Perceptual Color and Spatial Texture Features for Segmentation

Junqing Chen, Thrasyvoulos N. Pappas[a], Aleksandra Mojsilovic and Bernice E. Rogowitz[b]

[a]Dept. of Electrical and Computer Engineering, Northwestern University, USA
[b]IBM T.J. Watson Research Center, Hawthorne, NY 10532, USA

ABSTRACT

We develop spatially adaptive, low-level, color and spatial texture features based on perceptual principles about the processing of texture and color information. We then propose an algorithm that combines these features to obtain image segmentations that convey semantic information that can be used for content-based retrieval. Our focus is images of natural scenes. The color texture features are based on the estimation of spatially adaptive dominant colors, which on one hand, reflect the fact that the human visual system cannot simultaneously perceive a large number of colors, and on the other, the fact that image colors are spatially varying. The spatially adaptive dominant colors are obtained using a previously developed adaptive clustering algorithm for color segmentation. The spatial texture features are based on a steerable filter decomposition, which offers an efficient and flexible approximation of early processing in the human visual system. We use the local energy of the subband coefficients as a simple but effective characterization of spatial texture. A median filter is used to distinguish the energy due to region boundaries from the energy of the textures themselves. Texture feature estimation requires a finite neighborhood that limits spatial resolution, while color segmentation provides accurate and precise edge localization. By combining texture with color information, the proposed algorithm can obtain robust segmentations that are accurate and precise. The performance of the proposed algorithm is demonstrated in the domain of photographic images, including low resolution, degraded, and compressed images.

Keywords: Adaptive clustering algorithm, optimal color composition distance, steerable decomposition, CBIR, local median energy, spatially adaptive dominant colors

1. 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 image segmentation based on texture and color. Although significant progress has been made in texture segmentation (*e.g.*, Refs. 3–6) and color segmentation (*e.g.*, Refs. 7–9) separately, the area of combined spatial texture and color segmentation problem is still quite open and active.[10,11] 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.

Current algorithms for low-level feature extraction, such as color, texture, and shape are quite sophisticated and have met with considerable success. However, there is still a large gap between the high-level semantic concepts that human beings use for image retrieval and the low-level image features that most current algorithms are based on. We attempt to bridge this gap by developing algorithms that use low-level features based on perceptual models and principles about the processing of texture and color information. Such features and the resulting segmentations can then be correlated with high-level semantics and used to capture the semantic meaning of an image. We present an image segmentation algorithm

Further author information: (Send correspondence to T.N.P.)
J.C.: E-mail: jqchen@ece.northwestsern.edu
T.N.P.: E-mail: pappas@ece.northwestsern.edu, Address: ECE Dept., Northwestern Univ., 2145 Sheridan Rd, Evanston, IL 60208
A.M.: E-mail: aleksand@us.ibm.com
B.E.R.: E-mail: rogowt@us.ibm.com

that is based on spatially adaptive color and spatial texture features. An early version of this algorithm has appeared in Ref. 12. In this paper, we review the basic elements of the algorithm, with emphasis on perceptual considerations. We also propose several improvements that enhance the perceptual aspects of the algorithm.

The color and spatial texture features are first developed independently, and then combined to obtain an overall segmentation. As we pointed out in Ref. 12, texture feature estimation requires a finite neighborhood that limits spatial resolution, while color segmentation can provide accurate and precise edge localization. By combining texture with color information, the proposed algorithm can obtain robust, and at the same time, accurate and precise segmentations. The color texture features are based on the estimation of spatially adaptive dominant colors, which on one hand, reflect the fact that the human visual system cannot simultaneously perceive a large number of colors, and on the other, the fact that image colors are spatially varying. The spatially adaptive dominant colors are obtained using the adaptive clustering algorithm for segmentation proposed by Pappas.[7] The (spatial) texture features are based on a steerable filter decomposition, which offers an efficient and flexible approximation of early processing in the human visual system. We use the local energy of the subband coefficients as a simple but effective characterization of spatial texture. A median filter is used to distinguish the energy due to region boundaries from the energy of the textures themselves. As was shown in Ref. 12, median operators tend to respond to texture within uniform regions and suppress textures associated with transitions between regions. In contrast to texture analysis/synthesis techniques that use a large number of parameters to describe texture, our segmentation algorithm relies on only a few parameters to segment the image into simple yet meaningful texture categories.

The proposed algorithm includes an elaborate border refinement procedure, which extends the idea of the adaptive clustering algorithm used for the color segmentation[7] to color texture. In natural images, the intensity, color, and texture of a perceptually uniform region can change gradually but significantly from one side of a region to the other. The proposed algorithm adapts to such variations by estimating the color and texture parameters over a hierarchy of window sizes that progressively decreases as the algorithm converges to the final segmentation. Overall, the proposed algorithm provides a robust segmentation with accurate and precise boundaries.

The image segmentation results can be used to derive region-wide 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 humans may recognize them. A key to the success of the proposed approach is the recognition of the fact that it is not necessary to obtain a complete understanding of a given image: In many cases, the identification of a few key segments (such as "sky," "mountains," "people," etc.) may be enough to classify the image into a given category.[13] Thus, some of the regions will be classified as "complex" or "none of the above," and as such will still play a significant role in scene analysis.

The performance of the proposed algorithm is demonstrated in the domain of photographic images. It consists of a wide variety of images: outdoor and indoor scenes, including landscapes, cityscapes, plants, animals, people, and man-made objects. A challenging aspect of our work is that we attempt to accomplish the above tasks with relatively low resolution (*e.g.*, 200×200) and occasionally degraded or compressed images, just as humans can do it.

