MULTI-LEVEL OBJECT DESCRIPTION: COLOR OR TEXTURE

Pınar Duygulu, Abdurrahman Çarkacıoğlu, Fatoş Yarman-Vural
Middle East Technical University, Department of Computer Engineering,
Ankara, Turkey, TR-06531
{duygulu, carkaci, vural}@ceng.metu.edu.tr

ABSTRACT

In this study, we propose two descriptors on color and texture for content based segmentation. The descriptors are called "the most similar colors in the same neighborhood" and "the most similar textures in the same neighborhood", respectively. These descriptors are applied on a multi-level segmentation algorithm proposed by the authors of this study [1]. The segmentation algorithm decomposes the image either based on the color or texture descriptor. Two multi-level trees are generated, each of which consists of homogenous regions with respect to color or texture respectively. The levels of each tree correspond to the complete image of various degrees of detail. While extracting the objects at different sizes and detail, the multi-level tree representation of image avoids finding a universal threshold for image segmentation. By using the proposed color and texture descriptors, the images are segmented into meaningful objects. Various properties of the descriptors are compared and the findings are discussed.

1. INTRODUCTION

Object representation by a set of descriptors is a challenging problem for the content-based storage and retrieval of image databases. In order to get the semantic information, meaningful objects should be extracted from the image and the relations between them should be defined. This requires a robust segmentation method based on color or/and texture similarity, at the first place.

Segmentation is an old and difficult problem, which requires the selection of a set of similarity measure together with a threshold value for each measure. Since the concept of similarity is highly dependent on the image content, the descriptors and their thresholds to define the homogenous regions cannot be fully identified before the identification of the image content. On the other hand, the goal of segmentation is to identify the image content. This infinite loop of ambiguity can only be resolved by empirical trial and error method for selection of a "suitable" set of descriptors and their thresholds for each application domain.

Multi-level segmentation algorithm proposed in [1], handles the threshold problem for a given descriptor. The proposed algorithm generates a stack of images with homogeneous regions. The generated stack of images is ordered in a tree hierarchy, avoiding the use of an empirical segmentation threshold. At the bottom leave of the tree, there are the regions with exactly same color and/or texture, which are homogeneous according to the selected descriptor, and at the root of the tree the image is reduced to a single plane. As the levels approach to the root of the tree the most similar color or texture descriptors merge the neighboring regions of the previous level.

Color, texture and/or boundaries can be used for the homogeneity criteria in multi-level segmentation algorithm. Color is one of the widely used descriptors, since it is easily recognizable and the human visual system is capable of recognizing very large number of them. In [1], we used a color descriptor for our multi-level image segmentation algorithm. However, for highly textured or gray-scale images color is not sufficient for the segmentation of objects. Texture descriptors should also be somehow incorporated into the segmentation algorithm.

Gabor Filters are the most popular and successful descriptors for texture similarity problems. Gabor Filters with varying size and orientations are more appropriate for representing the texture. The mean and variance of the filters are used as texture descriptors. Using Gabor Filter descriptors it is possible to find the most similar textures in the same neighborhood. Thus, the descriptors can be used in the multilevel segmentation algorithm for generating a multi-level tree with the nodes homogenous with respect to texture.

The color descriptors are relatively simpler than the texture descriptors. Because; texture requires a collection of neighboring pixels for description. Therefore, as the window size for texture descriptor grows the segmentation resolution decreases, whereas the small window sizes reduce the efficiency of the texture descriptors. On the other hand, the initial regions are pixels with the same color and the regions grow according to the similarities defined by the Euclidean distance in ***Lab*** color space. Each pixel can be associated to a color. However, this is not valid for texture information. A single pixel does not give the textural information. The texture information of a pixel can be obtained from a group of pixels surrounding it. The texture class, which the pixel belongs, can only be captured if the size of the surrounding region is large enough. Thus, the multi-level segmentation algorithm using color descriptor cannot be directly applied to texture descriptor. Instead of selecting pixels having the same color as the initial homogeneous regions, multi-level segmentation algorithm is adopted for texture descriptor, so that regions with a fixed window size is selected and the most similar regions according to their texture features are grouped.

2. MULTI-LEVEL REPRESENTATION

In this section, we introduce the multi-level segmentation algorithm based on "the most similar colors in the same neighborhood" and "most similar textures in the same neighborhood".

2.1.1 Color Descriptor

Let the Euclidean distance between two neighboring pixels in the ***Lab*** color space be

$$E(k,l)= \sqrt{L^2(k,l) + a^2(k,l) + b^2(k,l)}$$

Given an image, two neighboring pixels in 4-neihborhood forms a homogeneous region R, if they have the closest colors in the image palette with respect to he Euclidean distance in ***Lab*** color space. Mathematically speaking,

$E(k,l) < E(m,n), \forall (k,l) \in \eta$, then $(k,l) \in R$,

where η indicates all he pixels in the image in 4-neighborhood.

