A CONTENT BASED MULTIMEDIA RETRIEVAL SYSTEM

Payel Saha
TCET, Thakur Village, Kandivali (E), Mumbai – 101,

Sudhir Sawarkar
D.M.C.E., Airoli-708,

ABSTRACT

Multimedia search and retrieval has become an active field for many contemporary information systems. This paper presents a scheme of retrieving a multimedia object, i.e. a video clip with audio. For video retrieval, the system searches a particular query video clip from a database of video clips by matching on the basis of motion vector analysis. For audio retrieval, the audio from the query is to be separated and matched using the fingerprint algorithm with all the audio files of the videos from the database and provide rankings to the matched files.

Key words: Multimedia, CBVR, Query, Image, Audio, Motion Compensation

I. INTRODUCTION

The increasing popularity of digital video content has made the demand of automatic, user-friendly and efficient retrieval of video collection becomes an important issue [11, 17]. VideoQ [3] is the first on-line content-based video search engine providing interactive object-based retrieval and spatiotemporal queries of video contents. Some commercial search engines; such as Google and Yahoo!, have started to extend their services to video searching on the Internet, and it is already possible to search for video clips by typing keywords. However, commonly adopted features, such as colour, texture, or motion are still insufficient to describe the rich visual content of a video clip. In the past few years, the area of content-based multimedia retrieval has attracted worldwide attention. Among the different types of features used in previous content-based video retrieval (CBVR) systems, the motion feature has played a very important role [13]. Multimedia search and retrieval has become an active field after the standardization of MPEG-7. The syntactic information used in MPEG-7 includes color, texture, shape and motion. The technology of moving-object tracking plays
an important role in those video retrieval systems. At first, motion and color information extracted from the MPEG-2 compressed stream to determine which pixels belong to moving objects and which pixels belong to static background for a period of time. This kind of technologies can help us find some interesting events from video data and avoid tedious searching processes.

II. SYSTEM FUNCTIONS

A video retrieval system consists of the search for a particular query video clip from a database of video clips. Following are the basic functions of the system:

1) Matching Video: The motion vector for each video query is unique and comparing it with that from the database will provide an ideal match. This is based on the assumption that the entire query video is available in the database.

2) Matching Audio: The audio from the query is separated and matched using the fingerprint algorithm with all the audio files of the videos from the database and provide rankings to the matched files.

3) Search rankings post processing: After the final processing of audio and motion vector, the system should provide rankings to the search results. This will be on the basis of the motion vector.

![Flowchart of the system](image.png)

**Fig. 1** Flowchart of the system
III. AUDIO PROCESSING

Audio retrieval implements a landmark-based audio fingerprinting system that is very well suited to identifying small, noisy excerpts from a large number of items. Each audio track is analysed to find prominent onsets concentrated in frequency, since these onsets are most likely to be preserved in noise and distortion. These onsets are formed into pairs, parameterized by the frequencies of the peaks and the time in between them. These values are quantized to give a relatively large number of distinct landmark hashes [18].

Each reference track is described by landmarks it contains and the times at which they occur. This information is held in an inverted index, which, for each of the 1 million distinct landmarks, lists the tracks in which they occur and when they occur in those tracks. To identify a query, it is similarly converted to landmarks. Then, the database is queried to find all the reference tracks that share landmarks with the queries, and the relative time differences between where they occur in the query and where they occur in the reference tracks. Once a sufficient number of landmarks have been identified as coming from the same reference track, with the same relative timing, a match can be confidently declared.

IV. MOTION DETECTION

Here, some image frames without moving objects are used to compute statistical quantities for the background scene. Then, the foreground pixels are detected and features extracted.

A. Background image

The initial background image modelling is carried out over first 50 image frames. With the assumption of no moving objects in the 50 image frames, the reference colour background image with a normal distribution is built. The background is modelled by computing the sample mean $\mu(x,y)$ and variance $\sigma^2(x,y)$ in the colour images over a sequence of 50 frames. These statistics are calculated separately for each one of the RGB components by using the following iterative formula [4]. For image frame $f = 1, \ldots, 50$, we have,

$$[\mu(x,y)]_f = [\mu(x,y)]_{f-1} + \frac{1}{f}([C(x,y)]_f - [\mu(x,y)]_{f-1})$$

(1)

$$[\sigma^2(x,y)]_f = \frac{f-1}{f} [\sigma^2(x,y)]_{f-1} + \frac{1}{f} ([\mu(x,y)]_f - [C(x,y)]_f)^2$$

