9.1

Image and Video Indexing and Retrieval

Michael A. Smith and Tsuhan Chen
Carnegie Mellon University

1 Introduction

The amount of digital content, in the form of images and video, has been increasing exponentially in recent years. With increasing computing power and electronic storage capacity, the potential for large digital image/video libraries is growing rapidly. In particular, the World Wide Web has seen an increased use of digital images and video, which form the base of many entertainment, educational, and commercial applications. As a result, it has become more and more challenging for a user to search for the relevant information among a large amount of digital images or video. Image and video libraries therefore need to provide easy informational access and the retrieval information must be easy to locate, manage and display.

As the size of accessible image and video collections grows to thousands of hours, potential viewers will need abstractions and technology that help them browse effectively and efficiently. Text-based search algorithms offer some assistance in finding specific images or segments among large video collections. In most cases, however, these systems output many irrelevant images or video segments to ensure retrieval of pertinent information. Intelligent indexing systems are essential for optimal retrieval of image and video data.

A typical content-based image/video retrieval system includes three major aspects: feature extraction, high dimensional indexing and system design [24]. Among the three aspects, high dimensional indexing is important for speed performance; system design is critical for appearance performance; and feature extraction is the key to accuracy performance. The accuracy performance of a retrieval system is very subjective and user-dependent. To a user, the similarity between objects is often high-level or semantic. However, features we can extract from objects are often low-level features, as most of them are extracted directly from digital representations of objects in the database. The gap between low-level features and high-level semantics has been the major obstacle to better retrieval performance.

In this chapter we explore the latest technologies in image and video retrieval. We describe several methods for extracting features that are used to measure image and video similarity in multimedia databases. We also describe techniques to bridge the gap between low-level features and high-level semantics.

1.1 Content Categories

Most commercial and academic research in image and video retrieval has focused on documentaries and broadcast news, although some experimentation has been devoted to sports, feature films, and stock footage. Documentary footage is very informative and follows many standard video production procedures. Access to public broadcast material is less stringent than private material since much of what they produce is educational, rather than commercial. There are three basic formats for documentary video: 1) factual, 2) historic, and 3) biographic. Broadcast news includes pre-recorded and live footage from producers such as CNN Headline News, ABC, NBC, and CBS. Most live news segments present a variety of challenging editing styles and effects that are not commonly used in pre-recorded segments. There are many different types of feature films. Most experiments test commercially successful films suitable for all audiences. For sports, most researchers examine video with recognizable formats and "plays" through the course of the event, such as football, basketball, baseball, soccer, and hockey.

1.2 Storage and Compression

In analysis of digital content, compression schemes offer increased storage capacity by utilizing statistical characteristics of images and video. Images and video are compressed and stored as discrete cosine transform (DCT) coefficients and motion vectors. One drawback to these compression schemes is loss in quality. Bitstreams created by lossy compression schemes, however, typically preserve some statistical information of the original video in an explicit manner. For example, the DCT coefficients preserve colors, texture, and other spatial domain characteristics, and motion vectors preserve object motion, camera pan and zoom, and other temporal characteristics. Lossless schemes, such as run length encoding (RLE) and Huffman coding, do not sacrifice quality but provide lower compression ratios. Furthermore, bitstreams created by lossless algorithms do not explicitly contain any statistical information of the original video. Many algorithms provide compression as high as 100 to 1, and often use DCT and motion compensation for compression. The parameters of the DCT may be used for video segmentation while the motion compensation statistics may be used as a form of optical flow, as discussed in Section 2.

2 Image and Video Features

A feature is defined as a descriptive parameter that is extracted from an image or video stream. Features may be used to interpret visual content, or as a measure for similarity in image and video databases. In this chapter, features are described in the following categories:

- Statistical Features—Features extracted from and image or video sequence without regard to content are described as statistical features. These include parameters derived from such algorithms as image difference and camera motion.
- Compressed Domain Features—A feature extracted from a compressed image or video stream without regard to content is described as a compressed domain feature.
- Content-Based Features—A feature derived for the purpose of describing the actual content in an image or video stream is a content-based feature.

In the sections that follow, we describe examples of each feature and potential applications in image and video databases.

2.1 Statistical Features

Certain features may be extracted directly from image pixels without regard to the content. These features include such analytical features as scene changes, motion flow and video structure in the image domain, and sound discrimination in the audio domain. In this section we describe techniques for image difference and motion analysis as statistical features. For more information on statistical image features, please see Chapters 4.7 and 4.8. Here we discuss features that are more related to video.

2.1.1 Image Difference

A difference measure between images serves as a feature to measure similarity. We describe two fundamental methods for image difference: absolute difference and histogram difference. The absolute difference requires less computation, but is generally more susceptible to noise and other imaging artifacts, as described below.

Absolute Difference. The image difference of two images is defined as the sum of the absolute difference at each pixel. The first image I_t is analyzed with a second image, I_{t-T}, at a temporal distance T. The difference value is defined as,

$$D(t) = \sum_{i=1}^{M} \left| I_{(t-T)}(i) - I_t(i) \right|$$

where M is the resolution, or number of pixels in the image. This method for image difference is noisy and extremely sensitive to camera motion and image degradation. When applied to sub-regions of the image, $D(t)$ is less noisy

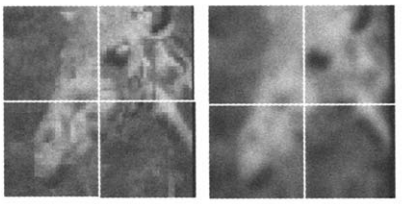

FIGURE 1 Left: original; Right: filtered (see color insert).

and may be used as a more reliable parameter for image difference.

$$D_s(t) = \sum_{j=S}^{H/n} \sum_{i=S}^{W/n} \left| I_{(t-T)}(i,j) - I_t(i,j) \right|$$

$D_s(t)$ is the sum of the absolute difference in a sub-region of the image, where S represents the starting position for a particular region and n, represents the number of sub-regions.

We may also apply some form of filtering to eliminate excess noise in the image and subsequent difference. For example, the image on the right in Fig. 1 represents the output of a Gaussian filter on the original image on the left. For more details on image enhancement, please refer to Chapter 3.

Histogram Difference. A histogram difference is less sensitive to subtle motion, and is an effective measure for detecting similarity in images. By detecting significant changes in the weighted color histogram of two images, we form a more robust measure for image correspondence. The histogram difference may also be used in sub-regions to limit distortion due to noise and motion.

$$D_H(t) = \sum_{v=1}^{N} \left| H_{(t-1)}(v) - H_t(v) \right|$$

The difference value, $D_H(t)$, will rise during scene changes, image noise, and camera or object motion. In the equation below, N represents the number of bins in the histogram, typically 256. Two adjacent images may be processed, although this algorithm is less sensitive to error when images separated by a spacing interval, D_i. D_i is typically on the order of 5 to 10 frames for video encoded at standard 30 fps. An empirical threshold may be set to detect values of $D_H(t)$ that correspond to scene changes. For inputs from multiple categories of video, an adaptive threshold for $D_H(t)$ should be used.

$$D_{H-R}(t) = \sum_{v=1}^{N} \left| H_{R(t-1)}(v) - H_{Rt}(v) \right|$$

$$D_{H-G}(t) = \sum_{v=1}^{N} \left| H_{G(t-1)}(v) - H_{Gt}(v) \right|$$

$$D_{H-B}(t) = \sum_{v=1}^{N} \left| H_{B(t-1)}(v) - H_{Bt}(v) \right|$$

$$D_{H-RGB}(t) = \frac{\sum \left(D_{H-R}(t) + D_{H-G}(t) + D_{H-B}(t) \right)}{3}$$

If the histogram is actually three separate sets for RGB, the difference may simply be summed. An alternative to summing the separate histograms is to convert the RGB histograms to a single color band, such as Munsell or LUV color. Color representations in digital imagery are discussed in Chapter 4.5.

2.1.2 Video Segmentation

An important application of image difference in video is the separation of visual scenes. A simple image difference represents one of the more common methods for detection of scene changes. The difference measures, $D(t)$ and $D_H(t)$, may be used to determine the occurrence of a scene change. By monitoring the difference of two images over some time interval, a threshold may be set to detect significant differences or changes in scenery. This method provides a useful tool for detecting scene cuts, but is susceptible to errors during transitions. A block-based approach may be used to reduce errors in difference calculations. This method is still subject to errors when subtle object or camera motion occurs.

The most fundamental scene change is the video cut. For most cuts, the difference between image frames is so distinct that accurate detection is not difficult. Cuts between similar scenes, however, may be missed when using only static properties such as image difference. Several research groups have developed working techniques for detecting scene changes through variations in image and histogram differencing. For more details on video segmentation technology, please see Chapter 4.9.

A histogram difference is less sensitive to subtle motion, and is an effective measure for detecting scene cuts and gradual transitions. By detecting significant changes in the weighted color histogram of each successive frame, video sequences can be separated into scenes. This technique is simple, and yet robust enough to maintain high levels of accuracy. An illustration of histogram based segmentation is shown in Fig. 2.

Scene Change Categories. There are a variety of complex scene changes used in video production, but the basic premise is a change in visual content. The video cut, as well as other scene change procedures are discussed below.

Fast Cut—A sequence of video cuts, each very short in duration, represents a fast cut. This technique heightens the sense of action or excitement. To detect a fast cut, we may

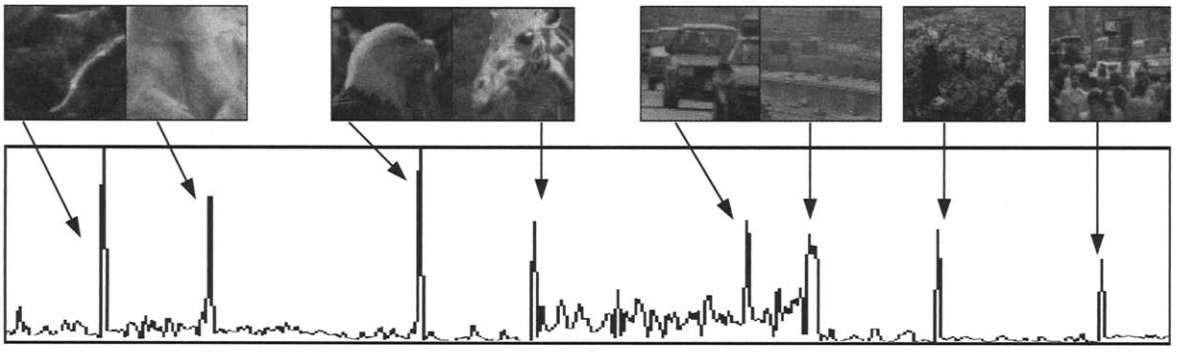

FIGURE 2 Histogram difference, $D_{H\text{-}RGB}(t)$, for scene segmentation.

look for a sequence of scene changes that are in close proximity.

Distance Cut—A distance cut occurs when the camera cuts from one perspective of a scene to another some distance away. This shift in distance usually appears as a cut from a wide shot to a close-up shot, or vice-versa.

Inter-cutting—When scenes change back and forth from one subject to another, we say the subjects are inter-cut. This concept is similar to the distance cut, but the images are separate and not inclusive of the same scenes. Inter-cutting is used to show a thought process between two or more subjects.

Dissolves and Fades—Dynamic imaging effects are often used to change from one scene to another. A common effect in all types of video is the fade. A fade occurs when a scene changes over time from its original color scheme to a black background. This procedure is commonly used as a transition from one topic to another. Another dynamic effect is the dissolve. Similar to the fade, this effect occurs when a scene changes over time and morphs into a separate scene. This transition is less intrusive and is used when subtle change is needed.

Wipes and Blends—These effects are most often used in news video. The actual format of each may change from one show to the next. A wipe usually consists of the last frame of a scene being folded like a page in a book. A blend may be shown as pieces of two separate scenes combined in some artistic manner. Like the fade and dissolve, wipes and blends are usually used to transition to a separate topic. In feature-films a wipe is often used to convey a change in time or location.

Alternative Segmentation Technology. An alternative form of scene segmentation involves the use of traditional edge detection characteristics. Edges in images are useful information about the changes in background and object distribution between scenes. An effective algorithm for detecting cuts and gradual transitions was developed at Cornell University using edge detection technology [22]. For more details on edge detection technology, please see Chapters 4.10 and 4.11.