The paper is organized as follows. In Section 2, we review the color texture feature extraction. Our new approach for spatial texture feature extraction is presented in Section 3. Section 4 discusses the proposed algorithm for combining the texture and color features to obtain an overall segmentation. Segmentation results and comparisons to other approaches are also presented in Section 4.

2. PERCEPTUAL COLOR TEXTURE FEATURES

Color has been used extensively as a low-level feature for image retrieval.[1,14–16] In this section, we discuss color texture features that take into account both image characteristics and human perception.

An important characteristic of human color perception is that the human eye cannot simultaneously perceive a large number of colors,[17] even though under appropriate adaptation, it can distinguish more than two million colors.[18] In addition, the number of colors that can be internally represented and identified in cognitive space is about 30.[19] A small set of color categories provides a very efficient representation, and more importantly, makes it easier to capture invariant properties in object appearance.[20] Based on the assumption that color categories are related to the statistical structure of the perceived environment, Yendrikhovskij[20] proposed algorithms (*e.g.*, K-means clustering) for computing color categories.

Along similar lines, subjective experiments reported by Mojsilovic *et al.* in Ref. 21 indicate that, when presented with various color patterns, humans are not able to perceive more than six or seven colors.

The idea of using a compact color representation in terms of dominant colors for image analysis was introduced by Ma *et al.*.[22] Their argument for using such a representation was based on the fact that most natural scenes require only a few colors to describe their color content without significantly affecting color quality. The representation they proposed consists of the dominant colors (expressed in *RGB* space) along with the percentage of occurrence of each color. Mojsilovic *et al.*[21] adopted this representation using an (approximately) perceptually uniform color space, such as *Lab* or *Luv*. It has been shown that the quality of image retrieval algorithms can be substantially improved by using such color spaces.[23]

In Ref. 12 we introduced the idea of *spatially adaptive dominant colors*. This is necessary because of the spatially varying image characteristics and the adaptive nature of the human visual system. An observer's notion of a blue or brown or green color is highly dependent on the surrounding colors; moreover, it varies with the lighting conditions and the colors of the display device. For example, a color vision study found that color appearance depends more on the local contrast rather than absolute contrast value.[24] Our color 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:

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

where each of the dominant colors, c_i, is a three dimensional vector in *Lab* space, and p_i are the corresponding percentages. $N_{x,y}$ represents the neighborhood around the pixel at location (x,y) and M is the total number of colors in the neighborhood. A typical value is $M = 4$. The color feature vector at each pixel is essentially a crude "local histogram" of the image.

In addition to modeling human perception and image characteristics, the use of spatially adaptive dominant colors, has other advantages compared to dominant colors that are fixed over an image or a collection of images. For example, one must first determine the number of dominant colors. This number must be big enough to capture the characteristics of the image or collection of images, and at the same time, it should not be too big in order to maintain a concise feature representation. One approach for finding the dominant colors is to use vector quantization (VQ)[25,26] to obtain a set of colors that minimize the mean-square quantization error across all images in the database. This approach was taken in Refs. 17, 22. However, it requires a large training set, which increases the computational burden for the codebook design. To overcome this problem, Mojsilovic *et al.*[27] proposed a technique, based on Fibonacci lattices, that provides uniform sampling of the chromaticity planes in *Lab* space. An alternative approach is to find the dominant colors of a given image (*e.g.*, using VQ). While the resulting colors may be useful in characterizing the image as a whole, the corresponding segmentation could be quite inadequate due to lack of spatial constraints and spatial adaptation.[7] The advantage of using the spatially adaptive dominant colors is that we only need to specify the number of dominant colors within a given neighborhood. We found that a small number, *e.g.*, four, is adequate. 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.

The spatially adaptive dominant colors are obtained by the adaptive clustering algorithm (ACA) proposed in Ref. 7 and extended to color in Ref. 8. The ACA is an iterative algorithm that uses spatial constraints in the form of Markov random fields (MRF). The initial estimate is obtained by the K-means algorithm,[28,29] 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 K-means algorithm is the same as the VQ techniques discussed above.) As the algorithm progresses, the dominant colors are updated 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.

Fig. 1 compares the adaptive dominant colors obtained by ACA[7] to the constant dominant colors obtained by the Comaniciu-Meer algorithm.[9] The latter is a relatively simple and quite effective algorithm that has been used for obtaining the dominant colors of an image. It is based on the "mean shift" algorithm for estimating density gradients, and essentially works with the image histogram, even though it also attempts to incorporate spatial constraints by imposing constraints on the connectivity of the detected regions. However, like the other approaches we discussed above, it does not take into consideration the fact that the dominant colors may be slowly varying across the image. The image resolution is 250×214 pixels. The examples for the Comaniciu-Meer algorithm were generated using the "oversegmentation" setting. Note the false contours in the Comaniciu-Meer 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,

(a) Original color image (b) Comaniciu-Meer algorithm (c) Adaptive clustering algorithm (ACA)

Figure 1. Color image segmentation (shown in color)

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 textured regions, like the mountain area, the ACA oversegments the image, while in other areas, like the forest it consolidates everything into one region. This depends on the size of the features and the color differences, and is controlled by the strength of the MRF. The latter can be chosen to control the amount of detail that is perceptually relevant.

