6G－2 New Indexing Method for Content-Based Video Retrieval and Clustering for MPEG Video Database

Zaher AGHBARI, Kunihiko KANEKO, and Akifumi MAKINOUCHI *

1 Introduction

Recently, multimedia information, especially video, have vastly increased; as a result, an automatic content-based retrieval became an important research issue in multimedia database. The basic units of a video are a 'frame' and a 'shot'. The results of content-based retrieval is a set of shots that satisfy the user query. In this paper, we introduce an 'average frame' as a logical representative frame of a shot. The average frame is a compact representation of 2 features: color distribution and their locality information of a shot. That would enable users to issue queries such as:

```
select S
from   S in shots
where  S.match(<locality_info>, <color info>)
```

Here, 'shots' is a set of shots in a database, and the predicate 'match' is the operator for retrieval. To retrieve a shot a user has to specify some location, or rather a field of movement which is an area in a frame within which the searched object is moving during the time of a shot, and then specify the amount of color(s) in reference to that specified location.

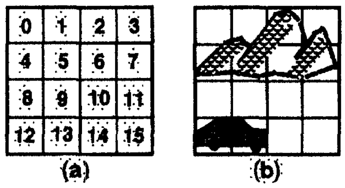

Figure 1: a video shot

2 Related Work

In the previous content-based video management systems, such as the QBIC [2], JACOB [1], video sequence are divided into independent shots where each shot is basically a single camera operation. Each shot is then represented by one or more frames, called keyframes. From those keyframes color information are extracted by computing one color histogram for every keyframe.

In those systems, a user queries a shot by estimating the amount of color(s) in reference to the whole frame of a video shot. For example, to query for the red car with black tires in Figure 1-b, the query should be as follows:

```
select all shots
match(red 10%, black 2%)
```

But, if there are some red objects in other regions of the frame, the above query will not be effective since the amount of red in the index (color histogram) of the shot is the cumulative value of all red objects in the frame.

Our system presents a solution to this problem via providing locality information to the color distributions by dividing each frame of a video shot into several subregions. Therefore, a user querying for a moving car, from left to right, has to specify the field of movement (locality information) and the amount of red and black colors in reference to the specified region. With this method, we were able to isolate the effect of other objects having the same color(s) in other regions of the frame, and the query would be as follows:

```
select S
from   S in shots
where  match(subregions [12, 13, 14, 15],
             red 40%, black 5%)
```

3 System Overview

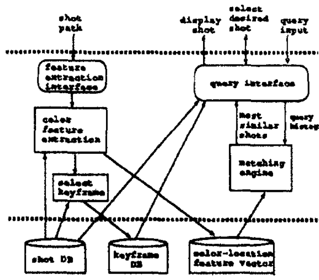

Figure 2: system layout

The system consists of two main functional modules: the first one is dedicated to the generation of color-location index vectors and the selection of keyframes that will best represent the corresponding shots, and the second module is dedicated to video database querying and matching.

3.1 Index Generation

First, the video sequence is divided logically into independent shots. Since the frames of an MPEG coded

*Graduate School of Information Science and Electrical Engineering, Department of Intelligent Systems, Kyushu University 6-10-1 Hakozaki, Higashi-ku, Fukuoka-shi 812-81, Japan

shot are close in terms of color distribution, only the intra-frame (I-frame) will be selected to contribute to the computation of the indexes. Then, the RGB full color space is quantized to 64 colors which is a manageable set of color. Each of the selected I-frames are, then, divided into 16 subregions which is a number that is not too large to be tedious for the user to specify queries with, nor too small to lose the effectiveness of locality provision. Then, a color histogram is computed for every subregion to represent the color information in that particular subregion. That captures the locality information. The average of all color histograms of the same subregion in all the frames of a shot are computed, as illustrated in Figure 3.

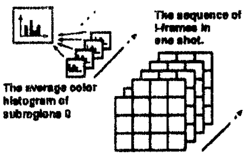

Figure 3: averaging color histograms

Each subregion is then represented by the 10 major colors in its computed average color histogram, since the rest of the colors usually average out to a negligible value as found from the testing of the system. The resulting average color histograms are considered as an abstract frame that sufficiently represents the corresponding shot and are used to construct the color-location feature vectors which are the indexes of the shots. Then, a frame that is closest to those computed average color histograms will be selected as a representative frame and stored in a keyframe database. These keyframes will then be displayed to the user representing the most similar shots to the user's query as a result of a matching process.

3.2 Matching Engine

Our matching method is derived from the Cross-Correlation method [2]. It takes into consideration the perceptual similarity between the different colors and reduces the contribution of background colors since only the colors specified in the user query and/or their perceptually similar colors will contribute to computation of the metric distance values between the user query and the shots' indexes. The metric equation is

$$d = \left| \sum_{i=1}^{m} [(I_i - J_i) + \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} * (I_j - J_j)] \right|$$

where I and J are the query histogram and target histogram, respectively. m and n are the number of colors in the query histogram and target histograms respectively. a_{ij} is the perceptual similarity coefficients between the different colors.

As a result of matching the user query against the shots DB, a similarity rank list is created which is sorted in ascending order on the computed distance value. It also contains pointer to the corresponding keyframes and shots.

4 Testing

Testing of the system was performed on 250 shots which were a mixture of animation, educational, and real movie video sources. Several queries were formulated to test the system. The results were encouraging and proved the validity of our method. The distinguishing characteristic of our method, is that a user can specify a location, a field of movement for moving objects, and then specify the amount of color(s) in reference to the specified location. For example, the following query was formulated to search for a shot that contains a moving pink ball from left to right in subregions 4, 5, 8, and 9, which is the second shot from the left shown in Figure 4, and the amount of pink in reference to those four subregions is about 30%. Figure 4 shows the first four shots in the resulting rank list. A comparison test between our matching method and the Cross-Correlation method showed that our method is more than three times faster with even better retrieved set of shots.

```
select S
from   S in shots
where  match(subregions [5, 6, 9, 10], pink 30%)
```

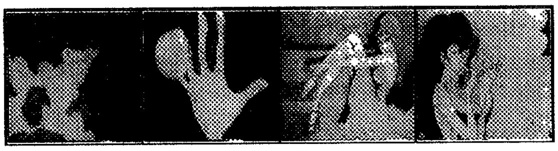

Figure 4: query for a pink a moving pink ball

5 Conclusion

In this paper we have presented a method to index video shots compactly by their color features and locality information. Thus enabling users to retrieve shots by specifying a certain location within a frame, field of movement for a moving object, and then specify the amount of color(s) in reference to the specified location. Our test results showed the effectiveness of such queries.

References

[1] M.L.Cascia, E.Ardizzone. JACOB:Just a content-based query system for video databases. ICASSP, May 7-10, 1996, Atlanta.

[2] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D, Lee, D. Perkovic, D. Steele, P. Yanker. Query by Image and Video Content: The QBIC System. IEEE, Sept. 1995.