An Effective Content-based Visual Image Retrieval System

Xiuqi Li[1], Shu-Ching Chen[2*], Mei-Ling Shyu[3], Borko Furht[1]
[1] *NSF/FAU Multimedia Laboratory, Florida Atlantic University, Boca Raton, FL 33431*
[2] *Distributed Multimedia Information System Laboratory*
School of Computer Science, Florida International University, Miami, FL 33199
[3] *Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33124*

Abstract

In this paper, an effective content-based visual image retrieval system is presented. This system consists of two main components: visual content extraction and indexing, and query engine. Each image in the image database is represented by its visual features: color and spatial information. The system uses a color label histogram with only thirteen bins to extract the color information from an image in the image database. A unique unsupervised segmentation algorithm combined with the wavelet technique generates the spatial feature of an image automatically. The resulting feature vectors are relatively low in dimensions compared to those in other systems. The query engine employs a color filter and a spatial filter to dramatically reduce the search range. As a result, queue processing is speeded up. The experimental results demonstrate that our system is capable of retrieving images that belong to the same category.

Keywords: Content-Based Image Retrieval, Multimedia Systems.

1. Introduction

The research in Image Retrieval began in the 1970s [1]. Initially, a text-based approach was adopted. In this approach, human first manually annotates each image using keywords, and then images are retrieved based on the keywords in the text annotation. There are two main disadvantages in this approach. One is that it requires a huge amount of human labor in the manual annotation when the image collection is large. The other one is that it is hard to precisely annotate the rich content of an image by humans due to perception subjectivity [1][2]. The text-based approach remained popular until early 1990s when many large-scale image collections emerged and the drawbacks of text-based approach became more and more notorious. A new content-based approach was then proposed and the research in content-based image retrieval has been active since then. In the content-based approach, images are retrieved directly based on their visual content such as color, texture, and shape [1][2]. Typically, a content-based image retrieval system consists of three components: feature extraction, feature indexing and retrieval engine. The feature extraction component extracts the visual feature information from the images in the image database, the feature indexing component organizes the visual feature information to speed up the query processing, and the retrieval engine processes the user query and provides a user interface [1][2].

A large number of content-based image retrieval systems have been built [1] such as QBIC [3], VisualSEEK [4], and Photobook [5]. In the QBIC system, content-based queries such as query by example image, query by sketch and drawing, and query by selected color and texture patterns are supported. The visual features include color, texture, and shape. Color is represented using a k-bin color histogram. Texture is described by an improved Tamura texture representation. Shape information includes area, circularity, eccentricity, major axis orientation, and moment invariants. KLT is used to reduce the dimension of the feature vectors and R* tree is the indexing structure. The later version integrated text-based query [1]. In the VisualSEEK system, both content-based query (query by example image and spatial relation pattern) and text-based query are supported. The system uses the following visual features: color represented by color set, texture based on wavelet transform, and spatial relationship between image regions. A binary tree is used to index on feature vectors [1]. The Photobook system is composed of a set of interactive tools for browsing and searching images. It supports query by example. The images are organized in three subbooks from which shape, texture, and face appearance features are extracted respectively [1][5].

* This research was supported in part by NSF CDA-9711582.

The differences between our system and the previous systems are in feature extraction and query strategy. For feature extraction, we propose a color label histogram to extract global color information. We quantize the color space into thirteen bins by categorizing the pixel colors into thirteen categories. The resulting color histogram is effective and efficient to obtain objects with similar colors. The spatial information of an image is automatically obtained using a unique unsupervised segmentation algorithm in combination with the wavelet technique. Our query strategy includes a color filter and a spatial filter, which greatly reduces the search range and therefore speeds up the query processing.

The rest of the paper is organized as follows. The system architecture is presented in Section 2. This section consists of four subsections, which describe color and spatial information extraction, feature indexing, similarity measure and query strategy. In Section 3, the experimental results are presented and discussed. The conclusion and future work are given in Section 4.

2. The image retrieval system

The architecture of our system is shown in Figure 1. There are two main components in the system. The first component is the visual content extraction and indexing. Each image in the image database is analyzed and the color and spatial information are generated using the color label histogram computation algorithm and the unsupervised segmentation algorithm respectively. The obtained features are stored in a feature database and organized in an efficient way for query retrieval. The second component is the query engine. It consists of a query user interface and a query processing subcomponent. Query by example image is supported in the system. When a user issues a query through the query user interface, the query processing subcomponent computes the similarity measure between the query image and each image in the search range. Two filters, the color filter and the spatial filter, are used to reduce the search range. The top N images similar to the query image are displayed in the query user interface.