An analysis of the global motion of a video sequence may also be used to detect changes in scenery. For example, when the error in optical flow is high, this is usually attributed to its inability to track a majority of the motion vectors from one frame to the next. Such errors can be used to identify scene changes. A motion-controlled temporal filter may also be used to detect dissolves and fades, as well as separate video sequences that contain long pans. The use of motion as a statistical feature is discussed in the following section. The methods for scene segmentation described in this section may be used individually or combined for more robust segmentation.

2.1.3 Motion Analysis

Motion characteristics represent an important feature in video indexing. One aspect is based on interpreting camera motion [1, 17]. Many video scenes have dynamic camera effects, but offer little in the description of a particular segment. Static scenes, such as interviews and still poses, contain essentially identical video frames. Knowing the precise location of camera motion can also provide a method for video parsing. Rather than simply parse a video by scenes, one may also parse a video according to the type of motion. An important kind of video characterization is defined not just by the motion of the camera, but also by motion or action of the objects being viewed.

An analysis of optical flow can be used to detect camera and object motion. Most algorithms for computing optical flow require extensive computation, and more often, researchers are exploring methods to extract optical flow from video compressed with some form of motion compensation. Section 3 describes the benefits of using compressed video for optical flow and other image features.

Statistics from optical flow may also be used to detect scene changes. Optical flow is computed from one frame to the next. When the motion vectors for a frame are randomly

distributed without coherency, this may suggest the presence of a scene change. In this sense, the quality of the camera motion estimate is used to segment video. Video segmentation algorithms often yield false scene changes in the presence of extreme camera or object motion. An analysis of optical flow quality may also be used to avoid false detection of scene changes.

Optical flow fields may be interpreted in many ways to estimate the characteristics of motion in video. Two such interpretations are the camera motion and object motion. For more details on motion analysis, please see Chapter 3.8.

Camera Motion. An affine model is used to approximate the flow patterns consistent with all types of camera motion.

$$u(x_i, y_i) = ax_i + by_i + c$$
$$v(x_i, y_i) = dx_i + ey_i + f$$

Affine parameters a, b, c, d, e, and f are calculated by minimizing the least squares error of the motion vectors.

$$\begin{bmatrix} \sum x^2 & \sum xy & \sum x & 0 & 0 & 0 \\ \sum xy & \sum x & \sum y & 0 & 0 & 0 \\ \sum x & \sum y & \sum N & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum x^2 & \sum xy & \sum x \\ 0 & 0 & 0 & \sum xy & \sum x^2 & \sum y \\ 0 & 0 & 0 & \sum x & \sum y & \sum N \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \sum ux \\ \sum uy \\ \sum u \\ \sum vx \\ \sum vy \\ \sum v \end{bmatrix}$$

We also compute average flow $\bar{v}$ and $\bar{u}$. Where $\bar{v}$ and $\bar{u}$,

$$\bar{u} = \sum_{i-0}^{N} ax_i + by_i + c$$
$$\bar{v} = \sum_{i-0}^{N} dx_i + ey_i + f$$

Using the affine flow parameters and average flow, we classify the flow pattern. To determine if a pattern is a zoom, we first check if there is the convergence or divergence point (x_0, y_0), where: $u(x_i, y_i) = 0$ and $v(x_i, y_i) = 0$. To solve for (x_0, y_0), the following relation must be true:

$$\begin{vmatrix} a & b \\ d & e \end{vmatrix} = 0$$

If the above relation is true, and (x_0, y_0) is located inside the image, then it must represent the focus of expansion. If $\bar{v}$ and $\bar{u}$, are large, then this is the focus of the flow and camera is zooming. If (x_0, y_0) is outside the image, and or are large, then the camera is panning in the direction of the dominant vector.

If the above determinant is approximately 0, then (x_0, y_0) does not exist and the camera is panning or static. If $\bar{v}$ or $\bar{u}$, are large, the motion is panning in the direction of the dominant vector. Otherwise, there is no significant motion and the flow is static. We may eliminate fragmented motion by averaging the results in a W frame window over time. Examples of the camera motion analysis results are shown in Fig. 3.

Object Motion. Object motion typically exhibits flow fields in specific regions of an image, while camera motion is characterized by flow throughout the entire image. The global distribution of motion vectors distinguishes between object and camera motion. The flow field is partitioned into a grid as shown in Fig. 4. If the average velocity for the vectors in a particular grid is high (typically > 2.5 pixels), then that grid is designated as containing motion. When the number of connected motion grids, G_m,

$$G_m(i) = \begin{cases} 0 & (G_m(i-1) = 0, G_m(i+1) = 0, \ldots, M) \\ 1 & \text{otherwise} \end{cases}$$

is high (typically $G_m > 7$), the flow is some form of camera motion. $G_m(i)$ represents the status of motion grid at position i and M represents the number of neighbors. A motion grid should consist of at least a 4×4 array of motion vectors. If G_m is not high, but greater than some small value (typically 2 grids), the motion is isolated in a small region of the image and the flow is probably caused by object motion. This result is averaged over a frame window of width W_A, just as with camera motion, but the number of object motion regions needed is typically on the order of 60%. This is 12 object motion frames for a typical W_A of 20 frames. Examples of the object motion analysis results are shown in Fig. 4.

2.1.4 Alternative Statistical Features

Texture: Analysis of image texture is useful in the discrimination of low interest video from video containing complex features. A low interest image may also contain uniform texture, as well as, uniform color or low contrast. Perceptual features for individual video frames were computed using common textual features such as, coarseness, contrast, directionality, and regularity. For more details on texture analysis, please see Chapters 4.7 and 4.9.

Shape and Position: The shape and appearance of objects may also be used as a feature for image correspondence. Color and texture properties will often change from one image to the next, making image difference and texture features less useful. An example of this is shown in Fig. 5, where the feature of interest is an anchorperson, but the color, texture and position of the subjects is different for each image.

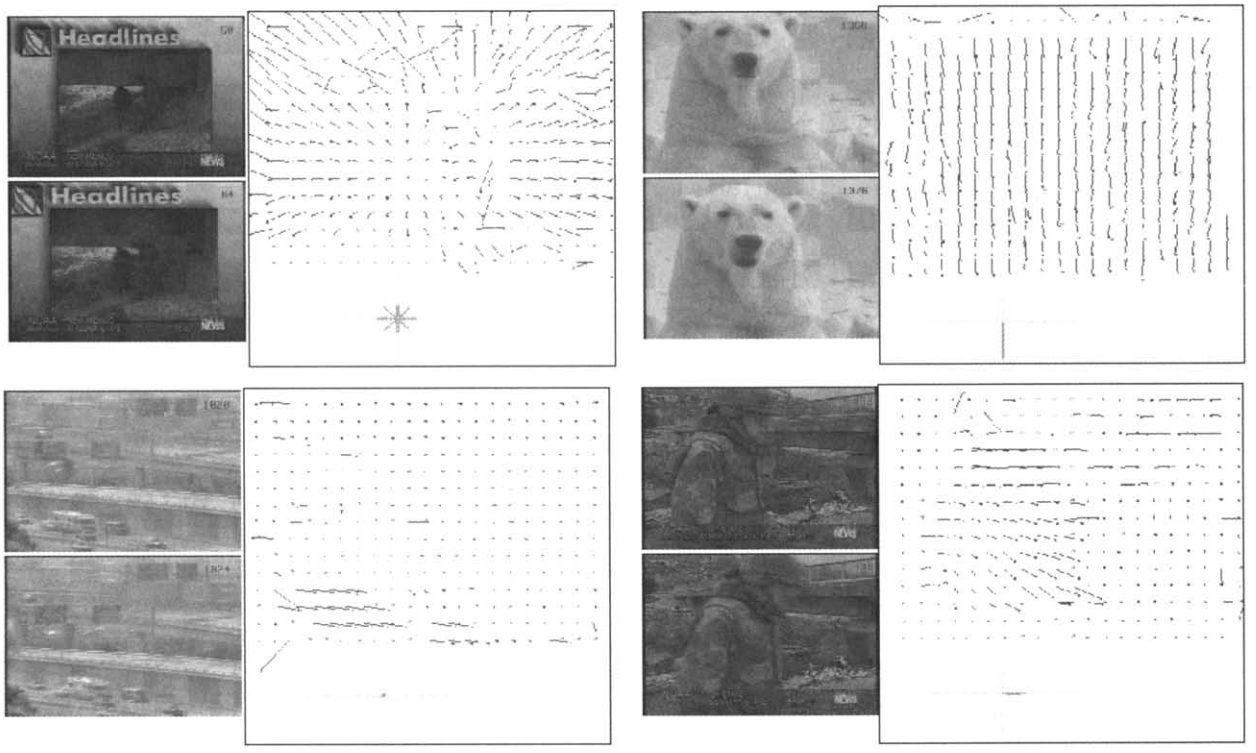

FIGURE 3 Optical flow fields for a pan (top right), zoom (top left), and object motion (see color insert).

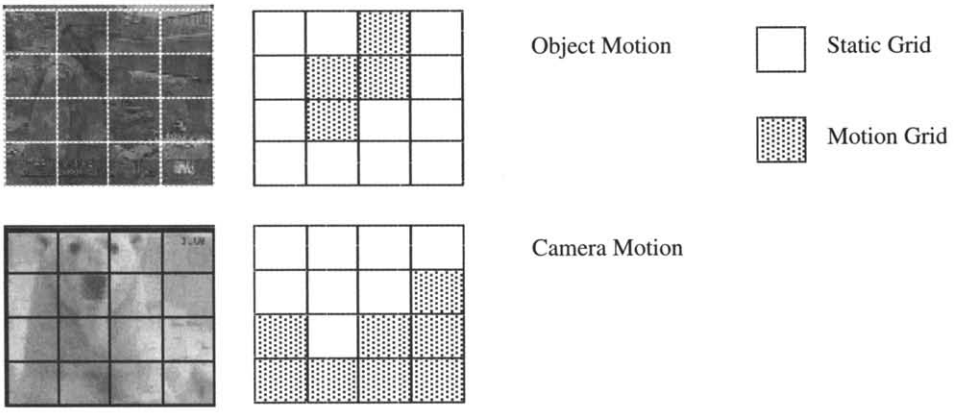

FIGURE 4 Camera and object motion detection (see color insert).

FIGURE 5 Images with similar shapes (human face and torso) (see color insert).

Commercial systems for shape based image correspondence are discussed in section 7. For more details on shape analysis, please see Chapter 4.6.

Audio Features: In addition to image features, certain audio features may be extracted from video to assist in the retrieval task. Loud sounds, silence, and single frequency sound markers may be detected analytically without actual knowledge of the audio content. Loud sounds imply a heightened state of emotion in video, and are easily detected by measuring a number of audio attributes, such as signal amplitude or power. Silent video may signify an area of less importance, and can also be detected with straightforward analytical estimates. A video producer will often use single frequency sound markers, typically a 1000 Hz tone, to mark a particular point in the beginning of a video. This tone may be detected to determine the exact point in which a video will start.

2.1.5 Hierarchic Video Structure

Most video is produced with a particular format and structure. This structure may be taken into consideration when analyzing particular video content. News segments are typically 30 minutes in duration and follow a rigid pattern from day to day. Commercials are also of fixed duration, making detection less difficult.

Another key element in video is the use of the black frame. In most broadcast video, a black frame is shown between a transition of two segments. In news broadcast this usually occurs between a story and a commercial. By detecting the location of black frames in video, a hierarchic structure may be created to determine transitions between segments. A black frame or any single intensity image may be detected by summing the total number of pixels in a particular color space, P_s.

$$P_s(t) = \sum_{i=1}^{M} \begin{cases} 0 & (i > I_{\text{high}} || i < I_{\text{low}}) \\ 1 & \text{otherwise} \end{cases}$$

In the detection of the black frame, I_{high}, the maximum allowable pixel intensity is on the order of 20% of the maximum color resolution (51 for a 256 bit image), and I_{low}, the minimum allowable pixel intensity, is 0. The separation of segments in video is crucial in retrieval systems, where a user will most likely request a small segment of interest and not an entire full-length video. There are a number of ways to detect this feature in video, the simplest being to detect a high number of pixels in an image that are within a given tolerance of being a black pixel.

