A

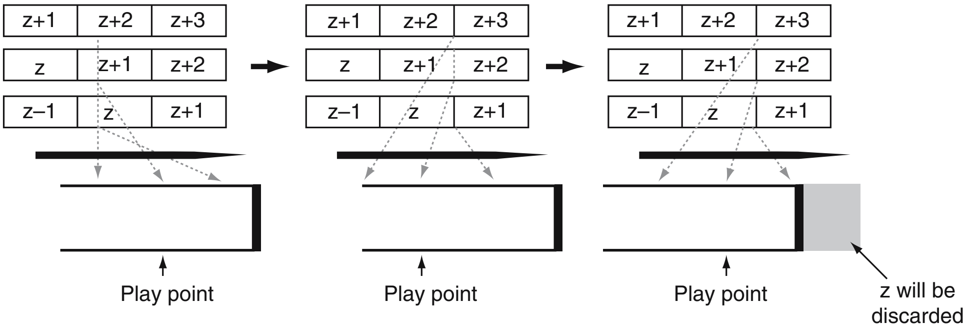
Active Buffer Management for Provision of VCR Functionalities

Definition

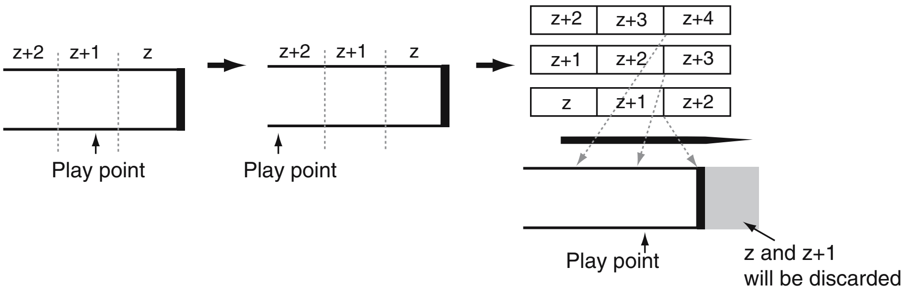
Active buffer management is used to adjust the contents of the buffer after execution of VCR functions in VoD systems.

The problem of providing VCR functions with the traditional buffering schemes is that the effects of VCR actions in the same direction are cumulative. When consecutive VCR actions in the same direction are performed, the play point will ultimately move to a boundary of the buffer. Thus, the active buffer management (ABM) scheme [1] was developed to use a buffer manager to adjust the contents of the buffer after the VCR functions such that the relative position of the play point can stay in the middle of the buffer. Figure 1 shows the basic operational principle of ABM in a staggered VoD system with no VCR actions. Assuming the buffer can hold three segments. At some point, the buffer downloads segments z, z + 1, z + 2 and the play point is in segment z + 1. If there is no VCR action, after a period of time, the play point will be at the start of segment z + 2. At this moment, in order to keep the play point in the middle, the client will download segment z + 3 and segment z will be discarded.

For the scenario with an interactive function, it is assumed that the buffer now holds segment *z*, *z* + 1, and *z* + 2. If a FF action as illustrated in Fig. 2 is issued and the play point moves to the end of segment *z* + 2, the



Active Buffer Management for Provision of VCR Functionalities. Figure 1. Active buffer management scheme without VCR action.



Active Buffer Management for Provision of VCR Functionalities. Figure 2. Active buffer management scheme with FF action.

buffer manager will select segment $z + 3$ and $z + 4$ to download. The play point will thus be moved back to the middle segment after one segment time. This is segment $z + 3$ in this case.

Cross-References

► Large-Scale Video-on-Demand System

References

1. Z. Fei, M.H. Ammar, I. Kamel, and S. Mukherjee, “Providing Interactive Functions for Staggered Multicast Near Video-on-Demand,” Proceedings of IEEE International Conference on Multimedia Computing and Systems ‘99, Vol. 2, 1999, pp. 949–953.

Adaptation Decision Taking Engine

► Utility Model-based Adaptation of Multimedia Content

Adaptive Educational Hypermedia Systems

Definition

Adaptive educational hypermedia systems include adaptive functionality based on three components: the document space, observations, and the user model.

To support re-usability and comparability of adaptive educational hypermedia systems, we give a component-based definition of adaptive educational hypermedia systems (AEHS), extending the functionality-oriented definition of adaptive hypermedia given by Brusilovsky [1]. AEHS have been developed and tested in various disciplines and have proven their usefulness for improved and goal-oriented learning and teaching. However, these systems normally come along as stand-alone systems — proprietary solutions have been investigated, tested and improved to fulfill specific, often domain-dependent requirements. This phenomenon is known in the literature as the *open corpus problem in AEHS* [2] which states that normally, adaptive applications work on a fixed set of documents which is defined at the design time of the system, and directly influences the way adaptation is implemented.

The logical definition of adaptive educational hypermedia given here focuses on the components of these systems, and describes which kind of processing information is needed from the underlying hypermedia system (*the document space*), the runtime information which is required (*observations*), and the user model characteristics (*user model*). *Adaptive functionality* is then described by means of these three components, or more precisely: how the information from these three components, the static data from the document space, the runtime-data from the observations, and the processing-data from the user model, is used to provide adaptive functionality. The given logical definition of adaptive educational hypermedia provides a language for describing adaptive functionality, and allows for the comparison of adaptive functionality in a well-grounded way, enabling the re-use of adaptive functionality in different contexts and systems. The applicability of this formal description has been demonstrated in [3].

An Adaptive Educational Hypermedia System (AEHS) is a Quadruple (DOCS, UM, OBS, AC) with:

- *DOCS* (Document Space): A finite set of first order logic (FOL) sentences with constants for describing documents (and knowledge topics), and predicates for defining relations between these constants.
- *UM* (User Model): A finite set of FOL sentences with constants for describing individual users (user groups), and user characteristics, as well as predicates and rules for expressing whether a characteristic applies to a user.
- *OBS* (Observations): A finite set of FOL sentences with constants for describing observations and predicates for relating users, documents/topics, and observations.
- *AC* (Adaptation Component): A finite set of FOL sentences with rules for describing adaptive functionality.

With the emerging Semantic Web, there is even more the need for comparable, re-usable adaptive functionality. If we consider adaptive functionality as a service on the Semantic Web, we need re-usable adaptive functionality, able to operate on an open corpus, which the Web is.

Cross-References

► Personalized Educational Hypermedia

References

1. P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia," User Modeling and User Adapted Interaction, Vol. 6, No. 2–3, 1996, pp. 87–129.
2. P. Brusilovsky, "Adaptive Hypermedia," User Modeling and User Adapted Interaction, Vol. 11, 2001, pp. 87–110.
3. N. Henze and W. Nejdl, "A Logical Characterization of Adaptive Educational Hypermedia," New Review of Hypermedia, Vol. 10, No. 1, 2004, pp. 77–113.

Adaptive Joint Source-Channel Coding for Scalable Video

Naeem Ramzan, Ebroul Izquierdo
Queen Mary University of London, UK

Synonyms

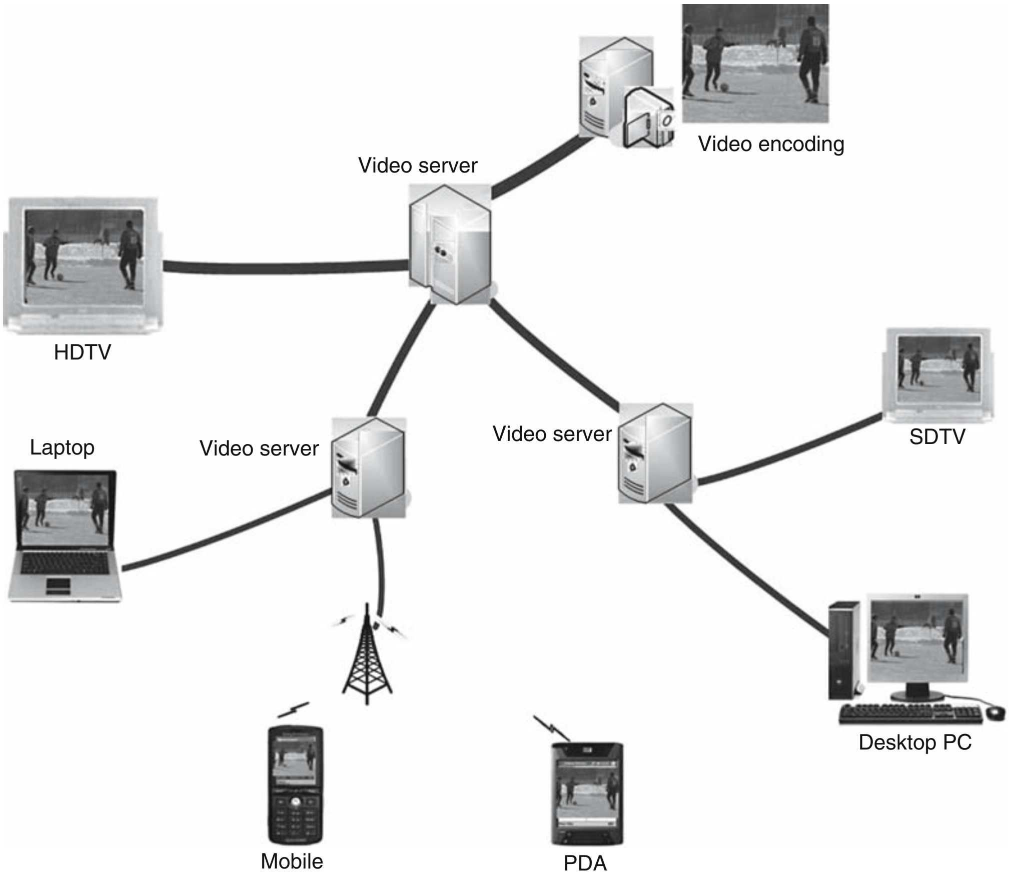
► Joint source-channel coding

Definition

Adaptive joint source-channel coding for scalable video minimizes the reconstructed video distortion at the decoder subject to a constraint on the overall transmission bit rate budget.

Introduction

Adaptive Joint Source-Channel Coding (AJSCC) optimizes the rate allocation for scalable source and channel coding efficiently. There are some scenarios where the AJSCC can play a vital role. Firstly, multi-user environments and multi-user heterogeneous networks are the most likely candidate for the AJSCC, as shown in Fig. 1. Secondly, the AJSCC will have maximal impact, when resource constraints are inflexible. In the scenario depicted in Fig. 1, the video server requires video contents of different fidelities, such as high quality material for storage and future editing and lower bit-rate content for distribution. In traditional



Adaptive Joint Source-Channel Coding for Scalable Video. Figure 1. Advantages of scalable video coding: one video bit-stream serves the different users.

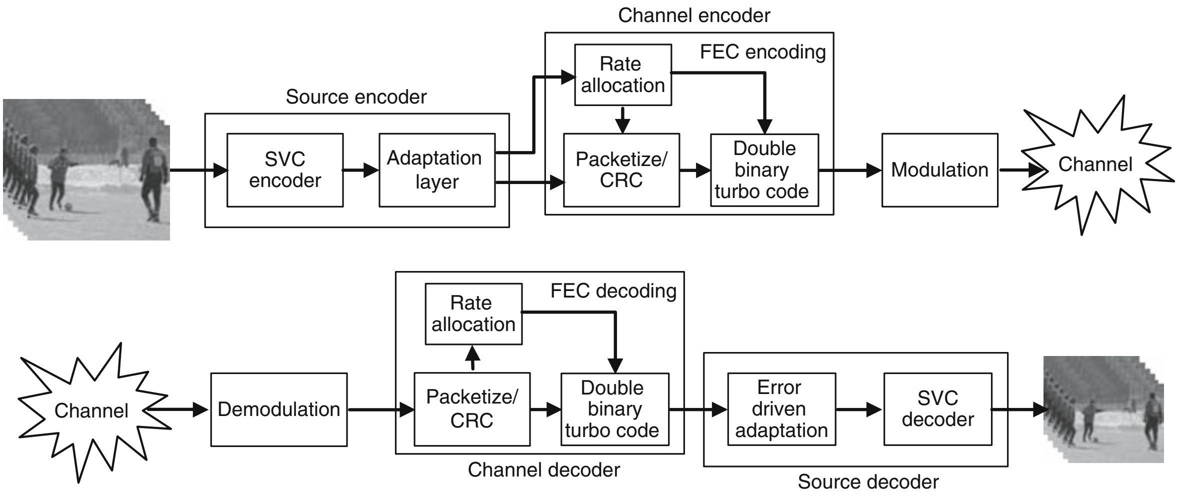
video communications over heterogeneous channels, the video is usually processed offline. Compression and storage are tailored to the targeted application according to the available bandwidth and potential end-user receiver or display characteristics. However, this process requires either transcoding of compressed content or storage of several different versions of the encoded video. None of these alternatives represent an efficient solution. Furthermore, video delivery over error-prone heterogeneous channels meets additional challenges such as bit errors, packet loss and error propagation in both spatial and temporal domains. This has a significant impact on the decoded video quality after transmission and in some cases renders useless the received content. Consequently, concepts such as scalability, robustness and error resilience need to be re-assessed to allow for both efficiency and adaptability according to individual transmission bandwidth, user preferences and terminals.

Scalable Video Coding (SVC) [1] promises to partially solve this problem by "encoding once and decoding many." SVC enables content organization in a hierarchical manner to allow decoding and interactivity at several granularity levels. That is, scalable coded bit-streams can efficiently adapt to the application requirements. The scenario shown in Fig. 1 can truncate the SVC encoded bit-stream at different points and decode it. The truncated bit-stream can be further truncated to some lower resolution, frame rate or quality. Thus, it is important to tackle the problems inherent to the diversity of bandwidth in heterogeneous networks and in order to provide an improved quality of services. SVC provides a natural solution for error-prone transmissions with a truncateable bit-stream.

The transmission of compressed video bit-stream over noisy and fading channels presents several challenging problems that remain to be resolved. Compressed video bit-stream is extremely vulnerable to bit errors. For example, the wired/wireless channels introduce bursts of errors and a vast amount of random errors due to signal attenuation and signal fading effects. In the case of SVC, the errors in the compressed bit-stream may propagate to severe degradation of video quality. Different channel coding methods may be used to make the video bit-stream more resilient to channel errors. In channel coding, turbo codes (TCs) [2] and related iterative-based approaches are currently among the state-of-the-art techniques.

Video Communication System

In a generic video communication chain, the video is first compressed by a source encoder to reduce the data rate. The compressed information is passed through the channel encoding module which protects the information from transmission errors typically using Forward Error Correction (FEC). A video communication system typically involves five modules, namely source coder, channel coder, communication channel, channel decoder and source decoder, as shown in Fig. 2. In the following, we describe these basic elements of the typical SVC digital communication system.



Adaptive Joint Source-Channel Coding for Scalable Video. Figure 2. Video Communication System.

- Source encoder: The source encoder efficiently converts the video into a stream of binary digits called bit-stream. The scalable video encoder organizes the encoded bit-stream in such a way that it can be adapted with respect to different user requirement expediently. This bit-stream is normally highly compressed and very sensitive to transmission errors;
- Channel encoder: The channel encoder protects the highly compressed bit-stream against errors introduced by the noisy communication channels. This is achieved by inserting additional bits called redundancy bits into the compressed bit-stream. The redundancy bits are used at the receiver to detect and possibly correct transmission errors. The output of the channel encoder is binary strings called codewords. A digital modulator is used to convert the codewords into a waveform which is the appropriate form for transmission;
- Communication channel: It is a physical medium through which a signal is sent from transmitter to receiver. It usually causes signal attenuation and introduces signal noise which may lead to severe loss or degradation of the quality of the reconstructed bit-stream;
- Channel decoder: The aim of the channel decoder is to reconstruct the original compressed bit-stream using the redundancy bits inserted by the channel encoder. Both the channel encoder design and the channel noise characteristics are taken into account in the design of the channel decoder;
- Source decoder: The purpose of the source decoder is to reconstruct the video.

Adaptive Joint Source-Channel Coding

Information theory [3] suggests that the source and channel coding can be considered separately but in a practical system, this is not possible. So we need to optimize the source and channel coding characteristic jointly in the AJSCC framework. The scalable source coding and advanced channel coding methods are used to enhance the performance of the system in AJSCC and are described below.

Scalable video coding: SVC encodes the video signal only once and produces a scalable bit-stream which can be truncated at various points according to specific user or channel requirement. In this context, the truncated bit-stream can still be decoded, offering a reduced resolution compared to the resolution provided by the original bit-stream without truncation. A scalable bit-stream can be easily adapted to fulfill different requirements for reconstructed resolution, producing flexible scaling in spatial, temporal, and quality domains. The compressed SVC bit-stream features a highly scalable yet simple structure. The smallest entity in the compressed bit-stream is called an atom, which can be added or removed from the bit-stream. The bit-stream is divided into a Group of Pictures (GOP). Each GOP is composed of a GOP header, the atoms and the allocation table of all atoms. Each atom contains the atom header, motion vectors data (some atom does not contain motion vector data) and texture data of a certain sub-band. In the main header of the SVC bit-stream, the organization of the bit-stream is defined so that the truncation is performed at different user points with low complexity.

Channel Coding: The main purpose of channel coding is to increase the reliability of data transmission. In channel coding, we normally add redundancy to the information data in order to provide error detection and correction capabilities at the receiver. Channel codes could be classified into two major categories: linear block codes; and convolutional codes. The encoder of the block code (n, k) divides the information data into blocks of k bits each and operates on them independently. It adds n-k redundant bits that are algebraically related to the k messages, thereby producing an overall encoded block of n bits called codeword with $n > k$, and $R = k/n$ is called a code rate. The Reed-solomon and LDPC codes are the good example of block code. In contrast to block codes, convolutional codes have memory mr. An (n, k) convolutional encoder converts k information symbols into a codeword of length, which depend not only the k information symbols but also on the previous mr symbols. Nowadays, TCs [2] are one of the best practical channel codes because of their exceptional performance at low Signal to Noise Ratio (SNR). It is based on two convolutional encoders that are separated by an interleaver of length k. One reason for their better performance is that turbo codes produce high weight codewords. Double binary TCs were introduced in the domain of TCs by Doulliard et al. [4]. These codes consist of two binary Recursive Systemic Convolutional (RSC) encoders of rate 2/3 and an interleaver of length k. Each binary RSC encoder encodes a pair of data bits and produces one redundancy bit, so the desired rate 1/2 is the natural rate of the double binary TCs. In this research work, we consider the 8-state double binary TCs with generators in octal notation are (15,13) for y_k.

Both the (RSC) encoders are identical and the data is applied in the form of blocks.

- First of all data is applied in the natural order of data when the switches are at position 1 in Fig. 3;
- Then the interleaved data through interleaver is applied when the switches are in position 2 as shown in Fig. 3.

Adaptive Joint Source-Channel Coding

AJSCC takes the advantage of the different sensitivities of scalable layers of the SVC bit-stream. Before starting the bit-allocation between source and channel coding, we will explain how the bits are arranged in the SVC compressed bit-stream that is exploited in AJSCC. For the sake of visualization and simplicity, the bit-stream can be represented in a $3D$ space with coordinates Q = quality, T = temporal resolution and S = spatial resolution for each GOP, as shown in Fig. 4. There exists a base layer in each domain that is called the 0-th layer and cannot be removed from the bit-stream. Therefore, in the example shown in Fig. 4, three quality, three temporal and three spatial layers are depicted. Each block named as atom has its own coordinates in (q, t, s) space.

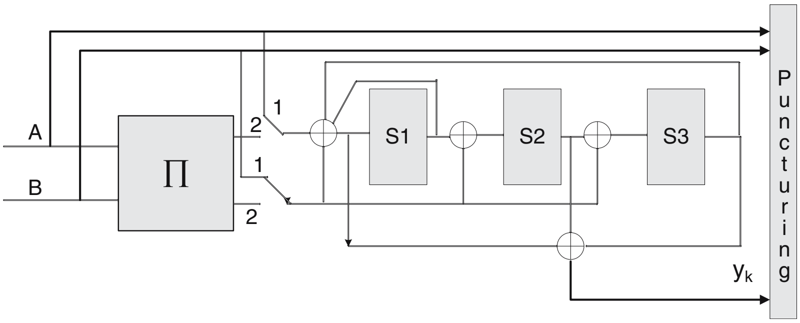
If we select particular subsets of atoms regarding desired quality layers, frame rate and resolution, the selected atoms are interlaced in the embedded bit-stream. Then the bit-stream is divided into parts regarding the domain that progresses most slowly. For example, using the default setting of the SVC, the quality layers progress most slowly. Then we packetize each quality layer using packets of equal length. During the packetizing of the scalable bit-stream, Cyclic Redundancy Check (CRC) bits are added to check the error-free delivery of packets at decoder side. Each quality layer is protected by FEC as shown in Fig. 5.