(2)

where $[.]_f$ denotes the corresponding value at frames $f$, and $[\mu(x,y)] = [C(x,y)]$ and $[\sigma^2(x,y)] = 0$ Equations (1, 2) can be shown by straightforward calculations that they yield the sample mean and variance over the first 50 image frames [4]. The sample mean is the background image. The background image and the variance of the $(x, y)^{th}$ pixel's RGB values over the first 50 image frames is given by [5]:

$$\mu(x,y) = [\mu_R(x,y), \mu_G(x,y), \mu_B(x,y)]$$

(3)

$$\sigma^2(x,y) = [\sigma^2_R(x,y), \sigma^2_G(x,y), \sigma^2_B(x,y)]$$

(4)

B. Background subtraction

The term “moving pixels” is defined as the “foreground.” In each new image frame $C(x,y)$, the foreground can be obtained by comparing their RGB values to the corresponding mean values. First a binary image $D(x,y)$ with the same dimension as the image $C(x,y)$ is created and all its pixel values are set to 0. The output of the background subtraction method is defined as follows:
A pixel \((x,y)\) is extracted as a foreground if it RGB values \(C(x,y)\), satisfies the absolute difference value with \(\mu(x,y)\), where the parameter \(\alpha\) can be adjusted to yield more or less foreground. 4\(\sigma\) is used as the threshold in background subtraction. If the threshold is too low (e.g. 1\(\sigma\)), it will cause too much fault-foreground; on the contrary, if the threshold is too high (e.g. 5\(\sigma\)), it will cause too much fault-background.

C. Background updating

The background cannot be expected to be stationary for a long period of time. An adaptive scheme makes a constant updating of background as linear combination of previous background image and current image frame. The recursive estimation of mean and variance can be performed using equations (6) and (7). Equations (6) and (7) update the background image and sample variance, respectively [5, 8, 9]:

\[
\mu(x, y)_t = \beta \cdot C(x, y)_t + (1 - \beta) \cdot \mu(x, y)_{t-1}
\]

\[
\sigma^2(x, y)_t = \beta \cdot (C(x, y)_t - \mu(x, y)_t)^2 + (1 - \beta) \cdot \sigma^2(x, y)_{t-1}
\]

where \(C(x,y)\) is the current image frame; \(\mu(x,y)_t\), \(\sigma^2(x,y)_t\) the mean and variance values that update the current image frame; \(\beta\) is the learning rate that determines the speed at which the distribution's parameters change (0<\(\beta\)<1) [7].

D. Shadow elimination

The shadow problem will cause redundant foreground (moving pixels) and decrease the system’s accuracy. Therefore, there is a need to eliminate the shadow after background subtraction. Before the shadow elimination process, some pre-processing is needed:

1. The matching part of the image \(|C(x,y) - \mu(x,y)|\) from the RGB to greyscale format is created.
2. Gaussian filter is used to remove isolation points and smooth the old grayscale image.
3. The gray-level distribution of the new gray scale image is done and the minimal gaussian value is taken as threshold. A support map \(C'(x,y)\) is a RGB colour image where pixel values are set to current image frame if it is greater than threshold; otherwise, set to 0. This method is defined as follows:

\[
C'(x, y) = \begin{cases} 
C(x, y), & \text{if (greyscale image) > threshold} \\
0, & \text{otherwise}
\end{cases}
\]

The equation (8) maintains the motion parts (foreground, shadow, highlight, noise) of \(C(x,y)\) while the non-motion parts are removed.

E. Noise elimination

From above processes, the obtained foreground image may still have some noises. Those noises come from lighting changes, illumination changes and false matches. To eliminate noises and improve the foreground, connected component labelling and morphological operations are applied to
noise elimination. Connected component labelling is used to label all pixels that are determined as foreground and count the area of all the labelling components.