2.2 Compressed Domain Features

In typical applications of multimedia databases, the materials, especially the images and video, are often in a compressed format. Given large amounts of compressed materials (e.g., MPEG), how do we index and retrieve the content rapidly? To deal with these materials, a straightforward approach is to decompress all the data, and utilize the same features as mentioned in previous section. Doing so, however, has some disadvantages. First, the decompression implies extra computation. Secondly, the process of decompression and re-compression, often referred to as "recoding", results in further loss of image quality. Finally, since the size of decompressed data is much larger than the compressed form, most hardware and CPU cycles are needed to process and store the data. The solution to these problems is to extract features directly from the compressed data. We call these the compressed-domain features, and these features can be useful for indexing and retrieval [1, 7, 19]. The question is how to explore unique information available in the compressed domain. We start by introducing a number of commonly used compressed-domain features:

The motion vectors that are available in all video data compressed using standards such H.261/H.263 and MPEG-1/2 are very useful. Analysis of motion vectors can be used to detect scene changes and other special effects such as dissolve, fade in and fade out. For example, if the motion vectors for a frame are randomly distributed without coherency, that may suggest the presence of a scene change. Segmentation of a field of motion vectors into regions of similar vectors can be used to detect moving objects and track their positions. They can also be used to derive camera motion such as zoom and pan [1, 17]. Essentially, since motion vectors represent a low-resolution optical flow in the video, they can be used to extract all information that can be extracted using the optical flow method.

The percentage of each type of block in a picture is also a good indicator of scene changes, too. For a P frame, a large percentage of intra blocks implies a lot of new information for the current frame that cannot be predicted from the previous frame. Therefore, such a P-frame indicates the beginning of a new scene right after a scene change. For a B frame, the ratio between the number of forward predicted locks and the number of backward predicted blocks can be used to conclude whether the scene change happens before this B-frame or after this B-frame. If the number of forward predicted blocks is larger than the number of backward predicted blocks, i.e., there is more correlation between the previous frame and the current B frame than there is between the current B frame and the following frames, then one can conclude that the scene change happens after the B-frame. If the number of forward predicted blocks is smaller than the number of backward predicted blocks, then one can conclude that the scene change happens before the B frame.

The DCT (discrete cosine transform) provides a decomposition of the original image in the frequency domain. Therefore, DCT coefficients form a natural representation of

texture in the original image. In addition to texture analysis, DCT coefficients can also be used to match images and to detect scene changes. If only the DC components are collected, we have a low-resolution representation of the original image, averaged over 8×8 blocks. This is very helpful because it means much less data to manipulate, and it is found that for some applications, DC components already contain sufficient information. For color analysis, usually only the DC components are used to estimate the color histogram. For scene change detection, usually only the DC components are used to compare the content in two consecutive frames.

Not only can information be extracted from the compressed data for indexing and retrieval, the parameters in the compression process that are not explicitly specified in the bitstream can be very useful as well. One example is the bit rate, i.e., the number of bits used for each picture. For intra coded video (i.e., no motion compensation), the number of bits per picture should remain roughly constant for a scene segment and should change when the scene changes. For example, a scene with simple color variation and texture requires fewer bits per picture compared to a scene that has detailed texture. For intercoding, the number of bits per picture is proportional to the action between the current picture and the previous picture. Therefore, if the number of bits for a certain picture is high, we can often conclude that there is a scene cut.

The compressed-domain approach does not solve all problems, though. To identify useful features from compressed data is typically difficult because each compression technique poses additional constrains, e.g., non-linear processing, rigid data structure syntax, and resolution reduction.

Another issue is that compressed-domain features depend on the underlying compression standard. For different compression standards, different feature extraction algorithms have to be developed. Ultimately, we would like to have new compression standards with maximal content accessibility. MPEG-4 and MPEG-7 already have considered this aspect. In particular, MPEG-7 is a standard that goes beyond the domain of "compression" and seeks efficient "representation" of image and video content. In conclusion, the compressed-domain approach provides significant advantages but also brings new challenges.

2.3 Content-based Features

Section 2.1 described a number of image and video features that can be extracted using well-known techniques in image processing. Section 2.2 described how many of these features are computed or approximated using encoded parameters in image and video compression. Although in both cases there is considerable understanding of the structure of the video, the features in no way estimate the actual image or video content.

In this section we describe several methods to approximate the actual content of an image or video. The desired result has less to do with analytic features such as color, or texture, and more with the actual objects within the image or video.

2.3.1 Object Detection

Identifying significant objects that appear in the video frames is one of the key components for video characterization. Several working systems have generated reasonable results for the detection of a particular object, such as human faces, text, or automobile. These limited domain systems have much greater accuracy than do broad domain systems that attempt to identify any object in the image.

Human Subjects. The "talking head" image is common in interviews and news clips, and illustrates a clear example of video production focussing on an individual of interest. A human interacting within an environment is also a common theme in video. The detection of a human subject is particularly important in the analysis of news footage. An anchorperson will often appear at the start and end of a news broadcast, which is useful for detecting segment boundaries. In sports, anchorpersons will often appear between plays or commercials.

The detection of humans in video is possible using a number of algorithms. Figure 6 shows examples of faces detected using the neural network arbitration method [11]. Most techniques are dependent of scale, and rely heavily on lighting conditions, limited occlusion, and limited facial rotation. For more details on edge detection technology, please see Chapter 10.6.

Captions and Graphics. Text and graphics are used in a variety of ways to convey content to the viewer. They are

FIGURE 6 Recognition of captions and faces [11] (see color insert).

most commonly used in broadcast news, where information must be absorbed in a short time. Examples of text and graphics in video are discussed below.

Text in video provides significant information as to the content of a scene. For example, statistical numbers and titles are not usually spoken but are included in captions for viewer inspection. Moreover, this information does not always appear in closed captions so detection in the image is crucial for identifying potentially important regions.

In news video, captions of the broadcasting company are often shown at low opacity as a watermark in a corner without obstructing the actual video. A ticker-tape is widely used in broadcast news to display information such as the weather, sports scores, or the stock market. In some broadcast news, graphics such as weather forecasts are displayed in a ticker-tape format with the news logo in the lower right corner at full opacity. Captions that appear in the lower third portion of a frame are almost always used to describe a location, person of interest, title, or event in news video. In Fig. 6, the anchorperson's location is listed.

Captions are used less frequently in video domains other than broadcast news. In sports, a score or some information about an ensuing play is often shown in a corner or border at low opacity. Captions are sometimes used in documentaries to describe a location, person of interest, title, or event. Almost all commercials use some form of captions to describe a product or institution, because their time is limited to only a few minutes.

For feature films a producer may use text at the beginning or end of a film for deliberate viewer comprehension, such as character listings or credits. A producer may also start a film with an introduction to the story being told. Throughout a film, captions may be used to convey a change in time or location, which would otherwise be difficult and time consuming for a video producer to create. A producer will seldom use fortuitous text in the actual video unless the wording is noticeable and easy to read in a short time. A typical text region can be characterized as a horizontal rectangular structure of clustered sharp edges, because characters usually form regions of high contrast against the background. By detecting these properties, we can extract potentially important regions from video frames that contain textual information. Most captions are high contrast text such as the black and white chyron commonly found in news video. Consistent detection of the same text region over a period of time is probable since text regions remain at an exact position for many video frames. This may also reduce the number of false detections that occur when text regions move or fade in and out between scenes.

A typical text region can be characterized as a horizontal rectangular structure of clustered sharp edges, because characters usually form regions of high contrast against the background. By detecting these properties we can extract regions from video frames that contain textual information. Figure 7 illustrates the process of text detection; primarily, regions of horizontal titles and captions. We first apply a global horizontal differential filter, F_{HD}, to the image.

$$F_{HD} = \begin{bmatrix} -1/2 & 1 & 1/2 \end{bmatrix}$$

An appropriate binary threshold (Chapter 2) should be set for extraction of vertical edge features. A smoothing filter, F_S, is then used to eliminate extraneous fragments, and to connect character sections that may have been detached.

$$F_S = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$$

Individual regions must be identified though cluster detection (Chapter 2). A bounding box, B_B, should be computed for selection of text regions. We now select clusters with bounding regions that satisfy constraints in cluster size, C_S, cluster fill-factor, C_{FF}, and horizontal-vertical aspect ratio.

$$C_s(n) = \sum_{i=1}^{P} C_n(i)$$

$$C_{FF}(n) = \frac{C_S(n)}{B_{B_{area}}(n)}$$

A sample set of parameters for the font size in Fig. 7 is listed below:

- Cluster size >70 pixels
- Cluster fill factor >.45

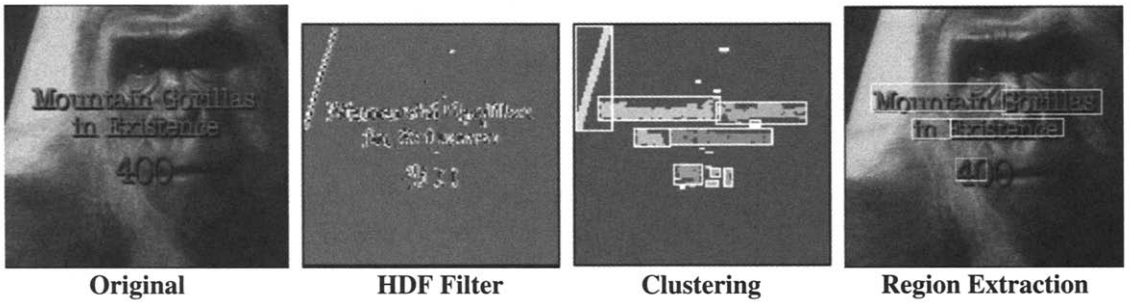

FIGURE 7 Text detection in video (see color insert).

- Horizontal-vertical aspect ratio >.75
- Maximum cluster height = 50 pixels
- Minimum cluster height = 10 pixels
- Maximum cluster width = 150 pixels
- Minimum cluster width = 15 pixels

A cluster's bounding region must have a small vertical-to-horizontal aspect ratio as well as satisfying various limits in height and width. The fill factor of the region should be high to insure dense clusters. The cluster size should also be relatively large to avoid small fragments. Other controlling parameters are listed below.

Finally, we examine the intensity histogram of each region to test for high contrast. This is because certain textures and shapes appear similar to text but exhibit low contrast when examined in a bounded region.

For some fonts a generic optical character recognition (OCR) package may accurately recognize video captions. For most OCR systems, the input is an individual character. This presents a problem in digital video since most of the characters experience some degradation during recording, digitization and compression. For a simple font, we can search for blank spaces between characters and assume a fixed width for each letter [13].

A graphic is usually a recognizable symbol, which may contain text. Graphic illustrations or symbolic logos are used to represent many institutions, locations, and organizations. They are used extensively in news video, where it is important to describe the subject matter as efficiently as possible. A logo representing the subject is often placed in a corner next to an anchorperson during dialogue. Detection of graphics is a useful method for finding changes in semantic content. In this sense, its appearance may serve as a scene break. Recognition of corner regions for graphics detection may be possible through an extension of the scene change technology. Histogram difference analysis, $D_{Hs}(t)$, of isolated image regions instead of the entire image can provide a simple method for detecting corner graphics. An example of a graphics logo detected with $D_{Hs}(t)$ is shown in Fig. 8. In this example, a change is detected in the upper corner, although no scene change is detected.

$$D_{Hs}(t) = \sum_{j=1}^{H/2} \sum_{i=W/2}^{W} \left| H_{(t-T)}(i, j) - H_t(i, j) \right|$$

Frame t Frame t+T

FIGURE 8 Graphics detection through sub-region histogram differencing (see color insert).

Articulated Objects. A particular object is usually the emphasis of a query in image and video retrieval. Recognition of articulated objects poses a great challenge and represents a significant step in content-based feature extraction. Many working systems have demonstrated accurate recognition of animal objects, segmented objects, and rigid objects such as planes or automobiles.

The recognition of a single object is only one potential use of image-based recognition systems. Discrimination of synthetic and natural backgrounds, or an animated or mechanical motion would yield a significant improvement content based feature extraction. For more information on object recognition, please see Chapters 4.6 and 4.7.