The ACA segments the image into color classes. At every pixel in the image, each class is represented by a color that is equal to the average color of the pixels in its neighborhood that belong to that class.[7] In the example of Fig. 1(c), each pixel is painted with the representative color of the class that it belongs to. Assuming that the dominant colors are slowly varying, we can assume that they are approximately constant in the immediate vicinity of a pixel. We can then count the number of pixels in each class within a given window, which together with the average color values, provides the color feature representation of the form (1) for the pixel.

The ACA provides a color feature vector for each image pixel. In Section 4, we will use these features, along with texture features we develop in the next section, to obtain the final image segmentation. For that we need a metric that measures the perceptual similarity between two color 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.[17,27] 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.[27] Various suboptimal solutions have been proposed.[17,22] Mojsilovic *et al.*[27] proposed the *Optimal Color Composition Distance (OCCD)*, which finds the optimal mapping between the dominant colors of two images, and thus, provides a better similarity measure. The OCCD 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 produces a close approximation of the optimal solution.

The pseudo code for the proposed OCCD implementation along with an illustrative example are shown in Figs. 2 and 3. 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 original OCCD, and thus, can be even more accurate than the original implementation. Once the color correspondences are established, the OCCD distance is then calculated as follows:

$$Dist(f_c^1, f_c^2) = \sum_{i=0}^{M} Dist(c_i^1, c_i^2) * p_i \tag{2}$$

where c_i^1, c_i^2, and p_i are the matched colors and corresponding percentage after the procedure outlines in Fig. 2.

1. Given two color 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. 3. Create an empty destination stack for each vector.
2. Select a pair of tokens (c_a, p_a) and (c_b, p_b), one from each feature vector, whose colors are closest, regardless of percentages.
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.

Figure 2. Steps of the simplified version of OCCD

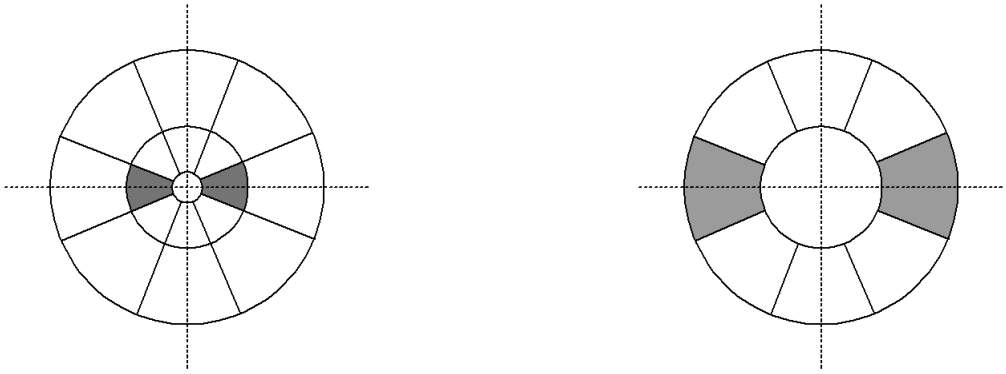
Figure 3. Example of the simplified version of OCCD

3. PERCEPTUAL FEATURES FOR SPATIAL TEXTURE

As we discussed in the introduction, the color and spatial texture features are developed independently. Thus, we use only the grayscale component of the image to obtain the spatial texture features, based on which we obtain an intermediate segmentation, which is then combined with the color texture features to produce the final segmentation. This is in contrast to the approaches described in Refs. 11, 17, where the color quantization/segmentation is used to obtain an achromatic pattern map which becomes the basis for texture feature extraction.

From the signal processing perspective, the most significant difference between color and texture is that the latter requires a finite neighborhood to be defined. In addition to the increased computational burden, this limits the resolution of texture segmentation. As we saw in the previous section, the ACA algorithm can provide very accurate and precise edge localization. A texture segmentation algorithm, on the other hand, requires neighborhood operations which make it difficult to define texture near region boundaries.

Like many of the existing algorithms for texture analysis and synthesis (*e.g.*, Refs. 5, 6, 30, 31), our approach is based on a multiscale frequency decomposition. Such decompositions have been widely used as descriptions of early visual processing in mammals. They have also been used in a number of applications, such as texture classification and segmentation,[31–38] texture analysis/synthesis,[36, 38–44] as well as image quality evaluation.[45] Examples of such decompositions are the Cortex transform[46] and the Gabor transform.[35, 36] The steerable pyramid decomposition[47–49] is very similar in spirit to the cortex transform, but offers more efficiency and flexibility. It can be designed to produce any number of orientation bands and it is steerable, *i.e.*, a filter of arbitrary orientation can be synthesized as a linear combination of a



(a) 2-level decomposition
horizontal bands of scale 2 shown in gray.

(b) 1-level decomposition
horizontal bands of scale 1 shown in gray.

Figure 4. Steerable Filter Decomposition.

