

Multimedia Systems: Content Based Indexing and Retrieval

Faisal Bashir, Shashank Khanvilkar, Ashfaq Khokhar, and Dan Schonfeld,
University of Illinois at Chicago

1: Introduction

2: Multimedia Storage and Encoding

2.1: Image Encoding Standards

2.1.1: Digital Compression and Coding of Continuous-Tone Still Images (JPEG)

2.2: Video Encoding Standards

2.2.1: MPEG – Moving Picture Expert Group

2.2.2: H.26X

3: Multimedia Indexing and Retrieval

3.1: Image Indexing and Retrieval

3.1.1: Low-level Feature-based Indexing

3.1.2: Spatial vs. Compressed Domain Processing

3.1.3: Segmentation

3.1.4: High-Dimensionality and Dimension-Reduction

3.1.5: Relevance Feedback

3.1.6: Existing CBIR Systems

3.2: Video Indexing and Retrieval

3.2.1: Temporal Segmentation

3.2.2: Video Summarization

3.2.3: Compensation for Camera and Background Movement

3.2.4: Feature-based Modeling

3.2.5: High-level Semantic Modeling

References

1: Introduction

Multimedia data, such as text, audio, images and video, is rapidly evolving as the main form for the creation, exchange, and storage of information in the modern era. Primarily, this is attributed to rapid advances in the three major technologies that determine its growth: VLSI technology that is producing greater processing power; broad-band networks (ISDN, ATM, etc) that are providing much higher bandwidth for many practical applications, and multimedia compression standards (JPEG, H.263, MPEG, MP3, etc) that enable efficient storage and communication. The combination of these three advances is spurring the creation and processing of increasingly high-volume multimedia data, along with its efficient compression and transmission over high-bandwidth networks. This current trend towards the removal of any conceivable bottleneck in using multimedia and its impact on a whole spectrum of users, from advanced research organizations to home users, has led to the explosive growth of visual information available in the form of digital libraries and online multimedia archives. According to a press release by Google Inc. in December 2001, the search engine offers access to over 3 billion web documents and its Image search comprises more than 330 million images. AltaVista has been serving around 25 million search queries per day in more than 25 languages, with its multimedia search featuring over 45 million images, videos and audio clips. This explosive growth of multimedia data accessible to users poses a whole new set of challenges relating to its storage and retrieval. The current technology of text-based indexing and retrieval implemented for relational databases does not provide practical solutions for this problem of managing huge multimedia repositories. Most of the commercially available multimedia indexing and search systems index the media based on keyword annotations and use standard text based indexing and retrieval mechanisms to store and retrieve multimedia data. There are often many limitations with this method of keywords based indexing and retrieval especially in the context of multimedia databases. First, it is often difficult to describe with human languages the content of a multimedia object, for example an image having complicated texture patterns. Second, manual annotation of text phrases for a large database is prohibitively laborious in terms of time and effort. Third, since users may have different interests in the same multimedia object, it is difficult to describe it with a complete set of key words. Finally, even if all relevant object characteristics are annotated, difficulty may still arise due to the use of different indexing languages or vocabularies by different users. As recently as in 1990's, these major drawbacks of searching visual media based on textual annotations were recognized to be unavoidable and this prompted a surging increase in interest in content-based solutions [16]. In content-based retrieval, manual annotation of visual media is avoided and indexing and retrieval is instead performed on the basis of media content itself. There have been extensive studies on the design of automatic *content-based indexing and retrieval* (CBIR) systems. For visual media these contents may include, color, shape, texture, motion, etc. For audio/speech data contents may include phonemes, pitch, rhythm, cepstral coefficients, etc. Studies of human visual perception indicate that there exists a gradient of sophistication in human perception, ranging from seemingly primitive inferences of shapes, textures, colors, etc. to complex notions of structures such as chairs, buildings, affordances, and to cognitive processes such as recognition of emotions and feelings. Given the multidisciplinary nature of the techniques for modeling, indexing and retrieval of multimedia data, efforts from many different communities of engineering, computer science and psychology have merged in the advancement of CBIR systems. But the field is still in its infancy and calls for more coherent efforts to make practical CBIR systems a reality. In particular, robust techniques are needed to develop semantically rich models to represent data,

computationally efficient methods to compress, index, retrieve, and browse the information, and semantic visual interfaces integrating the above components into viable multimedia systems.

This chapter presents a review of the state of the art research in the area of multimedia systems. In Section 2, we present a review of storage and coding techniques for different media types. Section 3 studies fundamental issues related to the representation of multimedia data and discusses salient indexing and retrieval approaches introduced in the literature. For the sake of compactness and focus, in this chapter we review only CBIR techniques for visual data, i.e. for images and videos, for the review of systems for audio data readers are referred to [28], [15].

2: Multimedia Storage and Encoding

Raw multimedia data requires vast amount of storage and therefore usually it is stored in compressed format. Slow storage devices, e.g., CD-ROMs and Hard Disk Drives, do not support playback/display of uncompressed multimedia data (especially, video and audio) in real-time. The term *Compression* refers to removal of *Redundancy* from data. The more we reduce redundancy in the data, the higher the *Compression Ratio* that we achieve. The method by which redundancy is eliminated in order to increase data compression is known as source coding. In essence, the same (or nearly the same) information is represented using fewer data bits. There are several other reasons behind the popularity of compression techniques and standards for multimedia data. For example:

- Compression extends the playing time of a storage device. With compression, more data can be stored in the same storage space.
- Compression allows miniaturization of hardware system components. With less data to store, the same playing time is obtained with less hardware.
- Tolerances of system design can be relaxed. With less data to record, storage density can be reduced making equipment which is more resistant to adverse environments and requires less maintenance.
- For a given bandwidth, compression allows faster information transmission.
- For a given bandwidth, compression allows a better-quality signal transmission.

These are the reasons that compression technologies have helped development of compressed domain based modern communication systems and compact and rugged consumer products. Although compression in general is a useful technology, and in the case of multimedia data, an essential one, it should be used with caution because it comes with some drawbacks as well. By definition, compression removes redundancy from signals. Redundancy is, however, essential in making data resistant to errors. As a result, compressed data is more sensitive to errors than uncompressed data. Thus, transmission systems using compressed data must incorporate more powerful error-correction strategies. Most of the text compression techniques such as the Lampel-Ziv-Welch codes are very sensitive to bit errors as an error in the transmission of the code table value results in bit errors every time that table location is accessed. This phenomenon is known as *error propagation*. Other variable-length coding techniques, such as Huffman coding, are also sensitive to bit errors. In real-time multimedia applications, such as audio and video communications, some error concealment must be used in case of errors.

The applications of multimedia compression are limitless. The *International Standards Organization (ISO)* has provided standards that are appropriate for a wide range of possible compression products. The video encoding standards by ISO,

developed by MPEG (Motion Pictures Expert Group), embraces video pictures from the tiny screen of a videophone to the high-definition images needed for electronic cinema. Audio coding stretches from speech-grade mono to multichannel surround sound. Data compression techniques are classified as *lossless* and *lossy* coding. In lossless coding, the data from the decoder is identical bit-for-bit with the original source data. Lossless coding generally provides limited compression ratios. Higher compression is possible only with lossy coding in which data from the decoder is not identical to the original source data and between them minor differences exist. Lossy coding is not suitable for most applications using text data, but is used extensively for multimedia data compression as it allows much greater compression ratios. Successful lossy codecs are those in which the errors are imperceptible to a human viewer or listener. Thus, lossy codecs must be based on an understanding of psycho-acoustic and psycho-visual perception and are often called *perceptive codecs*. In the following subsections, we provide a very brief overview of some of the multimedia compression standards. Section 2.1 presents image encoding standards and, Section 2.2 discusses several video encoding standards.

2.1: Image Encoding Standards

As noted earlier, the task of compression schemes is to reduce the redundancy present in raw multimedia data representation. Images contain three forms of redundancy, viz:

- *Coding Redundancy*: Consider the case of an 8 bit per pixel image i.e., each pixel in the image is represented with an 8-bit value ranging between 0-255, depending upon the local luminosity level in that particular area of the image. Since the grayscale value of some of the pixels may be small (around zero, for a blacker pixel), representing those pixels with the same number of bits as for the ones with a higher pixel value (brighter pixels) is not a good coding scheme. Also, some of the grayscale values in the image may be occurring more often than the others. A more realistic approach would be to assign shorter codes to the more frequent data. So, instead of using fixed-length codes as above, we prefer the *variable length coding* schemes (e.g., Shannon-Fano, Huffman, Arithmetic, etc.) in which the smaller and more frequently occurring grayscale values get shorter codes. If the gray levels of an image are encoded in such a way that uses more code symbols than absolutely necessary to represent each gray level, the resulting image is said to contain the coding redundancy.
- *Inter-Pixel Redundancy*: For the case of image data, redundancy will always be present if we explore only the coding redundancy and however rigorously we minimize it by using state of the art variable length coding techniques. The reason is that images are typically composed of objects that have a regular and somewhat predictable morphology and reflectance and the pixel values are highly correlated. The value of any pixel can be reasonably predicted from the value of its neighbors; so the information carried by each individual pixel is relatively small. To exploit this property of the images, it is often necessary to convert the visual information of the image into somewhat non-visual format that better reflects the correlation between the pixels. The most effective technique in this regard is to transform the image into frequency domain by taking the DFT (Discrete Fourier Transform) or DCT (Discrete Cosine Transform), or any other such transform.
- *Psycho-visual Redundancy*: This type of redundancy arises from the fact that human eye's response is not equally sensitive to all the visual information. The information that has less relative importance to the eye is said to be psycho-visually redundant. Human perception of the visual information does not involve the quantitative analysis of every pixel, rather the eye searches for some recognizable groupings of them to be interpreted as distinguished features in the image. This is the

reason that DC component of a small section of the image, which tells us about the average luminosity level of that particular section of the image, contains more visually important information than a high frequency AC component, which has the information regarding the difference between luminosity levels of some successive pixels. Psychovisual redundancy can be eliminated by throwing away some of the redundant information. Since the elimination of psycho-visually redundant data results in a loss of quantitative information and is an irreversible process, it results in lossy data compression. How coarse or how fine to quantize the data depends upon what quality and/or what compression ratio is required at the output. This stage acts as the tuning tap in the whole image compression model. On the same grounds, human eye's response to color information is not as sharp as it is for the luminosity information. So, more color information is psycho-visually redundant than the grayscale information simply because eye can't perceive finer details of colors while for grayscale values, it can.

Next we outline the details of one very popular image compression standard, JPEG, that is a result of collaborative effort between ITU-T and ISO.

2.1.1 Digital Compression and Coding of Continuous-Tone Still Images (JPEG)

The Joint Photographic Experts Group (JPEG) standard is used for compression of continuous-tone still images [25]. This compression standard is based on the Huffman and Run-Length encoding of the quantized Discrete Cosine Transform (DCT) coefficients of image blocks. The widespread use of the JPEG standard is motivated by the fact that it consistently produces compression ratios in excess of 20:1. The compression algorithm can be operated on both grayscale as well as multi-channel color images. The color data, if in psycho-visually redundant RGB format, is first converted to some more "compression-friendly" color model like YCbCr or YUV. The image is first broken down into blocks of size 8x8 called *data units*. After that, depending upon color model and decimation scheme for chrominance channels involved, *Minimum Code Units* (MCUs) are formed. A minimum code unit is the smallest unit which is processed for DCT, quantization and variable-length encoding subsequently. One example for the case of YUV 4:1:1 color model (each of chrominance component being half in width and half in length), is shown in figure 1. Here the MCU consists of four data units from Y component and one each from U and V.

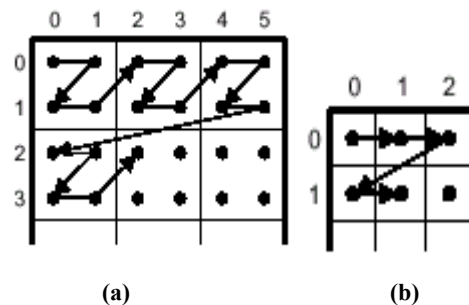


Figure 1: Interleaved data ordering of Y (a) and U & V components (b).