2.3.2 Audio and Language

An important element in video indexing creation is the audio track. Audio is an enormous source for describing video content. Words specific to the actual content, or "keywords" can be extracted using a number of language processing techniques [6, 16]. Keywords may be used to reduce indexing and provide abstraction for video sequences. There are many possibilities for language processing in video, but the audio track must first exist as an ASCII document or speech recognition is necessary.

Audio segmentation is needed to distinguish spoken words from music, noise and silence. Further analysis through speech recognition is necessary to align and translate these words into text. Audio selection is made on a frame by frame basis, so it is important to achieve the highest possible accuracy. At a sampling rate of 8 Khz, one frame corresponds to 267 samples of audio. Techniques in language understanding are used for selecting the most significant words and phrases.

In order to use the audio track, we must isolate each individual word. To transcribe the content of the video material, we recognize spoken words using a speech recognition system. Speaker-independent recognition systems have made great strides as of late and offer promise for application in video indexing [5]. Speech recognition works best when closed-captioned data is available. Captions usually occur in broadcast material, such as sitcoms, sports, and news. Documentaries and movies may not necessarily contain captions. Closed-captions have become more common in video material throughout the United States since 1985 and most televisions provide standard caption display.

2.3.3 Rule-based Features

The features described in previous sections may be used with rules that describe a particular type of video scene to create an additional set of content-based features [15]. By using examples from video production standards, we can identify a small set of heuristic rules. In some cases these rules involve the integration of image processing features with audio and language features. Below is a description of three rule-based features suitable for most types of video.

Introduction Scenes: The scenes prior to the introduction of a person usually describe their accomplishments and often precede scenes with large views of the person's face. A person's name is generally spoken and then followed by supportive material. Afterwards, the person's actual face is shown. If a scene contains a proper name, and a large human face is detected in the scenes that follow, we call this an introduction scene. Characterization of this type is useful when searching for a particular human subject because identification is more reliable than using the image or audio features separately. Introduction scenes must meet the following criteria:

$$\text{Scene}_{\text{Introduction}}(i) = \begin{cases} 1, & (\text{Face}_i = \text{TRUE} \;\&\&\; \text{WORD}_i = \text{PROPER\_NAME}) \\ 0, & (\text{otherwise}) \end{cases}$$

Adjacent Similar Scenes: The color histogram difference measure gives us a simple routine for detecting similarity between scenes. Scenes between successive shots of a human face usually imply illustration of the subject. For example, a video producer will often interleave shots of research between shots of a scientist. Images that appear between two similar scenes that are less than T_{SS} seconds apart are characterized as an adjacent similar scene. Scene(i) is an adjacent similar scene if it meets the following criteria:

$$\text{Scene}_{\text{Adjacent Similar}}(i) = \begin{cases} 1, & (\text{Scene}(i-T) = \text{Scene}(i+T)) \\ & \text{AND} \\ & \left(\left|\text{Scene}_{\text{Start}}(i-T) - \text{Scene}_{\text{Start}}(i+T)\right| < T_{SS}\right) \\ 0, & (\text{otherwise}) \end{cases}$$

where T_{SS} is on the order of 10 seconds or less.

Short Successive Scenes: Short successive shots often introduce an important topic. By measuring the duration of each scene, S_D, we can detect these regions and identify short successive sequences. A set of scenes is short successive if a group of 5 or more scenes meet the following criteria:

$$\text{Scene}_{\text{Short Successive}}(i) = \begin{cases} 1, & \left(\begin{array}{l} \text{Scene}_{\text{Duration}}(i-T) < S_D \;\&\&\; \text{Scene}_{\text{Duration}}(i+T) < S_D \;\&\& \\ \text{Scene}_{\text{Duration}}(i+2T) < S_D \;\&\&\; \text{Scene}_{\text{Duration}}(i+3T) < S_D \ldots \end{array}\right) \\ 0, & (\text{otherwise}) \end{cases}$$

where S_D for each scene is on the order of 3 seconds or less.

2.3.4 Embedded Video Features

A final solution for content-based feature extraction is the use of known procedures for creating video. Video production manuals provide insight into the procedures used during video editing and creation. There are many documents that describe the editing and production procedures for creating video segments, but one of the most recent is published by Pryluck [10].

One of the most common elements in video production is the ability to convey climax or suspense. Producers use a variety of different effects; ranging from camera positioning, lighting, and special effects to convey this mood to an audience. Detection of procedures such as these are beyond the realm of present image and language understanding technology. However, many of the important features described in Sections 2, 3, and 4 were derived from research in the video production industry.

Structural information as to the content of a video is a useful tool for indexing video. For example, the type of video being used (documentaries, news footage, movies and sports) and its duration may offer suggestions to assist in object recognition. In news footage, the anchorperson will generally appear in the same pose and background at different times. The exact locations of the anchorperson can then be used to delineate story breaks. In documentaries, a person of expertise will appear at various points throughout the story when topical changes take place. There are also many visual effects introduced during video editing and creation that may provide information for video content. For example, in documentaries the scenes prior to the introduction of a person usually describe their accomplishments and often precede scenes with large views of the person's face.

A producer will often create production notes that describe in detail action and scenery of a video, scene by scene. If a particular feature is needed for an application in image or video databases, the description may have already been documented during video production.

Another source of descriptive information may also embedded in the video stream in the form of timecode and geospatial (GPS/GIS) data. These features are useful in indexing precise segments in video or a particular location in spatial coordinates. Aeronautic and automobile surveillance video will often contain GPS data that may be used as a source for indexing.

3 From Low-Level Features to High-Level Semantics

A feature is good if and only if similar objects are close to each other in the feature space, and dissimilar objects are far apart. However, the concept of "similar" or "dissimilar" involves high-level semantics. The challenge therefore is to transform, or "warp," the low-level feature space to represent high-level semantic similarity/dissimilarity. A popular approach to finding the "warping" function is through learning. Given a small amount of training data, the system can automatically find the appropriate "warping" function to transform the low-level feature space to represent high-level semantics.

In learning, one must decide what to use as the ground truth data, i.e., what one wants the "warping" function to be optimized for. It turns out that there are two major forms that are most often used: similarity/dissimilarity (SD) and keyword annotations (KA). If we know that some objects are similar to each other while others are not, the feature space should be "warped" so that similar objects get closer, and dissimilar objects get farther. If the training data has some keyword annotated for each object, we want objects that share the same keywords get closer while otherwise get farther. Both SD and KA have their advantages and suitable applications. SD is convenient for the user, and it does not require any explanation why two objects are similar or not (sometimes the reason is hard to be presented to the system by the user). Therefore, SD is suitable for user optimized learning, e.g., to learn what the user really means by giving some examples. SD is almost exclusively used by relevance feedback—a very hot research topic today. KA is good for system maintainers to improve the general performance of the retrieval system. Recent work in video retrieval has shown an interesting shift from query by example (QBE) [24], [25] to query by keywords (QBK). The reason is that it allows users to specify queries with keywords, as they are used to in text retrieval. Moreover, KA allows the knowledge learned to be accumulated by simply adding more annotations, which is often not obvious when using SD. Therefore, by adding more and more annotations, the system maintainer can let the end-user feel that the system works better and better. Both SD and KA have their constraints, too. For example, SD is often too user-dependent and the knowledge obtained is hard to accumulate, while KA is often limited by a predefined small lexicon.

The learning process is determined not only by the form of the training data, but also by their availability. Sometimes all the training data are available before the learning, and the process can be done in one batch. We often call this off-line learning. If the training data are obtained gradually and the learning is progressively refined, we call it on-line learning. Both cases were widely studied in the literature. In the following sections, we will focus on applying these learning algorithms on retrieval systems to improve the similarity measure.

3.1 Off-Line Learning

If all the training data is available at the very beginning, learning can be done in one step. This kind of off-line learning is often applied before the system is provided to users. Most off-line learning systems handle keyword annotations (KA). The keywords are often given as a predetermined set, organized in different ways. For example, Basu et al. [26] defined a lexicon as relatively independent keywords describing events, scenes and objects. Many authors prefer the tree structure [27,28], as it is clean and easy to understand. Naphade et al. [29] and Lee et al. [30] used graph structure, which is appropriate if the relationship between keywords is very complex.

Once the training data is given, a couple of learning algorithms, parametric or non-parametric, can be used to learn the concepts behind the keywords. As far as the authors know, at least Gaussian mixture model (GMM) [26], support vector machine (SVM) [31], hybrid neural network [32], multi-nets [29], distance learning network [33] and kernel regression [27] have been studied in the literature. A common characteristic of these algorithms is that all of them can model potentially any distribution of the data. This is expected because we do not know how the objects that share the same concept are distributed in the low-level feature space. One assumption we can probably make is that in the low-level feature space, if two objects are very close to each other, they should be semantically similar, or be able to infer some knowledge to each other. On the other hand, if two objects are far from each other, the semantic link between them should be weak. Notice that because of the locality of the semantic inference, this assumption allows objects with the same semantic meaning to lie in different places in the feature space, which cannot be handled by simple methods such as linear feature reweighing. If the above assumption does not hold, probably none of the above learning algorithms will help improve the retrieval performance too much. The only solution to this circumstance might be to find better low-level features for the objects.

Different learning algorithms have different properties and are good for different circumstances. Take the Gaussian mixture model as an example. It assumes that the objects having the same semantic meaning are clustered into groups. The groups can lie at different places in the feature space, but each of them follows a Gaussian distribution. If the above assumptions are true, GMM is the best way to model the data: it is simple, elegant, easy to solve with algorithms such as expectation maximization (EM) [34] and sound in theoretical point of view. However, the above assumptions are very fragile: we do not know how many clusters the GMM will

have, and no real case will happen that each cluster is a Gaussian distribution. Despite the constraints, GMM is still very popular for its many advantages. Kernel regression (KR) is another popular machine learning technique. Instead of using a global model like GMM, KR assumes some local inference (kernel function) around each training sample. From the un-annotated object's point of view, to predict its semantic meaning, an annotated object that is closer will have a higher influence, and a farther one will have less. Therefore, it will have similar semantic meanings to its close-by neighbours. KR can model any distribution naturally, and also has sound theory behind it. The limitation of KR is that the kernel function is hard to select, and the number of samples needed to achieve a reasonable prediction is often high. Support vector machine (SVM) [35, 36] is a recent addition to the toolbox of machine learning algorithms that has shown improved performance over standard techniques in many domains. It has been one of the most favourite methods among researchers today. The basic idea is to find the hyperplane that has the maximum margin toward the sample objects. Margin here means the distance the hyperplane can move along its normal before hitting any sample object. Intuitively, the greater the margin, the less the possibility that any sample points will be misclassified. For the same reason, if a sample object is far from the hyperplane, it is less likely to be misclassified.

Although after applying the learning algorithm, the semantic model can be used to tell the similarity between any two objects already, most systems require a fusion step. The reason is that the performance of the statistically learned models is largely determined by the size of the training data set. Since often the training data is manually made, very expensive and thus small, it is risky to believe that the semantic model is good enough. In [27], semantic distance is combined with low-level feature distance through a weighting mechanism to give the final output, and the weight is determined by the confidence of the semantic distance. In [26], several GMM models are trained for each feature types, and the final result is generated by fusing the outputs of all the GMM models.

Keyword annotation is very expensive because it requires a lot of manual work. Chang and Li [37] proposed to employ another way of getting the ground truth data. They used 60,000 images as the original set and synthesize another set by 24 transforms such as rotation, scaling, cropping, etc. Obviously, images after the transforms should be similar to the one before the transform. They discovered a perceptual function called dynamic partial distance function (DPF). Synthesizing new images by transforms and using them as training data is not new. For example, people play this trick in face recognition systems when the training image set has very few images (e.g., only one). Despite the fact that transforms may not be complete as a model of similarity, this is a very convenient way of getting a lot of training data, and DPF seems to have reasonable performance as reported in [37].

3.2 On-Line Learning

Compared to off-line learning, on-line learning does not have the whole set of training data beforehand. The data are often obtained during the process, which makes the learning process a best effort one and highly dependent on the input training data, even the order they come in. However, on-line learning involves the interaction between the system and the user. The system can then quickly modify its internal model in order to output good results for each specific user. As discussed in Section 1, similarity measure in information retrieval systems is highly user-dependent. On-line learning's adaptive property makes it very suitable for such applications. In retrieval systems, on-line learning is used in three scenarios: relevance feedback, finding the query seed, and enhancing the annotation efficiency.

