

Adaptive Perceptual Color-Texture Image Segmentation

Junqing Chen, *Member, IEEE*, Thrasyvoulos N. Pappas, *Senior Member, IEEE*, Aleksandra Mojsilović, *Member, IEEE*, and Bernice E. Rogowitz *Senior Member, IEEE*

Abstract— We propose a new approach for image segmentation that is based on low-level features for color and texture. It is aimed at segmentation of natural scenes, in which the color and texture of each segment does not typically exhibit uniform statistical characteristics. The proposed approach combines knowledge of human perception with an understanding of signal characteristics in order to segment natural scenes into perceptually/semantically uniform regions. The proposed approach is based on two types of spatially adaptive low-level features. The first describes the local color composition in terms of spatially adaptive dominant colors, and the second describes the spatial characteristics of the grayscale component of the texture. Together, they provide a simple and effective characterization of texture that the proposed algorithm uses to obtain robust and, at the same time, accurate and precise segmentations. The resulting segmentations convey semantic information that can be used for content-based retrieval. The performance of the proposed algorithms is demonstrated in the domain of photographic images, including low resolution, degraded, and compressed images.

Index Terms— Content-based image retrieval (CBIR), adaptive clustering algorithm, optimal color composition distance (OCCD), steerable filter decomposition, Gabor transform, local median energy, human visual system (HVS) models.

EDICS Category: Trans. IP 2-SEGM (Segmentation)

I. INTRODUCTION

THE RAPID accumulation of large collections of digital images has created the need for efficient and intelligent schemes for image retrieval. Since manual annotation of large image databases is both expensive and time consuming, it is desirable to base such schemes directly on image content. Indeed, the field of Content-Based Image Retrieval (CBIR) has made significant advances in recent years [1], [2]. One of the most important and challenging components of many CBIR systems is scene segmentation.

This paper considers the problem of segmentation of natural images based on color and texture. Although significant

Manuscript received August 7, 2003; revised August 9, 2004. This work was supported by the National Science Foundation (NSF) under Grant No.CCR-0209006. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michael Schmitt.

J. Chen is with Unilever Research, Trumbull, CT 06611 USA (email: junqing.chen@unilever.com)

T. N. Pappas is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: pappas@ece.northwestern.edu).

A. Mojsilović and B. E. Rogowitz are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA (e-mail: aleksand@us.ibm.com, rogowitz@us.ibm.com)

progress has been made in texture segmentation (e.g., [3–7]) and color segmentation (e.g., [8–11]) separately, the area of combined color and texture segmentation remains open and active. Some of the recent work includes JSEG [12], stochastic model-based approaches [13–15], watershed techniques [16], edge flow techniques [17], and normalized cuts [18].

Another challenging aspect of image segmentation is the extraction of perceptually relevant information. Since humans are the ultimate users of most CBIR systems, it is important to obtain segmentations that can be used to organize image contents semantically, according to categories that are meaningful to humans. This requires the extraction of low-level image features that can be correlated with high-level image semantics. This is a very challenging problem. However, rather than trying to obtain a complete and detailed description of every object in the scene, it may be sufficient to isolate certain regions of perceptual significance (such as “sky,” “water,” “mountains,” etc.) that can be used to correctly classify an image into a given category, such as “natural,” “man-made,” “outdoor,” etc. [19]. An important first step towards accomplishing this goal, is to develop low-level image features and segmentation techniques that are based on perceptual models and principles about the processing of color and texture information.

A significant effort has been devoted recently to understanding perceptual issues in image analysis. This includes perceptual grouping of image contents (e.g., [18], [20], and [21]), perceptual modeling of objects (e.g., [22–24]), perceptual modeling of isolated textures for analysis/synthesis [25], [26], and perceptually based texture classification [27]. However, there has been relatively little work in applying perceptual principles to complex scene segmentation (e.g., [28]), which motivates our work. We focus on a broad domain of photographic images: outdoor and indoor scenes, landscapes, cityscapes, plants, animals, people, objects, etc. A challenging aspect of our work is that we attempt to accomplish both feature extraction and segmentation with relatively low resolution (e.g., 200×200 or lower) and occasionally degraded or compressed images, just as humans do. This is especially important since low resolution images are most frequently used within WWW documents. In addition, the advantage of low resolution images is that access and processing time are significantly reduced.

A. Motivation and Justification for the Proposed Approach

There are two main goals in this work. The first is to develop segmentation algorithms for images of natural scenes, in which

color and texture typically do not exhibit uniform statistical characteristics. The second is to incorporate knowledge of human perception in the design of underlying feature extraction algorithms.

Segmentation of images of natural scenes is particularly difficult because, unlike artificial images that are composed of more or less pure textures, the texture properties are not well defined. The texture characteristics of perceptually distinct regions are not uniform due to effects of lighting, perspective, scale changes, etc. Fig. 1 shows two manually segmented images. Even though the water and the sky in both images are quite distinct segments, the color varies substantially within each segment. Similarly, the spatial characteristics of the city, forest, and mountain segments are also distinct but do not have well defined uniform characteristics. The human visual system (HVS) is very good at accounting for the various effects mentioned above in order to segment natural scenes into *perceptually/semantically* uniform regions. However, it is extremely difficult to automatically segment such images, and existing algorithms have been only partially successful. The key to addressing this problem is in combining perceptual models and principles of texture and color processing with an understanding of image characteristics.

Recently, there has been considerable progress in developing perceptual models for texture characterization in the areas of texture analysis/synthesis and texture classification. Several authors have presented models for texture analysis and synthesis using multiscale frequency decompositions [26], [29–34]. The most recent and complete results were presented by Portilla and Simoncelli [26], who proposed a statistical model for texture images that is consistent with human perception. Their model is quite elaborate and captures a very wide class of textures. Similarly, there has been considerable activity in texture classification [3–5], [27]. The segmentation problem is quite different, however. Most of the work in texture analysis/synthesis and texture classification has been focused on isolated samples of well-defined textures with relatively uniform characteristics (*e.g.*, wavelet coefficients within each subband follow a certain distribution [35]). In addition, the methods for texture analysis, classification, and synthesis are designed to operate in high-resolution images (*e.g.*, 256×256 or 512×512 pixels), which allows for the precise estimation of a relatively large number of texture parameters (*e.g.*, several hundred in [26]). In contrast, we want to segment textures in thumbnail images, which may contain several textures with spatially varying characteristics. Thus, by necessity, our texture models have to be far simpler so their parameters can be robustly estimated from a few sample points. Note that, as we discussed above, for segmentation it is not necessary to characterize every possible texture, only some key texture features that can help discriminate between perceptually important regions.

B. Outline of Proposed Approach

We present an image segmentation algorithm that is based on spatially adaptive texture features. As illustrated in Fig. 2, we develop two types of features, one describes the local

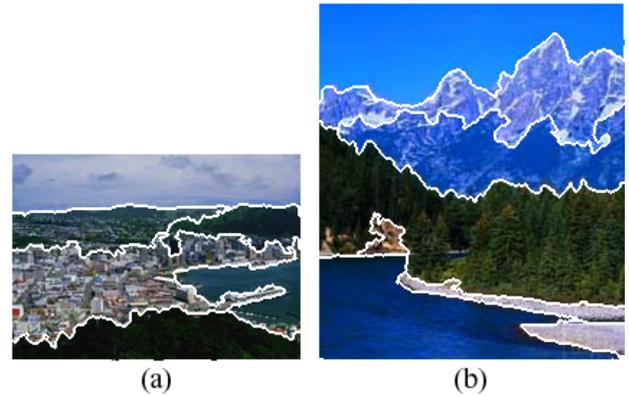


Fig. 1. Human Segmentations (images shown in color).

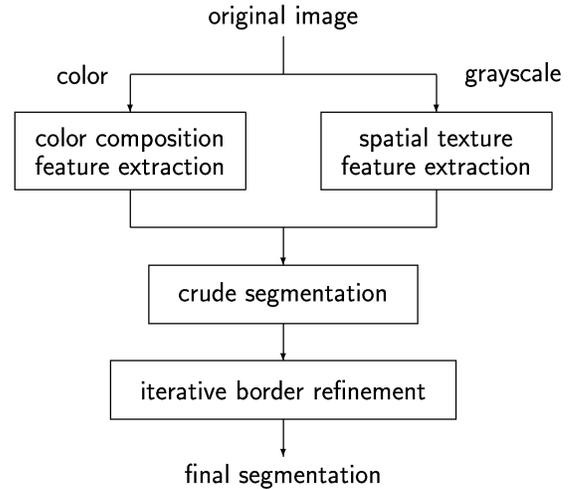


Fig. 2. Schematic of Proposed Segmentation Algorithm.

color composition, and the other the spatial characteristics of the grayscale component of the texture. These features are first developed independently, and then combined to obtain an overall segmentation.