We use the double binary TC as FEC to adapt the scalable video bit-stream. At the decoder output, the double binary TC's bit-rate is twice that of the binary TC because it decodes two bits at the same number of turbo iterations. Double binary TC is less sensitive to puncturing so we can easily use unequal error protection for scalable bit-stream with CRC bits.

Let us now formally state the problem. Suppose Q are the total quality layers in the SVC bit-stream and the error protection scheme for the SVC bit-stream is given $\Omega = \left(\pi_{m_0}, \ldots, \pi_{m_i}, \ldots, \pi_{m_{Q-1}}\right)$, which determines the allocation of the code rate across different Q quality layers and m takes all the values of available channel code rate. The expected distortion for the protection scheme Ω over the entire quality layers in the SVC bit-stream is defined as:

$$E[d](\Omega) = \sum_{i=0}^{Q-1} P_i(\Omega) d_i(\Omega), \qquad (1)$$

where d_i is the distortion of the bit-stream by using the data form the 0-th quality layer to the (i-1)-th quality layer only. $P_i(\Omega)$ is the probability that the (i-1)-th quality layer is received correctly and the first error is reported on the i-th quality layer. This can be defined as:



Adaptive Joint Source-Channel Coding for Scalable Video. Figure 3. Double binary turbo encoder.

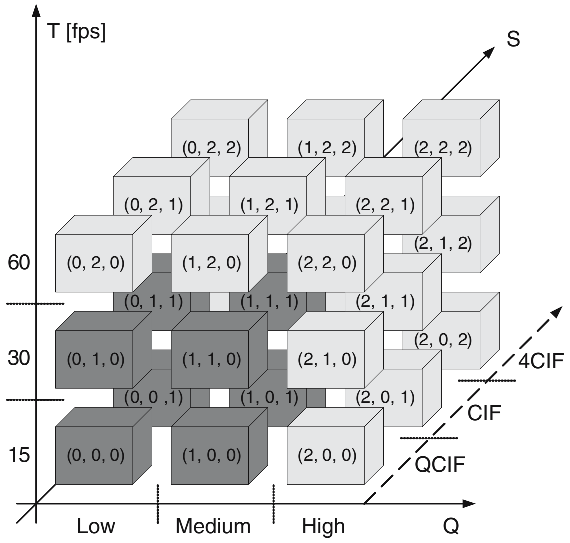
$$P_i(\Omega) = p(\pi_{m_i}) \prod_{j=1}^{i} \left(1 - p\left(\pi_{m_{j-1}}\right)\right)^j, \qquad (2)$$

where $p(\pi_{m_i})$ is the probability that the transmitted i-th quality layer is protected with code rate π_m and will be lost. Suppose an Optimal Protection Scheme (OPS) is $\Omega^* = \left(\pi^*_{m_1}, \ldots, \pi^*_{m_{Q-1}}\right)$ and is defined as:

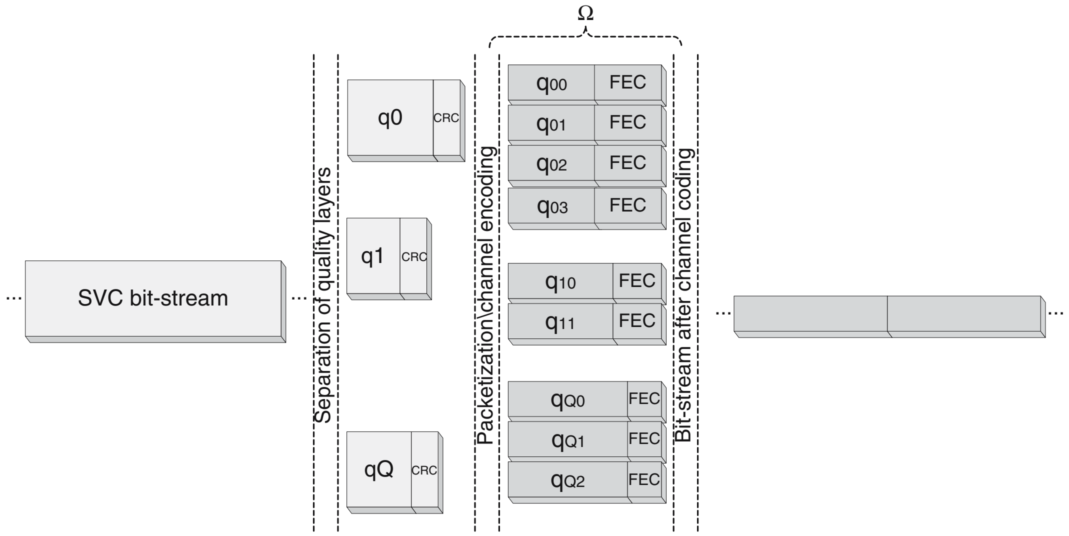
$$\Omega^* = \arg \min_{\Omega} E[d](\Omega). \qquad (3)$$

Now the problem converges to find the OPS. If we try to find this from (3), the algorithm complexity depends upon the number of quality layers Q and the available code rate π_m. In this case, as we have few quality layers and few channel code rates, it is practical to compute the OPS Ω^*. However if many quality layers are considered, then exhaustive computation may render the system impractical because of its huge complexity and dynamic programming is the only choice to solve the problem with limited complexity.

An algorithm [5] is proposed to find the OPS efficiently. Initially, the optimal Equal Error Protection (EEP) is found and then protection is iteratively increased from the lowest quality layers and decreased from the highest quality layer. Hence the protection scheme converges to an OPS in this iterative process. In short, more protection is applied to the important part of the bit-stream and a higher channel code rate is set for data with lower priority in the OPS. The lowest quality layer which contains the most important data (header, motion vectors and allocation tables) is protected with the lowest channel code rate and vice versa. The available code rates are: 1/3, 2/5, 1/2, 2/3, 3/4, 4/5 and 6/7.



Adaptive Joint Source-Channel Coding for Scalable Video. **Figure 4.** 3-D representation of a scalable video bit-stream in a GOP.



Adaptive Joint Source-Channel Coding for Scalable Video. **Figure 5.** AJSCC framework diagram with channel protection using CRC and FEC of different quality layers in SVC bit-stream.

At the decoder side, if a packet is error-corrupted, the CRC fails after channel decoding. We then point out the corresponding atom in the SVC bit-stream. If an atom (q_i, t_i, s_i) is corrupted after channel decoding or fails to qualify the CRC checks, then all the atoms which have higher index than i are removed by the error driven adaptation module outlined in Fig. 2. Finally, SVC decoding is performed to evaluate the overall performance of the system.

Conclusions

An efficient approach for adaptive joint source and channel coding for scalable video is presented. The proposed approach exploits the joint optimization of the SVC and a forward error correction method based on Turbo codes. Optimal protection scheme is used to minimize the end-to end distortion by considering the characteristics of scalable video bit-stream and channel coding at given channel conditions with limited complexity. The AJSCC provides a more graceful pattern of quality degradation as compared to conventional unequal error protection method in literature at different channel errors. The most advanced techniques for AJSCC for scalable bit-stream are explained in [6,7].

Cross-References

► Scalable Video Coding

References

1. M. Mrak, N. Sprljan, T. Zgaljic, N. Ramzan, S. Wan, and E. Izquierdo, "Performance evidence of software proposal for Wavelet Video Coding Exploration group, ISO/IEC JTC1/SC29/WG11/MPEG2006/M13146," 76th MPEG Meeting, Montreux, Switzerland, April 2006.
2. C. Berrou and A. Glavieux, "Near-Optimum Error-Correction Coding and Decoding: Turbo Codes," IEEE Transactions on Communications, Vol. 44, No. 10, October 1996, pp. 1261–1271.
3. C.E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27, 1948, pp. 379–423.
4. C. Doulliard and C. Berrou, "Turbo Codes with Rate-m/(m + 1) Constituent Convolutional Codes," IEEE Transactions on Communications, Vol. 53, No. 10, October 2005, pp. 1630–1638.
5. N. Ramzan and E. Izquierdo, "Scalable Video Transmission Using Double Binary Turbo Codes," Proceedings of 13th IEEE International Conference on Image Processing (ICIP), Atlanta, USA, October 2006.
6. N. Thomos, N.V. Boulgouris, and M.G Strintzis, "Wireless Image Transmission Using Turbo Codes and Optimal Unequal Error Protection," IEEE Transactions on Image Processing, Vol. 14. No. 11, November 2005, pp. 1890–1901.
7. N. Ramzan, S. Wan, and E. Izquierdo, "Joint Source-Channel Coding for Wavelet-Based Scalable Video Transmission Using an Adaptive Turbo Code," EURASIP Journal on Image and Video Processing, Vol. 2007, Article ID 47517, 12pp.

Advances in Image and Video Quality Assessment

KALPANA SESHADRINATHAN, ALAN C. BOVIK
University of Texas at Austin, Austin, TX, USA

Synonyms

► Image Fidelity Measurement; ► Video Fidelity Measurement

Definition

Objective measurement, by algorithm, of image and video quality in a way that agrees with human subjective assessment.

Introduction

In this article, we discuss methods to evaluate the quality of digital images and videos, where the final image is intended to be viewed by the human eye. The quality of an image that is meant for human consumption can be evaluated by showing it to a human observer and asking the subject to judge its quality on a pre-defined scale. This is known as *subjective assessment* and is currently the most common way to assess image and video quality. Clearly, this is also the most reliable method as we are interested in evaluating quality *as seen by the human eye*. However, to account for human variability in assessing quality and to have some statistical confidence in the score assigned by the subject, several subjects are required to view the same image. The final score for a particular image can then be computed as a statistical average of the sample scores. Also, in such an experiment, the assessment is dependent on several factors such as the display device, distance of viewing, content of the image, whether or not the subject is a trained observer who is familiar with processing of images etc. Thus, a change in viewing conditions would entail repeating the experiment! Imagine this process being repeated for every image that is encountered and it becomes clear why subjective studies are cumbersome and expensive. It would hence be extremely valuable to formulate some *objective measure* that can predict the quality of an image.

The problem of image and video quality assessment is to quantify the quality of an image or video signal *as seen by a human observer* using an objective measure. The quality assessment techniques that we present in this chapter are known as *full-reference* techniques, i.e. it is assumed that in addition to the test image whose quality we wish to evaluate, a "perfect" reference image is also available. We are, thus, actually evaluating the fidelity of the image, rather than the quality. Evaluating the quality of an image without a reference image is a much harder problem and is known as *blind* or *no-reference* quality assessment. Blind techniques generally reduce the storage requirements of the algorithm, which could lead to considerable savings, especially in the case of video signals. Also, in certain applications, the original uncorrupted image may not be available. However, blind algorithms are also difficult to develop as the interpretation of the content and quality of an image by the HVS depends on high-level features such as attentive vision, cognitive understanding, and prior experiences of viewing similar patterns, which are not very well understood. *Reduced reference* quality assessment techniques form the middle ground and use some information from the reference signal, without requiring that the entire reference image be available.

Why Do We Need Quality Assessment?

Image and video quality assessment plays a fundamental role in the design and evaluation of imaging and image processing systems. For example, the goal of image and video compression algorithms is to reduce the amount of data required to store an image and at the same time, ensure that the resulting image is of sufficiently high quality. Image enhancement and restoration algorithms attempt to generate an image that is of better visual quality from a degraded image. Quality assessment algorithms are also useful in the design of image acquisition systems and to evaluate display devices etc. Communication networks have developed tremendously over the past decade and images and video are frequently transported over optic fiber, packet switched networks like the Internet, wireless systems etc. Bandwidth efficiency of applications such as video conferencing and Video on Demand (VoD) can be improved using quality assessment systems to evaluate the effects of channel errors on the transported images and video. Finally, quality assessment and the psychophysics of human vision are closely related disciplines. Evaluation of quality requires clear understanding of the sensitivities of the HVS to several features such as luminance, contrast, texture, and masking that are discussed in detail in Section 4.1. Research on image and video quality assessment may lend deep insights into the functioning of the HVS, which would be of great scientific value.

Why is Quality Assessment So Hard?

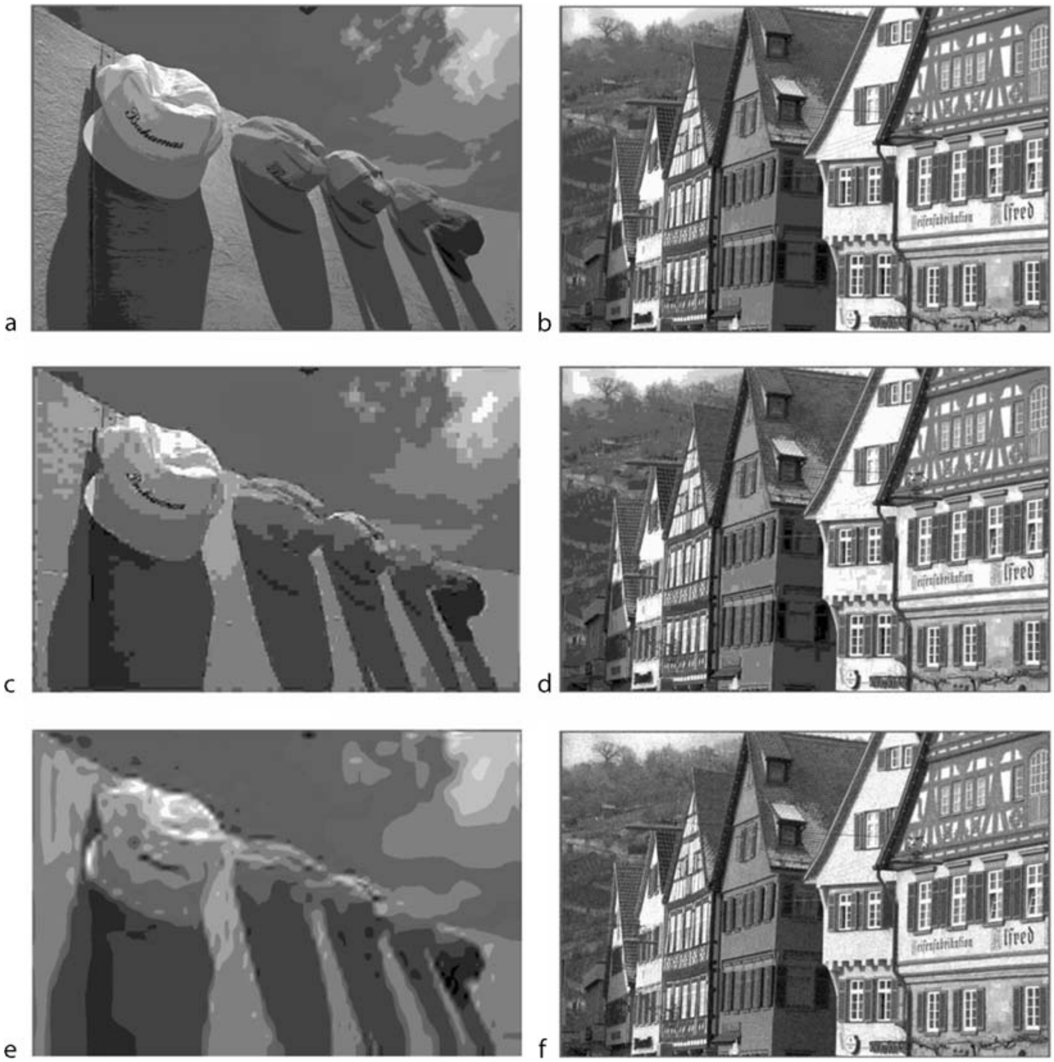
At first glance, a reasonable candidate for an image quality metric might be the Mean-Squared Error (MSE) between the reference and distorted images.

Let $\vec{R} = \{R_i, 1 \leq i \leq N\}$ and $\vec{T} = \{T_i, 1 \leq i \leq N\}$ represent N-dimensional vectors containing pixels from the reference and test image or video respectively. Then, the MSE between $\vec{R}$ and $\vec{T}$ is defined by

$$MSE(\vec{R}, \vec{T}) = \frac{1}{N} \sum_{i=1}^{N} (R_i - T_i)^2$$

MSE is the square of the Euclidean distance between the two vectors $\vec{R}$ and $\vec{T}$ in an N-dimensional space. Since the MSE is a monotonic function of the error between corresponding pixels in the reference and distorted images, it is a reasonable metric and is often used as a quality measure. Some of the reasons for the popularity of this metric are its simplicity, ease of computation and analytic tractability. However, it has long been known to correlate very poorly with visual quality [1]. A few simple examples are sufficient to demonstrate that MSE is *completely unacceptable* as a visual quality predictor. This is illustrated in Fig. 1 which shows several images, where the MSE of the distorted image with respect to the reference are identical, that have very different visual quality. The main reason for the failure of MSE as a quality metric is the absence of any kind of modeling of the sensitivities of the HVS.

The difficulties in developing objective measures of image quality are best illustrated by example. Figure 1 (a) and (b) show the original "Caps" and "Buildings" images respectively. Figure 1(c) and (d) show JPEG compressed versions of these images of approximately the same MSE. While the distortion in the "Buildings" image is hardly visible, it is visibly annoying in the "Caps" image. The perception of distortion varies with the actual image at hand and this effect is part of what makes quality assessment difficult. There is enormous diversity in the content of images used in different applications and even within images of a specific category, for example, the class of images



Advances in Image and Video Quality Assessment. Figure 1. (a) Original "Caps" image (b) Original "Buildings" image (c) JPEG compressed image, MSE = 160 (d) JPEG compressed image, MSE = 165 (e) JPEG 2000 compressed image, MSE = 155 (f) AWGN corrupted image, MSE = 160.

obtained from the real world. Consistent performance of a quality assessment algorithm irrespective of the specific image at hand is no easy task. Additionally, different kinds of distortion produce different characteristic artifacts and it is very difficult for a quality assessment algorithm to predict degradation in visual quality *across distortion types*. For example, JPEG produces characteristic blocking artifacts and blurring of fine details (Fig. 1(c) and (d)). This is due to the fact that it is a block-based algorithm that achieves compression by removing the highest frequency components that the HVS is least sensitive to. JPEG 2000 compression eliminates blocking artifacts, but produces ringing distortions that are visible in areas surrounding edges, such as around the edges of the caps in Fig. 1(e). Sub-band decompositions, such as those used in JPEG 2000, attempt to approximate the image using finite-duration basis functions and this causes ringing around discontinuities like edges due to Gibb's phenomenon. Figure 1(f) shows an image that is corrupted by Additive White Gaussian Noise (AWGN) which looks grainy, seen clearly in the smooth background regions of the image. This kind of noise is typically observed in a lot of imaging devices and images transmitted through certain communication channels. A generic image quality measure should predict visual quality in a robust manner across these and several other types of distortions.

Thus, it is not an easy task for a machine to automatically predict quality by computation, although the human eye is very good at evaluating the quality of an image almost instantly. We explore some state-of-the-art techniques for objective quality assessment in the following sections.

Approaches to Quality Assessment

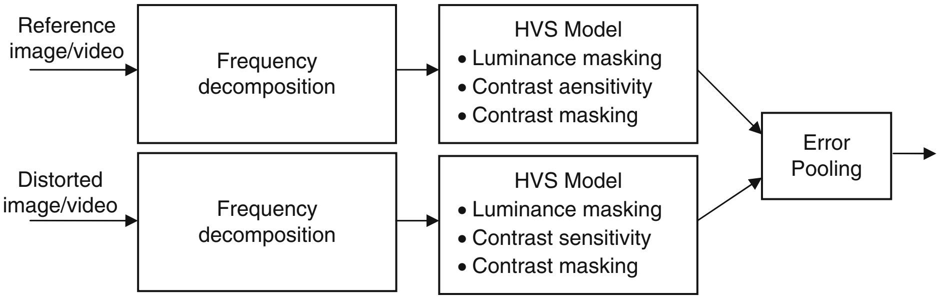
Techniques for image and video quality assessment can broadly be classified as models based on human vision and models based on signal statistics. Human vision based systems attempt to model the functioning of the HVS and characterize the sensitivities and limitations of the human eye to predict the quality of a given image. Approaches based on signal statistics, on the other hand, usually make some high-level assumption on the technique adopted by the human eye in evaluating quality and use this to develop a quality metric. In this chapter, we categorize several state-of-the-art quality assessment techniques into three main categories, namely HVS modeling based approaches, structural approaches and information theoretic approaches. Each of these paradigms in perceptual quality assessment is explained in detail in the following sections.