Each of the data units in an MCU is then processed separately. First a 2-D DCT operation is performed which changes the energy distribution of the original image and concentrates more information in low-frequencies. The DCT coefficients are then quantized to reduce the magnitude of coefficients to be encoded and to reduce some of the smaller ones to zero. The specs

Interframe coding is the main coding principal that is used in all standard video codecs to be explained shortly. First we will give theoretical discussion of temporal redundancy reduction and then explore some of the popular video compression standards in more detail.

Temporal redundancy is removed by using the differences between successive images. For static parts of the image sequence, temporal differences will be close to zero, and hence are not coded. Those parts which change between frames, either due to illumination variation or motion of objects, result in significant image error which needs to be coded. Image changes due to motion can be significantly reduced if the motion of the object can be estimated, and the difference taken on motion compensated image. To carry out *motion compensation*, first the amount and direction of moving objects has to be estimated. This is called *motion estimation*. The commonly used motion estimation technique in all the standard video codecs is the *Block Matching Algorithm* (BMA). In a typical BMA, a frame is divided into square blocks of N^2 pixels. Then, for a maximum motion displacement of w pixels per frame, the current block of pixels is matched against a corresponding block at the same co-ordinates but in previous frame within the square window of width $N+2w$. The best match on the basis of a matching criterion yields the displacement. Various measures such as Cross-Correlation Function (CCF), Mean Squared Error (MSE) and Mean Absolute Error (MAE) can be used in the matching criteria. In practical coders, MSE and MAE are more often used since it is believed that CCF does not give good motion tracking, especially when the displacement is not quite large. MSE and MAE are defined as:

$$MSE(i, j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N (f(m, n) - g(m+i, n+j))^2, -w \leq i, j \leq w$$

$$MAE(i, j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N |f(m, n) - g(m+i, n+j)|, -w \leq i, j \leq w,$$

where $f(m,n)$ represents the current block of N^2 pixels at coordinates (m,n) and $g(m+i,n+j)$ represents the corresponding block in the previous frame at new coordinates $(m+i, n+j)$. Motion estimation is one of the most computationally intensive parts of video compression standards and some fast algorithms for this have been reported. One such algorithm is Three-Step Search which is the recommended method for H.261 codecs to be explained ahead. It computes motion displacements up to 6 pixels per frame. In this method all eight positions surrounding the initial location with a step size of $w/2$ are searched first. At each minimum position the search step size is halved and the next eight positions are searched. The process is outlined in figure 3. This method, for w set as 6 pixels per frame, searches 25 positions to locate the best match.

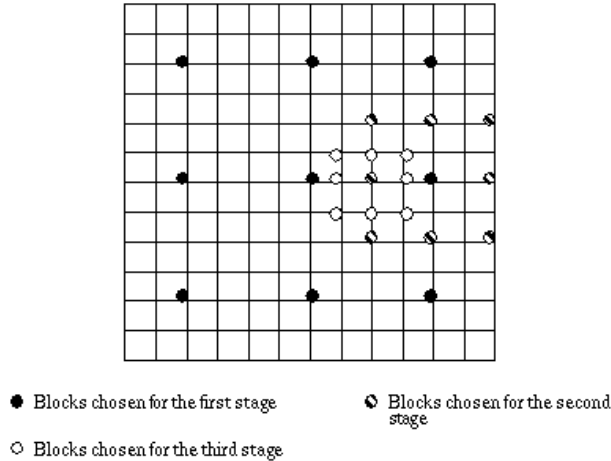


Figure 3: Example path for convergence of Three Step Search.

2.2.1: MPEG – Motion Photographic Expert Group

The Motion Picture Expert Group (MPEG) provides a collection of motion picture compression standards developed by the IEC. Its goal is to introduce standards for movie storage and communication applications. These standards include audio compression representation, video compression representation, and system representation. Below we briefly outline the details of three video compression standards developed by MPEG.

2.2.1.1: Coding of Moving Pictures for Digital Storage Media (MPEG-1)

The goal of MPEG-1 was to produce VCR NTSC (352 x 240) quality video compression to be stored on CD-ROM (CD-I and CD-Video formats) using a data rate of 1.2 Mbps. This approach is based on the arrangement of frame sequences into a Group Of Pictures (GOP) consisting of four types of pictures: I-Picture (Intra), P-Picture (Predictive), B-Picture (Bidirectional), and D-Picture (DC). I-Pictures are Intraframe JPEG encoded pictures that are inserted at the beginning of the GOP. P and B-Pictures are Interframe motion compensated JPEG encoded macroblock difference pictures that are interspersed throughout the GOP.¹ The system level of MPEG-1 provides for the integration and synchronization of the audio and video streams. This is accomplished by multiplexing and including timestamps in both the audio and video streams from a 90 KHz system clock [39].

In MPEG-1 due to the existence of several picture types, GOP is the highest level of hierarchy. The first coded picture in a GOP is an I-picture. It is followed by an arrangement for P- and B- pictures, as shown in Figure 4. GOP length is normally defined as the distance N between I-pictures. The distance between anchor I/P to P pictures is represented by M . The GOP can be any length but there has to be one I-picture in each GOP. Applications requiring random access, fast forward play or fast and reverse play may use short GOPs. GOP may also start at scene cuts otherwise motion compensation is not effective. Each picture is further divided into a group of macroblocks, called *slices*. The reason for defining slices is to reset the variable length code to prevent channel error propagation into the picture. Slices can have different sizes within a picture, and the division in

¹ MPEG-1 restricts the GOP to sequences of fifteen frames in progressive mode.

one picture need not be the same as in any other picture. The slices can begin and end at any macroblock in a picture, but the first slice has to begin at the top-left corner of the picture and the end of last slice must be the bottom right macroblock of the picture. Each slice starts with a *slice start code*, and is followed by a code that defines its position and a code that sets the quantization step size. Slices are divided into *macroblocks* of size 16x16 which are further divided into *blocks* of size 8x8, much like in JPEG.

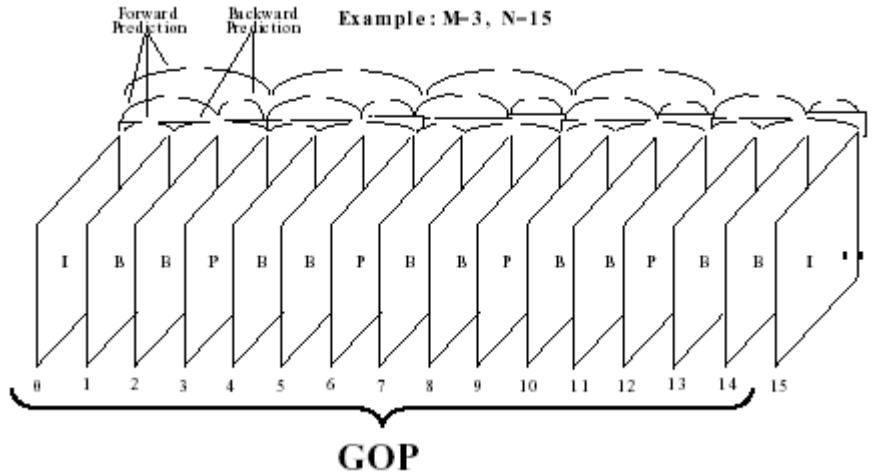


Figure 4: A typical MPEG-1 GOP.

The encoding process for MPEG-1 encoder is as follows: For a given macroblock, the coding mode is first chosen. This depends on the picture type, the effectiveness of motion compensated prediction in that local region and the nature of the signal within the block. Next, depending on the coding mode, a motion compensated prediction of the contents of block based on past and/or future reference pictures is formed. This prediction is subtracted from the actual data in the current macroblock to form an error signal. After that, this error signal is divided into 8x8 blocks and a DCT is performed on each block. The resulting two-dimensional 8x8 block of DCT coefficients is quantized and scanned in zigzag order to convert it into one-dimensional string of quantized coefficients like in JPEG. Finally, the side information for the macroblock, including the type, block-pattern and motion vectors along with DCT coefficients are coded. A unique feature of MPEG-1 standard is the introduction of B-pictures that have access to both past and future anchor points. They can either use past frame, called *forward* motion estimation, or the future frame for *backward* motion estimation as shown in figure 5.

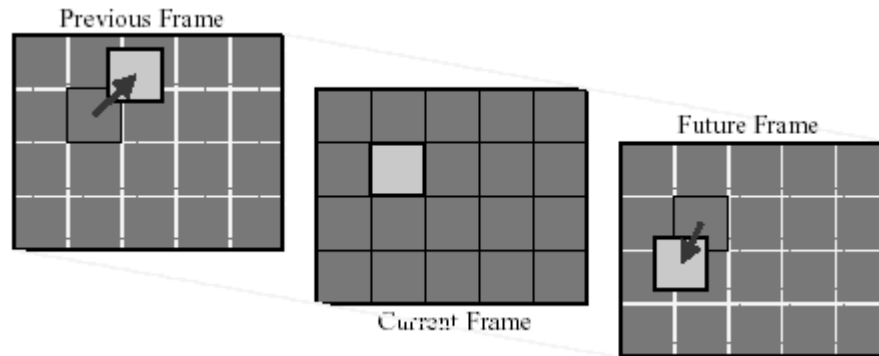


Figure 5: Motion estimation in B-pictures.

Such an option increases motion compensation efficiency especially when there are occluded objects in the scene. From the two forward and backward motion vectors, the encoder has a choice of choosing either any of the two or a weighted average of the two, where weights are inversely proportional to the distance of the B-picture with its anchor position.

2.2.1.2: Coding of High Quality Moving Pictures (MPEG-2)

The MPEG-1 standard was targeted for coding of audio and video for storage, where the media error rate is negligible. Hence MPEG-1 bitstream is not designed to be robust to bit errors. Also, MPEG-1 was aimed at software oriented image processing, where large and variable length packets could reduce the software overhead. The MPEG-2 standard on the other hand is more generic for a variety of audio-visual coding applications. It has to have the error-resilience for broadcasting and ATM networks. The aim of MPEG-2 was to produce broadcast-quality video compression and was expanded to support higher resolutions including High Definition Television (HDTV).² MPEG-2 supports four resolution levels: low (352 x 240), main (720 x 480), high-1440 (1440 x 1152), and high (1920 x 1080) [40]. The MPEG-2 compressed video data rates are in the range of 3—100 Mbps.³ Although the principles used to encode MPEG-2 are very similar to MPEG-1, it provides much greater flexibility by offering several profiles that differ in the presence or absence of B-Pictures, chrominance resolution, and coded stream scalability.⁴ MPEG-2 supports both progressive and interlaced modes.⁵ Significant improvements have also been introduced in the MPEG-2 system level. The MPEG-2 systems layer is responsible for the integration and synchronization of the Elementary Streams (ES): audio and video streams, as well as an unlimited number of data and control streams that can be used for various applications such as subtitles in multiple languages. This is accomplished by first packetizing the ESs thus forming the Packetized Elementary Streams (PES). These PESs contain timestamps from a system clock for synchronization. The PESs are subsequently multiplexed to form a single output stream for transmission in one of two modes: Program Stream (PS) and Transport Stream (TS). The PS is provided for error-free environments such as storage in CD-ROM. It is used for multiplexing PESs that share a common time-base, using long variable-length packets.⁶ The TS is designed for noisy environments such as communication over ATM networks. This mode permits multiplexing streams (PESs and PSs) that do not necessarily share a common time-base, using fixed-length (188 bytes) packets.

In the MPEG-2 standard, pictures can be interlaced while in MPEG-1, the pictures are progressive only. The dimensions of the units of blocks used for motion estimation/compensation can change. In the interlaced pictures, since the number of lines per field is half the number of lines per frame, then with equal horizontal and vertical resolutions, for motion estimation it might be appropriate to choose blocks of 16x8, i.e., 16 pixels over 8 lines. The second major difference between the two is *scalability*. The scalable modes of MPEG-2 video encoders are intended to offer interoperability among different services or to accommodate the varying capabilities of different receivers and networks upon which a single service may operate. MPEG-2 also has a choice of a different DCT coefficient scanning mode *alternate scan* as well as *zigzag scan*.

² The HDTV Grand Alliance standard has adopted the MPEG-2 video compression and transport stream standards in 1996.

³ The HDTV Grand Alliance standard video data rate is approximately 18.4 Mbps.

