Efficient Video Indexing and Retrieval by Color

2W−1

Zaher AGHBARI Kunihiko KANEKO Akifumi MAKINOUCHI
Graduate School of Information Science and Electrical Engineering,
Department of Intelligent Systems.
Kyushu University 6-10-1 Hakozaki,
Higashi-ku, Fukuoka-shi 812-8581, Japan

1 Introduction

Retrieving videos by content has been the focus of many researches due to the recent advances in data compression and networking technology. Consequently, new applications have emerged such as distance learning, digital libraries, public information systems, movies-on-demand, just to name a few. However, to fully utilize the potential benefits of these applications, a new breed of efficient and effective methods to index and retrieve videos are necessary.

Sequential scanning methods are impractical for video databases due to the large number of accessible videos. In addition, traditional indexing methods using the Spatial Access Methods (SAM), such as G-trees, B-trees, R-trees, ... etc., are not efficient in case of video data. That is because the high dimensionality causes more overlap among the *minimum bounding rectangles* (MBRs) of feature vectors; resulting in large number of search paths, thus slower response time [2]. For example, R-trees [3], or its variants, are not proven to be efficient when indexing high-dimensional color feature (64, or more, dimensions); however they are found to be very efficient when indexing low-dimensional data [1, 4]. This requirement calls for techniques to reduce dimensionality of features so that they can be efficiently indexed by a SAM, such as an R-tree.

In this paper, we propose a method to index and retrieve shots efficiently and effectively based on the colors of salient objects that exist in the shot. The idea is to extract keyframes that best represent the content of a shot and then from these keyframes salient objects are segmented. The colors of each object are represented by a color histogram. Each color histogram, which is highly dimensional, is mapped into a point in a low-dimensional distance space. These points are indexed by an R-tree in order to retrieve these points efficiently. Our method reduces the dimensionality and at the same time guarantees no *false dismissals*, points that satisfy the user query but not returned in the result.

2 Video System

Our video data are *shots* collected from the Internet. Each shot undergoes several processing steps: (1) a shot is divided into several *events*, where an event is a consecutive supsequence of frames that contain fixed number of meaningful objects, (2) two, or more, keyframes are extracted from each event, (3) salient objects are extracted from each keyframe. In this paper, we assume that shots are coded by an object-based video encoder such as MPEG-4. In MPEG-4 objects with arbitrary shapes are encoded apart from their background. Therefore, segmentation information of objects is provided in the video input stream.

From each extracted object, the color feature (color histogram) is computed. In this system, the RGB color space is quantized into 64 equally spaced colors, where each color consists of 3 components: red (*r*), green (*g*), and blue (*b*). A color histogram *C* contains the major 10 colors of object O_i. Thus, *C* of O_i at keyframe κ_j is defined by:

$$C : \{(r_1, g_1, b_1, p_1), \ldots, (r_l, g_l, b_l, p_l), \kappa_j\} \qquad (1)$$

Where, p_i is the percentage of color c_i that is represented by the r_i, g_i, and b_i, color components. We set $l = 10$, which is the maximum number of major.

2.1 Shot Indexing

The color histograms, which are points in high-dimensional color space, are first mapped into points in low-dimensional distance space using the *topological modeling* method. Then, we use an R-tree to organize and cluster these low-dimensional points. So that when a query is issued, the cluster (a small subset of the database) that is most similar to the user query is retrieved. This retrieved cluster contains all relevant shots and a few irrelevant shots, called *false alarms*. To remove those irrelevant shots we propose a two-step filtering process to refine the result.

Topological modeling method maps a color histogram into a 3-dimensional distance space. Where, the 3 dimensions are the 'red' (R_d), 'green' (G_d) and 'blue' (B_d) *distance* spaces (or topological spaces). As shown in Figure 6.a, mapping the color histogram into the $R_dG_dB_d$ distance space is achieved by, first, computing the distance $D(C, O_R)$ between the color histogram *C* and a *virtual* red object O_R that is assumed to be at the origin of the 'red' topological axis (distance space) and then a point representing a color histogram is placed in the 'red' topological axis based on the computed distance $D(C, O_R)$. The distance $D(C, O_R)$ is computed using Equation 4. Similarly, the distance $D(C, O_G)$ between *C* and a *virtual* green object O_G is computed, and also the distance $D(C, O_B)$ between *C* and a *virtual* blue object O_B is computed. Accordingly, the $D(C, O_G)$ and $D(C, O_B)$ points are placed on the 'green' topological axis and the 'blue' topological axis, respectively. Combining the three distances, as shown in Figure 6.b, will result in a point representing *C* in a 3-dimensional distance space.

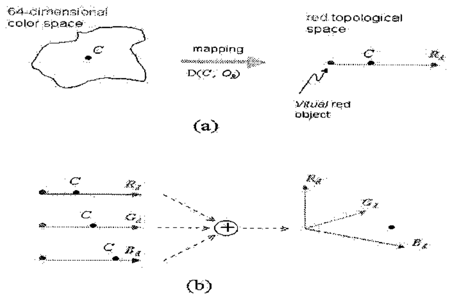

Fig. 6: (a) Mapping a color histogram C into a point in the 'red' topological space, (b) Combining the three topological spaces.

