CTex—An Adaptive Unsupervised Segmentation Algorithm Based on Color-Texture Coherence

Dana E. Ilea and Paul F. Whelan, *Senior Member, IEEE*

***Abstract*—This paper presents the development of an unsupervised image segmentation framework (referred to as CTex) that is based on the adaptive inclusion of color and texture in the process of data partition. An important contribution of this work consists of a new formulation for the extraction of color features that evaluates the input image in a multispace color representation. To achieve this, we have used the opponent characteristics of the RGB and YIQ color spaces where the key component was the inclusion of the Self Organizing Map (SOM) network in the computation of the dominant colors and estimation of the optimal number of clusters in the image. The texture features are computed using a multichannel texture decomposition scheme based on Gabor filtering. The major contribution of this work resides in the adaptive integration of the color and texture features in a compound mathematical descriptor with the aim of identifying the homogenous regions in the image. This integration is performed by a novel adaptive clustering algorithm that enforces the spatial continuity during the data assignment process. A comprehensive qualitative and quantitative performance evaluation has been carried out and the experimental results indicate that the proposed technique is accurate in capturing the color and texture characteristics when applied to complex natural images.**

***Index Terms*—Adaptive Spatial K-Means clustering, color-texture segmentation, multichannel texture decomposition, multispace color segmentation, SOM classification.**

I. Introduction

IMAGE segmentation is one of the most investigated subjects in the field of computer vision since it plays a crucial role in the development of high-level image analysis tasks such as object recognition and scene understanding. A review of the literature on image segmentation indicates that a significant amount of research has been dedicated to the development of algorithms where the color and texture features where analyzed alone [1]–[4]. The early color-texture segmentation algorithms were designed in conjunction with particular applications [5], [6] and they were generally restricted to the segmentation of images that are composed of scenes defined by regions with uniform characteristics. Segmentation of natural images is by far a more difficult task, since natural images exhibit significant inhomogeneities in color and texture [7], [8]. Thus, the complex characteristics associated with natural images forced researchers to approach their segmentation using features that locally sample both the color and texture attributes [9], [10]. The use of color and texture information collectively has strong links with the human perception, but the main challenge is the combination of these fundamental image attributes in a coherent color-texture image descriptor [11]–[16]. In fact, if we take into consideration that the textures that are present in natural images are often characterized by a high degree of complexity, randomness, and irregularity, the simple inclusion of color and texture is not sufficient and a more appropriate segmentation model would encompass three attributes such as color, texture, and composite elements that are defined by a large mixture of colors. Although this model is intuitive, there is no generally accepted methodology to include the texture and color information in the segmentation process.

In this regard, Mirmehdi and Petrou [17] approached the segmentation of color images from a perceptual point of view. In their paper, they calculate a multiscale perceptual image tower that is generated by mimicking a human observer when looking at the input image from different distances. The first stage of their algorithm deals with the extraction of the core color clusters and the segmentation task is defined as a probabilistic process that reassigns the noncore pixels hierarchically starting from the coarsest image in the tower to the image with the highest resolution. The main limitation of this algorithm is the fact that the color and texture features are not explicitly used and this causes problems in the analysis of their contribution in the overall segmentation process. Deng and Manjunath [18] proposed a different color-texture segmentation technique (also known as JSEG) that consists of two independent computational stages: color quantization and spatial segmentation. During the first stage the color information from the input image is quantized into a representative number of classes without enforcing the spatial relationship between pixels. The aim of this process is to map the image into a structure where each pixel is assigned a class label. The next stage of the algorithm enforces the spatial composition of the class labels using a segmentation criterion (J value) that samples the local homogeneity. The main merit of this paper is the use of color and texture information in succession and the authors argue that this approach is beneficial since it is difficult to analyze the color similarity and spatial relationship between the neighboring pixels at the same time. Other researchers adopted different strategies regarding the inclusion of texture and color in the segmentation process. Tan and Kittler [19] developed an image segmentation algorithm that is composed of two channels, one for texture representation and one for color description. The texture information is extracted by applying a local linear transform, while the color is sampled by the moments calculated from the

The authors are with the Vision Systems Group, School of Electronic Engineering, Dublin City University, Dublin, Ireland (e-mail: danailea@eeng.dcu.ie; paul.whelan@dcu.ie).

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

Digital Object Identifier 10.1109/TIP.2008.2001047

color histogram. This approach is extremely appealing, since the contribution of color and texture can be easily quantified in the segmentation process and it has been adopted by a large number of researchers [20]–[22]. Another related implementation has been proposed by Carson *et al* [23]. They developed a Blobworld technique that has been applied to the segmentation of natural images in perceptual regions. The central part of this algorithm is represented by the inclusion of the polarity, contrast and anisotropy features in a multiscale texture model that is evaluated in parallel with the color information that is sampled by the CIE Lab color components. The main advantage of the Blobworld algorithm consists in its ability to segment the image in compact regions and it has been included in the development of a content-based image retrieval system.

In this paper, we propose a flexible and generic framework (CTex) for segmentation of natural images based on color and texture. The developed approach extracts the color features using a multispace adaptive clustering algorithm, while the texture features are calculated using a multichannel texture decomposition scheme. Our segmentation algorithm is unsupervised and its main advantage resides in the fact that the color and texture are included in an adaptive fashion with respect to the image content. This paper is organized as follows. In Section II, an outline of the developed algorithm is provided and each component is briefly described. In Section III, the Gradient Boosted Forward and Backward (GB-FAB) anisotropic diffusion filtering is presented and discussed. In Section IV, the color extraction algorithm is detailed, while in Section V, the texture extraction method is presented. Section VI highlights the mathematical framework behind the integration of these two fundamental image attributes: color and texture, while, in Section VII, the experimental results are presented and analyzed. Section VIII concludes the paper.

II. Overview of the Developed color-Texture Segmentation Algorithm

The main components of the proposed segmentation framework are depicted in Fig. 1. The color and texture features are extracted independently, on two different channels. The color segmentation algorithm is the most sophisticated part of the CTex framework and involves a statistical analysis of the input image in the RGB and YIQ color representations. The original image (RGB color space) is first prefiltered with a GB-FAB anisotropic diffusion algorithm [24] that is applied to eliminate the image noise, artefacts and weak textures and improve the local color coherence. The second step extracts the dominant colors (initial seeds) from the filtered image and calculates the optimal number of clusters (k) using an unsupervised classification procedure based on the Self Organizing Maps (SOM). Next, the filtered RGB image is clustered using a K-Means algorithm where the cluster centers are initialized with the dominant colors and the number of clusters (k) calculated during the previous step. As illustrated in Fig. 1, the second stream of the color segmentation algorithm converts the original input image into the YIQ color representation that will be further subjected to similar procedures as the RGB image. The filtered YIQ image is clustered with a K-Means algorithm where the initial cluster centers are initialized with the dominant YIQ colors and the number of clusters is set to k (that has been calculated from the RGB data using the SOM procedure). The clustered RGB and YIQ images are concatenated to generate an intermediate image that will be further subjected to a 6-D multispace K-Means clustering that outputs the final color segmented image. The second major component of the CTex framework involves the extraction of the texture features from the original image over the entire spectrum of frequencies with a bank of Gabor filters calculated for different scales and orientations. The resulting color segmented image, texture images (the number of texture images is given by the number of scales and orientations of the Gabor filter bank) and the final number of clusters k are the inputs of the novel Adaptive Spatial K-Means clustering (ASKM) framework, which returns the final color-texture segmented image.

III. GB-FAB Anisotropic Filtering

The adaptive filtering technique that has been implemented for preprocessing the input image is an improvement of the forward and backward anisotropic diffusion (also called FAB [25], [27]). The FAB anisotropic diffusion is a nonlinear feature preserving smoothing technique that efficiently eliminates the image noise and weak textures from the image while preserving the edge information.

The original anisotropic diffusion filtering (PM) has been proposed by Perona and Malik [26] where they formulated the smoothing as a diffusive process that is performed within the image regions and suppressed at the regions boundaries. In order to achieve this behavior, they developed a mathematical framework where the central part is played by a diffusion function that controls the level of smoothing

$$\frac{\partial I(x,y,t)}{\partial t} = div[D(|\nabla I(x,y,t)|)\nabla I(x,y,t)] \qquad (1)$$

where $I(x,y)$ is the image data, $\nabla I\,(x,y,t)$ is the gradient operator at the position (x,y) at iteration t, $D(\cdot)$ represents the diffusion function and *div* is the divergence operator. The function D is usually implemented using an exponential function as illustrated in (2), where the parameter d controls the level of smoothing

$$D(|\nabla I(x,y,t)|) = e^{-(|\nabla I(x,y,t)|/d)^2}, d > 0. \qquad (2)$$

It can be observed that the diffusion function $D(\cdot)$ is bounded in the interval $(0,1]$ and decays with the increase of the gradient value ∇I. The PM filtering is an efficient feature preserving smoothing strategy, but it has stability problems caused by the offsets between the input and output image. Another problem associated with the standard PM filtering is that the diffusion function D acts aggressively upon medium gradients [see Fig. 2(a)] which results in the attenuation of medium edges in the smoothed image. To eliminate the limitations associated with the original PM formulation, the FAB anisotropic diffusion has been proposed [25]. The goal of the FAB diffusion function is to highlight the medium and large gradients that are noise independent and this is achieved by reversing the diffusion process. This can be implemented by applying two diffusions simultaneously: the forward diffusion that acts upon the low gradients that are usually caused by noise, while the backward diffusion is applied to reverse the diffusion process when dealing