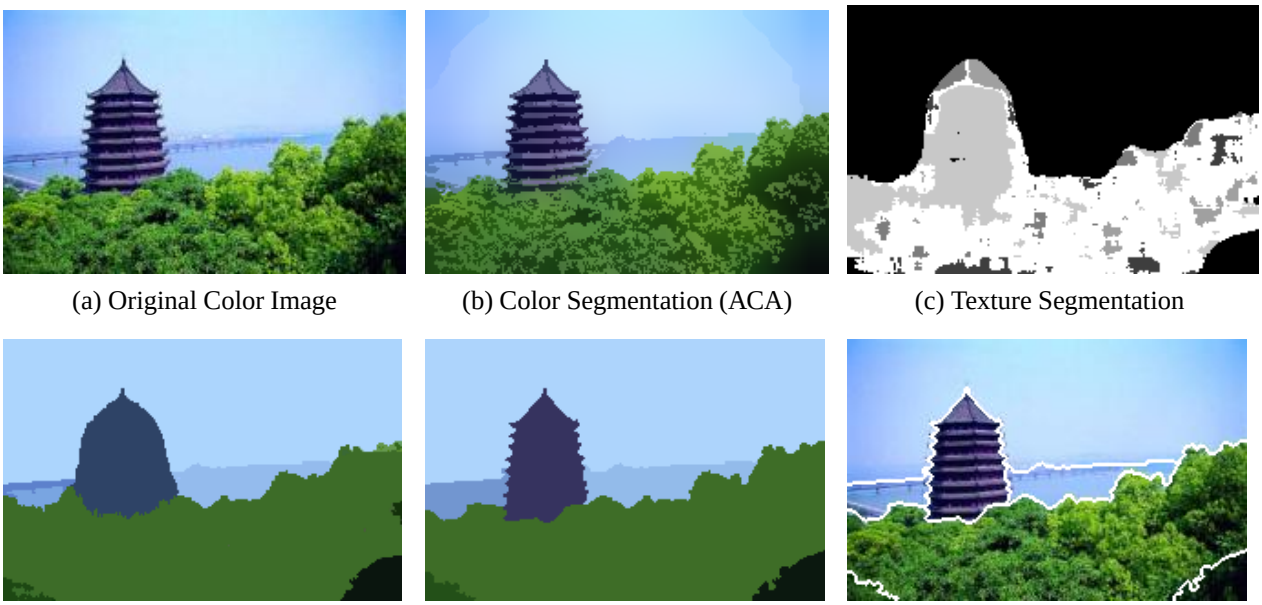
set of basis filters. It is thus ideally suited for the problem of texture analysis. On the other hand, in image compression applications, perceptual models have been mostly based on simpler separable subband and wavelet decompositions.[50–56] These decompositions can be regarded as crude approximations of the cortex transform. One of their drawbacks, as far as texture analysis is concerned, is that they cannot separate the two diagonal directions.

In contrast to the texture synthesis problem that requires an elaborate model in order to accurately synthesize a wide range of textures, the model for the segmentation problem can be quite crude. There are two fundamental differences that distinguish the segmentation problem from that of texture analysis/synthesis. First, in texture analysis/synthesis they consider isolated textures, and thus, do not have to worry about the resolution loss due the neighborhood operations necessary for texture definition. Second, they work with relatively high resolution images which allow the precise estimation of texture parameters. In contrast, we want to identify textures in thumbnail images which contain, not only one, but several textures. Thus, by necessity, our texture models have to be a lot simpler, so that we can determine their parameters from a few sample points. Of course, we know that the solution to this problem is possible, because the human visual system can do it.

As we discussed above, the steerable filter decomposition offers an efficient and flexible approximation of early processing in the human visual system. Thus, our spatial texture feature extraction is based on the steerable pyramid. Fig. 4 shows examples of frequency decompositions that can be obtained by 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.[3,57] Various nonlinear operations have been used to boost up the sparse subband coefficients. Chang and Kundu[3,57] used the average of coefficients' energy with in a small window. Manjunath and Ma[22,31] used both the mean and the standard deviation of the magnitude of the Gabor transform subband coefficients as texture features. We propose using the *median local energy* of the oriented filter outputs. This energy is computed by first squaring the subband coefficients and then taking the median over a small window. The advantage of the median filter is that it suppresses textures associated with transitions between regions, while it responds to texture within uniform regions.[12] The use of median local energy as a nonlinear operation also agrees with Graham and Sutter's conclusion[58–60] that nonlinearity operator in texture segregation must have accelerating/expansive nature.

We use a steerable filter decomposition[49] with four orientation bands (horizontal, vertical, $+45^o$, -45°) as shown in Fig. 4. Most researchers have used four to six orientation bands to approximate the orientation selectivity of the human visual system (*e.g.*, Refs. 46, 61). 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. 4 (b)) is adequate. Out of those, we only use the four orientation bands. The filters are applied to the grayscale component of the original image, and the median local energy is computed for each subband, followed by a 2-level K-means algorithm. This provides a classification into *smooth* and *non-smooth* regions as follows. If a pixel belongs to the low-energy cluster in all subbands, then it belongs to the smooth class, otherwise it is classified as non-smooth. The non-smooth pixels are then further classified into different orientation classes, by comparing the first and second maxima among the four subband coefficients. If the first and second maxima are not close, then the orientation is determined by the first maximum. Otherwise, the pixel is classified as *complex*, as



(a) Original Color Image (b) Color Segmentation (ACA) (c) Texture Segmentation

(d) Crude Segmentation (e) Final Segmentation (f) Final Segmentation (on original image)

Figure 5. Color and Texture Image Segmentation (a,b,d,e,f shown in color)

there is no dominant orientation. Thus, we can have up to six classes. Fig. 5 (c) shows an example of such a classification. The black areas correspond to smooth regions and the white areas correspond to complex regions. Different shades of gray (from light to dark) represent the *horizontal*, -45°, $+45^o$, and *vertical* regions. We also experimented with the discrete wavelet decomposition (DWT),[12] using a similar classification procedure, and found that the steerable pyramid produces superior results.

