COMPLEX WAVELET TRANSFORM-BASED COLOR INDEXING FOR CONTENT-BASED IMAGE RETRIEVAL

Nader Safavian and Shohreh Kasaei

Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
skasaei@sharif.edu

ABSTRACT

With the rapid establishment of digital libraries and multimedia databases, the need for an efficient search algorithm is also increasing. In this paper, a new approach for content-based image indexing and retrieval is presented. The proposed method is based on a combination of multiresolution analysis and color characteristics of the image. Also, in order to obtain better retrieval results, the image texture features are combined with the color features to form a powerful discriminating feature vector for each image. The texture features are obtained with the use of dual-tree *complex wavelet transform* (DT CWT) method. According to the new algorithm, the image is divided into different sublayers, each of which containing only pixels in areas with similar spatial frequency characteristics. Here, we present a computationally efficient algorithm to implement an efficient content-based image retrieval algorithm. Simulation results show the efficiency of the proposed algorithm.

Key words

Color indexing, texture analysis, content-based image retrieval, complex wavelet transform, and histogram.

1. INTRODUCTION

The *Content-based image retrieval* (CBIR) has become very attractive in the past decade. The first important step in every algorithm dealing with this matter is to find a way in describing the content of every image in the database. At first, swain and ballard [1] used simple color histogram to index each image. This method has been shown to be very effective and has become popular in indexing applications due to its low complexity

Color histograms are computationally efficient, and generally insensitive to small changes in camera position. However, a color histogram provides only a very coarse characterization of an image (and the information regarding the spatial positions of the color is not included). Consequently, two images with similar histograms can have dramatically different appearances. When working with large databases, the chance of having (visually) different images with similar histograms increases [2], and consequently the effectiveness of the method will be reduced.

To overcome this difficulty researchers have proposed different approaches. One common approach is to divide the image into several subimages and assign a separate color histogram to each subimage [3]. The problem with that method is that it is computationally expensive and it requires a huge amount of storage memory. Also, this method is vulnerable to translation and rotation of color images. Mandel et al. [4] reduce the computational complexity of color histogram in terms of its moments. In that research it is shown that the *Legendre* moments provide superior retrieval performances when compared to regular moments.

In other approaches, histograms are augmented with local spatial properties. A famous method of this type is the *color coherence vector* (CCV), by Pass and Zabih [5], where the image pixels in a given histogram bin are partitioned into two classes based on their spatial coherence. A pixel is assigned as coherent if it is in a continuous region with the same color and with a size larger than a minimum amount; and otherwise as incoherent. The performance of this method is better than that of the conventional histogram, but it suffers from low computation speed. Huang et al. [6] propose color correlogram which includes the spatial correlation of colors along with the global distribution of local spatial correlation of colors. It outperforms the CCV method but it also suffers from a very high computational cost. Han et al. [7] argue that a conventional color histogram considers neither the color similarity across different bins nor the color dissimilarity in the same bin, and therefore it is sensitive to noisy interference such as illumination changes and quantization error. To address theses concerns, they present a new color histogram representation called *fuzzy color histogram* (FCH), by considering the color similarity of each pixel's color associated to all the histogram bins through a fuzzy-set membership function. The problem with this method is that it is a rather complicated approach. Another problem associated with all the previous works is that they do not consider the way humans judge similarity of the retrieved images. Qiu and Lam [8] used frequency *layered color indexing* (LCI) in their research. They separated every image to different layers according to their frequency content. They used simple Gaussian filters [9] in their work. In our approach, we consider the human visual system properties and use them in proposing a new algorithm for image retrieval. Although color is the most effective cue for image retrieval, incorporating other features (such as texture) will enhance the image retrieval performance. In this paper, we propose a new method in which we take advantage of both color and texture properties of images to improve the performance of the image indexing and retrieval algorithm.

The paper is organized as follows. In Section 2 the details of the proposed technique along with a brief introduction to the dual-tree complex wavelet transform procedure are given. The simulation results are illustrated in Sections **3,** followed by the conclusion statement presented in Section 4.

2. PROPOSED ALGORITHM

The best judge regarding the performance of an image retrieval system is given by the final user; the human. Therefore, in order to propose a *"good"* retrieval system, we should take into account the way a human perceives similarity between images.

According to the frequency analysis theory [10], there are several frequency channels perceived by the human visual system. Each of these frequency channels responses only to a limited bandwidth of the image constructed on the *retina*. Consequently, we can consider them as bandpass filters and it would be convenient to think of the visual system as a filter bank consisting of several filters, where each filter's response covers certain areas of the spatial frequency spectrum.