The initial motivation for the proposed approach came from the adaptive clustering algorithm (ACA) proposed by Pappas [8]. ACA has been quite successful for segmenting images with regions of slowly varying intensity but oversegments images with texture. Thus, a new algorithm is necessary that can extract color textures as uniform regions and provide an overall strategy for segmenting natural images that contain both textured and smooth areas. The proposed approach uses ACA as a building block. It separates the image into smooth and textured areas, and combines the color composition and spatial texture features to consolidate textured areas into regions.

The color composition features consist of the dominant colors and associated percentages in the vicinity of each pixel. They are based on the estimation of *spatially adaptive dominant colors*. This is an important new idea, which on one hand, reflects the fact that the HVS cannot simultaneously perceive a large number of colors, and on the other, the fact that region colors are spatially varying. Note that there have been previous approaches based on the concept of extracting

the dominant colors in the image [27], [36], [37], however, none of them addresses the issue of spatial variations, which is one of the most common characteristics for images of natural scenes. Spatially adaptive dominant colors can be obtained using the ACA [8]. As we will see in Section II, the local intensity functions of the ACA can be used as spatially adaptive dominant colors. Finally, we propose a modified Optimal Color Composition Distance (OCCD) metric to determine the perceptual similarity of two color composition feature vectors [38].

The spatial texture features describe the spatial characteristics of the grayscale component of the texture, and are based on a multiscale frequency decomposition that offers efficient and flexible approximation of early processing in the HVS. We use the local energy of the subband coefficients as a simple but effective characterization of spatial texture. An important novelty of the proposed approach is that a median filter operation is used to distinguish the energy due to region boundaries from the energy of the textures themselves. We also show that, while the proposed approach depends on the structure of the frequency decomposition, it is relatively independent of the detailed filter characteristics.

The proposed segmentation algorithm combines the color composition and spatial texture features to obtain segments of uniform texture. This is done in two steps. The first relies on a multigrid region growing algorithm to obtain a *crude* segmentation. The segmentation is crude due to the fact that the estimation of the spatial and color composition texture features requires a finite window. The second step uses an elaborate border refinement procedure to obtain accurate and precise border localization by appropriately combining the texture features with the underlying ACA segmentation.

The novelty of the proposed approach is twofold. First, by using features that adapt to the local image characteristics, it can account for the nonuniformity of the textures that are found in natural scenes, namely the intensity, color, and texture of a perceptually uniform region can change gradually (but significantly) across a region. The proposed algorithm adapts to such variations by estimating the color composition texture parameters over a hierarchy of window sizes that progressively decrease as the algorithm converges to the final segmentation. Second, in contrast to texture analysis/synthesis techniques that use a large number of parameters to describe texture, it relies on only a small number of parameters that can be robustly estimated (and easily adapted) based on the limited number of pixels that are available in each region.

The paper is organized as follows. Section II presents the color composition texture features. The extraction of the spatial texture features is presented in Section III. Section IV discusses the proposed algorithm for combining the spatial texture and color composition features to obtain an overall segmentation. Segmentation results and comparisons to other approaches are presented in Section IV. The conclusions are summarized in Section V.

II. COLOR COMPOSITION TEXTURE FEATURES

Color has been used extensively as a low-level feature for image retrieval [1], [39–41]. In this section, we discuss new

color composition texture features that take into account both image characteristics and human color perception.

A. Motivation and Prior Work

An important characteristic of human color perception is that the human eye cannot simultaneously perceive a large number of colors [27], even though under appropriate adaptation, it can distinguish more than two million colors [42]. In addition, the number of colors that can be internally represented and identified in cognitive space is about 30 [43]. A small set of color categories provides a very efficient representation, and more importantly, makes it easier to capture invariant properties in object appearance [44].

The idea of using a compact color representation in terms of dominant colors for image analysis was introduced by Ma *et al.* [36]. The representation they proposed consists of the dominant colors along with the percentage of occurrence of each color.

$$f_c = \{(c_i, p_i), i = 1, \dots, N, p_i \in [0, 1]\} \quad (1)$$

where each of the dominant colors, c_i , is a three-dimensional (3-D) vector in *RGB* space, and p_i are the corresponding percentages. Mojsilović *et al.* [27] adopted this representation using an (approximately) perceptually uniform color space (*Lab*). It has been shown that the quality of image retrieval algorithms can be substantially improved by using such color spaces [45].

As implied by (1), the dominant colors in [27], [36], [37] are fixed over an image or a collection of images. There are a number of approaches for extracting the dominant colors [27], [36], [38], [46]. A relatively simple and quite effective algorithm that can be used for obtaining the dominant colors of an image is the color segmentation algorithm proposed by Comaniciu and Meer [10], which is based on the “mean-shift” algorithm for estimating density gradients and is, thus, known as the *mean-shift* algorithm in the literature. However, it does not take into consideration spatial variations in the dominant colors of a (natural) image. Another approach that assumes constant dominant colors, but takes into account the spatial distribution of the original image colors, is presented in [47]. It recognizes the fact that human visual perception is more sensitive to changes in smooth regions and quantizes the colors more coarsely in detailed regions.

The above dominant color extraction techniques rely on the assumption that the characteristic colors of an image (or class of images) are relatively constant, i.e., they do not change due to variations in illumination, perspective, etc. This is true for images of fabrics, carpets, interior design patterns, and other pure textures. The class of images that we are considering, however, is more general and includes indoor and outdoor scenes, such as landscapes, cityscapes, plants, animals, people, and man-made objects. To handle such images, one has to account for color and lighting variations in the scene. Thus, while the above approaches can provide colors that are quite useful in characterizing the image as a whole, the resulting color classification (segmentation) could be quite inadequate due to lack of spatial adaptation and spatial constraints [8].

In addition to the spatially varying image characteristics, one has to take into consideration the adaptive nature of the HVS [48]. For example, we perceive regions with spatially varying color as a single color.

B. Proposed Color Composition Features

In order to account for the spatially varying image characteristics and the adaptive nature of the HVS, we introduce the idea of *spatially adaptive dominant colors*. The proposed color composition feature representation consists of a limited number of locally adapted dominant colors and the corresponding percentage of occurrence of each color within a certain neighborhood:

$$f_c(x, y, N_{x,y}) = \{(c_i, p_i), i = 1, \dots, M, p_i \in [0, 1]\} \quad (2)$$

where each of the dominant colors, c_i , is a 3-D vector in *Lab* space, and p_i are the corresponding percentages. $N_{x,y}$ denotes the neighborhood around the pixel at location (x, y) and M is the total number of colors in the neighborhood. A typical value is $M = 4$. As we will see below, this number can vary in different parts of the image.

One approach for obtaining spatially adaptive dominant colors is the ACA proposed in [8] and extended to color in [9]. The ACA is an iterative algorithm that can be regarded as a generalization of the K -means clustering algorithm [46], [49] in two respects: it is adaptive and includes spatial constraints. It segments the image into K classes. Each class is characterized by a spatially varying characteristic function $\mu^k(x, y)$ that replaces the spatially fixed cluster center of the K -means algorithm. Given these characteristic functions, the ACA finds the segmentation that maximizes the *a posteriori* probability density function for the distribution of regions given the observed image. The algorithm alternates between estimating the characteristic functions and updating the segmentation. The initial estimate is obtained by the K -means algorithm (and, in particular, the implementation described in [50]), which estimates the cluster centers (i.e., the dominant colors) by averaging the colors of the pixels in each class over the whole image. The key to adapting to the local image characteristics is that the ACA estimates the characteristic functions $\mu^k(x, y)$ by averaging over a sliding window whose size progressively decreases. Thus, the algorithm starts with global estimates and slowly adapts to the local characteristics of each region. As we will see below, it is these characteristic functions $\mu^k(x, y)$ that are used as the spatially adaptive dominant colors.

Fig. 3 compares the adaptive dominant colors obtained by the ACA [8] to the constant dominant colors obtained by the mean-shift algorithm [10]. The image resolution is 250×214 pixels. The examples for the mean-shift algorithm were generated using the “oversegmentation” setting. Note the false contours in the mean-shift algorithm in the water and the sky. Also, while there are color variations in the forest region, the segment boundaries do not appear to correspond to any true color boundaries. The ACA on the other hand, smoothes over the water, sky, and forest regions, while capturing the dominant edges of the scene. Note that the ACA was developed for images of objects with smooth surfaces and no texture. Thus,

in many textured regions, like the mountain area, the ACA oversegments the image, but the segments do correspond to actual texture details. Thus, it preserves the essential color characteristics of the texture. In other textured areas, like the forest, the ACA consolidates everything into one region. In such cases, the color variations in the texture are not as significant and can be represented by their local average.

In contrast to the other approaches, the ACA is quite robust to the number of classes. This is because the gradual color adaptation makes it possible to use one color class to represent a wide range of similar colors, provided that they vary gradually over the image. In addition, as we move to another part of the image, the same color class can be used to represent an entirely different color. Thus, one of the advantages of using the ACA to obtain spatially adaptive dominant colors is that we only need to specify the parameter K , which then determines the maximum number of dominant colors ($M \leq K$) in any given region of the image. We found that a small number (e.g., $K = 4$) is quite adequate.

