Performance Analysis of Using Wavelet Transform in Content Based Video Retrieval System

YUK YING CHUNG, WAI KWOK JESS CHIN, XIAOMING CHEN,
*DAVID YU SHI, *ERIC CHOI, *FANG CHEN
School of Information Technologies, University of Sydney, NSW 2006, AUSTRALIA
* ATP Research Laboratory, National ICT for Australia, NSW 1430, AUSTRALIA

Abstract: - Multimedia Retrieval is one of the most important and fastest growing research areas in the field of multimedia technology. Large collections of scientific, artistic and commercial data comprising image, text, audio and video abound in the present information based society. There must be an effective and precise method of assisting users to search, browse and interact with these collections. This paper has proposed and implemented a content based video retrieval system using Haar wavelet, Daubechies D4 wavelet transform and five different types of clustering techniques. The experimental results show that the Haar wavelet with 3-Level transform can perform best with a high retrieval accuracy rate (89%).

Key-Words: - video retrieval, wavelet transform, clustering algorithms

1 Introduction

The large and growing amount of digital data and the development of the Internet highlight the need to develop sophisticated access methods that provide more than just simple text-based queries. Many programs have been developed with complex mathematical algorithms to allow the transformation of image or audio data in a way that enhances searching accuracy. However, it becomes difficult when dealing with large sets of multimedia data. This paper proposes and demonstrates a Content Based Video Retrieval System (CBVRS) based on wavelet transform and clustering algorithms.

Content-Based Video Retrieval Systems (CBVRS) allow similar features to effectively characterise the content of images and then uses such features in the retrieval process. Recent studies have proved that features such as colour, texture, shape and spatial position indeed possess a very high sematic value for image or video retrieval systems [1,2,3]. Such systems represent each image as a set of feature vectors which allows the assessment of similarity between images. IBM QBIC system [1] is an example of using colour, texture and shape as feature vectors.

Huge amounts of multimedia data now can be stored because of the advancement of technology and the low cost of hard disk. The major difficulty is to efficiently locate images and video frames within huge databases. A searching process would be time-consuming if it needed to search on the whole database which may contain one million pictures and videos. This paper describes the implementation of wavelet-based video retrieval system using clustering algorithms. The proposed CBVRS system can have high retrieval accuracy (89%) and 7 times faster than the non-clustering CBVRS.

Section 2 describes the proposed Content Based Video Retrieval System (CBVRS) using Haar wavelet and Daubechies D4 wavelet transform. Section 3 explains the five clustering algorithms: Forgy, K-means, Isodata, Two-level clustering and One-level clustering. Sections 4 and 5 present the experimental results and conclusion, respectively.

2 Content Based Video Retrieval Systems (CBVRS)

Nowadays people not only use pure images for daily purposes, but video is also a popular media for recording TV, diaries etc. As a consequence, effective methods for searching large databases are needed. This suggests the need for using Clustering Based Image Retrieval System for images or videos which allow users to search images or particular image frames according to their preferences.

The proposed Clustering Based Video Retrieval System (CBVRS) comprises three components to obtain desired images or frames efficiently. They are colour histogram, clustering process and searching process. The colour features of each image or video frame are represented by a colour histogram saved in the database. These features define the characteristics of images that are relevant to the database. As a result, the database can be relied upon to distinguish and classify images.

The databases can store huge amounts of

multimedia data because of the advancement of technology and the low cost of hard disk presently. The major difficulty is effectively locating images and video frames within databases. The searching process would be time-consuming if it needs to access the whole database, which may contain one million pictures and videos. Hence the clustering approach, and algorithms such as Isodata and K-means have been invented to reduce the searching time. Clustering refers to the process of grouping similar feature vectors together. The groups are termed "clusters". Clustering applied to a multimedia database groups frames and images into similar grey levels, colours and textures for determining various regions in the images. The "centroid" is an important term within a cluster. It is the average value of feature vectors in each cluster.

Searching would be based on a clustered or non-clustered histogram. A non-clustered approach would not be concerned with a "centroid" comparison. However the feature vector of the query feature vector is first compared with the centroid of each cluster through a clustered search. The query feature vector would only make a comparison with the most similar group as the distances between the query feature vector and the centroid of all clusters are sorted in order. In this paper, five clustering algorithms : Forgy, K-means, Isodata, Two-level clustering and One-level clustering were used to test the proposed CBVRS. Section 3 describes in more detail about these five clustering algorithms.

2.1 Wavelet Based Video Retrieval System (WBVRS)

In the proposed WBVRS, each image is processed using the following steps:

DWT – Images are analysed in the time-frequency domain using a 2D discrete wavelet transformation.