2.1. Feature extraction and indexing

Visual features must be extracted before images are retrieved. In our system, the color feature, represented by a 13-bin color label histogram, is computed. The spatial information, which is represented by class parameters, is obtained by applying an unsupervised segmentation algorithm combined with the wavelet technique to images.

2.1.1. Color extraction. The color feature is the most widely used visual feature in image retrieval because it is more robust to changes due to scaling, orientation, perspective and occlusion of images [2]. Humans perceive a color as a combination of three stimuli, R (red), G (Green), and B (Blue), which form a color space. Separating chromatic information and luminance information can generate more color spaces. To extract color information, a color space must be chosen first. There exist many color spaces. Examples are RGB, YIQ, YUV, CIE LAB, CIE LUV, HSV and its variants. None of them can be used for all applications [1][2][6][8][9][13]. RGB is the most commonly used color space primarily because color image acquisition and recording hardware are designed for this space. However, the problem of this space is the close correlation among the three components, which means that all three components will change as the intensity changes. This is not good for color analysis. YIQ and YUV are used to represent the color information in TV signals in color television broadcasting. Y encodes the luminance information and UV or IQ encodes the chromatic information. CIE LAB and CIE LUV are often used in measuring the distance between two colors because of its perceptual uniformity. That is, the Euclidian Distance between two colors represented in these two spaces matches the human perception. However, its transformation from the RGB space is computationally intensive and dependent on a reference white. H (Hue) S (Saturation) V (Value) and its variants are perceptual color spaces, while all the previous color spaces are not. By perceptual, we mean that the three components (H, S, and V) represent the color attributes associated with how human eyes perceive colors. Hue, which corresponds to the dominant wavelength of a given perceived color stimulus, represents the type of the color such as red, blue, and green. The strength, purity, or richness of a color is represented by Saturation. The color is perceived to be less saturated as more white light is added to it. Value (or intensity) is the amount of light perceived from a given color sensation. White and black are perceived as the maximum and minimum intensity, respectively [6]. In our system, the HSV color space is chosen for two reasons. First, it is perceptual, which makes HSV a proven color space particularly amenable to color image analysis [6][8][9]. Second, the benchmark results in [2] show that the color histogram in the HSV color space performs the best.

Many schemes, such as color histogram, color moments, color coherence vector, and color autocorrelogram, can be used to describe the color information in an image. Color histogram is the most widely used method since it is more robust to changes due to scaling, orientation, perspective, and occlusion of images [2]. Color histogram represents the joint distribution of three color channels in an image.