3.2.1 Relevance Feedback

Widely used in text retrieval, relevance feedback was first proposed by Rui et al. as an interactive tool in content-based image retrieval [38]. Since then it has been proven to be a powerful tool and has become a major focus of research in this area. Relevance feedback often does not accumulate the knowledge the system learned. That's because the end-user's feedback is often unpredictable and inconsistent from user to user, or even query to query. If the user who gives the feedback is trustworthy and consistent, feedback can be accumulated and added to the knowledge of the system, as was suggested by Lee et al. [30].

3.2.2 Query Concept Learner

In a query by example system, it is often hard to initialise the first query, because the user may not have a good example to begin with. Having got used to text retrieval engines such as Google, users may prefer to query the database by keyword. Many systems with keyword annotations can provide such kind of service. Chang et al. recently proposed the SVM active learning system [39] and MEGA system [40], which can be alternate solutions. SVM active learning and MEGA have similar ideas but with different tools. They both want to find a query-concept learner that learns query criteria through an intelligent sampling process. No example is needed as the initial query. Instead of browsing the database completely randomly, these two systems ask the user to provide some feedback and try to quickly capture the concept in the user's mind. The key to success is to maximally utilize the user's feedback and quickly reduce the size of the space that the user's concept lies in. Active learning is the answer.

Active learning is an interesting idea in the machine learning literature. While in traditional machine learning research, the learner typically works as a passive recipient

of the data, active learning enables the learner to use its own ability to respond to collect data and to influence the world it is trying to understand. A standard passive learner can be think of as a student that sits and listens to a teacher, while an active learner is a student that asks the teacher questions, listens to the answers and asks further questions based on the answer. Active learning has shown very promising results in reducing the number of samples required to finish a certain task.

In practice, the idea of active learning can be translated into a simple rule: if the system is allowed to propose samples and get feedback, always propose those samples that the system is most confused of, or that can bring the greatest information gain. Following the rule, SVM active learning becomes very straightforward. In SVM, objects far away from the separating hyperplane are easy to classify. The most confused objects are those that are close to the boundary. Therefore, during the feedback loop, the system will always propose the images closest to the SVM boundary for the user to annotate.

3.2.3 Efficient Annotation through Active Learning

Keyword annotation is a very expensive work, as it can only be done manually. It is natural to look for methods that can improve the annotation efficiency. Active learning turns out to be also suitable for this job. In [27], Zhang and Chen proposed a framework for active learning during the annotation. For each object in the database, they maintain a list of probabilities, each indicating the probability of this object having one of the attributes. During training, the learning algorithm samples objects in the database and presents them to the annotator. For each sampled object, each probability is set to be one or zero depending on whether or not the corresponding attribute is assigned by the annotator. For objects that have not been annotated, the learning algorithm estimates their probabilities with biased kernel regression. Knowledge gain is then defined to determine, among the objects that have not been annotated, which one is the most uncertain to the system of. The system then presents it as the next sample to the annotator.

Naphade et al. proposed a very similar work in [31]. However, they used a support vector machine to learn the semantics. They have essentially the same method as Chang et al.'s SVM active learning [39] to choose new samples for the annotator to annotate.

4 Retrieval Techniques

In Section 2, we described analytical and content-based features which can be extracted from image and video segments. In this section we describe techniques for establishing correspondence between these features.

4.1 Feature-based Retrieval (Statistical and Compressed)

Correspondence between analytic features is established with the difference measures described in Section 2. This is straightforward for image matching features, where a match is based on the minimum absolute difference, $D(t)$, or histogram difference, $D_H(t)$. In the case of features based on motion, texture and shape, the difference is based on the Euclidean distance between the parameters of the perspective feature. The difference measures may be applied to the entire image or a sub-region of the image for better correspondence between objects in the image. Regardless of the granularity in applying difference measures, a key problem with color image matching is that similar colors do not necessarily provide similar content, as seen in Fig. 9.

Image correspondence is important for identifying scenes that appear often in a video segment. The color histogram is not only useful for detecting scene changes, but serves as an adequate method for image correspondence. A histogram from the first video frame of each scene is stored and compared with that of video frames in subsequent scenes. An analysis of the entire image requires less computation than sub-region differencing, but the image match is less robust to foreground objects. Global image matching is particularly useful with images of uniform color and texture.

In news footage, an icon or logo is often used to symbolize the subject of the video. This icon is usually placed in the upper-quarter of the image. Although the background of the image remains the same, changes in this icon represent changes in content. By applying histogram differencing to a small region in the image we can detect changes in news icons. Processing sub-regions requires more computation, but the resulting image match is usually more robust. Objects that appear away from the background are usually easier to match with sub-region differencing. Sub-region differencing is also more affective with images of complex color and texture.

4.2 Content-based Retrieval

The main problem in image in video indexing is that users query content and most systems only match statistical features such as color and texture. Figure 10 illustrates two images of essentially identical content, but almost no similarities in color, shape, texture, or motion. In this case, the motion of the players is similar, but the angle of camera will yield two separate forms of object motion from the original video sequence.

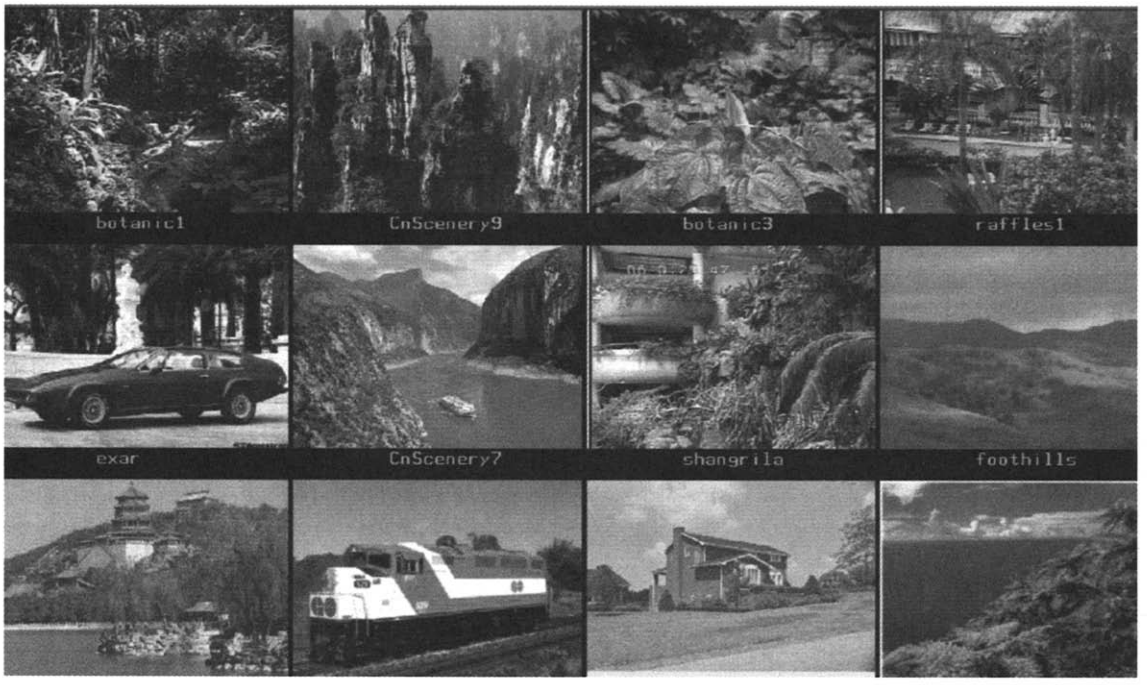

FIGURE 9 Images with similar color.

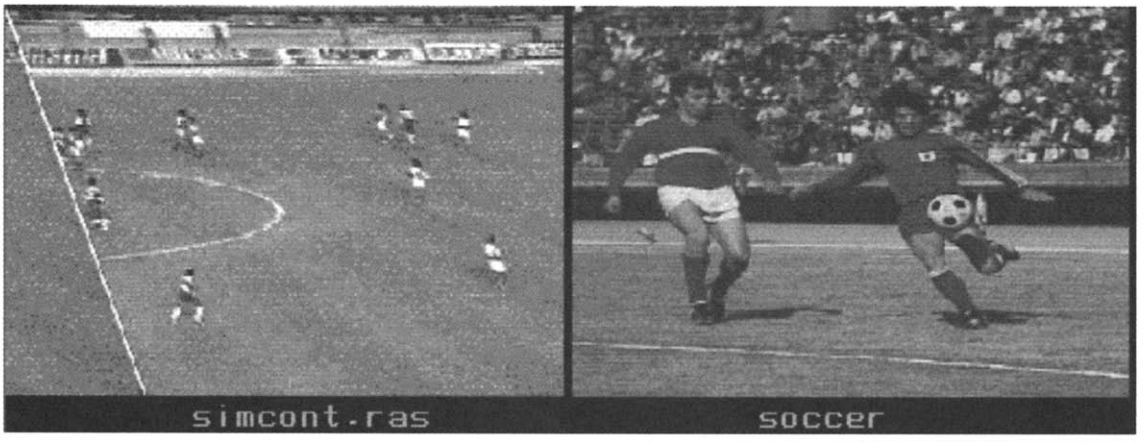

FIGURE 10 Images with similar content (see color insert).

Content matching attempts to correlate actual objects with a given query. The user is not limited to selections based on similar color properties, but rather a collection based on content. In this form of matching, the query may be an image or text. The content features, such as caption and face detection correspond to textual descriptions so a query need not be an image.

A number of content-based image and video systems are applicable to the features described in this chapter. In the table below, we list several potential query applications associated with content-based and statistical features.

Several working systems have demonstrated the potential of content-based matching for identifying specific objects and stories. Three of the more interesting systems are discussed below.

Name-It, is a system for matching a human face to a name in news video [12]. It approximates the likelihood of a particular face belonging to a name in close proximity within the transcript. Integrated language and image understanding technology make the automation of this system possible.

Spot-It, is a topological system that attempts to identify known characteristics in news video for indexing and classification [8]. It has reasonable success in identifying common video themes such as interviews, group discussions, and conference room meetings.

TABLE 1 Potential query applications

Query type	Associated feature
Pans or zooms in video	Camera motion
Action or moving objects	Object motion
Important scenes	Short sequences, adjacent similar scenes, introduction scenes
Human subjects	Face detection, introduction scenes, video text detection
Video captions	Video text detection
Subject location	GPS, video text detection
Image scenery	Color difference, texture
Name or description	Audio and language analysis
Simple objects	Color difference
Segment boundaries	Face detection, scene changes, black frames
Segment boundaries	Face dectection, scene changes, black frames

Pictorial transcripts, a working system at AT&T Research Laboratories has shown promising results in video summarization when closed-captions are used with statistical visual attributes [14]. CNN video is digitized and displayed in an HTML environment with text for audio and a static image for every paragraph. More than 3000 hours of processed video can be searched and browsed.

4.3 Considerations in Multimedia Databases

The retrieval of an image or video segment is often limited in practical multimedia databases. There are many factors to consider when creating an image or video database, such as optimization for large databases, the type of query, the presentation of results, and the measure of success.

Retrieval efficiency is an important concern for image and video databases. Flat file systems are sufficient when the size of a collection is moderate. However, a more robust solution is necessary when image and video libraries grow to several thousand units of data. Researchers have developed tree structure optimization systems that greatly reduce the search space by clustering image characteristics into small subsets for later retrieval [21].

4.3.1 Queries: Image or Text

For most image and video retrieval systems, the query is an image. When the comparison is based on analytic features, the results can often be ambiguous, as shown in section 5.1. Content-based features provide a more accurate match to the given query, but the results are based on image processing technology which is only capable of recognizing a small number of objects.

Text queries eliminate ambiguity in the query, and work only with content based features. There is still a dependence on content based feature extraction, but there is limited uncertainty in the query. This type of the query may also be used to match the title of the image or the transcript of the video.

4.3.2 Presentation of Results

The presentation of a query result is an important part of the image or visual query system. Presentations are usually visual and textural in layout. Textual presentations provide more specific information and are useful when presenting large collections of data. Visual, or iconic, presentations are more useful when the content of interest is easily recalled from imagery. This is quite often the case in stock footage video where there is no audio to describe the content. Section 7 describes current working systems for presentation of image and video results.