⁴ The MPEG-2 video compression standard, however, does not support D-Pictures.

⁵ The interlaced mode is compatible with the field format used in broadcast television interlaced scanning.

⁶ The MPEG-2 program stream (PS) is similar to the MPEG-1 systems stream.

2.2.1.3: Content-based Video Coding (MPEG-4)

The intention of MPEG-4 was to provide low bandwidth video compression at data rate of 64 Kbps that can be transmitted over a single N-ISDN B channel. This goal has evolved to the development of flexible, scalable, extendable and interactive compression streams that can be used with any communication network for universal accessibility (e.g., Internet and wireless networks). MPEG-4 is a genuine multimedia compression standard that supports audio and video as well as synthetic and animated images, text, graphics, texture, and speech synthesis [41]. The foundation of MPEG-4 is on the hierarchical representation and composition of Audio-Visual Objects (AVO). MPEG-4 provides a standard for the configuration, communication, and instantiation of classes of objects: The configuration phase determines the classes of objects required for processing the AVO by the decoder. The communication phase supplements existing classes of objects in the decoder. Finally, the instantiation phase sends the class descriptions to the decoder. A video object at a given point in time is a Video Object Plane (VOP). Each VOP is encoded separately according to its shape, motion, and texture. The shape encoding of a VOP provides a pixel map or a bitmap of the shape of the object. The motion and texture encoding of a VOP can be obtained in a manner similar to that used in MPEG-2. A multiplexer is used to integrate and synchronize the VOP data and composition information—position, orientation, and depth—as well as other data associated with the AVOs in a specified bitstream. MPEG-4 provides universal accessibility supported by error robustness and resilience, especially in noisy environments at very low data rates (less than 64 Kbps): bitstream resynchronization, data recovery, and error concealment. These features are particularly important in mobile multimedia communication networks.

2.2.2: H.26X

The H.26X provides a collection of video compression standards developed by the ITU-T. The main focus of this effort is to present standards for videoconferencing applications compatible with the H.310 and H.32X communication network standards. These communication network standards include video compression representation, audio compression representation, multiplexing standards, control standards, and system standards. The H.26X and MPEG standards are very similar with relatively minor differences due to the particular requirements of the intended applications.

2.2.2.1: Coding for Video Conferencing (H.261)

The H.261 standard has been proposed for video communications over ISDN at data rates of px64 Kbps. It relies on intra and inter-frame coding where integer-pixel accuracy motion estimation is required for inter mode coding [18].

2.2.2.2: Video Coding for Low Bit-Rate Communications (H.263)

The H.263 standard is aimed at video communications over POTS and wireless networks at very low data rates (as low as 18-64 Kbps). Improvements in this standard are due to the incorporation of several features such as half-pixel motion

estimation, overlapping and variable blocks sizes, bidirectional temporal prediction⁷, and improved variable-length coding options [19].

2.2.2.3: H.26L

The H.26L standard is designed for video communications over wireless networks at low data rates. It provides features such as fractional pixel resolution and adaptive rectangular block sizes.

3: Multimedia Indexing and Retrieval

As discussed in the previous sections, multimedia data poses its distinct challenges for modeling and representation. The huge amount of multimedia information now available makes it all the more important to organize these multimedia repositories in a structured and coherent way so as to make it more accessible to large number of users. In this section, we explore the problem of storing multimedia information in a structured form (indexing) and searching the multimedia repositories in an efficient manner (retrieval). Section 3.1 outlines an image indexing and retrieval paradigm. We first discuss the motivation for using content-based indexing and retrieval for images and then explore several different issues and research directions in this field. Section 3.2 highlights similar problems in the area of video indexing and retrieval. As with any other emerging field going through intellectual and technical exploration, the domain of content based access to multimedia repositories poses a number of research issues which cannot be summarized in a single concise presentation. One such issue is query language design for multimedia databases. The interested reader can refer to [11], [33], [24],[67].

3.1: Image Indexing and Retrieval

Due to the tremendous growth of visual information available in the form of images, effective management of image archives and storage systems is of great significance and an extremely challenging task indeed. For example, a remote sensing satellite, which generates seven band images including three visible and four infrared spectrum regions, produces around 5000 images per week. Each single spectral image, which corresponds to a 170 km x 185 km of the earth region, requires 200 Mega bytes of storage. The amount of data originated from satellite systems is already reaching a terabyte per day. Storing, indexing and retrieving such a huge amount of data by its contents, is a very challenging task. Generally speaking, data representation and feature based content modeling are two basic components required by the management of any multimedia database. As far as the image database is concerned, the former is concerned with image storage while the latter is related to image indexing and retrieval. Depending on the background of the research teams, different levels of abstractions have been assumed to model the data. As shown in Figure 6, we classify these abstractions into three categories based on the gradient model of human visual perception. In this figure, we also capture the mutual interaction of some of the disciplines of engineering, computer science, and cognitive sciences.

⁷ Bidirectional temporal prediction, denoted as a PB-picture, is obtained by coding two pictures as a group and avoiding the reordering necessary in the decoding of B-pictures.

Level I represents systems that model raw image data using features such as color histogram, shape and texture descriptors. This model can be used to serve the queries like “find pictures with dominant red color on a white background”. CBIR systems based on these models operate directly on the data, employing techniques from signal processing domain. Level II consists of derived or logical features involving some degree of statistical and logical inference about the identity of objects depicted by visual media. An example query at this level can be “find pictures of Eiffel Tower”. Using these models, systems normally operate on low-level feature representation, though they can also use image data directly. Level III deals with semantic abstractions involving a significant amount of high-level reasoning about the meaning and purpose of the objects or scenes depicted. An example of a query at this level can be “find pictures of laughing children”. As indicated at Level III of the figure, the artificial intelligence (AI) community has had the leading role in this effort. Systems at this level can take semantic representation based on input generated at level II.

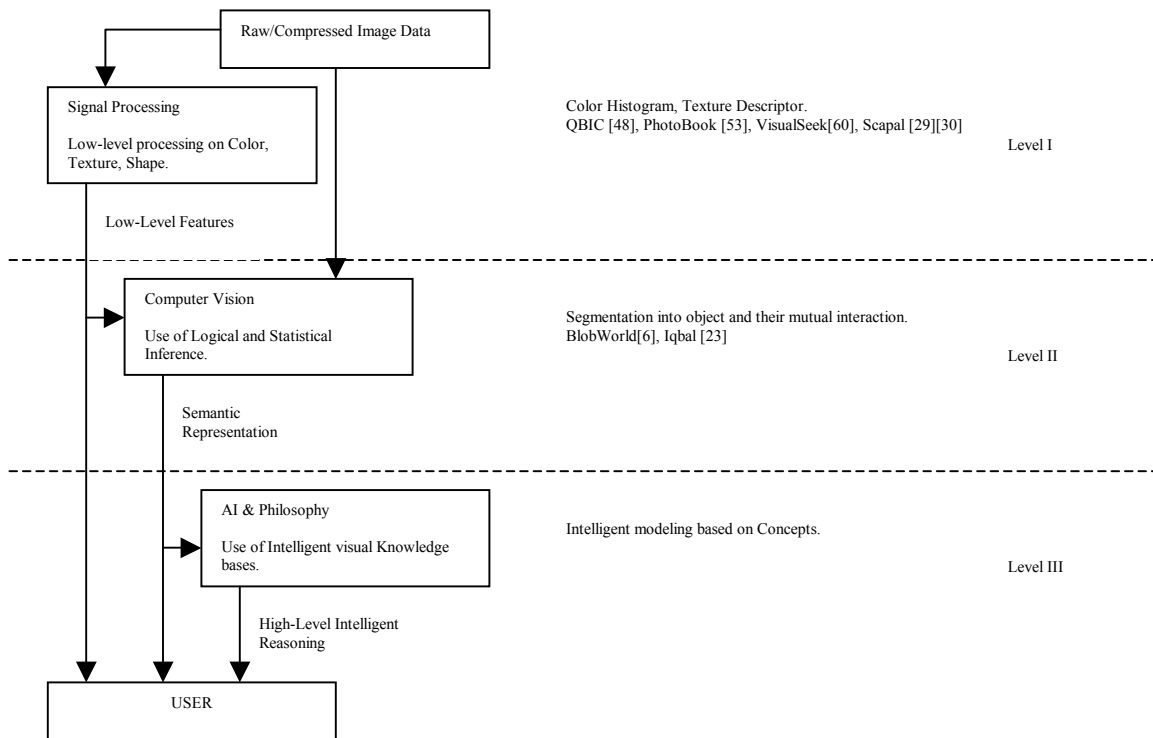


Figure 6: Classification of CBIR Techniques. Level I: Low-level physical modeling of Raw Image Data. Level II: Representation of derived or logical features. Level III: Semantic level abstractions.

In the following subsections we explore some of the major building blocks of content based image indexing and retrieval (CBIR) systems.

3.1.1: Low-level Feature based Indexing

Low-level visual feature extraction to describe image content is at the heart of CBIR systems. The features to be extracted can be categorized into general features and domain-specific features. The latter ones may include human faces, finger prints, human skin, etc. Feature extraction in the former context, i.e., from databases that contain images of wide ranging content containing images that do not portray any specific topic but come from various sources and are without common theme, is a very challenging job. One possible approach is to perform segmentation first and then extract visual features from segmented objects. But unconstrained segmentation of an object from the background is often not possible as there generally is no particular object in the image. Therefore, segmentation in such a case is of very limited use as a stage preceding feature extraction. The images thus need to be described as a whole unit and one should devise feature extraction schemes that do not require segmentation. This restriction excludes a vast number of well-known feature extraction techniques from low-level feature based representation: all boundary-based methods and many area-based methods. Basic pixel-value-based statistics, possibly combined with edge detection techniques, that reflect the properties of human visual system in discriminating between image patches can be used. Invariance to specific transforms is an issue of interest in feature extraction too. Feature extraction methods that are global in their nature or perform averaging over the whole image area are often inherently translation invariant. Other types of invariances, e.g., invariance to scaling, rotation, and occlusion, can be obtained with some feature extraction schemes by using proper transformations. Because of the perception subjectivity, there does not exist a single best representation for a given feature. For any given feature, there exist multiple representations that characterize the feature from different perspectives. The main features used in CBIR systems can be categorized into three groups, namely color features, texture features and shape features. In the following subsections, we will review the importance and implementation of each feature in the context of image content description.

3.1.1.1: Color

Color is one of the most widely used low-level features in the context of indexing and retrieval based on image content. It is relatively robust to background complication and independent of image size and orientation. Typically, the color of an image is represented through some color model. A color model is specified in terms of 3-D coordinate system and a subspace within that system where each color is represented by a single point. The more commonly used color models are RGB (red, green, blue), HSV (hue, saturation, value) and YIQ (luminance and chrominance). Thus the color content is characterized by 3-channels from some color model. One representation of color content of the image is by using color histogram. The histogram of a single channel of an image with values in the range $[0, L-1]$ is a discrete function $p(i) = n_i/n$, where i is the value of the pixel in current channel, n_i is the number of pixels in the image with value i , n is the total number of pixels in the image, and $i = 0, 1, 2, \dots, L-1$. For a three-channel image, we will have three such histograms. The histograms are normally divided into bins in an effort to coarsely represent the content and reduce dimensionality of subsequent matching phase. A feature vector is then formed by concatenating the three channel histograms into one vector. For image retrieval, histogram of query image is then matched against histogram of all images in the database using some similarity metric. One similarity metric that can be used in this context is Histogram Intersection. The intersection of histograms h and g is given by:

$$d(h, g) = \frac{\sum_{m=0}^{M-1} \min(h[m], g[m])}{\min\left(\sum_{m=0}^{M-1} h[m], \sum_{m=0}^{M-1} g[m]\right)}$$

In this metric, colors not present in the user's query image do not contribute to the intersection. Another similarity metric between histograms h and g of two images is Histogram Quadratic Distance, which is given by:

$$d(h, g) = \sum_{m_0=0}^{M-1} \sum_{m_1=0}^{M-1} (h[m_0] - g[m_0]) \cdot a_{m_0, m_1} \cdot (h[m_1] - g[m_1]),$$

where a_{m_0, m_1} is the cross-correlation between histogram bins based on the perceptual similarity of the colors m_0 and m_1 . One appropriate value for the cross-correlation is given by:

$$a_{m_0, m_1} = 1 - d_{m_0, m_1},$$

where d_{m_0, m_1} is the distance between colors m_0 and m_1 normalized with respect to the maximum distance. Color Moments have also been applied in image retrieval. The mathematical foundation of this approach is that any color distribution can be characterized by its moments. Furthermore, since most of the information is concentrated in the low-order moments, only the first moment (mean) and the second and third central moments (variance and skewness) can be used for robust and compact color content representation. Weighted Euclidean distance is then used to compute color similarity. To facilitate fast search over large-scale image collections, Color Sets have also been used as an approximation to color histograms. The color model used is HSV and the histograms are further quantized into bins. A color set is defined as a selection of the colors from quantized color space. Because color set feature vectors are binary, a binary search tree is constructed to allow fast search [59].