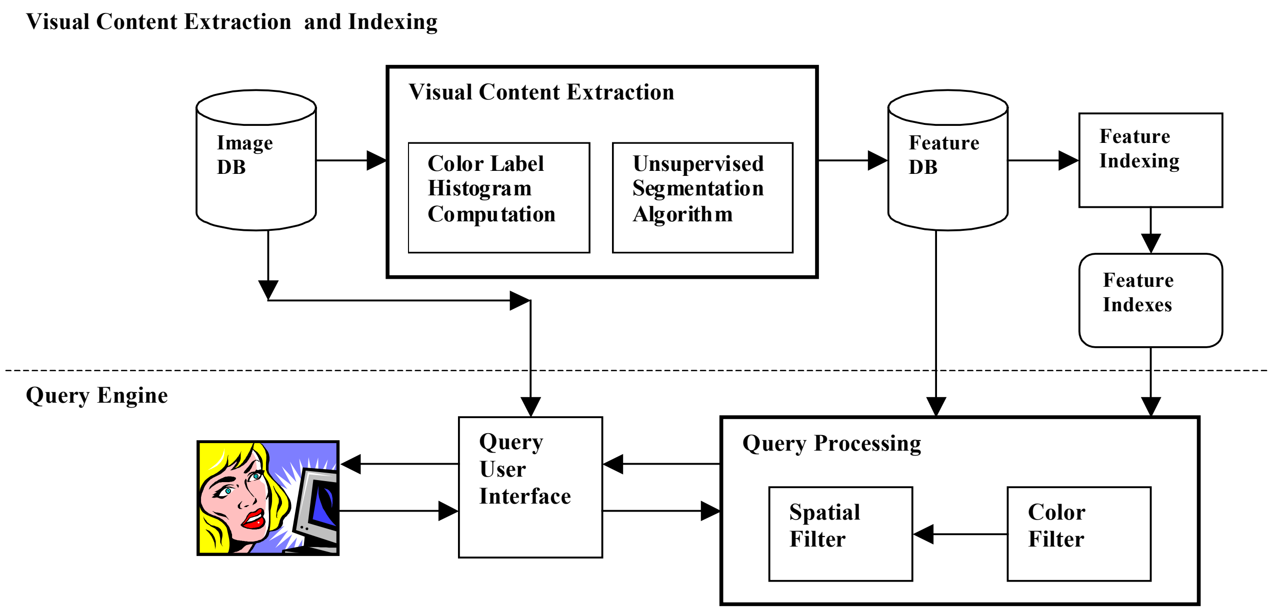

Figure 1. The system architecture

Therefore, it characterizes the global color information in an image. Color moments are the first few low-order moments of each color channel. It is a compact representation of the color distribution of an image. Color coherence vector is designed to take into account of the spatial distribution of color in an image. It is obtained by partitioning each histogram bin into two: one with coherent pixels and the other with incoherent pixels. Color autocorrelogram represents the probability of finding a pixel of some color at some distance from a pixel of the same color in an image. It characterizes both the global and spatial distribution of the color. In the performance evaluation experiments in [2], it is shown that the color histogram runs much faster than the color coherence vector and color autocorrelogram, performs almost as good as the color coherence vector, and does not perform much worse than the best color autocorrelogram. Therefore, color histogram is used in our system [1][2].

Because there are many different colors, to reduce the complexity in histogram computation, the color space needs to be quantized [2]. In our system, the color space is quantized through color categorization. All possible colors of the pixels are first classified into thirteen categories based on the H, S, and V value ranges. Each category is identified by an ID, and then each pixel is labeled as the ID of the category to which it belongs. Next, a color label histogram is built. The resulting color label histogram is computationally efficient and effective to obtain objects with similar colors. In addition, it reduces the dimension of the color feature vector.

The author in [6] used twelve categories, which are obtained from the experimental result based on the H, S, and V value ranges, to represent the dominant colors of color regions in an image. These twelve categories are black, white, red, bright red, yellow, bright yellow, green, bright green, blue, bright blue, purple, and bright purple. The Hue is partitioned into 10 color slices with 5 main slices (red, yellow, green, blue, and purple) and 5 transition slices. Each transition slice is counted in both adjacent main slices. In our approach, some modifications are made to compute the color histogram. Firstly, the difference between the bright chromatic pixels and the chromatic pixels is ignored to reduce the total number of bins. Therefore, bright red and dark red are considered to be in the same color category. Secondly, the transition color slices are considered as separate categories for histogram computation. Thirdly, a new category "gray" is added to consider all possible value ranges since some images in our image database contain the gray color. Hence, there are totally thirteen color categories, which are white, black, gray, red, red-yellow, yellow, yellow-green, green, green-blue, blue, blue-purple, purple, and purple-red.

2.1.2. Spatial information extraction. The spatial information is represented by the class parameters a_{nj}, where n is the class id and j is the parameter id. It is extracted by the unsupervised segmentation (SPCPE) algorithm [7][10][11][12], which partitions a gray-scale image into s regions that are mutually exclusive and totally inclusive. In the algorithm, a region is considered as class. In each class, there exist one or more segments that are similar to each other in some sense and may not be spatially adjacent to each other. Therefore, each image is partitioned into s classes and b segments. The SPCPE algorithm regards both the partitions C and the class parameters θ as random variables. It estimates the partition C and class parameters θ jointly using the Bayesian approach. Starting with an initial partition, the simultaneous estimation is performed in an iterative way [7][10][11][12].

In our experiments, we found that different initial partitions can produce very different segmentation results. Therefore, the wavelet decomposition coefficients are used in the initial partition generation for a better segmentation result. The idea is to partition the pixels based on the wavelet coefficient values.