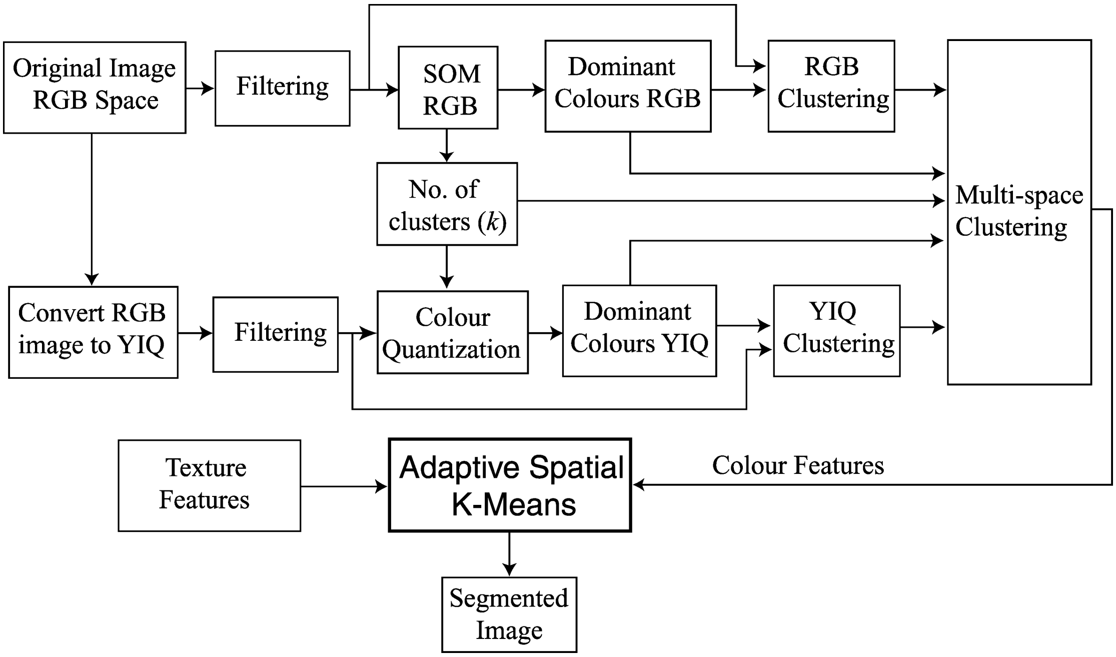
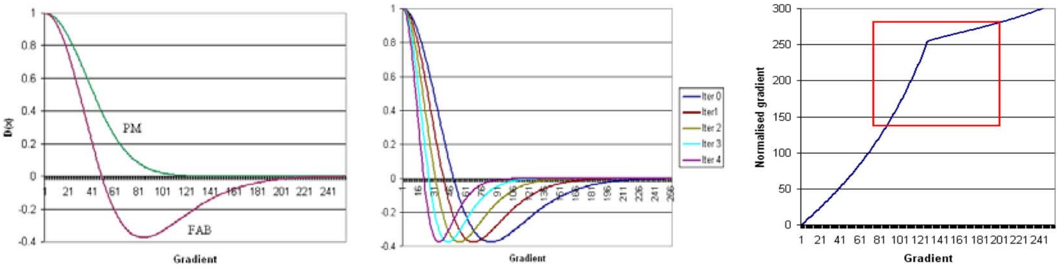


Fig. 1. Outline of the proposed CTex color-texture image segmentation framework.



(a) (b) (c)

Fig. 2. (a) Comparison between the FAB diffusion function and the standard PM diffusion function. (b) The effect of the cooling process. Note that the position where the curve intersects the x-axis is lowered at each iteration and this implies less smoothing. (c) Gradient boosting function. Note the amplification of the gradients with medium values—marked in the red box.

with medium gradients. This can be observed in Fig. 2(a) where it is shown that the FAB diffusion function [defined in (3)] becomes negative for medium values of the gradient. Nonetheless, since the D_{FAB} function is defined by two parameters the problems associated with stability are more difficult to control. To address these problems, Smolka and Plataniotis [27] proposed the inclusion of a time dependent cooling procedure [see Fig. 2(b)] where the values of the diffusion parameters are progressively reduced with the increase in the number of iterations [see (4)]

$$D_{FAB}(|\nabla I(x,y,t)|) = 2e^{-(|\nabla I(x,y,t)|/d_1(t))^2} - e^{-(|\nabla I(x,y,t)|/d_2(t))^2} \quad (3)$$

$$d_i(t+1) = d_i(t) \cdot \gamma, i = 1,2$$
$$d_i(t+1) < d_i(t), \gamma \in (0,1] \quad (4)$$

where $d_1(t=0)$ is the starting parameter, ∇I represents the image gradient, γ is a fixed parameter that takes values in the interval (0,1], $d_1(t)$ and $d_2(t)$ are the time dependent parameters that control the forward and backward diffusion respectively and t is the time or iteration step. In our implementation we set these parameters to the following default values: $d_1(t=0) = 40$, $d_2(t=0) = 2d_1(t=0) = 80$, and $\gamma = 0.8$.

While the FAB anisotropic diffusion eliminates some of the problems associated with the standard PM filtering strategy, the experimental results show that the smoothed data is still blurred especially around regions defined by medium gradients. To further reduce the level of blurriness, we proposed in [24] the inclusion of a boosting function [see (5)] to amplify the medium gradients [see Fig. 2(c)] and obtain much crisper image details. The gradient value in (3) will be replaced by the new "boosted" value

$$|\nabla I(x,y,t)| \leftarrow |\nabla I(x,y,t)| \left(1 + 2e^{-||\nabla I(x,y,t)|-m|/d_1(t)}\right) \quad (5)$$

where m is the median value of the gradient data. Fig. 3 illustrates the performance of the PM and GB-FAB anisotropic diffusion schemes when applied to a natural image defined by a high level of detail. Fig. 4 depicts the results obtained after the application of the color segmentation algorithm (full details are

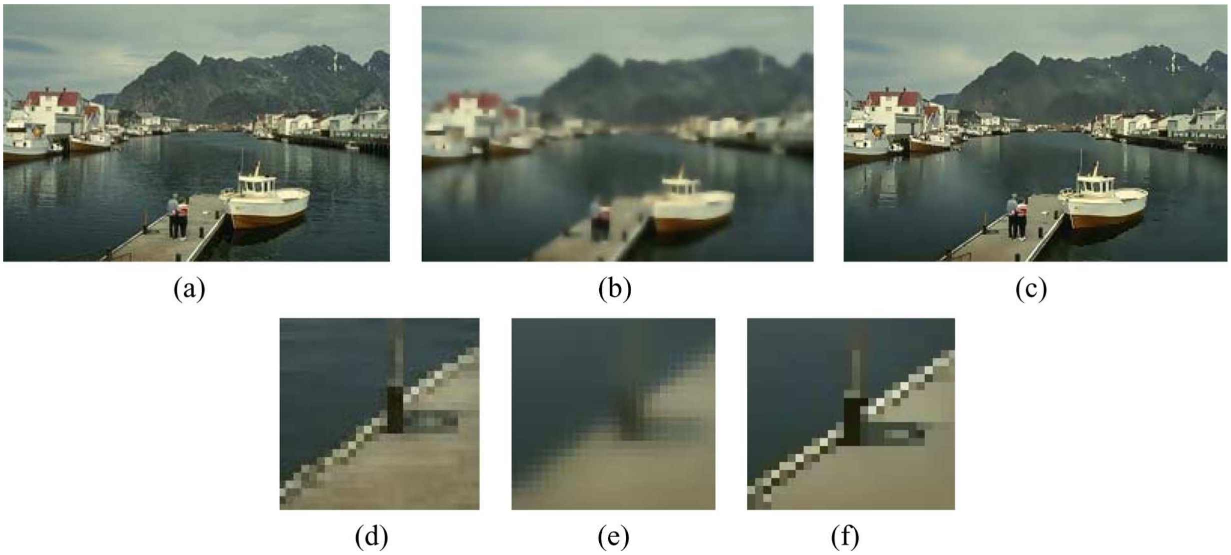


Fig. 3. (a) Original natural image. (b) PM filtered image $(d = 40)$. (c) GB-FAB filtered image $(d_1(t = 0) = 40, d_2(t = 0) = 80)$. (d) Close-up detail from the original image. (e) Close-up detail from the PM filtered image. (f) Close-up detail from the GB-FAB filtered image.