One major drawback of color histogram based approaches is the lack of explicit spatial information. Specifically, based on global color based representation, it is hard to distinguish between a red car on white background and a bunch of red balloons with white background. This problem is addressed by Khokhar et al [29], where they have used encoded quadtree spatial data structure to preserve the structural information in the color image. Based on a perceptually uniform color space CIE Lab*, each image is quantized into k bins to represent k different color groups. A color layout corresponding to pixels of each color in the whole image is formed for each bin and is represented by the corresponding encoded quadtree. This encoded quadtree based representation not only keeps the spatial information intact, but also results in a system that is highly scalable in terms of query search time.

3.1.1.2: Texture

An image can be considered as a mosaic of regions with different appearances, and the image features associated with these regions can be used for search and retrieval. Although no formal definition of *texture* exists, intuitively this descriptor provides measures of properties such as smoothness, coarseness, and regularity. These properties can generally not be attributed to the presence of any particular color or intensity. Texture corresponds to repetition of basic texture elements called texels. A texel consists of several pixels and can be periodic, quasi periodic or random in nature. Texture is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, fabric, etc. It contains important information about the structural

arrangement of surfaces and their relationship to the surrounding environment. The three principal approaches used in practice to describe the texture of a region are statistical, structural and spectral. Statistical approaches yield characterization of textures as smooth, coarse, grainy and so on. Structural techniques deal with the arrangement of image primitives, such as description of texture based on regularly spaced parallel lines. Spectral techniques are based on properties of Fourier spectrum and are used primarily to detect global periodicity in an image by identifying high-energy, narrow peaks in the spectrum [17]. Haralick et al [22] proposed the co-occurrence matrix representation of texture feature. This method of texture description is based on the repeated occurrence of some gray-level configuration in the texture; this configuration varies rapidly with distance in fine textures and slowly in coarse textures. This approach explores the gray level spatial dependence of texture. It first constructs a co-occurrence matrix based on the orientation and distance between image pixels and then extracts meaningful statistics from the matrix as texture representation. Motivated by the psychological studies in human visual perception of texture, Tamura et al [64] have proposed the texture representation from a different angle. They developed computational approximations to the visual texture properties found to be important in psychology studies. The six visual texture properties are *coarseness*, *contrast*, *directionality*, *linelikeness*, *regularity*, and *roughness*. One major distinction between the Tamura texture representation and co-occurrence matrix representation is that all the texture properties in Tamura representation are visually meaningful whereas some of the texture properties used in co-occurrence matrix representation may not. This characteristic makes Tamura texture representation very attractive in Image retrieval, as it can provide a friendlier user-interface.

The use of texture feature requires texture segmentation which remains a challenging and computationally intensive task. In addition, texture based techniques lack robust texture models and correlation with human perception.

3.1.1.3: Shape

Shape is an important criterion for matching objects based on their profile and physical structure. In image retrieval applications, shape features can be classified into global and local features. Global features are the properties derived from the entire shape such as roundness, circularity, central moments, and eccentricity. Local features are those derived by partial processing of a shape including size and orientation of consecutive boundary segments, points of curvature, corners and turning angle. Another categorization of shape representation is boundary-based and region-based. The former uses only outer boundary of the shape while the latter uses entire shape of the region. Fourier Descriptors and Moment invariants are the most widely used shape representation schemes. The main idea of Fourier Descriptor is to use the Fourier transformed boundary as the shape feature. Moment invariant technique uses region-based moments, which are invariant to transformations, as the shape feature. Hu et al [21] proposed a set of seven invariant moments derived from second and third moments. This set of moments is invariant to translation, rotation and scale changes. Finite Element Method (FEM) [52] has also been used as shape representation tool. FEM defines a stiffness matrix, which describes how each point on the object is connected to other points. The eigenvectors of the stiffness matrix are called modes and span a feature space. All the shapes are first mapped into this space and similarity is then computed based on the eigenvalues. Along the similar lines of Fourier Descriptors, Arkin et al [1] developed a Turning function based approach for comparing both convex and concave polygons.

3.1.2: Spatial vs. Compressed Domain Processing

Given the huge storage requirements of non-textual data, the vast volumes of images are normally stored in compressed form. One approach in CBIR systems is to first decompress the images and transform them into the format used by system. The default color format used by majority of the CBIR systems is RGB which is very redundant so unsuitable for storage, but very easy to process on and use for display. Once the raw image data has been extracted after decompression, any of the content modeling techniques can be applied to yield a representation of the image content for indexing. This process of decompressing the image before content representation obviously poses an overhead in the most likely scenario when more and more image content is being stored in compressed form due to the success of image coding standards like JPEG and JPEG-2000. A better approach towards this issue in many of the modern systems is to process compressed domain images as first class and default medium, i.e., compressed images should be operated upon directly. Since compressed domain representation either has all (if it is a loss-less coding scheme) or most of the important (if it is a lossy scheme, depending upon the quantization setting in encoder) image information intact, indexing based on image content can be performed from minimal decoding of compressed images.

Discrete Cosine Transform (DCT) is at the heart of JPEG still-image compression standard and many of the video compression standards, like MPEG- 1,2 and H.261. Sethi et al [56] have used DCT coefficients of encoded blocks to mark areas of interest in the image. This distinction can be used to give more preference to these areas when processing the image for content representation. Location of areas of interest looks for those parts of the image which show sufficiently large intensity changes. This is achieved by computing the variance of pixels in a rectangular window around the current pixel. In DCT domain, this translates to computing the AC energy according to the relationship:

$$E = \sum_{u=0}^7 \sum_{v=0}^7 F_{uv}^2 \quad (u, v) \neq (0,0),$$

where F_{uv} stands for AC coefficients in the block, and the encoded block is 8x8 as in image/video coding standards. They also propose fast coarse edge detection techniques using DCT coefficients. A comparison of their approach with edge detection techniques in spatial domain speaks in favor of DCT coefficients based edge detection because a coarse representation of edges is quite sufficient for content description purposes.

Since image coding and indexing are quite overlapping processes in terms of storage and searching, one approach is to unify the two problems in a single framework. There has been a recent shift in trends in terms of transformation used for frequency domain processing, from DCT to Discrete Wavelet Transform (DWT) because of its time-frequency and multi-resolution analysis nature. DWT has been incorporated in modern image and video compression standards like JPEG-2000 and MPEG-4. One such system that uses DWT for compression and indexing of images is proposed in Liang et al [36]. The wavelet based image encoding techniques depend on successive approximate quantization (SAQ) of wavelet coefficients in different subbands from wavelet decomposition. The image indexing from DWT encoded image is mainly based on significant coefficients in each subband. Significant coefficients in each subband are recognized as the ones whose magnitude is greater than certain threshold which is different at each decomposition level. The initial threshold is chosen to be half the maximum magnitude at first decomposition level, while successive thresholds are given by dividing the threshold at previous decomposition level by 2. During the coding process, a binary map called significant map is maintained so the coder knows the location of significant as well as insignificant coefficients. To index texture, a two-bin histogram of wavelet coefficient at each

subband is formed with the count of significant and insignificant wavelet coefficients in the two bins. For color content representation, YUV color space is used. A non-uniform histogram of 12 bins containing count of significant coefficients given each of the 12 thresholds is computed for each color channel. This way, three 12-bin histograms are computed for luminance (Y) and chrominance (U, V) channels. For each of the histogram, first, second and third order moments (mean, variance and skewness) are computed and used as indexing features for color.

3.1.3: Segmentation

Most of the existing techniques in current CBIR systems depend heavily on low-level feature based description of image content. So most existing approaches represent images based only on the “stuff” they are composed of with little regard to spatial organization of the low-level features. On the other hand, users of the CBIR systems often would like to find images containing particular objects (“things”). This gap between low-level description of the image content and the object they represent can be filled by performing segmentation on images to be indexed. Segmentation subdivides an image into its constituent parts or objects. Segmentation algorithms for monochrome images generally are based on one of two basic properties of gray-level values: discontinuity and similarity. In the first category, the approach is to partition an image based on abrupt changes in gray level. The principal areas of interest within this category are detection of isolated points and detection of lines and edges in an image. The principal approaches in the second category are based on thresholding, region growing, and region splitting and merging. The BlobWorld system proposed by Malik et al [6] is based on segmentation using the Expectation-Maximization algorithm on combined color and texture features. It represents the image as a small set of localized coherent regions in color and texture space. After segmented the image into small regions, a description of each region’s color, texture, and spatial characteristics is produced. Each image in their representation may be visualized by an ensemble of 2-D ellipses or “blobs”, each of which possesses a number of attributes. The number of blobs in an image is not very overwhelming to facilitate fast image retrieval applications and is typically less than ten. Each blob represents a region on the image which is roughly homogeneous with respect to color or texture. A blob is described by its dominant color, mean texture descriptors, spatial centroid and scatter matrix. The exact retrieval process is then performed on the blobs in query image. On similar lines but in domain-specific context, Iqbal et al [23] apply perceptual grouping to develop a CBIR system for images containing buildings. In their work, semantic interrelationships between different primitive image features are exploited by perceptual grouping to detect presence of manmade structures. Perceptual grouping uses concepts as grouping by proximity, similarity, continuation, closure, and symmetry to group primitive image features into meaningful higher-level image relations. Their approach is based on the observation that the presence of a manmade structure in an image will generate a large number of significant edges, junctions, parallel lines and groups, in comparison with an image with predominantly non-building objects. These structures are generated by the presence of corners, windows, doors, boundaries of the buildings, etc. The feature they extract from an image are hierarchically in nature and include *line segments, longer linear lines, L junctions, U junctions, parallel lines, parallel groups, significant parallel groups.*

Most of the segmentation methods discussed in image processing and analysis literature are automatic. A major advantage of this type of segmentation algorithms is that it can extract boundaries from large number of images without occupying the user’s time and effort. However, in an unconstrained domain, for non-preconditioned images, which is the case with image

CBIR systems, the automatic segmentation is not always reliable. What an algorithm can segment in this case is only regions, but not objects. To obtain high-level objects, human assistance is almost always needed for reliable segmentation.

