VECTOR-WAVELET BASED SCALABLE INDEXING AND RETRIEVAL SYSTEM FOR LARGE COLOR IMAGE ARCHIVES

Elif Albuz, Erturk Kocalar, and Ashfaq A. Khokhar

University of Delaware
Department of Electrical and Computer Engineering
Newark, DE 19716
{albuz, kocalar, ashfaq}@eecis.udel.edu

ABSTRACT

This paper presents an efficient content based indexing and retrieval mechanism based on vector wavelet coefficients of color images. We use highly decorrelated wavelet coefficient planes to acquire a search efficient feature space. The feature space is subsequently indexed using properties of the all the images in the database. Therefore the feature key of an image does not only correspond to the content of the image itself but also how much the image is different from the other images being stored in the database. The search time depends only on the number of images similar to the query image but not on the size of the entire database. The system is scalable and provides fast retrievals. We show that in a database of 1000 images, query search takes less than 50 *msec*, on a 266 MHz Pentium processor compared to several seconds of retrieval time in the earlier systems proposed in the literature.

1. INTRODUCTION

With the development of world-wide-web (www) and fast computer technology, the use of visual information has become a routine in scientific and commercial applications. Images and videos are being captured and archived in large databases and it is extremely important to develop efficient technologies to process, manipulate, and search large visual information archives. In this context, retrieving images by content has gained increasing interest. The existing image indexing systems deliver poor performance for large image archives. The most difficult part of the problem is to find feature vectors that represent image archives as close as possible and data structures that organize the feature vector space efficiently and thus speed up the search process. In addition, a feature vector has to be computationally inexpensive to facilitate query processing in real time. This paper presents an effective approach based on multiresolution properties of vector wavelet [13] features and a variant of a B-tree structure for efficient indexing and online retrieval of images in the database.

Currently, two main approaches exist that avoid human interaction: indexing of images based on features from raw image data [2, 1, 8, 9, 3, 4] such as pixel intensity, histogram, etc. and indexing based on coefficients in the compressed transform domain [11, 6, 7, 5, 12]. The general idea of content-based image databases is that for each image a feature vector closely representing the image is stored in the database. At the time of the query, feature vector of the query image is computed and the database is searched for the matching feature vectors. All the images corresponding to the matched feature vectors are presented as a result. Most of the proposed techniques in the image indexing literature have focussed on finding robust features that result in high similarity matches. The organization of the feature space in the database has been considered a separate problem. This approach has severely hampered the scalability of the proposed techniques. The performance of the retrieval systems degrades significantly as the size of the archives grows. We believe it is the right COMBINATION of the feature space and storage structures that yields scalable systems. Some features spaces are better organizable than the others. In most of the existing systems, the feature vector of the query image has to be compared sequentially against all the entries in the database. This results in large query response times that linearly depend on the database size.

In this paper, we treat the feature vector and its efficient indexing as a unified problem and suggest a solution in which query response time is relatively independent of the database size. We exploit energy compaction properties of the vector wavelet [13] and design suitable data structures for fast indexing and retrieval mechanisms. We believe that an indexing system is efficient if in some implicit terms the system provides information on the relationship of each image with the other images stored in the database. For example, how much different is one image compared to the others? How many different classes of images are present in the database? What are the properties of these classes in terms of the feature vector? The use of this information during query processing time will reduce the search space and expedite the query response. Therefore, instead of storing feature vector for each image separately, images that have similar features are clustered together and properties of each cluster are used to represent the corresponding images. We show that on Pentium II 266Mhz Windows NT system, using a database of 1000 images, query search takes less than 50 *msec*, compared to several seconds of retrieval time in the earlier systems. Given a query image, the ratio of the similar images over the number of images retrieved for a maximum quality search is 0.9, yielding efficiency close to 1.

The rest of the paper is organized as follows. Section 2 provides our approach for search efficient feature spaces for indexing color images. Section 3 describes indexing and query system. Experimental results and conclusions are presented in Section 4.