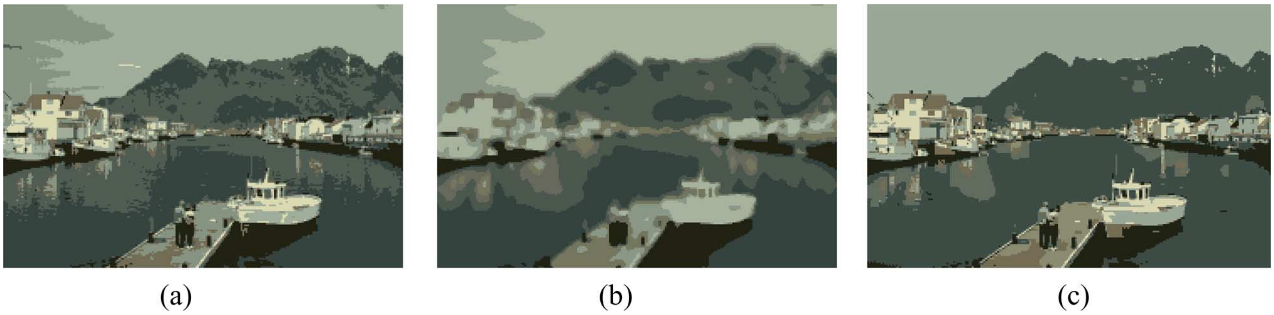


Fig. 4. Comparison results when the original image depicted in Fig. 3(a) has been subjected to color segmentation under the following preprocessing conditions: (a) no filtering, (b) PM filtering, (c) proposed GB-FAB filtering.

provided in Section IV) in conditions when the input image is subjected to no filtering, PM and GB-FAB anisotropic diffusion. In Fig. 4(a), it can be observed that a high level of over-segmentation is obtained when the image is not prefiltered. When the image is subjected to anisotropic diffusion we can conclude that the PM algorithm is not efficient in preserving the level of detail and the outlines of the objects in the image [see Fig. 4(b)] and significantly improved results are obtained when the input image is filtered with the proposed GB-FAB algorithm. For further details about the implementation and performance characterization of the GB-FAB adaptive filtering scheme, the reader may refer to [24].

IV. Color Segmentation Algorithm

A. *Dominant Colors Extraction. Automatic Detection of the Number of Clusters*

This section details the procedure employed to automatically determine the dominant colors and the optimal number of clusters from the filtered input image. For most space partitioning algorithms, the cluster centers are initialized either using a starting condition specified *a priori* by the user or by applying a random procedure that selects the cluster centers from the input data. The random selection proved to be inappropriate since this forces the clustering algorithms to converge to local minima [28]. Another parameter that has to be specified *a priori* is the final number of clusters. The performance of the clustering algorithms is highly influenced by the selection of this parameter and in this paper we propose an efficient solution to automatically detect the dominant colors and the final number of clusters in the image using a classification procedure based on the Self Organizing Maps (SOM). Using the SOM, we train a set of input vectors in order to obtain a lower dimensional representation of the input image in the form of a feature map that maintains the topological relationship and metric within the training set. The SOM networks were first introduced by Kohonen [29] and they became popular due to their ability to learn the classification of a training set without any external supervision. In our implementation, we created a 2-D SOM network that is composed of nodes or cells [see Fig. 5(a)]. Each node N_i ($i \in [1, M]$ where M is the number of nodes in the network) has assigned a 3-D weight vector (w_i) that matches the size of each element of the input vector. It is important to mention that the training dataset represented by the input image is organized as a 1-D vector V_j ($j = 1 \ldots n$, where n is the total number of pixels in the image) in a raster scan manner. Each

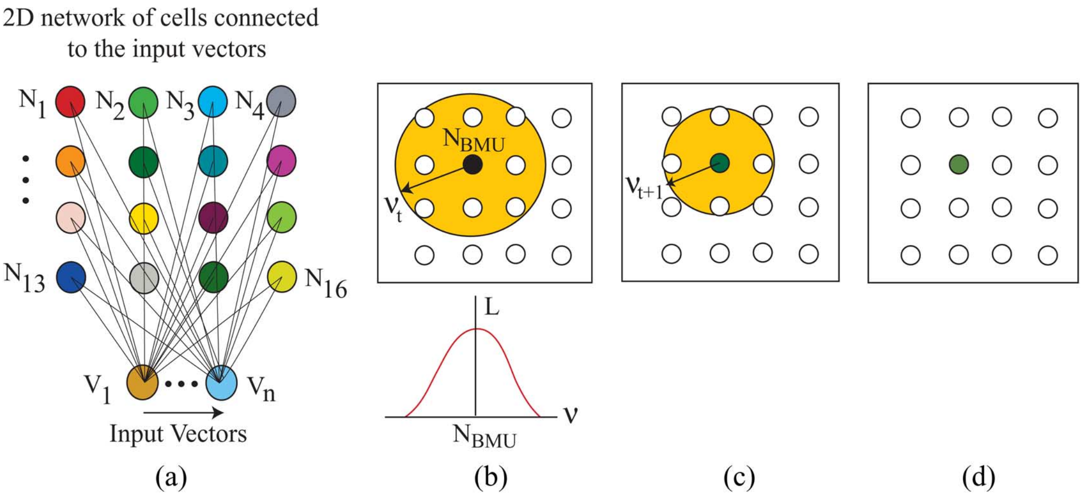


Fig. 5. (a) A 2-D SOM network. (b) The neighborhood of $\boldsymbol{N}_{\mathrm{BMU}}$ at iteration t. The learning process of each cell's weight follows a Gaussian function, i.e., it is stronger for cells near node N_{BMU} and weaker for distant cells. (c), (d) The radius $\nu(t)$ is progressively reduced until it reaches the size of one cell (N_{BMU}).

element of the training set V_j is defined by a 3-D vector whose components are the normalized R, G, B values of the pixels in the image and is connected to all the cells of the network [see Fig. 5(a)].

In line with other clustering schemes, before starting the training procedure we need to initialize the weights w_i for all cells in the network. In practice, the random initialization is usually adopted when working with SOM networks [30] and this is motivated by the fact that after several hundreds of iterations, the corresponding values of the initial random weights will change in accordance to the color content of the image. This procedure has been applied in [31] where the authors initialized the SOM network by randomly picking color samples from the input image. However, the random selection of the starting condition is sub-optimal since the algorithm can be initialized on outliers. Therefore, we propose to initialize the weights of the nodes in the SOM network with the dominant colors that are represented by the peaks (P_i) in the 3-D color histogram, calculated from the image that has been subjected to color quantization. This is achieved by applying a color quantization procedure that consists of re-sampling linearly the number of colors on each color axis. It has been experimentally demonstrated [32] that a quantization value of 8 is sufficient to sample the statistical relevant peaks in the 3-D histogram. Thus, the quantized version of the input image is re-mapped so that the initial number of grey levels in all color bands 256 × 256 × 256 is now reduced to 8 × 8 × 8. After constructing the 3-D histogram in the quantized color space, the peaks P_i in relation to the desired number of dominant colors are selected by applying a quicksort algorithm. Considering that the size of the SOM lattice is four by four (i.e., $M = 16$ cells), the first 16 highest histogram peaks are sufficient to accurately sample the dominant colors in the image.

Once the initialization is completed $(w_i \leftarrow P_i, i \in [1, 16])$, the classification procedure is iteratively applied and consists of assigning the input vectors to the cell in the network whose corresponding weight values are most similar. The node in the SOM network that returns the smallest Euclidean distance is declared the best matching unit (BMU)

$$\mathrm{BMU} = \underset{i\in[1,16]}{\arg\min} \|V_j - w_i\| \quad j \in [1, n]. \tag{6}$$

The weights of the N_{BMU} and of the nodes situated in its neighborhood are updated using the following learning rule:

$$\begin{aligned} w_i(t+1) &= w_i(t) + L(t)[V_j(t) - w_i(t)] \\ \text{if } \|N_{\mathrm{BMU}} - N_i\| &\le \nu(t) \\ w_i(t+1) &= w_i(t), \text{if } \|N_{\mathrm{BMU}} - N_i\| > \nu(t). \end{aligned} \tag{7}$$

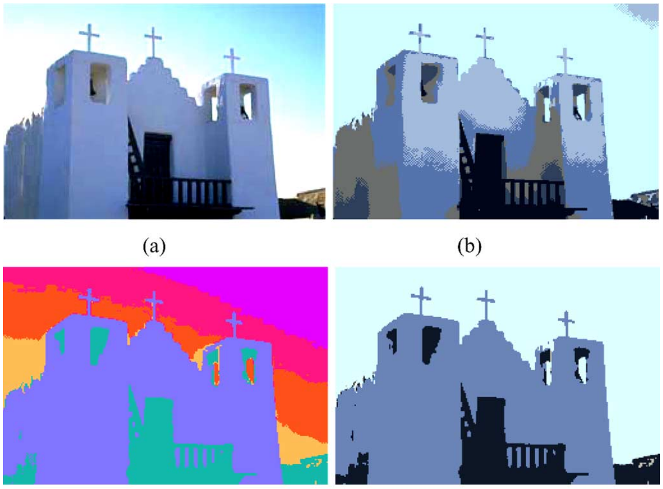
In (7), t is the iteration step, $\nu(t)$ is the neighborhood radius and $L(t)$ is the learning rate. The size of the radius $\nu(t)$ and the strength of the learning rate $L(t)$ are exponentially reduced with the increase in the number of iterations [see Fig. 5(b)–(d)]. The SOM algorithm is iterated until convergence (radius ν reaches the size of N_{BMU}) and the final weights of the 2-D network are the dominant colors of the input image.

