14.1 Introduction

With rapidly growing collections of images, news programs, music videos, movies, digital television programs, and training and education videos on the Internet and corporate intranets, new tools are needed to harness digital media for different applications, ranging from image and video cataloging, to media archival and search, multimedia authoring and synthesis, and smart browsing. In recent years, we have witnessed the growing momentum in building systems that can query and search video collections efficiently and accurately for desired video segments, just in the manner text search engines on the web have enabled easy retrieval of documents containing a required piece of text located on a server anywhere in the world. Digital video archival and management systems are also important to postproduction houses, broadcast studios, stock footage houses, and advertising agencies working with large videotape and multimedia collections, to enable integration of content in their end-to-end business processes. Further, since the digital form of videos enables rapid content editing, manipulation, and synthesis, there is burgeoning interest in building cheap, personal desktop video production tools.

An image and video content management system must allow archival, processing, editing, manipulation, browsing, and search and retrieval of image and video data for content repurposing, new program production, and other multimedia interactive services. Annotating or describing images and videos manually through a preview of the material is extremely time consuming, expensive, and unscalable. A content management system, for example, in a digital television studio serves different sets of users, including the program producer who needs to locate material from the studio archive, the writer who
needs to write a story about the airing segment, the editor who needs to edit in the desired clip, the librarian who adds and manages new material in the archive, and the logger who annotates the material in terms of its metadata such as medium ID, production details, and other pertinent information about the content that allows it to be located easily. Therefore, a content management tool must be scalable and highly available, ensure integrity of content, and enable easy and quick retrieval of archived material for content reuse and distribution. Automatic extraction of image and video content descriptions is highly desirable to ease the pain of manual annotation and to provide a consistent language of content description when annotating large video collections.

In order to answer user queries during media search, it is crucial to define a suitable representation for the media, their metadata, and the operations to be applied to them. The aim of a data model is to introduce an abstraction between the physical level (data files and indices) and the conceptual representation, together with some operations to manipulate the data. The conceptual representation corresponds to the conceptual level of the American National Standards Institute (ANSI) relational database architecture [106], where algebraic optimization and algorithm selections are performed. Optimization at the physical level (data files and indices) consists of defining indices and selecting the right access methods to be used in query processing.

This chapter surveys techniques used to extract descriptions of multimedia data (mainly image and video) through automated analysis and current database solutions in managing, indexing, and querying of multimedia data. The chapter is divided into two parts: multimedia data analysis and database techniques for multimedia. The multimedia data analysis part comprises Section 14.2, which presents common features used in image databases and the techniques to extract them automatically, and Section 14.3, which discusses audio and video content analysis, extraction of audiovisual descriptors, as well as the fusion of multiple media modalities. The second part comprises Section 14.4, which presents an example of database models defined for multimedia data, and Section 14.5, which discusses similarity search in multimedia. Finally, Section 14.6 concludes the chapter.

### 14.2 Image Content Analysis

Existing work in image content analysis could be coarsely categorized into two groups based on the features employed. The first group indexes an image based on low-level features such as color, texture, and shape, while the second group attempts to understand the semantic content of images by using mid- to high-level features, as well as by applying more complex analysis models. This section surveys representative work in both groups.

#### 14.2.1 Low-Level Image Content Analysis

Research in this area proposes to index images based on low-level features, which are easy to extract and fast to implement. Some well-known content-based image retrieval (CBIR) systems such as query by image content (QBIC) [39], multimedia analysis and retrieval system (MARS) [54], WebSEEK [119], and Photobook [102] have employed these features for image indexing, browsing, and retrieval, with reasonable performance achieved. However, due to the low-level nature of these features, there still exists a gap between the information revealed by these features and the real image semantics. Obviously, more high-level features are needed to truly understand the image content. In the following, we review some commonly used image features, including color, texture, and shape.

##### 14.2.1.1 Color

Color is one of the most recognizable elements of image content. It is widely used as a feature for image retrieval due to its invariance to image scaling, translation, and rotation. Key issues in color feature extraction include the selection of color space and the choice of color quantization scheme.

A color space is a multidimensional space in which different dimensions represent different color components. Theoretically, any color can be represented by a linear combination of the three primary colors.
Multimedia Databases

14-3

colors, red (R), green (G), and blue (B). However, the RGB color space is not perceptually uniform, and equal distances in different areas and along different dimensions of this space do not correspond to equal perception of color dissimilarity. Therefore, other color spaces, such as CIELAB and CIEL\*u*\*v*, have been proposed. Other widely used color spaces include YCbCr, YIQ, YUV, HSV, and Munsell spaces. Readers are referred to [20] for more detailed descriptions on color spaces. The MPEG-7 standard [85], formally known as the “multimedia content description interface,” has adopted the RGB, YCbCr, and HSV color spaces, as well as linear transformation matrices with reference to RGB.

Quantization is used to reduce the color resolution of an image. Since a 24 bit color space contains 2^{24} distinct colors, using a quantized map can considerably decrease the computational complexity of color feature extraction. The most commonly used color quantization schemes are uniform quantization, vector quantization, tree-structured vector quantization, and product quantization. MPEG-7 supports linear and nonlinear quantization, as well as quantization via lookup tables.

Three widely used color features for images (also called color descriptors in MPEG-7) are global color histogram, local color histogram, and dominant color. The global color histogram captures the color content of the entire image while ignoring information on spatial layout. Specifically, a global color histogram represents an image \( I \) by an \( N \)-dimensional vector \( H(I) = [H(I,j), j = 1, 2, ..., N] \), where \( N \) is the total number of quantized colors and \( H(I,j) \) is the number of pixels having color \( j \).

In contrast, the local color histogram representation considers the position and size of each individual image region so as to describe the spatial structure of the image color. For instance, Stricker and Dimai [123] segmented each image into five nonoverlapping spatial regions, from which color features were extracted and subsequently used for image matching. In [96], a scalable blob histogram was proposed, where the term blob denotes a group of pixels with a homogeneous color. This descriptor is able to distinguish images, containing objects of different sizes and shapes, without performing color segmentation.

The dominant color representation is considered to be one of the major color descriptors due to its simplicity and its association with human perception. Various algorithms have been proposed to extract this feature. For example, Ohm and Makai took the means of color clusters (the cluster centroids) as dominant colors of images [97]. Considering that human eyes are more sensitive to changes in smooth regions than changes in detailed ones, Deng et al. proposed the extraction of dominant colors by hierarchically merging similar and unimportant color clusters while leaving distinct and more important clusters untouched [34].

Other commonly used color feature representations include color correlogram [52] and color moments [123]. Specifically, the color correlogram captures the local spatial correlation of pairs of colors and is a second-order statistic on the color distribution. It is rotation, scale, and (to some extent) viewpoint invariant. Moreover, it is designed to tolerate moderate changes in appearance and shape due to viewpoint changes, camera zoom, noise, and compression. Therefore, color correlogram representations have been widely used in video copy detection [92] and video concept modeling and detection [140]. On the other hand, color moments are mainly adopted to overcome undesirable color quantization effects and are defined as the moments of the distribution formed by RGB triplets in a given image. Note that by using a proper combination of moments it is possible to normalize against photometric changes. Such combinations are called color moment invariants. Color moments and color moment invariants have been applied to applications such as CBIR.