In other words, two neighboring pixels are merged into a homogeneous region, when the Euclidean distance between the colors of them has the smallest possible value in the color palette of the image. This descriptor is called as "the closest colors in the same neighborhood".

2.1.2 Texture Descriptor

Texture descriptor proposed in this study is based on the Gabor Filter function, which can be defined as follows:

$$G(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right]\cdot \exp[2\pi jWx]$$

where σ_x and σ_x indicates the filter variance in x and y directions and W is the window size of the texture. In this study, we use Gabor Filters with 4 different sizes and 6 orientations. Note that there are total of 24 filters, thus 24 means and 24 variances are evaluated for each region. Therefore a region is represented by 48 real number (24 means+24 variances). Numerical studies indicate that Gabor filter looses its power when the window size is less then 8x8. Since we require the smallest possible window size for segmentation, the smallest window size is selected as 8x8 and gradually increased in the study.

Merging strategy for the texture descriptor can be summarized as: At any level, merge two regions when

- Regions are spatially connected.
- Regions have the most similar texture.

Normalized Euclidean distance is used as a texture similarity metric. The textural distance between region T_l and region T_k can be given as

$$TD(T_l, T_k) = \sum_{i=1}^{48} \frac{|T_{l,i} - T_{k,i}|}{\sigma_i} ,$$

where $\mathbf{T}_{l,i}$ and $\mathbf{T}_{k,i}$ are the entries of the descriptor vector which consists of the mean and variances in region *l* and *k*, respectively. It can be argued that different sizes of regions effects the performance of the texture descriptor. This is particularly true when the texels are bigger than the filter size. However, this argument is not valid when the texels are smaller than the filter size. Adaptive Gabor Filter can be defined that cope with the above problem.

2.1.3 Multi-Level Segmentation Algorithm

By using the above descriptors it is possible to generate a stack of images $I_{1, \ldots\ldots ,}$ I_M starting from the original image and iteratively growing the regions in the image I_j, from the homogeneous regions of image I_{j-1}. In the initial image, the neighboring pixels with the same color automatically forms the homogeneous regions if color is used as the descriptor. If the texture descriptor is used, since the necessary information about the texture cannot be gained from a single pixel, initial regions are selected as the windows with a fixed size.

Two trees are independently generated based on the color and texture descriptors. The lowest levels of these trees consist of nodes of equal-color pixels or smallest size texture windows. Let, $\{R_{i,j}\}$, i=1,....,N, be a set of homogeneous regions in level j of a tree. Then, an image at this level can be represented as mutually exclusive and collectively exhaustive set of homogeneous regions as

$$I_j = \bigcup_{i=0}^{N} R_{i,j} , \qquad \text{for } j=1,.....,M$$

Note that $R_{i,j-1} \subseteq R_{i,j}$ for j=1,......,M.

3. COMPARISON EXPERIMENTS

The experiments are performed on an image database with images of various texture and color contents. In the experiments, we applied color and texture descriptors, separately. Image:I, II and III indicate sample images for this paper. The sample images are selected among many images with high (Image:I), medium (Image:II) and low (Image:III) texture content.

The multi-level trees are formed with image stacks for each descriptor. Due to the space limitations, only the top 8 levels of the stacks are shown for each image, in figures 1, 2 and 3. Note that, at each level, the number of regions is decreased by one, starting from the lowest level with all of the pixels in the image and ending at the root with one regions.

Image:I Image:II Image:III

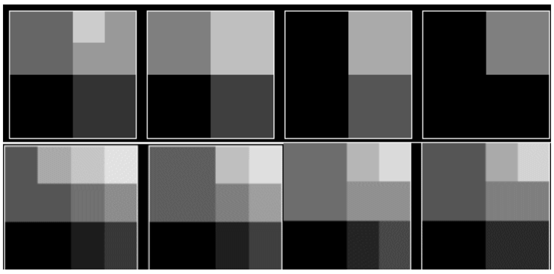
Image:I is artificially generated by combining four textures, while Image:II and III are colored..

3.1. Multi-Level Segmentation with Texture

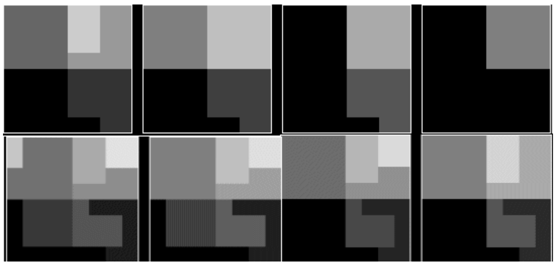
Multi-Level Segmentation algorithm with texture descriptor is applied to each image. Resulting top 8 image stacks are shown in figure 1, 2 and 3. Three different window sizes, 8x8, 16x16 and 32x32 are used for determining the Gabor Filter mean values and variances.