3.1.4: High-Dimensionality and Dimension-Reduction

It is obvious from above discussion that content-based image retrieval is a high-dimensional feature vector matching problem. To make the systems truly scalable to large size image collections, there are two factors to be taken care of. First, the dimensionality of the feature space needs to be reduced in order to achieve the embedded dimension. Second, efficient and scalable multidimensional indexing techniques need to be adapted to index the reduced but still high-dimensional feature space. In the context of dimensionality reduction, a transformation of the original data set using Karhunen-Loeve Transform (KLT) can be used. KLT features data-dependant basis functions obtained from a given data set and achieves the theoretical ideal in terms of compressing the data set. An approximation to KLT given by Principal Component Analysis (PCA) gives a very practical solution to the computationally intensive process of KLT. PCA, introduced by Pearson in 1901 and developed independently by Hotelling in 1933, is probably the oldest and best known of the techniques of multivariate analysis. The central idea of PCA is to reduce the dimensionality of a data set in which there are a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This reduction is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so the first few retain most of the variation present in *all* of the original variables. Computation of the principal components reduces to the solution of an eigenvalue-eigenvector problem for a positive-semidefinite symmetric matrix. Given that x is a vector of p random variables, the first step in PCA evaluation is to look for a linear function $\alpha'_1 x$ of the elements of x which has maximum variance, where α_1 is a vector of p constants $\alpha_{11}, \alpha_{12}, \dots, \alpha_{1p}$. Next, look for a linear function $\alpha'_2 x$, uncorrelated with $\alpha'_1 x$, which has maximum variance, and so on. The k th derived variable $\alpha'_k x$ is the k th PC. Up to p PCs can be found, but in general most of the variation in x can be accounted for by m PCs, where $m \ll p$. If the vector x has known covariance matrix Σ , then the k th PC is given by an orthonormal linear transformation of x as $y_k = \alpha'_k x$ where α_k is an eigenvector of Σ corresponding to its k th largest eigenvalue λ_k . Consider an orthogonal matrix Φ_q with α_k as the k th column and containing $q \ll p$ columns corresponding to q PCs, then it can be shown that for the transformation $y = \Phi_q x$, the determinant of covariance matrix for transformed data set y , $\det(\Sigma_y)$ is maximized. The statistical importance of this property follows because the determinant of a covariance matrix, which is called the *generalized variance*, can be used as a single measure of spread for a multivariate random variable. The square root of the generalized variance, for a multivariate normal distribution, is proportional to the 'volume' in p -dimensional space which encloses a fixed proportion of the probability distribution of x . For multivariate normal x , the first q PCs are therefore q linear functions of x whose joint probability distribution has contours of fixed probability which encloses the maximum volume [26]. If the data vector x is normalized by its variance and autocorrelation matrix instead of covariance matrix is used, then above-mentioned optimality property and derivation of PCs still hold. For the efficient computation of PCs, at least in context of PCA rather than general eigenvalue problems, Singular Value Decomposition (SVD) has been termed as the best approach available [7].

Even after the dimension of the data set has been reduced, the data set is still almost always fairly high-dimensional. There have been contributions from three major research communities in this direction, i.e., Computational Geometry, Database Management and Pattern Recognition. The history of multi-dimensional indexing techniques can be tracked back to middle 1970's when Cell methods, quad-tree and k-d tree were first introduced. However, their performance was far from satisfactory. Pushed by the then urgent demand of spatial indexing from GIS and CAD systems, Guttman proposed the R-tree indexing structure in 1984. Some good reviews of various indexing techniques in the context of image retrieval can be found in [66]. In [30], Khokhar et al deal with the feature vector formation and its efficient indexing as one problem and suggest a solution in which query response time is relatively independent of the database size. They exploit energy compaction properties of the vector wavelets and design suitable data structures for fast indexing and retrieval mechanisms.

3.1.5: Relevance Feedback

The problem of content based image retrieval is different than the conventional computer vision based pattern recognition task. The fundamental difference between the two is that while in latter we are looking for exact match for the object to be searched with as small and as accurate a retrieved list as possible. But in the former, the goal is to extract as many of the “similar” objects as possible, the notion of similarity being very loose as compared to the notion of exact match. Also, the human user is the indispensable part in the former. Early literature and CBIR systems emphasized on fully automatic operation. But this approach didn't take into account the fact that ultimate end-user of the CBIR system is human, and that image is inherently a subjective media, i.e., the perception of image content is very subjective and same content can be interpreted differently by users having different search criteria. This human perception subjectivity has different levels to it, i.e., one users might be more interested in a different dominant feature of the image than the other; or the two might be interested in the same feature (say, texture) but the perception of a specific texture might be different for the two users. Recent drive is more towards how humans perceive image content and how can we integrate such a “human model” into the image retrieval systems. Huang et al [55] have reported a formal model of CBIR system with relevance feedback integrated into it. They first initialize retrieval system with uniformly distributed weights for each feature. Then user's information need is distributed among all the features. The similarity is then computed on the basis of weights by user's input and retrieval results are displayed to the user. The user marks each retrieved result as *highly relevant*, *relevant*, *no-opinion*, *irrelevant* and *highly irrelevant* according to his information needs and perception subjectivity. The system updates its weights and goes back into the loop.

3.1.6: CBIR Systems: QBIC (Query By Image Content)—IBM Corp.

The field of content-based indexing and retrieval has been an active area of research for the past few decades. The research effort that has gone into development of techniques towards this problem has led to some very successful systems currently available as commercial products as well as other research systems available for the academic community. Some of these CBIR systems include Virage [3], Netra [43], PhotoBook [53], VisualSeek [60], WebSeek [61], MARS [42], BlobWorld [6] etc. A comparative study of many of the CBIR systems can be found in [65]. In this section, for illustrative purposes we review one example of a commercial CBIR system known as QBIC (Query By Image Content).

QBIC is the first commercial content-based image retrieval system. Developed by IBM Almaden Research Centre, it is an open framework technology, which can be utilized for both static and dynamic image retrieval. QBIC has undergone several

iterations since it was first reported [48]. QBIC allows users to graphically pose and refine queries based on multiple visual properties including color, shape and texture. QBIC supports several query types: simple, multi-feature, and multi-pass. A simple query involves only one feature. For example, identify images that have a color distribution similar to the query image. A complex query involves more than one feature, which can take the form of a multi-feature or a multi-pass query. For example, identify images that have similar color and texture features. With the multi-feature query the system searches through the different types of feature data in the database in order to identify similar images. All feature classes have equal weightings during the search, and all feature tables are searched in parallel. In contrast, with a multi-pass query the output of an initial search is used as the basis for the next search. The system reorganizes the search results from a previous pass based on the "feature distances" in the current pass. For example, identify images that have a similar color distribution, and then reordering the results based on color composition. With multi-feature and multi-pass queries, users can weight features to specify their relative importance. QBIC technology has been incorporated into several IBM software products, including DB2 Image Extender and Digital Library. QBIC supports several matching features including color, shape, and texture. The global color function computes the average RGB colors within the entire image for both the dominant color and the variation of color throughout the entire image. Similarity is based on the three average color values. The local color function computes the color distribution for both the dominant color and the variation for each image in a predetermined 256 color space. Image similarity is based on the similarity of the color distribution. The shape function analyses images for combinations of area, circularity, eccentricity, and major axis orientation. All shapes are assumed to be non-occluded planar shapes allowing each shape to be represented as a binary image. The texture function analyses areas for global coarseness, contrast and directionality features.

3.2: Video Indexing and Retrieval

As compared to content-based image indexing and retrieval, which has been an active area of research since the 1970's, the field of content-based access to video repositories is still gaining due attention. As discussed at the beginning of section 3.1, human visual perception displays a gradient of sophistication, ranging from seemingly primitive inferences of shapes, textures, colors, etc. to complex notions of structures such as chairs, trees, affordances, and to cognitive processes such as recognition of emotions and feelings. Given the multidisciplinary nature of the techniques for modeling, indexing and retrieval of visual data, efforts from many different communities have merged in the advancement of content-based video indexing and retrieval (CBVIR) systems. Depending on the background of the research teams, different levels of abstractions have been assumed to model the data. As shown in Figure 7, we classify these abstractions into three categories based on the gradient model of human visual perception specifically in the context of CBVIR systems. In this figure, we also capture the mutual interaction of some of the disciplines of engineering, computer science, and cognitive sciences.

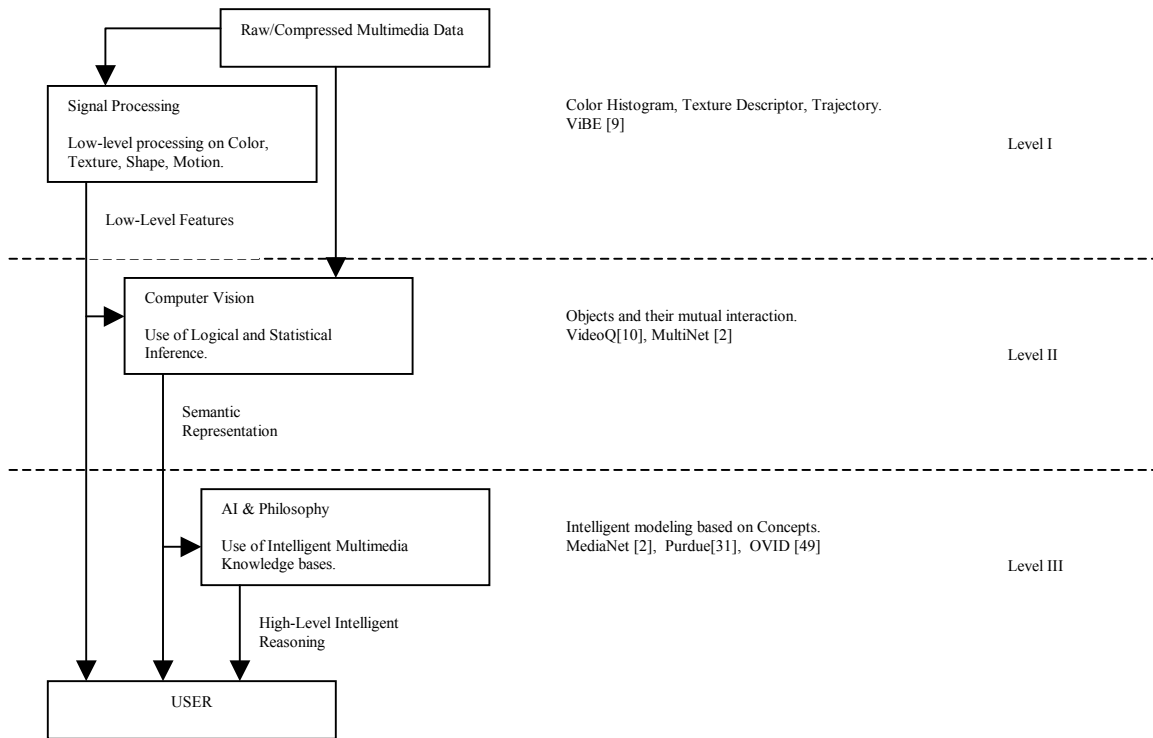


Figure 7: Classification of Content Modeling Techniques. Level I: Low-level physical modeling of Raw Video Data. Level II: Representation of derived or logical features. Level III: Semantic level abstractions.

Level I represents systems that model raw video data using features such as color histogram, shape and texture descriptors, or trajectory of objects. This model can be used to serve the queries like “shots of object with dominant red color and moving from left corner to right”. CBVIR systems based on these models operate directly on the data, employing techniques from signal processing domain. Level II consists of derived or logical features involving some degree of statistical and logical inference about the identity of objects depicted by visual media. An example query at this level can be “shots of Sears Tower”. Using these models, systems normally operate on low-level feature representation, though they can also use video data directly. Level III deals with semantic abstractions involving a significant amount of high-level reasoning about the meaning and purpose of the objects or scenes depicted. An example of a query at this level can be “shots depicting human suffering or sorrow”. As indicated at Level III of the figure, the artificial intelligence (AI) community has had the leading role in this effort. Systems at this level can take semantic representation based on input generated at level II.

Despite the diversity in modeling and application of CBVIR systems, most systems usually rely on similar video processing modules. In the following subsections, we explore some of the broad classes of modules typically used in a content-based video indexing and retrieval system.

3.2.1: Temporal Segmentation

Video data can be viewed hierarchically, where at the lowest level, video data is made up of *Frames*; a collection of frames that result from single camera operation depicting one event is called a *Shot*; a complete unit of narration which consists

of a series of shots or a single shot that takes place in a single location and that deals with a single action defines a *Scene* [27]. CBVIR systems rely on the visual content at distinct hierarchical levels of the video data. Although the basic representation of raw video is provided in terms of a sequence of frames, the detection of distinct shots and scenes is a complex task.

