Video Content-Based Retrieval

Waleed E. Farag
Indiana University of Pennsylvania, USA

INTRODUCTION

Recently, multimedia applications have undergone 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 immature. 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 (Acuna, Marcos, Gomez, & Bussler, 2005) have become widely accessible via the Web. These portals use search engines that adopt keyword-based search models in order to access the stored information, but the inaccurate search results of these search engines is a known issue.

For multimedia data, describing unstructured information (such as video) using textual terms is not an effective solution because they 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 (Tešic & Smith, 2006), so that two persons may describe a single image by 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, 2005). 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 (Marques & Furht, 2002):

1. The **database population phase** consists 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 (Hanjalic, 2002), 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 indexes to key frames and hence to shots. Temporal and other features (e.g., object motion) are used also.
2. In the **retrieval phase,** a query is presented to the system that in turns performs similarity matching operations and returns similar data (if found) back to the user.

It is worth mentioning that a growing trend in current content-based retrieval systems is the application of contextual constraints to enrich those systems with additional metadata (Davis, King, Good, & Sarvas, 2004). The use of context makes video retrieval systems both content-based and context-based systems at the same time. Besides, context-based techniques try to improve the retrieval performance by using associate contextual information, other than those derived from the media content (Hori & Aizawa, 2003).

In this article, each of the above stages will be reviewed and expounded. Background, current research directions, and outstanding problems will also 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 (Farag & Abdel-Wahab, 2002b), *shot boundary detection* and *key frames extraction.* 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 (except in the case of gradual transitions). 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 many algorithms for compressing and representing digital video data, the MPEG family (Watkinson, 2004) is the most famous one and the current international standard. In MPEG, spatial compression is achieved through the use of a Discrete Cosine Transform (DCT)-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 those reference frames. 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.

Several research techniques were proposed to perform the shot detection task for both cuts and gradual transitions. For instance, template matching and histogram comparison are commonly used. Statistical models are also proposed as in Hanjalic (2002). Farag and Abdel-Wahab (2002b) proposed the use of supervised learning systems in order to detect shout boundaries. Moreover, other techniques such as finite state machine and support vector machines are used to identify various types of transitions (Liu, Gibbon, Zavesky, Shahraray, & Haffner, 2007).

KEY FRAMES SELECTION

The second stage in most video analysis systems is the process of *key frames* (KFs) selection (Deb, 2005), which aims to abstract every 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, hence 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 an active area 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. Cooper and Foote (2005) introduced the use of linear discriminant analysis to select representative frames. The VCR system (Farag & Abdel-Wahab, 2002a) employs two algorithms to select KFs, *accumulated frames summation* (AFS) and *absolute luminance differences* (ALD). The AFS is a dynamically adapted algorithm that uses two levels of threshold adaptation, one based on the input dimension while the second level relies on a shot activity criterion to further improve the performance and reliability of the selection. AFS employs the accumulated summation of luminance differences of corresponding *direct current* (DC) coefficients in successive frames. The second algorithm, ALD, uses absolute luminance difference between the summation of all DC terms in a frame and the same summation of the next frame. It utilizes a statistical criterion for the shot-by-shot adaptation level. A comprehensive review and classifications of video abstraction techniques introduced by various researchers in the field is presented in Truong and Venkatesh (2007). That work reviewed different methodologies that use still images (key frames) and moving pictures (video skims) to abstract video data and provide fast overviews of the video content.

FEATURE EXTRACTION

To facilitate access to large video databases, the stored data need to be organized; a straightforward way to do such organization is through 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 indexes 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 video data content such as from selected key frames in order to represent the original data. These indexes are then used as metadata, and any further similarity