14.2.1.2 Texture

Texture refers to visual patterns with properties of homogeneity that do not result from the presence of only a single color or intensity. Tree bark, clouds, water, bricks, and fabrics are some examples of textures. Typical texture features include contrast, uniformity, coarseness, roughness, frequency, density, and directionality, which contain important information about the structural arrangement of surfaces as well as their relationship to the surrounding environment. So far, much research effort has been devoted to texture analysis due to its usefulness and effectiveness in applications such as pattern recognition, computer vision, and image retrieval.
There are two basic types of texture descriptors: statistical and transform based. The statistical approach explores the gray-level spatial dependence of textures and extracts meaningful statistics as the texture representation. For example, Haralick et al. [47] proposed the representation of textures using co-occurrence matrices that capture gray-level spatial dependence. They also performed a statistical analysis of the spatial relationships of lines, as well as the properties of their surroundings. Interestingly, Tamura et al. addressed this topic from a totally different viewpoint [125]: based on psychological measurements, they claimed that the six basic textural features should be coarseness, contrast, directionality, line-likeness, regularity, and roughness. Two well-known CBIR systems, QBIC and MARS, have adopted this representation. Other texture representations may involve a subset of the aforementioned six features, such as contrast, coarseness, and directionality.

Two other commonly used statistical texture features are edge histograms [85] and histograms of oriented gradient (HOG) [29]. Edge histograms describe the distribution of edges in terms of both frequency and directionality of the brightness gradients within an image. HOG extends the edge orientation histogram or histogram of gradient directions by means of computation over a dense grid of uniformly spaced cells. Furthermore, improved performance is often achieved within overlapping local contrast normalization. Edge histograms and HOG features have been widely applied in image and video object detection and recognition [45].

In recent years, local binary pattern (LBP) feature has gradually gained in popularity, especially in computer vision applications. Specifically, the LBP operator is defined as a grayscale invariant texture measure, derived from a general definition of texture in a local neighborhood. Its properties of invariance under monotonic gray-level changes, together with its computational simplicity, have made it a very good feature for applications such as facial expression recognition [151].

Commonly used transforms for texture extraction include the discrete cosine transform (DCT), the Fourier–Mellin transform, the polar Fourier transform, the Gabor transform, and the wavelet transform. Alata et al. [5] proposed the classification of rotated and scaled textures using a combination of the Fourier–Mellin transform and a parametric 2D spectrum estimation method (Harmonic Mean Horizontal Vertical). In [130], Wan and Kuo reported their work on texture feature extraction for Joint Photographic Experts Group (JPEG) images based on the analysis of DCT–AC coefficients. Chang and Kuo [23] presented a tree-structured wavelet transform, which provided a natural and effective way to describe textures that have dominant middle- to high-frequency subbands. Readers are referred to [20] for a detailed description on texture feature extraction.

Building upon Gabor features, Oliva and Torralba proposed the GIST feature for the characterization of scenes [98]. GIST features concatenate spatially pooled Gabor filter responses in different scales and orientations in different spatial blocks of the image. They have been successfully applied to directly represent, understand, and recognize image scenes, bypassing the segmentation and processing of individual objects or regions. Natseve et al. have also successfully applied GIST (along with other features) for video copy detection and video event detection in their text retrieval conference—video retrieval evaluation (TRECVID) 2010 effort [92].

14.2.1.3 Shape

Compared to color and texture, the representation of shape is inherently much more complex. Two major steps are required to extract a shape feature: object segmentation and shape representation.

Although object segmentation has been studied for decades, it remains a very difficult research topic in computer vision. Existing image segmentation techniques include the global threshold-based approach, the region growing approach, the split and merge approach, the edge-detection-based approach, the color- and texture-based approach, and the model-based approach. Generally speaking, it is difficult to achieve perfect segmentation due to the complexity of individual object shapes, as well as the existence of shadows and noise.

Existing shape representation approaches can be categorized into the following three classes: boundary-based representation, region-based representation, and their combination. Boundary-based
representations emphasize the closed curve that surrounds the shape. Numerous models have been proposed to describe this curve, including chain codes [42], polygons [28], circular arcs [38], splines [33], explicit and implicit polynomials [61], boundary Fourier descriptors, and Universidade Nova de Lisboa (UNL) descriptors [16]. Region-based representations, on the other hand, emphasize the area within the closed boundary. Various descriptors have been proposed to model interior regions, such as moment invariants [77], Zernike moments [62], morphological descriptor [81], and pseudo-Zernike moments. Generally speaking, region-based moments are invariant to image’s affine transformations. Readers are referred to [147] for more details.

Recent work in shape representation includes the finite element method (FEM), the turning function, and the wavelet descriptor. Besides the earlier work in 2D shape representation, research effort has been devoted to 3D shape representation. Readers are referred to [20] for more detailed discussions on shape features.

14.2.2 Key Points and Bag of Visual Words

14.2.2.1 Key Point Extraction

Key points (or points of interest) are used to describe local features of images. The most popular technique is the scale invariant feature transform (SIFT) proposed by Lowe in 2004 [82]. SIFT is designed to be scale, rotation, and shift invariant so that it can be reliably applied to object recognition and image matching. The four basic steps in extracting SIFT features include scale-space extrema detection, key point localization, orientation assignment, and key point descriptor formation.

Based upon SIFT, several other variants have been subsequently proposed, including the speeded up robust features (SURFs) [10], the principal component analysis-based representation of SIFT features (PCA-SIFT) [60], and the rotation-invariant feature transform (RIFT) [66]. SURF is a high-performance scale- and rotation-invariant interest point detector/descriptor, which has been claimed to approximate or even outperform SIFT with respect to repeatability, distinctiveness, and robustness. PCA-SIFT differs from SIFT in terms of calculation of the image gradients and the definition of gradient regions. It also applies PCA to further reduce the feature dimension. Finally, RIFT is a rotation-invariant generalization of SIFT, constructed using circular normalized patches, which are divided into concentric rings of equal width. A gradient orientation histogram is then computed within each ring.

Key point extraction methods such SIFT represent an image as a large collection of feature vectors, each of which describes a specific key point in the image. Visual words are obtained by grouping similar key points into clusters. Hence, a visual word represents a specific visual pattern and the set of visual words represent a visual word vocabulary. An image can then be represented as a collection (or “bag”) of visual words. The entire process from the key point extraction to the final visual word vectors is illustrated by Figure 14.1.

14.2.3 Mid- to High-Level Image Content Analysis

Research in this area attempts to index images based on semantics content in the form of salient image objects. To achieve this goal, various mid- to high-level image features, as well as more complex analysis models, have been proposed. An example reported in [34] uses a low-dimensional color indexing scheme based on homogeneous image regions. After first applying a color segmentation method (JSEG) to obtain homogeneous regions, the colors within each region are quantized and grouped into a small number of clusters. Finally, color centroids together with their relative weights were used as feature descriptors.

Some approaches attempt to understand image content by learning its semantic concepts. For example, Minka and Picard [88] developed a system that first generated segmentations or groups of image
regions using various feature combinations. The system was later improved by learning from the user input to decide which combinations best represent predetermined semantic categories. The system, however, requires supervised training over various parts of the image. In contrast, Li et al. proposed the detection of salient image regions based on segmented color and orientation maps without any human intervention [74].

