ADAPTIVE IMAGE DATABASE USING WAVELETS

Shantanu Chakrabartty, Milutin Stanacevic and Trac D. Tran

Department of Electrical and Computer Engineering
Johns Hopkins University, Baltimore MD 21218
E-mail: shantanu@jhu.edu, miki@jhu.edu, trac@jhu.edu

ABSTRACT

We propose a novel approach to image database design, using wavelets and their inherent hierarchical nature. Simple feature vectors are computed from the images and used to tune the database storage structure, yielding the maximum performance. This approach is more efficient than conventional approaches in terms of retrieval rates and storage space. The database needs less number of parameters to be stored and less computation has to be done to retrieve images which are similar to a query image. Moreover, the simplicity of the work leads to a design of a more scalable system.

1. INTRODUCTION

Due to the advent of the world wide web and prevalence of digital media in today life, there is an increasing demand of organizing data in such a manner that contained information can be efficiently archived and retrieved. Archiving and retrieval of images, as digital media, definitely introduce a problem as the content of information in each image has a large variance. Therefore, the only measure by which such information can be achieved is through similarity measures. The question that can be posed is how similar two images are or a problem of finding a list of images that are similar to a query image.

But even with this formulation, we cannot expect that the problem has been solved, because similarity measure itself is a widely debatable subject. A side view of a persons face is similar to the front view of the same persons face. However, such degree of similarity detection would require sophisticated processing and enhancement of similarity metrics.

In literature numerous schemes have been proposed to address the question of similarity metric. One of them is mutual information between two images [2]. However, if we carefully examine the classical paradigm of approaching the archiving and retrieval problem, it seems that similarity metrics and storage data structures for the images have been dealt separately, by indexing images based on features extracted from raw image data [3, 4, 5], or indexing based on coefficients in the compressed transform domain [6, 7, 8, 9]. Recently, new approaches [10, 11] have been proposed that include the right combination of the feature space and storage structures.

In this paper we propose scheme that facilitates efficient retrieval of images. The rule of thumb followed in designing databases is that archiving or storage of image data can take much more time, but the search for a similar image, given a query image, should be very fast. Therefore, indexing mechanism has to be applied so image database can be suitably built. Wavelet transform

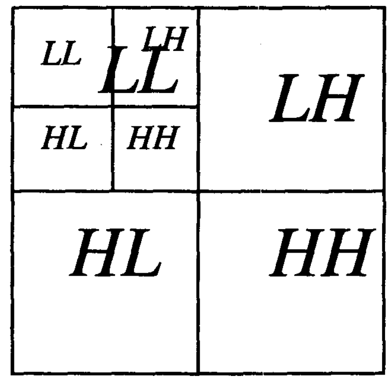

Figure 1: Wavelet decomposition of an image.

enables extraction of information from the image data at different time and frequency resolutions, which can be used for classification of images and hence can be a very useful mechanism for design of hierarchical databases with efficient retrieval. Moreover, we propose the storage architecture that kind of binds the similarity measure together, leading to more efficient storage.

Section 2 gives the overview of the approach we have used for solving the problem. Section 3 explains the results and finally section 4 draws conclusions on those results and elucidates the possible future extension of this work.

2. ADAPTIVE DATABASE DESIGN

Wavelet transform is able to give a multi-resolution representation of the signal, since each frequency component can be analyzed with a different resolution and scale [1]. Our signal is a $2-D$ image, where the 'time' domain is the spatial location of pixels and the 'frequency' domain is the intensity variation between adjacent pixels. For an N x M image, the first transformation step decomposes the signal into four sub-images of size $N/2$ x $M/2$, representing the sub-bands in the frequency domain. The obtained sub-images are labelled as LL, LH, HL and HH, where L and H represent low and high frequency information, respectively, and the first and the second position refer to the horizontal and to the vertical direction, respectively. The second transformation level decomposes the LL sub-image, obtaining four images of size $N/4$ x $M/4$, and so on. Figure 1 shows the decomposition of the frequency domain at different scale levels.

0-7803-6514-3/00/$10.00©2000 IEEE

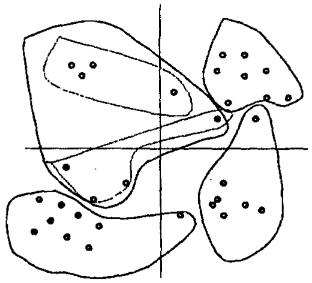

Figure 2: Database design as a special case of vector quantization

2.1. Assumptions

The problem of image classification and retrieval has a very broad spectrum. We try to approach a subset of this class of problems. As a result, our work was subjected to the following assumptions. However, the algorithm which we propose in the later sections can be applied to a more generalized framework.