Texture segmentation of Image:I is shown in Figure 1.a, b and c, for various window sizes. The best representation is achieved, when the window size is taken as 32x32. For smaller window sizes the Gabor filter cannot capture the characteristics of distinct textures.

Window Size=32x32



Window Size=16x16



Window Size=8x8

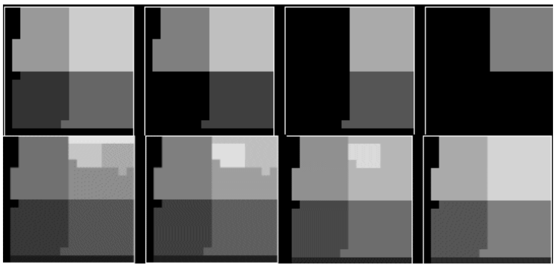
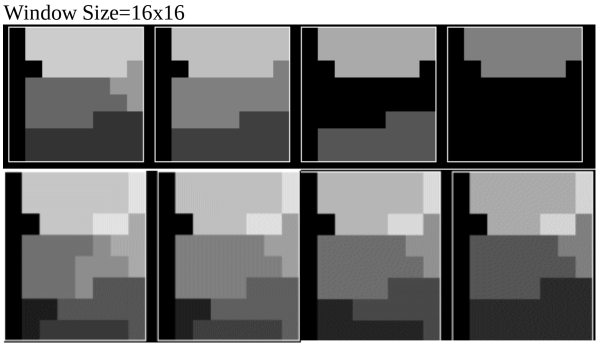


Figure 1. Top 8 levels of the stack for Image:1.

Window Size=16x16



Window Size=8x8

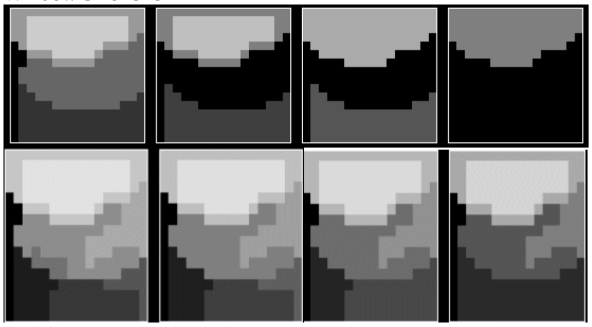
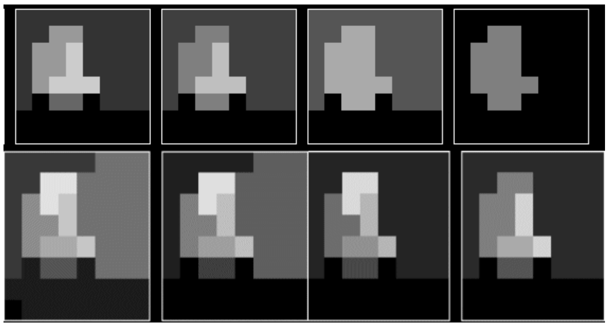


Figure 2. Top 8 levels of the stack for Image:2.

Window Size=16x16



Window Size=8x8

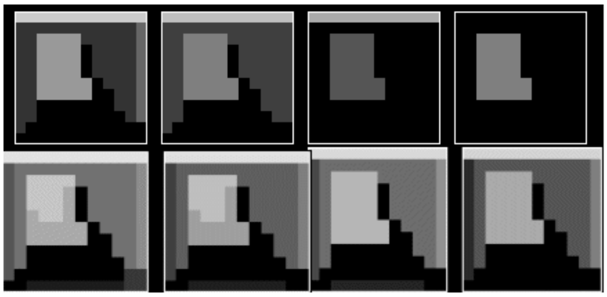


Figure 3. Top 8 levels of the stack for Image:3.

Multi-Level segmentation with texture descriptor algorithm cannot always produces the best segmentation as it is seen from the segmentation of images II and III. Figure 2.a, b and c indicate the first 8-level of the image stack. Although at level 1, the object and the background are successfully identified, other levels cannot be regarded as a good segmentation. This is basically because of the window size. Even for the smallest window, the segmentation is too curse to capture the fine boundary of the object.

The window size makes an important difference when the objects with homogenous regions are smaller then the window. The effect of the window size on segmentation is demonstrated in Figure 2 and Figure 3. When the window size is relatively large, the segmentation can only capture the big objects. Small objects are automatically eliminated for these cases. On the other hand, when the window size is relatively small, small object can be captured easily. However, this time, the objects with large texture primitives are missed.