Targeting the construction of a moderately large lexicon of semantic concepts, Naphade et al. proposed a support vector machine (SVM) learning system for detecting 34 visual concepts, including 15 scene concepts (e.g., outdoors, indoors, landscape, cityscape, sky, beach, mountain, and land) and 19 object concepts (e.g., face, people, road, building, tree, animal, text overlay, and train) [91]. Using the TREC 2002 benchmark corpus for training and validation, this system has achieved reasonable performance with moderately large training samples.
With the proliferation of photo sharing services and general purpose sites with photo sharing capabilities, such as Flickr and Facebook, automated image annotation and tagging have become a problem of great interest [35,107,115,129]. In fact, image annotation has been extensively studied by several research communities, including the image retrieval, computer vision community, multimedia and web, and human–computer interface communities. Several benchmark evaluation campaigns such as Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) and TRECVID have been launched to encourage and evaluate algorithms and systems developed along this line [56,117,128]. In [50], Houle et al. proposed a knowledge propagation algorithm as a neighborhood-based approach to effectively propagate knowledge associated to a few objects of interest to an entire image database. In [114,121], an ontology is utilized for image annotation. The keywords are organized in a hierarchical structure to help remove semantic ambiguities. Photo sharing services such as Facebook and Flickr contain community-contributed media collections, and applying social knowledge can help rerank, filter, or expand image tags. Both [43] and [115] were Flickr image tag recommendation systems developed based on tag concurrence and intertag aggregation. Liu et al. further incorporated image similarity into tag co-occurrence and evaluated the system with subjective labeling of tag usefulness [79]. In [135], an end-to-end image tagging system was presented along with a detailed user evaluation of the accuracy and value of the machine-generated image tags. Several important issues in building an image tagging system were addressed, including the design of tagging vocabulary, taxonomy-based tag refinement, classifier score calibration for effective tag ranking, and selection of valuable tags as opposed to accurate ones.

Finally, there is a recent trend to adopt a new computing paradigm, known as data-intensive scalable computing (DISC), to keep up with the increasingly high-volume multimedia modeling and analysis. DISC proposes the use of a data parallel approach to processing Big Data, which are large volumes of data in the order of terabytes or petabytes [19]. DISC has defined a new research direction, and with the emergence of Big Data, research effort has been devoted to large-scale data processing. For example, in [140], Yan et al. proposed an algorithm called robust subspace bagging (RB-SBag) for large-scale semantic concept modeling, along with its MapReduce implementation (MapReduce [30–32] was proposed by Google as a leading example of DISC). RB-SBag combines both random subspace bagging and forward model selection into a unified approach. Evaluations conducted on more than 250K images and several standard TRECVID benchmark data sets have showed that the RB-SBag approach can achieve a more than ten-fold speedup over baseline SVM-based approaches.

### 14.3 Video Content Analysis

Video content analysis, which consists of visual content analysis, audio content analysis, as well as multimodality analysis, has attracted an enormous interest in both academic and corporate research communities. This research appeal, in turn, encourages areas that are primarily built upon content analysis modules such as video abstraction, video browsing, and video retrieval to be actively developed. In this section, a comprehensive survey of these research topics will be presented.

#### 14.3.1 Visual Content Analysis

The first step in video content analysis is to extract its content structure, which could be represented by a hierarchical tree exemplified in Figure 14.2 [72]. As shown, given a continuous video bitstream, we first segment it into a series of cascaded video shots, where a shot contains a set of contiguously recorded image frames. Because the content within a shot is always continuous, in most cases, one or more frames (known as key frames) can be extracted to represent its underlying content. However, while the shot forms the building block of a video sequence, this low-level structure does not directly correspond to the video semantics. Moreover, this processing often leads to a far too fine segmentation of the video data. That risks the obscuration of its semantic meaning.
Most recent work attempts to understand the video semantics through the extraction of the underlying video scenes, where a scene is defined as a collection of semantically related and temporally adjacent shots that depicts and conveys a high-level concept or story. A common approach to video scene extraction is to group semantically related shots into a scene.

Nevertheless, not every scene can be associated with a meaningful theme. For example, in feature films, there are certain scenes that are used only to establish story environment without involving any thematic topics. Therefore, it is necessary to find important scenes that contain specific thematic topics, such as dialogue or sports highlight. In the area of surveillance video analysis, users are likely to be interested only in video segments that contain specific or abnormal patterns or behaviors. In this chapter, such video units or segments are referred to as events.

Finally, it is also important to automatically annotate videos or detect specific video concepts, so as to better understand video content and facilitate video browsing, search, and navigation at a semantic level. This section reviews previous work on the detection of video shots, scenes, and events, as well as video annotation.

### 14.3.1.1 Video Shot Detection

A shot can be detected by identifying the camera transitions, which may be either abrupt or gradual. Abrupt transition, also called a camera break or cut, occurs when the content changes significantly between two consecutive frames. In contrast, a gradual transition is usually associated with special effects such as a dissolve, wipe, fade-in, and fade-out, where a smooth change in content is observed over a consecutive sequence of frames.

Existing work in shot detection can be generally categorized into the following five classes: pixel based, histogram based, feature based, statistics based, and transform based. The pixel-based approach detects the shot change by counting the number of pixels, which have changed from one frame to the next. However, while this approach may be the simplest way to detect the content change between two frames, a pixel-based method may be too sensitive to object and camera motion. Consequently, the histogram-based approach, which detects content change by comparing the histograms of neighboring frames, has gained more popularity as histograms are invariant to image rotation, scaling, and transition. In fact, it has been reported that histogram methods can achieve good trade-offs between the accuracy and speed [78].

The feature-based approach applies features such as intensity edge [146] and visual rhythm [26] to identify the shot boundary. Other technologies such as image segmentation and object tracking have also been employed. On the other hand, information such as mean and standard deviation of pixel intensities are exploited by statistics-based approach. Finally, transform-based methods use hidden Markov model (HMM) or Markov random field (MRF) to model shot transitions.
The topic of shot detection has matured in recent years, as the current research literature contains approaches that can perform reasonably well on real-world applications [145].

14.3.1.2 Video Scene and Event Detection

Existing scene detection approaches can be classified into two: model-based and model-free. With the former category, specific structure models are constructed for up specific video applications by exploiting scene characteristics, discernible logos, or marks. For example, in [148], temporal and spatial structures were defined for the parsing of TV news, where the temporal structure was modeled by a series of shots, including anchorperson shots, news shots, commercial break shots, and weather forecast shots. Meanwhile, the spatial structure was modeled by four frame templates, with each containing either two anchorpersons, one anchorperson, one anchorperson with an upper-right news icon, or one anchorperson with an upper-left news icon. Some other work in this direction has tried to integrate multiple media cues such as visual, audio, and text (closed captions or audio transcripts) to extract scenes from real TV programs.

Compared to the model-based approach, which has very limited applications, the model-free approach can be applied to very generic situations. Work in this area can be categorized into three classes according to the use of visual, audio, or audiovisual cues. In visual-based approaches, color or motion information is utilized to locate the scene boundary. For example, Yeung et al. proposed to detect scenes by grouping visually similar and temporally close shots [142]. Moreover, they also constructed a transition graph to represent the detected scene structure. Compressed video sequences were used in their experiments. Other work in this area has involved the cophenetic dissimilarity criterion or a set of heuristic rules to determine the scene boundary.

Pure audio-based work was reported in [150], where the original video was segmented into a sequence of audio scenes such as speech, silence, music, speech with music, song, and environmental sound based on low-level audio features. In [89], sound tracks in films and their indexical semiotic usage were studied based on an audio classification system that could detect complex sound scenes as well as the constituent sound events in cinema. Specifically, it has studied the car chase and the violence scenes for action movies based on the detection of their characteristic sound events such as horns, sirens, car crashes, tires skidding, glass breaking, explosions, and gunshots.