Transitions or boundaries between shots can be abrupt (Cut) or they can be gradual (Fade, Dissolve, Wipe). Traditional temporal segmentation techniques have focused on cut detection, but there has been increasing research activity on gradual shot boundary detection as well. Most of the existing techniques reported in the literature detect shot boundary by extracting some form of feature for each frame in the video sequence, then evaluating a similarity measure on features extracted from successive pairs of frames in the video sequence, and finally declaring the detection of a shot boundary if the feature difference conveyed by the similarity measure exceeds a threshold. One such approach is presented in [45] in which two difference metrics, Histogram Distance Metric (HDM) and Spatial Distance Metric (SDM), are computed for every frame pair. HDM is defined in terms of 3-channel linearized histograms computed for successive frame pair f_i and f_{i+1} as follows:

$$D_h(f_i, f_{i+1}) = \frac{1}{M \times N} \sum_{j=1}^{256 \times 3} |H_i(j) - H_{i+1}(j)|,$$

where H_i represents the histogram of frame f_i and $M \times N$ is the dimension of each frame. For each histogram, 256 uniform quantization levels for each channel are considered. SDM is defined in terms of the difference in intensity levels between successive frames at each pixel location. Let $I_{i,j}(f_k)$ denote the intensity of a pixel at location (i,j) in the frame f_k , then the spatial distance operator is defined as:

$$d_{i,j}(f_k, f_{k+1}) = \begin{cases} 1 \dots \dots, & |I_{i,j}(f_k) - I_{i,j}(f_{k+1})| > \epsilon \\ 0 \dots \dots, & \text{otherwise} \end{cases}$$

SDM is then computed as follows:

$$D_s(f_k, f_{k+1}) = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N d_{i,j}(f_k, f_{k+1})$$

These two distances are then treated as a 2-D feature vector and an unsupervised K-Means clustering algorithm is used to group shot boundaries into one cluster. For a review of major conventional shot boundary detection techniques, refer to [4] which also provides a comparison between five different techniques based on pixel difference from raw data, DCT coefficients difference and motion compensated difference. Due to the huge amount of data to be processed required in the case of full-frame pixel difference based methods as well as their susceptibility to intensity differences caused by motion, illumination changes, and noise, many novel techniques, beyond the scope of the review presented in [4], have been proposed in both the compressed as well as uncompressed domain. We shall now present a brief overview of some of the recent advances in shot detection.

In [54], a frequency domain correlation approach is proposed. This approach relies on motion estimation information obtained by use of template matching. That is, for each 32×32 block in a given frame, the best matching block in corresponding neighborhood in the next frame is sought by calculating the normalized cross correlation in the frequency domain as:

$$\rho(\varepsilon) = \frac{F^{-1} \left\{ \hat{x}_1(\varpi) \times \hat{x}_2^*(\varpi) \right\}}{\sqrt{\int |\hat{x}_1(\varpi)|^2 d\varpi \cdot \int |\hat{x}_2(\varpi)|^2 d\varpi}},$$

where ε and ϖ are the spatial and frequency coordinate vectors, respectively, $\hat{x}_i(\varpi)$ denotes the Fourier transform of frame $x_i(\varepsilon)$, F^{-1} denotes the inverse Fourier transform operation and $*$ is the complex conjugate. Next, the mean and standard deviation of the correlation peaks for each block in the whole image are calculated and the peaks beyond one standard deviation away from the mean are discarded, thus making the technique more robust to sudden local changes in a small portion of the frame. An average mean is then computed from this pruned data. This average match measure is then compared to the average match of the previous pair and a shot boundary is declared if there is a significant decrease in this similarity match feature.

A novel approach proposed in [38] argues that at the shot boundary, the contents of new shot differ from contents of the whole previous shot instead of just the previous frame. They proposed a recursive Principal Component Analysis- based generic approach, which can be built upon any feature extracted from frames in a shot, and generates a model of the shot trained from features in previous frames. Features from the current frame are extracted and a shot boundary is declared if the features from the current frame do not match the existing model by projecting the current feature onto the existing eigenspace.

In an effort to cut back on the huge amount of data available for processing and emphasizing on the fact that in video shots, while objects may appear or disappear, the background stays much the same and follows the camera motion within one shot, Oh et. al. [51] have proposed a background tracking (BGT) approach. A strip along top, left and right border of the frame, covering around 20% of frame area, is taken as fixed background area (FBA). A signature—1-D vector called transformed background area (TBA)—formed from the Gaussian pyramid representation of the FBA, is computed. Background tracking is achieved by a 1-D correlation matching between two TBAs obtained from successive frames. Shot detection is declared if the background tracking fails as characterized by a decrease in the correlation matching parameter. This approach has been reported to detect and classify both abrupt and gradual scene changes.

Observing the fact that single features can't be used accurately in a wide variety of situations, Delp et. al. [8] have proposed to construct a high-dimensional feature vector, called Generalized Trace (GT), by extracting a set of features from each DC frame. For each frame, GT contains the number of intra coded as well as forward- and backward-predicted macroblocks, histogram intersection of current and previous frames for Y, U and V color components, and standard deviation of Y, U and V components for the current frame. GT is then used in a binary regression tree to determine the probability that each frame is a shot boundary. These probabilities are then used to determine the frames that most likely correspond to the shot boundary.

Hanjalic [20] has put together a nice analysis of the shot boundary detection problem itself, identifying major issues that need to be considered, along with a conceptual solution to the problem in the form of a statistical detector based on minimization of average detection-error probability. The thresholds used in their system are defined at the lower level modules of the detector system. The decision making about the presence of a shot boundary is left solely to a parameter-free detector, where all of the indications coming from different low-level modules are combined and evaluated.

Schonfeld et. al. [32], [34] present a scene change detection method using stochastic sequential analysis theory. The DC data from each frame is processed using Principal Component Analysis to generate a very low-dimensional feature vector Y_k corresponding to each frame. These feature vectors are assumed to form an i.i.d. sequence of multidimensional random vectors having Gaussian distribution. Scene change is then modeled as change in the mean parameter of this distribution. Scene change detection is formulated as a hypothesis testing problem and the solution is provided in terms of a threshold on a generalized likelihood ratio. Scene change is declared at frame k when the maximum value of the sufficient statistic g_k evaluated over frame interval j to k , as:

$$g_k = \max_{1 \leq j \leq k} \left\{ \frac{k-j+1}{2} (X_j^k)^2 \right\},$$

exceeds a preset threshold. Here X_j^k is defined as:

$$X_j^k = \left[(\bar{Y}_j^k - \theta_0)^T \Sigma^{-1} (\bar{Y}_j^k - \theta_0) \right]^{1/2}.$$

In this expression, \bar{Y}_j^k is the mean of feature vectors Y in the current frame interval j to k , and θ_0 is the mean of Y in an initial training set frame interval consisting of M frames. This approach, which is free from human fine-tuning, has been reported to perform equally well for both abrupt and gradual scene changes.

3.2.2: Video Summarization

Once the video clip has been segmented into atomic units based on visual content coherence, the next step is to compactly represent the individual units. This task is the major block in summarizing video content using a table of content approach. It also facilitates efficient matching between two shots at query time for content-based retrieval. Most existing systems represent video content by using one representative frame from the shot, called a *keyframe*. Keyframe-based representation has been recognized as an important research issue in content-based video abstraction. The simplest approach towards this problem is to use the first frame of each shot as a keyframe [44]. Although the approach is simple, it is limited since each shot is allotted only one frame for its representation irrespective of the complexity of the shot content. Also, the choice of the first frame over other frames in the shot is arbitrary. In order to have more flexibility in keyframe-based representation of a video shot, Zhang et. al. [68] propose to use multiple frames to represent each shot. They use criteria such as color content change and zoom-in type of effects in shot content to decide on the keyframes to be used to represent each shot. A technique for shot content representation and similarity measure using subshot extraction and representation is presented in [37]. This approach uses two content descriptors, Dominant Color Histogram (DCH) and Spatial Structure Histogram (SSH), to measure content variation and to represent subshots. They first compute the quantized HSV (Hue, Saturation and Value) color histogram for each frame. Next, the dominant local maxima positions in each frame's histogram are identified and tracked throughout the shot. After tracking, only the colors with longer durations are retained as dominant colors of the shot. Histogram bins are finally weighted by the duration of each bin in the whole shot. SSH is computed based on spatial information of color-blobs. For each blob, histograms are computed for the area, position, and deviation.

Delp et. al. [9] represent a shot using a tree structure called a *shot tree*. This tree is formed by an agglomerative clustering technique performed on individual frames in a shot. Starting at the lowest level with each frame representing a

cluster, the algorithm iteratively combines the two most similar frames at a particular level into one cluster at the next higher level. The process continues until a single cluster represented by one frame for the whole shot is obtained. This approach unifies the problem of scene content representation for both browsing and similarity matching. For browsing, only the root node of the tree (keyframe) is used, while for similarity matching two or three levels of tree can be used, employing standard tree matching algorithms.

Another approach to video summarization based on a low-resolution video clip has been proposed by Lelescu and Schonfeld [35]. In this approach, a low-resolution video clip is provided by an efficient representation of the DC frames of the video shot using an iterative algorithm for the computation of principal component analysis (PCA). Efficient representation of the DC frames is obtained by their projection onto the eigenspace characterized by the dominant eigenvectors for the video shot. The eigenvectors obtained by PCA can also be used for conventional keyframe representation of the video shot by considering the similarity of frames in the video shot to the eigenvectors with the largest eigenvalues.

3.2.3: Compensation for Camera and Background Movement

The motion content in a video sequence is the result of either camera motion (pan, zoom, tilt) or object and background motion. Panning motion of a camera is defined as rotation along the horizontal axis while tilt is rotation along the vertical axis. The major concern in motion content-based video indexing and retrieval is almost always the object's motion and not the camera effects. This is because while querying video indexing and retrieval systems, users tend to be more interested in the maneuvering of the objects in the scene, and not in the way camera is being tilted, rotated or zoomed with respect to the object. The problem is that true motion of object cannot be assessed unless the camera motion is compensated for. This problem always arises in case of a video of a moving object -recorded with a mobile camera. Similarly, quite often we have object motion as well as background movement in the video sequence. In such -cases, background movement needs to be differentiated from object movement to give the true object motion. Once these motions have been separated, the object trajectory can be obtained and video data can be indexed based on this motion cue along with other features.

Oh et. al. [50] express different motions accurately by estimating motions from the camera and object. First, they measure the total motion (TM) in a shot. This is achieved by computing the accumulated quantized pixel differences on all pairs of frames in the shot. Before computing pixel differences, the color at each pixel location is quantized into 32 bins to reduce the effect of noise. Once the total motion has been estimated, each frame in the shot is checked for the presence of camera motion -. They detect - pan and tilt of camera, and if present, the amount and direction of camera motion is computed. Object motion (OM) is computed by a technique similar to the computation of TM, after compensation of camera motion -. Bouthemy et. al. [5] address the problem of shot change detection as well as camera motion estimation in a single framework. The two objectives are met by computing, at each time instant, the dominant motion in the image sequence represented by a 2D affine motion model. From each frame pair in the video sequence, a statistical - module estimates the motion model parameters-and the support for motion (i.e., the area of motion) in the successive frame. A least-squares motion estimation module - then computes the confidence of the motion model and maps significant motion parameters onto predefined camera motion classes. These classes include pan, tilt, zoom and many of their combinations.

3.2.4: Feature-based modeling

Most of the contribution to- the problem of content-based video indexing from the signal processing community has been in the direction of modeling visual content by using low-level features. Since video is formed by a collection of images, most of the techniques that model visual content rely on extracting image-like features from the video sequence. Visual features can be extracted from keyframes or the sequence of frames after the video sequence has been segmented into shots. In this section, we analyze different low-level features that can be used to represent the visual content of a video shot.

3.2.4.1: Temporal Motion Features

Video is a medium which is very rich in dynamic content. Motion stands out as the most distinguishing feature to index video data. Motion cue is hard to extract since computation of the motion trail often involves generation of optical flow. The problem of computing optical flow between successive frames of the image sequence is recognized to be computationally intensive, so few systems use motion cue to a full extent. The optical flow represents a two-dimensional field of instantaneous velocities corresponding to each pixel in the image sequence. Instead of computing the flow directly on image brightness values, it is also possible to first process the raw image sequence for contrast, entropy or spatial derivatives. The computation of optical flow can then be performed on these transformed pixel brightness values instead of the original images in an effort to reduce the computational overhead. In either case, a relatively dense flow field is obtained at each pixel in the image sequence. Another approach to the estimation of object motion in the scene can be performed by using a feature-matching based method. This approach involves computation of relatively sparse but highly discriminatory features in a frame. The features can be points, lines or curves and are extracted from each frame of the video sequence. Inter-frame correspondence is then established between these features in order to compute the motion parameters in the video sequence.