B. Selection of the Optimal Number of Clusters

To obtain the optimal number of clusters, we propose a multistep technique that progressively reduces the number of dominant colors resulting after the SOM classification procedure. In the first step, the pixels in the image are mapped to the final weights of the cells in the SOM network based on the minimum Euclidean distance [see (8)]

$$V_j \leftarrow w_g, g = \underset{i\in[1,16]}{\arg\min} \|V_j - w_i\| \quad j \in [1, n]. \tag{8}$$

The resulting color map can be viewed as a preliminary clustering of the input image. In the second step, a confidence map is constructed where the cumulative smallest distances between



(a)

(b)

(c)

(d)

Fig. 6. (a) Original image. (b) The clustered image in the RGB color space (the number of clusters determined using the SOM procedure is $k = 6$). (c) The clustered image in the YIQ color space ($k = 6$). (d) The final multispace color segmentation result. The final number of clusters is 4.

TABLE I

CONFIDENCE MAP CORRESPONDING TO IMAGE IN FIG. 6(A)

Seed	*Confidence value*	*No. of samples*
c_1	0.072752	17119
c_2	0.099610	14333
c_3	0.077165	18651
c_4	0.103173	2172
c_5	0.070406	17442
c_6	0.081271	16448
c_7	0.075945	17361
c_8	0.201127	2384
c_9	0.066440	16668
c_{10}	0.097796	15595
c_{11}	0.137216	16942
c_{12}	0.105167	19678
c_{13}	0.067716	15011
c_{14}	0.119089	17125
c_{15}	0.188415	21717
c_{16}	0.130097	9791

the weights of the SOM network and pixels in the image are recorded [see (9)]. For all pixels V_j labelled with w_i, we define

$$\text{confidence}(w_i) = \frac{\sum\limits_{j \in D_{w_i}} \|V_j - w_i\|}{\text{no\_pixels\_labelled}(w_i)} \qquad i \in [1, 16] \tag{9}$$

where D_{w_i} is the image domain defined by the pixels V_j that are closest to the weights w_i. The confidence map returns a weighted measure between the variance within the cluster and the number of pixels in the cluster. The lower its value is, the more reliable the estimate w_i is. The confidence map calculated for the example depicted in Fig. 6(a) is shown in Table I. The last step determines the final number of clusters by evaluating the intercluster variability. To achieve this, we construct the similarity matrix where the Euclidean distances between the weights of any neighboring nodes in the SOM network are stored.

If this distance is smaller than a predefined intercluster threshold, then the node that has the highest confidence value is eliminated. This process is iteratively repeated until the distance between the weights of all adjacent nodes in the SOM network is higher than the predefined threshold value. In our implementation, we set the intercluster threshold to 0.3 and this value proved to be optimal for all images analyzed in our study.

C. Multispace Color Segmentation

RGB is a perceptually nonuniform color space and one of its limitations is the fact that the chrominance and intensity components are not explicitly defined. In order to overcome this drawback, the second stream of the color segmentation algorithm extracts additional color features from the YIQ representation of the image. We have adopted this approach in order to exploit the opponent characteristics of the RGB and YIQ color spaces. As mentioned earlier, in the YIQ color representation, the chrominance components (I and Q) are separated from the luminance component (Y) and as a result the shadows and local inhomogeneities are generally better modeled than in the RGB color space. Colors with high degrees of similarity in the RGB space may be difficult to distinguish, while the YIQ representation may provide a much stronger discrimination.

The YIQ image goes through similar operations as the previously analyzed RGB image. Initially, it is filtered using the GB-FAB anisotropic diffusion detailed in Section III and then it is further processed using a K-Means clustering algorithm. The key issue in the extraction of the color features from the YIQ image is the fact that the parameter k that selects the number of clusters for the K-Means algorithm is set to the same value that has been obtained after the application of the SOM procedure to the image represented in the RGB color space (see Section IV-A and B). Thus, the parameter k performs the synchronization between the RGB and YIQ channels by forcing the K-Means algorithms applied to the RGB and YIQ images (see Fig. 1) to return the same number of clusters. The dominant colors from the YIQ image that are used to initialize the initial clusters for K-Means algorithm are determined using the same procedure based on color quantization that has been applied to initialize the weights of the SOM network (see Section IV-A).

The next step of the color extraction algorithm consists in the concatenation of the color features calculated from the RGB and YIQ images where each pixel in the image is defined by a 6-D vector whose components are the R, G, B, Y, I, Q values of the clustered RGB and YIQ images. In the final step, the RGB-YIQ data is clustered with a 6-D K-Means algorithm where the number of clusters is again set to k and the cluster centers are initialized with the dominant colors that were used to initialize the K-Means algorithms that have been applied to cluster the RGB and YIQ images. Fig. 6 illustrates the performance of the developed multispace color segmentation algorithm when compared to the results obtained when the input image is analyzed in the RGB and YIQ color representations.

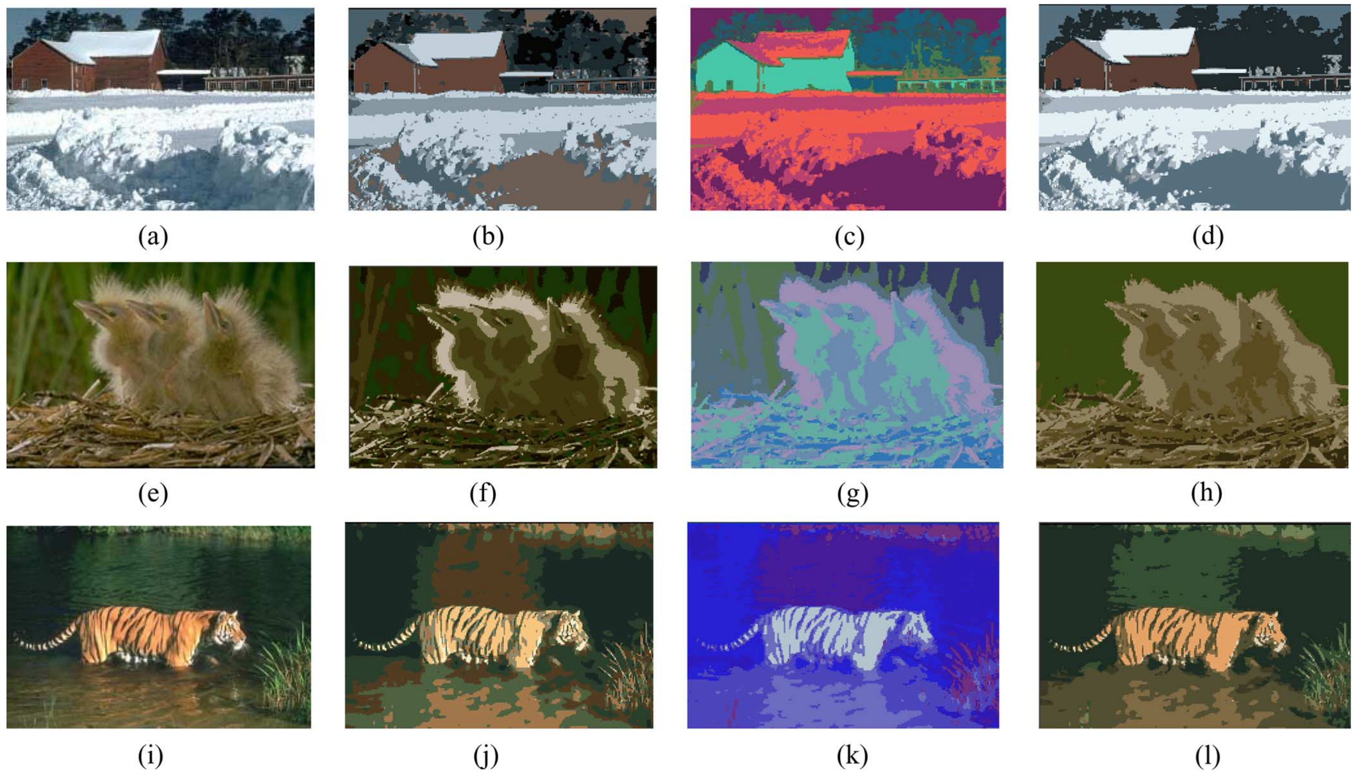


Fig. 7. Color segmentation results. (a), (e), and (i): Original natural images showing complex color-texture characteristics. (b), (f), and (j): RGB clustered images. (c), (g), and (k): YIQ clustered images. (d), (h), and (l): Multispace color segmentation results. The final number of clusters in images (d), (h), and (l) are 8, 5, and 9, respectively.

It is important to note that, during the multispace partitioning process, some clusters from the initial set may disappear as the clusters become more compact with the increase in the number of iterations (this is achieved by applying a cluster merging procedure that re-labels the adjacent clusters whose centers are close in the RGB-YIQ representation).

At this stage, we need to address why we have adopted the approach to analyze the RGB and YIQ images in succession and then fusing the results from the K-means algorithms using multidimensional clustering rather than clustering directly the RGB-YIQ data. Our approach is motivated by the fact that the initialization of the 6-D SOM network using the procedure based on color quantization is unreliable while the 6-D histogram calculated from the RGB-YIQ data is sparse and the peaks are not statistically relevant. Our approach circumvents this issue while it attempts to find the optimal result by fusing the clustered RGB and YIQ images which have a reduced dimensionality that is sampled by the parameter k (as opposed to the high dimensionality of the original RGB-YIQ data).