However, due to the difficulty of precisely locating the scene boundaries based on audio cues alone, more recent research has concentrated on the integration of multiple media modalities for more robust results [37, 53, 120]. For example, three types of media cues including audio, visual, and motion were employed by [53] to extract semantic video scenes from broadcast news. They considered two types of scenes, N-type (normal scenes) characterized by a long-term consistency of chromatic composition, lighting conditions, and sound and M-type (montage scenes) characterized by visual features for such attributes as lighting conditions, location, and time of creation. N-type scenes were further classified into pure dialogue scenes (with a simple repetitive visual structure among the shots), progressive scenes (linear progression of visuals without repetitive structure), and hybrid scenes (dialogue embedded in a progressive scene) [124]. An integration of audio and visual cues was reported in [73], where audio cues including ambient noise, background music, and speech were cooperatively evaluated with visual feature extraction in order to precisely locate the scene boundary. Special movie editing patterns were also considered in this work.

Compared to that of a scene, the concept of an event is more subjectively defined as it can take on different meanings for different applications many assign. For example, an event could be the highlight of a sports video or an interesting topic in a video document. In [84], a query-driven approach was presented for the detection of topics of discussion by using image and text contents of query foil presentation. While multiple media sources were integrated in their framework, identification results were mainly evaluated in the domain of classroom presentations due to the special features adopted. In contrast, work on sports highlight extraction mainly focuses on detecting the announcer’s speech, the audience ambient speech noise, the game-specific sounds (e.g., the sounds of batting in baseball), and various other background
noises (e.g., the cheering of audiences). Targeting the analysis of movie content, Li et al. proposed the
detection of three types of events, namely, 2-speaker dialogues, multispeaker dialogues, and hybrid
events, by exploiting multiple media cues and special movie production rules [75,95].

In the computer vision community, event detection usually refers to abnormal event detection or
abnormal object activity detection/recognition in surveillance videos. For example, in [8], Basharat et al.
first built the trajectories of objects by tracking them using an appearance model and then, based on the
measurement of the abnormality of each trajectory, detecting anomalous behaviors such as unusual path
taken by pedestrians or moving object with abnormal speed and size. Similar ideas were also proposed in
[104], where a nonparametric object trajectory representation was employed to detect abnormal events.
In [63], a space–time MRF model was proposed to detect abnormal activities in videos. The nodes in the
MRF graph corresponded to a grid of local regions in the video frames, and neighboring nodes in both
space and time were associated with links. Experimental results on several surveillance videos showed
that the MRF model could not only localize atomic abnormal activities in a busy video scene but also
capture global-level abnormalities caused by irregular interactions between local activities. Also aiming
at modeling both spatial and temporal activities, Xiang and Gong developed an HMM to incrementally
and adaptively detect abnormal object behaviors [134].

In contrast with the detection of moving objects, Tian et al. proposed a framework for the robust effi-
cient detection of abandoned and removed objects in complex surveillance videos based on background
subtraction and foreground analysis complemented by tracking [127]. Three Gaussian mixture models
(GMMs) were applied to model the background, as well as to detect static foreground regions. Context
information of the foreground masks was also exploited to help determine the type of static regions.
Finally, a person-detection process was integrated to distinguish static objects from stationary people.
The proposed method has been tested on a large set of real-world surveillance videos and has demonstra-
ted its robustness to quick changes in lighting and occlusions in complex environments.

A very comprehensive and scalable object and event detection, indexing, and retrieval system, the IBM
Smart Surveillance Solution, was presented in [41]. The system allows user to define criteria for alerts with
reference to a specific camera view. The criteria can be parked car detection, trip wire, etc. Moreover, it
automatically generates descriptions for the events that occur in the scene and stores them in an indexed
database to allow users to perform rapid search. Analytics supported by this system include moving object
(such as people and vehicle) detection and tracking, object classification with calibration, object behavior
analysis, color classification, and license plate recognition. The system has been successfully deployed in
several large cities such as Chicago and New York City for public safety monitoring in urban environments.

In recent years, with the proliferation of videos on social media sites such as YouTube, detecting real-
world events from user-uploaded videos and monitoring and tracking their usage and life cycle have
attracted increasing interests from researchers in both multimedia analysis and social media mining com-
munities. Cha et al. conducted a large-scale YouTube measurement study to characterize content category
distributions, as well as to track exact duplicates of popular videos [22]. In [136], Xie et al. developed a
large-scale event monitoring system for YouTube videos based on proposed visual memes. A meme is a
short video segment frequently remixed and reposted by different authors. Once memes were extracted
from tens of thousands of YouTube videos, the authors built a graph model to model the interactions
among them, with people and content as nodes and meme postings as links. Then using this model they
were able to derive graph metrics that capture content influence and user roles. The authors observed that
over half of the event videos on YouTube contained remixed content, that about 70% of video creators
participated in video remixing, and that over 50% of memes were discovered and reposted within 3 h after
their first appearance.

14.3.1.3 Video Annotation

The automatic annotation of videos at the semantic concept level has recently emerged as a very active
topic within the multimedia research community. Video annotation entails a multilabeling process
whereby a video clip is associated with one or more labels. For example, a video clip can be tagged
as “garden,” “plant,” “flower,” and “lake” simultaneously. The concepts of interest in this case usually include a wide range of categories describing scenes, objects, events, and certain named entities.

Current research on video annotation follows three general paradigms: individual-concept detection, context-based concept fusion (CBCF), and integrated multilabeling [105]. Research involving the first paradigm generally uses binary classification to detect each individual concept, thus treating a collection of concepts as isolated points. A typical approach in this case is to build a binary detector for each concept and create either a presence or absence label based on video content modeling and analysis. An alternative is to stack a set of binary detectors into a single discriminative classifier [113]. The left branch in Figure 14.3 illustrates a typical process flow.

Solutions proposed according to the individual-concept detection paradigm have only achieved limited success, as they do not model the inherent correlations between real-world concepts. For example, the presence of “sky” often occurs together with the presence of “cloud,” whereas “ocean” normally does not co-occur with “truck.” Furthermore, while simple concepts can be directly modeled from low-level features, it could be rather difficult to learn individual models for such complex concepts as “political protest.” Instead, complex concepts could be inferred from other more primitive concepts based on correlation. For example, if the concepts of “people,” “walking,” and “banners” are detected, then they likely indicate the presence of the concept “political protest.”

In contrast, studies of context-based concept fusion add a second step after individual-concept detection by fusing multiple concepts together. A CBCF-based active learning approach was proposed in [123] where user annotations were solicited for a small number of concepts from which the remainder of the concept could be inferred. In [132], Wu et al. proposed an ontology-based multiclassification learning method or video concept detection. Each concept was first independently modeled by a classifier, and then a predefined ontology hierarchy was applied to improve the detection accuracy of individual classifiers. An alternative fusion strategy was proposed in [49], where logistic regression (LR) was applied to fuse individual detections. In [118], Smith et al. described a two-step discriminative model fusion (DMF) approach to derive unknown or indirect relationship among concepts by constructing model vectors based on detection scores of individual classifiers. An SVM was then trained to refine the detection results of these classifiers. The typical flow of this process is illustrated by the center branch in Figure 14.3.

While it is intuitive to exploit the contextual relationship among concepts to improve their detection accuracy, CBCF approaches do not always provide stable performance improvement. The reasons are twofold. First, detection errors in individual classifiers can be propagated to the fusion step, thereby degrading the overall performance. Second, the training data for the conceptual fusion may be insufficient, leading to overfitting. Models learned in this way usually lack generalization capability.

Recently, Qi et al. proposed a third paradigm that simultaneously classifies concepts and models their correlations using a so-called correlative multilabel (CML) framework [105]. This approach, illustrated by the right-most branch (Figure 14.3), follows the principle of least commitment [86] in the sense that both learning and optimization are performed in a single step, thus avoiding the error propagation problem of CBCF. Moreover, since the entire training data are simultaneously used for modeling both individual concepts and their correlations, the risk of overfitting is significantly reduced. Experiments conducted on the benchmark TRECVID 2005 data sets have demonstrated its superior performance over other state-of-the-art approaches.