HVS-based Approaches

Most HVS-based approaches can be summarized by the diagram shown in Fig. 2. The initial step in the process usually involves the decomposition of the image into different spatial-frequency channels in the case of still images. In the case of video sequences, the "Frequency Decomposition" block decomposes the video into different spatial *and temporal* frequency channels. It is well known that cells in the visual cortex are specialized and tuned to different spatial frequencies and orientations. Experimental studies indicate that the radial frequency selective mechanisms have constant octave bandwidths and the orientation selectivity is a function of the radial frequencies. Several transforms have been proposed to model the spatial frequency selectivity of the HVS which are used to decompose the image into different spatial frequency channels. In the case of video sequences, it is believed that two kinds of temporal mechanisms exist in the early stages of processing in the visual cortex, one low-pass and one band-pass. HVS-based video quality assessment algorithms decompose the video sequence into different temporal frequency channels using low-pass and band-pass temporal filters, in addition to the spatial frequency decomposition.

The perception of brightness is not a linear function of the luminance and this effect is known as luminance masking. In fact, the threshold of visibility of a brightness pattern is a linear function of the background luminance. In other words, brighter regions in an image can tolerate more noise due to distortions before it becomes visually annoying. The Contrast Sensitivity Function (CSF) provides a description of the frequency response of the HVS, which can be thought of as a band-pass filter. For example, the HVS is less sensitive to higher spatial frequencies and this fact is exploited by most compression algorithms to encode images at low bit rates, with minimal degradation in visual quality. Most HVS-based approaches use some kind of modeling of the luminance masking and contrast sensitivity properties of the HVS as shown in Fig. 2.

In Fig. 1, the distortions are clearly visible in the "Caps" image, but they are hardly noticeable in the "Buildings" image, despite the MSE being the same. This is a consequence of the contrast masking property of the HVS, wherein the visibility of certain



Advances in Image and Video Quality Assessment. Figure 2. Block diagram of HVS-based quality metrics.

image components is reduced due to the presence of other strong image components with similar spatial frequencies and orientations at neighboring spatial locations. Thus, the strong edges and structure in the "Buildings" image effectively mask the distortion, while it is clearly visible in the smooth "Caps" image. Usually, a HVS-based metric incorporates modeling of the contrast masking property, as shown in Fig. 2.

In developing a quality metric, a signal is first decomposed into several frequency bands and the HVS model specifies the maximum possible distortion that can be introduced in each frequency component before the distortion becomes visible. This is known as the Just Noticeable Difference (JND). The final stage in the quality evaluation involves combining the errors in the different frequency components, after normalizing them with the corresponding sensitivity thresholds, using some metric such as the Minkowski error. The final output of the algorithm is either a spatial map showing the image quality at different spatial locations or a single number describing the overall quality of the image.

Different proposed quality metrics differ in the models used for the blocks shown in Fig. 2. Notable amongst the HVS-based quality measures for still images are the Visible Difference Predictor [2], the Teo and Heeger model [3], Lubin's model [4], Watson's DCTune (for DCT-based compression systems) [5] and the Visual Signal to Noise Ratio [6]. Popular HVS-based video quality assessment systems include Sarnoff's JNDMetrix technology [7], Watson's Digital Video Quality (DVQ) metric [8], Moving Pictures Quality Metric (MPQM) [9], Perceptual Distortion Metric (PDM) [10] and Continuous Video Quality Evaluation (CVQE) [11].

Structural Approaches

In structural approaches to image quality assessment, it is hypothesized that loss of visual quality can be equated to the loss of *structural information* from a scene. The loss in structural information is quantified using statistics calculated from the reference and test signals and is used to predict the quality of an image or video sequence [12, 13]. In Fig. 1, the distorted versions of the "Buildings" image and the "Caps" image have the same MSE with respect to the reference image. The bad visual quality of the "Caps" image can be attributed to the structural distortions in both the background and the objects in the image. The structural philosophy can also accurately predict the good visual quality of the "Buildings" image, since the structure of the image remains almost intact in both distorted versions.

Structural information is defined as those aspects of the image or video that are independent of the luminance and contrast, since the structure of various objects in the scene is independent of the brightness and contrast of the image. The Structural SIMilarity (SSIM) algorithm, also known as the Wang-Bovik Index partitions the quality assessment problem into three components, namely luminance, contrast and structure comparisons.

Let $\vec{R} = \{R_i, 1 \leq i \leq N\}$ and $\vec{T} = \{T_i, 1 \leq i \leq N\}$ represent N-dimensional vectors containing pixels from the reference and test image or video respectively. Then, the Wang-Bovik Index between $\vec{R}$ and $\vec{T}$ is defined by

$$SSIM(\vec{R}, \vec{T}) = \left(\frac{2\mu_R\mu_T + C_1}{\mu_R^2 + \mu_T^2 + C_1}\right)^{\alpha} \left(\frac{2\sigma_R\sigma_T + C_2}{\sigma_R^2 + \sigma_T^2 + C_2}\right)^{\beta} \left(\frac{\sigma_{RT} + C_3}{\sigma_R\sigma_T + C_3}\right)^{\gamma} \quad (1)$$

where

$$\mu_R = \frac{1}{N}\sum_{i=1}^{N} R_i, \sigma_R = \left(\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \mu_R)^2\right)^{1/2},$$

$$\sigma_{RT} = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - \mu_R)(T_i - \mu_T)$$

C_1, C_2 and C_3 are small constants added to avoid numerical instability when the denominators of the fractions are small. α, β and γ are non-negative constants that control the relative contributions of the three different measurements to the Wang-Bovik Index.

The three terms in the right hand side of Eqn. (1) are the luminance, contrast and structure comparison measurements respectively. μ_R and μ_T are estimates of the mean luminance of the two patches and hence, the first term in Eqn. (1) defines the luminance comparison function. It is easily seen that the luminance comparison function satisfies the desirable properties of being bounded by 1 and attaining the maximum possible value if and only if the means of the two images are equal. Similarly, σ_R and σ_T are estimates of the contrast of the two patches and the second term in Eqn. (1) defines the contrast comparison function. Finally, the structural comparison is performed between the luminance and contrast normalized signals, given by $(\vec{R} - \mu_R)/\sigma_R$ and $(\vec{T} - \mu_T)/\sigma_T$. The correlation or inner product between these signals is an effective measure of the structural similarity. The correlation between the normalized vectors is equal to the correlation coefficient between the original signals $\vec{R}$ and $\vec{T}$, which is defined by the third term in Eqn. (1). Note that the Wang-Bovik Index is also bounded by 1 and attains unity if and only if the two images are equal.

In quality assessment of still images, the patches $\vec{R}$ and $\vec{T}$ are extracted locally from each image using a window. The final SSIM index is computed as a mean of the local quality indices. An extension of the SSIM index, known as the Multi-Scale SSIM (MS-SSIM) index, that decomposes the image into multiple scales before quality computation was also proposed [14]. Experimental studies on a large database of images show that both SSIM and MS-SSIM are competitive with several other state-of-the-art quality metrics [12,15].

For video sequences, the patches $\vec{R}$ and $\vec{T}$ are extracted locally from each frame of the video sequence using a window. The local quality indices are combined using a heuristic weighting function that accounts for the reduced visibility of spatial detail when the speed of motion is large to obtain a final quality index for the entire video in [13]. An improvement of the weighting function used in [13] was suggested in [16], where a more scientific approach is taken to designing these weights. In [16], motion information is computed from the video sequence and psychophysical models are used to estimate the information content and perceptual uncertainty of the motion information. These are then combined and used as weights in pooling the local quality indices and experimental studies show that the performance of this new algorithm is significantly better than [13].

Video quality assessment algorithms based on the SSIM philosophy in [13, 16] operate on the video frame by frame to compute local quality indices. Although motion information in the video is used to weight these local quality indices to generate a final quality score for the video, motion information is not used directly in the local quality computation. A new paradigm for video quality assessment based on the SSIM philosophy that incorporates motion modeling, known as the V-SSIM index, was proposed in [17]. Some spatial artifacts occur within video frames that do not arise from temporal processes, such as blocking from DCT coefficient quantization in MPEG; ringing from quantization in block-free codecs such as Motion JPEG-2000; mosaic patterns; and false contouring. Spatio-temporal artifacts arise from spatio-temporal processes including trailing ghosts behind moving objects; artifacts from motion compensation mismatches in MPEG; mosquito effect and stationary area fluctuations in MPEG; jitter from transmission delays or losses; and motion blur from acquisition or display devices (e.g., LCD). Operating on the video frame by frame captures spatial distortions in the video, but fails to adequately capture the temporal distortions. Additionally, the HVS is extremely sensitive to the speed and direction of movement of objects in a scene, a skill that is crucial in our day-to-day lives. Since the goal of a quality assessment algorithm is to match human perception, it is important to model motion in developing an accurate video quality assessment system. The V-SSIM index attempts to achieve these objectives by suggesting a novel framework for video quality assessment that incorporates motion information in computing the local quality indices.

In V-SSIM, short video segments are modeled as translating image patches. The video is filtered using a bank of *spatio-temporal* Gabor filters and the outputs of these Gabor filters are first used to estimate the optical flow or motion vector at each pixel of the reference video sequence. In the next step, these motion vectors are used to construct a set of motion-compensated responses that filter both the reference and test videos along the motion trajectory of the reference. Such an approach can capture both spatial and *temporal* distortions that alter the motion information in the test video with respect to the reference. The performance of the V-SSIM index has also been experimentally validated and is shown to be competitive with other video quality assessment algorithms [17].

Information Theoretic Approaches

Information theoretic approaches attempt to quantify the *loss in the information* that the HVS can extract from a given test image, as compared to the original reference image [18]. *Mutual information* between two random sources is a statistical measure that quantifies the amount of information one source contains about the other. In other words, assuming the distorted and reference images to be samples obtained from two random sources, mutual information measures the distance between the distributions of these sources. Information theoretic approaches use this measure to quantify the amount of information that the human eye can obtain from a given image, to develop a metric that correlates well with visual quality. The Visual Information Fidelity (VIF) criterion, also known as the Sheikh-Bovik Index, assumes that the distorted image is the output of a communication channel that introduces errors in the reference image that passes through it. The HVS is also assumed to be a communication channel that limits the amount of information that can pass through it.

Photographic images of natural scenes exhibit striking structures and dependencies and are far from random. A random image generated assuming an independent and identically distributed Gaussian source, for example, will look nothing like a natural image. Characterizing the distributions and statistical dependencies of natural images provides a description of the subspace spanned by natural images, in the space of all possible images. Such probabilistic models have been studied by numerous researchers and one model that has achieved considerable success is known as the Gaussian Scale Mixture (GSM) model [19]. This is the source model used to describe the statistics of the wavelet coefficients of reference images in the Sheikh-Bovik Index. Let $\vec{R}$ represent a collection of wavelet coefficients from neighboring spatial locations of the original image. Then, $\vec{R}{\sim}z\vec{U}$, where z represents a scalar random variable known as the mixing density and $\vec{U}$ represents a zero-mean, white Gaussian random vector. Instead of explicitly characterizing the mixing density, the maximum likelihood estimate of the scalar z is derived from the given image in the development of the Sheikh-Bovik Index.

Let $\vec{T}$ denote the corresponding coefficients from the distorted image. The distortion channel that the reference image passes through to produce the distorted image is modeled using

$$\vec{T} = g\vec{R} + \vec{v}$$

This is a simple signal attenuation plus additive noise model, where g represents a scalar attenuation and $\vec{v}$ is additive Gaussian noise. Most commonly occurring distortions such as compression and blurring can be approximated by this model reasonably well. This model has some nice properties such as analytic tractability, ability to characterize a wide variety of distortions and computational simplicity. Additionally, both reference and distorted images pass through a communication channel that models neural noise in the HVS. The HVS neural noise model is given by

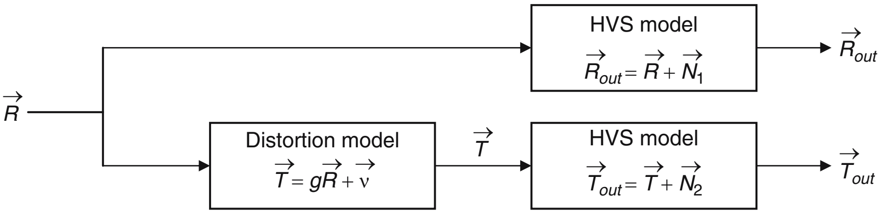
$$\vec{R}_{out} = \vec{R} + \vec{N}_1, \vec{T}_{out} = \vec{T} + \vec{N}_2$$

where $\vec{N}_1$ and $\vec{N}_2$ represent additive Gaussian noise, that is independent of the input image. The entire system is illustrated in Fig. 3.

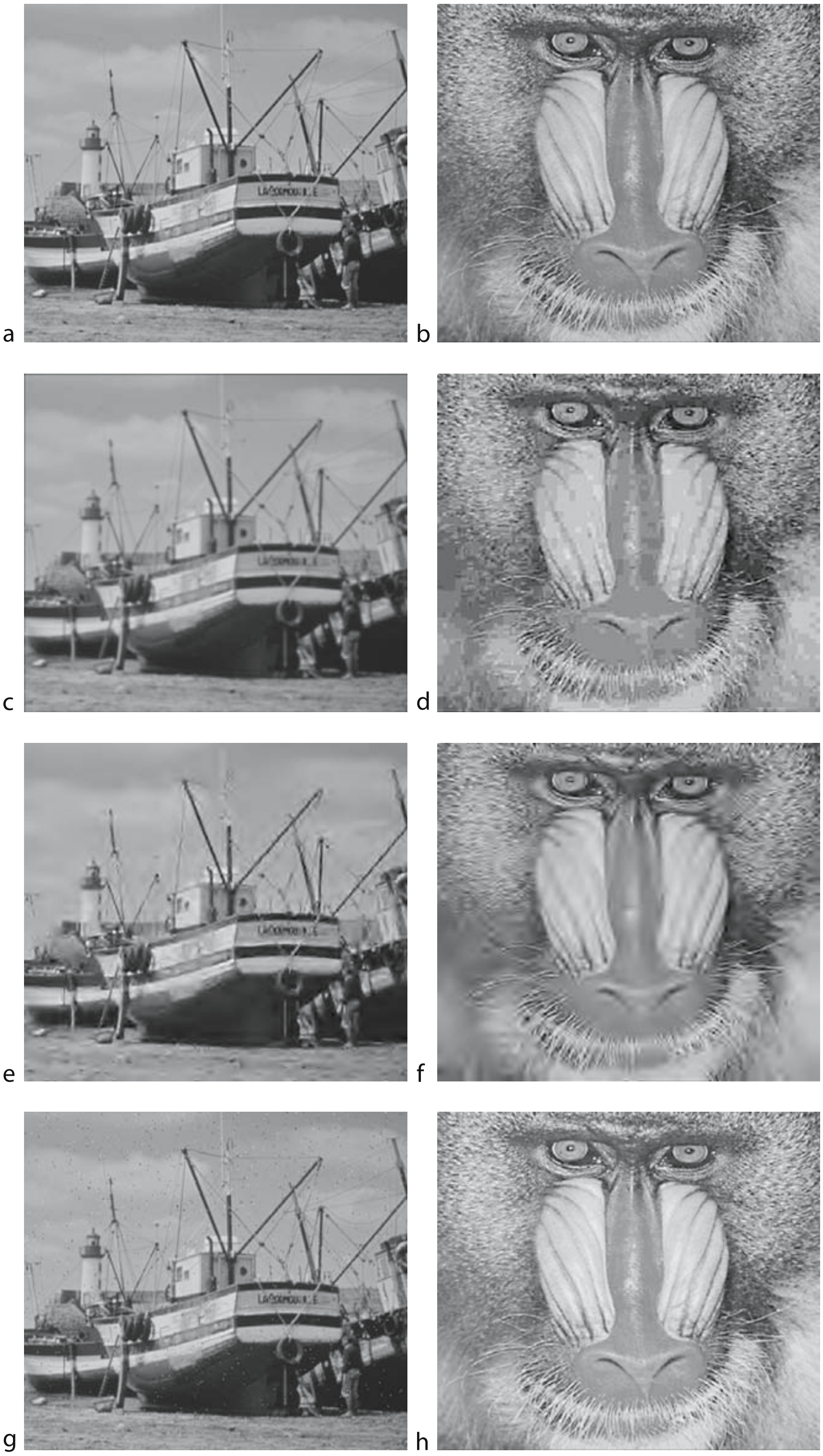
The VIF criterion is then defined for these coefficients using

$$VIF = \frac{I(\vec{T}; \vec{T}_{out}/z)}{I(\vec{R}; \vec{R}_{out}/z)} \tag{2}$$

$I(\vec{T}; \vec{T}_{out}/z)$ represents the mutual information between $\vec{T}$ and $\vec{T}_{out}$, conditioned on the estimated value of z. The denominator of Eqn. (2) represents the amount of information that the HVS can extract from the original image. The numerator represents the amount of information that the HVS can extract from the distorted image. The ratio of these two quantities hence is a measure of the amount of information in the distorted image relative to the reference image and has


$\vec{R}$
HVS model
$\vec{R}_{out} = \vec{R} + \vec{N}_1$
$\vec{R}_{out}$
Distortion model
$\vec{T} = g\vec{R} + \vec{v}$
$\vec{T}$
HVS model
$\vec{T}_{out} = \vec{T} + \vec{N}_2$
$\vec{T}_{out}$

Advances in Image and Video Quality Assessment. Figure 3. Block diagram of the Sheikh-Bovik quality assessment system.



Advances in Image and Video Quality Assessment. Figure 4. Illustration of the Wang-Bovik and Sheikh-Bovik indices. (a) Original ‘Boats’ image (b) Original ‘Mandrill’ image (c) Gaussian Blurring, SSIM = 0.85, VIF = 0.25 (d) JPEG compression, SSIM = 0.54, VIF = 0.07 (e) JPEG2000 compression, SSIM = 0.78, VIF = 0.11 (f) JPEG2000 compression, SSIM = 0.48, VIF = 0.05 (g) Salt and Pepper noise, SSIM = 0.87, VIF = 0.38 (h) Mean shifted, SSIM = 0.99, VIF = 1.

been shown to correlate very well with visual quality. Closed form expressions to compute this quantity have been derived and further details can be found in [18].

Figure 4 illustrates the power of the Wang-Bovik and Sheikh-Bovik indices in predicting image quality. Notice that the relative quality of the images, as predicted by both indices, is the same and agrees reasonably well with human perception of quality. In the Video Quality Experts Group (VQEG) Phase I FR-TV tests [20], which provides performance evaluation procedures for quality metrics, logistic functions are used in a fitting procedure to obtain a non-linear mapping between objective/subjective scores first. Hence, the differences in the absolute values of quality predicted by the two algorithms are not important.

The VIF index was extended to video by filtering the video using simple spatial and temporal derivative kernels and applying the VIF model for still images on the resulting representation [21]. A precursor to the VIF index for still images was known as the Information Fidelity Criterion (IFC) [22]. Motion models and spatio-temporal distortion measurement were used in a framework similar to the V-SSIM index to construct an IFC index for video sequences [23].

Natural scene modeling is in some sense a dual of HVS modeling, as the HVS has evolved in response to the natural images it perceives. The relation between this approach and certain HVS-based approaches has been studied [22]. The use of the GSM model results in the source coefficients being Gaussian distributed when conditioned on the mixing density. Use of the linear channel model results in $(\vec{T}, \vec{R})$ being jointly Gaussian distributed, when conditioned on the mixing density. In the case of jointly Gaussian random variables, the mutual information and the correlation coefficient between them satisfy a simple monotonic relationship. Recent work has established the equivalence of the Information Fidelity Criterion and the structure term of the MS-SSIM index, which is the correlation coefficient between reference and test image patches after a multi-scale decomposition [24]. This indicates a conceptual unification of different approaches to the quality assessment problem discussed in this chapter, which is useful in understanding the science of quality assessment.

Conclusions

In this chapter, we have attempted to present a short survey of full reference image and video quality assessment techniques. Considerable progress has been made in the field of quality assessment over the years, especially in the context of specific applications like compression and half-toning. No-reference quality assessment is still in its infancy and is likely to be the thrust of future research in this area.

