Content Based Video Retrieval

Shweta Ghodeswar¹, B.B. Meshram²
¹Department of Electrical Engineering, VJTI Mumbai,
²Department of Computer Technology, VJTI Mumbai,

Abstract
A system that supports video content-based indexing and retrieval has, in general, two stages: The first one is the database population stage, performs the following tasks: Video segmentation: Segment the video into constituent shots, Key frame selection: Select one frame or more to represent each shot and Feature extraction: Extract low-level and other features from key frames or their inter-relationships in order to represent these frames, hence shots. The second stage, the retrieval subsystem processes the presented query (usually in form of QBE), performs similarity matching operations, and finally displays results to the user. In this paper we talk about video segmentation, key frame selection and feature extraction.

Key Words: Feature extraction, Key frame extraction, Video representation, Video segmentation.

1. Introduction

Multimedia information systems are increasingly important with the advent of broadband networks, high-powered workstations, and compression standards. Since visual media requires large amounts of storage and processing, there is a need to efficiently index, store, and retrieve the visual information from multimedia database.

Video has become an important element of multimedia computing and communication environments, with applications as varied as broadcasting, education, publishing and military intelligence. However, video will only become an effective part of everyday computing environments when we can use it with the same facility that we currently use text.

The human derivation of textual descriptions from video contents has limitations; it is subjective, incomplete, biased, inaccurate and time consuming. Many of the principles, ideas, techniques and algorithms devised for content based image retrieval can be extended to video retrieval, but such extensions are not as simple as giving to each video frame the same treatment one would give to individual images. Indexing each individual frame as an image would produce a prohibitively high amount of redundant metadata, wasting valuable storage and processing resources. Moreover, because video is a structured medium in which actions and medium in time and space convey stories, a video program must be viewed as a document, not a non-structured sequence of frames.

The challenges behind the design and implementation of content based video indexing, browsing and retrieval systems have attracted many researchers from many disciplines, ranging from image processing, to databases to speech recognition, to mention just a few. Such a broad range of expertise has been recognized by the research community as necessity to tackle this complex problem. It is widely accepted that successful solutions to the problem of understanding and indexing video data will require a blend of information from different sources such as speech, sound, text and images.

As computers networks improve and digital image and video libraries become more readily available to homes and offices via the Internet, the problem of bandwidth and one’s ability to access relevant data become more challenging. Even now, the task of viewing and browsing through the vast amount of images and video data with conventional VCR-like interfaces is tedious, time-consuming, and unbearable. In order to make these databases widely usable, we need to develop a more effectively method for content selection, data retrieval and browsing.

Content-based Video Retrieval (CBVR) systems appear like a natural extension (or merge) of Content-based Image Retrieval (CBIR) systems. However, there are a number of factors that are ignored when dealing with images which should be dealt with when using videos. These factors are primarily related to the temporal information available from a video document.

Spatial Scene Analysis on video can be fully transferred from CBIR but temporal analysis is the uniqueness about video. Temporal Information induces the concept of motion for the objects present in the document.

The temporal analysis has following four different levels of granularity:
• **Frame level:** Each frame is treated separately. There is no temporal analysis at this level.

• **Shot-level:** A shot is a set of contiguous frames all acquired through a continuous camera recording. Only the temporal information is used.

• **Scene-level:** A scene is a set of contiguous shots having a common semantic significance.

• **Video-level:** The complete video object is treated as a whole.

The three types of Shot-level are as follows:

- **Cut:** A sharp boundary between shots. This generally implies a peak in the difference between color or motion histograms corresponding to the two frames surrounding the cut.

- **Dissolve:** The content of last images of the first shots is continuously mixed with that of the first images of the second shot.

- **Wipe:** The images of the second shot continuously cover or push out of the display that of the first shot.

The rest of the paper is organized as follows: Section 2 deals with video representation. Section 3 deals with video segmentation. Section 4 presents key frame selection. In Section 5 we talk about feature extraction. Finally we conclude with conclusions in Section 6.

### 2. Video Representation

Video sequences are rich in information, large in storage requirements, and have a time dimension. Thus it is extremely useful to have effective video representation and abstraction tools so that video content can be represented compactly.