Xie et al. proposed probabilistic visual concept trees for modeling large visual semantic taxonomies and demonstrated their effective use in visual concept detection in a limited setup [137]. They considered three salient attributes for a real-world visual taxonomy: concept semantics, image appearance, and data statistics. They proposed a multifaceted concept forest structure within which these attributes are expressed in the form of parent–child relationships, mutually exclusive relationships, and multiple aspect labelings. They then used the proposed visual concept tree model to encode the multifaceted concept forest under uncertainty. The probabilistic visual concept trees were designed to extend the functionalities of several predecessors, such as the Bayes Net and tree-structured taxonomy [138].
Low-level features

Concept model vector

Conceptual fusion

First paradigm: Individual detectors

Second paradigm: CBCF

Third paradigm: Integrated multilabel approach

FIGURE 14.3 Flowcharts for three different paradigms of typical multilabel video annotation methods. (From Qi, G. et al., ACM Multimedia, 1, 17, 2007.)
Multimedia Databases

They are also capable of supporting inference with the junction tree algorithm. Evaluations conducted over 60K web images with a taxonomy of 222 concepts have shown that probabilistic visual concept trees are capable of effective representation of visual semantic taxonomies, including the encoding of hierarchy, mutually exclusion, and multifaceted concept relationships under uncertainty.

14.3.2 Audio Content Analysis

Existing research on content-based audio data analysis is still quite limited and can be categorized into the following two classes.

14.3.2.1 Audio Segmentation and Classification

One basic problem in audio segmentation is the discrimination between speech and music, which are the two most important audio types. In general, a relatively high accuracy can be achieved when distinguishing speech from music, since their respective signals are significantly different in their spectral distributions as well as their temporal change patterns. A general solution is to first extract various audio features such as the average zero-crossing rate (ZCR) and the short-time energy from the signals, then distinguish the two sound types based on the feature values. For example, 13 audio features calculated in time, frequency, and cepstrum domains were employed in [112] for the purpose of classification. The authors also examined and compared several popular classification schemes, including the multi-dimensional Gaussian maximum a posteriori estimator, the GMM, a spatial partitioning scheme based on k-d-trees, and a nearest neighbor classifier.

State-of-the-art classification algorithms are usually based more on sound. For example, Wyse and Smoliar classified audio signals into three types, including “music,” “speech,” and “others” [133]. Music was first detected based on the average length of the interval during which peaks were within a narrow frequency band; speech was then separated out by tracking the pitches. Research in [103] was devoted to analyze the signal’s amplitude, frequency, and pitch. Simulations were also conducted on human audio perception, the results of which were utilized to segment the audio data and to recognize the music component. More recently, Zhang and Kuo presented an extensive feature extraction and classification system for audio content segmentation and classification purposes [149]. Five audio features including energy, average ZCR, fundamental frequency, and spectral peak tracks were extracted to fulfill the task. A two-step audio classification scheme was proposed in [83], where in the first step, speech and nonspeech were discriminated using k-nearest neighbor (K-NN) and line spectral pair vector quantization (LSPVQ) classification schemes. In the second step, the nonspeech signals were further classified into music, environment sounds, and silence using a feature thresholding scheme. In [70], an SVM-based audio classification mechanism was proposed to assist the content analysis of instructional videos such as presentations and lectures.

In [143], Yu et al. proposed the combination of local summarization and multilevel locality-sensitive hashing (LSH) for multivariate audio retrieval. Based on spectral similarity, they divide each audio track into multiple continuously correlated periods of variable length. This removes a significant part of the redundant information but provides support for more compact but yet accurate descriptions. They then compute weighted mean chroma features for each of these periods. Then, by exploiting the characteristics of the content description, they adapted a two-level LSH scheme for efficiently delineating a narrow relevant search region. In [144], Yu et al. proposed MultiProbe histograms as global summaries of audio feature sequences that retain local temporal acoustic correlations by concatenating major bins of adjacent chroma features. They then exploited the order statistics of the MultiProbe histograms to more efficiently organize and probe LSH tables. The resulting approximate retrieval method is faster than exact K-NN retrieval while still achieving useful accurate rates.

14.3.2.2 Audio Analysis for Video Indexing

Five different video classes, including news report, weather report, basketball, football, and advertisement, were distinguished in [80] using both multilayer neural networks (MNNs) and HMMs.
Features such as the silence ratio, the speech ratio, and the subband energy ratio were extracted to fulfill this task. It was shown that while MNN worked well in distinguishing among reports, games, and advertisements, it had difficulty in classifying different types of reports or games. On the contrary, the use of HMM increased the overall accuracy, but it could not achieve good classification of the five video types. In [101], features such as the pitch, the short-time average energy, the band energy ratio, and the pause rate were first extracted from the coded subband of an MPEG audio clip; these features were then integrated to characterize the clip as either silence, music, or dialogue. Another approach to video indexing based on music and speech detection was proposed in [87], where image processing techniques were applied to the spectrogram of the audio signals. The spectral peaks of music were recognized by applying an edge-detection operator and the speech harmonics were detected with a comb filter.

Li et al. [69] proposed to the detection of discussion scenes in instructional videos by identifying scenes that contain narrations and dialogues using on an audio classification scheme.

### 14.3.3 Fusion of Multiple Modalities

Recently, researchers have begun to focus their attention on the analysis of video content by integrating multiple modalities including audio, visual, and textual cues. Multiple media sources usually complement each other to convey the complete semantics of the content. Consequently, by taking all sources into account, we can achieve better understanding of the content.

A general solution for the fusion of multiple modalities is to first perform individual visual, audio, or textual content analysis to obtain separate sets of analysis results and then to combine them based on fusion rules, perhaps under a probabilistic framework. Another popular way of media integration is to employ different information sources at different processing stages. For example, we can first employ visual cues to generate coarse-level results, then introduce audio cues to refine the results. However, when and how to efficiently and effectively integrate multiple media sources still remain an open issue in need of further study.

In [36,71], a system called metadata automated generation for instructional content (MAGIC) was presented, which aimed to assist content authors and course developers in generating metadata for learning materials and information assets, so as to enable wider reuse of these objects across departments and organizations. The MAGIC metadata generation environment consisted of a set of text and video processing tools to recover the underlying narrative structures of learning materials, including text analysis components for extracting titles and keywords, a taxonomy and classification system for automatically assigning documents to specific taxonomy, and audio and video segmentation and analysis components for segmenting learning videos. Various video analysis algorithms and machine-learning techniques have been applied to both visual features extracted from the video track and aural features computed from the audio track. The segments were further annotated with text features extracted from time-stamped closed caption text based on advanced text analysis. Experiments conducted on a midsized set of videos from various Department of Homeland Security agencies have demonstrated the advantage of integrating multiple modalities in video content analysis to assist more intuitive and effective video browsing, search, and authoring.

An atomic topical segment detection framework was proposed in [76] for instructional videos to facilitate topic-based video navigation and offer efficient video authoring. A comprehensive text analysis component was first developed to extract informative text cues such as keyword synonym sets and sentence boundary information from video transcripts. These text cues were then integrated with various audiovisual cues, such as silence/music breaks and speech similarity, to identify topical segments. Encouraging user feedback has been gathered regarding the usefulness of such system.