Cross-References

▶ Frequency Domain Representations for 3-D Face Recognition
▶ Range Image Quality Assessment by Structural Similarity
▶ Structural Similarity Index Based Optimization
▶ Video Quality Assessment Over Wireless Channels

References

1. B. Girod, "What's Wrong with Mean-Squared Error," in A.B. Watson (ed.), Digital Images and Human Vision, MIT Press, 1993, pp. 207–220.
2. S. Daly, "The Visible Difference Predictor: An Algorithm for the Assessment of Image Fidelity," Proceedings of the SPIE, Vol. 1616, 1992, pp. 2–15.
3. P.C. Teo and D.J. Heeger, "Perceptual Image Distortion," Proceedings of the SPIE, Vol. 2179, 1994, pp. 127–141.
4. J. Lubin, "The Use Of Psychophysical Data And Models In The Analysis Of Display System Performance," in A.B. Watson (ed.), Digital Images and Human Vision, MIT Press, 1993, pp. 163–178.
5. A.B. Watson, "DCTune: A Technique for Visual Optimization of DCT Quantization Matrices for Individual Images," Society for Information Display Digest of Technical Papers, vol. 24, 1993, pp. 946–949.
6. D.M. Chandler and S.S. Hemami, "VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images," IEEE Transactions on Image Processing, Vol. 16, No. 9, September 2007, pp. 2284–2298.
7. Sarnoff Corporation, "JNDMetrix Technology," 2003. http://www.sarnoff.com/products_services/video_vision/jndmetrix/downloads.asp.
8. A.B. Watson, J. Hu, and J.F. McGowan III, "Digital Video Quality Metric Based on Human Vision," Journal of Electronic Imaging, Vol. 10, No. 1, January 2001, pp. 20–29.
9. C.J. van den Branden Lambrecht and O. Verscheure, "Perceptual Quality Measure Using a Spatiotemporal Model of the Human Visual System," Proceedings of SPIE International Society of Optical Engineers, Vol. 2668, No. 1, March 1996, pp. 450–461.
10. S. Winkler, "Perceptual Distortion Metric for Digital Color Video," Proceedings of SPIE International Society of Optical Engineers, Vol. 3644, No. 1, May 1999, pp. 175–184.
11. M. Masry, S.S. Hemami, and Y. Sermadevi, "A Scalable Wavelet-Based Video Distortion Metric and Applications," IEEE Transaction on Circuits and Systems for Video Technology, vol. 16, 2006, pp. 260–273.

12. Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image Quality Assessment: From Error Visibility To Structural Similarity," IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004, pp. 1–14.
13. Z. Wang, L. Lu, and A.C. Bovik, "Video Quality Assessment Based On Structural Distortion Measurement," Signal Processing: Image Communication, vol. 19, February 2004, pp. 121–132.
14. Z. Wang, E.P. Simoncelli, and A.C. Bovik, "Multiscale Structural Similarity For Image Quality Assessment," Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, Vol. 2, November 2003, pp. 1398–1402.
15. H.R. Sheikh, M.F. Sabir, and A.C. Bovik, "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," IEEE Transactions on Image Processing, Vol. 15, No. 11, November 2006, pp. 3440–3451.
16. Z. Wang and Q. Li, "Video Quality Assessment Using A Statistical Model Of Human Visual Speed Perception," Journal of the Optical Society of America A, Vol. 24, No. 12, December 2007, pp. B61–B69.
17. K. Seshadrinathan and A.C. Bovik, "A Structural Similarity Metric for Video Based on Motion Models," IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 1, April 2007, pp. I-869–I-872.
18. H.R. Sheikh and A.C. Bovik, "Image Information And Visual Quality," IEEE Transactions on Image Processing, Vol. 15. No. 2, February 2006, pp. 430–444.
19. M.J. Wainwright and E.P. Simoncelli, "Scale Mixtures Of Gaussians And The Statistics Of Natural Images," in S.A. Solla, T.K. Leen, and K.R. Mueller (eds.), Advances in Neural Information Processing Systems, Cambridge, Vol. 12, MIT Press, MA, May 2000, pp. 855–861.
20. VQEG, "Final Report From The Video Quality Experts Group On The Validation Of Objective Models Of Video Quality Assessment," March 2000. http://www.vqeg.org
21. H.R. Sheikh and A.C. Bovik, "A Visual Information Fidelity Approach to Video Quality Assessment," The First International Workshop on Video Processing and Quality Metrics for Consumer Electronics, Scottsdale, AZ, January 2005.
22. H.R. Sheikh, A.C. Bovik, and G. de Veciana, "An Information Fidelity Criterion For Image Quality Assessment Using Natural Scene Statistics," IEEE Transactions on Image Processing, Vol. 14, No. 12, December 2005, pp. 2117–2128.
23. K. Seshadrinathan and A.C. Bovik, "An Information Theoretic Video Quality Metric Based On Motion Models," Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics, Scottsdale, Arizona, January 2007.
24. K. Seshadrinathan and A.C. Bovik, "Unified Treatment Of Full Reference Image Quality Assessment Algorithms," submitted to the IEEE Transactions on Image Processing, February 2008.

Ambient Media Intelligence

► Streaming Multimedia Information

Analyzing Person Information in News Video

Shin'ichi Satoh
National Institute of Informatics, Tokyo, Japan

Synonyms

► Face-name association in news video

Definition

Analyzing person information in news video includes the identification of various attributes of a person, such as face detection and recognition, face-name association, and others.

Introduction

Person information analysis for news videos, including face detection and recognition, face-name association, etc., has attracted many researchers in the video indexing field. One reason for this is the importance of person information. In our social interactions, we use face as symbolic information to identify each other. This strengthens the importance of face among many types of visual information, and thus face image processing has been intensively studied for decades by image processing and computer vision researchers. As an outcome, robust face detection and recognition techniques have been proposed. Therefore, face information in news videos is rather more easily accessible compared to the other types of visual information.

In addition, especially in news, person information is the most important; for instance, "*who* said this?," "*who* went there?," "*who* did this?," etc., could be the major information which news provides. Among all such types of person information, "*who* is this?" information, i.e., face-name association, is the most basic as well as the most important information. Despite its basic nature, face-name association is not an easy task for computers; in some cases, it requires in-depth semantic analysis of videos, which is never achieved yet even by the most advanced technologies. This is another reason why face-name association still attracts many researchers: face-name association is a good touchstone of video analysis technologies.

This article describes about face-name association in news videos. In doing this, we take one of the earliest attempts as an example: Name-It. We briefly describe its mechanism. Then we compare it with corpus-based natural language processing and information retrieval techniques, and show the effectiveness of corpus-based video analysis.

Face-Name Association: Name-It Approach

Typical processing of face-name association is as follows:

- Extracts faces from images (videos)
- Extracts names from speech (closed-caption (CC) text)
- Associates faces and names

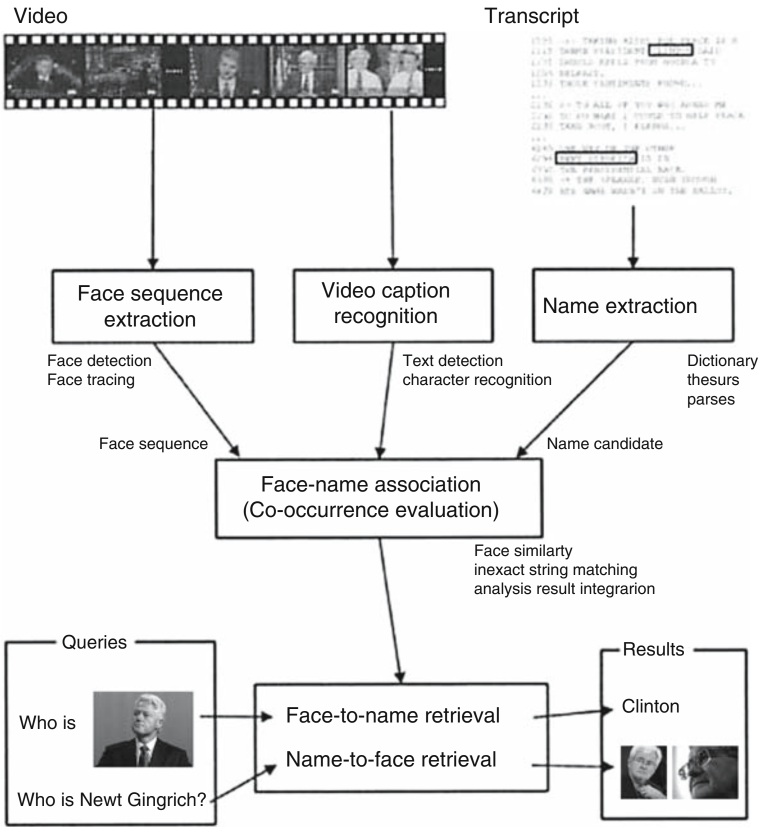
This looks very simple. Let's assume that we have a segment of news video as shown in Fig. 1. We don't feel any difficulty in associating the face and name when we watch this news video segment, i.e., the face corresponds to "Bill Clinton" even though we don't know the person beforehand. Video information is composed mainly of two streams: visual stream and speech (or CC) stream. Usually each one of these is not direct explanation of another. For instance, if visual information is shown as Fig. 1, the corresponding speech will not be: "The person shown here is Mr. Clinton. He is making speech on...," which is the direct explanation of the visual information. If so the news video could be too redundant and tedious to viewers. Instead they are complementary each other, and thus concise and easy to understand for people. However, it is very hard for computers to analyze news video segments. In order to associate the face and name shown in Fig. 1, computers need to understand visual stream so that a person shown is making speech, and to understand text stream that the news is about a speech by Mr. Clinton, and thus to realize the person corresponds to Mr. Clinton. This correspondence is shown only implicitly, which makes the analysis difficult for computers. This requires image/video understanding as well as speech/text understanding, which themselves are still very difficult tasks.

6902 >>> PRESIDENT CLINTON MET
6963 WITH FRENCH PRESEIDENT
6993 JACQIES CHIRAC TODAY
7023 AT THE WHITE HOUSE.
7083 MT. CLINTON SAID HE WELCOMED
7113 FRANCE'S DECISION TO END
7143 ITS NUCLEAR TEST PROGRAM
7204 IN THE PACIFIC AND PLEDGED
7234 TO WORK WITH FRANCE TO BAN
7264 FUTURE TESTS.

Analyzing Person Information in News Video. Figure 1. Example of news video segment.

Name-It [1] is one of the earliest systems tackling the problem of face-name association in news videos. Name-It assumes that image stream processing, i.e., face extraction, as well as text stream processing, i.e., name extraction, are not necessarily perfect. Thus the proper face-name association cannot be realized only from each segment. For example, from the segment shown in Fig. 1, it is possible for computers that the face shown here can be associated with "Clinton" or "Chirac," but the ambiguity between these cannot be resolved. To handle this situation, Name-It takes a corpus-based video analysis approach to obtain sufficiently reliable face-name association from imperfect image/text stream understanding results.

The architecture of Name-It is shown in Fig. 2. Since closed-captioned CNN Headline News is used as news video corpus, given news videos are composed of a video portion along with a transcript (closed-caption text) portion. From video images, the system extracts faces of persons who might be mentioned in transcripts. Meanwhile, from transcripts, the system extracts words corresponding to persons who might appear in videos. Since names and faces are both extracted from videos, they furnish additional timing information, i.e., at what time in videos they appear. The association of names and faces is evaluated with a "co-occurrence" factor using their timing information. Co-occurrence of a name and a face expresses how often and how well the name coincides with the face in given news video archives. In addition, the system also extracts video captions from video images. Extracted video captions are recognized to obtain text information, and then used to enhance the quality of face-name association. By the co-occurrence, the system collects ambiguous face-name association cues, each of which is obtained from each news video segment, over the entire news video corpus, to obtain sufficiently reliable face-name association results.



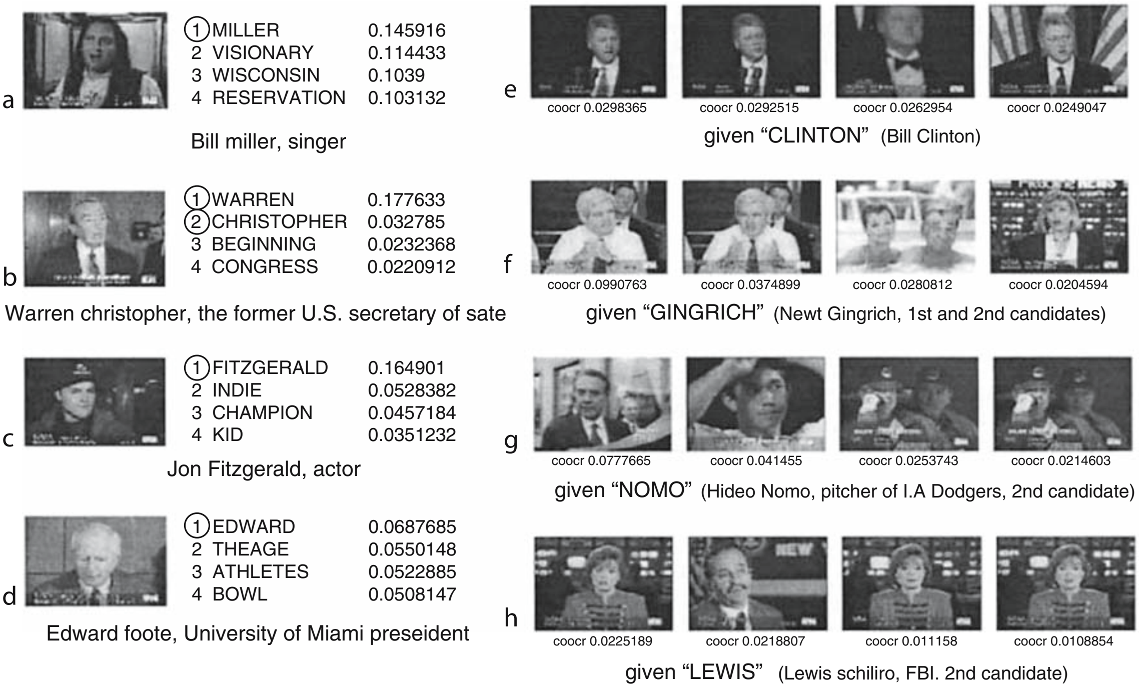
Analyzing Person Information in News Video. Figure 2. The architecture of Name-It.

Figure 3 shows the results of face-name association by using five hours of CNN Headline News videos as corpus.

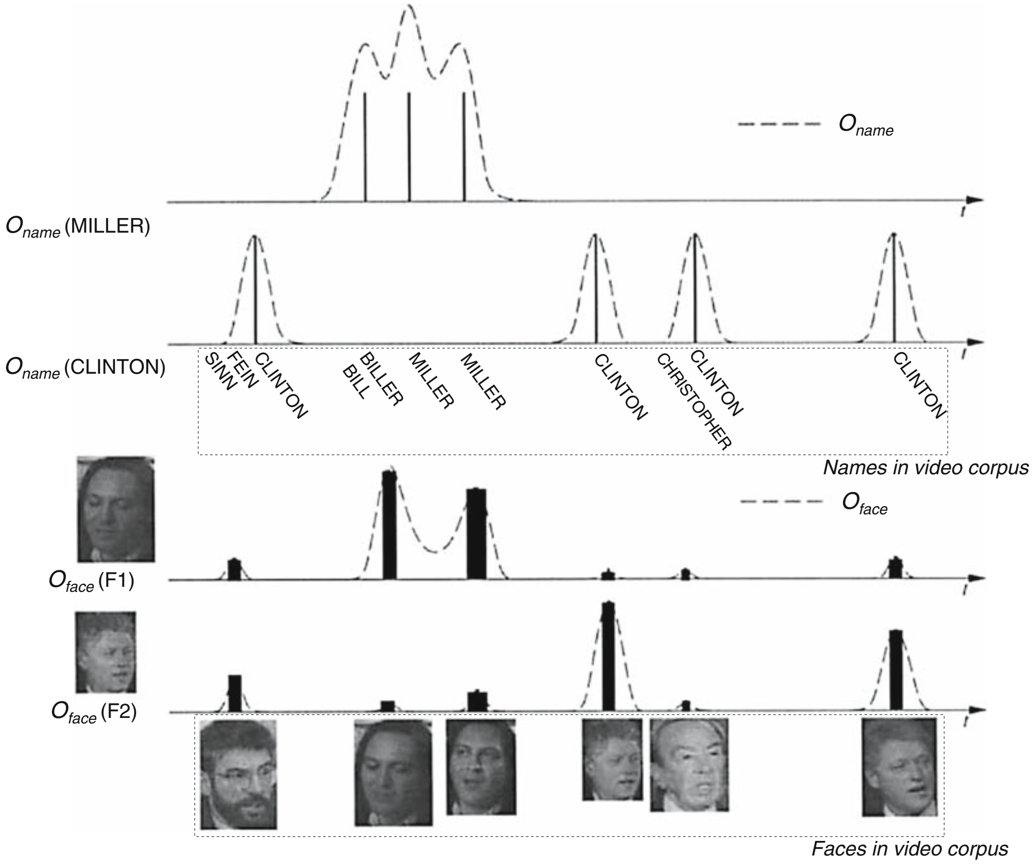
A key idea of Name-It is to evaluate co-occurrence between a face and name by comparing the occurrence patterns of the face and name in news video corpus. To do so, it is obviously required to locate a face and name in video corpus. It is rather straight forward to locate names in closed-captioned video corpus, since closed-caption text is symbol information. In order to locate faces, a face matching technique is used. In other words, by face matching, face information in news video corpus is symbolized. This enables co-occurrence evaluation between faces and names. Similar techniques can be found in the natural language processing and information retrieval fields. For instance, the vector space model [2] regards that documents are similar when they share similar terms, i.e., have similar occurrence patterns of terms. In Latent Semantic Indexing [3], terms having similar occurrence patterns in documents within corpus compose a latent concept. Similar to these, Name-It finds face-name pairs having similar occurrence patterns in news video corpus as associated face-name pairs. Figure 4 shows occurrence patterns of faces and names. Co-occurrence of a face and name is realized by correlation between occurrence patterns of the face and name. In this example, “MILLER” and F1, “CLINTON” and F2, respectively, will be associated because corresponding occurrence patterns are similar.

Conclusions and Future Directions

This article describes about face-name association in videos, especially Name-It, in order to demonstrate the effectiveness of corpus-based video analysis. There are potential directions to enhance and extend corpus-based face-name association. One possible direction is to elaborate component technologies such as name extraction, face extraction, and face matching. Recent advanced information extraction and natural language processing techniques enable almost perfect name extraction from text. In addition, they can provide further information such as roles of names in sentences and documents, which surely enhances the face-name association performance.



Analyzing Person Information in News Video. Figure 3. Face and name association results.



Analyzing Person Information in News Video. Figure 4. Face and name occurrence patterns.

Advanced image processing or computer vision techniques will enhance the quality of symbolization of faces in video corpus. Robust face detection and tracking in videos is still challenging task (such as [4]. In [5] a comprehensive survey of face detection is presented). Robust and accurate face matching will rectify the occurrence patterns of faces (Fig. 4), which enhances face-name association. Many research efforts have been made in face recognition, especially for surveillance and biometrics. Face recognition for videos could be the next frontier. In [6] a comprehensive survey for face recognition is presented. In addition to face detection and recognition, behavior analysis is also helpful, especially to associate the behavior with person's activity described in text.

Usage of the other modalities is also promising. In addition to images, closed-caption text, and video captions, speaker identification provides a powerful cue for face-name association for monologue shots [7].

In integrating face and name detection results, Name-It uses co-occurrence, which is based on coincidence. However, as mentioned before, since news videos are concise and easy to understand for people, relationship between corresponding faces and names is not so simple as coincidence, but may yield a kind of video grammar. In order to handle this, the system ultimately needs to "understand" videos as people do. In [8] an attempt to model this relationship as temporal probability distribution is presented. In order to enhance the integration, we need much elaborated video grammar, which intelligently integrate text processing results and image processing results.

It could be beneficial if corpus-based video analysis approach is applied to general objects in addition to faces. However, obviously it is not feasible to realize detection and recognition of many types of objects. Instead, in [9] one of the promising approaches is presented. The method extracts interest points from videos, and then visual features are calculated for each point. These points are then clustered by features into "words," and then a text retrieval technique is applied for object retrieval for videos. By this, the method symbolizes objects shown in videos as "words," which could be useful to extend corpus-based video analysis to general objects.