One of the most effective methods of determining whether a video is relevant and for locating a relevant video segment is browsing. The traditional video operations (play, fast forward, and fast reverse) that are used for browsing are sequential and thus time consuming. Compact representation of video allows the user to see the main video contents quickly without going through the video sequentially.

A hierarchical video browser consists of a number of levels, from the video title, to groups of shots, to shots, and to individual frames. Representative frames are displayed at each level. Subsequent levels are displayed when selected. Using the hierarchical video browser, the user can find relevant shots or frames quickly. For example, in Figure 1, the user first selected video 3 based on the title and cover frame and a collection of video shot groups were displayed to the user. Based on the visual information displayed in each video shot group, the user found group 1 interesting and selected it. All the shots in group 1 were displayed with an r frame for each shot.

![Figure 1. A hierarchical video browser](image)

The user then selected shot 2 and all frames in shot 2 were displayed. Normally after very few interactions, the user is able to locate the video sections of interest.

### 3. Video Segmentation

Usually, a video is created by taking a set of shots and composing them together using specified composition operators. A shot is usually a piece of video taken with a fixed set of cameras, each of which has a constant relative velocity. In general, a shot may have many associated attributes such as the duration of the shot, the type(s) of camera(s) used, and so on.

A shot composition operator is an operation that takes two shots, $S_1$ and $S_2$, and a duration $t$ as input and merges the two shots into a composite shot within time $t$. Thus, for example, suppose we wish to compose together two shots $S_1$ and $S_2$, and suppose these two shots have duration $t_1$ and $t_2$ respectively. If $f$ is a shot composition operator, then

$$ f(S_1, S_2, t) $$

creates a segment of video of length $(t_1 + t_2 + t)$. $S_1$ is first shown and then undergoes a continuous transformation...
over time interval t, leading to the presentation of $S_2$ next. $f(S_1, S_2, t)$ then is a continuous sequence of video. In general, a video as a whole may be represented as

$$f_d(...f_d(f(S_1,S_2,t),S_3,t_2)...,S_{n+1},t_n)$$

Thus, a video is typically created by composing together a set of video shots using a finite set of video composition operations. Examples of video composition operations:

- **Shot concatenation**: It is the simplest form of video composition operator. It concatenates the two shots (even if the transition is not smooth). If shotcat is a shot concatenation operator, then t must be zero: that is, whenever we invoke shotcat $(S_1, S_2, t)$, the third argument t must be set to zero.

- **Spatial composition**: The best known type of spatial composition operation is a translate operation, which causes two successive shots to be overlaid one on top of the other. For instance, suppose we want to show shot $S_1$ first, followed by shot $S_2$. This is done by first overlaying shot $S_1$ on top of shot $S_2$ and then moving (i.e. translating) shot $S_1$ away, thus exposing shot $S_2$.

- **Chromatic composition**: There are two well-known kinds of chromatic compositions-fades and dissolves. Both these operations are chromatic scaling operations that try to continuously transform each pixel $(x, y)$ in the first shot into the corresponding pixel in the second shot. The smoothness of this transformation depends both upon the degree of difference between the two shots and the amount of time available for the composition operation.

Video segmentation, or shot change detection, involves identifying the frame(s) where a transition takes place from one shot to another. In cases where this change occurs between two frames, it is called a cut or a break.

Examples of transitions that occur gradually over several frames include fades, dissolves, wipes, and other special effect edits. These shots may be further classified according to camera motion.

Video segmentation is the technique which can generate object shape information from video sequences. It is a key operation for MPEG-4 since without video segmentation the content based coding functionalities cannot be realized. It is also a very important operation for feature extraction of video sequences, which can be used for MPEG-7 to represent and index video data or for other intelligent signal processing applications, such as recognition.

Video segmentation can be used for many real-time applications for MPEG-4 such as video phone, video conference and content based video camcorder.

### 4. Key frame Selection

Typically in retrieval applications, a video sequence is subdivided in time into a set of shorter segments each of which contains similar content. These segments are represented by 2-D representative images called "key-frames" that greatly reduce amount of data that is searched. However, key frames do not describe the motions and actions of objects within the segment.