The ACA segments the image into color classes, as shown in Fig. 3 (d). At every pixel in the image, each class is represented by the characteristic function $\mu^k(x, y)$, i.e., a color that is equal to the average color of the pixels in its neighborhood that belong to that class [8]. In the example of Fig. 3(c), each pixel is painted with the representative color of the class that it belongs to. Since the characteristic functions (dominant colors) are slowly varying, we can assume that they are approximately constant in the immediate vicinity of a pixel. Thus, the color composition feature representation of the form (2) at each point in the image consists of the (up to) K characteristic colors of each class and the associated percentage of pixels within a given window. Note that, given an ACA segmentation, the color feature vectors can be computed using a different window size, by averaging the colors of each class in the window and computing the percentage of pixels in each class.

C. Color Composition Similarity Metric

We now define a metric that measures the perceptual similarity between two color composition feature vectors. Based on human perception, the color composition of two images (or image segments) will be similar if the colors are similar and the total areas that each color occupies are similar [27], [38]. The definition of a metric that takes into account both the color and area differences, depends on the mapping between the dominant colors of the two images [38]. Various suboptimal solutions have been proposed [27], [36]. Mojsilovic *et al.* [38] proposed the OCCD, which finds the optimal mapping between the dominant colors of two images and, thus, provides a better similarity measure. The OCCD, which is closely related to the *earth mover’s distance* [51],¹ overcomes the (significant) problems of the other metrics, but in general, requires more computation. However, since we are primarily interested in comparing image segments that contain only a few colors (at most four), the additional overhead for the OCCD is reasonable. Moreover, we introduce an efficient implementation of OCCD for the problem at hand that

¹For a comparison of the two metrics, see [38].

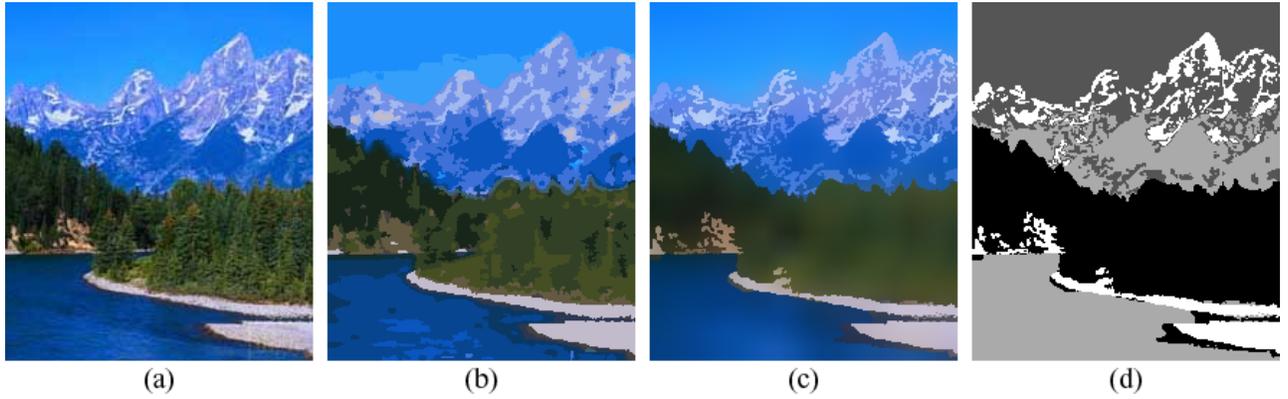


Fig. 3. Color Image Segmentation (a,b,c shown in color). (a) Original Color Image. (b) Mean Shift Algorithm. (c) ACA. (d) ACA Color Classes.

produces a close approximation of the optimal solution. The steps of the proposed OCCD implementation are as follows:

- 1) Given two color composition feature vectors f_c^1 and f_c^2 , create a stack of tokens (colors and corresponding percentages) for each feature vector, as shown in Fig. 4. Create an empty destination stack for each vector.
- 2) Select a pair of tokens (c_a, p_a) and (c_b, p_b) with nonzero percentages, one from each feature vector, whose colors are closest.
- 3) Move the token with the lowest percentage (e.g., (c_a, p_a)) to the destination stack. Split the other token into (c_b, p_a) and $(c_b, p_b - p_a)$, and move the first to the corresponding destination stack.
- 4) Repeat above steps with the remaining colors, until the initial stacks are empty.

An illustrative example is shown in Fig. 4. Note that even though this implementation is not guaranteed to result in the optimal mapping, in practice, given the small number of classes, it produces excellent results. On the other hand, it avoids the quantization error introduced by the original OCCD, and thus, can be even more accurate than the original implementation. Once the color correspondences are established, the OCCD distance is calculated as follows:

$$\mathcal{D}_c(f_c^1, f_c^2) = \sum_{i=0}^M \mathbf{d}(c_i^1, c_i^2) * p_i \quad (3)$$

where c_i^1 , c_i^2 , and p_i are the matched colors and corresponding percentage after the color matching process described above, and $\mathbf{d}(\cdot)$ is the distance in some color space. We use the Euclidean distance in *Lab* space.

III. SPATIAL TEXTURE FEATURES

As we discussed in the introduction, the color composition and spatial texture features are developed independently. We use only the grayscale component² of the image to derive the spatial texture features, which are then combined with the color composition features to obtain an intermediate crude segmentation. This is in contrast to the approaches described

²The grayscale component is obtained as a standard linear combination of gamma corrected RGB values.

Source Stack	Destination Stack
f_c^1 : (■,30)(■,30)(■,20)(■,20) f_c^2 : (■,40)(■,30)(■,30)	
f_c^1 : (■,30)(■,20)(■,20) f_c^2 : (■,10)(■,30)(■,30)	(■,30) (■,30)
f_c^1 : (■,20)(■,20) f_c^2 : (■,10)(■,30)	(■,30)(■,30) (■,30)(■,30)
f_c^1 : (■,20) f_c^2 : (■,10)(■,10)	(■,30)(■,30)(■,20) (■,30)(■,30)(■,20)
	(■,30)(■,30)(■,20)(■,10)(■,10) (■,30)(■,30)(■,20)(■,10)(■,10)

Fig. 4. Example of Simplified Version of OCCD (shown in color).

in [12], [27], where the color quantization/segmentation is used to obtain an achromatic pattern map which becomes the basis for texture feature extraction.

A. Motivation and Prior Work

Like many of the existing algorithms for texture analysis and synthesis (e.g., [5], [6], [26], [29–34], [52–57]), our approach is based on a multiscale frequency decomposition. Examples of such decompositions are the Cortex transform [58], the Gabor transform [30], [59], the steerable pyramid decomposition [60–62], and the discrete wavelet transform (DWT) [63], [64], which can be regarded as a crude approximation of the cortex transform. We base our spatial texture feature extraction on one of the more accurate approximations of the visual cortex, the steerable pyramid decomposition, which can be designed to produce any number of orientation bands. The proposed methodology, however, can make use of any of the decompositions mentioned above. Fig. 5 shows examples of frequency decompositions that can be obtained with the steerable pyramid.

One of the most commonly used features for texture analysis in the context of multiscale frequency decompositions is the energy of the subband coefficients [3–7], [65]. Various nonlinear operations have been used to boost up the sparse subband coefficients [3], [36], [57], [65]. Our approach is

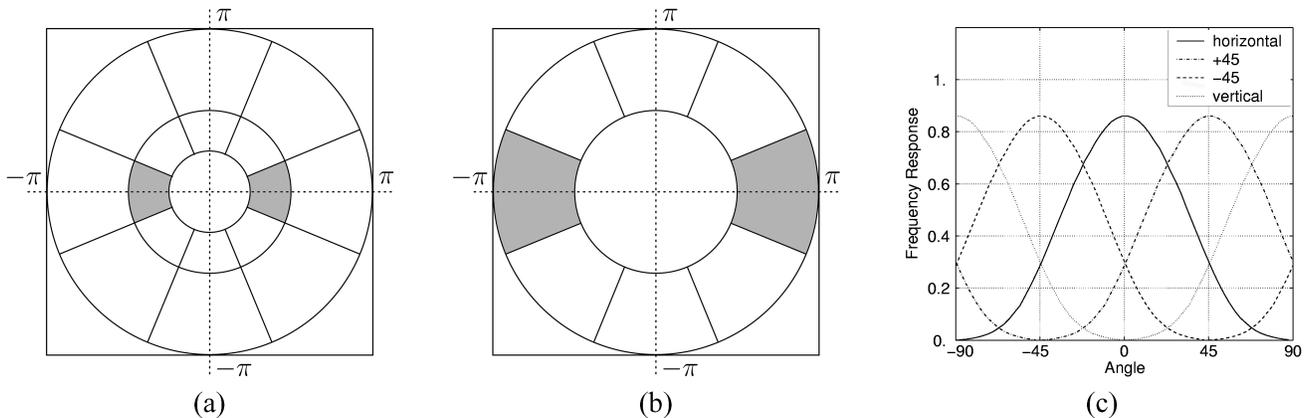


Fig. 5. Steerable filter decomposition. (a) Ideal two-level decomposition. (b) Ideal one-level decomposition (Horizontal bands shown in gray). (c) Circular cross section of real steerable filter frequency response.

based on the *local median energy* of the subband coefficients, where the energy is defined as the square of the coefficients. As we saw in the introduction, the advantage of the median filter is that it suppresses textures associated with transitions between regions, while it responds to texture within uniform regions. The use of median local energy as a nonlinear operation also agrees with Graham [66] and Graham and Sutter [67], [68], who conclude that a nonlinear operator in texture segregation must have accelerating/expansive nature.