4.3.3 Testing and Evaluation

In image and video databases, accuracy is based on the relevance of the output set of images or video to a particular query. A user defines the level of quality, therefore, the evaluation of an image or video retrieval system cannot be based on traditional analytic measures. The accuracy of these systems is purely subjective, which requires that some human intervention take place during evaluation.

User studies or some form of subjective rating are essential during the design and development of an image and video database system. Researchers have successfully demonstrated the utility of user-studies in testing image and video retrieval applications [3,4]. A subject is generally shown a query and asked to rank the resulting image or video segments on a scale. For example, in a video database the user might be asked to rate the quality of selection on a scale ranging from "high relevance" to "low relevance". An example of a user-study interface for video retrieval is shown in Fig. 11.

5 Video Access and Browsing

With the size of video collections growing to thousands of hours, technology is needed to effectively browse segments in a short time without losing the content of the video. Simplistic browsing techniques, such as increased playback speed and skipping video frames at fixed intervals, reduce video viewing time. However, increased video rates eliminate the majority of the audio information and distorts much of the image information, and displaying video sections at fixed intervals merely gives a random estimate of the overall content. An ideal browser would display only the video pertaining to a segment's content, suppressing irrelevant data.

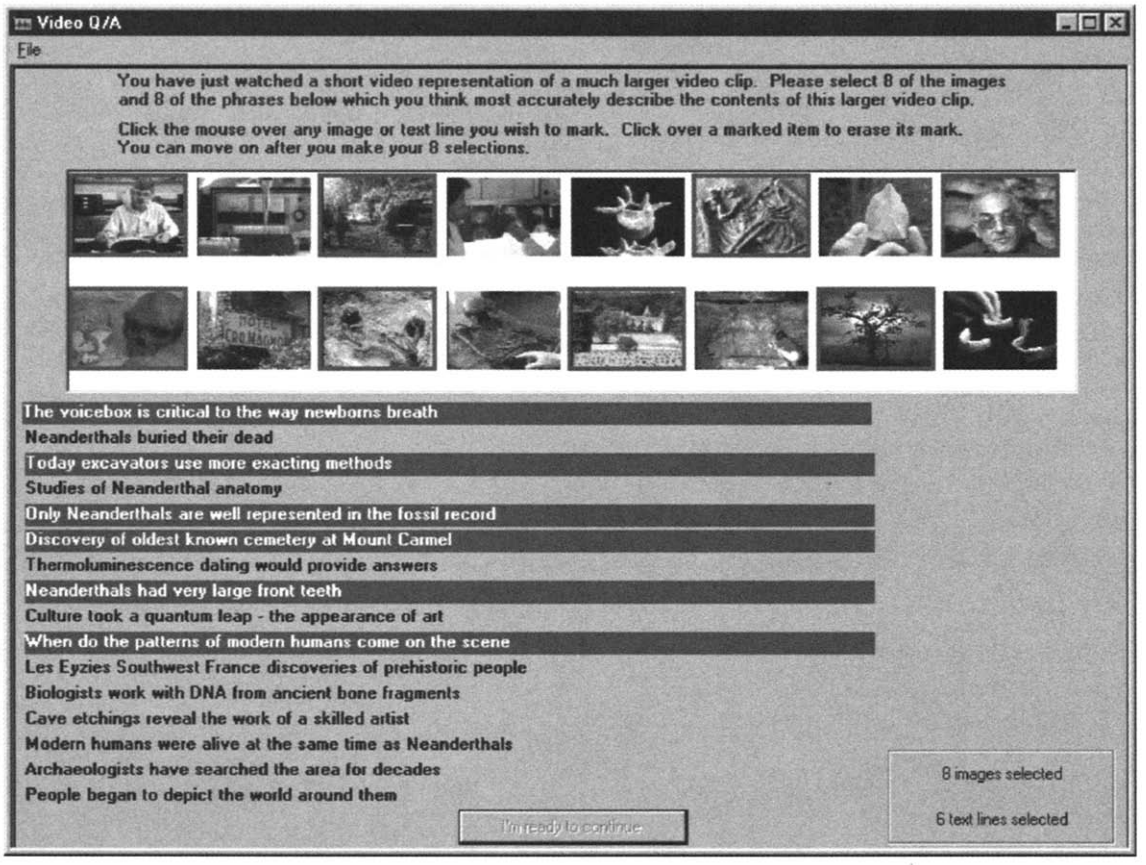

FIGURE 11 Video retrieval user-study interface [3] (see color insert).

A multimedia abstraction ideally preserves and communicates the essential content of a video segment via a compact representation. Examples of multimedia abstractions include short text titles and single thumbnail images. Another commonly used abstraction presents an ordered set of representative, "thumbnail" images simultaneously on a computer screen. Image statistics, such as histogram analysis and texture, camera structure and scene changes are the dominant factors in these systems. While these abstractions have proven useful in various contexts, their static nature ignores video's temporal dimension.

In addition, these abstractions often concentrate exclusively on the image content and neglect the audio information carried in a video segment. Preliminary investigations suggest that the opposite emphasis offers greater value. The video skim, as illustrated in Figure 12, is one of the first systems to integrate technology in image, language, and audio understanding for browsing and summarization [15]. Recently, researchers have proposed browsing representations based on information within the video. These systems rely on the motion in a scene, placement of scenes breaks, but not on integrated image and language understanding.

Although the video skimming work represents the one of the first experiments in integrating language and image understanding, there are a number of efforts that combine language and image understanding as of late. The application of technology integration is different for these systems, however, they all demonstrate the advantages of using multiple modalities in video characterization and summarization. Examples of these systems are discussed below.

Browsing through clustering: This system was designed to cluster image regions for browsing digital video [20]. It uses many of the image statistics mentioned earlier, but it attempts to process scene transitions rather than just process individual frames.

High rate keyframe browsing: The Digital Library Research Group at the University of Maryland, College Park, has conducted a user study to test optimal frame rates for keyframe based browsing [4]. They use many of the same image analysis techniques mentioned earlier to extract keyframes, and they quantify their research through studies of a video slide show interface at various frame rates.

Video abstracts: The Movie Content Analysis (MoCA) group in Mannheim, Germany has created a system for movie abstraction based on the occurrence of image statistics and audio frequency analysis to detect dialogue scenes [9].

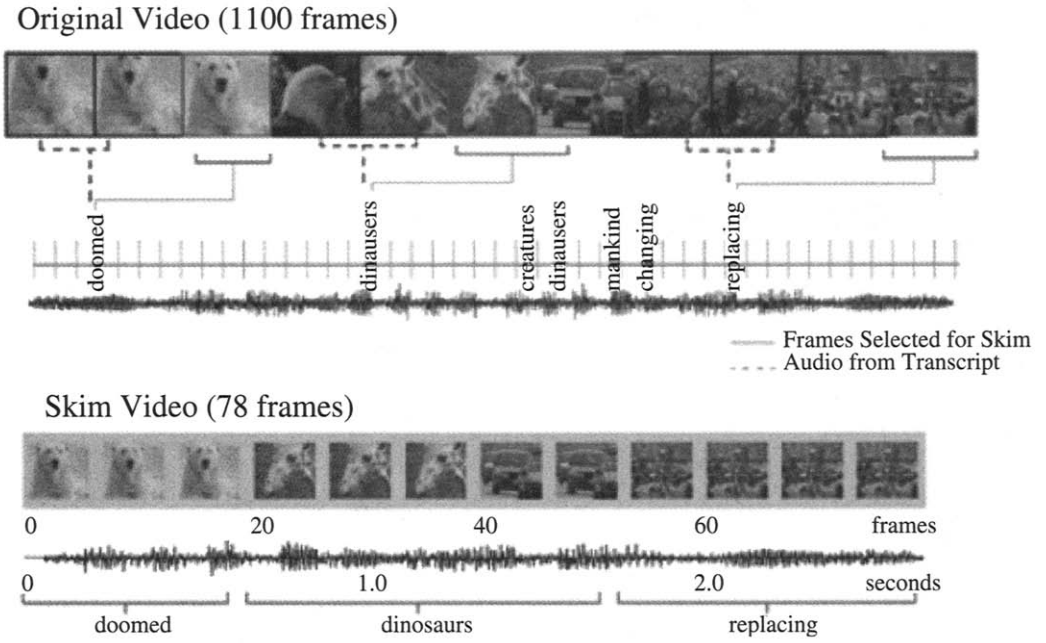

FIGURE 12 Illustration of video skimming (see color insert).

6 The MPEG-7 Standard

As pointed out earlier in this chapter, instead of trying to extract relevant features, manually or automatically, from original or compressed video, a better approach for content retrieval should be to design a new standard in which such features, often referred to as meta-data, are already available. MPEG-7, an ongoing effort by the Moving Picture Experts Group, is exactly working toward this goal, i.e., the standardization of meta-data for multimedia content indexing and retrieval.

MPEG-7 is an activity that is triggered by the growth of digital audiovisual information. The group strives to define a "multimedia content description interface" to standardize the description of various types of multimedia content, including still pictures, graphics, 3D models, audio, speech, video, and composition information. It may also deal with special cases such as facial expressions, personal characteristics.

The goal of MPEG-7 is exactly the same as the focus of this chapter, i.e., to enable efficient search and retrieval of multimedia content. Once finalized, it will transform the text-based search and retrieval (e.g., keywords) as is done by most of the multimedia databases nowadays, into a content-based approach, e.g., using color, motion, or shape information. MPEG-7 can also be thought of as a solution to describing multimedia content. If one looks at PDF (portable document format) as a standard language to describe text and graphic documents, then MPEG-7 will be a standard description for all types of multimedia data, including audio, images, and video.

Compared with earlier MPEG standards, MPEG-7 possesses some essential differences. For example, MEPG-1, 2 and 4 all focus on the representation of audiovisual data, but MPEG-7 will focus on representing the "meta-data" (information about data). MPEG-7, however, may utilize the results of previous MPEG standards, e.g., the shape information in MPEG-4 or the motion vector field in MPEG-1/2.

Figure 13 shows the scope of the MPEG-7 standard. Note that feature extraction is outside the scope of MPEG-7, so is the search engine. This is owing to one approach constantly taken by most of the standard activities, i.e., "to standardize the minimum". Therefore, the analysis (feature extraction) should not be standardized, so that after MPEG-7 is finalized, various analysis tools can still be further improved over time. This also leaves room for competition among vendors and researchers. This is similar to that MPEG-1 does not specify motion estimation, and that MPEG-4 does not specify segmentation algorithms. Likewise, the query process (the search engine) should not be standardized. This allows the design of search engines and query languages to adapt to different application domains, and also leaves room for further improvement and competition. Summarizing, MPEG-7 takes the approach that standardizes only what is necessary so that the description for the same content may adapt to different users and different application domains.

One goal of MPEG-7 is to provide a standardized method for describing features of multimedia data. For images

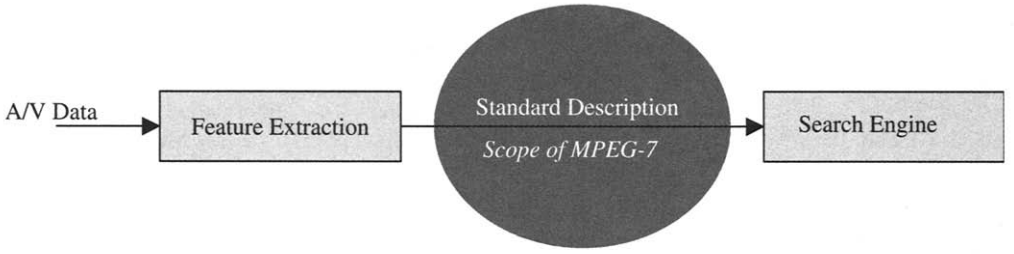

FIGURE 13 The scope of MPEG-7.

and video, colors or motion are example features that are desirable in many applications. MPEG-7 will define a certain set of descriptors to describe these features. For example, the color histogram can be a very suitable descriptor for color characteristics of an image, and motion vectors (as commonly available in compressed video bitstreams) form a useful descriptor for motion characteristics of a video clip. MPEG-7 also uses the concept of description scheme (DS) which means a framework that defines the descriptors and their relationships. Hence, the descriptors are the basic of a description scheme. Description then implies an instantiation of a description scheme. MPEG-7 not only want to standardize the description, it also wants the description to be efficient. Therefore, MPEG-7 also considers compression techniques to turn descriptions into coded descriptions. Compression reduces the amount of data that need to be stored or processed. Finally, MPEG-7 will define a description definition language (DDL) that can be used to define, modify, or combine descriptors and description schemes. Summarizing, MPEG-7 will standardize a set of descriptors and DS's, a DDL, and methods for coding the descriptions.