- All the images in our database are in gray scale. The color image can be divided into the corresponding color channels and the discrete wavelet transform (DWT) would be applied separately to each channel. Feature vectors would have increased dimensionality, so the color information would improve retrieval performance in terms of image similarity.
- All the images in our database have been set to the same size. This step helps to simplify the work in terms of calculating the same number of wavelet coefficients.
- Image database size has been chosen to be small. This limitation is more due to availability of wide spectrum of images. But, our conviction is if we can gain insight into the principles based on few images, then those can be easily extended to large set of images. In principle, large sets of images would help our purpose because it would lead to a better classification as those samples would be representative of the real world.
- As mentioned above, we assume that these small set of images that we use represent the broad spectrum of images that can be obtained in real life.

2.2. Conceptual Reasoning

Adaptive database design is a form of hierarchical vector quantization. Consider a set of two dimensional feature vectors that have been extracted from the training set. Figure 2 shows the vector constellation of the training set. Usual vector quantization would try to find out the cluster centers by finding out the points in this space which are closest to the data points. For a database design, there is an additional constraint, the number of points that belong to the cluster centers should be equal. This constraint results in balancing the database tree we want to construct.

Figure 2 shows the formation of first four regions. The dotted line shows that the points in a region are clustered into sub-regions in the similar fashion. This hierarchical clustering continues till each cluster has less than or equal to 4 vectors.

As we can see in the Figure 2 the above scheme could group two dissimilar vectors together, resulting the database miss[1]. These cases are taken care of if we choose the threshold values in the parameter space appropriately, as it will be shown later. Moreover, as mentioned, we need a rejection criterion for avoiding false matches which could employ measuring the mutual information content of the query image and the matched image.

2.3. Algorithm Design

There are number of ways for solving the adaptive database problem. The one we describe is a sub-optimal solution and search for better solutions lies in future work. We use a projection mechanism to demonstrate the effectiveness of the above scheme. Suppose we have a set of images ρ that constitute our training set which will form our database. Let each element of the set be ρ_m where $m\epsilon(1, N)$, N being the training set size.

2.4. Similarity Metric

The function of the similarity metric is to extract the maximum information from the training set. We want to find out a function that maximizes the variance and also extract maximum information from it. We achieve variance maximization by choosing training set that represents the broad spectrum of image queries that is possible. For extraction of maximum of information from the image we propose the following steps:

- Applying wavelet decomposition to extract four region as previously mentioned.
- Calculating the entropy as similarity metrics of each of these regions to obtain a 4 dimensional vector.
- Normalize each of the dimensions over the whole training set so that the maximum value of that dimension value is equal to unity, which allows unbiased representation of each dimension.

The similarity metrics used is entropy, calculated using Shannon's entropy definition [12]. The maximum entropy (maximum disorder) of a 256 x 256 size image with one byte per pixel is 5.545. Zero is minimum entropy (maximum order). As similarity metrics, in the conducted research, the total energy of the sub band, as the sum of squares of the coefficients, was also used. Due to high bias of energy in the LL region, the algorithm which we propose below did not give expected results. Moreover, the high bias also affected the precision involved in calculating the feature vectors and the corresponding parameters due to denoising errors. For wavelet decomposition Daubechies wavelet of fourth order is used.

At the end of this computation, we obtain N x 4 dimensional vector T, whose elements are the feature vectors, obtained from training images using the above procedure.

2.5. Parameter space

A 4 dimensional vector from each image is obtained, representing a type or a class of images. However, we want to minimize retrieval time, too. To minimize retrieval time, the training set is split more evenly into four classes. In principle, an optimum choice of

[1]A database miss is defined as the probability that a query for an image results in an image that is not the most similar image to query image in the database

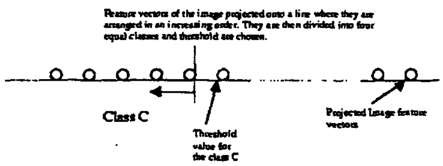

Figure 3: Projected image vectors onto a line, from which the thresholds shown are determined.

parameter size is 4, which is equal to the number of classes we are seeking for. One way of estimating these weights would be by hand labeling the training data and using regression to find the values of the parameters. However, hand labeling of a huge database is a very tedious task, so we will resort to unsupervised learning or vector quantization to obtain clustering of the training data about which the variance is minimum. Hence, our algorithm for parameter estimation is:

- Obtain 4 cluster centers using K-means clustering method of vector quantization from the 4 dimensional training data vectors, which is a 4x4 matrix *A*.
- Label those cluster centers into four categories namely C_1, C_2, C_3, C_4, which form the 4x1 label vector *C*.
- Solve the equation *Aw*=*C* where w_1, w_2, w_3, w_4 are the four elements of the weight vector *w*. We used regression to solve this to avoid possible matrix inversion problems but the faster methods can be used. At the end of this step we obtain $w = A^{-1}C$.