B. Proposed Spatial Texture Features

We use a steerable filter decomposition with four orientation subbands (horizontal, vertical, $+45^\circ$, -45°) as shown in Fig. 5. Most researchers have used four to six orientation bands to approximate the orientation selectivity of the HVS (e.g., [58], [69]). Since the images are fairly small, we found that a one-level decomposition (lowpass band, four orientation bands, and highpass residue, as shown in Fig. 5(b)) is adequate. Out of those we use only the four orientation bands. Our goal is to identify regions with a dominant orientation (*horizontal*, *vertical*, $+45^\circ$, -45°); all other regions will be classified as *smooth* (not enough energy in any orientation) or *complex* (no dominant orientation).

Fig. 5(c) shows a circular cross section of the steerable filter responses. Note that there is a large overlap between neighboring filters. Thus, even when there is a dominant orientation, the response of the neighboring filters will be quite significant, especially when the texture orientation falls between the main orientations of the steerable filters. Therefore, it is the maximum of the four coefficients that determines the orientation at a given pixel location.³

The spatial texture feature extraction consists of two steps. First, we classify pixels into smooth and nonsmooth categories. Then we further classify nonsmooth pixels into the remaining categories. Let $s_0(x, y)$, $s_1(x, y)$, $s_2(x, y)$, and $s_3(x, y)$

represent the steerable subband coefficient at location (x, y) that corresponds to the horizontal (0°), diagonal with positive slope ($+45^\circ$), vertical (90°), and diagonal with negative slope (-45°) directions, respectively. We will use $s_{\max}(x, y)$ to denote the maximum (in absolute value) of the four coefficients at location (x, y) , and $s_i(x, y)$ to denote the subband index that corresponds to that maximum.

A pixel will be classified as smooth if there is no substantial energy in any of the four orientation bands. As we discussed above, a median operation is necessary for boosting the response to texture within uniform regions and suppressing the response due to textures associated with transitions between regions. A pixel (x, y) is classified as smooth if the median of $s_{\max}(x', y')$ over a neighborhood of (x, y) is below a threshold T_0 . This threshold is determined using a two-level K -means algorithm that segments the image into smooth and nonsmooth regions. A cluster validation step is necessary at this point. If the clusters are too close, then the image may contain only smooth or nonsmooth regions, depending on the actual value of the cluster center.

The next step is to classify the pixels in the nonsmooth regions. As we mentioned above, it is the maximum of the four subband coefficients, $s_i(x, y)$, that determines the orientation of the texture at each image point. The texture classification is based on the local histogram of these indices. Again, a median type of operation is necessary for boosting the response to texture within uniform regions and suppressing the response due to textures associated with transitions between regions. This is done as follows. We compute the percentage for each value (orientation) of the index $s_i(x', y')$ in the neighborhood of (x, y) . Only the nonsmooth pixels within the neighborhood are considered. If the maximum of the percentages is higher than a threshold T_1 (e.g., 36%) and the difference between the first and second maxima is greater than a threshold T_2 (e.g., 15%), then there is a dominant orientation in the window and the pixel is classified accordingly. Otherwise, there is no dominant orientation, and the pixel is classified as complex. The first threshold ensures the existence of a dominant orientation and the second ensures its uniqueness. An example is presented in Fig. 6. The grayscale component of the original color image is shown in Fig. 6(a). In Fig. 6(b),

³In [70], we used the closeness of the 1st and 2nd maxima of the four subband coefficients as an indication of a complex region. However, such a criterion misclassifies, as complex, textures with orientations that fall between the main orientations of the steerable filters, for which the responses of the two filters are close. Using sharper orientation filters will narrow the range of misclassified orientations, but will not entirely eliminate the problem.

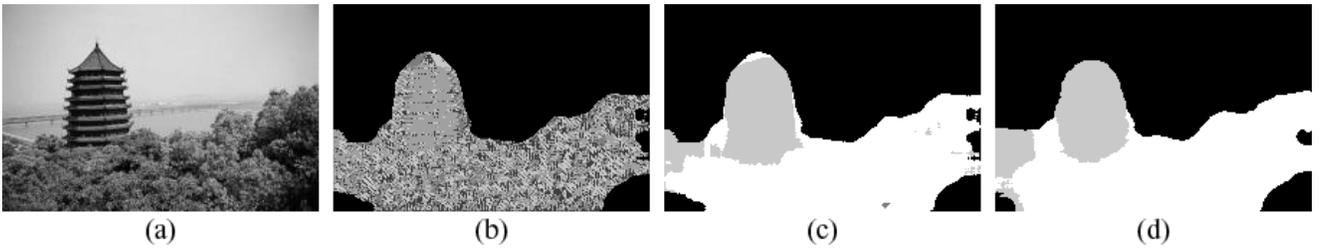


Fig. 6. Texture Map Extraction. (a) Grayscale Component of Original Image. (b) Smooth (black) and Nonsmooth (different shades of gray) Regions Using Steerable Filter Decomposition. (c) Texture Classes Using Steerable Filter Decomposition. (d) Texture Classes Using Gabor Decomposition. Texture window size = 23×23 .

the smooth regions are shown in black, and the nonsmooth regions are shown in different shades of gray representing the indices s_i of the subband coefficients with maximum energy. Fig. 6(c) shows the resulting texture classes, where black denotes smooth, white denotes complex, and light gray denotes horizontal textures. (There are no diagonal textures in this example.) The window for the median operation was 23×23 .

C. Spatial Texture Similarity Metric

To measure the similarity between two spatial texture features f_t^1 and f_t^2 , we define the following distance:

$$\mathcal{D}_t(f_t^1, f_t^2) = \begin{cases} 0 & \text{if } f_t^1 = f_t^2 \\ t_{i,j} & \text{if } f_t^1 = i \neq f_t^2 = j \end{cases} \quad (4)$$

where $t_{i,j}$ is a threshold that will, in general, depend on the combination of texture classes (smooth, horizontal, vertical, $+45^\circ$, -45° , and complex); in the following, we will assume two different values for $t_{i,j}$, one for within nonsmooth texture classes (e.g., $t_{i,j} = 2.5$) and the other for between smooth and nonsmooth classes (e.g., $t_{i,j} = 5$). This metric will be used in combination with the color metric to determine the overall similarity between two texture (color composition and spatial texture) feature vectors. The value of $t_{i,j}$ represents the penalty for inconsistent color composition and spatial texture classification. The idea is that, if the spatial texture classes are the same, then we allow for more color variation. If they are not the same, the colors have to be more similar in order for pixels to belong to the same class.

D. Implementation Details and Other Considerations

In the texture class extraction procedure, we found that the window size for median operator is of critical importance. It must be large enough to capture the local texture characteristics, but not too large to avoid border effects. Our experiments indicate that window sizes in the range of 17×17 to 25×25 pixels are suitable for the steerable filter decomposition. A more careful determination of the window size should be based on subjective experiments. Note also, that the window size depends on the specific decomposition. For example, we found that the DWT requires smaller window sizes [71]. That is because in the DWT the subbands are downsampled, while in the steerable decomposition that we use they are not. The window size also depends on the extent of the analysis filters.

We have also experimented with alternative ways to obtain the smooth vs. nonsmooth classification. For example, we tried an approach similar to the one described in [71], whereby the local median energy of each subband coefficient is computed first, followed by a two-level K -means. A pixel is then classified as smooth if all subbands belong to the low energy class. This leads to similar results but involves more computation. Another approach is to apply K -means to the vector of the local median energies of the four subband coefficients. We found that the proposed algorithm has the best performance in terms of accuracy and robustness, as well as computational efficiency.

We also considered a number of alternative decompositions. In [70], [71] we compared the performance of the DWT and the steerable filter decomposition using similar classification procedures, and found that the steerable filter decomposition produces superior results. As we discussed above, this is mainly due to the fact that the DWT does not separate the two diagonal directions. A number of other filter banks that generate complete/over-complete orientational decompositions can be used instead of the steerable filters. For example, we tried a one-level, four-orientation Gabor decomposition⁴ with the rest of the procedure unchanged, and found that its performance is comparable to that of the steerable filters. Fig. 6(d) shows the resulting texture class map. Note that because of the “max” operator, using sharper orientation filters will not lead to better texture classification.

IV. SEGMENTATION ALGORITHM

In this section, we present an algorithm that combines the color composition and spatial texture features to obtain the overall image segmentation.