Pioneering work in using motion to describe video object activity has been presented by Dimitrova and Golshani [13], who use macroblock tracing and clustering to derive trajectories and then compute similarity between these raw trajectories. In , they have proposed a three-level motion analysis methodology. Starting from the extraction of trajectory of a macro-block in an MPEG video, followed by averaging all trajectories of the macro-blocks of objects, and finally relative position and timing information among objects, a dual hierarchy of spatio-temporal logic is established for representing video. More recently, Schonfeld and Lelescu [57] have developed a video tracking and retrieval system known as *VORTEX*. In this system, a bounding box is used to track an object throughout the compressed video stream. This is accomplished by exploiting the motion vector information embedded in the coded video bitstream. A k-means clustering of the motion vectors is used to avoid occlusions. An extension of this approach to the tracking of the boundary of the object in the raw video stream has been presented in [58]. After initial segmentation of the object contour, an adaptive block matching process is used to predict the object contour in successive image sequences.

Further research has also been devoted to the indexing and retrieval of object trajectories. One such system that makes use of low-level features extracted from objects in the video sequence with particular emphasis on object motion is VideoQ [10]. Once the object trajectory has been extracted, modeling of this motion trail is essential for indexing and retrieval applications. A trajectory in this sense is a set of 2-tuples $\{(x_k, y_k) : k = 1, \dots, N\}$, where (x_k, y_k) is the location of the object's centroid in the k-th frame and the object has been tracked for a total of N frames. The trajectory is treated as separable in its x and y-coordinates and the two are processed separately as 1-D signals. VideoQ models object trajectory based on physical features like acceleration, velocity, and arc length. In this approach, the trajectory is first segmented into smaller units called

subtrajectories. The motivation of this is two-fold. First, modeling of full object trajectories can be very computationally intensive. Second, there might be many scenarios where a part of the object trajectory is not available due to occlusion, etc. Also, the user might be interested in certain partial movements of the objects. Physical feature-based modeling is used to index each subtrajectory using acceleration, velocity, etc. These features are extracted from the original subtrajectory by fitting it with a second-order polynomial as in the following equation:

$$r(t) = (x(t), y(t)) = 0.5at^2 + v_0t,$$

$$a = (a_x, a_y) = \text{acceleration}, v_0 = (v_x, v_y) = \text{velocity}.$$

where $r(t)$ is the parametric representation of the object trajectory.

3.2.4.2: Spatial Image Features

Low-level image representation features can be extracted from keyframes in an effort to efficiently model the visual content. At this level, any of the techniques from representation of image indexing schemes can be used. The obvious candidates for feature space are color, texture, and shape. Thus, features used to represent video data have conventionally been the same ones used for images, extracted from keyframes of the video sequence, with - additional motion features used to capture temporal aspects of video data. In [46], Nephade et. al. first segment the video spatio-temporally obtaining regions in each shot. Each region is then processed for feature extraction. They use a linearized HSV histogram having 12 bins per channel as the color feature. The HSV color space is used because it is perceptually closer to human vision as compared to the RGB space. The three histograms corresponding to the three channels (hue, saturation, and value) are then combined into one vector of dimension 36. Texture is represented by gray-level co-occurrence matrices at four orientations. Also, shape is captured by moment invariants. A similar approach proposed by Shih-Fu Chang et. al. [10] uses quantized CIE-LUV space as the color feature, three Tamura texture measures (coarseness, contrast, and orientation) as texture feature, as well as shape components and motion vectors. All these features are extracted from objects detected and tracked in video sequence after spatio-temporal segmentation.

3.2.5: High-level Semantic Modeling

As pointed out earlier, higher level indexing and retrieval of visual information, as depicted at level II or level III in Figure 1, requires semantic analysis that is beyond the scope of many of the low-level feature-based techniques. One important consideration that many existing content modeling schemes overlook is the importance of the multi-modal nature of video data comprising of a sequence of images along with associated audio and, in many cases, textual captions. Fusing data from multiple modalities improves the overall performance of the system. Many of the content modeling schemes based on low-level features work on Query By Example (QBE) paradigm in which the user is required to submit a video clip or an image illustrating the desired visual features. At times this constraint becomes prohibitive when an example video clip or image depicting what the person is seeking is not at hand. Query By Keyword (QBK) offers an alternative to QBE in the high-level semantic representation. In this scenario, a single keyword or a combination of many can be used to search through the video database. However, this requires more sophisticated indexing because keywords summarizing the video content need to be generated during the indexing stage. This capability can be achieved by incorporating knowledge base into video indexing and

retrieval systems. There has been a drive towards incorporating intelligence into CBVIR systems and we will look into some intelligence-based ideas and systems in this section. Modeling video data and designing semantic reasoning-based Video Database Management Systems (VDBMSs) facilitate high-level querying and manipulation of video data. A prominent issue associated with this domain is development of formal techniques for semantic modeling of multimedia information. Another problem in this context is the design of powerful indexing, searching, and organization methods for multimedia data.

3.2.5.1: Multimodal Probabilistic Frameworks

Multimedia indexing and retrieval presents a challenging task of developing algorithms that fuse information from multiple media to support queries. Content modeling schemes operating in this domain have to bridge the gap between low-level features and high-level semantics often called the semantic gap. This effort has to take into account the information from audio as well as from video sources. Nephade et. al. [46] have proposed the concept of *Multiject*, a Multimedia Object. A multiject is the high-level representation of a certain object, event, or site having features from audio as well as from video. It has a semantic label, which describes the object in words. It also has associated multi-modal features (including both audio and video features) which represent its physical appearance. It has an associated probability of occurrence in conjunction with other objects in the same domain (shot). Experiments using multijects concepts from three main categories of objects (e.g., airplane), sites (e.g., indoor) and events (e.g., gunshot) have been conducted. Given the multimodal feature vector \vec{X}_j of the j^{th} frame and assuming uniform priors on the presence or absence of any concept in any region, the probability of occurrence of each concept in the j^{th} frame is obtained from Bayes' rule as:

$$P(R_{ij} = 1 | \vec{X}_j) = \frac{P(\vec{X}_j | R_{ij} = 1)}{P(\vec{X}_j | R_{ij} = 1) + P(\vec{X}_j | R_{ij} = 0)},$$

where R_{ij} is a binary random variable taking value 1 if the concept i is present in frame j . During the training phase, the identified concepts are given labels and the corresponding Multiject consists of a label along with its probability of occurrence and multimodal feature vector. Multijects are then integrated at the frame level by defining frame level features $F_i, i \in \{1 \dots N\}$ (N is the number of concepts the system is being trained for) in the same way as for R_{ij} . If M is the number of regions in the current frame, then given $\chi = \{\vec{X}_1, \dots, \vec{X}_M\}$, the conditional probability of Multiject i being present in any region in the current frame is:

$$P(F_i = 1 | \chi) = \max_{j \in \{1, \dots, M\}} P(R_{ij} = 1 | \vec{X}_j).$$

Observing the fact that semantic concepts in videos do not appear in isolation, but rather interact and appear in context, their interaction is modeled explicitly and a network of multijects, called *Multinet* is proposed [47]. A framework based on multinet takes into account the fact that presence of some multijects in a scene boosts the detection of other semantically related multijects, and reduces the chances for others. Based on this multinet framework, spatio-temporal constraints can be imposed to enhance detection, support inference, and impose a priori information.

3.2.5.2: Intelligence-Based Systems

The next step towards future CBVIR systems will be marked by the introduction of intelligence into the systems as they need to be capable of communicating with the user, understanding audio-visual content at a higher semantic level, and reasoning and planning at a human level. Intelligence is referred to as the capabilities of the system to build and maintain situational or world models, utilize dynamic knowledge representation, exploit context, and leverage advanced reasoning and learning capabilities. An insight into human intelligence can help better understand users of CBVIR systems and construct more intelligent systems. Ana et. al. [2] propose an intelligent information system framework, known as *MediaNet*, which incorporates both perceptual and conceptual representations of knowledge based on multimedia information in a single framework by augmenting the standard knowledge representation frameworks with the capacity to include data from multiple media. It models the real world by concepts, which are real world entities and relationships between those concepts that can be either semantic (car Is-A-Subtype-Of vehicles) or perceptual (donkey Is-Similar-To mule). In *MediaNet*, concepts can be as diverse-natured as living entities (Humans), inanimate objects (Car), events in the real world (Explosion), or certain property (Blue). Media representation of the concepts involves data from heterogeneous sources. Multimodal data from all such sources is combined using the framework which intelligently captures the relationships between its various entities.

3.2.5.3: Semantic Modeling and Querying of Video Data

Owing to its distinguished characteristics from textual or image data – very rich information content, temporal as well as spatial dimensions, unstructured organization, massive volume, and complex and ill-defined relationship among entities – robust video data modeling is an active area of research. The most important issue that arises in the design of Video Database Management Systems (VDBMSs) is the description of structure of video data in a form appropriate for querying, sufficiently easy for updating, and compact enough to capture the rich information content of the video. The process of designing the high-level abstraction of raw video to facilitate various information retrieval and manipulation operations is the crux of VDBMSs. To this end, current semantic-based approaches can be classified into *segmentation-based* and *stratification-based*. The drawback of the former approaches is lack of flexibility and incapability of representing semantics residing in overlapping segments. The latter models, however, segment contextual information of video instead of simply partitioning it.

SemVideo [62] presents a video model in which semantic content having unrelated time information are modeled as ones that do; also, not only is the temporal feature used for semantic descriptions, but also the temporal relationships among themselves are components of the model. The model encapsulates information about *Videos*, each being represented by a unique identifier; *Semantic Objects*, description of knowledge about video having a number of attribute-value pairs; *Entities*, any of the above two; *Relationships*, an association between two entities. Many functions are also defined that help in organizing data and arranging relations between different objects in the video. Tran et. al. [63] propose a graphical model, *VideoGraph*, that supports not only the Event Description, but also Inter-Event Description that describes the temporal relationship between two events – a functionality overlooked by most of the existing video data models. They also have a provision for exploiting incomplete information by associating the temporal event with a Boolean-like expression. A query language based on their framework is proposed in which query processing involves only simple graph traversal routines.

Khokhar et. al. [31] introduce a multi-level architecture for video data in which semantics are shared among various levels. An object-oriented paradigm is proposed for management of information at higher levels of abstraction. For each video sequence to be indexed, they first identify objects inside the video sequence, their sizes and locations, their relative positions

and movements, and this information is finally encoded in a spatio-temporal model. Their approach integrates both intra- and inter-clip modeling and uses both bottom-up as well as top-down object-oriented data abstraction concepts. Decler et. al. [12] have developed a data model that goes one step beyond the existing stratification-based approaches using *Generalized intervals*. Here instead of a time segment to be associated with a description, a set of time segments is associated with a description – an approach that allows handling with a single object all occurrences of an entity in a video document. They also propose a declarative, rule-based, constraint query language that can be used to infer relationships from information represented in the model, and to intentionally specify relationships among objects.