In the texture classes extraction procedure, we found that the window size for median operator is of critical importance. The window size 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 window size determination should be based on careful subjective experiments. Note also, that the window size depends on the specific decomposition, as we found that the DWT requires smaller window sizes.[12] That's because the DWT is downsampled while the steerable subbands that we use do not involve any downsampling. The window size is also related to the extent of the analysis filters.

4. SEGMENTATION ALGORITHM

We now consider an algorithm that combines the color and spatial texture features to obtain the overall image segmentation. It starts with the texture classification we discussed in the previous section and uses the color texture features to refine it.

First, we consider the smooth texture regions. As we discussed in Section 2, the ACA was developed for images with smooth regions. Thus, in those regions, we should rely on the ACA for the final segmentation. However, some region merging may be necessary. Thus, in the smooth regions, we find all the connected segments that belong to different color classes, and then merge neighboring segments if the average color difference across the common border is below a given threshold. Finally, any remaining small color segments that are close to non-smooth texture regions, are relabeled as non-smooth, so that they can be considered in the next step.

For all other texture classes (horizontal, vertical, $+45^o$, -45°, and complex), we use a region growing algorithm to obtain a *crude* segmentation. We use a multi-grid approach. We start with pixels located on a coarse grid, and compute the color features with a window size equal to twice the grid spacing, *i.e.* with a 50% window overlap. A pair of pixels belong to the same region if the color features are similar in the OCCD sense. The threshold is higher for pixels that belong to the same texture class (*i.e.*, easier to merge), and lower for pixels in different texture classes. In addition, we incorporate

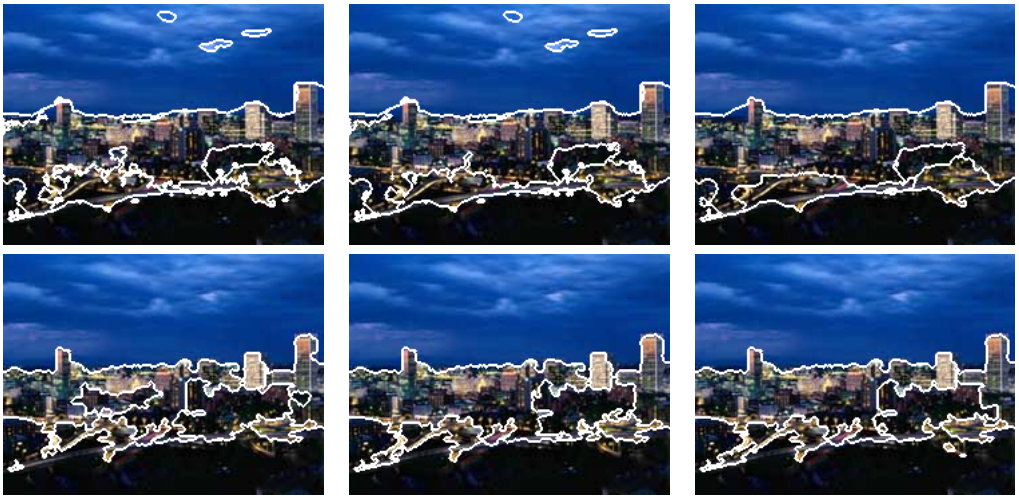


Figure 6. Illustrating the necessity of spatial constraints. Top row shows crude segmentations and bottom row are final segmentations. From left to right $\beta = 0,\ .1,\ .8$

MRF-type spatial constraints in the feature distance measure. That is, a pixel is more likely to belong to a region if many of its neighbors belong to the same region. The feature distance measure of (2) now becomes:

$$Dist(f_c^1, f_c^2) = \sum_{i=0}^{M} Dist(c_i^1, c_i^2) * p_i + \beta N_i \tag{3}$$

where β represents the strength of the spatial constraint and N_i represents the number of neighboring pixels with the same label as the one being checked. Since the MRF constraint is symmetric, it is necessary to iterate a few times for a given grid spacing. The grid spacing is then reduced, and the procedure is repeated until the spacing is equal to one pixel. Fig. 5 (d) shows an example of the resulting crude segmentation. In this figure, as well as Fig. 5 (e), the different segments have been painted by the average color of the region. Fig. 6 shows examples of crude segmentations obtained with different values of the parameter β. Note that in this case, we have superimposed the region boundaries on the original image.

Finally, we refine the crude segmentation by adaptively adjusting the borders using the color texture features. The approach is similar to that of ACA,[7] and is illustrated in Fig. 7. The dotted line represents the real 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 feature vector of the form (1), and a larger window to obtain a localized estimate of the region characteristics. For each texture segment that the window overlaps, we obtain a separate color feature vector, that is, we find the average color and percentage for each of the dominant colors. We then use the OCCD criterion to decide which segment the current pixel should belong to. As in Ref. 7, an MRF-type constraint is added to insure region smoothness. This is repeated for each pixel in a raster scan. A few iterations are necessary for convergence. The iterations converge when the number of pixels that change class is below a given threshold. We then reduce the window sizes and repeat the procedure. We use a series of window pairs starting from 35/5 and ending with 11/3. 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. Otherwise, the region is not a valid candidate. A reasonable choice for the threshold for deciding whether a region should be a valid candidate is to use the product of the two window sizes divided by 2. The refinement procedure is applied to the whole image except the smooth regions, where as we saw, the ACA provides accurate segmentation. The final segmentation results are shown in Fig. 5(e). Additional segmentation results are shown in Fig. 8.