The smooth and nonsmooth regions are considered separately. As we discussed in Section II, the ACA was developed for images with smooth regions. Thus, in those regions, we can rely on the ACA for the final segmentation. However, some region merging may be necessary. Thus, in the smooth regions, we consider all pairs of connected neighboring segments, and merge them if the average color difference across the common border is below a given threshold. The color difference at each point along the border is based on the spatially adaptive dominant colors provided by ACA, which thus provides a natural and robust region merging criterion. Finally, any remaining

⁴The Gabor filters we used are of size 9×9 pixels and we used the same filter design and parameters as that in [57].

small color segments⁵ that are connected to nonsmooth texture regions are considered together with the nonsmooth regions, and are assumed to have the same label as any nonsmooth region they are connected to. Fig. 7 shows the different stages of the algorithm; (a) shows an original color image, (b) shows the ACA segmentation (dominant colors), (c) shows the texture classes, and (d) and (e) show the color segments in the smooth regions before and after the merging operation. The nonsmooth regions are shown in white, while the smooth regions have been painted by the average color of each connected segment.

We now consider the nonsmooth regions, which have been further classified into horizontal, vertical, $+45^\circ$, -45° , and complex categories. These categories must be combined with the color composition features to obtain segments of uniform texture. We obtain the final segmentation in two steps. The first combines the color composition and spatial texture features to obtain a *crude* segmentation, and the second uses an elaborate border refinement procedure, which relies on the color information to obtain accurate and precise border localization.

A. Crude Segmentation

The crude segmentation is obtained with a multigrid region growing algorithm. We start with pixels located on a coarse grid in nonsmooth regions, and compute the color composition features using a window size equal to twice the grid spacing, i.e., with 50% overlap with adjacent horizontal or vertical windows. Only pixels in nonsmooth regions and smooth pixels that are neighbors with nonsmooth pixels are considered. Note that the color composition features are computed at the full resolution; it is the merging only that is carried out on different grids. The merging criterion, which we discuss below, combines the color composition and spatial texture information.

Ideally, a pair of pixels belong to the same region, if their color composition features are similar and they belong to the same spatial texture category. Thus, to determine if a pair of pixels belong to the same region, we compute the distance between their feature vectors $f^1 = (f_c^1, f_t^1)$ and $f^2 = (f_c^2, f_t^2)$, which include both the color composition and spatial texture features:

$$\mathcal{D}(f^1, f^2) = \mathcal{D}_c(f_c^1, f_c^2) + \mathcal{D}_t(f_t^1, f_t^2) \quad (5)$$

where $\mathcal{D}_c(\cdot)$ and $\mathcal{D}_t(\cdot)$ were defined in the previous sections.

In addition, we incorporate spatial constraints in the form of Markov random fields (MRFs). Using an MAP formulation similar to that of [8], whereby the conditional density of the observation is Gaussian and the *a priori* density of the class assignments is MRF, a pixel is assigned to the class that minimizes the following function:

$$\mathcal{D}(f^0, f^i) + \beta(M^i - N^i) \quad \text{all } i \quad (6)$$

where f^0 is the feature vector of the current pixel, f^i is the feature vector of its i th neighbor, N^i (M^i) is the number of nonsmooth neighbors that belong to the same (different) class as the i th neighbor, and β represents the strength of the

spatial constraint. Thus, a pixel is more likely to belong to a class when many of its neighbors belong to the same class. In order to allow new classes to be created, we arbitrarily set the feature distance between the current pixel and a pixel in a new class equal to a threshold t_0 . Note that because of the MRF constraint, the likelihood of appearance of a new class decreases as β increases.

Since the MRF constraint is symmetric, it is necessary to iterate a few times for a given grid spacing. The grid spacing and window size are then reduced by a factor of two, and the procedure is repeated until the spacing is equal to one pixel. Fig. 7 (f) shows an example of the resulting crude segmentation. Fig. 8 shows examples of crude segmentations obtained with different values of the parameter β . Note that in Fig. 7 (d), (e), (f), (g), and in Fig. 8 the different segments have been painted by the average color of the region, while in Fig. 7 (d) and (e) white represents nonsmooth regions.

B. Border Refinement Using Adaptive Clustering

Once the crude segmentation is obtained, we refine it by adaptively adjusting the borders using the color composition texture features. The approach is similar to that of the ACA [8], and is illustrated in Fig. 9. The dotted line represents the actual boundary and the solid line denotes the boundary location in the current iteration. For each pixel in the image, we use a small window to estimate the pixel texture characteristics, i.e., a color composition feature vector of the form (2), and a larger window to obtain a localized estimate of the region characteristics. For each texture segment that the larger window overlaps, we obtain a separate color composition feature vector, that is, we find the average color and percentage for each of the dominant colors. We then use the OCCD criterion to determine which segment has a feature vector that is closest to the feature vector of the small window, and classify the pixel accordingly. An MRF constraint similar to the one in (6) is added to insure region smoothness. The above procedure could be repeated for each pixel in a raster scan. To save computation, however, we only consider pixels on the border between nonsmooth segments or between smooth and nonsmooth segments. (The borders between smooth segments have already been fixed.) A few iterations are necessary for convergence. The iterations converge when the number of pixels that change class is below a given threshold (e.g., equal to the average of the widths of the two windows). We then reduce the window sizes and repeat the procedure. For example, we use a series of window pairs starting from 35/5 and ending with 11/3. (The window size is odd so that they are symmetric.)

One of the important details in the above procedure is that each of the candidate regions in the larger window must be large enough in order to obtain a reliable estimate of its texture attributes. If the area of a segment that overlaps the larger window is not large enough, then the region is not a valid candidate. A reasonable choice for the threshold for the overlapping area is to use the product of the window sizes divided by 2.

As we mentioned above, the refinement procedure is applied to the whole image except the smooth regions, where as we

⁵For example, we used a threshold equal to the area of a one pixel wide narrow strip, whose length is equal to a half of the maximum image dimension.

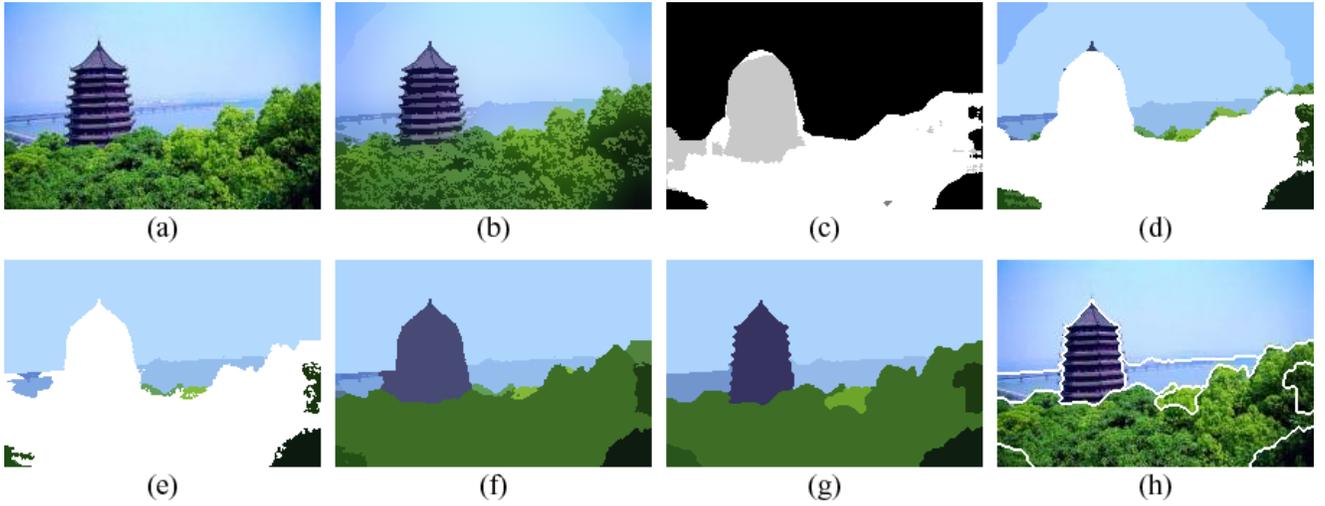


Fig. 7. Color and Texture Image Segmentation (a,b,d,e,f,g,h shown in color). (a) Original Color Image. (b) Color Segmentation (ACA). (c) Texture Classes. (d) Smooth Regions Before Merging (e) Smooth Regions After Merging (f) Crude Segmentation. (g) Final Segmentation. (h) Final Segmentation (on original image). Texture window size = 23×23 and $\beta = 0.8$. White regions in (c) denote complex regions. White regions in (d) and (e) denote nonsmooth regions.

saw, the ACA provides accurate segmentation and no refinement is necessary. Moreover, it is easy and interesting to explain why the border refinement procedure, which is designed for nonsmooth textures, will not work in the smooth regions. Let us assume we have a border between two smooth regions as shown in Fig. 9. Let the local feature be $f_c(x, y, N_{x,y}) = \{(c_1, p_1), (c_2, p_2)\}$ and the features of the two segments be $f_c^a(x, y, N_{x,y}) = \{(c_a, 1)\}$ and $f_c^b(x, y, N_{x,y}) = \{(c_b, 1)\}$. Note that these are smooth segments, and thus, each is characterized by one color. Since the colors are slowly varying, we have $c_1 \approx c_a$ and $c_2 \approx c_b$. Thus, the OCCD feature distances between the local feature and the two segment features become

$$D_c(f_c, f_c^a) = \mathbf{d}(c_1, c_a) * p_1 + \mathbf{d}(c_2, c_a) * p_2 \approx \mathbf{d}(c_b, c_a) * p_2$$

$$D_c(f_c, f_c^b) = \mathbf{d}(c_1, c_b) * p_1 + \mathbf{d}(c_2, c_b) * p_2 \approx \mathbf{d}(c_a, c_b) * p_1$$

where $\mathbf{d}(\cdot)$ represents the distance between the dominant colors in a given color space as we saw in (3). Thus, the OCCD feature distances are actually determined by the percentages of the colors, and, hence, the refinement will lead to the wrong results.