Let Y = $\{y_{i,j},\ i, j = 0, \ldots, M-1\}$ be the image intensity matrix. Assume there are 2 classes, whose probability densities are $p_1(y_{ij})$ and $p_2(y_{ij})$, respectively. The algorithm assumes that the pixels in the same class cluster around a 2D polynomial function are given as:

$y_{ij} = a_{n0} + a_{n1}\, i + a_{n2}\, j + a_{n3}\, ij$, for $\forall\ (i,j)$ such that $y_{ij} \in S$n, n = 1, 2, where Sn denotes class n and $a_{n0} \sim a_{n3}$ are the class parameters for class n. Let $C = \{c_1, c_2\}$ be the partition variable, and $\theta = \{\theta_1, \theta_2\}$ be the class parameters with $\theta_n = (a_{n0}, a_{n1}, a_{n2}, a_{n3})^T$. The algorithm estimates $\boldsymbol{C}$ and θ as that which maximizes the *a posterior probability* of the partition variable and class parameter variable given the image data Y, denoted as $(\hat{c}, \hat{\theta})_{MAP}$.

$$(\hat{c}, \hat{\theta})_{MAP} = Arg \max_{(C,\theta)} P(c, \theta \mid Y) = Arg \max_{(C,\theta)} P(Y \mid c, \theta) P(c, \theta)$$

Under some reasonable assumptions and by using mathematical transformation, the previous equation then becomes:

$$(\hat{c}, \hat{\theta})_{map} = \underset{(C,\theta)}{Arg \max} P(Y \mid C, \theta)$$

$$= \underset{(C,\theta)}{Arg \min} J(C_1, C_2, \theta_1, \theta_2)$$

$$J(C_1, C_2, \theta_1, \theta_2) =$$

$$\sum_{y_{ij} \in C_1} - \ln p_1(y_{ij}; \theta_1) + \sum_{y_{ij} \in C_2} - \ln p_2(y_{ij}; \theta_2)$$

After relabelling, the partition in the current iteration and the previous iteration are compared, the algorithm stops when there is no change between the two partitions. Otherwise, it enters another iteration.

During the initial partition generation, the images are first processed using wavelet at level one to extract salient points in the horizontal, vertical and diagonal subbands. For each wavelet subband, a candidate initial partition is generated by labeling all pixels in the original image that correspond to the salient points in that subband as one class and the rest of the pixels as the other class. This generates three candidate initial partitions. The final initial partition is chosen to be the one with the least cost $\boldsymbol{J}$ from the three candidate initial partitions. Experimental results show that the wavelet technique doubles the precision of the segmentation result that uses the random initial partition generation.

The color feature extracted in our system is a 13-dimension color label histogram vector, and the spatial feature is two 4-dimensional class parameter vectors.

2.2. Query strategy

To compare two images, the similarity/distance between them must be calculated. There are a number of similarity/distance measures for measuring the similarity/distance of different feature vectors. Examples are *L_1-Distance*, *L_2-Distance*, and *Quadratic Distance Metric* [2].

In our system, the color label histogram and the class parameters are used for image comparison. *L_1-Distance* is chosen for measuring the distance between two color label histogram vectors because it is commonly used in histogram comparison. The *L_1-Distance* between two color label histograms of the query image q and the i^{th} image in the image database can be represented by the following formula [2]:

$$D_{color}^{(q,i)} = \sum_{j=1}^{M} \left| H_j^{(q)} - H_j^{(i)} \right|,$$

where H_j is the j^{th} bin and M is the total number of bins.

The *L_2-Distance (*also called *Euclidian Distance)* is used to compare the class parameter vectors because the parameters in each class are assumed to be independent. The Euclidian Distance between the class parameters of the query image q and that of the i^{th} image in the database is:

$$D_{spatial}^{(q,i)} = \sum_{n=1}^{2} \sqrt{\sum_{j=0}^{3} \left(a_{nj}^{(q)} - a_{nj}^{(i)} \right)^2},$$

where n refers to the class n and a_{nj} refers to the j^{th} class parameter for class n.

To rank the images in the database based on the measurement of their similarity to the query image, the total distance in both color and spatial information should be used. Because the distance value in color and the distance value in spatial information may be at different scales, normalization is required. In our system, the ratio of the color and spatial information to their corresponding maximum distances is used. Therefore, the total distance is given as:

$$D^{(q,i)} = D_{color}^{(q,i)} / Max(D_{color}^{(q,i)}; i = [1, K]) + D_{spatial}^{(q,i)} / Max(D_{spatial}^{(q,i)}; i = [1, K])$$

where K is the total number of images in the image database.