F. Motion vector computation
The process of computing motion vectors comprise of two steps:

1. Motion estimation.
2. Motion compensation.

1. Motion estimation
In compressed domain video indexing and retrieval, feature used is motion vectors. Motion is the most significant feature in video which represents two dimensional temporal change of video content [8] despite the conventional image features including color, texture and shape. It is possible to distinguish video and images in terms of motion. It is well known that a motion vector field is usually composed of camera motion, object motion, and noises. The global motion in a video is mostly contributed by camera motion. Therefore, the following four-parameter global motion model, which is fast and also valid for most videos [9], is used to estimate the camera motion from the motion vector field.

\[
\dot{M}V_{\text{cam}} = \begin{pmatrix} \text{zoom} \\ -\text{rotate} \end{pmatrix} \begin{pmatrix} \text{rotate} \\ \text{zoom} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \text{pan} \\ \text{tilt} \end{pmatrix} \tag{9}
\]

If the underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame. The idea behind block matching is to divide the current frame into a matrix of ‘macro blocks’ that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame.

The search area for a good macro block match is constrained up to p pixels on all fours sides of the corresponding macro block in previous frame. This ‘p’ is called as the search parameter. Larger motions require a larger p, and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macro block is taken as a square of side 16 pixels, and the search parameter p is 7 pixels. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block [1].

In MEPG video coding, each frame is divided into non-overlapping macro blocks (MBs) of size 16 ×16. For each MB, the motion vector reveals its displacement between the reference frame and current P-frame. The motion vector consists of a horizontal component, x, and a vertical component, y. Let \( mv = (mv^x, mv^y) \) denote the forward motion vector of a MB in a P-frame [10].

The motion object detection is obtained as:

Compute the magnitude, \( \rho \), and angle, \( \theta \), of the motion vector:

\[
\rho = \sqrt{(mv^x)^2 + (mv^y)^2} \tag{10}
\]
\[
\theta = \tan^{-1}\left(\frac{mv^x}{mv^y}\right) \tag{11}
\]
2. Motion compensation

In general, Motion compensation is an algorithmic technique employed in the encoding of video data for video compression, for example in the generation of MPEG-2 files. Motion compensation describes a picture in terms of the transformation of a reference picture to the current picture. The reference picture may be previous in time or even from the future. When images can be accurately synthesized from previously transmitted or stored images, the compression efficiency can be improved.

Using motion compensation, a video stream will contain some full (reference) frames; then the only information stored for the frames in between would be the information needed to transform the previous frame into the next frame.

Here block motion compensation (BMC) is used. In BMC, the frames are partitioned in blocks of pixels (i.e. macroblocks of 16×16 pixels in MPEG). Each block is predicted from a block of equal size in the reference frame. The blocks are not transformed in any way apart from being shifted to the position of the predicted block. This shift is represented by a motion vector. To exploit the redundancy between neighbouring block vectors, (e.g. for a single moving object covered by multiple blocks) it is common to encode only the difference between the current and previous motion vector in the bit-stream.

Block motion compensation divides up the current frame into non-overlapping blocks, and the motion compensation vector tells where those blocks come from. The source blocks typically overlap in the source frame.

V. COLOR EXTRACTION AND SIMILARITY MEASUREMENT

A. Color Descriptors

Color descriptors of images and video can be global and local. Global descriptors specify the overall color content of the image but with no information about the spatial distribution of these colors. Local descriptors relate to particular image regions and, in conjunction with geometric properties of these latter, describe also the spatial arrangement of the colors.

B. Color Histograms

Color histogram is the most widely used method owing to its robustness to scaling, orientation, perspective, and occlusion of images. The joint distribution of the three color channels is denoted by the histogram. The human perspective to color is a merger of three stimuli, R (red), G (Green), and B (Blue), which form a color space.

A colour histogram \[h(image) = (h_k(image) \ k = 1,\ldots,K)\] is a K-dimensional vector such that each component \(h_k(image)\) represents the relative number of pixels of colour \(C_k\) in the image, that is, the fraction of pixels that are most similar to the corresponding colour. To build the colour histogram, the image colours should be transformed to an appropriate colour space and quantized according to a particular codebook of the size \(K\).

The retrieval system typically contains two mechanisms: similarity measurement and multidimensional indexing. Similarity measurement is used to find the most similar objects. Multidimensional indexing is used to accelerate the query performance in the search process.