The final segmentation results are shown in Fig. 7(g) and (h). Additional segmentation results are shown in Fig. 10; the resolution of the images varies from 180×149 to 214×250 pixels. Most of the images shown were found in the Internet; example (e) comes from the Berkeley image database [72]. Fig. 11 shows the segmentation results obtained by JSEG [12], a segmentation algorithm that is also based on texture and color. We chose the “no merge” option for the JSEG examples shown. Thus, in comparing with the results of the proposed algorithm in Fig. 10, one should keep in mind that the JSEG images are oversegmented. It is fair to assume that a reasonable region merging step could be applied, even though the JSEG merging criterion does not work that well. Thus, for example, there are no significant differences between the two algorithms in the forest area of example (b) or the flower area of example (c). On the other hand, there are

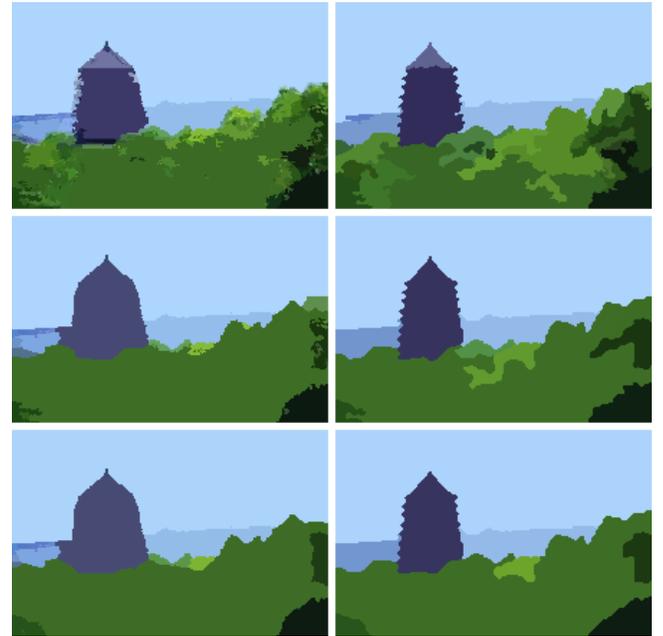


Fig. 8. Illustrating the effects of spatial constraints (images shown in color). Left row shows crude segmentations and right row shows final segmentations. From top to bottom $\beta = 0.0, 0.5, 1.0$. Texture window size = 23×23 .

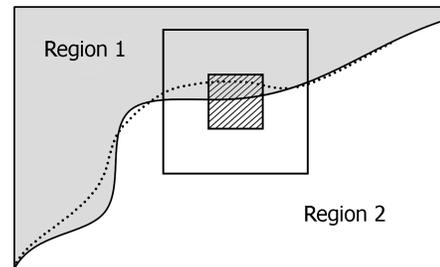


Fig. 9. Illustration of Border Refinement.

significant differences in (g) that cannot be eliminated with region merging, e.g., around the boat or the boundary between the city and the forest at the top of the picture. Similarly, there are significant differences in example (i), where the tower behind the train is segmented well by our algorithm, but is merged with one of the sky segments by JSEG. Note that in example (h), the color of the sky is too close to the color of the mountains, and thus, both algorithms merge part of the mountains with the sky. Note that the proposed algorithm occasionally also oversegments some textured regions, e.g., in the lower left corner of example (a) and the forest area of example (b). For such cases, a region merging criterion similar to the one we described for the smooth regions can be applied to the textured regions. Fig. 11 (a), (b), (i), (h), and (j) also demonstrate that the proposed algorithm can handle color and texture gradients.

V. CONCLUSION

We presented a new approach for image segmentation that is based on low-level features for color and texture. It is aimed at segmentation of natural scenes, in which the color and texture of each segment does not typically exhibit uniform statistical characteristics. The proposed approach combines knowledge of human perception with an understanding of signal characteristics in order to segment natural scenes into perceptually/semantically uniform regions.

The proposed approach is based on two types of spatially adaptive low-level features. The first describes the local color composition in terms of spatially adaptive dominant colors, and the second describes the spatial characteristics of the grayscale component of the texture. Together they provide a simple and effective characterization of texture that can be used to obtain robust, and at the same time, accurate and precise segmentations. The performance of the proposed algorithms has been demonstrated in the domain of photographic images, including low resolution, degraded, and compressed images. As we have shown, one of the strengths of the algorithm is that it can handle color and texture gradients, which are commonly found in perceptually uniform regions of natural scenes.

The image segmentation results can be used to derive region-specific color and texture features. These can be combined with other segment information, such as location, boundary shape, and size, in order to extract semantic information. Such semantic information may be adequate to classify an image correctly, even though our segmentation results may not always necessarily correspond to semantic objects as perceived by human observers.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) under Grant No.CCR-0209006. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

- [1] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions and open issues," *J. Visual Communication and Image Representation*, vol. 10, pp. 39–62, Mar. 1999.
- [2] W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 1349–1379, Dec. 2000.
- [3] A. Kundu and J.-L. Chen, "Texture classification using QMF bank-based subband decomposition," *Computer Vision, Graphics, and Image Processing. Graphical Models and Image Processing*, vol. 54, pp. 369–384, Sept. 1992.
- [4] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Processing*, vol. 2, pp. 429–441, Oct. 1993.
- [5] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, pp. 1549–1560, Nov. 1995.
- [6] T. Randen and J. H. Husoy, "Texture segmentation using filters with optimized energy separation," *IEEE Trans. Image Processing*, vol. 8, pp. 571–582, Apr. 1999.
- [7] G. V. de Wouwer, P. Scheunders, and D. Van Dyck, "Statistical texture characterization from discrete wavelet representations," *IEEE Trans. Image Processing*, vol. 8, pp. 592–598, Apr. 1999.
- [8] T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Processing*, vol. SP-40, pp. 901–914, Apr. 1992.
- [9] M. M. Chang, M. I. Sezan, and A. M. Tekalp, "Adaptive Bayesian segmentation of color images," *Journal of Electronic Imaging*, vol. 3, pp. 404–414, Oct. 1994.
- [10] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *IEEE Conf. Computer Vision and Pattern Recognition*, (San Juan, Puerto Rico), pp. 750–755, June 1997.
- [11] J. Luo, R. T. Gray, and H.-C. Lee, "Incorporation of derivative priors in adaptive Bayesian color image segmentation," in *Proc. Int. Conf. Image Processing (ICIP-98)*, vol. III, (Chicago, IL), pp. 780–784, Oct. 1998.
- [12] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 800–810, Aug. 2001.
- [13] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and texture-based image segmentation using EM and its application to content based image retrieval," in *ICCV*, pp. 675–682, 1998.
- [14] D. K. Panjwani and G. Healey, "Markov random-field models for unsupervised segmentation of textured color images," *PAMI*, vol. 17, pp. 939–954, Oct. 1995.
- [15] J. Wang, "Stochastic relaxation on partitions with connected components and its application to image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 6, pp. 619–636, 1998.
- [16] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 11, pp. 1530–1544, 1997.
- [17] W. Ma and B. S. Manjunath, "Edge flow: a technique for boundary detection and image segmentation," *IEEE Trans. Image Processing*, vol. 9, pp. 1375–1388, Aug. 2000.
- [18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 888–905, Aug. 2000.
- [19] A. Mojsilovic and B. Rogowitz, "Capturing image semantics with low-level descriptors," in *Proc. Int. Conf. Image Processing (ICIP-01)*, (Thessaloniki, Greece), pp. 18–21, Oct. 2001.
- [20] A. Amir and M. Lindenbaum, "A generic grouping algorithm and its quantitative analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 168–185, Feb. 1998.
- [21] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 1053–1074, Oct. 2001.
- [22] T. P. Minka and R. W. Picard, "Interactive learning using a society of models," *Pattern Recognition*, vol. 30, pp. 565–581, Apr. 1997. Special issue on Image Databases.
- [23] A. Jaimes and S. F. Chang, "Model-based classification of visual information for content-based retrieval," in *Storage and Retrieval for Image and Video Databases VII*, vol. Proc. SPIE, Vol. 3656, (San Jose, CA), pp. 402–414, 1999.
- [24] J. Fan, Z. Zhu, and L. Wu, "Automatic model-based object extraction algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1073–1084, Oct. 2001.