References

1. J. Yang, M. Chen, and A. Hauptmann, "Finding Person X: Correlating Names with Visual Appearances," Proceedings of the International Conference on Image and Video Retrieval (CIVR'04), Dublin, Ireland, Lecture Notes in Computer Science, Vol. 3115, 2004, pp. 270–278.
2. R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," Addison-Wesley, Reading, MA, 1999.
3. S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, Vol. 41, 1990, pp. 391–407.
4. R.C. Verma, C. Schmid, and K. Mikolajcayk, "Face Detection and Tracking in a Video by Propagating Detection Probabilities," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, 2003, pp. 1216–1228.
5. M.-H. Yang, D.J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, 2002, pp. 34–58.
6. W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," ACM Computing Surveys, Vol. 35, No. 4, 2003, pp. 399–458.
7. M. Li, D. Li, N. Dimitrova, and I. Sethi, "Audio-Visual Talking Face Detection," Proceedings of the International Conference on Multimedia and Expo (ICME 2003), Vol. 1, pp. 473–476, 2003.
8. C.G.M. Snoek and A.G. Haptmann, "Learning to Identify TV News Monologues by Style and Context," CMU Technical Report, CMU-CS-03–193, 2003.
9. J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," Proceedings of the International Conference on Computer Vision (ICCV 2003), Nice, France, Vol. 2, 2003, pp. 1470–1477.
10. S. Satoh, Y. Nakamura, and T. Kanade, "Name-It: Naming and Detecting Faces in News Videos," IEEE MultiMedia, Vol. 6, No. 1, January-March, 1999, pp. 22–35.

Application of Computational Intelligence in Content Adaptation

► Neural Networks in Multimedia Content Adaptation

Applications of Face Recognition and Novel Trends

Definition

A number of contemporary civilian and law enforcement applications require reliable recognition of human faces.

Nowadays, machine recognition of human faces is used in a variety of civilian and law enforcement applications that require reliable recognition of humans. Identity verification for physical access control in buildings or security areas is one of the most common face recognition applications. At the access point, an image of someone's face is captured by a camera and is matched against pre-stored images of the same person. Only if there is a match, access is permitted, e.g., the door opens. For high security areas, a combination with card terminals is possible, so that a double check is performed. Such face recognition systems are installed for example in airports to facilitate the crew and airport staff to pass through different control levels without having to show an ID or passport [1].

To allow secure transactions through the Internet, face verification may be used instead of electronic means like passwords or PIN numbers, which can be easily stolen or forgotten. Such applications include secure transactions in e- and m-commerce and banking, computer network access, and personalized applications like e-health and e-learning. Face identification has also been used in forensic applications for criminal identification (mug-shot matching) and surveillance of public places to detect the presence of criminals or terrorists (for example in airports or in border control). It is also used for government applications like national ID, driver's license, passport and border control, immigration, etc.

Face recognition is also a crucial component of ubiquitous and pervasive computing, which aims at incorporating intelligence in our living environment and allowing humans to interact with machines in a natural way, just like people interact with each other. For example, a smart home should be able to recognize the owners, their family, friends and guests, remember their preferences (from favorite food and TV program to room temperature), understand what they are saying, where are they looking at, what each gesture, movement or expression means, and according to all these cues to be able to facilitate every-day life. The fact that face recognition is an essential tool for interpreting human actions, human emotions, facial expressions, human behavior and intentions, and is also an extremely natural and non-intrusive technique, makes it an excellent choice for ambient intelligence applications [2].

During the last decade wearable devices were developed to help users in their daily activities. Face recognition is an integral part of wearable systems like memory aids or context-aware systems [2]. A real-world application example is the use of mini-cameras and face recognition software, which are embedded into an Alzheimer's patient's glasses, to help her remember the person she is looking at [2].

Cross-References

► Face Recognition

References

1. A. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 1, January 2004, pp. 4–20.
2. A. Pentland and T. Choudhury, "Personalizing Smart Environments: Face Recognition for Human Interaction," IEEE Computer Magazine, Vol. 33, No. 2, February 2000, pp. 50–55.

Architecture of Commercial News Systems

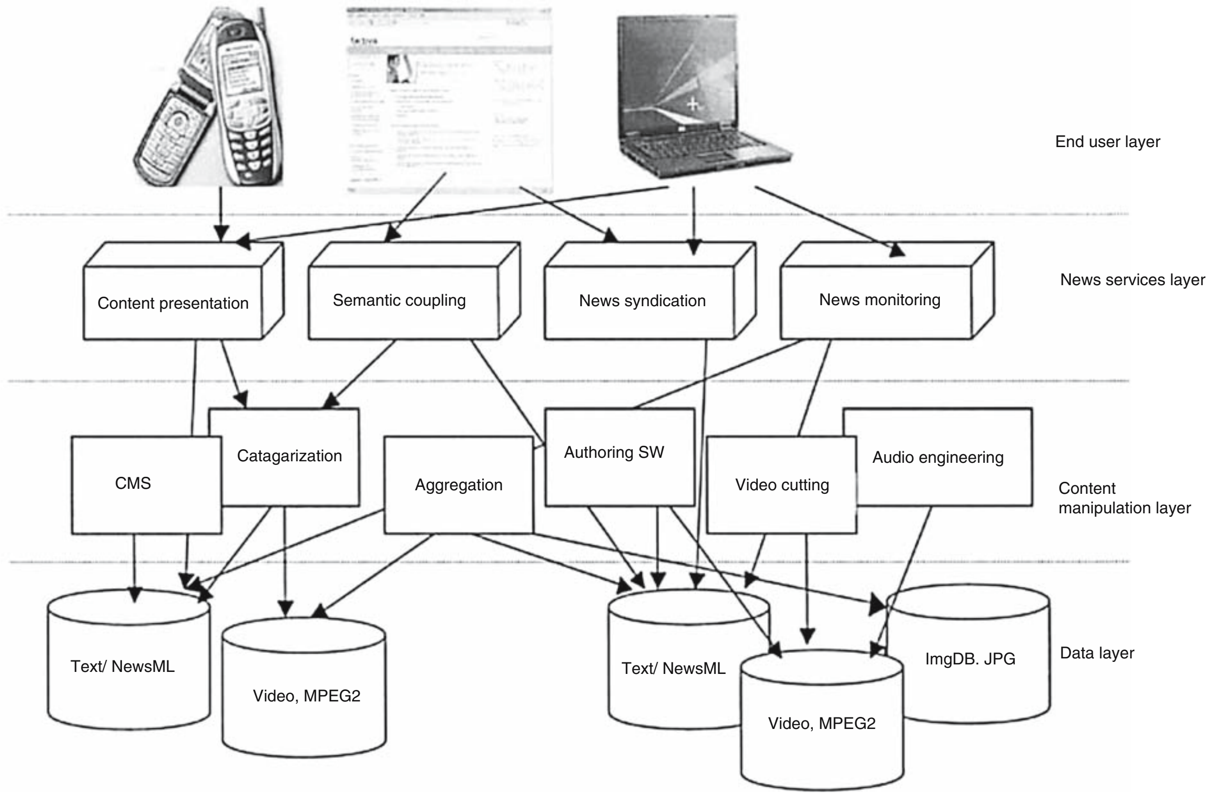
Definition

The architecture of commercial news systems is based on a layered approach consisting of the following layers: data layer, content manipulation layer, news services layer, and end user layer.

Multimedia news is presented as content of commercial services by national and international agencies and organizations all over the world. The news community is researching for solutions in different application areas, such as high level semantic analysis, provisioning and management of mixed information, and distribution and presentation of media data to satisfy requirements dictated by business scenarios.

Modern news networks, news syndication services, media observation and international news exchange networks are following the customer needs and provide specific services within the multimedia news application domain. The most-used presentation platform for multimedia news herein is the World Wide Web, providing all facets of news aggregation, manipulation, and dissemination as discussed in "Multimedia news systems." Appropriate Web applications integrate multimedia services in complex environments [1] and modern web-based content management systems (WCMS) handle all assets of multimedia news data for personalized user-oriented news presentation.

Multimedia news systems typically follow a layered architecture approach as shown in Fig. 1. The data layer contains multimedia data that are stored in modern appropriate formats like NewsML for texts, modern image formats such as JPG, PNG, etc. and current versions of multimedia encoding (e.g., MPEG versions) for audio and video files.



Architecture of Commercial News Systems. Figure 1. Multimedia news systems layering.

The content manipulation layer provides access to the multimedia data via specific tools that provide methods to control and access news contents along the various transitions in the content lifecycle. The news services layer includes gateways that provide structured and standardized access to the contents by end user applications. Within this layer, tools and services of content provider networks take care of the presentation and distribution of multimedia contents. Most providers run multimedia gateways such as streaming servers or web services and sites to present the multimedia contents.

The top layer presents the end user environment, providing access to multimedia news services based on direct access to multimedia gateways or special services of the news services layer such as multi-agency full-text search engines, semantically coupling services or commercially related gateways like billing servers or subscription access gateways.

Cross-References

▶ Multimedia News Systems

References

1. E. Kirda, M. Jazayeri, C. Kerer, and M. Schranz, "Experiences in Engineering Flexible Web Services," IEEE Multimedia, Vol. 8, No. 1, January–March 2001, pp. 58–65.

Architecture of Object-based Multimedia Storage Systems

▶ Large-Scale Object-Based Multimedia Storage Systems

Area of Interest Management

▶ Networking for Massively Multiuser Online Gaming

Artificial Intelligence (AI)-based Techniques for Finding a Sequence of Adaptation Operations

► Knowledge-based Multimedia Adaptation Decision-taking

Audio and Video Information in Multimedia News Systems

Definition

Contemporary news systems contain today visual media including audio and video information.

Historically most relevant information concerning progress in sciences and the growth of general information within human knowledge and accessible to mankind has been documented in written text and occasionally described in images and maps. The technological developments of the twentieth century has brought tools and communication channels that increased the ways of communicating and distributing news concerning all facets of human live in various ways. Multimedia news services involve now besides enormous amounts of digitized textual information also images, audio and video material, supported by multiple technical tools for seamless integration and modern distribution.

Image captures people's thrills, emotions, and concerns. Art can shock or inspire, and news images cover most relevant events over the entire globe. Most readers depend on visual and multimedia contents to understand the world around them and as a basis for further creative activities. The popularity of visual media such as photos, videos, and animations attests to their mainstream acceptance.

Technically, multimedia integration has been adopted to middleware software and content management applications to allow a seamless integration with current news management and distribution applications. XML standards like SMIL provide synchronization and integration frameworks and document description standards that cope with traditional and future text information and multimedia data that support modern user's needs.

Modern multimedia services integrate content types of text, images, audio data and streams and current video formats such as MPEG2, MPEG4, MPEG7 and multimedia broadcasting technologies and initiatives like MHP and DVB for providing future interactive digital news access via television, internet and mobile devices.

General purpose news editing systems integrate easy-to-use textual interfaces, mostly based on Web architectures, with modern multimedia features like video studios, composing for example MPEG4 audio-visual scenes (cf. MPEG4 STUDIO in [1]).

Modern Internet protocols like HTTP for web-based presentation of text and images, and streaming protocols such as RTSP for audio and video streaming cover the distribution and presentation services of multimedia news systems. Besides the technical enabling of news presentation and distribution, multimedia news services have to face legal and commercial constraints. Recent misuse by illegal file sharing or copyright violence has introduced security topics to multimedia news services. Besides customer oriented access restrictions and modern billing systems, multimedia news have been enriched with up-to-date encrypting mechanisms. Especially for audio and video formats, the time constraints in encoding and decoding live streams have introduced challenges which are met by modern approaches such as encryption and watermarking for copyrighted MPEG [2]. Multimedia data security is vital for multimedia commerce. Early cryptography have focused and solved text data security. For multimedia applications, light weight encryption algorithms as discussed in [3] are attractive and appropriate.

Cross-References

► Multimedia News Systems

References

1. K. Cha and S. Kim, "MPEG-4 STUDIO: An Object-Based Authoring System for MPEG-4 Contents," Multimedia Tools and Applications, Vol. 25, No. 1, January 2005, pp. 111–131.
2. D. Simitopoulos, N. Zissis, P. Georgiadids, V. Emmanouilidis, and M. Strintzis, "Encryption and Watermarking for the Secure Distribution of Copyrighted MPEG Video on DVD," Multimedia Systems, Vol. 9, No. 3, September 2003, pp. 217–227.
3. B. Bhargava, C. Shi, and S. Wang, "MPEG Video Encryption Algorithms," Multimedia Tools and Applications, Vol. 24, No. 1, September 2004, pp. 57–79.

Audio Compression and Coding Techniques

Jauvane C. de Oliveira
National Laboratory for Scientific Computation,
Petropolis, Brazil

Synonyms

▶ Audio codecs

Definition

Audio compression and coding techniques are used to compress audio signals and can be based on sampling or on signal processing of audio sequences.

Audio is the most important medium to be transmitted in a conference-like application. In order to be able to successfully transmit audio through a low bandwidth network, however, one needs to compress it, so that its required bandwidth is manageable.

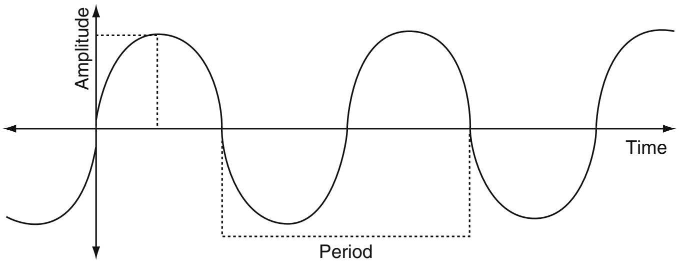
Introduction – Audio Properties

Sound is a phenomenon that happens due to the vibration of material. Sound is transmitted through the air, or some other elastic medium, as pressure waves that are formed around the vibrating material. We can consider the example of strings of a guitar, which vibrate when stroked upon. The pressure waves follow a pattern named *wave form* and occur repeatedly at regular intervals of time. Such intervals are called a *period*. The amount of periods per second denotes what is known as the *frequency* of sound, which is measured in Hertz (Hz) or cycles per second (cps) and is denoted by *f*. *Wavelength* is the space the wave form travels in one period. It may also be understood as the distance between two crests of the wave. The waveform is denoted by λ. Yet with regard to the wave form, the intensity of the deviation from its mean value denotes the *amplitude* of the sound. Figure 1 shows an example of an audio signal, where we can observe both its amplitude and period. The velocity of sound is given by $c = f\lambda$. At sea level and 208 C (688 F), $c = 343$ m/s.

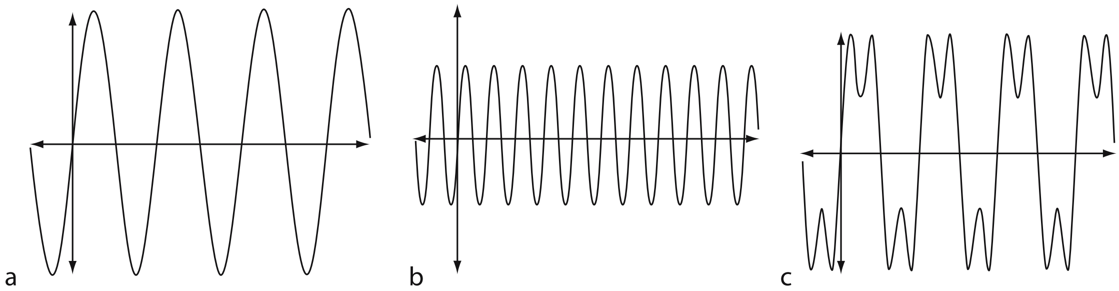
A sound wave is an analog signal, as it assumes continuous values throughout the time. Using a mathematical technique called Fourier Analysis one can prove that any analog signal can be decomposed as a, possibly infinite, summation of single-frequency sinusoidal signals (See Fig. 2). The range of frequencies which build up a given signal, i.e., the difference between the highest and lowest frequency components, is called *signal bandwidth*. For a proper transmission of an analog signal in a given medium that must have a bandwidth equal or greater than the signal bandwidth. If the medium bandwidth is lower than the signal bandwidth some of the low and/or high frequency components of the signal will be lost, which degrades its quality of the signal. Such loss of quality is said to be caused by the *bandlimiting channel*. So, in order to successfully transmit audio in a given medium we need to either select a medium whose bandwidth is at least equal to the audio signal bandwidth or reduce the signal bandwidth so that it *fits* in the bandwidth of the medium.

Audio Digitization Codec

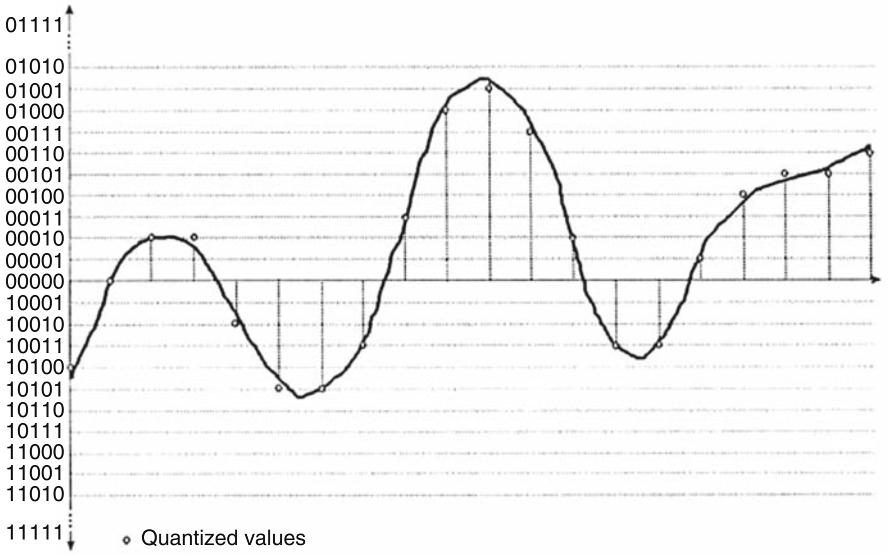
In order to process audio in a computer, the analog signal needs to be converted into a digital representation. One common digitization technique used is the Pulse Code Modulation (PCM). Basically we'll set a number of valid values in the amplitude axis and later we will measure the amplitude of the wave a given number of times per second. The measurement at a given rate is often referred to as *sampling*. The sampled values are later rounded up or down to the closest valid value in the amplitude axis. The rounding of samples is called *quantization*, and the distance from one value to the next refers to as a *quantization interval*. Each quantization value has a well-defined digital bitword



Audio Compression and Coding Techniques. Figure 1. Sound wave form with its amplitude and period.



Audio Compression and Coding Techniques. Figure 2. (a, b) Two sinusoidal components and (c) its resulting summation.



Audio Compression and Coding Techniques. Figure 3. Digitization: samples (vertical dashed), quantized values (dots) and bitwords (left).

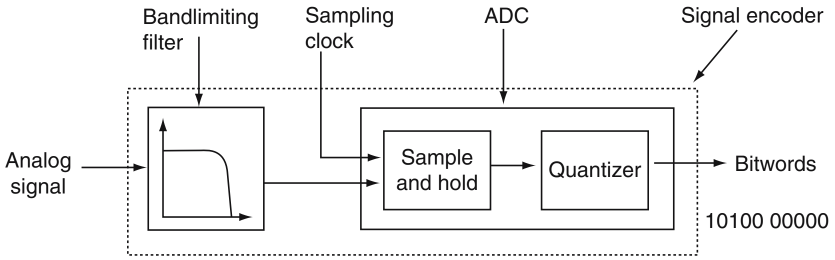
to represent it. The analog signal is then represented digitally by the sequence of bitwords which are the result of the sampling + quantization. Figure 3 shows this procedure, whose digital representation of the signal is 10100 00000 00010 00010 10010 10101 10101 10011 00011 01000 01001 00111 00010 10011 10011 00001 00100 00101 00101 00110.

Harry Nyquist, a physicist who worked at AT&T and Bell Labs, developed in 1927 a study with regard to the optimum sampling rate for a successful digitization of an analog signal. The *Nyquist Sampling Theorem* states that the sampling frequency must be greater than twice the bandwidth of the input signal in order to allow a successful reconstruction of the original signal out of the sampled version. If the sampling is performed at a frequency lower than the *Nyquist Frequency* then the number of samples may be insufficient to reconstruct the original signal, leading to a distorted reconstructed signal. This phenomenon is called *Aliasing*.

One should notice that for each sample we need to round it up or down to the next quantization level, which leads to what is called *quantization error*. Such procedure actually distorts the original signal. *Quantization noise* is the analog signal which can be built out of the randomly generated quantization errors.

In order to reconstruct the analog signal using its digital representation we need to interpolate the values of the samples into a continuous time-varying signal. A bandlimiting filter is often employed to perform such procedure.