Clustering – Images are classified into different groups using colour deviation and the wavelet coefficients.

2.1.1 Wavelet Based Feature Vectors

In WBVRS, both the Harr [5] wavelet and Dubechies [6] wavelet were used.

2.1.1.1 Three-Level Wavelet Transform

Three-level wavelet transform (both Haar and Daubechies D4 wavelets) was applied to an image, separating the image into different frequency bands.

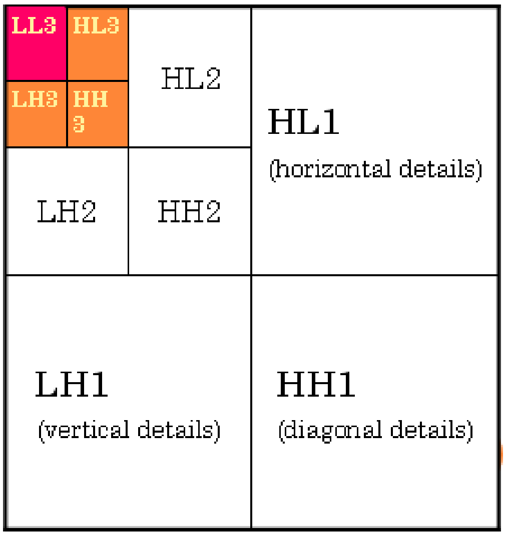

Fig. 1 3-Level Wavelet Transform

The 28 feature vectors are obtained from below:

(D0, D1, D2, D3, ⟵ Color variation
H0, H1, H2, H3, H4, H5, H6, H7, ⟵ Color Histogram component
H8, H9, H10, H11, H12, H13, H14, H15,
H16, H17, H18, H19, H20, H21, H22, H23)

The first four elements of the feature vector D0 – D3 are the colour variations (mean deviation) in Y colour channel in the LL3, HL3, HH3, LH3 bands. The mean deviation is the mean of the absolute deviations of a set of data about the data's mean. For a sample size *N*, the mean deviation is defined by

$$\mathrm{MD} \equiv \frac{1}{N}\sum_{i=1}^{N} |x_i - \bar{x}|,$$

where $\bar{x}$ is the mean of the distribution.

The H0 - H23 are the colour histograms in three colour channels in the LL2 band. The feature vectors produced by this algorithm have two components: the colour variation and the colour histogram.

The D0 - D3 comprise the colour variation component of the feature vector, and they describe the colour changes of an image. For example, an image with similar colours will have low value in colour variation while an image with sharp and frequent colour changes will have high value. The D0 - D3 can be considered as a very approximate description of an image, and D0 - D3 is used in the first-level clustering stage.

The H0 - H23 elements compose the colour histogram component of the feature vector: they are the histograms in the LL3 band in three colour-channels with each colour channel divided into eight groups. H0 - H7 are the histograms in the Y colour channel; H8 - H15 are the histograms in the Cb colour channel; and H16 - H23 are histograms in the Cr colour channel. The histogram information is a more detailed description of an image, especially the colour distribution of an image. The H0 - H23 range is used in the second-level clustering.

2.1.1.2 Two-Level Wavelet Transform

Two-level wavelet transform is similar to the Three-level wavelet transform, and it only applies to 2-level transform of the image. Fig.2 shows the frequency bands of Two-level wavelet transform.

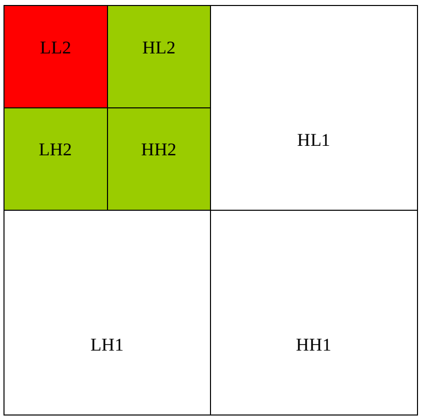

Fig. 2 Two- Level Wavelet Transform

Similar to Three-level transform, Two-level transform has 28 dimensions and it was formed by 2 color components of an image, they are colour distribution and color variation.

The feature vectors of the wavelet are described as

(D0, D1, D2, D3, ← Color variation
H0, H1, H2, H3, H4, H5, H6, H7, ← Color Histogram component
H8, H9, H10, H11, H12, H13, H14, H15,
H16, H17, H18, H19, H20, H21, H22, H23)

The first 4 elements of the feature vector D0 to D3 are the colour variation in Y colour channel in the LL2, HL2, and HH2 bands. The H0 to H23 are the colour histograms in three colour channels in the LL2 band.