In an image, a *busy* or *sharp* area consists of high frequency components and a *smooth* area contains lower frequency components. The flat areas can be associated with the interior of the objects or backgrounds while the busy areas can be textured surfaces or object boundaries. Physically, different frequency components in an image may be regarded to different objects or boundaries. As a result, judging the similarity of images by the human visual system can occur as follows: an image is decomposed into different frequency components and the corresponding components of different images are compared. We have based our algorithm on this philosophy. A schematic block diagram of the proposed algorithm is presented in Fig. 1. As can be seen in this figure, we first pass an image through a filter bank. The output of each filter is used to define a separate layer. Each individual layer (which contains pixels with similar frequency distributions) is used for its own index. Then, we combine these features to construct the total feature vector of the image. Also to compare two images, we compare the index of each layer with the index of its counterpart in another image.

To design our algorithm we should consider two important issues. The first is how to implement the filter bank and the second is how to index each layer.

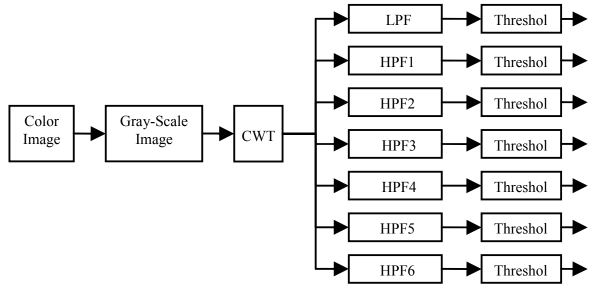

Figure 1. Block diagram of the proposed layer separation stage.

For the filter bank we use the CWT [11] to take advantage of its numerous special properties. In Section 2.1 we represent a brief description of the CWT. Since the information of the busy and sharp areas along with the information of the flat areas is present in the gray-scale components of the image, for the texture analysis stage, we convert the color (RGB) image into a gray-scale image before implementing the CWT onto it. Multiple thresholds are then applied to the outputs of the filters. We obtain the binary images using (1) and the layers using (2).

$$b_k(i,j) = \begin{cases} 1, & if \quad T_{k-1} \le y(i,j) \le T_k \\ 0, & \quad\quad Otherwise \end{cases} \tag{1}$$

$$L_k(i,j) = \begin{cases} x(i,j), & if \quad b_k(i,j) = 1 \\ Empty, & \quad Otherwise \end{cases} \tag{2}$$

2.1. Dual-Tree Complex Wavelet Transform

Kingsbury in his distinguished work introduced the dual-tree complex wavelet transform [11]; which is similar to the Gabor filter but it is orthogonal and also can be computed faster than that. The frequency responses of the CWT are shown in Fig. 2. Just like a typical Gabor filter, there are 6 orientations at each of the 4 scales (the number of scales is arbitrary, but the number of orientations is fixed).

The main advantages of the DT CWT over the real DWT are that the complex wavelets are approximately shift invariant (meaning that the obtained texture features are likely to be invariant to the translations in images) and have separate subbands for positive and negative orientations (6 orientations). Note that the conventional separable real wavelets suffers from the lack of shift invariance, provide just three orientations while having a poor directional selectivity and also cannot distinguish between 45° and -45° directions.

The CWT attains these properties by replacing the tree structure of the conventional wavelet transform with a dual tree. At each scale, one of the trees produces the real part of the dual-tree complex wavelet coefficients, while the other tree produces the imaginary part.

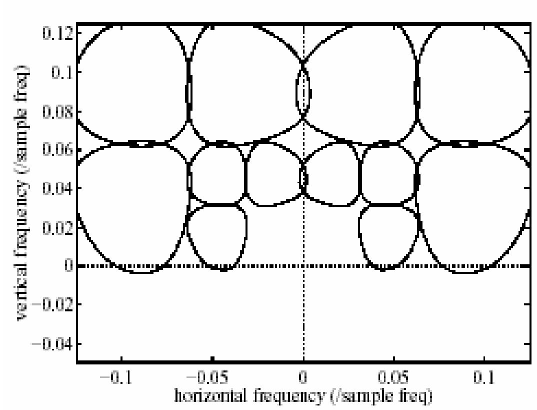

Figure 2. Contours of 70% peak magnitudes of the CWT filters at scales of 3 and 4 (adopted from [11]).