Figure 4 shows a typical audio encoder. Basically we have a bandlimiting filter followed by an Analog-to-Digital Converter (ADC). Such converter is composed of a circuit which samples the original signal as indicated



Audio Compression and Coding Techniques. Figure 4. Signal encoder.

by a sampling clock and holds the sampled value so that the next component, a quantizer, can receive it. The quantized, in its turn, receives the sampled value and outputs the equivalent bitword for it. The bandlimiting filter is employed to ensure that the ADC filter won't receive any component whose Nyquist rate could be higher than the sampling clock of the encoder. That is, the bandlimiting filter cuts off frequencies which are higher than half of the sampling clock frequency.

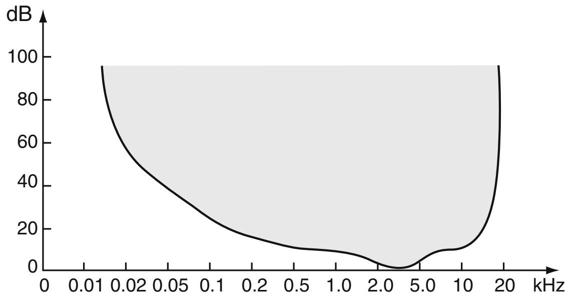
The audio decoder is a simpler device that is composed of a Digital-to-Analog Converter (DAC), which receives the bitwords and generates a signal that maintains the sample value during one sampling interval until the next value gets decoded. Such "square" signal then goes through a low-pass filter, also known as *reconstruction filter*, which smoothens it out to what would be equivalent to a continuous-time interpolation of the sample values.

The Human Hearing/Vocal Systems and Audio Coding

The human hearing system is capable of detecting by sounds whose components are in the 20 Hz–20 KHz range. The human voice, in the other hand, can be characterized in the 50 Hz–10 KHz range. For that reason, when we need to digitize human voice, a 20 Ksps (samples per second) sampling rate is sufficient according to the Nyquist Sampling Theorem. More generally, since we can't hear beyond 20 KHz sinusoidal components, a generic sound such as music can be properly digitized using a 40 Ksps sampling rate.

The above-mentioned characteristics of the human audio-oriented senses can be used to classify sound processes as follows:

1. Infrasonic: 0–20 Hz
2. Audiosonic: 20 Hz–20 KHz
3. Ultrasonic: 20 kHz–1 GHz
4. Hypersonic: 1 GHz–10 THz



Audio Compression and Coding Techniques. Figure 5. Human hearing sensibility.

The human hearing system is not linearly sensible to all frequencies in the audiosonic range. In fact the curve shown in Fig. 5 shows the typical hearing sensibility to the various frequencies.

With regard to the quantization levels, using linear quantization intervals, it is usual to use 12 bits per sample for voice encoding and 16 bits per sample for music. For multi-channel music we'd use 16 bits for each channel. We can then find that we would use respectively 240, 640, and 1280 Kbps for digitally encoded voice, mono and stereo music. In practice, however, since we have much lower network bitrate available than those mentioned herein, we'll most often use a lower sampling rate and number of quantization levels. For telephone-quality audio encoding, for instance, it is common to sample at 8 Ksps, obviously cutting off sinusoidal components with frequencies over 4 KHz in order to comply with the Nyquist sampling theorem.

Sampling Based Audio Compression Schemes

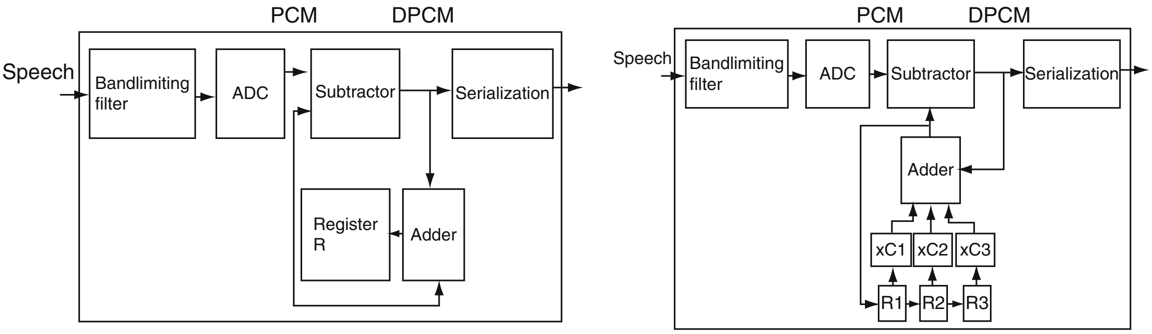
There are a number of standard compression schemes, which are based on the samples and that are not specific for any type of audio, being hence useable for both voice and music, with the appropriated adaptations with regard to the frequency range considered.

Pulse Code Modulation (*PCM*)*:* Pulse Code Modulation, defined in the ITU-T Recommendation G.711 [1], is a standard coding technique defined for voice encoding for transmission over telephone lines. A typical telephone line has a bandwidth limited to the range from 200 Hz to 3.4 KHz. For this rate a 6.8 Ksps sampling frequency would suffice, but in order to accommodate low quality bandlimiting filters, an 8 Ksps sampling frequency is employed. PCM uses 8 bits per sample rather than 12, with a compression/expansion circuit being used to achieve a sound quality equivalent to a normal 12 bits per sample encoding. Basically what the compression/expansion circuit does is to indirectly implement non-linear quantization levels, i.e. the levels are closer together for smaller samples and farther apart for larger ones. That minimizes quantization error for smaller samples, which leads to a better overall audio quality. Instead of really using logarithmic quantization levels, the signal is compressed and later linear quantized. The result is nevertheless equivalent to quantizing with logarithmic distributed quantization levels. There are two standard compression/expansion circuits: u-Law, which is used in the North America and Japan; and A-law, which is used in Europe and other countries. With that a telephone-like audio coded with PCM reaches a total of 64 Kbps.

Compact Disc Digital Audio (*CD-DA*)*:* Music has sinusoidal components with frequencies in the 20 Hz–20 KHz range. That requires at least a 40 Ksps sampling rate. In practice a 44.1 Ksps is used to accommodate filter discrepancies. Each sample is then ended with 16 bits using linear quantization levels. For stereo recordings there shall be 16 bits for each channel. Such coding scheme reaches a total of 705.6 Kbps for mono and 1.411 Mbps for stereo music.

Differential Pulse Code Modulation (*DPCM*)*:* Further compression is possible in an audio signal through the analysis of typical audio samples. If we analyze a sound wave form, we can see that at the Nyquist sampling rate the wave change from one sample to the next is not very abrupt, i.e. the difference between two consecutive samples is much smaller than the samples themselves. That allows one to naturally use a sample as a prediction to the next one, having to code just the difference to the previous rather than the each sample separately. The difference between two samples can be coded with a smaller number of bits, as its maximum value is smaller than the sample itself. That’s the motto behind Differential PCM, or DPCM. Typical savings are of about 1 bit per sample; hence a 64 Kbps voice stream gets compressed to 56 Kbps. The problem with this coding scheme is that quantization errors can accumulate if the differences are always positive (or negative). More elaborated schemes may use various previous samples that are mixed together using *predictor coefficients*, which consists of proportions of each previous sample that is to be used to build the final prediction. Figure 6 shows both the DPCM encoder with a single and three previous values used for prediction.

Adaptive Differential Pulse Code Modulation (*ADPCM*). Extra compression can be achieved by varying the number of bits used to encode different signal components, depending on their maximum amplitude. The former ITU-T G.721 Recommendation,



Audio Compression and Coding Techniques. Figure 6. DPCM encoders with single (left) and third order predictions (right).

now part of the ITU-T G.726 Recommendation [2], uses the same principle as DPCM, but using a eight-order prediction scheme, with either 6 or 5 bits per sample for a total of 32 or 16 Kbps. The ITU-T G722 Recommendation [3] adds another technique called *subband coding*. Such technique consists of extending the speech bandwidth to 50 Hz–7 KHz (rather than cutting off at 3.4 KHz like in PCM) and passing the signal through two filters: the first allows only frequencies from 50 Hz to 3.5 KHz while the second allows only frequencies from 3.5 to 7 KHz. The two signals are named *lower subband* and *upper subband* signals. Each subband signal is then sampled independently, respectively at 8 and 16 Ksps, and quantized using specific tables. The bitstreams are finally merged together in a last stage. This standard leads to 64, 56, or 8 Kbps. We should notice that this standard reaches higher voice quality encoding as it also considers higher frequencies in the 3.4–7 KHz range. Yet another ADPCM-based standard, ITU-T G.726 Recommendation [2], also uses the subband coding technique described above, but considering only 50 Hz–3.4 KHz components, with bitstreams at 40, 32, 24 or 16 Kbps.

Adaptive Predictive Coding (APC): Further compression can be achieved if we use adaptive coefficient predictors, which is the basis for a compression technique called Adaptive Predictive Coding, where such coefficients change continuously based on characteristics of the audio signal being encoded. An audio sequence is split into small audio segments, each of which is then analyzed aiming at selecting optimum predictive coefficients. Such compression scheme can reach 8 Kbps with reasonable quality.

Digital Signal Processing Based Audio Compression Schemes

We call psycho-acoustic system what comprises those two systems. The first consists of all electrical/nervous systems linked to the communication from the senses to the brain and vice-versa and the latter comprises the generation/capture of sound which is transmitted through a given medium, such as the air, to/from the other party of the communication. The human speech is generated by components that come from the diaphragm all the way up to the human lips and nose. Through analysis of the human voice and the psycho-acoustic model of the human being, there is a class of compression schemes which makes use of digital signal processing circuits that are inexpensive as of today inexpensive. In this section, we describe a number of those compression schemes.

Linear Predictive Coding (LPC): The Linear Predictive Coding is based on signal processing performed in the source audio aiming at extracting a number of its perceptual features. Those are later quantized and transmitted. At the destination such perceptual features feed a voice synthesizer which generates a sound that can be perceived as the original source audio. Although the sound does sound synthetic, this algorithm reaches very high compression rates, leading to a low resulting bitstream. Typical output bitstreams reach as low as 1.2 Kbps.

The perceptual features that are commonly extracted from voice signals are pitch, period, loudness as well as voice tract excitation parameters. *Pitch* is related to the frequency of the signal, *period* is the duration of the signal and *loudness* relates to the power of the signal. The *voice tract excitation parameters* indicated if a sound is voice or unvoiced. *Voiced* sounds involve vibrations of the human vocal cords while *unvoiced* sounds do not. Lastly *vocal tract model coefficients* are also extracted. Such coefficients indicate probable vocal tract configuration to pronounce a given sound. Such coefficients later feed a basic vocal tract model which is used to synthesize audio at the destination.

Code-excited LPC (CELP): A group of standards which are based on a more elaborate model of the vocal tract is also used. Such model is known as Code Excited Linear Prediction (CELP) model and is one of various models known as enhanced excitation LPC models. This compression scheme achieves better sound quality than LPC. Standards such as ITU-T G.728 [4], G.729 [5], G.723.1 [3] are based in CELP. Those standards achieve respectively 16, 8, and 5.3 or 6.3 Kbps. The price paid for such low final bitrate is the time it takes for the encoding to be performed. Respectively 0.625, 23, and 67.5 ms.

Perceptual Coding: If we expect to compress generic audio such as music, the previous LPC and CELP are not useable, as those are based on a vocal tract model for audio synthesis. Perceptual Coding is a technique which exploits the human hearing system limitations to achieve compression with not perceived quality loss. Such compression scheme also requires digital signal processing, to analyze the source audio, before it gets compressed. Features explored include: (1) the human hearing sensibility, as shown in Fig. 5, where we can cut off signal components whose frequencies have an amplitude which is below the minimum shown, i.e., if a signal component at 100 Hz is under 20 dB it is not inaudible, (2) frequency masking, which consists of the fact that when we hear a sound that is composed of several waves, if a loud wave is close (frequency-wise) to a low wave, the low wave is not heard because of the sensitivity curve for the human ear, as shown in Fig. 5, that gets distorted for frequencies around a given loud wave, much like if the sensitivity levels were pushed up a bit, (3) temporal masking, which consists of the fact that when we hear a loud sound we get deaf for quieter sounds for a short period. When we hear an explosion, for instance, we can't hear quieter noises for a while. All those inaudible sounds can be fully discarded and go unnoticed.

MPEG-Audio: The Motion Picture Expert Group (MPEG), set by ISO to define a number of standards related to multimedia applications that use video and sound, defined a number of MPEG audio coders based on Perceptual Coding. Basically a source audio is sampled and quantized using PCM with a sampling rate and number of pixels per sample determined by the application. In a next step the bandwidth is split in 32 frequency subbands using analysis filters. Such subbands go through a Discrete Fourier Transform (DFT) filter to convert the samples to the frequency domain. In a further step, using the human hearing limitations some frequencies are cut off. For the remaining audible components, quantization accuracy is selected along with the equivalent number of bits to be used. That way, less quantization (and more bits) can be used for the frequencies for which we are most sensible to, such as the range from 2 to 5 KHz. In the decoder, after dequantizing each of the 32 subband channels, the subbands go through the synthesis filter bank. That component generates PCM samples which are later decoded to generate an analog audio. The ISO Standard 11172–3 [6] defines three levels of processing through layers 1, 2, and 3; the first being the basic mode and the other two with increasing level of processing associated, respectively with higher compression or better sound quality if bitrate is kept constant.

Dolby AC-1, AC-2, and AC-3: Other coding schemes based on Perceptual Coding are the Dolby AC-1, AC-2, and AC-3. AC stands for *acoustic coder*. Dolby AC-1 is basically a standard for satellite FM relays and consists of a compression scheme based in a low-complexity psychoacoustic model where 40 subbands are used at a 32 Ksps sampling rate and fixed bit allocation. The fix bit allocation avoids the need to submit the bit allocation information along with the data. Dolby AC-2 is used by various PC sound cards, producing hi-fi audio at 256 Kbps. Even tough the encoder uses variable bit allocations, there is no need to send that information along with the data because the decoder also contain the same psychoacoustic model used by the encoder, being able to compute the same bit allocations. In the negative side, if any change is to be made in the model used by the encoder all decoders need to be changed as well. The decoder needs to have the subband samples to feed the psychoacoustic model for its own computation of bit allocations, reason why each frame contains the quantized samples as well as the encoded frequency coefficients from the sampled waveform. That information is known as the *encoded spectral envelope* and that mode of operation is known as *backward adaptive bit allocation mode*. Dolby AC-3 uses both backward and forward bit allocation principles, which is known as *hybrid backward/forward adaptive bit allocation mode*. AC-3 has defined sampling rates at 32, 44.1, and 48 Ksps and uses 512 subband samples per block, of which only 256 subsamples are updated in each new block, since the last 256 subbands of the previous block become the first 256 subbands of the new block.

Cross-References

► Human Vocal System, Human Hearing System

References

1. ITU-T G.711 Recommendation, "Pulse Code Modulation (PCM) of Voice Frequencies," International Telecommunication Union, Telecommunication Standardization Sector.
2. ITU-T G.726 Recommendation, "40, 32, 24, 16 Kbit/s adaptive differential pulse code modulation (ADPCM)," International Telecommunication Union, Telecommunication Standardization Sector.
3. ITU-T G.722 Recommendation, "7 KHz Audio-coding Within 64 Kbits/s," International Telecommunication Union, Telecommunication Standardization Sector.
4. ITU-T G.728 Recommendation, "Coding of speech at 16 Kbit/s using low-delay code excited linear prediction," International Telecommunication Union, Telecommunication Standardization Sector.
5. ITU-T G.729 Recommendation, "Coding of speech at 8 Kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)," International Telecommunication Union, Telecommunication Standardization Sector.

6. ISO/IEC 11172–3 “Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 3: Audio,” International Organization for Standardization.
7. T.F. Quartieri, “Speech Signal Processing – Principles and Practice,” Prentice Hall, Englewood Cliffs, NJ, 2001.
8. B. Gold and N. Morgan, “Speech and Audio Signal Processing – Processing and Perception of Speech and Music,” Wiley, New York, 2000.
9. F. Halsall, “Multimedia Communications – Applications, Networks, Protocols and Standards,” Addison Wesley, Reading, MA, 2001.
10. R. Steinmetz and K. Nahrstedt, “Multimedia Fundamentals Volume I – Media Coding and Content Processing,” Prentice Hall, Englewood Cliffs, NJ, 2002.
11. ITU-T G.723.1 Recommendation, “Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 Kbit/s,” International Telecommunication Union, Telecommunication Standardization Sector.

Audio Conferencing

Definition

Audio conferencing allows participants in a live session to hear each other.

The audio is transmitted over the network between users, live and in real-time. Audio conferencing is one component of teleconferencing; the others are video conferencing, and data conferencing. Since the audio must be encoded, transmitted, and decoded in real-time, special compression and transmission techniques are typically used. In a teleconferencing system that is ITU-T H.323 [1] compliant, the G.711 [2] audio codec, which is basically uncompressed 8-bit PCM signal at 8 KHz in either A-Law or A-Law format, must be supported. This leads to bitrates of 56 or 64 Kbps, which are relatively high for audio but supported by today’s networks.

Support for other ITU-T audio recommendations and compression is optional, and its implementation specifics depend on the required speech quality, bit rate, computational power, and delay. Provisions for asymmetric operation of audio codecs have also been made; i.e., it is possible to send audio using one codec but receive audio using another codec. If the G.723.1 [3] audio compression standard is provided, the terminal must be able to encode and decode at both the 5.3 Kbps and the 6.3 Kbps modes. If a terminal is audio only, it should also support the ITU-T G.729 recommendation [4]. Note that if a terminals is known to be on a low-bandwidth network (<64 Kbps), it does not need to disclose capability to receive G.711 audio since it won’t practically be able to do so.

To transfer the live audio over the network, a protocol such as the RTP (Real-time Transport Protocol) [5], or simple UDP (User Datagram Protocol) is used.

Cross-References

► Teleconferencing

References

1. International Telecommunication Union, Telecommunication Standardization Sector H.323 Recommendation – Packet-based multimedia communications systems, July 2003.
2. International Telecommunication Union, Telecommunication Standardization Sector G.711 Recommendation – Pulse code modulation (PCM) of voice frequencies, November 1988.
3. International Telecommunication Union, Telecommunication Standardization Sector G.723.1 Recommendation – Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 Kbit/s, March 1996.
4. International Telecommunication Union, Telecommunication Standardization Sector G.729 Recommendation – Coding of speech at 8 Kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP), March 1996.
5. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” IETF RFC 1889, January 1996.

Audio Streaming

Shervin Shirmohammadi[1], Jauvane C. de Oliveira[2]
[1]University of Ottawa, Ottawa, ON, Canada
[2]National Laboratory for Scientific Computation, Petropolis, Brazil

Definition

Audio streaming refers to the transfer of audio across the network such that the audio can be played by the receiver(s) in real-time as it is being transferred.

Introduction

Audio streaming can be for various live media, such as the Internet broadcast of a concert, or for stored media, such as listening to an online jukebox. Real-time transfer and playback are the keys in audio streaming. As such, other approaches, such as downloading an entire file before playing it, are not considered to be streaming. From a high-level perspective, an audio streaming system needs to address three issues: audio compression, dissemination over the network, and playback at the receiver.

Audio Compression

Whether the audio is coming from a pre-stored file, or is captured live, it needs to be compressed to make streaming practical over a network. Uncompressed audio is bulky and most of the time not appropriate for transmission over the network. For example, even a low-quality 8 KHz 8-bit speech in the PCM format can take from 56 to 64 Kbps; anything with higher quality takes even more bandwidth. Compression is therefore necessary for audio streaming. Table 1 shows a list of several streaming standards, with the typical target bitrate, relative delay and usual target applications.

It should be noted that the delays disclosed in Table 1 are based on the algorithm. For example, for PCM at 8 KHz sampling, we have one sample at every 0.125 ms. Sample based compression schemes, such as PCM and ADPCM, are usually much faster than those that achieve compression based on the human vocal system or psychoacoustic human model like MP3, LPC, and CELP. So, for delay-conscious applications such as audio streaming, which needs to be in real-time, one may select a sample-based encoding scheme, bandwidth permitting; otherwise one of the latter would be a better choice as they achieve higher compression. For a detailed discussion about the audio compression schemes please see the "Compression and Coding Techniques, Audio" article. In the streaming context, the Real Audio (ra) and Windows Media Audio (wma) formats, from Real Networks and Microsoft Corp. respectively, are also used quite often.