3 Clustering Algorithms

Many retrieval systems calculate the features similarity between the query image and all images in the database and rank the images by sorting their similarities. The problem of this full search approach is very time consuming for large scale databases.

The retrieval time of this full search is the sum of the time to calculate similarity time $\mathbf{T}_{sim}$ and the time to sort the images in the database according to their similarity $\mathbf{T}_{sort}$. Time for the full search $\mathbf{T}_{full}$:

$$\mathbf{T}_{full} = \mathbf{T}_{sim} + \mathbf{T}_{sort}$$
$$\mathbf{T}_{full} = \mathbf{n}\mathbf{T}_{sim(1)} + \boldsymbol{O}(\mathbf{nlog\ n})$$

Where
n : number of images in the database
$\mathbf{T}_{sim}$: total time to calculate the similarity
$\mathbf{T}_{sort}$: total time to rank n images
$\mathbf{T}_{sim(1)}$: time to calculate similarity between two images
$\boldsymbol{O}$**(nlog n)**: time to sort n elements

If the images in the database are clustered, the retrieval time is the sum of

1) The time to calculate the similarity between the query and the cluster centers,
2) The time to calculate the similarity between the query and the images in the nearest clusters, and
3) The time to rank the images in step 2.

Time for cluster search $\mathbf{T}_{cluster}$:
$$\mathbf{T}_{cluster} = \boldsymbol{k}\,\mathbf{T}_{sim(1)} + \boldsymbol{l}\,\mathbf{T}_{sim(1)} + \boldsymbol{O}(\boldsymbol{l} \log \boldsymbol{l})$$
Where $\boldsymbol{k}$: number of clusters
$\boldsymbol{l}$: number of images in the clusters nearest to the query

Since $\boldsymbol{k} \ll \mathbf{n}$ and $\boldsymbol{l} \ll \mathbf{n}$, therefore the clustering search time $\mathbf{T}_{cluster}$ should be much smaller than the full search time $\mathbf{T}_{full}$ (i.e. $\mathbf{T}_{cluster} \ll \mathbf{T}_{full}$).

In this work, three different clustering algorithms: Forgy, K-Means and Isodata were tested. Sections 3.1, 3.2 and 3.3 provide an introduction to these three algorithms.

3.1 Forgy

The forgy algorithm is one of the simplest partitional clustering algorithms [4]. Assume k clusters are to be created. k samples, which are also called “seed points”, are chosen either randomly or specifically. These k samples are assigned to k clusters, respectively. Each cluster’s centroid is calculated with the seed sample’s feature vector information. For the remaining samples, the nearest cluster to each sample is determined, and the sample is assigned to that cluster. This completes the initialisation process. After initialisation, centroids are required to recompute. Then, all samples are examined. If any sample needs to change cluster and if there is a change, the centroids of the affected clusters are to be recomputed again. This step is repeated until no samples change their clusters anymore, which means that the algorithm is completed.

3.2 K-Means

The K-Means clustering algorithm [4] is as follow:

1. Begin with k clusters, each consisting of one of the first k samples. For each of the remaining (n-k) samples, find the centroid nearest it. Put the sample in the cluster identified with this nearest centroid. After each sample is assigned, recompute the centroid of the altered cluster.
2. Go through the data a second time. For each sample, find the centroid nearest it. Put the sample in the cluster identified with this nearest centroid. Do not recompute any centroid in this step.
3. If no samples change cluster in Step 2, stop.
4. Go to Step 2.

3.3 Isodata

Compared to other clustering algorithms, the Isodata [4] tries to minimize the squared error by assigning samples to the nearest centroid. It deals with *k* clusters where *k* is allowed to range over an interval that includes the number of clusters requested by the user. The following steps need to be followed:

1. Initialize the cluster centroids to the seed points.
2. Find the cluster centroid for each sample.
3. Compute the new centroids.
4. If more than one sample has changed clusters and the number of iteration is less than *start_i*, go to step 2.
5. Discard clusters and samples if there are fewer than *e-min* samples.
6. If the number of clusters is greater than or equal to 2 x *c_no* or the number of this iteration is even, go to step 7; otherwise, go to step 8.
7. If the distance between two centroids is less than *d_min*, merge these clusters and update the centroid; otherwise, repeat this step and go to step 8.
8. If the number of clusters is less than or equal to *c_no* / 2 or the number of this iteration is odd, go to step 9; otherwise, go to step 10.
9. Find the standard deviation for the complete original set of samples. If none, compute the mean within the cluster. Compute the centroids of these two clusters. If the distance between these centroids is greater than or equal to 1.1 x *d_min*, replace the original cluster by these two clusters; otherwise, do not split the cluster.
10. If there is no change in cluster when the number of iterations greater than *i_max*, stop. Otherwise, take the centroids of the clusters as new seed points and go to step 2.

where:

c_no – the desired number of clusters
e_min – the minimum number of samples per cluster
d_min – the minimum distance between centroids without merging
start_i – maximum number of iteration in the first part.
i_max -- maximum number of iteration in the main part.