References

- [1] Arkin E.M., Chew L., Huttenlocher D., Kedem K., Mitchell J., “An efficiently computable metric for comparing polygonal shapes”, IEEE Trans. On Patt. Recog. & Mach. Intell., 13(3), March 1991.
- [2] Benitez A.B., Smith J.R., and Chang S.F., “MediaNet: A Multimedia Information Network for Knowledge Representation”, Proceedings of the SPIE 2000 Conference on Internet Multimedia Management Systems (IS&T/SPIE-2000), Vol. 4210, Boston, MA, Nov 6-8, 2000.
- [3] Bach J.R., Fuller C., Gupta A., Hampapur A., Horwitz B., Humphrey R., Jain R., Shu C.F., “The Virage image search engine: An open framework for image management”, Proc. SPIE Storage and Retrieval for Image and Video Databases.
- [4] Borecsky J.S., Rowe L.A., “Comparison of video shot boundary detection techniques”, In Proceedings of SPIE, vol. 26670, pages 170-179, 1996.
- [5] Bouthemy P., Gelgon M., Ganansia F., “A unified approach to shot change detection and camera motion characterization”, Research Report IRISA, No 1148, November 1997.
- [6] Carson C., Belongie S., Greenspan H., Malik J., “Region-based Image Querying”, CVPR '97 workshop on content-based access of image and video libraries.
- [7] Chambers, J.M., “Computational Methods for Data Analysis”, Wiley, New York, 1977.
- [8] Chen J-Y, Taskiran C., Albiol A., Delp E.J., Bouman C.A., “ViBE: A Compressed Video Database Structured for Active Browsing and Search”, submitted to IEEE Transactions on Multimedia 2001.
- [9] Chen J-Y, Taskiran C., Albiol A., Delp E.J., Bouman C.A., “ViBE: A Compressed Video Database Structured for Active Browsing and Search”, submitted to IEEE Transactions on Multimedia 2001.
- [10] Chang S.F., Chen H., Meng J., Sundaram H., Zhong D., “A Fully Automated Content-Based Video Search Engine Supporting Spatiotemporal Queries”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 5, September 1998.
- [11] Catarci T., Costabile M.F., Leviardi S., Batini C., "Visual Query Systems for Databases: A Survey," Technical Report Rapporto di Ricerca SI/RR 95/17, Dipartimento di Scienze dell'Informazione, Universita degli Studi di Roma, October 1995.
- [12] Declair C., Hacid M.H., Kouloumdjian J., “A Database Approach for Modeling and Querying Video Data”, 15th International Conference on Data Engineering, Sydney, Australia, 1999.
- [13] Dimitrova N., Golshani F., “Motion recovery for video content classification”, ACM Transactions on Information Systems, 13:4, Oct. 1995, pp. 408 – 439.
- [14] Dimitrova N., Golshani F., “Px for Semantic Video Database Retrieval”, Proceeding of the ACM Multimedia'94, San Francisco, CA, pp. 219-226.
- [15] Foote J., "An Overview of Audio Information Retrieval," in *Multimedia Systems*, vol. 7 no. 1, pp. 2-11, ACM Press/Springer-Verlag, January 1999.
- [16] Goodrum A. A., “Image Information Retrieval: An Overview of Current Research”, Informing Science, Special Issue on Information Sciences, Vol 3 No 2, 2000.
- [17] Gonzalez R.C., Woods R.E., “Digital image processing”, Addison Wesley Longman , Inc., 1992.
- [18] H.261: “ITU-T Recommendation H.261, video codec for audiovisual services at px64 kbits/sec”, Geneva (1990).
- [19] Draft ITU-T Recommendation H.263, “Video coding for low bit rate communication”, July 1995.
- [20] Hanjalic A., “Shot-Boundary Detection: Unraveled and Resolved?”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 2, February 2002.
- [21] Hu M.K., “Visual pattern recognition by moment invariants, computer methods in image analysis”, IRE transactions on Information theory, 8, 1962.

- [22] Haralick R.M., Shanmugam K., Dinstein I., "Texture features for image classification", IEEE Trans. On Sys, Man, and Cyb, SMC-3(6), 1973.
- [23] Iqbal Q., Aggarwal J.K., "Using structure in content-based image retrieval", Proc. Of the IASTED international conference Signal and Image Processing (SIP), Oct. 18-21, 1999, Nassau, Bahamas, pp. 129-133.
- [24] Hibino, S. & Rundensteiner, E. (1995), "A Visual Query Language for Identifying Temporal Trends in Video Data", Proc. of the 1995 International Workshop on Multi-Media Database Management Systems. Los Alamitos, CA: IEEE Society Press, 74-81.
- [25] ITU, "Information Technology – Digital Compression & Coding of Continuous Tone Still Images-Requirements and Guidelines. T.81", ITU, 1993.
- [26] Jolliffe, I.T., "Principal Component Analysis", Springer-Verlag, New York, 1986.
- [27] Monaco J., "How to Read a Film: The Art, Technology, Language, History, and Theory of Film and Media", Oxford University Press, New York, NY, 1977.
- [28] Khokhar A., Ansari R., Malik H., "Content-Based Audio Indexing and Retrieval: An Overview", Tech. Report, Multimedia Systems Lab, UIC, Multimedia-2003-1.
- [29] Khokhar A., Albuz E., Kocalar E., "Quantized CIELab* Space and Encoded Spatial Structure for Scalable Indexing of Large Color Image Archives", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2000, ICASSP '00, Volume: 6, 2000.
- [30] Khokhar A., Albuz E., Kocalar E., "Vector Wavelet based Image Indexing and Retrieval for Large Color Image Archives," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99), March 1999.
- [31] Khokhar A., Day Y.F., Ghafoor A., "A Framework for Semantic Modeling of Video Data for Content-Based Indexing and Retrieval", ACM Multimedia, 1999.
- [32] Lelescu D., Schonfeld D., "Real-time scene change detection on compressed multimedia bitstream based on statistical sequential analysis," Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 1141-1144, New York, New York, 2000.
- [33] Kaushik S., Rundensteiner E. A. "SVIQUER: A Spatial Visual Query and Exploration Language" 9th Intern. Conf. on Database and Expert Systems Applications - DEXA'98, LNCS N. 1460, pp. 290-299, 1998.
- [34] Lelescu D., Schonfeld D., "Statistical sequential analysis for real-time scene change detection on compressed multimedia bitstream," IEEE Transactions on Multimedia, to appear, 2003.
- [35] Lelescu D., Schonfeld D., "Video skimming and summarization based on principal component analysis," Proceedings of the IFIP/IEEE International Conference on Management of Multimedia Networks and Services, pp. 128-141, Chicago, Illinois, 2001. Also appeared in Management of Multimedia on the Internet, Lecture Notes in Computer Science, E.S. Al-Shaer and G. Pacifici (Eds.), Springer-Verlag, pp. 128-141, 2001.
- [36] Liang K.C., JayKuo C.C., "WaveGuide: A joint wavelet based image representation and description system", IEEE transactions on image processing, vol. 8, no. 11, November 1999.
- [37] Lin T., Zhang H.J., and Shi Q-Y, "Video Content Representation for Shot Retrieval and Scene Extraction", International Journal of Image & Graphics, Vol. 1, No. 3, July 2001.
- [38] Liu X.M., Chen T., "Shot Boundary Detection Using Temporal Statistics Modeling", ICASSP 2002., Orlando, FL, U.S.A., May 2002.
- [39] MPEG-1: "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbps", ISO/IEC 1117-2: Video, (November 1991).
- [40] MPEG-2: "Generic coding of moving pictures and associated audio information", ISO/IEC 13818-2 Video, Draft International Standard, (November 1994).
- [41] MPEG-4 video verification model version-11, ISO/IEC JTC1/SC29/WG11, N2171, Tokyo, (March 1998).
- [42] Mehrotra S., Chakrabarti K., Ortega M., Rui Y., Huang T.S., "Multimedia Analysis and Retrieval System", Proc. Of 3rd Int. workshop on Information Retrieval Systems, 1997.

- [43] Ma W.Y., Manjunath B.S., “Netra: A toolbox for navigating large image databases”, Proc. IEEE Intl. Conf. On Image Processing, 1997.
- [44] Nagasaka A., Tanaka Y., “Automatic video indexing and full-video search for object appearances”, in Visual Database Systems II, 1992.
- [45] Naphade M.R., Mehrotra R., Fermant A. M., Warnick J., Huang T.S., Tekalp A. M., “A High Performance Shot Boundary Detection Algorithm using multiple cues”, Proc. I.E.E.E. International Conference on Image Processing, Volume 2, pages 884-887, Oct 1998, Chicago, IL.
- [46] Naphade M.R., Kristjansson T., Frey B., Huang T.S., “ Probabilistic Multimedia Objects Multijets: A novel Approach to Indexing and Retrieval in Multimedia Systems”, Proc. I.E.E.E. International Conference on Image Processing, Volume 3, pages 536-540, Oct 1998, Chicago, IL.
- [47] Naphade M.R., Kozintsev I.V., Huang T.S., “A Factor Graph Framework for Semantic Video Indexing”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 1, January 2002.
- [48] Niblack W., Zhu X., Hafner J. L., Breuel T., Ponceleon D. B., Petkovic D., Flickner M. D., Upfal E., Nin S. I., Sull S., Dom B. E., Yeo B-L, Srinivasan S., Zivkovic D., Penner M., “Updates to the QBIC System”, In Proceedings of Storage and Retrieval for Image and Video Databases VI. San Jose, California, USA, SPIE, 1997.
- [49] Oomoto E., Tanaka K., “OVID: Design and Implementation of a Video-Object Database System”, IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 4, August 1993, pp. 629-643.
- [50] Oh J., Chowdary T., “An efficient technique for measuring of various motions in video sequences”, Proceedings of 2002 International Conference on Imaging Science, Systems and Technology (CISST’02), Las Vegas, NV, June 2002.
- [51] Oh J.H., Hua K.A., Liang N., “A Content-based Scene Change Detection and Classification Technique using Background Tracking”, Proc. of IS&T/SPIE conference on Multimedia Computing and Networking 2000. pp. 254-265 Jan. 24 - 28, 2000, San Jose, CA.
- [52] Pentland A., Picard R.W., Sclaroff S., “PhotoBook: Content-based manipulation of image databases”, International Journal of Computer Vision, 1996.
- [53] Pentland A., Picard A.W., Sclaroff S., “PhotoBook: content-based manipulation of image databases”, International Journal of Computer Vision, 1996.
- [54] Porter S. V., Mirmehdi M., Thomas B.T., “Video Cut Detection using Frequency Domain Correlation”, In Proceedings of the 15th International Conference on Pattern Recognition, pages 413--416. IEEE Computer Society, September 2000.
- [55] Rui Y., Huang T.S., Ortega M., Mehrotra S., “Relevance feedback: A power tool in interactive content-based image retrieval”, IEEE Trans. On Circuits and Systems for Video Technology, Special issue on Interactive Multimedia Systems for the internet, Sept. 1998.
- [56] Shen B., Sethi I.K., “Direct feature extraction from compressed images”, SPIE vol. 2670. Storage & Retrieval for Image and Video Databases IV, 1996.
- [57] Schonfeld D., Lelescu D., “VORTEX: Video retrieval and tracking from compressed multimedia databases—multiple object tracking from MPEG-2 bitstream,” (Invited Paper). Journal of Visual Communications and Image Representation, Special Issue on Multimedia Database Management, vol. 11, pp. 154-182, 2000.
- [58] Schonfeld D., Hariharakrishnan K., Raffy P., Yassa F., “Object tracking using adaptive block matching,” Proceedings of the IEEE International Conference on Multimedia and Expo, Baltimore, Maryland, 2003, to appear.
- [59] Smith J.R., Chang S.F., “Tools and techniques for color image retrieval”, in IS & T/SPIE proceedings Vol. 2670, Storage & Retrieval for image and video databases IV, 1995.
- [60] Smith J. R., Chang S. F., “VisualSeek: A fully automated content-based image query system”, Proc. ACM Multimedia 96, 1996.
- [61] Smith J.R., Chang S.F., “Visually searching the web for content”, IEEE Multimedia Magazine, 4(3):12-20, Summer 1997.
- [62] Tran D. A., Hua K. A., Vu K. “Semantic Reasoning based Video Database Systems”, Proc. of the 11th Int'l Conf. on Database and Expert Systems Applications, pp. 41-50, September 4-8, 2000, London, England.

- [63] Tran D.A., Hua K.A., and Vu K. "VideoGraph: A Graphical Object-based Model for Representing and Querying Video Data", In the proc. of ACM Int'l Conference on Conceptual Modeling (ER 2000), October 9-12, Salt Lake city, USA.
- [64] Tamura H., Mori S., and Yamawaki T., "Texture features corresponding to visual perception", IEEE Trans. On Sys, Man, and Cyb, SMC-8(6), 1978.
- [65] Venters C.C., Cooper M., "A Review of Content-based Image Retrieval Systems", University of Manchester, JISC Technology Applications Program (JTAP) report 01/07/00.
- [66] White D., Jain R., "Similarity indexing: Algorithms and performance", in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.
- [67] Zhang C., Meng W.e., Zhang Z., Zhong u.Wu., "WebSSQL: A Query Language for Multimedia Web Documents", Proc. of the IEEE Conference on Advances in Digital Libraries (*ADL'00*) , Washington, D.C., May 2000.
- [68] Zhang H., Wu J., Zhong D., and Smoliar S.W., "An integrated system for content-based video retrieval and browsing", Pattern Recognition, Vol. 30, no.4, pp. 643-658, 1997.