Additional color segmentation results are illustrated in Fig. 7 when our color segmentation algorithm has been applied to three natural images with inhomogeneous color-texture characteristics. It can be noted that the shapes of the objects follow the real boundaries present in the original image and the small and narrow objects are not suppressed during the color segmentation process [see the background fence in Fig. 7(d) and the birds' eyes in Fig. 7(h)]. The segmentation results illustrated in Fig. 7 indicate that the color information alone is not sufficient to describe the regions characterized by complex textures, such as the straws in Fig. 7(h) or the tiger's fur in Fig. 7(l). Therefore, we propose to complement the color segmentation with texture features that are extracted using a texture decomposition technique based on Gabor filtering.

V. Texture Extraction

There has been a widely accepted consensus among vision researchers that filtering an image with a large number of oriented band pass filters such as Gabor represents an optimal approach to analyze textures [33]–[35]. Our approach implements a multichannel texture decomposition and is achieved by filtering the input textured image with a 2-D Gabor filter bank that was initially suggested by Daugman [36] and later applied to texture segmentation by Jain and Farrokhnia [34]. The 2-D Gabor function that is used to implement the even-symmetric 2-D discrete filters can be written as

$$G_{\sigma,f,\varphi}(x,y) = \exp\left(-\frac{x'^2+y'^2}{2\sigma^2}\right)\cos\left(2\pi f x\prime + \varphi\right) \quad (10)$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. In (10), the parameter σ represents the scale of the Gabor filter, θ is the orientation and f is the frequency parameter that controls the number of cycles of the cosine function within the envelope of the 2-D Gaussian (ϕ is the phase offset and it is usually set to zero to implement 2-D even-symmetric filters). The Gabor filters are band pass filters where the parameters σ, θ, and f determine the sub-band that is covered by the Gabor filter in the spatial-frequency domain. The parameters of the Gabor filters are chosen to optimize the trade-off between spectral selectivity

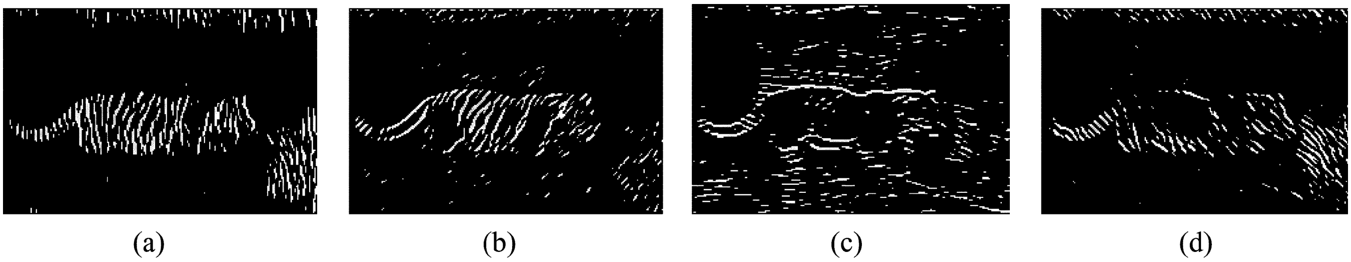


Fig. 8. Texture features extracted for the natural image depicted in Fig. 7(i) using Gabor filter with four orientations: (a) 0°, (b) 45°, (c) 90°, and (d) 135°.

and the size of the bank of filters. Typically, the central frequencies are selected to be one octave apart and for each central frequency is constructed a set of filters corresponding to four (0°, 45 °, 90°, 135°) or six orientations (0°, 30°, 60°, 90°, 120°, 150°). Fig. 8 shows the textures features extracted from a natural image, when the Gabor filters are calculated using four orientations.

VI. Color and Texture Integration

We propose to integrate the color and texture features using a spatially adaptive clustering algorithm. The inclusion of the texture and color features in an adaptive fashion is a difficult task [37], [38] since these attributes are not constant within the image. Thus, the application of standard clustering techniques to complex data such as natural images will lead to over-segmented results since the spatial continuity is not enforced during the space partitioning process. In this paper, our aim is to develop a space-partitioning algorithm that is able to return meaningful results even when applied to complex natural scenes that exhibit large variations in color and texture. To achieve this we propose a new clustering strategy called ASKM whose implementation can be viewed as a generalization of the K-Means algorithm. The ASKM technique attempts to minimize the errors in the assignment of the data-points into clusters by sampling adaptively the local texture continuity and the local color smoothness in the image. The inputs of the ASKM algorithm are: the color segmented image, the texture images and the final number of clusters k that has been established using the SOM based procedure (see Section IV). The main idea behind ASKM is to minimize an objective function J based on the fitting between the local color and local texture distributions calculated for each data-point (pixel) in the image and the color and texture distributions calculated for each cluster. This approach is motivated by the fact that the color-texture distribution enforces the spatial continuity in the data partitioning process since the color and texture information are evaluated in a local neighborhood for all pixels in the image. The local color distribution for the data-point at location (x,y) is calculated as follows:

$$H_C^{s\times s}(x,y) = \bigcup_{b\in[1,k]} h_C^{s\times s}(x,y,b), \text{ where}$$
$$h_C^{s\times s}(x,y,b) = \sum_{p=(x-s/2)}^{x+s/2} \sum_{q=(y-s/2)}^{y+s/2} \delta\left(C\left(p,q\right),b\right) \text{ and}$$
$$\delta\left(i,j\right) = \begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases} \tag{11}$$

where $H_C^{s\times s}(x,y)$ is the local color distribution calculated from the color segmented image C in the neighborhood of size $s\times s$ around the data-point at position (x,y) and k is the number of clusters. In (11), the union operator $\cup$ defines the concatenation of the individual histogram bins $h_C^{s\times s}(x,y,b), b\in[1,k]$ that are calculated from the color segmented image C.

The local texture distribution $H_T^{s\times s}(x,y)$ is obtained by concatenating the distributions $H_{T_j}^{s\times s}(x,y)$ as follows:

$$\begin{aligned} H_{T_j}^{s\times s}(x,y) &= \bigcup_{b\in[0,255]} h_{T_j}^{s\times s}(x,y,b) \\ h_{T_j}^{s\times s}(x,y,b) &= \sum_{p=(x-s/2)}^{x+s/2} \sum_{q=(y-s/2)}^{y+s/2} \delta\left[T_j\left(p,q\right),b\right] \\ j &\in [1,\alpha] \end{aligned} \tag{12}$$
$$\begin{aligned} H_T^{s\times s}(x,y) &= \bigcup_{j\in[1,\alpha]} H_{T_j}^{s\times s}(x,y) \\ &= \left[H_{T_1}^{s\times s}, H_{T_2}^{s\times s}, \ldots, H_{T_\alpha}^{s\times s}\right] \end{aligned} \tag{13}$$

where T_j is the jth Gabor filtered image and α is the total number of texture orientations. In our implementation the pixel values of the texture images T_j are normalized in the range $[0,255]$.

In order to accommodate the color-texture distributions in the clustering process, we replaced the global objective function of the standard K-Means algorithm with the formulation shown in (14). The aim of the ASKM algorithm is the minimization of the objective function J that is composed of two distinct terms that impose the local coherence constraints. The first term optimizes the fitting between the local color distribution for the data point under analysis and the global color distribution of each cluster, while the second term optimizes the fitting between the local texture distribution for the same data point with the global texture distribution of each cluster [see (14), shown at the bottom of the next page].

In (14), k is the number of clusters, $s\times s$ defines the size of the local window, $H_C^{s\times s}(x,y)$ and $H_T^{s\times s}(x,y)$ are the local color and the local texture distributions calculated for the pixel at position (x,y) respectively, H_C^i and H_T^i are the color and texture distributions for the cluster with index i respectively. The similarity between the local color-texture distributions and the global color-texture distributions of the clusters is evaluated using the Kolmogorov–Smirnov (KS) metric

$$\mathrm{KS}\left(H_a,H_b\right) = \sum_{i\in[0,\mathrm{hist\_size}]} \left| \frac{h_a(i)}{n_a} - \frac{h_b(i)}{n_b} \right| \tag{15}$$

where n_a and n_b are the number of data points in the distributions H_a and H_b, respectively. The main advantage of the KS metric over other similarity metrics such as G-statistic and the Kullback divergence [39] is the fact that the KS metric is normalized and the result of the comparison between the distributions H_a and H_b is bounded in the interval [0, 2]. The fitting between the local color-texture distributions and global color-texture distributions of the clusters is performed adaptively for multiple window sizes in the interval $[3 \times 3]$ to $[25 \times 25]$. The evaluation of the fitting between the local and global distributions using a multiresolution approach is motivated by the fact that the color composition of the texture in the image is not constant and the algorithm adjusts the window size until it is achieved the best fit value. It is important to note that the global color-texture distributions H_C^i and H_T^i are updated after each iteration and the algorithm is executed until convergence.