3.4 Two-level Clustering

All 28 feature vectors from section 2.1.1.1 and 2.1.1.2 use Forgy, K-Means and Isodata clustering algorithms. The system compares the query feature vectors to the centroid of each cluster by sorting, the distances between each centroid to the query feature vectors. The detailed comparison is made inside the cluster that is the closest to the query feature vectors.

Two-level clustering consists of two stages, as shown in Figure 3. In the initial stage, the first-level clustering is processed based on the color variation component of the feature vector. The image database is divided into a small number of clusters so that feature vectors are similar in terms of color variations within each cluster. After the first-level clustering, the centriod of a first-level cluster is the average color variations of all of the feature vectors in that cluster. At the later stage, each cluster obtained from the first-level clustering is further divided into sub-clusters based on the color histograms component. This is considered as the second-level clustering. The centroid of a sub-cluster is the average histograms of all of the feature vectors in that cluster.

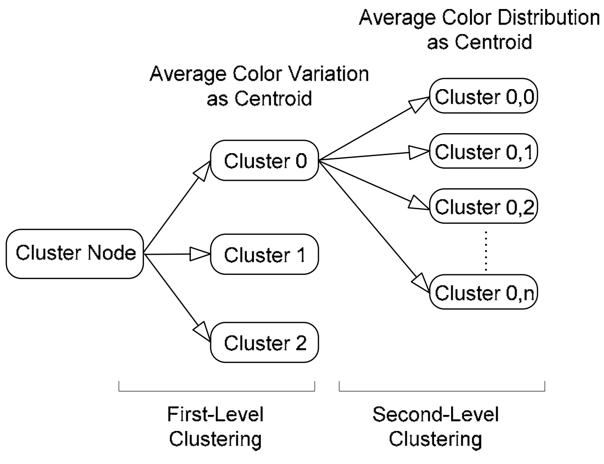

Fig. 3 The block diagram of Two-level Clustering

3.5 One-level Clustering

Fig.4 shows one-level clustering, which consists of one-step process. The feature vectors that were extracted by wavelet transform are partitioned according to their similarity. The centroid of each cluster, which represents the average feature vectors will then be calculated.

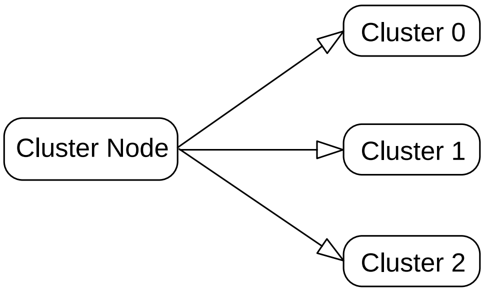

Fig. 4 The block diagram of One-level Clustering

4 Testing Results

4.2 Wavelet based CBIRS

A series of 25 video clips was taken with different objects (e.g, dog, bowl, clock, etc.), with over 1,000 frames within each video clip. With a 12 seconds video clip, over 1,000 image frames are produced if every single image frame were extracted. In this experiment, the first image on the first frame was captured, followed by every 10th image frame until the last frame. The total number of image frames captured from a video clip can be represented by

Number of image frame extracted = Total Number of Frames / 10

As different video format vary in their use of color encoding systems, the experiment described in this paper follows the Australian video signal standard, which is the PAL (Phase-Alternating Line) system. The PAL colour system is usually used with a video format that has 625 lines per frame and a research rate of 50 interlaced fields per second (or 25 full frames per second). By adopting the above formula, it follows that if images are captured in every 10th frame, each image, therefore, can be seen as if it were captured every 0.4 second. The average number of image captured from the selected video clip is 40. With 25 video clips being selected, therefore, an additional 1000 images were inserted into the image database. With the already-existing 1,500 images, the total number of images used were 2,500 in the tests.

