Video Content-Based Retrieval Techniques

Waleed E. Farag
Zagazig University, Egypt

INTRODUCTION

Recently, multimedia applications are undergoing explosive growth due to the monotonic increase in the available processing power and bandwidth. This incurs the generation of large amounts of media data that need to be effectively and efficiently organized and stored. While these applications generate and use vast amounts of multimedia data, the technologies for organizing and searching them are still in their infancy. These data are usually stored in multimedia archives utilizing search engines to enable users to retrieve the required information.

Searching a repository of data is a well-known important task whose effectiveness determines, in general, the success or failure in obtaining the required information. A valuable experience that has been gained by the explosion of the Web is that the usefulness of vast repositories of digital information is limited by the effectiveness of the access methods. In a nutshell, the above statement emphasizes the great importance of providing effective search techniques. For alphanumeric databases, many portals (Baldwin, 2000) such as *google*, *yahoo*, *msn*, and *excite* have become widely accessible via the Web. These search engines provide their users keyword-based search models in order to access the stored information, but the inaccurate search results of these search engines is a known drawback.

For multimedia data, describing unstructured information (such as video) using textual terms is not an effective solution because the information cannot be uniquely described by a number of statements. That is mainly due to the fact that human opinions vary from one person to another (Ahanger & Little, 1996), so that two persons may describe a single image with totally different statements. Therefore, the highly unstructured nature of multimedia data renders keyword-based search techniques inadequate. Video streams are considered the most complex form of multimedia data because they contain almost all other forms, such as images and audio, in addition to their inherent temporal dimension.

One promising solution that enables searching multimedia data, in general, and video data in particular is the concept of content-based search and retrieval (Deb, 2004). The basic idea is to access video data by their contents, for example, using one of the visual content features. Realizing the importance of content-based searching, researchers have started investigating the issue and proposing creative solutions. Most of the proposed video indexing and retrieval prototypes have the following two major phases (Flinkner et al., 1995):

- **Database population phase** consisting of the following steps:
- **Shot boundary detection.** The purpose of this step is to partition a video stream into a set of meaningful and manageable segments (Idris & Panchanathan, 1997), which then serve as the basic units for indexing.
- **Key frames selection.** This step attempts to summarize the information in each shot by selecting representative frames that capture the salient characteristics of that shot.
- **Extracting low-level features from key frames.** During this step, some of the low-level spatial features (color, texture, etc.) are extracted in order to be used as indices to key frames and hence to shots. Temporal features (e.g., object motion) are used too.
- **The retrieval phase.** In this stage, a query is presented to the system that in turns performs similarity matching operations and returns similar data (if found) back to the user.

In this article, each of the these stages will be reviewed and expounded. Moreover, background, current research directions, and outstanding problems will be discussed.

VIDEO SHOT BOUNDARY DETECTION

The first step in indexing video databases (to facilitate efficient access) is to analyze the stored video streams. Video analysis can be classified into two stages: shot boundary detection and key frames extraction (Rui, Huang & Mchrotra, 1998a). The purpose of the first stage is to partition a video stream into a set of meaningful and manageable segments, whereas the second stage aims to abstract each shot using one or more representative frames.

In general, successive frames (still pictures) in motion pictures bear great similarity among themselves, but this generalization is not true at boundaries of shots. A shot is a series of frames taken by using one camera. A frame at a boundary point of a shot differs in background and content from its successive frame that belongs to the next shot. In a nutshell, two frames at a boundary point will differ significantly as a result of switching from one camera to another, and this is the basic principle that most automatic algorithms for detecting scene changes depend upon.

Due to the huge amount of data contained in video streams, almost all of them are transmitted and stored in compressed format. While there are large numbers of algorithms for compressing digital video, the MPEG format (Mitchell, Pennebaker, Fogg & LeGall, 1997) is the most famous one and the current international standard. In MPEG, spatial compression is achieved through the use of a DCT-based (Discrete Cosine Transform-based) algorithm similar to the one used in the JPEG standard. In this algorithm, each frame is divided into a number of blocks (8x8 pixel), then the DCT transformation is applied to these blocks. The produced coefficients are then quantized and entropy encoded, a technique that achieves the actual compression of the data. On the other side, temporal compression is accomplished using a motion compensation technique that depends on the similarity between successive frames on video streams. Basically, this technique codes the first picture of a video stream (I frame) without reference to neighboring frames, while successive pictures (P or B frames) are generally coded as differences to that reference frame(s). Considering the large amount of processing power required in the manipulation of raw digital video, it becomes a real advantage to work directly upon compressed data and avoid the need to decompress video streams before manipulating them.

A number of research techniques was proposed to perform the shot segmentation task such as template matching, histogram comparison, block-based comparison, statistical models, knowledge-based approach, the use of AC coefficients, the use of motion vectors, and the use of supervised learning systems (Farag & Abdel-Wahab, 2001a, 2001c).

KEY FRAMES SELECTION

The second stage in most video analysis systems is the process of KFs (Key Frames) selection (Rui, Huang & Mchrotra, 1998) that aims to abstract the whole shot using one frame or more. Ideally, we need to select the minimal set of KFs that can faithfully represent each shot. KFs are the most important frames in a shot since they may be used to represent the shot in the browsing system, as well as be used as access points. Moreover, one advantage of representing each shot by a set of frames is the reduction in the computation burden required by any content analysis system to perform similarity matching on a frame-by-frame basis, as will be discussed later. KFs selection is one of the active areas of research in visual information retrieval, and a quick review of some proposed approaches follows.

Clustering algorithms are proposed to divide a shot into M clusters, then choose the frame that is closest to the cluster centroid as a KF. An illumination invariant approach is proposed that applies the color constancy feature to KFs production using hierarchical clustering. The VCR system (Farag & Abdel-Wahab, 2001b, 2001c) uses two algorithms to select KFs (AFS and ALD). The AFS is a dynamic adapted algorithm that uses two levels of threshold adaptation—one based on the input dimension, and the second relying upon a shot activity criterion to further improve the performance and reliability of the selection. AFS employs the accumulated frame summation of luminance differences of DC frames. The second algorithm, ALD, uses absolute luminance difference and employs a statistical criterion for the shot-by-shot adaptation level, the second one.

FEATURE EXTRACTION

To facilitate access to large video databases, the stored data need to be organized; a straightforward way to do such organization is the use of index structures. In case of video databases we even need multi-dimension index structures to account for the multiple features used in indexing. Moreover, we are in need of tools to automatically or semi-automatically extract these indices for proper annotation of video content. Bearing in mind that each type of video has its own characteristics, we also need to use multiple descriptive criteria in order to capture all of these characteristics.

The task of the feature extraction stage is to derive descriptive indexes from selected key frames in order to represent them, then use the indexes as metadata. Any further similarity matching operations will be performed over these indexes and not over the original key frames data. Ideally, content-based retrieval (CBR) of video should be accomplished based on automatic extraction of content semantics that is very difficult. Thus, most of the current techniques only check the presence of semantic primitives or calculate low-level visual features. There are mainly two major trends in the research community to extract indices for proper video indexing and annotation. The first one tries to automatically extract these indices,