The extra redundancy allows a significant reduction of aliasing terms and causes the complex wavelets to become approximately shift invariant. Translation causes large changes to the phase of the wavelet coefficients, but their magnitude (and hence energy) is much more stable.

By using even and odd filters alternately in these trees, it is possible to achieve overall complex impulse responses with symmetric real parts and antisymmetric imaginary parts. In Fig. 3, the structure of the dual tree CWT is illustrated.

In our proposed method, we extract features of mean, standard deviation and kurtosis of the magnitude of each of the six subimages of the CWT at the first stage as the texture features.

3. EXPERIMENTAL RESULTS

As indicated in Fig. 1, there are one low-pass filter and six high-pass filters in the CWT. We use the low-pass filter to construct a layer corresponding to low frequency distribution components of the image (which relate to the background or interior parts) and six outputs of the high-pass filters to build the layers indicating details (such as boundaries or textured parts).

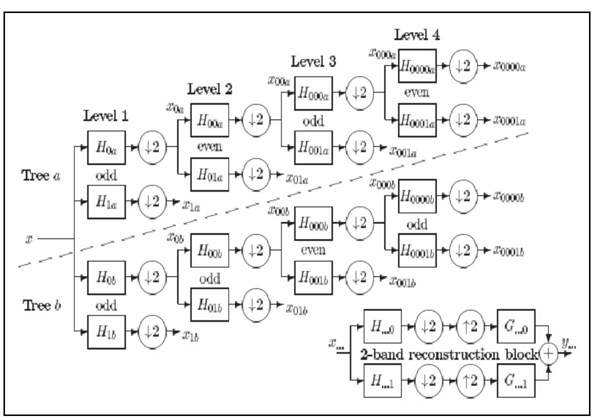

Figure 3. Dual trees of real filters for the implementation of the DT CWT (adopted from [11]).

The threshold values are chosen to be the maximum pixel values of the output filtered images divided by two. Therefore if we scale the maximum pixel value of each subimage to 1, we would set $T_{k-1} = .5$ and $T_k = 1$ in equation (1). To make the algorithm computationally efficient we have used only three high-pass filters instead of six and a 54-bin color quantizer ({H, S, V}: {6, 3, 3}) to build the index of each layer. As such, the feature vector length is 234 ($4\times 54 + 18 = 234$). To compare two histograms, the L_1-norm distance metric is used. If $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ denote the histograms of the two counterpart layers, then their difference would be:

$$D(X,Y) = \sum_i \frac{|x_i - y_i|}{1 + x_i + y_i} \tag{3}$$

3.1. Retrieval Performance Evaluation

Our database consists of images mainly downloaded from: http:// www. cs. washington. edu/ research /imagedatabase /groundtruth/ as well as other images gathered in our laboratory; containing more than 2000 images. Our database is divided into different categories, where each category contains similar and related images. The ground truth answers are also provided. Therefore, each image has an index indicating the ground truth answers.

We have implemented the algorithm using Matlab 6.5 package on a Pentium IV (2 GHz), and have run the algorithms on this database. For each image, the feature extraction stage of our algorithm took only about 2 seconds.

We used three different criteria to evaluate the performance of the algorithm as follows:

$$P_j = \frac{Number\ \ of\ \ retrieved\ \ and\ \ relevant\ \ elements\ \ in\ \ the\ \ first\ \ j\ \ position}{j} \tag{4}$$

$$R_j = \frac{Number\ \ of\ \ retrieved\ \ and\ \ relevant\ \ elements\ \ in\ \ the\ \ first\ \ j\ \ position}{Total\ \ number\ \ of\ \ relevant\ \ elements\ \ in\ \ the\ \ collection} \tag{5}$$

In Table 1, the amounts of these criteria for five different categories contained in our database are listed. Also, we have proposed another recall criterion which is a more appropriate measure. If Q_i is the ith query image and $Q_i(1), Q_i(2), ..., Q_i(N_i)$ are the N_i correct answers to this query, then we propose to use the following recall measure:

$$AR(l) = \sum_i \left(\frac{\left| \{ Q_i(j) | rank(Q_i(j)) < l \} \right|}{N_i} \right) \tag{6}$$

In Fig. 4, the recall performance of the histogram, the correlogram and our method are shown. Figures 5 and 6 show the performance of the proposed algorithm when retrieving the image shown in the top left on the figure.

Table1. Comparison of Different Methods
[C: Correlogram, O: Our method].