2. INDEXING COLOR IMAGES

In the color image domain, dealing with multiple spaces, several researchers have utilized different color transformations such as RGB, YUV, YIQ, etc. The main idea has been to consider it as a three dimensional problem and extend the results for single dimension to multiple dimension, considering indexing in each color space independent of the others. For example, construct multiple indexing systems, one for each color space, and then combine the results based on a user-specified preference. On the other hand multidimensional indexing data structures have been employed to handle all the color spaces simultaneously. The performance of these approaches varies depending on the application. However, a human eye doesn't see an image and all of its colors separately. In fact it looks at the composite signal for an approximation and then looks into the detailed signals for a precise match [10]. We propose to take a similar approach for computer based indexing and retrieval of color images. Since information in multiple spaces is highly correlated, we propose to decorrelate the information across the three planes such that a single composite plane is computed that has the maximum energy compaction of all the planes. We can use the plane with highest energy as the primary plane for first level indexing of color images. Once the search space has been limited to a smaller subset of images, a more detailed comparison can be applied using other all the color planes as well. In this way the main data structure that contains a much larger set of images may still be simple and efficient thus providing a scalable search efficient engine. For detailed searches in multiple planes on a smaller subset of images, multidimensional spatial data structures are explored.

Vector-valued wavelets have been recently studied in [13] for decorrelating vector-valued signals not only in the time/spatial domain but also between the components in the vector. There have been other linear transformations for decorrelating the three components of a color image, such as YIQ. However, vector-valued wavelets have two advantages over the existing methods: A) vector-valued wavelets have better energy compaction between the components than other linear transformations; B) vector-valued wavelets treat the correlations between the components and also between the pixels together and therefore jointly and optimally decorrelate all the correlations.

Notationly vector-valued wavelets are similar to single wavelets. There are two filters: the "lowpass" filter $\mathbf{H}(w)$ with $\mathbf{H}(0) = I_N$ and the "highpass" filter $\mathbf{G}(w)$ with $\mathbf{G}(0) = 0_N$, where I_N is the NxN identity matrix and 0_N is the NxN all zero matrix. The difference is that these two filters are NxN matrix polynomials, i.e. each component of the matrix is a polynomial, where N is the size of the vector. Since these filters are no longer time-invariant, the "lowpass" and "highpass" are interpreted in a different way, for further details see [13].

The orthogonality of the vector-valued wavelets induced from $\mathbf{H}(w)$ and $\mathbf{G}(w)$ is equivalent to the paraunitariness of their polyphase matrix $\varepsilon(w)$. The conditions $\mathbf{H}(0) = I_N$ and $\mathbf{G}(0) = 0_N$ are equivalent to

$$\varepsilon(0) = \frac{1}{2}\begin{pmatrix} I_N & I_N \\ I_N & -I_N \end{pmatrix}$$

Another way to construct vector-valued wavelets is as follows. Set $\mathbf{H}(w) = \mathbf{H}_O(w)\mathbf{C}$ and $\mathbf{G}(w) = \mathbf{G}_O(w)\mathbf{C}$ where $\mathbf{H}_O(w)$ and $\mathbf{G}_O(w)$ are constructed from single wavelets, such as Daubechies wavelets, and $\mathbf{C}$ is the N by N discrete cosine transform (DCT). The vector-valued wavelets constructed above are called DCT assisted vector-valued wavelets.