We have a set of parameters *w*, obtained from the training data. The next step would be to classify the training data equally into four classes. The proposed algorithm is as follows

- Calculate $P = w^T T$, which will give the projection vector *P*, that represents the projection of the training data vectors into a projection space.
- Sort the column matrix P to obtain a sorted order P_s.
- Divide the sorted column matrix into four equal parts and choose the boundary values of the division as the threshold values t_i, where $i\epsilon(1, .., 4)$. Therefore, the training set is divided into four categories $\rho_{1,1}, \rho_{1,2}, \rho 1, 3, \rho 1, 4$ based on the parameters *(w ,t)* which form the parameter space for the first level namely $\theta_{1,1}$, where $\rho_{i,j}$ means image subcategory number j at level i, and $\theta_{i,j}$ means parameter space j at level i. Figure 3 shows the sequence of the projected image vectors onto a line, from which the threshold is determined.

For each of the 4 sub-categories of images, we apply the following procedure:

- Calculate the level one decomposition of the wavelet coefficients of the LL portion of the image. This means that we apply wavelet decomposition on the previously obtained LL portion of the image.

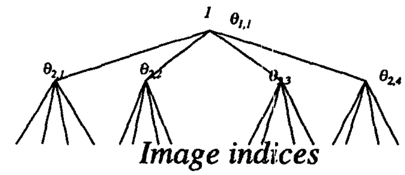

Figure 4: Tree structure of the database. Each level has a set of parameters given by $\theta_{i,j}$, consisting of weights $w_{i,j}$ and thresholds $t_{i,j}$. The leaf contains the index of the image in the training set.

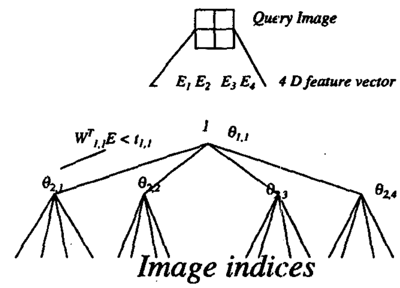

Figure 5: Search through the database based on the query image. The vector E is extracted and $w^T E$ is computed to decide which class the image belongs to. In that sub-class, the new feature vector E, based on level 2 wavelet decomposition, is formed and the similar procedure is repeated until the leaf is reached.

- For this LL decomposition, we proceed with the re-estimation procedure of parameters and for the next level namely $\theta_{2,1}, \theta_{2,2}, \theta_{2,3}, \theta_{2,4}$. In this fashion, we continue on until each leaf has a prescribed minimum number of images.

We have built a tree based on the training set, with each node constituting a parameter $\theta_{i,j}$, a set of weights w_k and thresholds t_k. The structure of the tree is shown in Figure 4.

2.6. Query Procedure

The procedure of image retrieval from the database given a query image is described.

- Given a query image wavelet decomposition is applied to obtain four regions LL, LH, HH, HH.
- The entropy of each region is calculated to obtain the 4 dimension vector, which is the representative of this image.
- We calculate $p = w_{i,j}^T E$ where E is the 4 dimensional feature vector of the query image. Now starting from $j = 1$ to 4 if $p < t_{i,j}$ then the query image belongs to class (*i*+1, *j*). At the next level of iteration the parameters that would be used are $\theta_{i+1,j}$.
- The above procedure is continued until we reach a leaf which will give us a list of image keys through which we can do a linear search to find out the most similar image.

The above procedure is shown in schematic in Figure 5.

3. RESULTS

We tested the algorithm using a set of 37 images out of which 32 of them were used as training set and 5 were used as a test set. The following table shows some tentative values of the entropies that are obtained from two images which look pretty different to illustrate the point of using entropy as feature vector. The table below shows some of the entropy values for some of the images that are used as training set.

Image number	E1	E2	E3	E4
21	1.0	0.7695	0.76629	0.7195
20	0.8897	0.9322	0.9463	0.9589
24	0.9299	0.9161	0.77384	0.7819
15	0.953	1.0	0.8919	0.93189

As we can see, the images 21, 20, 24 and 15 are very distinct from each other and therefore have different entropy values. The following figures show the training set we used, all arranged in a random fashion as shown in Figure 6 and then we show the result of application of our algorithm shown in Figure 7. The results show that the images which are similar to each other are clustered near each other, which proves our hypothesis of using entropy as a similarity metric and the tree algorithm as a clustering mechanism.