The query engine in our system employs the idea of filtering to reduce the search ranges at different stages so as to speed up the query processing. Two filters are used. One is the color filter, and the other is the spatial filter. The color filter uses small thresholds to filter out images in the database that are dissimilar to the query image in color and therefore need not be compared in the second stage. The color filter effectively eliminates around eighty percent of the images in the database from the search range for the later stage. The images passing the color filter are sorted based on the color distances between them and the query image. In the second stage, the spatial filter computes the distance in spatial information between the query image and those images that pass the color filter. A threshold is also used in the spatial filter to eliminate those images that are similar to the query image in color and but are dissimilar to the query image in spatial information. This avoids the unnecessary computation time at the third stage. About fifty percent of images passing the color filter are removed by the spatial filter. The images in the search range are not sorted at the second stage because the final ranking is not based solely on the distance in spatial information. At the last stage, the total normalized distances between the images passing the two filters and the query image are calculated. The images in the query result are then sorted based on the total normalized distances. The top six or less images similar to the query image are displayed in the query user interface as the final query result.

3. Experimental results

The image database in our current system contains 500 images that are downloaded from yahoo (www.yahoo.com) and corbis (www.corbis.com). The image categories are diverse. There are images with objects in the sky, in the water or ocean, or on the grass, images with green trees or plants, images with mountains under different time periods (daytime, sunset and nighttime) or different weather situations (cloudy and sunny), etc. All images are of size 256x192.

Figure 2 is the query result for Image 1. The top row is the query image. The images listed in the next two rows are the top six images returned by the query. These images are displayed based on their ranks. Clearly all six images contain major color blue and objects under a blue background. Besides, the top three images and the image in Rank 5 are similar to the query image in a semantic sense in that all of them contain objects under the blue sky. The images in Rank 4 and Rank 6 are also similar to the query image because they contain objects in the ocean. The ocean and the sky are all blue colors.

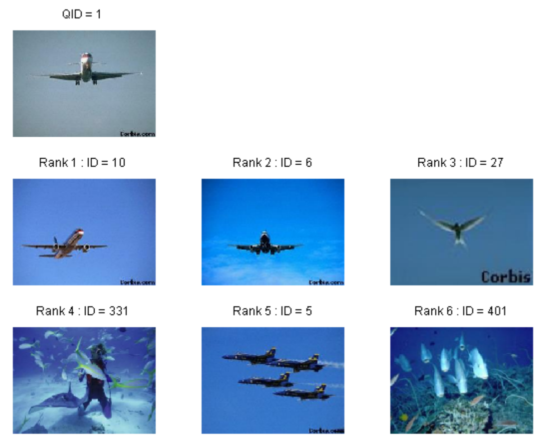

Figure 2. The query result of Image 1

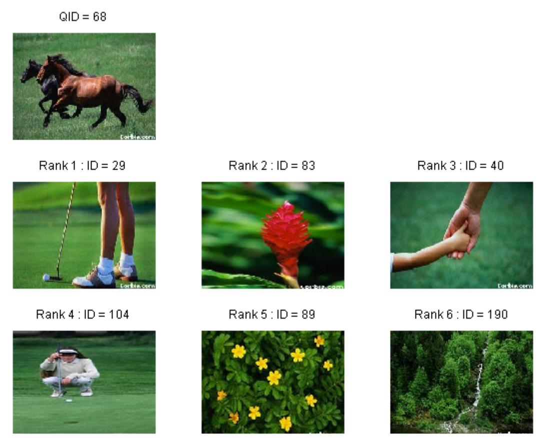

Figure 3. The query result of Image 68

The query result of Image 68 is shown in Figure 3. The images are displayed in a similar way as in Figure 2. The query image is in the first row and the top six images are listed based on their ranks in the second row and the third row. It's very easy to see that all images contain green grass or green plants. The top four images are similar to the query image semantically because all of them contain objects in the green background.

Figure 4 shows the query result of Image 232. The query image and the top images with their ranks and IDs are displayed in the same manner as those in the previous figures. Only four images are returned by the query. We can easily identify the objects under the blue sky or the blue ocean. Therefore they are similar to each other in a semantic way. Moreover, all objects are located in the similar position in the query image and the top four images.