For an example color image, the energy compaction characteristics of different transformations in the wavelet domain are shown in Figure 1. Clearly RGB color space decorrelated using DCT-assisted vector valued wavelets has the highest standard deviation among its planes in all the subbands. We choose the plane with the highest energies as a primary plane for indexing images. Figures 2 shows classification of a sample set of images using wavelet energy values in all the color planes and Figure 3 shows the same using the highest energy plane in the DCT-assisted vector-wavelet. The classifications are identical and

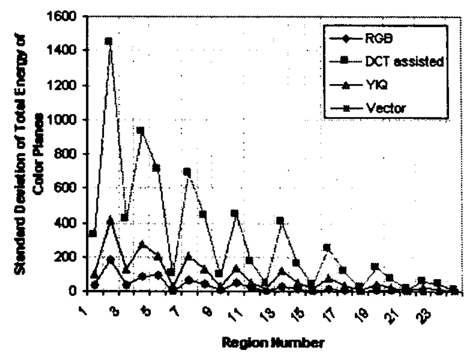

Figure 1: Comparisons of different decorrelation schemes

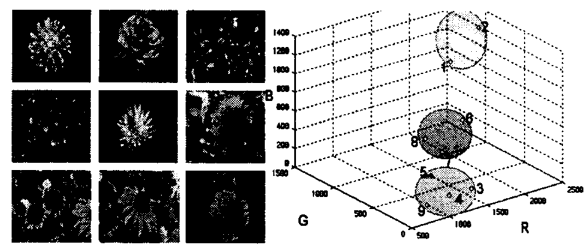

Figure 2: Classification of sample set of images using multiple color planes

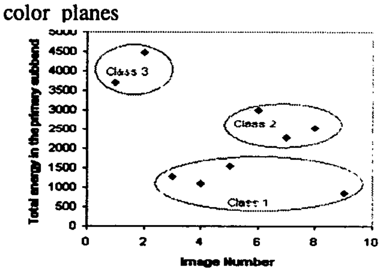

Figure 3: Classification of sample set of images using highest energy plane based on DCT assisted vector wavelets

encourage the use of this methodology for primary indexing of large color image archives. In the following we provide a simple mechanism to extract classes of features from a single plane.

3. INDEXING AND RETRIEVAL SYSTEM

To the best of our knowledge, few of the existing solutions, such as [11, 5], have addressed the problem of feature extraction and issues associated with the database construction. In most of the existing systems, the query time increases almost linearly with the database size. To improve the response time, either simple features were used or quality of the intended result was lowered. On the other hand, complex feature vectors resulted in good similarity measures, but complex feature vectors take more time to compute and may not be organizable efficiently in a database. The relationship between the feature complexity, the access time, and the retrieval quality is important. Up to now, these have been considered as three separate problems. However, when individual solutions were brought together, often the resulting systems do not scale very well. Considering the problem as a whole, we utilize an efficient feature space and associated indexing mechanisms that result in low access time and high retrieval quality.

3.1 Feature Computation

Feature computation is based on the energy properties of wavelet subbands. For each image, after the wavelet transformation, the coefficients of the highest energy plane is divided into natural wavelet subbands. For each subband, the local features are computed. Currently we are using the total energy of each subband as a feature to represent the contents of the region. However instead of summing up all the coefficients, we use the sum of squares of the coefficients in each region. Using a space-filling curve such as Morton order, we store these regional features in an array. This array represents the contents of the image, however we also need the similarity of this image and the rest of the images in the database.

The second step of feature computation is to identify the similarity of the image. This is accomplished by calculating the *key* of the image. Key represents how much a region is similar to the corresponding region of other images in the database. Once the regional features of all the images have been computed, the distribution of the energy spectrum for each region is examined considering all the images in the database and an energy histogram is built for each region as shown in Figure 4. The histogram of each region is further divided into logical groups or classes or buckets. The buckets are enumerated in sequential order. Each region in the image will belong to a bucket in each regional histogram. Images that have similar regional energy values will be close to each other in the corresponding regional histogram and thus are placed in the same bucket. The buckets can be fixed size or variable size. The width of the bucket defines the distance between two similar images when comparing their corresponding regions. For each image, its relationship with the other images in the database is represented by its *key*. Key is constructed as follows: Using a specific traversal order of the regions, i.e. Morton order, the bucket numbers of each region are concatenated together. In other words, the key consists of the bucket numbers of each region. An example computation is given in Figure 4.