Fig. 9 shows the segmentation results obtained by JSEG,[11] a segmentation algorithm that is also based on texture and color. We chose "no merge" option for the JSEG examples shown. Thus, in comparing with the results of the proposed algorithm in Fig. 8, 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. Thus, for example, there are no significant differences between the two algorithms

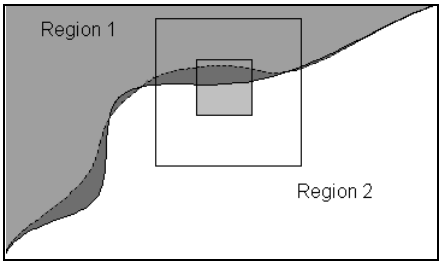


Figure 7. Illustration of border refinement.

in the forest area of example (a) or the water area of example (b). On the other hand, there are significant differences in example (f) that cannot be eliminated with regions 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 the segmentation of the train and the tracks in example (c). Finally, we should point out that in examples (b) and (d), the color of the sky is too close to the color of the mountains and cityscape.

ACKNOWLEDGMENTS

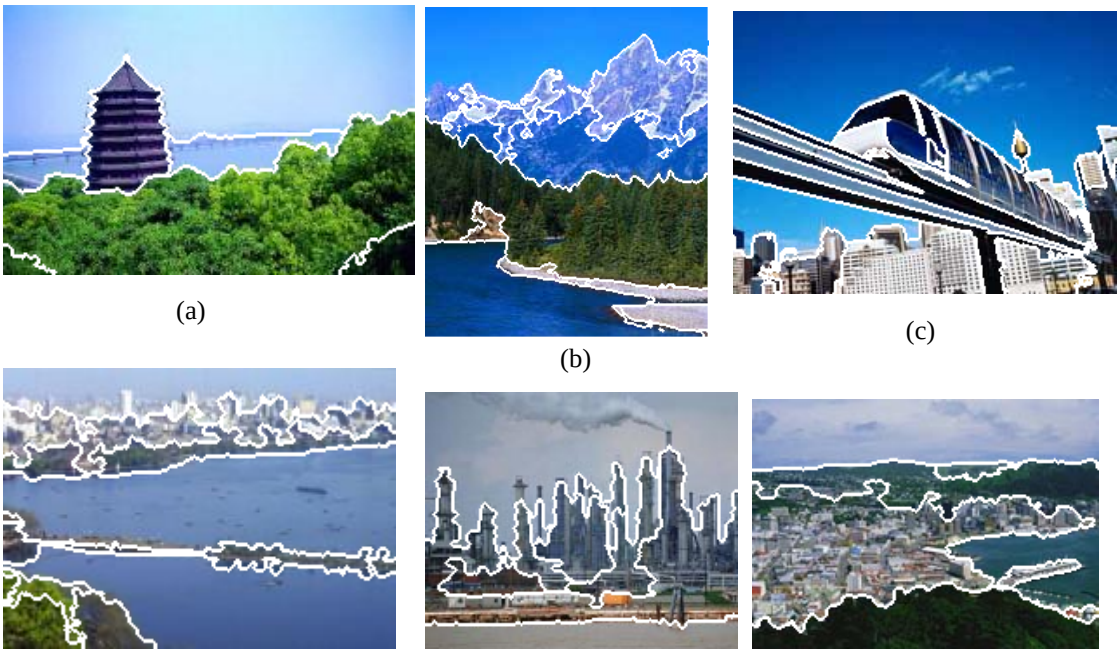
This material is based upon work supported by the National Science Foundation under Grant No.CCR-0209006. Any opinions, findings and conclusions or recomendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

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* **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 Analysis and Machine Intell.* **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* **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* **2**, pp. 429–441, Oct. 1993.
5. M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Proc.* **4**, pp. 1549–1560, Nov. 1995.
6. T. Randen and J. H. Husoy, "Texture segmentation using filters with optimized energy separation," *IEEE Trans on Image Processing* **8**, pp. 571–582, Apr. 1999.
7. T. N. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Processing* **SP-40**, pp. 901–914, Apr. 1992.
8. M. M. Chang, M. I. Sezan, and A. M. Tekalp, "Adaptive Bayesian segmentation of color images," *Journal of Electronic Imaging* **3**, pp. 404–414, Oct. 1994.
9. D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *IEEE Conf. Computer Vision and pattern Recognition*, pp. 750–755, (San Juan, Puerto Rico), June 1997.
10. 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.
11. Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Analysis and Machine Intel.* **23**, pp. 800–810, Aug. 2001.
12. J. Chen and T. N. Pappas, "Adaptive image segmentation based on color and texture," in *Proc. ICIP-02*, **III**, pp. 777–780, (Rochester, NY), Sept. 2002.
13. A. Mojsilovic and B. Rogowitz, "Capturing image semantics with low-level descriptors," in *Proc. ICIP-01*, pp. 18–21, (Thessaloniki, Greece), Oct. 2001.