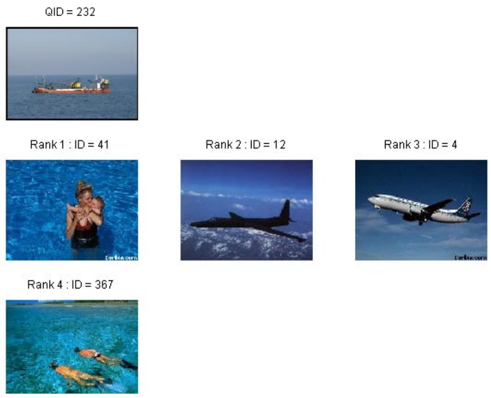

Figure 4. The query result of Image 232

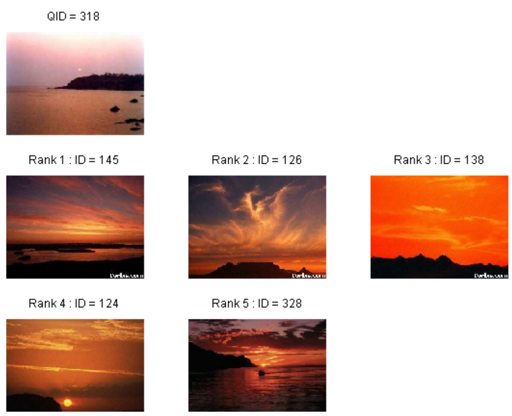

Figure 5. The query result of Image 318

The query result of Image 318 is illustrated in Figure 5. Similarly, the query image and the top images with their ranks and IDs are displayed in the figure. It can be seen from this figure that all images contain mostly the red and black colors, and therefore they are similar to each other in colors. In addition, all images are in the same category because all of them are mountains at sunset and do not contain any clear objects under a clear background.

4. Conclusion and future work

In this paper, we propose an effective content-based image retrieval system that consists of the visual content extraction and indexing component and the query engine component. There are three differences between our system and other systems. First, a color label histogram with only thirteen bins is used in our system. It effectively and efficiently describes the global color information by classifying the pixel colors into a small number of color categories. Second, a unique unsupervised segmentation algorithm in combination with the wavelet technique is utilized to automatically extract the spatial information from the images in the database. The dimension of the resulting feature vectors is low, which is good for feature indexing. Third, two filters are used to reduce the search ranges. As a result, query processing is speeded up.

In the future, we plan to add more images to our image database. Wavelet-based texture feature and the SPCPE algorithm with more classes will be integrated into our current system. Machine learning algorithms will also be considered to improve the retrieval precision.

5. References

[1] Y. Rui and T.S. Huang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," *Journal of Visual Communication and Image Representation*, 1999.

[2] W.-Y. Ma and H. Zhang, "Content-Based Image Indexing and Retrieval," *Handbook of Multimedia Computing*, CRC Press, 1999.

[3] M. Flicker, H. Sawhney, etc., "Query by Image and Video Content: The QBIC System," *IEEE Computers,* Sep. 1995.

[4] J.R. Smith and S.-F. Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System," *Proc. of ACM Multimedia 96*, 1996.

[5] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases," *International Journal of Computer Vision*, 1996.

[6] D. Androutsos. *Efficient Indexing and Retrieval of Color Image Data Using a Vector-Based Approach*. Ph.D. Dissertation, University of Toronto, 1999.

[7] S. Sista and R. Kashyap, "Bayesian Estimation For Multiscale Image Segmentation," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* Phoenix, AZ, March 1999.

[8] H.D. Cheng, X.H. Jiang, etc. "Color Image Segmentation: Advances and Prospects," *Pattern Recognition*, 2001.

[9] H.D. Cheng and Y. Sun, "A Hierarchical Approach to Color Image Segmentation Using Homogeneity," *IEEE Transactions on Image Processing*, Dec. 2001.

[10] S.-C. Chen, S. Sista, M.-L. Shyu, and R.L. Kashyap, "An Indexing and Searching Structure for Multimedia Database Systems," *IS&T/SPIE Conference on Storage and Retrieval for Media Databases 2000*, pp. 262-270, 2000.

[11] S.-C. Chen, M.-L. Shyu, and C. Zhang, "An Unsupervised Segmentation Framework For Texture Image Queries," *Proc. of the 25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC)*, pp. 569-573, October 2001.

[12] S.-C. Chen, M.-L. Shyu, C. Zhang, and R.L. Kashyap, "Video Scene Change Detection Method Using Unsupervised Segmentation and Object Tracking," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 57-60, 2001

[13] Y. Alp Aslandogan and C.T. Yu, "Techniques and Systems for Image and Video Retrieval," *IEEE Trans. On Knowledge and Data Engineering*, Vol.11, No.1, Jan/Feb 1999.