Category	P_5	P_{10}	P_{15}	P_{20}	R_5	R_{10}	R_{15}	R_{20}	
Cherries	1	1	1	1	.04	.08	.2	.3	O
	.8	.8	.9	.85	.04	.06	.12	.16	C
Football Field	1	1	1	.9	.21	.46	.78	.81	O
	1	.9	.8	.85	.18	.41	.71	.78	C
Red Rose	1	1	1	1	.12	.28	.29	.46	O
	.8	.7	.8	.8	.11	.2	.3	.39	C
Greenland	1	1	.9	.8	.14	.23	.31	.36	O
	1	.9	.8	.7	.11	.19	.28	.31	C
Ancient Species	1	1	1	.9	.09	.2	.31	.46	O
	1	1	.8	.83	.09	.21	.25	.41	C

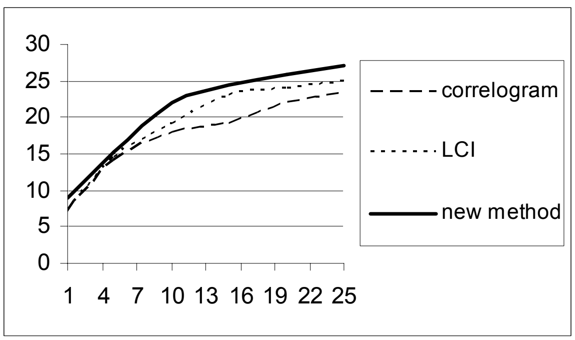

Figure 4. Comparison of three methods according to (6).

4. CONCLUSIONS

In this paper we proposed a new method for content-based image retrieval. We tried to take into account the human visual system properties an incorporated the texture features to obtain better retrieval results. We utilized the dual-tree complex wavelet transform which is shown to be very efficient for extracting texture features from images and we showed its superiority over the Gabor filter. The experimental results show the efficiency of the proposed algorithm.

ACKNOWLEDGEMENT

The authors would like to thank Prof. Nick G. Kingsbury from University of Cambridge for providing us with the Matlab codes of the dual-tree complex wavelet transform. This research was in part supported by a grant from ITRC.

5. REFERENCES

[1]. M.Swain and D. Ballard, “Color Indexing,” Int. J. Comput. Vis., vol. 7, pp. 11-32, 1991.
[2]. A. W. M. Smeulders *et al.*, “Content-Based Image Retrieval at the end of the Early Years,” IEEE Trans. on Pattern Anal. Machine Intell., vol. 22, pp.1349-1380, Dec. 2000.
[3]. M. Stricker and A. Dimai, “Spectral Covariance and Fuzzy Regions for Image Indexing,” Mach. Vis. Applicat., vol. 10, pp. 66-73, 1977.
[4]. M. K. Mandel and T. Aboulnasr, “Fast Wavelet Histogram Techniques for Image Indexing”, IEEE Trans. On Consumer Electron., vol. 42. Aug. 1996.
[5]. G. Pass and R. Zabih, “Histogram Refinement for Content–Based Image Retrieval,” in Proc. IEEE Workshop on Applications of Computer Vision,1996, pp. 96-102.
[6]. J. Huang et al., “Spatial Color Indexing and Applications,” Int. J. Comput. Vis., pp.245-268, 1999.
[7]. J. Han and K. Ma, “Fuzzy Color Histogram and Its Use in Color Image Retrieval”,. IEEE Trans. on Image Processing, vol. 11, No. 8. Aug. 2002
[8]. G. Qiu and K. Lam. “Frequency Layered Color Indexing for Content-Based Image Retrieval,” IEEE Trans. on Image Processing, vol. 12, No. 1, Jan. 2003.
[9]. P .J. Burt and E. H. Adelson, “The Laplacian Pyramid as a Compact Image Code,” IEEE Trans. on Commun., vol. COM-31, pp. 532-540, 1983.
[10]. A. Mojsilovic et al. , “Matching and Retrieval based on the Vocabulary and Grammar of Color Patterns,” IEEE Trans. on Image Processing, vol. 9, pp. 38-54, Jan. 2000.
[11]. N. Kigsbury, “Image Processing with Complex Wavelets”, Phil. Trans. R. Soc. Royal Society, 1999.

Figure 5. Performance of the proposed retrieval algorithm. The image in the top left is the query image and images from left to right and top to bottom are the retrieved images.(Due to the lack of related images in the database, it has retrieved the last three images that have been the most related images contained in the database.)

Figure 6. Performance of the proposed retrieval algorithm. The image in the top left is the query image and images from left to right and top to bottom are the retrieved images.