14. M. Swain and D. Ballard, “Color indexing,” *Int. J. Computer Vision* **7**(1), pp. 11–32, 1991.
15. 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*, **Proc. SPIE, Vol. 1908**, pp. 173–187, (San Jose, CA), 1993.
16. B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, “Color and texture descriptors,” *IEEE Trans. Circuits and Systems for Video Technology* **11**, pp. 703–715, June 2001.
17. 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 Proc.* **1**, pp. 38–54, Jan. 2000.
18. M. R. Pointer and G. G. Attridge, “The number of discernible colors,” *Color Research and Application* **23**(1), pp. 52–54, 1998.
19. G. Derefeldt and T. Swartling, “Color concept retrieval by free color naming,” *Displays* **16**, pp. 69–77, 1995.
20. S. N. Yendrikhovskij, “Computing color categories,” in *Human Vision and Electronic Imaging V*, B. E. Rogowitz and T. N. Pappas, eds., **Proc. SPIE Vol. 3959**, (San Jose, CA), Jan. 2000.
21. A. Mojsilović, J. Kovačević, D. Kall, R. J. Safranek, and S. K. Ganapathy, “The vocabulary and grammar of color patterns,” *IEEE Trans. Image Proc.* **9**, pp. 417–431, Mar. 2000.
22. 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., **Proc. SPIE, Vol. 3016**, pp. 496–507, (San Jose, CA), Feb. 1997.
23. J. Smith and S. F. Chang, “Single color extraction and image query,” in *Proc. IEEE Int. Conf. Image Proc.*, 1995.
24. K. R. Gegenfurtner and L. T. Sharpe, eds., *Color Vision: From Genes to Perception*, Cambridge University Press, 1999.
25. Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.* **COM-28**, pp. 84–95, Jan. 1980.
26. R. M. Gray, “Vector quantization,” *IEEE ASSP Mag.* **1**, pp. 4–29, Apr. 1984.
27. A. Mojsilović, J. Hu, and E. Soljanin, “Extraction of perceptually important colors and similarity measurement for image matching, retrieval, and analysis,” *IEEE Trans. Image Proc.* **11**, pp. 1238–1248, Nov. 2002.
28. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison, 1974.
29. R. M. Gray and Y. Linde, “Vector quantizers and predictive quantizers for Gauss-Markov sources,” *IEEE Trans. Commun.* **COM-30**, pp. 381–389, Feb. 1982.
30. A. C. Bovik, M. Clark, and W. S. Geisler, “Multichannel texture analysis using localized spatial filters,” *IEEE Trans. Pattern Anal. Machine Intell.* **12**, pp. 55–73, Jan. 1990.
31. B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Trans. Pattern Analysis and Machine Intel.* **18**, pp. 837–842, Aug. 1996.
32. J. R. Bergen and E. H. Adelson, “Visual texture segmentation based on energy measures,” *Journal of the Optical Society of America A* **3**, pp. 99–, 1986.
33. M. R. Turner, “Texture discrimination by Gabor functions,” *Biol. Cybern.* **55**, pp. 71–82, 1986.
34. J. Malik and P. Perona, “Preattentive texture discrimination with early vision mechanisms,” *Journal of the Optical Society of America A* **7**, pp. 923–932, 1990.
35. J. G. Daugman and D. M. Kammen, “Pure orientation filtering: A scale invariant image-processing tool for perception research and data compression,” *Behavior Research Methods, Instruments, and Computers* **18**(6), pp. 559–564, 1986.
36. M. Porat and Y. Y. Zeevi, “Localized texture processing in vision: Analysis and synthesis in gaborian space,” *IEEE Trans. Biomed. Eng.* **36**(1), pp. 115–129, 1989.
37. D. Dunn and W. E. Higgins, “Optimal Gabor filters for texture segmentation,” *IEEE Trans. Image Processing* **4**, pp. 947–964, July 1995.
38. J. Portilla, R. Navarro, O. Nestares, and A. Tabernero, “Texture synthesis-by-analysis based on a multiscale early-vision model,” *Optical Engineering* **35**(8), pp. 2403–2417, 1996.
39. D. Cano and T. H. Minh, “Texture synthesis using hierarchical linear transforms,” *Signal Processing* **15**, pp. 131–148, 1988.
40. K. Popat and R. W. Picard, “Novel cluster-based probability model for texture synthesis, classification, and compression,” in *Proc. SPIE Visual Communications ’93*, (Cambridge, MA), 1993.