VII. Experiments and Results

A large number of experiments were carried out to assess the performance of the proposed color-texture segmentation framework. These tests were conducted on synthetic and natural image datasets and the results were quantitatively and qualitatively evaluated. The first tests were performed on a dataset of 33 mosaic images and the segmentation results were evaluated by analyzing the errors obtained by computing the displacement between the border pixels of the segmented regions and the region borders in the ground truth data. The second set of experiments was performed on the Berkeley database [41] that is composed of natural images characterized by various degrees of complexity with respect to color and texture information. The purpose of this investigation was to obtain a comprehensive quantitative performance evaluation of our algorithm with respect to the correct identification of perceptual regions in the image and the level of image detail. In order to illustrate the validity of the proposed scheme, we have compared the results returned by the CTex algorithm against those returned by the well-established JSEG color-texture segmentation algorithm developed by Deng and Manjunath [18]. For the CTex algorithm, the parameters required by the anisotropic diffusion and color segmentation method are discussed in Sections III and IV and are left to the default values in all experiments. The texture features are extracted using Gabor filters and the experiments were conducted using a filter bank that samples the texture in four orientations (0°, 45°, 90°, and 135°) and the scale and frequency parameters were set to $\sigma = 3.0$ and $f = 1.5/2\pi$, respectively. The JSEG algorithm is a standard color-texture segmentation benchmark and in all experiments we used the implementation made available online by the authors (http://vision.ece.ucsb.edu/segmentation/jseg/software/). JSEG involves three parameters that have to be specified by the user (the color quantization threshold, the scale and the merge threshold) and in our study we have set them to the values suggested by the authors (255, 1.0, and 0.4, respectively).

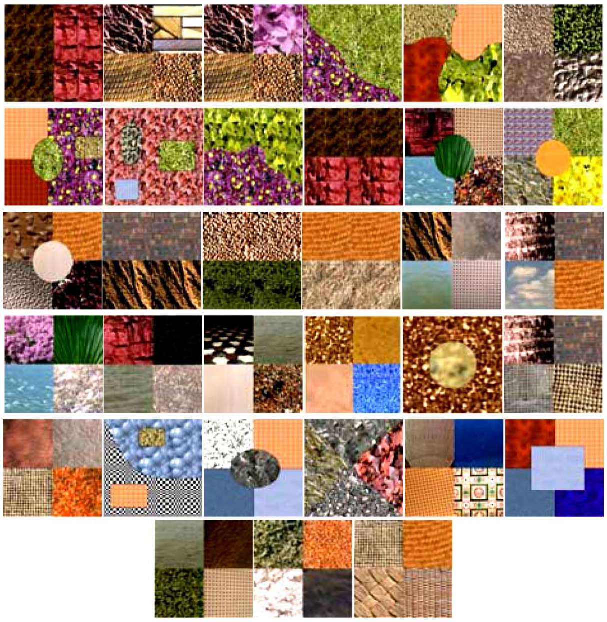


Fig. 9. Database of 33 mosaic images used in our experiments. These images are labelled from 01 to 33 starting from the upper left image in a raster scan manner.

A. Experiments Performed on Mosaic Images

Since the ground truth data associated with complex natural images is difficult to estimate and its extraction is highly influenced by the subjectivity of the human operator, we performed the first set of tests on synthetic data where the ground truth is unambiguous. Therefore, we executed the CTex and JSEG algorithms on a database of 33 mosaic images (image size 184×184) that were created by mixing textures from VisTex [40] and Photoshop databases. The mosaics used in our experiments consist of various texture arrangements that also include images where the borders between different regions are irregular. The suite of 33 mosaic images is depicted in Fig. 9.

The segmentation accuracy of the CTex and JSEG algorithms is estimated by calculating the Euclidean distances between the pixels situated on the border of the regions present in the segmented results and the border pixels present in the ground truth data. To evaluate the segmentation errors numerically, we calculate the mean, standard deviation and r.m.s errors that measure the border displacement between the ground truth and the segmented results. The experimental data is depicted in Table II and shows that the overall mean errors (shown in bold) calculated for

$$J = \sum_{x=1}^{\text{width}} \sum_{y=1}^{\text{height}} \left\{ \sum_{i=1}^{k} \left[\min_{s \in [3\times3,\ldots,25\times25]} KS\left(H_C^{s\times s}(x,y), H_C^i\right) + \min_{s \in [3\times3,\ldots,25\times25]} KS\left(H_T^{s\times s}(x,y), H_T^i\right) \right] \right\} \qquad (14)$$

TABLE II
POINT TO CURVE ERRORS BETWEEN THE GROUND TRUTH AND SEGMENTED RESULTS GENERATED BY THE CTEX AND JSEG ALGORITHMS. THE MEAN AND R.M.S ERRORS ARE GIVEN IN PIXELS

Image	CTex			JSEG		
	Mean	**St_dev**	**r.m.s**	**Mean**	**St_dev**	**r.m.s**
01	0.45	0.52	0.69	0.14	0.35	0.38
02	1.00	1.00	1.42	12.92	22.94	26.33
03	0.68	0.87	1.10	3.36	6.84	7.62
04	0.00	0.11	0.11	0.62	0.69	0.93
05	0.51	0.54	0.75	0.14	0.37	0.40
06	2.15	4.51	5.00	3.19	5.17	6.08
07	1.19	0.90	1.50	1.00	1.33	1.67
08	0.71	0.65	0.97	0.63	0.78	1.00
09	1.36	1.03	1.70	0.66	0.76	1.01
10	1.28	1.92	2.31	0.05	0.23	0.24
11	0.70	0.80	1.07	3.64	6.05	7.06
12	0.78	0.82	1.13	0.38	0.66	0.76
13	0.75	0.68	1.01	1.63	2.36	2.87
14	0.97	0.84	1.28	0.80	1.37	1.59
15	0.35	0.55	0.66	0.34	0.49	0.59
16	0.32	0.46	0.56	0.34	0.48	0.59
17	0.87	0.71	1.12	12.22	22.79	25.86
18	0.96	0.70	1.19	0.65	0.53	0.84
19	0.66	0.83	1.06	0.64	0.68	0.93
20	1.08	1.04	1.50	2.99	4.75	5.62
21	1.28	0.83	1.52	3.28	5.05	6.02
22	0.91	0.48	1.03	0.29	0.52	0.59
23	0.90	0.65	1.12	0.55	0.55	0.78
24	0.99	0.78	1.26	1.01	0.91	1.36
25	0.71	0.96	1.20	0.38	0.53	0.65
26	2.15	1.59	2.67	4.42	6.83	8.13
27	1.03	1.10	1.51	0.13	0.36	0.38
28	2.32	2.36	3.31	2.30	2.74	3.58
29	0.79	0.73	1.08	0.90	1.04	1.38
30	1.00	0.84	1.31	3.54	8.88	9.56
31	0.95	0.89	1.30	0.55	0.68	0.88
32	0.86	0.73	1.13	0.46	0.57	0.74
33	0.74	0.91	1.18	24.54	30.43	39.09
Overall	**0.95**	**0.97**	**1.38**	**2.68**	**4.20**	**5.01**

CTex are smaller than the overall mean errors calculated for the JSEG algorithm.

In our experiments, we noticed that the JSEG algorithm performs well in the identification of the image regions defined by similar color-texture properties, but it fails to determine accurately the object borders between the regions that are characterized by similar color compositions. This can be observed in Fig. 10, where the images where the JSEG produced the most inaccurate results are shown (images 02, 11, 17, and 33). These images are difficult to segment since they are defined by regions with inhomogeneous texture characteristics and the color contrast between them is low. Although the task to segment the images shown in Fig. 10 is very challenging, the experimental results (see Fig. 10 and Table II) indicate that the CTex algorithm is able to produce more consistent segmentation results where the errors in boundary location are significantly smaller than those generated by the JSEG algorithm.

B. *Experiments Performed on Natural Images*

We have tested the proposed CTex segmentation algorithm on a large number of complex natural images in order to evaluate its performance with respect to the identification of perceptual color-texture homogenous regions. To achieve this goal, we have applied our technique to Berkeley [41], McGill [42], and Outex [43] natural images databases that include images characterized by nonuniform textures, fuzzy borders, and low image contrast. The experiments were conducted to obtain a quantitative and qualitative evaluation of the performance of the CTex color-texture segmentation framework. In order to illustrate its validity, we have compared our segmentation results

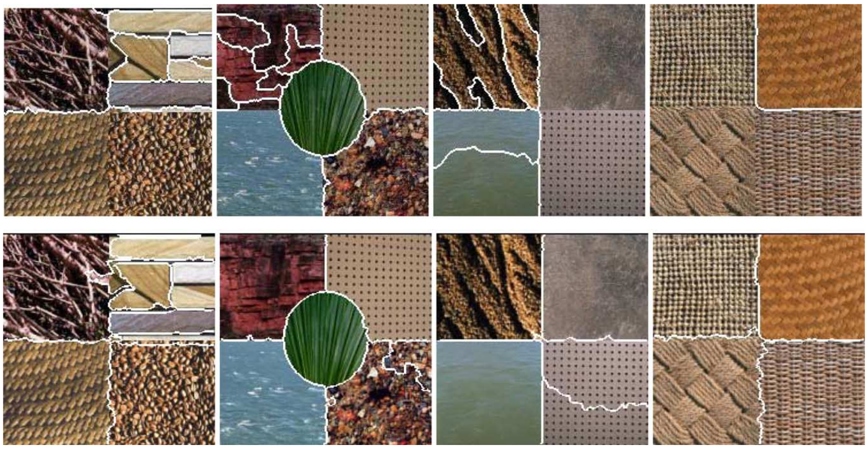


Fig. 10. Color texture segmentation results when JSEG and CTex algorithms where applied to images 02, 11, 17, and 33. First row: JSEG segmentation results. Second row: CTex segmentation results.