3.1. Computational Efficiency

Since the only way of testing the fast retrieval time and comparing it against alternative approaches would be to code efficient algorithms for each approach and implementing it on the same platform using the same training set, there is still work left in implementation aspect of proposed algorithm. However, theoretically and conceptually we can prove that faster retrieval rates are achieved compared to the alternative methods. The reasons why faster retrieval rates are achieved are as follows:

- The tree which is our primary data structure is a quaternary tree, which leads to a $O(log_4)$ complexity.
- The algorithm makes the tree balanced due to division of the images equally amongst each class after regression.
- The parameter space $\theta_{i,j}$, which is actually our storage space, requires very less as compared to alternative approaches. In principle, the storage space required is $0(N)$, where N is the size of the training set, which shows that there are no additional parameters that need to be stored for the training set of images other than the tree parameters.
- The order of complexity of query time can be computed as follows. Let the computational time for computing a wavelet decomposition and entropy calculation for M x M image be α units. The next level decomposition would then take $\alpha/2$ and the next level would take $\alpha/4$ and so on. At each level, we compute $w^T E$, as described earlier, which is just a 1x4 matrix into 4x1 matrix multiplication and hence that computation time can be neglected as compared to the wavelet decomposition time. The total time taken until we find the class of images to which the query image is similar, is given by

$$T = \sum_{H} \alpha/2^k \tag{1}$$

where H is the height of the tree which is approximated equal to $log_4 N$. Therefore, we obtain the computational time approximately equal to $2\alpha(1 - 1/N)$, which means that the asymptotic complexity is a constant equal to 2α and independent of the size of the database.

For the linear search, the asymptotic complexity for the same approach is αN, which is not acceptable for very large databases.

4. CONCLUSIONS

In this paper, we have proposed a novel approach of combining the feature vectors and the data structures leading to an efficient implementation in terms of retrieval time and storage space.

4.1. Future Work

The demonstrated work represents a minimal subset of the whole problem. A more exhaustive approach would take into consideration following points.

- A huge training set that comprises of more varieties of images.
- A more sophisticated similarity metric than entropy to generate feature vectors. A more intuitive approach would be partitioning of the image into several regions and then generating feature vectors from them.
- A better clustering mechanism than K-means clustering would definitely improve the performance, since K-means clustering is a local maximization which highly depends on the choice of initial points.
- Extension of the above approach to color images and images of irregular size.
- One aspect, we have ignored in our work, was the use of transforms to capture the effect of rotational similarity of the images. This would in turn mean a different similarity metric or more number of feature vectors. Whatsoever, the algorithmic implementation of this approach would not change.
- One important aspect which we would like to include is the rejection criterion, the development of a condition which would return a NULL value if the query image does not have a similar image in database. This would definitely help in reducing the number of false matches and it could be done when the query procedure has identified the leaf where the index of the similar image resides. If the rejection criterion is not met between the similar image and the query image then a NULL is returned.
- Characterization has to be done with regards to the worst case performance in terms of database misses.

5. REFERENCES

[1] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley, MA: Wellesley-Cambridge Press, 1998.

[2] B. Liu and S.F. Ling, "Signal classification by mutual information based discriminating pursuit," *TENCON*, vol. 2, pp 530-535, 1996.

Figure 6: Training set of images used for database creation

Figure 7: Location of the images of the training set at the leaves of the database

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

[4] 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,* vol. 3 (3/4), pp 231-262, Jul 1994.

[5] F. Idris and S. Panchanathan, "Image indexing using vector quantization," *Proceedings in Storage and Retrieval for Image and Video Databases-III,* vol. **2420**, pp 373-380, Feb 1995.

[6] C.E. Jacobs, A. Finkelstein and D.H. Salesin, "Fast multiresolution image querying," *Proceedings of SIGGRAPH,* pp 278-280, 1995.

[7] S. Ardizzoni, I. Bartolini and M. Patella, "Windsurf: Region-Based Image Retrieval Using Wavelets," *Database and Expert Systems Applications,* pp 167-173, 1999.

[8] V. Castelli, C.S. Lim, J. Turek and I. Kontoyiannis, "Progressive classification in the compressed domain for large EOS satellite databases," *ICASSP,* vol. **4**, pp 2199-2202, 1996.

[9] K.C. Liang and C.C.J. Kuo, "Retrieval and progressive transmission of wavelet compressed images," *ISCAS,* vol. **2**, pp 1464-1467, 1997.

[10] E. Albuz, E. Kocalar and A.A. Khokhar, "Vector-wavelet based scalable indexing and retrieval system for large color image archives," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. **6**, pp 3021-3024, 1999.

[11] M.G. Albanesi, M. Ferretti and A. Giancane, "A compact wavelet index for retrieval in image database," *International Conference on Image Analysis and Processing,* pp 927-932, 1999.

[12] R. Chiou, M.A.G.V. Ferreira, C.T. Silva and A.E. Kaufman, "Wavelet and entropy analysis combination to evaluate diffusion and correlation behaviors," *Computer Graphics and Image Processing,* pp 119-125, 1997.