MPEG-7 has a large variety of applications, such as digital libraries, multimedia directory services, broadcast media selection, multimedia authoring. Here are some examples. With MPEG-7, the user can draw a few lines on a screen to retrieve a set of images containing similar graphics. The user can also describe movements and relations between a number of objects to retrieve a list of video clips containing these objects with the described temporal and spatial relations. Also, for a given content, the user can describe actions and then get a list of scenarios where similar.

7 Conclusion

Image and video retrieval systems have been primarily based on statistical analysis of image pixels. With increasing efforts in feature-based analysis and extraction, these systems are becoming usable and efficient in retrieving perceptual content. Powerful content-based indexing and retrieval tools can be developed for image/video archives, complementing the traditional text-based techniques. There are no "best" features for "all" image domains. It's a matter of creating a good "solution" using multiple features for a specific application.

References

[1] Akutsu, A. and Tonomura, Y. "Video Tomography: An Efficient Method for Camerawork Extraction and Motion Analysis," Proc. of ACM Multimedia '94, Oct., 1994, San Francisco, CA, 349–356.

[2] Arman, F., Hsu, A., and Chiu, M.-Y. "Image Processing on Encoded Video Sequences," Multimedia Systems, 1994, 1, 211–219.

[3] Christel, M.G., Winkler, D.B., and Taylor, C.R. Improving Access to a Digital Video Library. In Human-Computer Interaction: INTERACT97, the 6th IFIP Conf. On Human-Computer Interaction (Sydney, Australia, July 14–18, 1997).

[4] Ding, L., et al. "Previewing Video Data: Browsing Key Frames at High Rates Using a Video Slide Show Interface," Proceedings of the International Symposium on Research Development and Practice in Digital Libraries, Tsukuba Science City, Japan, November 1997, 151–158.

[5] Hauptmann, A. and Smith, M. "Text, Speech, and Vision for Video Segmentation," AAAI Fall 1995 Symposium on Computational Models for Integrating Language and Vision.

[6] Mauldin, M. "Conceptual Information Retrieval: A Case Study in Adaptive Partial Parsing," Kluwer Press, September 1991.

[7] Meng, J. and Chang, S.-F. "Tools for Compressed-Domain Video Indexing and Editing," SPIE Conference on Storage and Retrieval for Image and Video Database, San Jose, Feb. 1996.

[8] Nakamura, Y. and Kanade, T. "Semantic Analysis for Video Contents Extraction—Spotting by Association in News Video." Proceedings of the Fifth ACM International Multimedia Conference, October, 1997.

[9] Pfeiffer, S. Lienhart, R., Fischer, S., and Effelsberg, W. "Abstracting Digital Movies Automatically," Journal of Visual Communication and Image Representation, 7, 4, 345–53, Dec. 1996.

[10] Pryluck, C., Teddlie, C., and Sands, R. "Meaning in Film/Video: Order, Time and Ambiguity," Journal of Broadcasting 26, 685–695, 1982.

[11] Rowley, H., Baluja, S., and Kanade, T. "Neural network-based face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, January 1998.

[12] Satoh, S., Kanade, T., and Smith, M. "NAME-IT: Association of Face and Name in Video," Computer Vision and Pattern Recognition, June, 1997, San Juan, Puerto Rico.

[13] Sato, T. Kanade, T., Hughes, E., and Smith, M. "Video OCR for Digital News Archives," IEEE Workshop on Content-Based Access of Image and Video Databases (CAIVD'98), Bombay, India, January, 1998.

[14] Shahraray, B. and Gibbon, D. "Authoring of Hypermedia Documents of Video Programs," Proceedings of the Third ACM Conference on Multimedia, 401–409, San Francisco, CA, November, 1995.

[15] Smith, M.A. and Kanade, T. "Video Skimming and Characterization Through the Combination of Image and Language Understanding Techniques," Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 1997.

[16] "TREC 93," Proceedings of the Second Text Retrieval Conference, D. Harmon, editor, sponsored by ARPA/SISTO, August 1993.

[17] Tse, Y.T. and Baker, R.L. "Global Zoom/Pan Estimation and Compensation for Video Compression," Proceedings of ICASSP 1991, 2725–2728.

[18] Wactlar, H.D., Kanade, T., Smith, M.A., and Stevens, S.M. "Intelligent Access to Digital Video: Informedia Project," IEEE Computer, 29, 5 (May 1996), 46–52.

[19] Wang, H. and Chang, S.F. "A Highly Efficient System for Automatic Face Region Detection in MPEG Video Sequences," IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Multimedia Systems and Technologies, 1997.

[20] Yeung, M., Yeo, B., Wolf, W., and Liu, B. Video Browsing Using Clustering and Scene Transitions on Compressed Sequences. Proceedings IS&T/SPIE Multimedia Computing and Networking, February 1995.

[21] Flickner, M., et al. "Query by Image Content," IEEE Computer, 23–32, September 1995.

[22] Zabih, R., Miller, J., and Mai, K. "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," Proceedings of the ACM International Conference on Multimedia, San Francisco, CA, November 1995.

[23] Zhang, H.J., Tan, S., Smoliar, S., and Yihong, G. "Video Parsing, Retrieval and Browsing: an Integrated and Content-based Solution," Proceedings of the ACM International Conference on Multimedia, San Francisco, CA, November 1995.

[24] Rui, Y. and Huang, T.S. "Image Retrieval: Current Techniques, Promising, Directions and Open Issues," Journal of Visual Communication and Image Representation, 10, 4, April 1999.

[25] Flicker, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M, Hafner, J, Lee, D., Petkovic, D., Steele, D., and Yanker, P. "Query by Image and Video Content: The QBIC System," IEEE Computer, 23–32, 28, 9, 1995.

[26] Basu, S. Naphade, M., and Smith, J.R. "A Statical Modeling Approach to Content Based Retrieval," IEEE ICASSP, 2002.

[27] Zhang, C. and Chen, T. "An Active Learning Framework for Content Based Information Retrieval," IEEE Trans. on Multimedia, Special Issue on Multimedia Database, 260–268, 4, 2, June 2002.

[28] Park, Y. "Efficient Tools for Power Annotation of Visual Contents: A Lexicographical Approach," ACM Multimedia, 426–428, 2000.

[29] Naphade, M.R., Kozintsev, I., Huang, T.S., and Ramchandran, K. "A Factor Graph Framework for Semantic Indexing and Retrieval in Video," Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, 2000.

[30] Lee, C.S., Ma, W.-Y., and Zhang, H.J. "Information Embedding Based on User's Relevance Feedback for Image Retrieval," Invited Paper, SPIE Int. Conf. Multimedia Storage and Archiving Systems IV, Boston, 19–22, Sep. 1999.

[31] Naphade, M.R., Lin, C.Y., Smith, J.R., Tseng, B. and Basu, S. "Learning To Annotate Video Databases," SPIE Conference on Storage and Retrieval on Media Databases, 2002.

[32] Ma, W.Y. and Manjunath, B.S. "Texture Features and Learning Similarity," IEEE Proceedings CVPR '96, 425–430, 1996.

[33] Squire, D. McG. "Learning a Similarity-based Distance Measure for Image Database Organization from Human Partitionings of an Image Set," IEEE Workshop on Applications of Computer Vision (WACV'98), 88–93, 1998.

[34] Dempster, A.P., Laird, N.M., and Rubin, D.B. "Maximum-likelihood from incomplete data via the EM algorithm", J. Royal Statist. Soc., Ser. B, 1–38, 39, 1, 1977.

[35] Burges, C. "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, 121–167, 2, 2, 1998.

[36] Cristianini, N. and Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[37] Chang, E., and Li, B. "On Learning Perceptual Distance Function for Image Retrieval," IEEE ICASSP, 2002.

[38] Rui, Y., Huang, T.S., Ortega, M., and Mehrotra, S. "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," IEEE Trans. On Circuits and Systems for Video Technology, 644–655, 8, 5, Sep. 1998.

[39] Tong, S. and Chang, E. "Support Vector Machine Active Learning for Image Retrieval," ACM Multimedia, 2001.

[40] Chang, E. and Li, B. "MEGA—The Maximizing Expected Generalization Algorithm for Learning Complex Query Concepts," UCSB Technical Report, August 2001.

FIGURE 9.1.1 Left: original; Right: filtered.

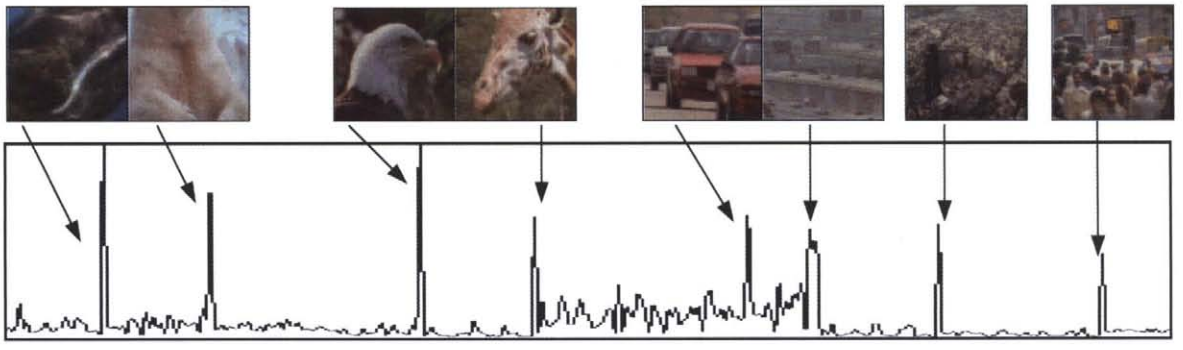

FIGURE 9.1.2 Histogram difference, $D_{H\text{-}RGB}(t)$, for scene segmentation.

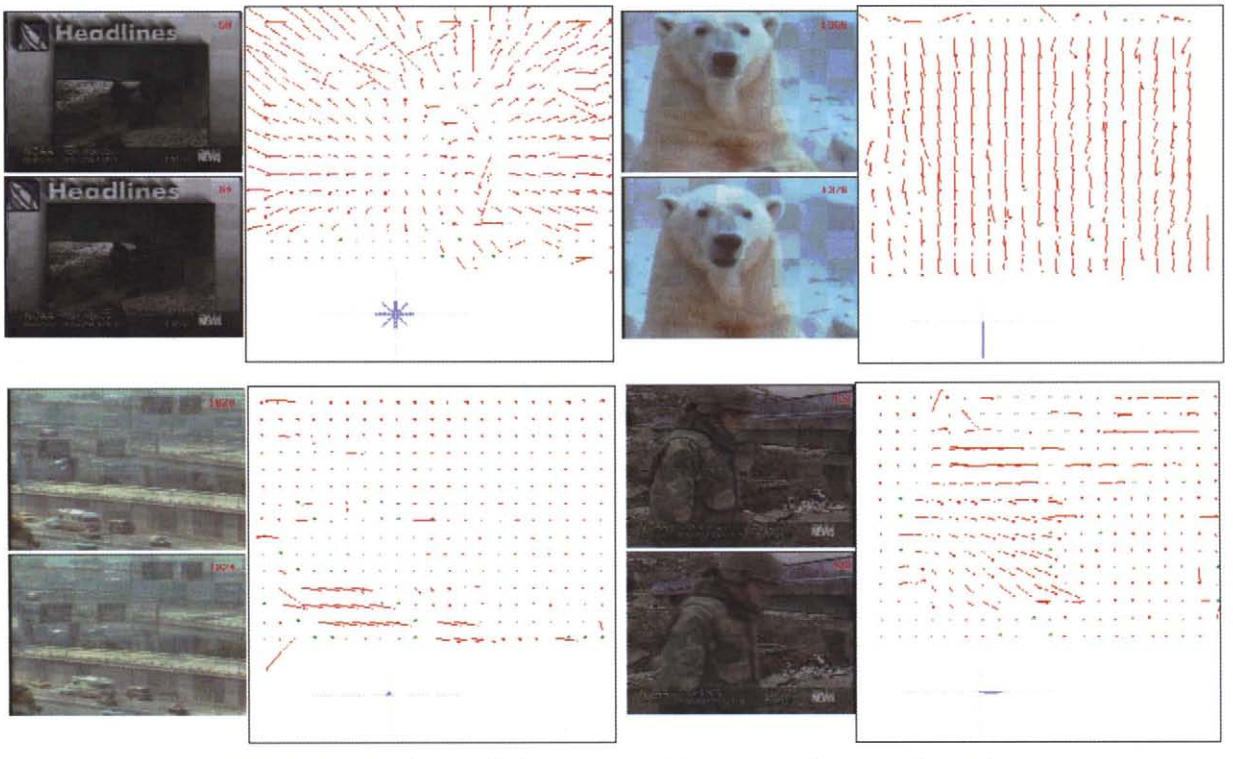

FIGURE 9.1.3 Optical flow fields for a pan (top right), zoom (top left), and object motion.

