccurrence	92.9 ± 7.6	94.2 ± 12.4	90.3 ± 15.8	92.9 ± 10.7
	Multilevel	GraphRLM	91.8 ± 6.8	93.8 ± 9.9	86.3 ± 18.4	92.4 ± 7.8
	Region growing	Object cooccurrence	81.9 ± 14.1	82.8 ± 29.9	72.3 ± 35.4	79.0 ± 25.2
	Multilevel	Object cooc., ObjectSeg, GraphRLM	93.6 ± 7.7	93.3 ± 13.6	92.3 ± 14.6	93.2 ± 11.1

to examine the robustness and stability of the segmentations. The average F-scores are found to be 1.58%, 2.00%, 1.74%, and 1.26% for $K = 2$, 3, 4, and 5, respectively. These results indicate that the segmentations slightly change from one run to another. The visual results also support this finding. The reader is referred to the technical report [21] for the visual results of an example image obtained over these 30 runs.

IV. CONCLUSION

This paper presents a new algorithm for unsupervised segmentation of histopathological images. This algorithm defines a set of new high-level texture descriptors to represent the prior knowledge in a tissue and uses them in an effective multilevel segmentation algorithm. The experiments on 200 colon tissue images show that the proposed algorithm is effective to obtain higher quality results compared to the other algorithms.

The proposed *MLSeg* algorithm has two main contributions: the introduction of object cooccurrence features and the use of multilevel segmentation. We implement two more algorithms to understand the effects of these two contributions separately. In the first algorithm, we extract the *graphRLM* features [15] to quantify the objects and achieve segmentation by the same multilevel segmentation. In the second one, we use the same cooccurrence features to quantify the objects but achieve segmentation by region growing used in [15]. The results of these two algorithms are compared to the *MLSeg* algorithm in Table V when $K = 2$ and 3. It shows that it is not possible to achieve the highest results with one of these two contributions. Thus, both of them are crucial in the proposed algorithm.

Obviously, it is possible to obtain better results by integrating different feature types into the multilevel segmentation algorithm. To explore this, we combine three types of features: the object cooccurrence features proposed by our study, the objectSEG features [14], and the graphRLM features [15]. The results in Table V show that one may slightly increase the performance by combining different features.

The *MLSeg* algorithm can be used as a part of a two-phase algorithm for whole slide image segmentation. In the first phase, empty areas of a whole slide can be separated from histological sections using a simple algorithm. Then, in the second phase, the histological sections are segmented into their homogeneous regions using the *MLSeg* algorithm; a typical slide contains 75–100 5× images of histological sections. Our implementation uses Java for feature extraction and MATLAB for multilevel segmentation. The computational time for a single image is approximately 7–8 min using a computer with a Intel Xeon 2.27 GHz processor and 12 GB of RAM. However, it is possible to obtain significant speedups by implementing the algorithm with C or C++ and using an optimized compiler. This would be considered as the future work.

REFERENCES

[1] P.-W. Huang and C.-H. Lee, "Automatic classification for pathological prostate images based on fractal analysis," *IEEE Trans. Med. Imaging*, vol. 28, no. 7, pp. 1037–1050, Jul. 2009.

[2] 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.

[3] S. Doyle, M. Feldman, J. Tomaszewski, and A. Madabhushi, "A boosted Bayesian multi-resolution classifier for prostate cancer detection from digitized needle biopsies," *IEEE Trans. Biomed. Eng.*, 2011, to be published, doi: 10.1109/TBME.2010.2053540.

[4] L. Zheng, A. W. Wetzel, J. Gilbertson, and M. J. Becich, "Design and analysis of a content-based pathology image retrieval system," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 249–255, Dec. 2003.

[5] F. Yu and H. H. S. Ip, "Semantic content analysis and annotation of histological image," *Comput. Biol. Med.*, vol. 38, no. 6, pp. 635–649, Jun. 2008.

[6] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 841–852, Apr. 2010.

[7] 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, Feb. 2010.

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

[9] M. Mete, X. Xu, C.-H. Fan, and G. Shafirstein, "Automatic delineation of malignancy in histopathological head and neck slides," *BMC Bioinformat.*, vol. 8, no. Suppl 7, pp. S17–S29, Nov. 2007.

[10] 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, Feb. 2009.

[11] D. Romo, E. Romero, and F. Gonzales, "Learning regions of interest from low level maps in virtual microscopy," *Diagnostic Pathology*, vol. 6, no. Suppl 1, p. 522, Mar. 2011.

[12] K. Mosaliganti, F. Janoos, O. Irfanoglu, R. Ridgway, R. Machiraju, K. Huang, J. Saltz, G. Leone, and M. Ostrowski, "Tensor classification of N-point correlation function features for histology tissue segmentation," *Med. Image Anal.*, vol. 13, no. 1, pp. 156–166, Feb. 2009.

[13] N. Signolle, M. Revenu, B. Plancoulaine, and P. Herlin, "Wavelet-based multiscale texture segmentation: Application to stromal compartment characterization on virtual slides," *Signal Process.*, vol. 90, no. 8, pp. 2412–2422, Aug. 2010.

[14] 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, Jun. 2009.

[15] A. B. Tosun and C. Gunduz-Demir, "Graph run-length matrices for histopathological image segmentation," *IEEE Trans. Med. Imag.*, vol. 30, no. 3, pp. 721–732, Mar. 2011.

[16] S. C. Zhu and A. Yuille, “Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996.

[17] A. Strehl and J. Ghosh, “Cluster ensembles—A knowledge reuse framework for combining multiple partitions,” *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Jan. 2003.

[18] L. I. Kuncheva and C. J. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy,” *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, May 2003.

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

[20] R. M. Haralick, “Statistical and structural approaches to texture,” *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[21] A. C. Simsek, A. B. Tosun, C. Aykanat, C. Sokmensuer, and C. Gunduz-Demir. (2012). “Multilevel segmentation of histopathological images using cooccurrence of tissue objects: Supplementary material,” Dept. Computer Engineering, Bilkent University, Ankara, Turkey, Tech. Rep. BU-CE-1202 [Online] Available: at http://www.cs.bilkent.edu.tr/tech-reports/2012/BU-CE-1202.pdf

[22] 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.

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

[24] X. Z. Fern and C. E. Brodley, “Solving cluster ensemble problems by bipartite graph partitioning,” in *Proc 21th Int. Conf. Machine Learn.*, 2004, pp. 36–43.

[25] 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.

[26] O. R. Duda, E. P. Hart, and G. D. Stork, *Pattern Classification*. New York: Wiley Interscience, 2001.

Authors, photographs and biographies not available at the time of publication.