3.2 Database Construction and Query Processing

The way image key is constructed plays an important role in the database construction. The image key consists of the bucket numbers of the regions concatenated based on the Morton order traversal of the regions. An image will not be similar to the images that have a different bucket numbers and there is no need to check them during the query. If two images are different in the primary region it is very likely that will be different in the subsequent regions as well. A variant of a B-tree data structure where the height of the tree corresponds to the total number of regions is used for database structure. The root level corresponds to the first numer in the key and the second level corresponds to the second number in the key. The number of edges going out from root to the next level is equal to the number of buckets in the first region. The edges going out from a node in the second level represent the classes in the second region. The leaves of the tree contain the images that have exactly the same image key. Therefore database allows similar images to be stored close to each other before querying. Insertion of an image into the database is done by scanning the image key from left to right and traversing the tree down taking the corresponding edges. The height

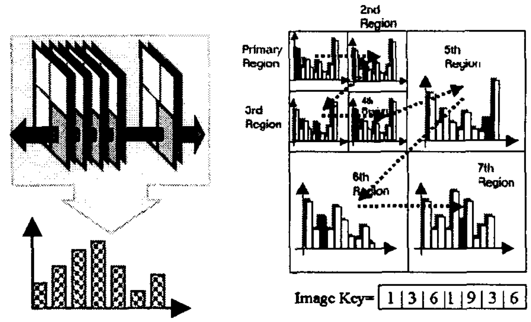

Figure 4: Class identification across images and subsequent calculation of the feature key using all the regions.

of the tree depends on the size of the image key, which is proportional to the number of subbands being used to build and search the database.

Given a query image, the image key of the query image is computed using the previously calculated regional histograms. Then starting from the first bucket number in the query image key, the tree is traversed down. At the end of the traversal, a leaf is reached. If there is any, all the images in the leaf are presented to the user in a thumbnail fashion. Depending on the traversal order used in indexing, portions of the key can be used to vary the quality of query result. If only a partial match of the image keys is required, then the tree traversal would stop before reaching a leaf and the images represented by the subtree originating from that node down will be presented to the user. In order to further refine the query results, all the images present in a bucket can be examined using more detailed features.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

The indexing and query mechanisms proposed in Section 3 have been implemented using the Rapid Application Development Tool on a Windows NT based system. The implementation consists of two main parts: the indexing of all the images in a database, and the development of query system and the associated user interface. The user query system allows users to sketch pictures, provide an example image file, or make changes in the query image. Currently our database consists of one thousand 512x512 pixels natural images. The results presented in this paper are based on indexing of features from a single plane.

We compare the performance of our method with the other systems proposed in the literature. We convincingly show that our approach is more robust, fast, and scalable compared with the earlier approaches. The feature vector size is proportional to the logarithm of the number of classes, and is bounded by $R\log_2 C$ where R is the number of regions used to index images and C is the number of classes in each region. Thus, if we use 8 bits to represent a class number and use maximum number of regions, i.e. $\log_2 512 * 3 + 1 = 28$, the feature vector size is at most 28 bytes. We are using only 10 regions to index images of size 512x512. Table 1, column 4, compares the space requirements for storing feature vectors.

System	Image Size	Database Size	Feature Vector Size	Search Time
LiangKuo[7]	192x128	2119	212bytes	NA
WBIIS [11]	128x128	10,000	≥768 bytes	3.3sec
QBIC[2]	100x100	1000	NA	2-40sec
U. Wash.[5]	128x128	1093	O(m) bytes	47.46sec
Udel	**512x512**	**1000**	**10bytes**	**50msec**

Table 1: Comparison of different image indexing systems.

In our system the query search time depends on the number of different image classes and it is relatively independent of the database size. Compared with the other systems, our implementations run several orders of magnitude faster than the fastest time reported in the literature. The time to calculate the feature vector is also important since it directly adds to the query response time. We are unable to determine this data for the existing systems from the literature. However, based on the complexity of the feature employed in most of these systems, we believe our feature computing time compares fairly well. In a database of 1000 images, it takes about 170 msec to compute the image key for the input query image. Note that this does not include the time to compute the wavelet coefficients.