In order to examine the different kinds of feature extraction algorithms and clustering techniques, the experiment was designed to have combinations of each of the algorithms tested with one clustering technique. A total number of 20 query images was selected and excluded from the image database. The following table shows the result of the experiments (results show both the accuracy percentage and speed performance (in brackets, measured in seconds)):

	Haar Wavelet (2-level transform)	**Haar Wavelet (3-level transform)**	**Daubechies Wavelet (2-level transform)**	**Daubechies Wavelet (3-level transform)**	**Performance average (in seconds)**
No Clustering	87% (0.3469)	**89%** (0.3734)	81% (0.3659)	81% (0.422)	**0.37705**
(Forgy)	81.5% (0.0358)	82% (0.0376)	79.5% (0.0702)	77.5% (0.0438)	**0.04685**
(K-Means)	84.5% (0.1079)	85% (0.1328)	79% (0.1173)	80.5% (0.1423)	**0.125075**
(Isodata)	88% (0.1139)	86% (0.1264)	79.5% (0.1272)	73.5% (0.1454)	**0.128225**
Accuracy Average	85.25%	85.5%	79.75%	78.125%	

Fig.5 Comparison of Haar and Daubechies Wavelet with 3 different clustering algorithms

The experimental results depicted in Fig. 5 clearly show that the clustering technique boosts up performance by 3 to 9 times. Forgy clustering performed reasonably well, compared with other clustering techniques, taking only 0.04685 seconds on average to perform a query.

In terms of accuracy, the Haar Wavelet with 3-level transformation performed slightly better than the Haar Wavelet with 2-level transformation. The interesting finding was that the choice of clustering technique does not affect the searching accuracy. Fig.6 shows the examples of 20 query images used in the tests. Fig. 7 shows the searching result of Haar wavelet (2-level transform) with one-level clustering using Isodata. Fig. 8 shows the searching result of Haar wavelet (3-level transform) with one-level clustering using K-means. Both Fig.7 and Fig.8 have 100% searching accuracy.

5 Conclusion

Multimedia Information Retrieval is an important research area. In this paper, we have proposed and implemented a content based video retrieval system based on the Haar wavelet and Daubechies D4 wavelet transform. The experimental results proved that wavelet transform would perform better than the colour-based algorithm when dealing with video sequence images. The Haar wavelet with 3-Level transform performed the best with high searching accuracy rate (89%). The results in Section 4 show that the clustering algorithms in the proposed CBVRS system can speed up the searching time by 7 times in One-Level clustering with Forgy.

References:

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

[2] A.Pentland, R.W. Picard, and S. Sclaroff, "Photobook: *Content-based manipulation of image databases*, In B.Furht, editor, Multimedia Tools and applications, Chapter 2, pp43-80. Kluwer Academic Publisher, 1996.

[3] J.R. Smith,"Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression. *PhD thesis*, *Columbia University*, 1997.

[4] Eric W. Wesisstein, "K-Means Clustering Algorithm" http://mathworld.wolfram.com/KMeansClusteringAlogirthm.html

[5] W.W. Weisstein," Harr Function", http://mathworld.wolfram.com/HaarFunction.html

[6] J. J. Benedetto, M. Frazier,"*Wavelet: Mathematics and Applications* Boca Raton, FL: CRC Press 1994.

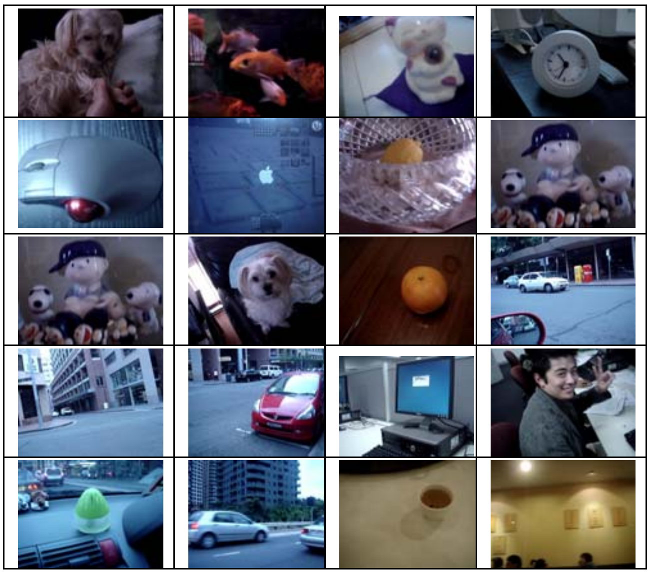

Fig.6 The examples of 20 query images used for the tests

Fig.7 Searching result of Haar Wavelet (2-level transform) with one-level clustering (Isodata)

Fig.8 Searching result of Haar Wavelet (3-level transform) with one-level clustering (K-means)