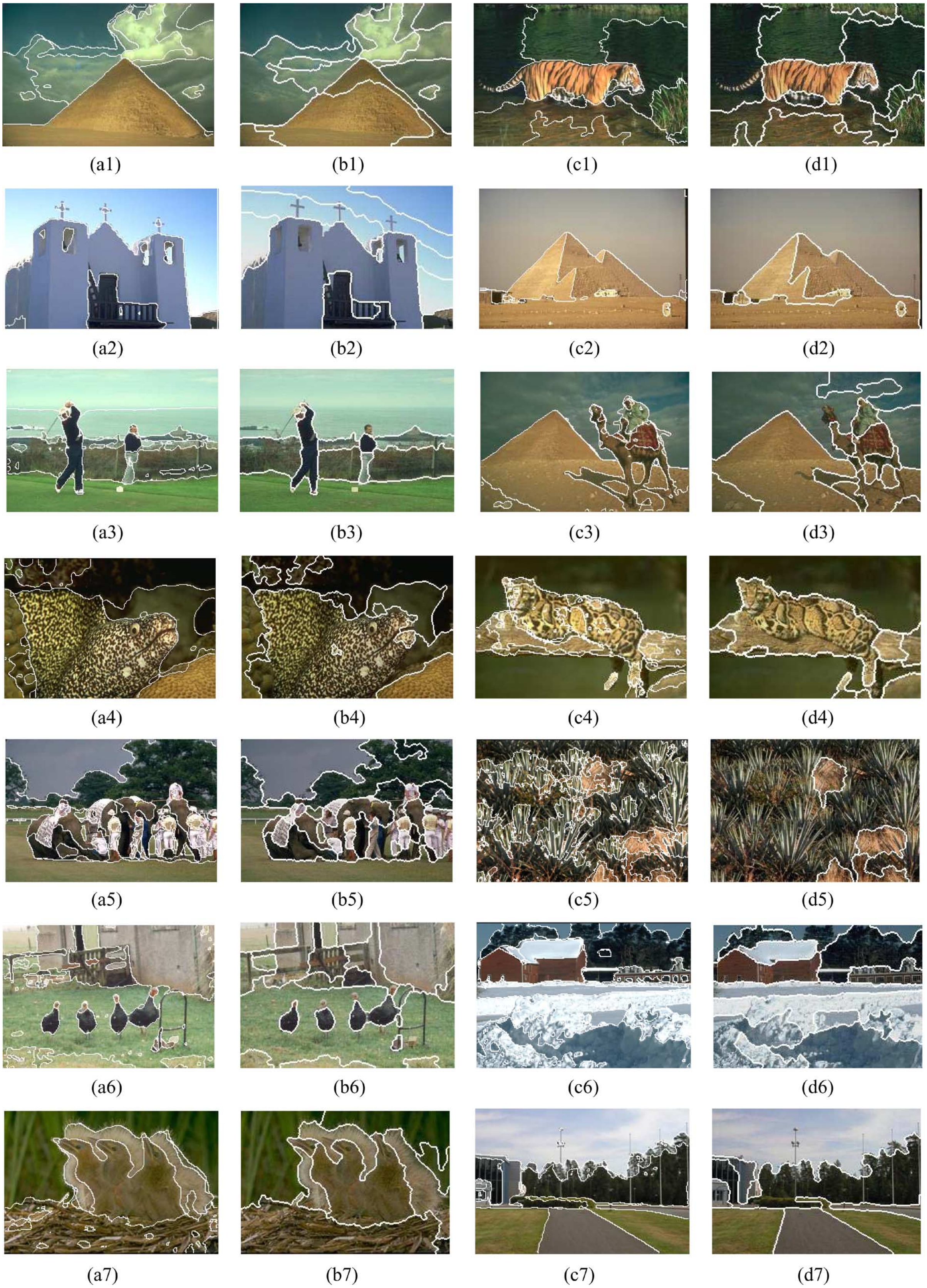
with the ones obtained using the JSEG segmentation algorithm. Although JSEG has a very different computational architecture then CTex, this comparison is appropriate since both algorithms include the color and texture attributes in the segmentation process.

While the tests conducted on McGill [42] and Outex [43] databases only allowed a qualitative evaluation (since no ground truth data is available), the tests performed on the Berkeley database allowed us to conduct both qualitative and quantitative evaluations since this database provides a set of manual segmentations for each image. In this paper, the quantitative measurements were carried out using the Probabilistic Rand index (PR) [44]. The PR index performs a comparison between the obtained segmentation result and multiple ground-truth segmentations by evaluating the pairwise relationships between pixels. In other words, the PR index measures the agreement between the segmented result and the manually generated ground-truths and takes values in the range $[0, 1)$, where a higher PR value indicates a better match between the segmented result and the ground-truth data.

Table III depicts the mean and the standard deviation of the PR values that are calculated when the CTex and JSEG algorithms were applied to all 300 images in the Berkeley database. As it can be observed in Table III, the CTex algorithm achieved a mean value of 0.80 while the mean value for JSEG is 0.77. The relative small difference between the quantitative results shown in Table III is motivated by the fact that the ground truth images from the Berkeley database are in general under-segmented since the manual annotation of these images was performed to highlight only the perceptual uniform regions. Obviously, this testing scenario favored the JSEG algorithm while the goal of the CTex framework is to achieve image segmentation at a high level of image detail. This can be observed in Fig. 11 where a number of segmentation results achieved after the application of CTex and JSEG on natural images are illustrated. The results depicted in Fig. 11 indicate that although the overall performance of the JSEG algorithm is good, it has difficulty in the identification of the image regions defined by low color contrast (see Fig. 11 b3, d3, b4, d4, and d5) and small and narrow details (see Fig. 11 d1, b2, b3, d3, and b6). These results also indicate that the CTex technique was able to produce consistent results with respect to the border localization of the perceptual regions and the level of image detail (see Fig. 11 c1, a2, a3, c3, c5, and a6) and shows better ability than the JSEG algorithm in handling the local inhomogeneities in texture and color. The elimination of the small and narrow image details in the segmented results by the JSEG algorithm is caused by two factors. The first is generated by the fact that the region growing that implements the segmentation process performs the seed expansion based only on the J values that sample the texture complexity rather than a texture model and the spatial continuity is evaluated in relative large neighborhoods. The second factor that forces the JSEG algorithm to eliminate the small regions from the segmented output is related to the procedure applied to determine the initial seeds for the region growing algorithm. In the original implementation proposed by Deng and Manjunath [18] the initial seeds correspond to minima of local J values and to prevent the algorithm to be trapped in local minima the authors imposed a size criterion for the candidate seed region. In contrast to this approach, the algorithm detailed in this paper (CTex) evaluates the color and texture information using explicit models (distributions) and the spatial continuity is enforced during the adaptive integration of the color and texture features in the ASKM framework. As illustrated in (14), the ASKM algorithm is able to adjust the size of the local color and texture distributions to the image content and this is an important advantage of our algorithm while the level of image detail in the segmented output is preserved. In Fig. 11, segmentation results of natural images from McGill [42] and Outex [43] databases are also included. For clarity purposes the segmentation borders for both algorithms were superimposed on the original image.

TABLE III
PERFORMANCE EVALUATION OF THE CTEX AND JSEG ALGORITHMS CONDUCTED ON THE BERKELEY DATABASE

	PR Index$_{\text{mean}}$	PR Index$_{\text{standard\_deviation}}$
JSEG	0.77	0.12
CTex	0.80	0.10



(a1) (b1) (c1) (d1)

(a2) (b2) (c2) (d2)

(a3) (b3) (c3) (d3)

(a4) (b4) (c4) (d4)

(a5) (b5) (c5) (d5)

(a6) (b6) (c6) (d6)

(a7) (b7) (c7) (d7)

Fig. 11. Segmentation of natural images using the CTex and JSEG algorithms. First and third columns (a), (c): CTex segmentation results. Second and fourth columns (b), (d): JSEG segmentation results.

It is useful to note that some small erroneous regions are generated by our approach that are caused by the incorrect adaptation of the window size [see (14)] when dealing with small image regions defined by step transitions in color and texture. This problem can be addressed by applying a postprocessing merging procedure, but this will lead to a reduction in the level of image detail in the segmented data. However, this problem is difficult to tackle taken into consideration the unsupervised nature of the CTex algorithm, but the experimental results indicate that our segmentation framework is robust in determining the main perceptual regions in natural images. One potential solution to address this problem is to include in the ASKM formulation a new regularization term that penalizes the weak discontinuities between adjacent regions by calculating a global spatial continuity cost. This can be achieved by embedding the ASKM process into an energy minimization framework.

Due to the large number of distributions that have to be calculated during the ASKM data clustering process, the computational complexity of the CTex algorithm is higher than that associated with JSEG. For instance, CTex segments a mosaic image of size 184 × 184 in 85 s, while JSEG requires only 6 s, but it is worth mentioning that the implementation of the CTex algorithm has not been optimized with respect to the minimization of the computational cost. The experiments have been conducted using a 2.4-GHz AMD X2 4600 PC and running Windows XP.