Similarity measurement plays an important role in retrieval. A query frame is given to a system which retrieves similar videos from the database. The distance metric can be termed as similarity measure, which is the key-component in Content Based Video Retrieval. In conventional retrieval, the Euclidean distances between the database and the query are calculated and used for ranking. The query frame is more similar to the database frame if the distance is smaller. If \(x\) and \(y\) are 2D feature vectors of database index frame and query frame respectively [14].
Then Euclidean distance is given as:

\[ D = \sqrt{\sum (x_i - y_i)^2} \]  \hspace{1cm} (12)

VI. GRAPHIC USER INTERFACES

The program runs with the help of two separate but linked Graphic User Interfaces (GUIs). The main GUI is used to obtain all the details of the query clip. Various computations required for this purpose are performed using different pushbuttons. The other retrieved GUI displays the final result at the end and works on the video files from the database.
The different functions performed by the main GUI are explained below.

A. Selection of video query
   The ‘Select Video’ pushbutton allows selecting any one video clip. The video clip is selected directly from the folder and played in media player. This is the query clip. The size of this clip is small and should be of minimum 5 seconds. It is a video file in AVI format.

B. Separation of audio
   The audio is separated from the query clip, which is in raw audio format i.e. in PCM format and played in a media player. This is an uncompressed audio format. This is executed by ‘Retrieved Audio’ pushbutton.

   After extracting the raw audio, it is required to convert it into standard lossy compressed .mp3 format for further processing. As most of the audio files used today are in mp3 format, which has a reduction in size than the raw audio. This can be done using any external converter within no time.

   A separate code is used to process this mp3 audio clip further. The code searches for the best possible match of the clip with the all audio files in the database which are again in mp3 format. From the listed matches, the top three best matches are obtained and sent for video processing.

C. Calculation of motion vector
   The first 50 frames of the query are taken under consideration for further processing and saved. The image data, matrix or 3-D array (RGB) of values specifying the colour of each rectangular area defining the image is obtained and the current colours are mapped.

   The next step is to detect the motion from the frames extracted. This process again includes different steps. After detecting motion, labeling is done; a new matrix contains label for the connected components of the input image (frame) is created. The input image can have any dimension; the label matrix (L) is of the same size as input image. The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make

---

**Fig. 3 Main Gui**

---
up one object; the pixels labeled 2 make up a second object; and so on. The default connectivity is 8 for two dimensions, 26 for three dimensions. This process is basically to labeling the background. Then at last motion vectors are obtained and motion compensation is done.

Fig 4 Axis displaying motion in the query clip

The videos that have been given the best rank in the audio retrieval are stored in a .mat file called X.mat. This file is then loaded to provide the list of the top 3 ranks on the basis of audio fingerprint matching. The computation begins after selecting ‘Click Here’ pushbutton.

The motion vectors for these videos are calculated by running the code from the initial main.m in a loop. Further Euclidean distance is found out between the frames of the query and those of the videos in the loop as well. If the Euclidean distance is found to be zero i.e. an exact match on the basis of colour is found, that portion of the query is played in a figure.

When processing is out of the loop, motion vectors are compared with the target motion vector and the difference is found out. The differences are arranged in ascending order thus providing us with the final rankings.

VII. RESULTS

The videos that have been given the best rank in the audio retrieval are stored in a .mat file called X.mat. This file is then loaded to provide the list of the top 3 ranks on the basis of audio fingerprint matching. The computation begins after selecting ‘Click Here’ pushbutton.

The motion vectors for these videos are calculated by running the code from the initial main.m in a loop. Further Euclidean distance is found out between the frames of the query and those of the videos in the loop as well. If the Euclidean distance is found to be zero i.e. an exact match on the basis of colour is found, that portion of the query is played in a figure.

When processing is out of the loop, motion vectors are compared with the target motion vector and the difference is found out. The differences are arranged in ascending order thus providing us with the final rankings.
The final result of this CBVR system yields three best possible matches of the small query clip selected earlier. As the ‘Play’ pushbutton is clicked, that particular video is played separately in any media player as shown in the figure below.

![Final Retrieved Video Clip](image)

**Fig 5** Final Retrieved Video Clip

**ACKNOWLEDGMENT**

The authors are grateful to the colleagues of respective Institutes for their motivation, and help towards the completion of this paper, as well as for providing valuable advice.

**REFERENCES**


Jau-Ling Shih, Ming-Chieh Chuang, “A Video Object Retrieval System using Motion and Color Features”, Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan, R.O.C.


Heinrich A. van Nieuwenhuizen, Willie C. Venter and Leenta, “The Study and Implementation of Shazam’s Audio Fingerprinting Algorithm for Advertisement Identification”, M.J. Grobler School of Electrical, Electronic and Computer Engineering North-West University, Potchefstroom Campus, South Africa.