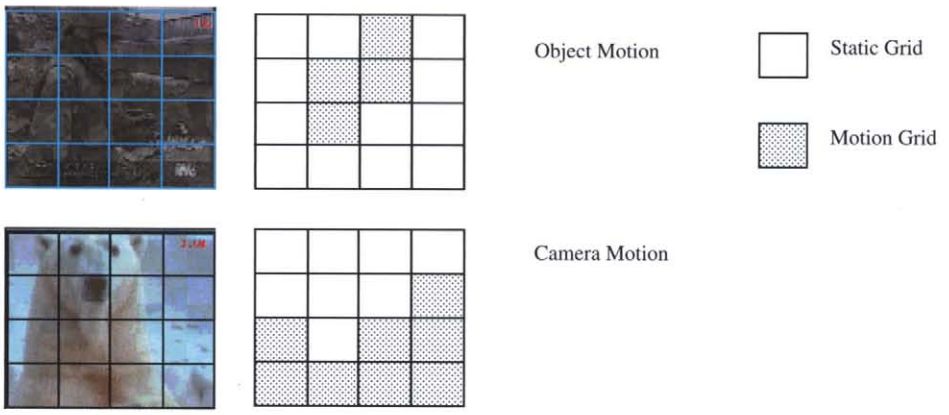

FIGURE 9.1.4 Camera and object motion detection.

FIGURE 9.1.5 Images with similar shapes (human face and torso).

FIGURE 9.1.6 Recognition of captions and faces [11].

FIGURE 9.1.8 Graphics detection through sub-region histogram differencing.

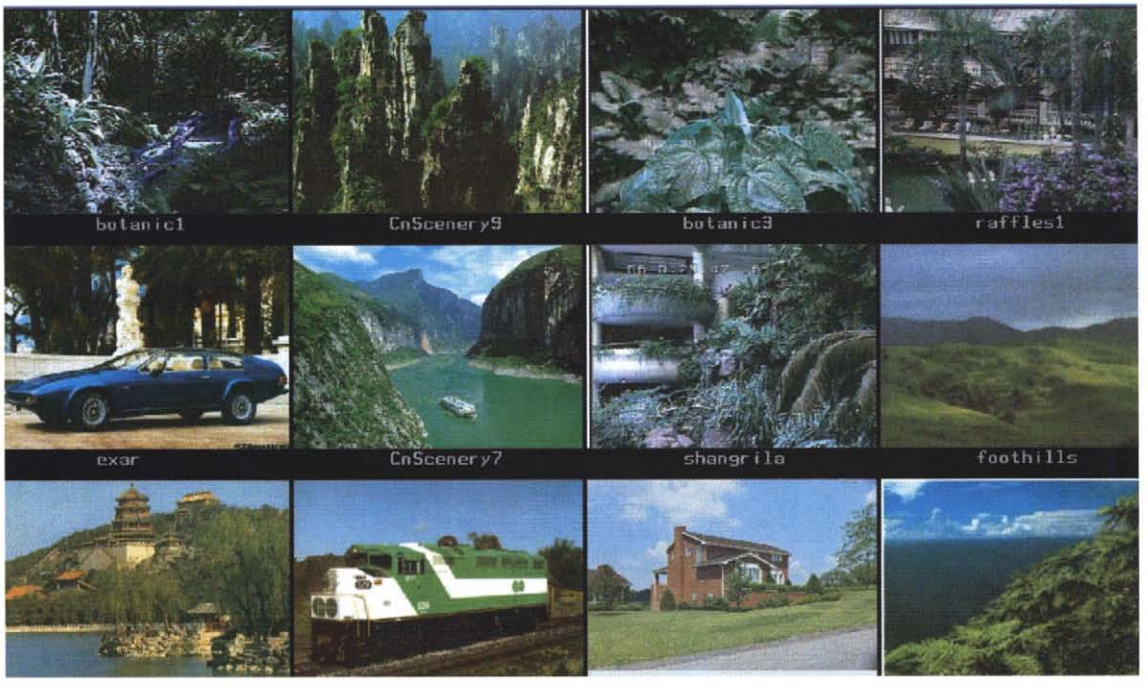

FIGURE 9.1.9 Images with similar color.

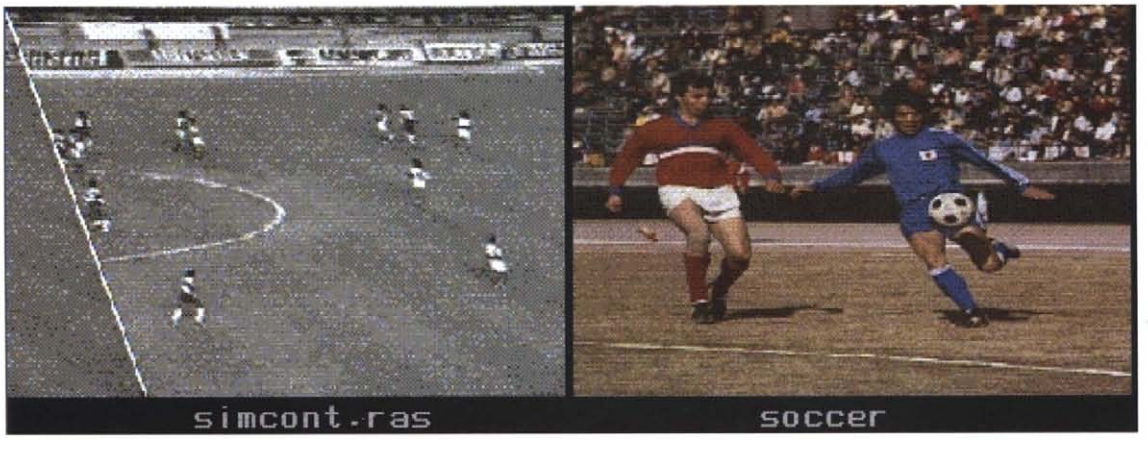

FIGURE 9.1.10 Images with similar content.

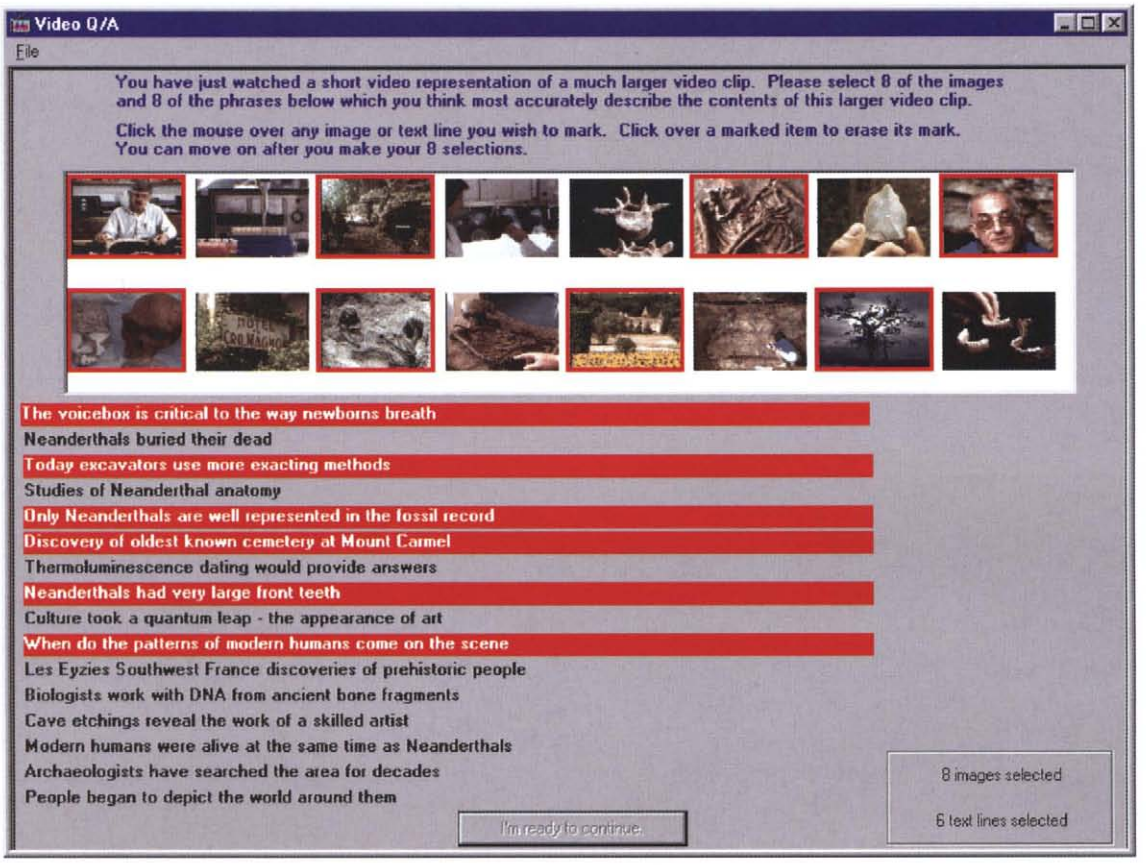

FIGURE 9.1.11 Video retrieval user-study interface [3].

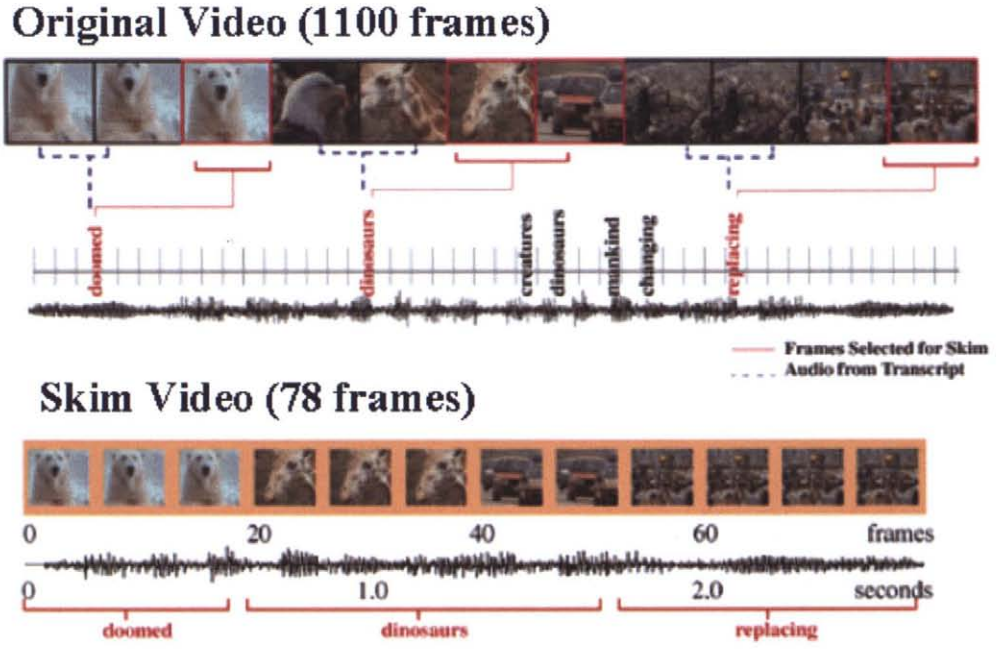

FIGURE 9.1.12 Illustration of video skimming.