Aiming at the detection of affective content in movies, Xu et al. exploited cues from multiple modalities to represent emotions and emotional atmosphere [139]. They first generated midlevel representations from low-level features by integrating audio sounds, dialogue, and subtitles with the visual analysis and then applied them to infer the affective content. Experiments carried out on different movie genres have validated the effectiveness of their approach.
A comprehensive study and large-scale tests of web video categorization methods were reported in [141]. Considering that web videos are characterized by a much higher diversity of quality, subject, style, and genres compared with traditional video programs, the authors concluded that multiple modalities should be employed. In addition to the application of low-level visual and audio features, they proposed two new modalities: a semantic modality and a surrounding text modality. The semantic modality was represented by three types of features: concept histogram, visual word vector model, and visual word latent semantic analysis (LSA). Meanwhile, the text modality was represented by titles, descriptions, and tags of web videos. A comprehensive experiment on evaluating the effectiveness of proposed feature representations with three different classifiers, including SVM, GMM, and manifold ranking, was conducted over 11K web videos, and a mean average precision of 0.56 has been achieved.

An earlier work on classifying news videos using multimodal information was reported in [24]. The authors first applied Fisher’s linear discriminant (FLD) to select a smaller number of discriminative features from a large set of features extracted from image, audio, transcript, and speech; they then concatenated the projections into a single synthesized feature vector to represent the scene content. Using SVM as the classifier, they have achieved good performance on anchor and commercial detection in the TREC 2003 video track benchmark. A later development of this work was described in [48], wherein a hybrid approach to improving semantics extraction from news video was proposed. Through extensive experiments, the authors demonstrated the value of careful parameter tuning, exploitation of multiple feature sets and multilingual linguistic resources, applying text retrieval approaches for image features, as well as establishing synergies between multiple concepts through undirected graphical models.

A novel visual feature-based machine-learning framework called IBM multimedia analysis and retrieval system (IMARS) was presented in [6, 93], which aimed to make digital photos and videos searchable through large-scale semantic modeling and classification. IMARS consists of two important components. First, the multimedia analysis engine applies machine-learning techniques to model semantic concepts in images and videos from automatically extracted visual descriptors. It then automatically assigns labels to unseen content to improve its searching, filtering, and categorization capabilities. Second, the multimedia search engine combines content-based, model-based, semantic concept-based, and speech-based text retrieval for more effective image and video searching.

### 14.3.4 Video Abstraction

Video abstraction, as the name implies, generates a short summary of a long video. A video abstract is a sequence of still or moving images, which represent the video essence in a very concise way. Video abstraction is primarily used for video browsing and is an indispensable part of any video indexing and retrieval system.

There are two fundamentally different kinds of video abstracts: still- and moving-image abstracts. The still-image abstract, also known as a static storyboard or video summary, is a small collection of key frames extracted or generated from the underlying video source. The moving-image abstract, also known as a moving storyboard or video skim, consists of a collection of image sequences, as well as the corresponding audio abstract extracted from the original sequence. It is itself a video clip but of considerably shorter length than the original.

There are basically two types of video skim: the summary sequence and the highlight. A summary sequence is used to provide users an impression about the entire video content, while a highlight generally only contains a relatively small number of interesting video segments. A good example of a video highlight is the movie trailer, which only shows a few very attractive scenes without revealing the story’s ending.

Compared to video skimming, video summarization has attracted much more research interest. Based on the way the key frames are extracted, existing work in this area can be categorized into the following three classes: sampling based, shot based, and segment based.
FIGURE 14.4 A mosaic image generated from a panning sequence.

Most of the earlier work on summarization work was sampling based, where key frames were either randomly chosen or uniformly sampled from the original video. Sampling is the simplest way to extract key frames, yet it often fails to adequately represent the video content. More sophisticated methods thus tend to extract key frames by adapting to the dynamic video content using shots. Because a shot is taken within a continuous capture period, a natural and straightforward approach is to extract one or more key frames from each shot. Features such as color and motion have been applied to find the optimal key frames in this case. Mosaics have also been proposed for the representation of shots. One such example is shown in Figure 14.4 [85].

A major drawback of using one or more key frames for each shot is that it does not scale well for long video sequences. Therefore, researchers have proposed to work with an even higher-level video units or segments. A video segment could be a scene, an event, or even the entire sequence.

14.4 Modeling and Querying Multimedia Data

Most research prototypes and applications in multimedia directly organize and search for the vectors obtained from content analysis (see Section 14.5). However, a few database models for multimedia have been proposed in the 1990s and early 2000s. A database model defines the theoretical foundation by which the data are stored and retrieved. The models were proposed as extensions of existing database models (relational, object-oriented, and semistructured models).

14.4.1 Multimedia Database Model as an Extension of Object-Relational Models

Defining a multimedia data model as an extension of the relational or object-relational model is appealing, as the database world is dominated by the relational database model. In [110], an image is stored in a table \( T(h: \text{Integer}, x_1: X_1, \ldots, x_n: X_n) \), where \( h \) is the image identifier and \( x_i \) is an image feature attribute of domain (or type) \( X_i \) (note that classical attributes can be added to this minimal schema). The tuple corresponding the image \( k \) is indicated by \( T[k] \). Each tuple is assigned a real-valued score \( \zeta \) such that \( T[k] \).

\( \zeta \) is a distance between the image \( k \) and the current query image. The value of \( \zeta \) is assigned by a scoring operator \( \Sigma_f(s) \) given a scoring function \( s: \Sigma_f(s)[k] \cdot \zeta = s(T[k] \cdot x_1, \ldots, T[k] \cdot x_n) \).

Since many image queries are based on distance measures, a set of distance functions \((d: X \times X \rightarrow [0,1])\) is defined for each feature type \( X \). Given an element \( x: X \) and a distance function \( d \) defined on \( X \), the scoring function \( s \) assigns \( d(x) \), a distance from \( x \) to every element of \( X \). In addition, a set of score combination operators \( \hat{\circ} : [0,1] \times [0,1] \rightarrow [0,1] \) is defined. New selection and join operators defined on the image table augmented with the scores allow the selection of the \( n \) images with lowest scores, the images whose scores are less than a given score value \( \rho \) and the images from a table \( T = T(h: \text{Integer}, x_1: X_1, \ldots, x_n: X_n) \) that match images from a table \( Q = Q(h: \text{Integer}, y_1: Y_1, \ldots, y_n: Y_n) \), based on score combination functions as follows:

- **K-NNs:** \( \sigma_l^*(\Sigma_f(s)) \) returns the \( k \) rows of the table \( T \) with the lowest distance from the query.
- **Range query operator:** \( \sigma_l^* < \Sigma_f(s) \) returns all the rows of the table \( T \) with a distance less than \( \rho \).
- **\( \hat{\circ} \) Join:** \( T \bowtie Q \) joins the tables \( T \) and \( Q \) on their identifiers \( h \) and returns the table \( W = W(h: \text{Integer}, x_1: X_1, \ldots, x_n: X_n, y_1: Y_1, \ldots, y_n: Y_n) \). The distance in the table \( W \) is defined as \( W \cdot d = T \cdot d \hat{\circ} Q \cdot d \).
In [111], the same authors proposed a design model with four kinds of feature dependencies that can be exploited for the design of efficient search algorithms. For multimedia data models defined as the object-relational model, an extension of SQL called SQL/MM was proposed as a standard query language [122].

### 14.4.2 Multimedia Database Model as Extension of Object-Oriented Models

Multimedia content description is application dependent. The vectors and related similarity functions are defined for specific applications. Because they allow the user to provide methods for specific classes, object-oriented models have been extended to support multimedia data [68,99]. The distributed image database management system (DISIMA) model [99] is an example of a multimedia data model defined as an extension of an object-oriented data model. In the DISIMA model, an image is composed of physical salient objects (regions of the image) whose semantics are given by logical salient objects representing real-world objects. Both images and physical salient objects can have visual properties. The DISIMA model uses an object-oriented concept and introduces three new types: image, physical salient objects, and logical salient objects as well as operators to manipulate them.