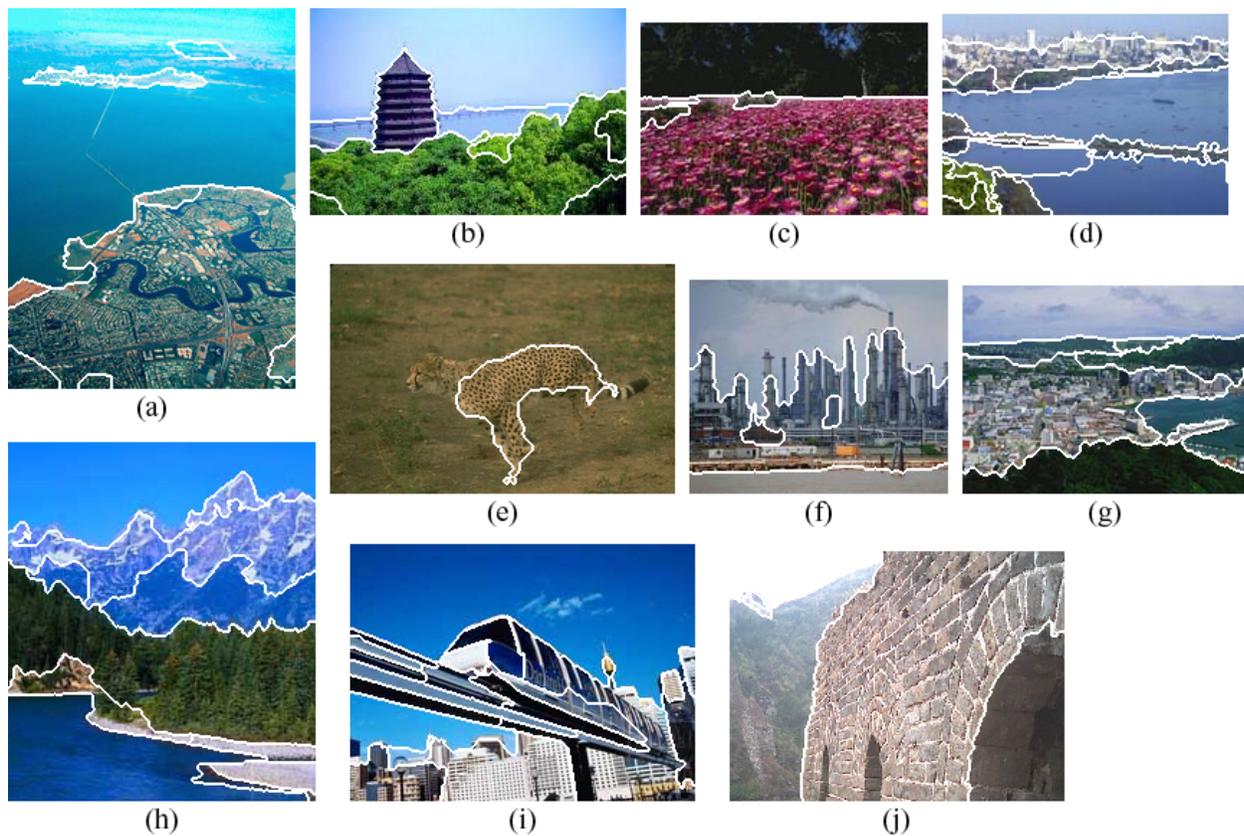


Fig. 10. Image Segmentation Based on Steerable Filter Decomposition (images shown in color). Texture window size = 23×23 and $\beta=0.8$. Edges are superimposed on original images.

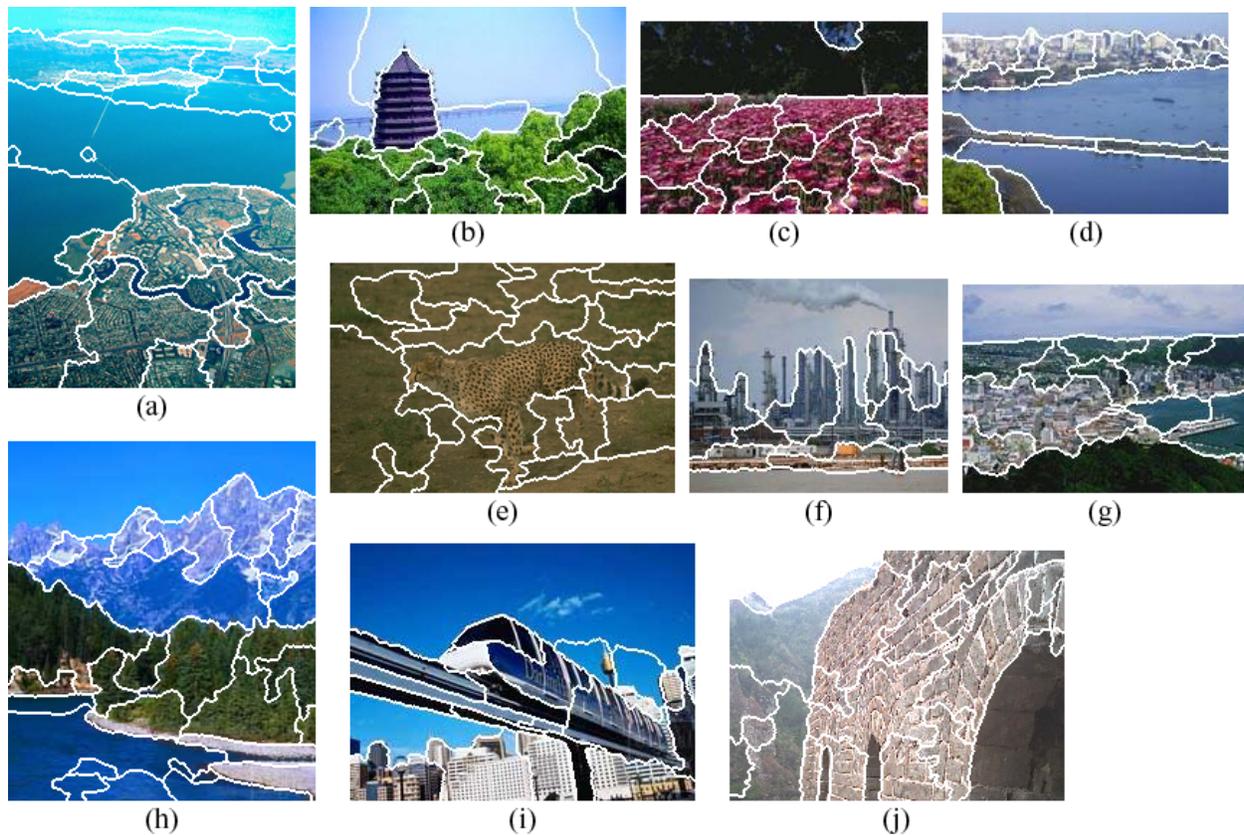


Fig. 11. Image Segmentation Using JSEG [12] With Least Merge Setting (images shown in color). Edges are superimposed on original images.