Choosing key frames of scenes allows us to capture most of the content variations, due at least to camera motion, while at the same time excluding other key frames which may be redundant.

The ideal method of selecting key frames would be to compare each frame to every other frame in the scene and select the frame with the least difference from other frames in terms of a given similarity measure. Obviously, this requires extensive computation and is not practical for most applications. On the other hand, choosing the first frame seems to be the natural choice, as all the rest of the frames in the scene can be considered to be logical and continuous extensions of the first frame, but it may not be the best match for all the frames in the scene. In a more general framework, we would like to choose frames with the greatest content or index potential --- for example, frames with text, or frames with clear unoccluded objects. Other factors that influence the choice include encoding patterns. For example, the frequency of I frames may affect the choice, as I frames represent the best candidates for key frame selection since they have the actual DCT coefficients which form the spatial component of the data. If I frames occur fairly frequently, then the first I frame can be chosen as the key frame. It is, however, possible to have an entire scene with no I frame, in which case, alternate measures are required.

We have found it is, in general, sufficient to select the first frame of each scene as a key frame. This is based in the observation that cinematographers attempt to "characterize" a shot with the first few frames, before beginning to track or zoom to a close-up.
5. Feature Extraction

Once objects of interest have been segmented from the input video sequences, feature extraction is used to identify abnormal activities such as excessive human movement that may, for instance, indicate that fighting is taking place. We focus on two features: excessive movement detection and head top points detection. Excessive movement $EM$ is detected by comparing moving regions and areas occupied by segmented objects. Suppose $m_{i-1}^T$ is the mask of an object of interest and $m_{i}^T$ is the mask of moving regions between two consecutive frames $F_{i-1}$ and $F_{i}$. Then $EM$ for those frames is given by

$$EM_i = m_{i}^T / m_{i-1}^T$$

$EM$ measures the ratio of positional change between the two frames. A value of $EM$ that is close to zero indicates that light or no movement is detected. On the other hand, $EM$ can slightly exceed unity when there is much movement between two consecutive frames. It is found empirically that a threshold of $(T_{EM})$ of 0.5 is suitable for most situations. Thus excessive movement is detected between $F_{i-1}$ and $F_{i}$ if the value of $EM_i > 0.5$.

Head top points are used as feature points for tracking the movement of objects. Head top points can provide useful information about body motion because body movements often lead to changes in head position. A head top point is defined as the highest peak of a segmented object of interest. For example, a head top point can indicate whether an object is adult-height or child-height. We can also track head top points to determine if a person is standing up or lying on the floor. In a event that a previously standing object has fallen on the floor, we can also determine how long the object remains relatively stationary (with a small value of $EM$) using a timer.

Feature extraction technique combines excessive movement detection and head top points analysis to detect a variety of situations. For example, if two children are left unattended in a room, then a number of possibilities can happen. If excessive movement is detected and the head top points correspond to two children who are close to each other then the children may be playing or fighting. If subsequently, one of the children falls on the floor as detected by head top points analysis and little motion is detected then this may be a cause for concern as the child may be injured. On the other hand, if wrestling is detected as excessive movement at the same time when a third, adult height object is unexpectedly detected, then an intruder may be present.

6. Conclusion

Every video in the database is segmented into several shots. Second, for each shot, one or more key frames are selected, and then a feature vector for each key frame is computed. Since every database video contains a sequence of key frames, there exists a sequence of feature vectors for the database video. The sequences of feature vectors of database videos are stored in the feature database. Third, the given query video is also segmented into several shots. For each shot, one or more key frames are selected. A feature vector is computed for each selected key frame. So, there also exists a sequence of feature vectors for the query video. Then, we use a dynamic programming approach to compute the similarity between the sequence of feature vectors of the query video and each sequence of feature vectors in the feature database. The database videos with the similarity higher than a predefined threshold are output and returned to the user.

Content-based video indexing and retrieval is an active area of research with continuing attributions from several domain including image processing, computer vision, database system and artificial intelligence.

7. References