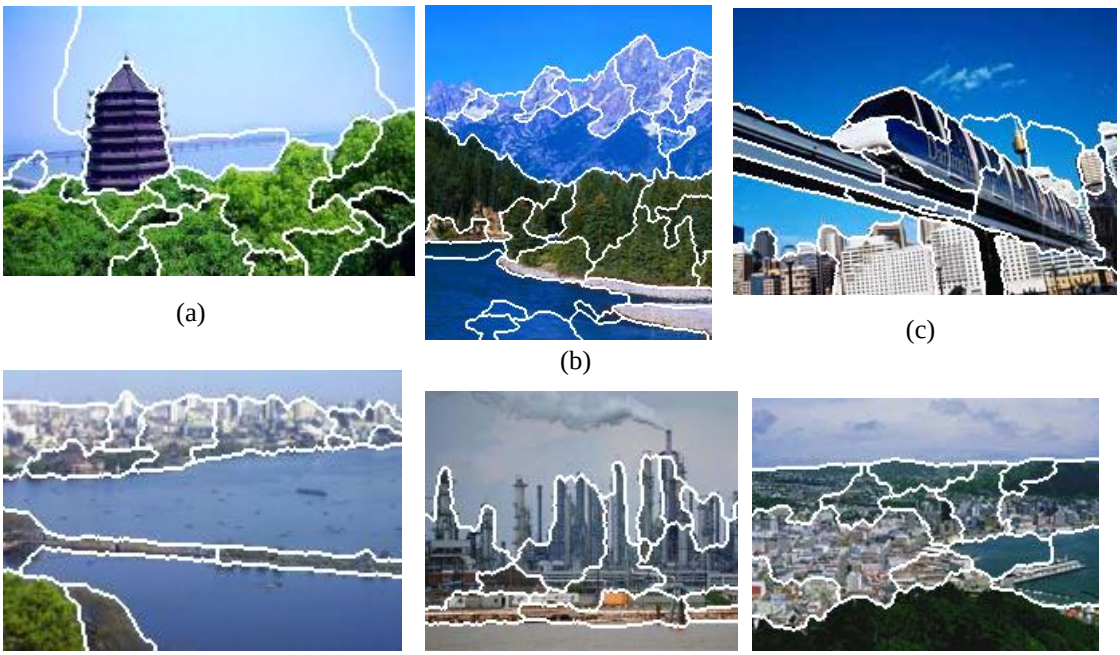
41. D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. ICIP-95, vol. III*, pp. 648–651, (Washington, DC), Oct. 1995.
42. S. Zhu, Y. N. Wu, and D. Mumford, "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling," in *IEEE Conf. Computer Vision and pattern Recognition*, pp. 693–696, 1996.
43. J. S. De Bonet and P. A. Viola, "A non-parametric multi-scale statistical model for natural images," *Adv. in Neural Info. Processing Systems* **9**, 1997.
44. J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statictics of complex wavelet coefficients," *Int. J. Computer Vision* **40**, pp. 49–71, Oct. 2000.
45. T. N. Pappas and R. J. Safranek, "Perceptual criteria for image quality evaluation," in *Handbook of Image and Video Processing*, A. C. Bovik, ed., pp. 669–684, Academic Press, 2000.
46. A. B. Watson, "The cortex transform: Rapid computation of simulated neural images," *Computer Vision, Graphics, and Image Processing* **39**, pp. 311–327, 1987.
47. W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Analysis and Machine Intel.* **13**, pp. 891–906, Sept. 1991.
48. E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multi-scale transforms," *IEEE Trans. Inf. Th.* **38**, pp. 587–607, Mar. 1992.
49. E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proc. ICIP-95, vol. III*, pp. 444–447, (Washington, DC), Oct. 1995.
50. R. J. Safranek and J. D. Johnston, "A perceptually tuned sub-band image coder with image dependent quantization and post-quantization data compression," in *Proc. ICASSP-89*, **3**, pp. 1945–1948, (Glasgow, Scotland), May 1989.
51. A. B. Watson, "DCT quantization matrices visually optimized for individual images," in *Human Vision, Visual Proc., and Digital Display IV*, J. P. Allebach and B. E. Rogowitz, eds., **Proc. SPIE, Vol. 1913**, pp. 202–216, (San Jose, CA), Feb. 1993.
52. R. J. Safranek, "A JPEG compliant encoder utilizing perceptually based quantization," in *Human Vision, Visual Proc., and Digital Display V*, B. E. Rogowitz and J. P. Allebach, eds., **Proc. SPIE, Vol. 2179**, pp. 117–126, (San Jose, CA), Feb. 1994.
53. D. L. Neuhoff and T. N. Pappas, "Perceptual coding of images for halftone display," *IEEE Tr. Im. Proc.* **3**, pp. 341–354, July 1994.
54. R. Rosenholtz and A. B. Watson, "Perceptual adaptive JPEG coding," in *Proc. ICIP-96, vol. I*, pp. 901–904, (Lausanne, Switzerland), Sept. 1996.
55. I. Höntsch and L. J. Karam, "Apic: Adaptive perceptual image coding based on subband decomposition with locally adaptive perceptual weighting," in *Proc. ICIP-97, vol. I*, pp. 37–40, (Santa Barbara, CA), Oct. 1997.
56. I. Höntsch, L. J. Karam, and R. J. Safranek, "A perceptually tuned embedded zerotree image coder," in *Proc. ICIP-97, vol. I*, pp. 41–44, (Santa Barbara, CA), Oct. 1997.
57. M. M. Chang, A. M. Tekalp, and M. I. Sezan, "Bayesian segmentation of MR images using 3-D Gibbsian priors," in *Proc. SPIE, Image and Video Processing*, **1903**, pp. 122–133, 1993.
58. N. Graham, "Non-linearities in texture segregation," in *CBIA Foundation Symposium*, G. R. Bock and J. A. Goode, eds., **184**, pp. 309–329, New York: Wiley, 1994.
59. N. Graham and A. Sutter, "Spatial summation in simple(fourier) and complex(non-fourier) texture channels," *Vision Research* **38**, pp. 231–257, 1998.
60. N. Graham and A. Sutter, "Normalization: contrast-gain control in simple(fourier) and complex(non-fourier) pathways of patter vision," *Vision Research* **40**, pp. 2737–2761, 2000.
61. 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, The MIT Press, Cambridge, MA, 1993.



(a) (b) (c)

(e) (d) (f)

Figure 8. Image Segmentation based on Steerable Decomposition. Edges are imposed on original images. Original images dimmed where necessary for the edges to be visible. Parameters used are texture window 17x17, β=0.1.



(a) (b) (c)

(e) (d) (f)

Figure 9. Image Segmentation from JSEG.[11] Least merge setting used.