Each shot S_i is represented by several keyframes $\kappa_1, \kappa_2, \ldots, \kappa_{Nk}$. Where, *Nk* is the number of keyframes extracted from a shot. Each of these keyframes contains at least one object. However, the number of objects $N_o(\kappa_j)$ in each keyframe may be different. Thus, each shot is represented by $N_o(S_i)$ points in low-dimensional distance space.

$$N_o(S_i) = \sum_{j=1}^{Nk} N_o(\kappa_j) \qquad \ldots\ldots\ (2)$$

2.1 Shot Retrieval

Each specified color histogram in a query Q will be represented by one query point qp_i. Therefore, Q is represented by m points, which is equal to the number of specified color histograms in Q.

$$Q = qp_1, qp_2, \ldots, qp_m \qquad \ldots\ldots(3)$$

First Filtering Process: Each query point qp_i is matched with the points in the R-tree and the cluster (a small subset of the points in the R-tree) that is most similar to qp_i is retrieved. As a consequence, m clusters (refer to Equation 3) are retrieved. These clusters are merged and a list of shots is generated. The list also contains values that represent the number of retrieved points N_p that belong to each shot. Where,$1 \leq N_p \leq m$. The number of retrieved points N_p is used as an Initial filtering step. The shots whose $N_p < m/2$ are removed from the list because their minimum distance D_p from Q is greater than 0.5.

Second Filtering Process: After the first filtering process, the remaining list of shots (whose $N_p \geq m/2$) may contain false alarms. To remove them, we provide the *sequential_match* algorithm to match Q with each of the remaining shots using the original feature vectors of Q and each shot, where the distance function is:

$$D(I, J) = \left| \sum_{i=1}^{g} [(I_i - J_i) + \sum_{\substack{j=1 \\ j \neq i}}^{l} a_{ij}(I_j - J_j)] \right| \qquad \ldots.(4)$$

Where, a_{ij} is a perceptual similarity between two colors (I_j, J_j). g and l are the number of colors in I and J, respectively. Since these remaining shots are only a very small subset of the all the shots in the database, this second filtering process will not degrade performance much.

3 System Evaluation

We conducted two types of experiments to measure the effectiveness and efficiency of our method: specifically, the first type of experiments measures the precision and recall of the method. The second type of experiments measures how efficient our method is in locating a wanted shot. These experiments were performed on a collection of 190 shots categorized into sports (such as bike racing, car racing, skiing, soccer, football), movie, and animation.

Precision & Recall: Specifically, *precision* and *recall* metrics are used to measure the effectiveness of a system. Where, *precision is the ability of a system to reject false alarms* and *recall is the ability of a system to retrieve all relevant shots.* To measure precision and recall of our method, we issued 10 sample queries to search for 10 randomly selected shots. Then, we compared the returned list of shots of each sample query with its ground truth. The average recall curve, Figure 2, shows a smooth increase of the recall measure as the number of returned shots increases. On the other hand, the average precision is quiet high when the number of returned shots is small and the curve starts to slope down as the number of returned shots increases. That is due to the fact that the chance of having some false alarms increases as the number of returned shots increases.

Response Time: the response time to queries is *the elapsed time from the start of execution of a query till the receipt of the result (till the end of the second filtering process).* The average response time of 25 sample queries, where the number of objects is varied from 1 to 5, is shown in Figure 3. Notice, our method reduces the response time to queries by about 70% as compared to the sequential scanning method. This result is a considerable saving to the search time of shots.

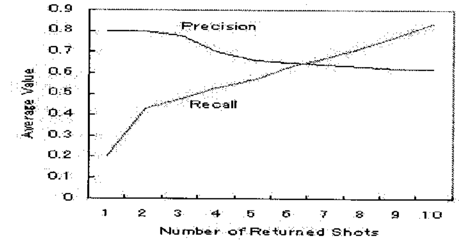

Fig. 2: *Precision and recall curves*

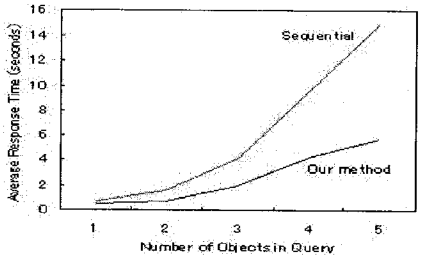

Fig. 3: *Comparison between our method and a sequential method*

4 Conclusion

The conducted experiments showed that our method is efficient and effective as compared to a sequential scanning method. Also, from these experiments, we conclude that the color feature of an object that exists in a shot can be sufficiently represented by a point in 3-dimensional distance space.

References:

[1] C.Faloutsos and K.Lin. *A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets.* ACM SIGMOD, May 1995.

[2] C.Faloutsos, M.Ranganathan and Y.Manolopoulos. *Fast Subsequence Matching in Time-Series Databases.* ACM SIGMOD, May 1994.

[3] A.Guttman. *R-trees: a dynamic index structure for spatial searching.* ACM SIGMOD, pp.47-47, June 1984.

[4] B.Yi, H.V.Jagadish, and C.Faloutsos. *Efficient Retrieval of Similar Time Sequences Under Time Warping.* ICDE98, Orlando, Florida, Feb.23-27, 1998.