Dissemination over the Network

Unlike elastic traffic such as email or file transfer, which are not severely affected by delays or irregularities in transmission speed, continuous multimedia data such as audio and video are inelastic. These media have a "natural" flow and are not very flexible. Interruptions in audio while streaming it is undesirable and creates a major problem for the end user because it distorts its real-time nature. It should be pointed out that delay is not always detrimental for audio, as long as the flow is continuous. For example, consider a presentational application where the audio is played back to the user with limited interaction capabilities such as play/pause/open/close. In such a scenario, if the entire audio is delayed by

Audio Streaming. Table 1. Characteristics of Compression Schemes

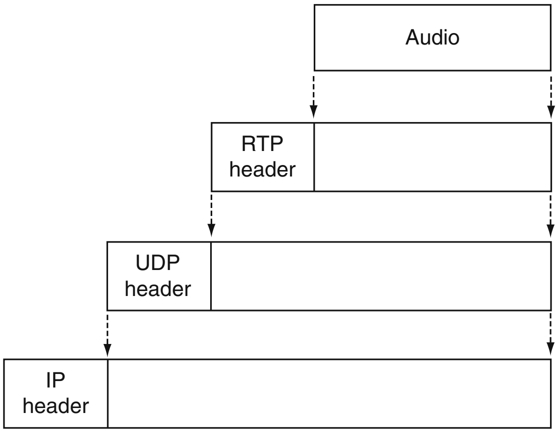
Standard	Compression	Target bitrate (Kbps)	Audio quality	Relative delay (ms)	Application
G.711 [1]	PCM + compression	64	Good Voice	0.125	PSTN/ISDN telephony
G.722 [2]	ADPCM w/subband coding	64 56/48	Excellent voice Good voice	Slightly higher than PCM	Audio conferencing
G.723.1 [3]	CELP	6.3 5.3	Good voice Fair voice	67.5	Video and Internet telephony videoconferencing
G.726 [4]	ADPCM w/subband coding	40/32 24/16	Good voice Fair voice		Telephony at reduced bitrates conferencing
G.728 [5]	CELP	16	Good voice	0.625	Low delay/low bitrate telephony
G.729 [6]	CELP	8	Good voice	25 ms	Telephony in cellular networks; Simultaneous telephony and data fax
LPC-10	LPC	2.4/1.2	Poor voice		Telephony in military networks
MP3 [7]	Perceptual coding	32–320	Music FM to CD quality		Music streaming

a few seconds, the user's perception of it is not affected due to lack of a reference point, as long as there are no interruptions. However, for a conversational application such as audio conferencing, where users interact with each other, audio delay must not violate certain thresholds because of the interaction and the existence of reference points between the users.

The transport protocol used for audio streaming must be able to handle the real-time nature of it. One of the most commonly-used real-time protocols for audio streaming is the Real-time Transport protocol (RTP), which is typically used with the Real Time Streaming Protocol (RTSP) for exchanging commands between the player and media server, and sometimes used with the Real -time Transport Control Protocol (RTCP) for Quality of Service (QoS) monitoring and other things. These protocols are briefly discussed next.

Real-time Transport Protocol (RTP)

To accommodate the inelasticity of audio streaming, there is a need for special networking protocols. The most common such protocol is the Real-time Transport Protocol (RTP) [8]. It is usually implemented as an application-level framing protocol on top of UDP, as shown in Fig. 1. It should be noted that RTP is named as such because it is used to carry real-time data; RTP itself does not guarantee real-time delivery of data. Real-time delivery depends on the underlying network; therefore, a transport-layer or an application-layer protocol cannot guaranty real-time delivery because it can't control the network. What makes RTP suitable for multimedia data, compared to other protocols, are two of its header fields: Payload Type, which indicates what type of data is being transported (Real Audio, MPEG Video, etc.), and Timestamp, which provides the temporal information for the data. Together with the Sequence Number field of the RTP header, these fields enable real-time playing of the audio at the receiver, network permitting. RTP supports multi-point to multi-point communications, including UDP multicasting.



Audio Streaming. Figure 1. RTP in the TCP/IP protocol suite.

Real-Time Transport Control Protocol (RTCP)

RTP is only responsible for transferring the data. For more capabilities, RTP's companion protocol the Real-time Transport Control Protocol (RTCP) can be used [8]. RTCP is typically used in conjunction with RTP, and it also uses UDP as its delivery mechanism. RTCP provides many capabilities; the most used ones are:

1. *QoS feedback*: The receiver can report the quality of their reception to the sender. This can include number of lost packets or the round-trip delay, among other things. This information can be used by the sender to adapt the source, if possible. Note that RTCP does not specify how the media should be adapted – that functionality is outside of its scope. RTCP's job is to inform the sender about the QoS conditions currently experienced in the transmission. It is up to the sender to decide what actions to take for a given QoS condition.
2. *Intermedia synchronization*: Information that is necessary for the synchronization of sources, such as between audio and video can be provided by RTCP.
3. *Identification*: Information such as the e-mail address, phone number, and full name of the participants can also be provided.
4. *Session Control*: Participants can send small notes to each other, such as "stepping out of the office," or indicate they are leaving using the BYE message, for example.

Real-Time Streaming Protocol (RTSP)

Unlike RTP which transfers real-time data, the Real Time Streaming Protocol (RTSP) [9] is only concerned with sending commands between a receiver's audio player and the audio source. These commands include *methods* such as SETUP, PLAY, PAUSE, and TEARDOWN. Using RTSP, an audio player can setup a session between itself and the audio source. The audio is then transmitted over some other protocol, such as RTP, from the source to the player. Similar to RTP, RTSP is not real-time by itself. Real-time delivery depends on the underlying network.

A Typical Scenario

Figure 2 demonstrates a typical scenario of audio streaming.

Here, the client first goes to a web site, where there is a link to the audio. Upon clicking on that link, the webs server sends to the receiver the URL of where the audio can be found. In the case of an RTSP session, the link looks something like rtsp://www.audioserver.com/audio.mp3. Note that the Web server and the audio server do not have to be the same entity; it is quite possible to separate them for better maintenance. After receiving the above link, the client's player established an RTSP link with the audio server through the SETUP command. The client can then interact with the server by sending PLAY, STOP, PAUSE, and other commands. Once the audio is requested for playing, RTP is used to carry the actual audio data. At the same time, RTCP can be used to send control commands between the client and the server. Finally, the session is finished with the TEARDOWN command of RTSP.

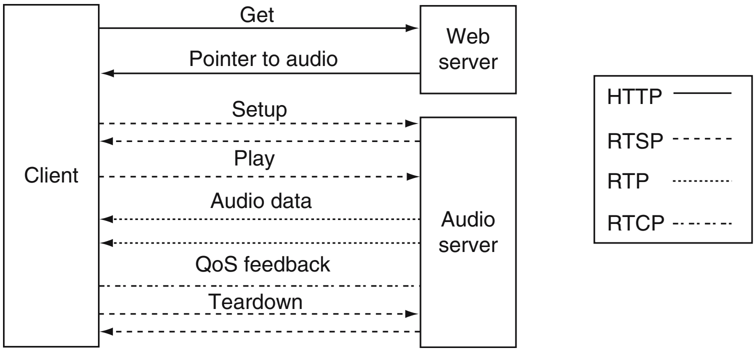
HTTP Streaming

HTTP streaming is an alternative to using RTP. The idea here is that the player simply requests the audio from the web server over HTTP, and plays it as the audio data comes in from the Web server. The disadvantages of this approach are the lack of RTP/RTCP features discussed above, and the fact that HTTP uses TCP which is not considered a real time protocol, especially under less-than ideal network conditions. As such, there can be more interruptions and delay associated with HTTP streaming compared to RTP streaming. However, HTTP streaming is used quite commonly for cases where the receiver has a high-speed Internet connection such as DSL and the audio bandwidth is not very high. In these cases, using HTTP streaming can be justified by its advantages; namely, the fact that HTTP is always allowed to go through firewalls, and that HTTP streaming is easier to implement as one can simply use an existing Web server.

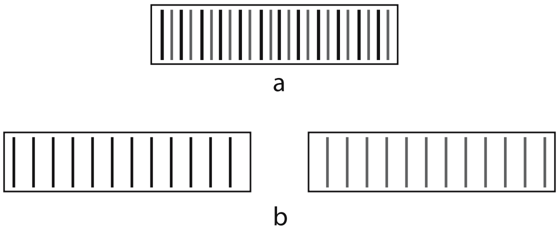
Playback at the Receiver

Although the coding and transmission techniques described above significantly contribute to the audio steaming process, the ultimate factor determining the real-time delivery of audio is the network condition. As mentioned above, delay severely affects audio in conversational applications. But for presentational applications, delay is less detrimental as long as it has a reasonable amount for a given application and is relatively constant. However, even for presentational applications, the variance of delay, known as jitter, has an adverse effect on the presentation. In order to smoothen out the delay, the player at the receiver's end usually buffers the audio for a certain duration before playing it. This provides a "safety margin" in case the transmission is interrupted for short durations. Note that the buffer cannot be too large, since it makes the user wait for too long before actually hearing the audio, and it cannot be too short since it won't really mitigate the effect of jitter in that case.

An extension to the above buffering technique is the faster-than-natural transmission of audio data. Depending on the buffer size of the receiver, the sender can transmit the audio faster than its normal playing



Audio Streaming. Figure 2. A typical sequence of events when streaming audio from a Web site.



Audio Streaming. Figure 3. (a) Transmission of consecutive samples, (b) Transmission of alternate samples.

speed so that if there are transmission interruptions, the player has enough data to playback for the user. This technique would work for stored audio, but it does not apply to live applications where the source produces audio at a natural speed.

Interleaved Audio

Interleaved audio transmission is a technique that is sometimes used to alleviate network loss and act as a packet loss resilience mechanism [10, 11]. The idea is to send alternate audio samples in different packets, as opposed to sending consecutive samples in the same packet. The difference between the two approaches is shown in Fig. 3. In Fig. 3(a) we see 24 consecutive samples being transmitted in one packet. If this packet is lost, there will be a gap in the audio equal to the duration of the samples. In Fig. 3(b) we see the interleaved approach for the same audio sample in Fig. 3(a), where alternate samples are being sent in separate packets. This way if one of the packets is lost, we only lose every other sample and the receiver will hear a somewhat distorted audio as opposed to hearing a complete gap for that duration. In case of stereo audio, we can adapt this technique to send the left channel and the right channel in separate packets, so that if the packet for one of the channels is lost, the receiver still hears the other channel.

Cross-References

- ▶ Audio Streaming
- ▶ Interleaved Audio
- ▶ Networking Protocols for Audio Streaming
- ▶ Streaming Audio Player

References

1. ITU-T G.711 Recommendation, "Pulse Code Modulation (PCM) of Voice Frequencies," International Telecommunication Union, Telecommunication Standardization Sector.
2. ITU-T G.722 Recommendation, "7 KHz Audio-coding Within 64 Kbits/s," International Telecommunication Union, Telecommunication Standardization Sector.
3. ITU-T G.723.1 Recommendation, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 Kbit/s," International Telecommunication Union, Telecommunication Standardization Sector.
4. ITU-T G.726 Recommendation, "40, 32, 24, 16 Kbit/s adaptive differential pulse code modulation (ADPCM)," International Telecommunication Union, Telecommunication Standardization Sector.
5. ITU-T G.728 Recommendation, "Coding of speech at 16 Kbit/s using low-delay code excited linear prediction," International Telecommunication Union, Telecommunication Standardization Sector.
6. ITU-T G.729 Recommendation, "Coding of speech at 8 Kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)," International Telecommunication Union, Telecommunication Standardization Sector.
7. ISO/IEC 11172–3 "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio," International Organization for Standardization.
8. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889, January 1996.
9. H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," IETF RFC 2326, April 1998.
10. D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," IETF RFC 2250, January 1998.
11. R. Finlayson, "A More Loss-Tolerant RTP Payload Format for MP3 Audio," IETF RFC 3119, June 2001.

Augmented Reality

Definition

Augmented reality is a system that enhances the real world by superimposing computer-generated information on top of it.

Virtual Reality (VR) is the technology that provides almost real and/or believable experiences in a synthetic or virtual way. Augmented Reality (AR) can be thought of as a variation of VR. In the original publication [1] which coined the term, (Computer-) Augmented Reality was introduced as the opposite of VR: instead of driving the user into a purely-synthesized informational environment, the goal of AR is to augment the real world with synthetic information such as visualizations and audio. In other words, AR is a system that enhances the real world by superimposing computer-generated information on top of it. VR technologies completely

Augmented Reality. **Figure 1.**

immerse a user inside a synthetic environment. While immersed, the user can not see the real world around him/her. In contrast, AR allows the user to see the real world, but superimposes computer-generated information upon or composed with the real world. Therefore, AR supplements reality, rather than completely replacing it. Combining 3D graphics with the real world in a 3D space is useful in that it enhances a user's perception of and interaction with the real world. In addition, the augmented information, such as annotations, speech instructions, images, videos, and 3D models, helps the user perform real world tasks. Figure 1 shows a wearable computer used for the implementation of AR of an industry training application.

Cross-References

► Virtual and Augmented Reality

References

1. P. Wellner, W. Mackay, and R. Gold (Eds.), "Special Issue on Computer Augmented Environments: Back to the Real World," Communications of the ACM, Vol. 36, No. 7, July 1993.

Authoring and Specification

Definition

Authoring and specification tools provide development environment for multimedia applications and presentations.

Numerous multimedia authoring tools and specification techniques have been produced across both commercial and research domains. While many articles focus on particular niche, ranging from specific media (e.g., image editing) to complex multimedia scenarios, this article specifically overviews those that have addressed various aspects of multimedia synchronization.

Early efforts in the specification of multimedia synchronization were based on temporal intervals [1] and a broad array of Petri net based techniques, such as [2]. Language-based constructs started with HyTime [3], a SGML-based (Standardized General Markup Language) document language which offered inter-object hyperlinking, scheduling and synchronization. The evolution of such techniques broadened to support the various classes of media synchronization (content, space and time) and included efforts such as MHEG [4] and PREMO (Presentation Environment for Multimedia Objects) [5]. MHEG offered a platform independent, "final" non-editable specification for real-time multimedia presentation, synchronization and interactivity, while PREMO provided a framework/middleware-approach to facilitate a standardized development environment for multimedia applications. Based on a conceptual framework, the latter's major goal was to provide a standard way to integrate emerging technologies from different media and presentation techniques to graphics packages and networks. Another series of research efforts addressed authoring of interactive multimedia presentations through a trio of projects which focused on the expression, specification and provision of media synchronization (DEMAIS, FLIPS and N-Sync [6]).

Recently, the SMIL (Synchronized Multimedia Integration Language) [7] markup language was developed by the World Wide Web Consortium (W3C) http://www.webopedia.com/TERM/S/W3C.html based on the extensible Markup Language (XML). It does not define any specific media format, but defines how various multimedia components should be played together or in sequence (including position, visibility, scheduling and duration). Complex multimedia scenario content is distributed amongst different servers, sent as independent streams http://www.webopedia.com/TERM/S/streaming.html (e.g., audio, video, text and images) and rendered together as a single unified presentation according to the SMIL specification.

Cross-reference

► Multimedia Synchronization – Area Overview

References

1. J.F. Allen, "Maintaining Knowledge about Temporal Intervals," Communications of the ACM, Vol. 26, 1983, pp. 832–843.
2. M. Diaz and P. Sénac, "Time Stream Petri Nets: A Model for Multimedia Streams Synchronization," Proceedings of International Conference on Multimedia Modeling, Singapore, 1993, pp. 257–273.
3. ISO, "Hypermedia/Time-Based Structure Language: HyTime (ISO 10744)," 1992.
4. T. Meyer-Boudnik and W. Effelsberg, "MHEG Explained," IEEE Multimedia, Vol. 2, 1995, pp. 26–38.
5. I. Herman, N. Correia, D.A. Duce, D.J. Duke, G.J. Reynolds, and J. Van Loo, "A Standard Model for Multimedia Synchronization: PREMO Synchronization Objects," Multimedia Systems, Vol. 6, 1998, pp. 88–101.
6. B.P. Bailey, J.A. Konstan, and J.V. Carlis, "DEMAIS: Designing Multimedia Applications with Interactive Storyboards," Proceedings of the Ninth ACM International Conference on Multimedia, Ottawa, Canada, 2001, pp. 241–250.
7. W3C, http://www.w3.org/AudioVideo, 2005.

Automated Lecture Capturing

► Automated Lecture Recording

Automated Lecture Recording

Gerald Friedland[1], Wolfgang Hürst[2], Lars Knipping[3]

[1]International Computer Science Institute, Berkeley, CA, USA

[2]Utrecht University, Utrecht, The Netherlands

[3]Berlin University of Technology, Berlin, Germany

Synonyms

► Automated lecture capturing; ► Presentation recording

Definition

The term automated lecture recording describes processes that aim to capture instructional classroom or lecture-hall events with a minimal use of manual processing.

Introduction

Providing recorded lectures for download or online viewing over the Internet has become popular at many universities. Educational institutions use such recordings as a means to make classes available to more people. In addition, providing classroom recordings online can improve education for their local students – for example, to cope with conflicting dates, to catch up with presentations they missed due to illness, or by simply offering them the ability to repeat parts of selected lectures when preparing for exams. Recorded courses are usually made available for download through a web browser or for online viewing via streaming. Recent trends also include the distribution of lecture recordings via podcasting [1].

Traditionally, lectures have generally been recorded as videos. Sometimes, the live event is also transmitted over the Internet. Initially, standard audio and video encoding as well as broadcasting applications were used for this process. The main reasons for the use of such tools in early approaches for lecture recording were primarily their commercial availability and the straightforward handling of state-of-the-art Internet broadcasting software. However, such tools were developed to suit TV broadcasts and were normally not designed to record typical classroom presentations. As a consequence, their operation draws too much attention from the presenter or might even demand technical staff for operating them.

If the aim is to do lecture recording on a bigger scale (i.e., recording several if not all classes of a term), using additional personnel to produce the recordings, manually post-process them, and archive them for download over the Internet is an approach which is often too time and cost intensive. As a consequence, by the late 1990s, *automated lecture recording* had become a research field. Automatic lecture recording basically aims to produce recorded lectures as a side-effect of a live session by being minimally invasive in the classroom event, not requiring constant supervision by humans, and automating the post-processing and archiving of the produced files as much as possible.

The respective approaches normally capture the audio of an instructional lecture (i.e., the voice of the teacher), slides presented during the presentation together with timestamps and annotations, as well as a video of the lecturer. Whereas audio and slides are obviously considered to be essential for reviewing a lecture recording (see, for example, [2]), the question of whether the video of the presenter is important for the learning process has become a subject of much discussion in the e-learning community (see, for example, [3]). For example, on the one hand such a video can improve attentiveness and create a better level of trust to the viewer; on the other hand, it might lead to the split attention problem and complicate the layout process. Hence, it is sometimes omitted, especially in situations with limited availability of storage or bandwidth. Traditionally, lecture recording and live broadcasting often went hand in hand, since the availability of tools and devices for one of these tasks generally simplifies the implementation of the other one as well. In this article, we will not elaborate the broadcasting problem but mainly focus on the recording issue.

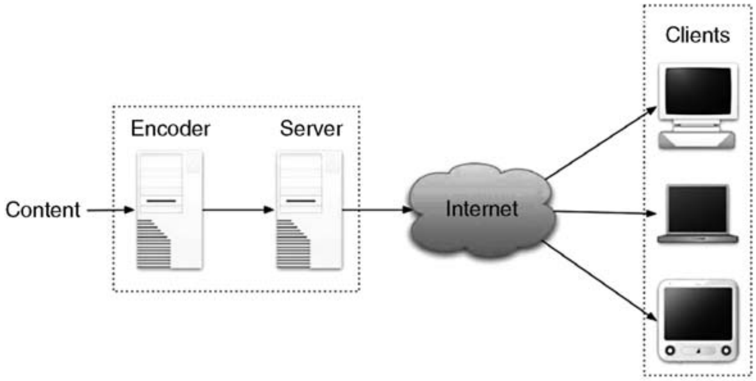
The research done in the last few years has shown that there is no unique solution for the automatic lecture recording problem but that different approaches exist, each of which offers different advantages to the user. In the following, we start by giving a short overview of common technologies normally involved in lecture recording. Then we present three representative systems illustrating different aspects involved in the lecture recording process. We close with a summary identifying issues of current and future work related to automated lecture recording.

Non-Automated Lecture Recording

Most automatic lecture recording projects are built on top of standard commercial Internet broadcasting systems. It is therefore useful to take a more detailed look at the three most important ones, namely the Windows Media Platform by Microsoft, Inc., the products from RealNetworks, Inc., and QuickTime by Apple, Inc.

The main scope of commercial encoders is the transmission of audio and video data via the Internet and their digital archival in files. Each of the systems consists of a three-layer architecture which contains an encoder part, a server part, and a client part. Figure 1 illustrates a typical architecture.