The performance of the indexing scheme can be assessed from the ratio of relevant images to retrieved images. In our scheme, the images are clustered before they are queried. So the ratio of relevant images to the total retrieved images is quite high and the exact matches are immediately returned. The database could be searched for different levels of quality. A higher value for the search quality parameter implies a closer match. For example, a search quality value of 4 implies that comparison should be made based on first 4 regions only. And a search quality of 10 implies that exact matches should be made, because all of the regions will be matched. Note that we are using only 10 regions in the current implementation. Figure 5 presents results for different input images.

5. REFERENCES

[1] C. C. Chang and S. Y. Lee. Retrieval of similar pictures on pictorial databases. *Pattern Recognition.* 24(7):675--680, 1991.

[2] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems,* 3(3/4):231--262, July 1994. RJ 9453.

[3] K. Hirata and T. Kato. Query by visual example. In *Advances in Database Technology EDBT '92, Third International Conference on Extending Database Technology,* Vienna, Austria, march 1992. Springer-Verlag.

[4] F. Idris and S. Panchanathan. Image indexing using vector quantization. In *Proceedings in Storage and Retrieval for Image and Video Databases-III,* volume 2420, pages 373--380, San Jose, CA, feb 1995.

[5] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95,* 1995.

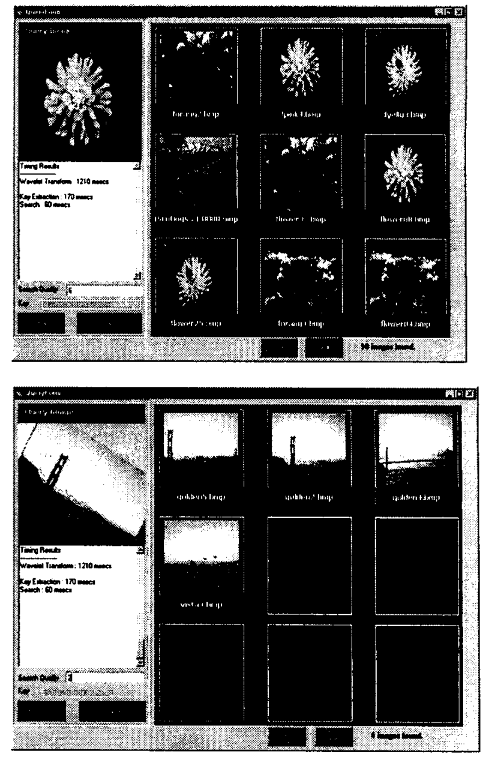

Figure 5: Query results with different search qualities.

[6] K.-C. Liang and C.-C. J. Kuo. Wavelet-compressed image retrieval using successive approximation quantization (saq) features. In *SPIE Voice, Video, and Data Communications,* Dallas, Texas, november 1997.

[7] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Proceedings: Storage and Retrieval for Image and Video Databases II,* pages 34--47, San Jose, CA, february 1994. SPIE.

[8] John R. Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. *ACM Multimedia,* 1996.

[9] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision,* 7(1):11--32, 1991.

[10] B.A. Wandell. *Foundations of Vision.* Sinauer Associates, Inc., Sunderland, MA, 1995.

[11] WangWie. J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. *Journal of Digital Libraries,* 1997.

[12] W.Y.Ma and B.S.Manjunath. Pictorial queries: Combining feature extraction with database search. Technical Report 18, University of California at Santa Barbara, Dept. of Electrical Engineering, 1994.

[13] X.-G. Xia, J. S. Geronimo, D. P. Hardin, and B. W. Suter. Design of prefilters for discrete multiwavelet transforms.*IEEE Trans. on Signal Processing,* 44:25--35, January 1996.