VIII. Conclusion

In this paper, we presented a new segmentation algorithm where the color and texture features are adaptively evaluated by a clustering strategy that enforces the spatial constraints during the assignment of the data into image regions with uniform texture and color characteristics. The main contribution of this work resides in the development of a novel multispace color segmentation scheme where an unsupervised SOM classifier was applied to extract the dominant colors and estimate the optimal number of clusters in the image. The second strand of the algorithm dealt with the extraction of the texture features using a multichannel decomposition scheme based on Gabor filtering. The inclusion of the color and texture features in a composite descriptor proved to be effective in the identification of the image regions with homogenous characteristics. The performance of the developed color-texture segmentation algorithm has been quantitatively and qualitatively evaluated on a large number of synthetic and natural images and the experimental results indicate that our algorithm is able to produce accurate segmentation results even when applied to images characterized by low resolution and low contrast.

References

[1] M. Tuceryan and A. K. Jain, "Texture analysis," in *Handbook of Pattern Recognition and Computer Vision*, C.H. Chen, L.F. Pau, and P.S.P. Wang, Eds., 2nd ed. Singapore: World Scientific, 1998, pp. 207–248.

[2] H. D. Cheng, X. H. Jiang, Y. Sun, and J. L. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, Dec. 2001.

[3] A. Materka and M. Strzelecki, "Texture analysis methods—A review," Rep. COST B11, Inst. Electron., Tech. Univ. Lodz, 1998.

[4] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr., 1999.

[5] K. Y. Song, M. Petrou, and J. Kittler, "Defect detection in random color textures," *Image Vis. Comput.*, vol. 14, pp. 667–683, 1996.

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

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

[8] C. H. Yao and S. Y. Chen, "Retrieval of translated, rotated and scaled color textures," *Pattern Recognit.*, vol. 36, no. 4, pp. 913–929, Apr., 2003.

[9] M. A. Hoang, J. M. Geusebroek, and A. W. M. Smeulders, "Color texture measurement and segmentation," *Signal Process.*, vol. 85, no. 2, pp. 265–275, 2005.

[10] J. Freixenet, X. Munoz, J. Marti, and X. Llado, "Color texture segmentation by region-boundary cooperation," in *Proc. 8th Eur. Conf. Computer Vision*, Prague, Czech Republic, 2004, pp. 259–261.

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

[12] H. Permuter, J. Francos, and I. Jermyn, "A study of Gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognit.*, vol. 39, no. 4, pp. 695–706, 2006.

[13] M. Vanrell, R. Baldrich, A. Salvatella, R. Benavente, and F. Tous, "Induction operators for a computational color-texture representation," *Comput. Vis. Image Understand.*, vol. 94, no. 1-3, pp. 92–114, 2004.

[14] Y. G. Wang, J. Yang, and Y. C. Chang, "Color-texture image segmentation by integrating directional operators into JSEG method," *Pattern Recognit. Lett.*, vol. 27, no. 16, pp. 1983–1990, 2006.

[15] L. Shi and B. Funt, "Quaternion color texture segmentation," *Comput. Vis. Image Understand.*, vol. 107, pp. 88–96, 2007.

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

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

[18] 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.

[19] T. S. C. Tan and J. Kittler, "Color texture analysis using color histogram," *IEE Proc. Vision, Image, Signal Process.*, vol. 141, no. 6, pp. 403–412, Dec. 1994.

[20] P. Nammalwar, O. Ghita, and P. F. Whelan, "Experimentation on the use of chromaticity features, local binary pattern and discrete cosine transform in color texture analysis," in *Proc. 13th Scandinavian Conf. Image Analysis*, Goteborg, Sweden, 2003, pp. 186–192.

[21] S. Liapis and G. Tziritas, "Color and texture image retrieval using chromaticity histograms and wavelet frames," *IEEE Trans. Multimedia*, vol. 6, no. 5, pp. 676–686, Mar. 2004.

[22] T. Mäenpää and M. Pietikäinen, "Classification with color and texture: Jointly or separately?," *Pattern Recognit. Lett.*, vol. 37, no. 8, pp. 1629–1640, Aug., 2004.

[23] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1026–1038, Aug. 2002.

[24] D. E. Ilea and P. F. Whelan, "Adaptive pre-filtering techniques for color image analysis," in *Proc. Int. Machine Vision & Image Processing Conf.*, Maynooth, Ireland, Sep. 2007, pp. 150–157.

[25] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Forward-and-backward diffusion processes for adaptive image enhancement and denoising," *IEEE Trans. Image Process.*, vol. 11, no. 7, pp. 689–703, Jul. 2002.

[26] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

[27] B. Smolka and K. N. Plataniotis, *On the Coupled Forward and Backward Anisotropic Diffusion Scheme for Color Image Enhancement*. London, U.K.: Springer Verlag, 2002, vol. 2383, Lecture Notes in Computer Science, pp. 70–80.

[28] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the K-Means algorithm," *Pattern Recognit. Lett.*, vol. 20, pp. 1027–1040, 1999.

[29] T. Kohonen, *Self-Organising Maps*, 3rd ed. New York: Springer Verlang, 2001, vol. 30, pt. Springer Ser. Information Sciences.

[30] G. Dong and M. Xie, "Color clustering and learning for image segmentation based on neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 925–936, Jul. 2005.

[31] S. H. Ong, N. C. Yeo, K. H. Lee, Y. V. Venkatesh, and D. M. Cao, "Segmentation of color images using a two-stage self-organising network," *Image Vis. Comput.*, vol. 20, pp. 279–289, 2002.

[32] D. E. Ilea and P. F. Whelan, "Color image segmentation using a self-initializing EM algorithm," presented at the Int. Conf. Visualization, Imaging and Image Processing, Spain, Aug. 2006.

[33] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 55–73, Jan., 1990.

[34] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filtering," *Pattern Recognit.*, vol. 33, pp. 1167–1186, 1991.

[35] T. Randen and J. H. Husoy, "Texture segmentation using filters with optimized energy separation," *IEEE Trans. Image Process.*, vol. 8, no. 4, pp. 571–582, Apr. 1999.

[36] J. G. Daugman, "Complete discrete 2D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 7, pp. 1169–1179, Jul. 1988.

[37] D. E. Ilea and P. F. Whelan, "Automatic segmentation of skin cancer images using adaptive color clustering," in *Proc. China-Ireland Int. Conf. Information and Communications Technologies*, Hangzhou, China, 2006, pp. 348–351.

[38] D. E. Ilea and P. F. Whelan, "Color image segmentation using a spatial K-means clustering algorithm," in *Proc. Irish Machine Vision and Image Processing Conf.*, 2006, pp. 146–153.

[39] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," *Comput. Vis. Image Understand.*, vol. 84, no. 1, pp. 25–43, 2001.

[40] Vision Texture (VisTex) Database, Media Lab., Mass. Inst. Technol., Cambridge, MA [Online]. Available: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html

[41] 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," *ICCV*, pp. 416–425, 2001.

[42] A. Olmos and F. A. A. Kingdom, McGill Calibrated Color Image Database 2004 [Online]. Available: http://tabby.vision.mcgill.ca

[43] Outex Natural Images Database [Online]. Available: http://www.outex.oulu.fi

[44] R. Unnikrishnan and M. Hebert, "Measures of similarity," presented at the IEEE Workshop on Computer Vision Applications 2005.

Dana E. Ilea received the B.Eng. degree in electronic engineering and computer science from Transilvania University, Brasov, Romania, in 2005. She is currently pursuing the Ph.D. degree in computer vision at the Vision Systems Group at Dublin City University, Dublin, Ireland.

Currently, she is a member of the Vision Systems Group at Dublin City University. Her research interests are in the areas of image processing, texture and color analysis, and medical imaging.

Paul F. Whelan (S'84–M'85–SM'01) received the B.Eng. (Hons) degree from NIHED, Dublin, Ireland, in 1985, the M.Eng. degree from the University of Limerick, Ireland, in 1990, and the Ph.D. degree from the University of Wales, Cardiff, U.K., in 1994.

During the period of 1985–1990, he was with ISI, Ltd., and later Westinghouse (WESL), where he was involved in the research and development of industrial vision systems. He was appointed to the School of Electronic Engineering, Dublin City University (DCU), in 1990, and is currently Professor of Computer Vision (Personal Chair). He founded the Vision Systems Group in 1990 and the Centre for Image Processing and Analysis in 2006, and currently serves as its Director. As well as publishing over 150 peer-reviewed papers, he has coauthored two monographs, namely *Intelligent Vision Systems for Industry* (Springer, 1997) and *Machine Vision Algorithms in Java: Techniques and Implementation* (Springer, 2000). He has also coedited three books, including *Selected Papers on Industrial Machine Vision Systems* (SPIE, 1994). His research interests include image segmentation and its associated quantitative analysis (specifically mathematical morphology, texture analysis) with applications in computer vision and medical imaging (computer aided detection and diagnosis).

Prof. Whelan is a Chartered Engineer (IET). He served on the governing board of the International Association for Pattern Recognition (IAPR) and is a past President of the Irish Pattern Recognition and Classification Society. He is an HEA-PRTLI and Science Foundation Ireland (SFI) funded principal investigator.