Images and related data are manipulated through predicates and operators defined on images; physical and logical salient objects are used to query the images. They can be directly used in calculus-based queries to define formulas or in the definition of algebraic operators. Since the classical predicates \( \{=, <, \leq, >, \geq \} \) are not sufficient for images, a new set of predicates were defined for use on images and salient objects:

- **Contain predicate**: Let \( i \) be an image and \( o \) be an object with a behavior \( pso \) that returns the set of physical salient objects associated with \( o \): \( \text{contains}(i,o) \iff \exists p \in o \cdot pso \land p \in i \cdot pso \).
- **Shape similarity predicates**: Given a shape similarity metric \( d_{\text{shape}} \) and a similarity threshold \( \varepsilon_{\text{shape}} \), two shapes \( s \) and \( t \) are similar with respect to \( d_{\text{shape}} \), if \( d_{\text{shape}}(s,t) \leq \varepsilon_{\text{shape}} \). In other words, \( \text{shape\_similar}(s,t,\varepsilon_{\text{shape}}) \iff d_{\text{shape}}(s,t) \leq \varepsilon_{\text{shape}} \).
- **Color similarity predicates**: Given two color representations \( (c_1,c_2) \) and a color distance metric \( d_{\text{color}} \), the color representations \( c_1 \) and \( c_2 \) are similar with respect to \( d_{\text{color}} \), if \( d_{\text{color}}(c_1,c_2) \leq \varepsilon_{\text{color}} \).

Based on the previously defined predicates, several operators are defined: \( \text{contains} \) or \( \text{semantic join} \) (to check whether a salient object is found in an image), the \( \text{similarity join} \) that is used to match two images or two salient objects with respect to a predefined similarity metric on some low-level features (color, texture, shape, etc.), and \( \text{spatial join} \) on physical salient objects:

- **Semantic join**: Let \( S \) be a set of semantic objects of the same type with a behavior \( pso \) that returns, for a semantic object, the physical salient objects it describes. The \( \text{semantic join} \) between an image class extent \( I \) and the semantic object class extent \( S \), denoted by \( I \bowtie_{\text{contains}} S \), defines the elements of \( I \times S \) where for \( i \in I \) and \( s \in S \), \( \text{contains}(i,s) \) is true.
- **Similarity join**: Given a similarity predicate \( \text{similar} \) and a threshold \( \varepsilon \), the \( \text{similarity join} \) between two sets \( R \) and \( S \) of images or physical salient objects, denoted by \( R \bowtie_{\text{similar}(r,i,s,\varepsilon)} S \) for \( r \in R \) and \( s \in S \), is the set of elements from \( R \times S \) where the behaviors \( i \) defined on the elements of \( R \) and \( j \) on the elements of \( S \) return some compatible metric data type \( T \) and \( \text{similar}(r \cdot i,s \cdot j, \varepsilon) \) (holds true the behaviors \( i \) and \( j \) can be the behaviors that return color, texture, or shape).
- **Spatial join**: The spatial join of the extent of two sets \( R \) and \( S \), denoted by \( R \bowtie_{\theta,\text{inj}} S \), is the set of elements from \( R \times S \) where the behaviors \( i \) defined on the elements of \( R \), and \( j \) on the elements of \( S \), return some spatial data type, \( \theta \) is a binary spatial predicate, and \( R \cdot i \) stands in relation \( \theta \) to \( S \cdot j \) (\( \theta \) is a spatial operator like north, west, northeast, intersect).

The previously defined predicates and the operators are the basis of the declarative query languages MOQL [67] and Visual MOQL [100]. The DISIMA model was later extended to support video data [25]. MOQL is an extension of the Object Query Language (OQL) [21]. Most extensions introduced to OQL by MOQL involve the where clause, in the form of four new predicate expressions:
Technical Foundations of Data and Database Management

Spatial expression, temporal expression, contains predicate, and similarity expression. The spatial expression is a spatial extension that includes spatial objects, spatial functions, and spatial predicates. The temporal expression deals with temporal objects, functions, and predicates for videos. The "contains" predicate is used to specify the salient objects contained in a given media object. The similarity predicate checks if two media objects are similar with respect to a given metric. Visual MOQL uses the DISIMA model to implement the image facilities of MOQL.

14.4.3 Example of Semistructured Model

MPEG-7, the multimedia content description interface, is an ISO metadata standard defined for the description of multimedia data [44]. The MPEG-7 standard aims at helping with searching, filtering, processing, and customizing multimedia data through specifying its features in a “universal” format. MPEG-7 does not specify any applications, but rather the representation format for the information contained within the multimedia data is represented, thereby supporting descriptions of multimedia made using many different formats. The objectives of MPEG-7 include creation of methods to describe multimedia content, manage data flexibly, and globalize data resources.

MPEG-7 essentially provides two tools: the description definition language (MPEG-7 DDL) [1] for the definition of media schemes and an exhaustive set of media description schemes (DSs) mainly for media low-level features. The predefined media DSs are composed of visual feature descriptor schemes [2], audio feature descriptor schemes [3], and general multimedia DSs [4]. Media description through MPEG-7 is achieved through three main elements: descriptors (Ds), DSs, and a DDL. The descriptors essentially describe a feature or distinctive aspect of the multimedia data. An example of a descriptor would be the camera angle used in a video document. The Ds organizes the descriptions, specifying the relationship between descriptors. DSs, for example, would represent how a picture or a movie would be logically ordered. The DDL is used to specify the schemes, as well as to allow modifications and extension to the schemes.

The MPEG-7 DDL is a superset of XML Schema [126], the W3C schema definition language for XML documents. The extensions to XML Schema comprise support for array and matrix data types as well as additional temporal data types. Because MPEG-7 media descriptions are XML documents that conform to the XML Schema definition, it is natural to suggest XML database solutions for the management of MPEG-7 documents, as proposed in [64]. Current XML database solutions are text oriented, and MPEG-7 encodes nontextual data. Directly applying current XML database solutions to MPEG-7 would lower its expressive power, as only textual queries will be allowed.

The models presented in this section are representative of the solutions proposed for multimedia data; however, all have significant limitations. Object or object-relational models assume that multimedia data are well structured, with their content clearly defined. This is not always the case. For example, the content description of an image as it relates to salient objects relies on segmentation and object recognition, both of which are still active research topics. Multimedia content is often represented as a collection of vectors with specific similarity functions, which cannot be represented using the semistructured data models that are commonly applied to text data. Multimedia data are complex data whose representations can involve structured data (multimedia metadata), text (multimedia semantics), as well as vectors (multimedia content). A simple extension of existing data models is not sufficient, especially as the retrieval of multimedia data involves browsing and search as well as classical text-based queries.

14.5 Similarity Search in Multimedia Databases

The aim of a similarity query is to select and return a subset of the database with the greatest resemblance to the query object, all according to a measure of object-to-object similarity or dissimilarity. Data dissimilarity is very commonly measured in terms of distance metrics such as the Euclidean distance, other Lp metrics, the Jaccard distance, or the vector angle distance (equivalent to cosine similarity);
many other measures exist, both metric and nonmetric. The choice of measure generally depends on the types of descriptors used in modeling the data.