- [25] E. P. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," in *Proc. Int. Conf. Image Processing (ICIP-98)*, vol. 1, (Chicago, IL), pp. 62–66, Oct. 1998.
- [26] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Computer Vision*, vol. 40, pp. 49–71, Oct. 2000.
- [27] A. Mojsilović, J. Kovačević, J. Hu, R. J. Safranek, and S. K. Ganapathy, "Matching and retrieval based on the vocabulary and grammar of color patterns," *IEEE Trans. Image Processing*, vol. 1, pp. 38–54, Jan. 2000.
- [28] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 142–159, Feb. 2000.
- [29] D. Cano and T. H. Minh, "Texture synthesis using hierarchical linear transforms," *Signal Processing*, vol. 15, pp. 131–148, 1988.
- [30] M. Porat and Y. Y. Zeevi, "Localized texture processing in vision: Analysis and synthesis in gaborian space," *IEEE Trans. Biomed. Eng.*, vol. 36, no. 1, pp. 115–129, 1989.
- [31] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. Int. Conf. Image Processing (ICIP-95)*, vol. III, (Washington, DC), pp. 648–651, Oct. 1995.
- [32] J. Portilla, R. Navarro, O. Nestares, and A. Taberero, "Texture synthesis-by-analysis based on a multiscale early-vision model," *Optical Engineering*, vol. 35, no. 8, pp. 2403–2417, 1996.
- [33] S. Zhu, Y. N. Wu, and D. Mumford, "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 693–696, 1996.
- [34] J. S. De Bonet and P. A. Viola, "A non-parametric multi-scale statistical model for natural images," *Adv. in Neural Info. Processing Systems*, vol. 9, 1997.
- [35] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," *IEEE Trans. Image Processing*, vol. 11, pp. 146–158, Feb. 2002.
- [36] W. Y. Ma, Y. Deng, and B. S. Manjunath, "Tools for texture/color based search of images," in *Human Vision and Electronic Imaging II* (B. E. Rogowitz and T. N. Pappas, eds.), vol. Proc. SPIE, Vol. 3016, (San Jose, CA), pp. 496–507, Feb. 1997.
- [37] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin, "An efficient color representation for image retrieval," *IEEE Trans. Image Processing*, vol. 10, pp. 140–147, Jan. 2001.
- [38] A. Mojsilović, J. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval, and analysis," *IEEE Trans. Image Processing*, vol. 11, pp. 1238–1248, Nov. 2002.
- [39] M. Swain and D. Ballard, "Color indexing," *Int. J. Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [40] W. Niblack, R. Berber, W. Equitz, M. Flickner, E. Glaman, D. Petkovic, and P. Yanker, "The QBIC project: Querying images by content using color, texture, and shape," in *Storage and Retrieval for Image and Video Data Bases*, vol. Proc. SPIE, Vol. 1908, (San Jose, CA), pp. 173–187, 1993.
- [41] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 703–715, June 2001.
- [42] M. R. Pointer and G. G. Attridge, "The number of discernible colors," *Color Research and Application*, vol. 23, no. 1, pp. 52–54, 1998.
- [43] G. Derefeldt and T. Swartling, "Color concept retrieval by free color naming," *Displays*, vol. 16, pp. 69–77, 1995.
- [44] S. N. Yendrikhovskij, "Computing color categories," in *Human Vision and Electronic Imaging V* (B. E. Rogowitz and T. N. Pappas, eds.), vol. Proc. SPIE Vol. 3959, (San Jose, CA), Jan. 2000.
- [45] J. Smith and S. F. Chang, "Single color extraction and image query," in *Proc. IEEE Int. Conf. Image Proc.*, 1995.
- [46] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
- [47] Y. Deng, S. Kenney, M. S. Moore, and B. S. Manjunath, "Peer group filtering and perceptual color image quantization," in *Proc. IEEE Int. Symposium on Circuits and Systems VLSI*, vol. 4, (Orlando, FL), pp. 21–24, June 1999.
- [48] P. K. Kaiser and R. M. Boynton, *Human Color Vision*. Washington, DC: Optical Society of America, 1996.
- [49] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Addison, 1974.
- [50] R. M. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Commun.*, vol. COM-30, pp. 381–389, Feb. 1982.
- [51] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. IEEE International Conference on Computer Vision*, (Bombay, India), 1998.
- [52] J. R. Bergen and M. S. Landy, "Computational modeling of visual texture segregation," in *Computational Models of Visual Processing* (M. S. Landy and J. A. Movshon, eds.), pp. 253–271, Cambridge, MA: MIT Press, 1991.
- [53] M. R. Turner, "Texture discrimination by Gabor functions," *Biol. Cybern.*, vol. 55, pp. 71–82, 1986.
- [54] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55–73, Jan. 1990.
- [55] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America A*, vol. 7, pp. 923–932, 1990.
- [56] D. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. Image Processing*, vol. 4, pp. 947–964, July 1995.
- [57] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 837–842, Aug. 1996.
- [58] A. B. Watson, "The cortex transform: Rapid computation of simulated neural images," *Computer Vision, Graphics, and Image Processing*, vol. 39, pp. 311–327, 1987.
- [59] J. G. Daugman and D. M. Kamen, "Pure orientation filtering: A scale invariant image-processing tool for perception research and data compression," *Behavior Research Methods, Instruments, and Computers*, vol. 18, no. 6, pp. 559–564, 1986.
- [60] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 891–906, Sept. 1991.
- [61] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multi-scale transforms," *IEEE Trans. Inform. Theory*, vol. 38, pp. 587–607, Mar. 1992.
- [62] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proc. ICIP-95*, vol. III, (Washington, DC), pp. 444–447, Oct. 1995.
- [63] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.
- [64] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [65] T. Chang and C.-C. J. Kuo, "Texture segmentation with tree-structured wavelet transform," in *Proc. IEEE-SP Int. Symp. Time-Frequency and Time-Scale Analysis*, pp. 543–546, Oct. 1992.
- [66] N. Graham, "Non-linearities in texture segregation," in *CIBA Foundation Symposium* (G. R. Bock and J. A. Goode, eds.), vol. 184, pp. 309–329, New York: Wiley, 1994.
- [67] N. Graham and A. Sutter, "Spatial summation in simple(fourier) and complex(non-fourier) texture channels," *Vision Research*, vol. 38, pp. 231–257, 1998.
- [68] N. Graham and A. Sutter, "Normalization: contrast-gain control in simple(fourier) and complex(non-fourier) pathways of patten vision," *Vision Research*, vol. 40, pp. 2737–2761, 2000.
- [69] S. Daly, "The visible differences predictor: an algorithm for the assessment of image fidelity," in *Digital Images and Human Vision* (A. B. Watson, ed.), pp. 179–206, Cambridge, MA: The MIT Press, 1993.
- [70] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, "Perceptual color and texture features for segmentation," in *Human Vision and Electronic Imaging VIII* (B. E. Rogowitz and T. N. Pappas, eds.), vol. Proc. SPIE Vol. 5007, (Santa Clara, CA), pp. 340–351, Jan. 2003.
- [71] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, "Adaptive image segmentation based on color and texture," in *Proc. Int. Conf. Image Processing (ICIP-02)*, vol. 2, (Rochester, NY), pp. 789–792, Sept. 2002.
- [72] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, (Vancouver, Canada), July 2001.

PLACE
PHOTO
HERE

Junqing Chen (S'02) was born in Hangzhou, China. She received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China in 1996 and 1999, respectively, and the Ph.D. degree in electrical engineering from Northwestern University, Evanston, IL, in 2003.

During the summer of 2001, she was a student intern at the Visual Analysis Group at the IBM T. J. Watson Research Center in New York. From January to July 2004, she was a Postdoctoral Fellow at Northwestern University. In August, 2004, she joined Unilever Research at Edgewater, NJ as imaging scientist. Her research interests include image and signal analysis, perceptual models for image processing, image and video quality, and machine learning.

PLACE
PHOTO
HERE

Bernice E. Rogowitz (SM'04) received her B.S. degree from Brandeis University, her Ph.D. degree from Columbia University, in experimental psychology, and was an NIH Postdoctoral Fellow in the Laboratory of Psychophysics at Harvard University.

Bernice is currently the Program Director for Research Effectiveness at the IBM T. J. Watson Research Center in New York. Previously, she managed the Visual Analysis Group at IBM Research, which conducted experimental research in human perception, built interactive software tools for representing and exploring data, and worked with customers to develop these methods within the context of real-world problems. Her research includes publications in human spatial and color vision, visualization, and perceptually-based semantic approaches to image analysis and retrieval.

In 1988, Dr. Rogowitz founded the IS&T/SPIE Conference on Human Vision and Electronic Imaging, which she continues to co-chair. She has served on the board of the IS&T from 1997-2002, was elected an IS&T Fellow in 2000 and a Senior Member of the IEEE in 2004.

PLACE
PHOTO
HERE

Thrasyloulos N. Pappas (M'87, SM'95) received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1979, 1982, and 1987, respectively. From 1987 until 1999, he was a Member of the Technical Staff at Bell Laboratories, Murray Hill, NJ. In September 1999, he joined the Department of Electrical and Computer Engineering at Northwestern University as an associate professor. His research interests are in image and video compression, video transmission

over packet-switched networks, perceptual models for image processing, model-based halftoning, image and video analysis, video processing for sensor networks, audiovisual signal processing, and DNA-based digital signal processing.

Dr. Pappas has served as chair of the IEEE Image and Multidimensional Signal Processing Technical Committee, associate editor and electronic abstracts editor of the IEEE Transactions on Image Processing, technical program co-chair of ICIP-01 and the Symposium on Information Processing in Sensor Networks (IPSN-04), and since 1997 he has been co-chair of the SPIE/IS&T Conference on Human Vision and Electronic Imaging. He is also co-chair for the 2005 IS&T/SPIE Symposium on Electronic Imaging: Science and Technology.

PLACE
PHOTO
HERE

Aleksandra (Saška) Mojsilović (S'93, M'98) was born in Belgrade, Yugoslavia. She received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Belgrade, Yugoslavia, in 1992, 1994, and 1997, respectively. Since then she has worked at IBM Research (2000-present), Bell Laboratories (1998-2000) and as a Faculty Member at the University of Belgrade (1997-1998). Her research interests include multidimensional signal processing, pattern recognition, modeling, forecasting and human perception.

In 2001 Dr. Mojsilović has received the Young Author Best Paper Award from the IEEE Signal Processing Society. She is a member of the IEEE Multidimensional Signal Processing Technical Committee and an Associate Editor for the IEEE Transactions on Image Processing.