3.2. Multi-Level Segmentation with Color

Multi-Level Segmentation algorithm with color descriptor is applied to the same sample images. Resulting top 8 levels of stack of images are shown for each case.

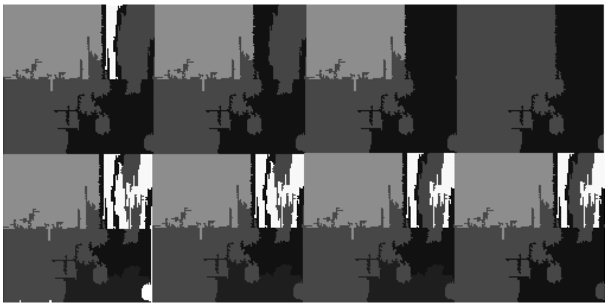


Figure 4. Top 8 levels of a Multi-Level Segmentation Stack of Image:I using color.

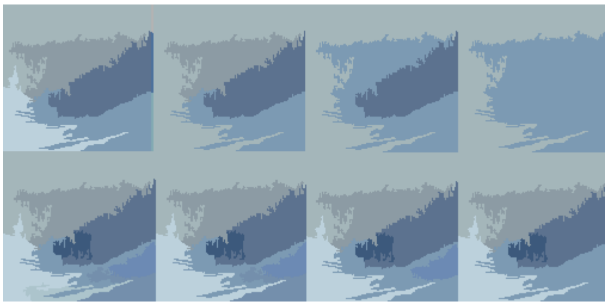


Figure 5. Top 8 levels of a Multi-Level Segmentation Stack of Image:II using color.

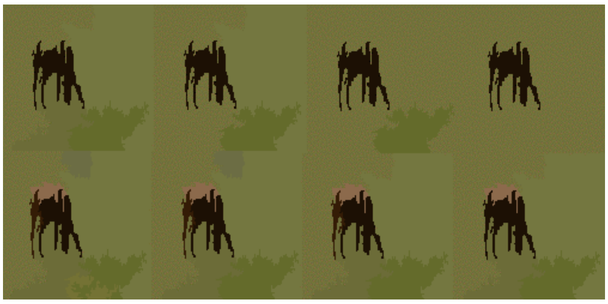


Figure 6. Top 8 levels of a Multi-Level Segmentation Stack of Image:III using color.

As it is seen from the Figure 4, Multi-level segmentation algorithm with color descriptor cannot segment Image:I as good as texture descriptor do. Comparison of segmentation of Image:II with respect to color and texture descriptors indicates slightly better result with texture segmentation. However, Image:III is better when segmented by color descriptor, when comparing the texture approach.

4. DISCUSSION

Content-based image description for storage and retrieval requires appropriate utilization of the color, texture and shape information related to the content of the images. The multi-level segmentation scheme avoids the empirical selection of a threshold in segmenting the image into meaningful objects. Merging pixels by just looking its color distance may not be appropriate when considering textured images or images having only the gray values. Thus, textural information should be employed to this schema. The best results can be obtained by the combination of color, texture and/or boundary information. However, the combination of different descriptors is not easy and requires the observation of their separate performances. In this study, we applied color and texture descriptors separately and compared the results of segmentation based on the proposed descriptors..

Although, individual pixels hold color information, texture information cannot be obtained from a single pixel. Texture features is comparable when the window size is fixed. Selection of this fixed window size is a very difficult and important problem. The choice of window size affects capturing the textural information versus the quality of the segmentation in reverse directions. When the window size is small enough, the region boundaries can be captured accurately. However, regions with large textural information cannot be captured if the window size is small. One alternative to avoid this problem is to choose a small window size at initial levels. Then, at higher levels, the window size can be gradually increased to capture the large areas with texture.

The single use of color descriptor is quite successful when the image has uniform regions with no texture. Thus, a pre-processing can be applied in order to find out the degree of texture in the image content. Texture descriptor can then be applied if the image is reasonably textured.

5. REFERENCES

[1] Y.Xu, P.Duygulu, E.Saber, M.Tekalp, F.T. Yarman-Vural, "Object Based Image Retrieval Based On Multi-Level Segmentation", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2000), June 2000.

[2] Y.Rui, T.S.Huang, S.Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues", *Journal of Visual Communication and Image Representation*, 10, 39-62 (1999)

[3] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and retrieval of image data" IEEE Trans. On PAMI, Vol. 18, No.8, pp 837-842, Aug 1996

[4] L.Shapiro, G.Stockman, A new Computer Vision Textbook, Chapter 8, http://www.cs.washington.edu/education/courses/576