Similarity queries are of two main types: range queries and K-NN queries. Given a query object \( q \in \mathcal{U} \), a data set \( S \subseteq \mathcal{U} \), and a distance function \( d: \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_+^* \), where \( \mathcal{U} \) is a data domain, range queries and K-NN queries can be characterized as follows:

- Range queries report the set \( \{ v \in S | d(q,v) \leq \epsilon \} \) for some real value \( \epsilon \geq 0 \).
- K-NN queries report a set of \( k \) objects \( W \subseteq S \) satisfying \( d(q,u) \leq d(q,v) \) for all \( u \in W \) and \( v \in S \setminus W \).

Note that a K-NN query result is not necessarily unique, whereas a range query result is. Of the two types of similarity queries, K-NN queries are arguably more popular, perhaps due to the difficulty faced by the user in deciding a range limit when the similarity values are unknown or when they lack an interpretation meaningful to the user.

Most similarity search strategies seek to reduce the search space directly by means of an index structure. Spatial indices are among the first such structures; a great many examples exist, including k-d-trees [12], B-trees [11], R-trees [46], X-trees [13], SR-trees [59], and A-trees [108]. Such structures explicitly rely on knowledge of the data representation for their organization, which makes them impractical when the representational dimension is very high. Moreover, they usually can handle only queries with respect to Euclidean or other \( L_p \) distance measures. Later, indices were developed for use with general metric data, such as multi-vantage-point trees [17], geometric near-neighbor access trees (GNAT) [18], and \( M \)-trees [27]. Unlike spatial indices, metric indices rely solely on pairwise distance values to make their decisions. Many more examples of spatial and metric data structures can be found in [109].

Both spatial and metric search structures do not in general scale well due to an effect known as the curse of dimensionality, so-called due to the tendency of the query performance of indexing structures to degrade as the data dimensionality increases. With respect to a given query item, as the data dimensionality increases, the distribution of distances from a given query object to the data set objects changes: these distances tend to concentrate more tightly around their mean value, leading to a reduction in the ability of the similarity measure to discriminate between relevant and nonrelevant data objects [14]. Evidence shows that as the data dimension approaches 20, the performance of such search structures as the \( R^* \)-tree, SS-tree, and SR-tree typically degrades to that of a sequential scan of the entire data set [14]. However, the dimensionality of multimedia feature vectors is usually much higher than that which can ordinarily be dealt with using spatial or metric indices. For example, color histograms typically span at least 64 dimensions and sometimes many more. Applications using bag-of-words modeling use vectors of many thousands of dimensions to represent the data [141].

More recently, many other approaches have been developed for both exact and approximate K-NN search, in an attempt to partially circumvent the curse of dimensionality. For exact search, Koudas et al. [65] proposed a method that used a bitmap data representation to facilitate the search process. Jagadish et al. [57] developed iDistance, in which high-dimensional objects are transformed into 1D values, which are then indexed for K-NN search using a \( B^+ \)-tree. Beygelzimer et al. [15] introduced the cover tree index, in which the search is performed by identifying a set of nodes whose descendants are guaranteed to include all nearest neighbors.

For approximate search, Andoni et al. [7] developed the MedScore technique, in which the original high-dimensional feature space is transformed into a low-dimensional feature space by means of random projections. For each dimension in the new feature space, a sorted list is created. The search is done by aggregating these sorted lists to find the objects with the minimum median scores. The well-known threshold algorithm [40] is used here for the aggregation. Indyk and Motwani [55] proposed a popular hashing-based search framework called LSH, in which pairs of similar objects have higher probabilities of mapping to the same hash location than pairs of dissimilar objects. Many variants and improvements upon LSH have subsequently been proposed, such as spectral hashing [131], where the hash bits are determined by thresholding a subset of eigenvectors of the Laplacian of the similarity graph on the data objects.
All of these recently proposed methods attempt to improve upon the performance of their predecessors on high-dimensional data, through the use of domain reduction techniques: via the direct reduction of either the data set (sampling) or attribute set (projection) or via hashing or other data domain transformations. Generally speaking, these methods can cope with data of up to several hundred dimensions before succumbing to the effects of the curse of dimensionality. Further improvements in performance can then be sought through the use of parallelism, as is possible using MapReduce [30], or other forms of distributed processing [9]. Ensembles of similarity indices have also been considered. The fast library for approximate nearest neighbors (FLANN) uses an ensemble of randomized $k$-$d$-tree indices [116] and hierarchical $K$-means trees [94], along with automatic parameter tuning of these indices for optimized performance [90]. It has been shown to achieve very good practical performance for many data sets with up to approximately 200 dimensions.

Recent advances have been made as to the indexability of multimedia data sets of very large scale, in both data set size and data set dimensionality. For data sets of extreme size, a product quantization-based approach for approximate nearest neighbor search has recently been developed that has been successfully applied to the indexing of two billion SIFT vectors of dimension 128, taken from a set of one million images [58]. After decomposing the space into a Cartesian product of low-dimensional subspaces, each subspace is quantized separately. The Euclidean distance between two vectors can be efficiently estimated from short codes composed of the vectors’ quantization indices. Inverted list techniques are then used to assemble a final query result.

For data sets of extreme dimensionality, Houle and Sakuma [51] proposed an approximate search index called a spatial approximation sample hierarchy (SASH), a multilevel structure recursively constructed by building a SASH on a large random data sample and then connecting the remaining objects to several of their approximate nearest neighbors from within the samples. For lower-dimensional data (on the order of 200 dimensions or less), the overheads associated with their sampling strategy generally render the SASH index less competitive than ensemble methods such as FLANN [90] or the heavily optimized product quantization method of [58]. However, the use of near neighbors for distance discrimination can often overcome the effects of the curse of dimensionality for sparse data sets of extremely high dimension (with both the number of vectors and the dimensionality in the millions), as sometimes arise in applications where the data are modeled using the bag-of-words representation.

### 14.6 Conclusion

In this chapter, we have surveyed multimedia databases in terms of data analysis, data querying, and indexing. In multimedia databases, raw multimedia data are unfortunately of limited use due to its large size and lack of interpretability. Consequently, they are usually coupled with descriptive data (including both low-level features and mid- to high-level semantic information) obtained by analyzing the raw media data. Various types of features and concepts related to image, video, and audio data are thus required along with some popular techniques for extracting them. Some latest work on integrating multiple media modalities to better capture media semantics has also been reviewed.

Retrieving a specific image, video, or song that a user has in mind remains a challenging task. Currently, the search relies more on metadata (structured data or textual description) than on the media content. The general problem with images and videos is that their digital representations do not convey any meaningful information about their content. Often to allow semantic search, an interpretation must be added to the raw data, either manually or automatically. Manual annotation of images and video is tedious, and automatic inference of the semantics of images and videos is not always accurate. Multimedia data annotation is still an active research topic, and existing approaches motivate users either to annotate images by simplifying the procedure [35,107,129] or to annotate images in an automatic or semiautomatic way [50,123,132]. In [115], Flickr generates candidate tag lists derived from user-defined tags of each image by computing tag co-occurrence and then recommending new tags by aggregating and ranking the candidates. The ideal solution in which the user does not intervene or manually annotates objects of interest is yet to be proposed.
A related and pressing issue is the lack of a complete data model for multimedia data. Although the idea of extending existing multimedia data models is an appealing one, models defined for classical data cannot effectively handle multimedia data. Metadata such as names, dates, and locations can indeed be stored in a relational database; however, for other types, a relational database does not suffice. For example, the content of an image can also be described using text, and high-dimensional vectors can be described using SIFT descriptors. Across differing real-life applications, different media types are used to describe the same information. This is the case for news applications, where an individual event can be covered by TV channels using video, by radio stations using audio, and by newspapers using text and images. Any successful multimedia data model must be sufficiently general so as to be able to address each of these issues.

References


© 2014 by Taylor & Francis Group, LLC