The encoder captures live audio and video content and delegates the compression to a codec provided by the operating system. Pre-recorded content is handled by using the codec as a converter. Current encoders feature flexible encoding modes with constant and variable bit rates. In addition to stream-capturing devices (such as sound cards, video cards, or FireWire interfaces), encoders are also able to read still images and capture screen shots. Most encoders can capture the entire desktop screen, individual windows, or a region and broadcast or encode it to files. Encoders normally provide a user interface to control everything necessary for live event production, such as pre-defining playlists and switching between live and pre-recorded sources. Several encoders also support controlling conventional hardware devices attached to the computer, providing commands like rewind, play, or pause to be sent to digital video cameras and videotape recorders. Generally, time-code data is also captured from the original source for frame-accurate seeking. Several encoders are also able to integrate presentation slides; for example Microsoft Producer also encodes Microsoft PowerPoint presentations.



Automated Lecture Recording. Figure 1. Typical architecture of a commercial multimedia broadcasting server. These systems usually assume that the content is given. In reality, however, the production of content can be resource-consuming, especially in a classroom environment. Image from [4].

The server part is able to deliver either a live stream or pre-encoded content over the Internet. For live streaming, a so-called broadcast publishing point connects to an encoder and sends the stream to compatible clients. A publishing point is a computer with a properly configured web server and a running streaming server. Today, most servers can also stream files that were encoded by encoders of different origin. Servers may also get their input stream from the output of another server. This allows for load balancing when many people connect to a certain broadcast at once.

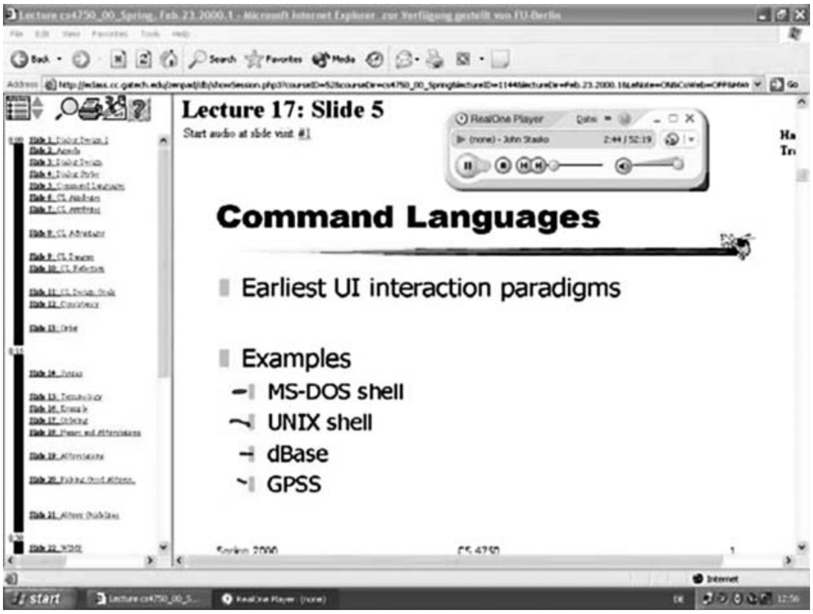
The client program is able to receive a stream sent by a server and to play back files stored on the local hard disk. The Windows Media Player is part of the operating system Windows. Apple's QuickTime Player is part of the operating system Mac OS X. The RealPlayer is part of several operating systems in handheld devices, such as mobile phones. Both the QuickTime Player and the RealPlayer are available on other platforms as well. Usually, the players install themselves into the web browser. They are integrated as plug-ins and are automatically invoked by the browser once a user accesses an associated file.

Classroom 2000/eClass

As stated before, many approaches have been developed in the past few years which highlight different aspects, and therefore, propose different solutions for the lecture recording problem. The *Classroom 2000* project [5,6] is noteworthy because it was one of the earliest approaches which went beyond pure manually operated recording and transmission of live lectures but used computer supported classrooms to generate lecture recordings as a side-effect. It was developed at the Georgia Institute of Technology in 2000, and was renamed to eClass [7]. Figure 2 shows a snapshot of the replay of a Classroom 2000 lecture in a web browser.

Classroom 2000 consists of a prototype classroom environment and a software system with the goal to "capture the rich interaction that occurs in a typical university lecture." The instructor uses an electronic whiteboard system, where the computer screen content is projected either from behind or from the front. A pen-tracking system simulates mouse movements and can be used for handwriting input and to control the applications which are used for the presentation. Audio recording is done using two dynamic microphones attached to the ceiling of the classroom. Video recording is done using a front camera for the instructor, a rear camera for the classroom and a document camera to capture non-electronic documents. The prototype classroom also featured a radio tuner, a VCR, and a DAT player/recorder for the instructor to be able to use non computer-based media during the lecture.

The recording software technically consists of several components that are briefly described in the



Lecture cs4750_00_Spring, Feb 23.2000.1 - Microsoft Internet Explorer zur Verfügung gestellt von FU-Berlin

Lecture 17: Slide 5

Start audio at slide visit #1

Command Languages

- Earliest UI interaction paradigms
- Examples
  - MS-DOS shell
  - UNIX shell
  - dBase
  - GPSS

Spring 2000 CS 4750 1

Automated Lecture Recording. Figure 2. A replay of an eClass lecture with slides and audio. Image from [8].

following. A presentation component called "ZenPad" is used to present pre-specified slides during the lecture. The program also allows for a simple free-hand annotation of slides. For use as whiteboard, the instructor adds an empty slide. The web browser used during the lecture is configured to use a custom proxy server keeping track of every URL visited. A program called "StudPad" allows for student interaction in the classroom. The program distributes the ZenPad content of the presentation computer to any number of student computers. The students can then add private notes. The so-called "ZenStarter" program was used to integrate the different components. The program triggers the "RealProducer" simultaneously with the ZenPad to record optional audio and video of the lecture. After the recording session a program called "StreamWeaver" builds HTML pages including links to the time-stamped slide positions that enable navigation inside the recording. The program converts all presented slides, including the last state of the annotation, to GIF images and creates a list of links from the logs of the custom proxy server.

The created lecture can be replayed remotely using a web browser. The system replays audio, a small video, any presented slides with static handwritten annotations, and all web links visited during the lecture. A detailed evaluation of the usage of the system and a discussion about the lessons learned can be found in [9].

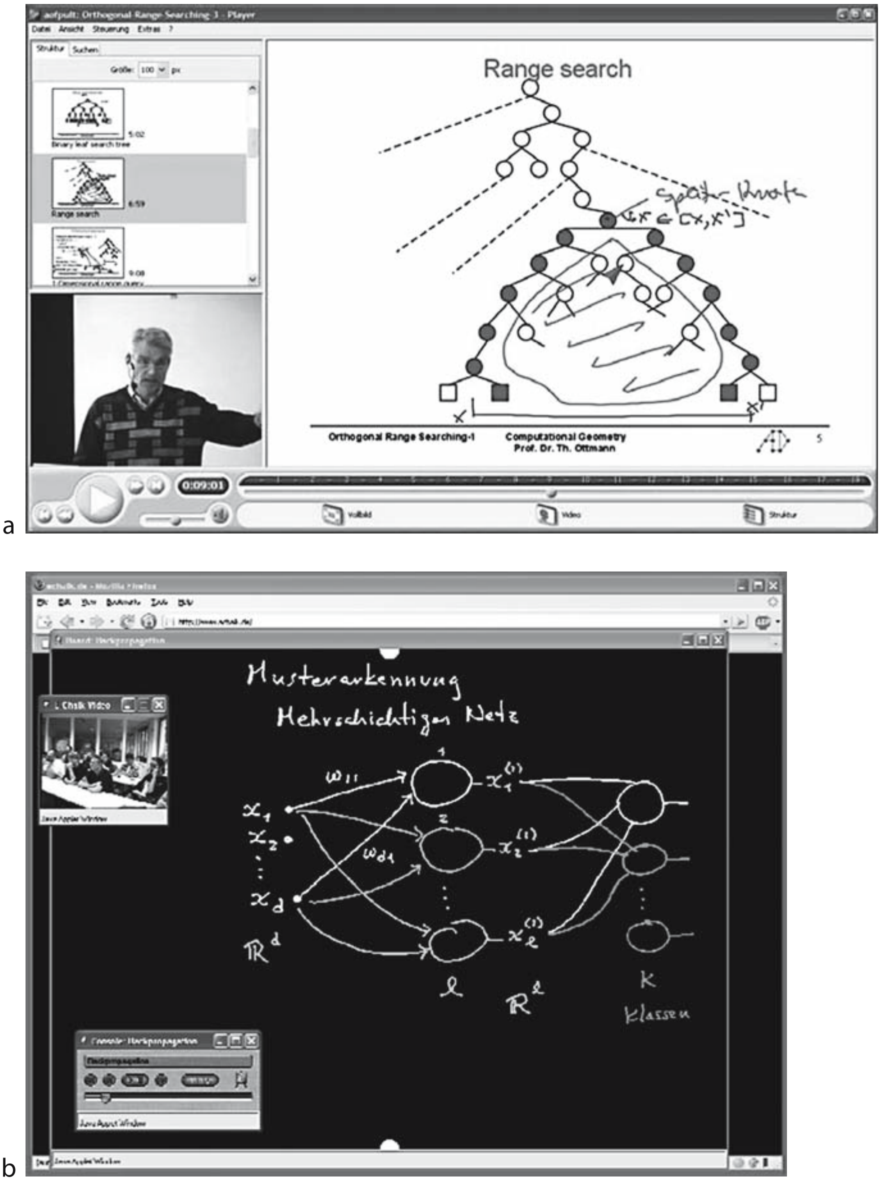
Authoring on the Fly

Authoring on the Fly (AOF) was developed at about the same time as Classroom 2000 at the University of Freiburg, Germany. It is noteworthy for two reasons. First, like E-Chalk (see below), it is one of the few academic approaches which went beyond the status of a research prototype and became a commercial product: *Lecturnity* [10] – which is developed and distributed by the imc AG – is based on research results achieved in the AOF project. Second, although addressing the same basic problem, the project focused on different aspects and thus created different solutions for some problems compared to Classroom 2000. A snapshot of the replay of a lecture recorded with Lecturnity is shown in Fig. 3.

One of the main characteristics of AOF is that it produces a vector-based recording of the slides and annotations presented during a lecture in contrast to common approaches which rely on a bitmap-based recording. Vector-based recording has several advantages including lower data rates and easier post-processing. In addition, having an object-based representation of the data normally simplifies the creation of indexes for lecture recordings (see [11], for example) and the development of advanced navigation functionalities, which increase the usability of such files (see [12], for example). Bitmap-based approaches normally produce a video by recording snapshots of the screen, thus requiring advanced techniques, such as image analysis and optical character recognition to achieve similar results. However, the advantage comes at a price: Vector-based recording can only be done if deeper knowledge of the involved objects is available and accessible. Hence, such a recording is usually restricted to the information presented with certain tools. In the AOF project, recording was mainly limited to slides and related annotations presented with the proprietary whiteboards developed as part of the project – *AOFwb* (see [13]) and *mlb* (mlb was developed in a cooperation with the group of Wolfgang Effelsberg at the University of Mannheim, Germany).

The Lecturnity system still uses a proprietary recording format for vector-based capturing of slides and annotations, but enables users to easily switch to screen capturing if necessary, thus offering a good compromise by combining the advantages of both worlds – using object-based recording whenever possible but falling back on bitmap-based capturing if necessary. A detailed discussion about the advantages and disadvantages of the two approaches can be found in [14].

Another characteristic of the AOF system is the stream-based capturing approach. Different media streams which appear in the classroom are captured and stored separately – an approach which offers several advantages including higher flexibility during replay, for example by adapting the replay quality of singular streams to changing bandwidths. AOF realizes a so-called master-slave synchronization model. The audio – which contains the voice of the lecturer and is generally considered to be the most important information stream in a lecture recording (cf. Introduction) – serves as a master stream. All other streams, such as slides and annotations or a video image of the lecturer, are synchronized to this master stream during replay. Thus, a high quality replay of the audio is guaranteed in contrast to the visual streams where small delays or variations in the frame rate



Automated Lecture Recording. Figure 3. (a) A replay of Lecturnity lecture with slides, audio, video of the lecturer, and thumbnail index for easy navigation. (b) A replay of an eChalk session.

are normally tolerable. Further advantages of such a master-slave synchronization approach for lecture recording and replay as well as a detailed technical description of the implementation realized in AOF can be found in [15].

E-Chalk

Most automated lecture recording systems were designed to capture slide-based presentations, often enriched with annotation capabilities. The E-Chalk system proposes a different approach: In 2000, upon initiation by Raul Rojas, the *E-Chalk* project started with the idea of "creating an update of the traditional chalkboard" [16]. The underlying idea was that the chalkboard has unique didactic advantages over slide presentations that were to be preserved for the future by creating a blackboard-based automatic lecture recording system. Especially in natural sciences, traditional chalk-and-talk lectures have not been replaced by slide presentations. Among the reasons given was the fact that a teacher is slowed down while actively developing the content on the board, making it easier for students to follow the lecture. A chalkboard usually makes it much easier to convey the train of thought that leads to a solution of a problem, rather than overwhelming listeners with bullet-pointed results. For a detailed discussion on the philosophy of E-Chalk see [4].

E-Chalk was designed to work together with an electronic pen input device, like a tablet PC or some kind of pen-active wall display. The system presents the user an initially empty screen, the chalkboard, to draw and write on. In addition to the bare drawing functionality, the system offers a set of additional features that make this teaching system unique. For example, the presenter is able to paste multimedia elements on the screen, such as images from the Web or interactive Java Applets. Queries for textual or graphical results can be submitted to computer algebra systems running in the background (e.g. Wolfram Mathematica or Waterloo Maple) or to Web services [17]. An open SDK is provided for developing so-called Chalklets, custom mini-applications that use exclusively pen strokes as input and output. [18]. Other features include a handwriting recognition for mathematical formulas [19] and integration of Google's Image Search to allow spontaneous use of arbitrary images.

In contrast to most other automated lecture recording systems, E-Chalk does not organize lecture content into pages. Instead, the board has a virtually infinite length. An E-Chalk session can be replayed by a Java-capable browser, cf. Fig. 3.

The presentation software stores a session as a collection of files including an HTML index file embedding appropriate player Applets. Each of the content streams recorded – the animated board image, the audio stream, and an optional video – is handled by a different Applet. The Applets communicate with each other to ensure synchronization. A fourth Applet, a control panel, is provided for VCR-like navigation. The system also offers a live transmission of a session. A PDF file is also generated as a static copy of the board content for printing.

The board stream is represented using a vector format, resulting in rather low bandwidth requirements without quality loss. In practice, the board's bandwidth requirement is negligible compared to the bandwidth used by audio and optional video, particularly since (proprietary) audio stream codecs between 24 and 256 kbps can be chosen [8].

Alternatively, the combined board and audio streams of a recording can be transformed into MPEG-4 video or into a Java Midlet to provide replay on mobile devices, such as PDAs, cell-phones, or Apple's iPod [20].

Summary

Preparing, creating, and post-processing lecture recordings by using traditional approaches for broadcasting and TV production is a time-consuming and expensive task and therefore not practical at most universities. Automated lecture recording offers a solution to this problem. However, research has shown that there is no single, "best" approach. Depending on the situation and context, different procedures can be chosen. Based on the successful research done in the past years, several commercial systems for automated lecture recording exist today. As a result, recording lectures and providing these recordings to students has become an established process at many universities and the number of institutions offering this service seems to increase constantly. Despite the achieved results and the existence of commercial tools, automated lecture recording remains an active research area. Current research aims to automate all stages of the production process. For example, the work done by [21] introduces a virtual cameraman producing higher quality video recordings than existing approaches. Other researchers address the post-processing problem – for example, the automatic production of searchable indexes [11] or semantic annotations [22]. Finally, the ability to replay recorded lectures on mobile devices such as cell phones or mobile media players requires the development of new, adaptive zooming and scaling approaches [20].

References

1. D.J. Malan, "Podcasting Computer Science E-1," ACM SIGCSE Bulletin, Vol. 39, No. 1, March 2007, pp. 389–393.
2. D.J. Gemmell, C.G. Bell, "Noncollaborative Telepresentations Come Of Age," Communications of the ACM, Vol. 40, No. 4, April 1997, 79–89.
3. R. Mertens, G. Friedland, A. Krüger, To See or Not To See: "Layout Constraints, the Split Attention Problem and their Implications for the Design of Web Lecture Interfaces," World Conference on E-Learning, in Corporate, Government, Healthcare & Higher Education (E-Learn 2006), October 2006, Honolulu, HI, USA.
4. G. Friedland, "Adaptive Audio and Video Processing for Electronic Chalkboard Lectures," Doctoral dissertation, Freie Universität Berlin, Department of Computer Science, October 2006.
5. G.D. Abowd, "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment," IBM Systems Journal, Vol. 38, No. 4, pp. 508–530.

6. J.A. Brotherton, "Enriching Everyday Experiences through the Automated Capture and Access of Live Experiences. eClass: Building, Observing and Understanding the Impact of Capture and Access in an Educational Domain," Ph. D. thesis, Georgia Institute of Technology, College of Computing, Atlanta, GA, 2001.
7. eClass project web site, see http://www.cc.gatech.edu/fce/eclass/ (accessed on March 2008).
8. L. Knipping, "An Electronic Chalkboard for Classroom and Distance Teaching," Doctoral Dissertation, Freie Universität Berlin, Department of Computer Science, February 2006.
9. J.A. Brotherton, G.D. Abowd, "Lessons Learned from eClass: Assessing Automated Capture and Access in the Classroom," ACM Transactions on Computer-Human Interaction (TOCHI), Vol. 11. No. 2, June 2004, pp. 121–155.
10. Lecturnity web site, see http://www.lecturnity.de/en/products/lecturnity/ (accessed on March 2008).
11. W. Hürst, N. Deutschmann, "Searching in Recorded Lectures," Proceedings of the World Conference on E-Learning in Corporate Government, Healthcare & Higher Education (E-Learn 2006), AACE, Honolulu, USA, pp. 2859–2852, October 2006.
12. W. Hürst, G. Götz, "Interface Issues for Interactive Navigation and Browsing of Recorded Lectures and Presentations," Proceedings of ED-MEDIA 2004, AACE, Lugarno, Switzerland, June 2004.
13. J. Lienhard, G. Maass, "AOFwb: A New Alternative for the MBone Whiteboard wb," Proceedings of ED-Media '98, Freiburg, Germany, June 1998.
14. W. Hürst, "Automatic Lecture Recording for Lightweight Content Production," in Margherita Pagani (Ed.) "Encyclopedia of Multimedia Technology and Networking" (2nd edn), to appear 2008.
15. W. Hürst, R. Müller, "A Synchronization Model for Recorded Presentations and its Relevance for Information Retrieval," Proceedings of the Seventh ACM international conference on Multimedia (Part 1), pp. 333–342, October/November 1999.
16. R. Rojas, L. Knipping, G. Friedland, B. Frötschl, "Ende der Kreidezeit – Die Zukunft des Mathematikunterrichts," DMV Mitteilungen, pp. 32–37, Berlin, 2001.
17. G. Friedland, L. Knipping, R. Rojas, C. Zick, "Mapping the Classroom into the Web: Case Studies from Several Institutions," Proceedings of the 12th EDEN Annual Conference, Rhodos, June 2003.
18. L. Knipping, M. Liwicki, "Chalklets: Developing Applications for a Board Environment," Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM 2006), pp. 907–914, San Diego, December 2006.
19. G. Friedland, L. Knipping, E. Tapia, "Web Based Lectures Produced by AI Supported Classroom Teaching," International Journal of Artificial Intelligence Tools (IJAIT), special issue of AI Techniques in Web-Based Educational Systems, Vol. 13, No. 2, pp. 367–382, June 2004.
20. A. Lüning, G. Friedland, L. Knipping, R. Rojas, "Visualizing Large-Screen Electronic Chalkboard Content on Handheld Devices," Proceedings of the Ninth IEEE Symposium on Multimedia, Second International Workshop on Multimedia Technologies for E-Learning (MTEL), pp. 369–379, Taichung, Taiwan, December 2007.
21. F. Lampi, S. Kopf, M. Beny, W. Effelsberg, "A Virtual Camera Team for Lecture Recording," to appear in IEEE Multimedia 2008.
22. S. Repp, S. Linckels, Ch. Meinel, Towards to an Automatic Semantic Annotation for Multimedia Learning Objects, Proceedings of Educational Multimedia and Multimedia Education (EMME), pp. 19–26, Augsburg, Germany.

Automatic Generation of Video Documentaries

STEFANO BOCCONI
University of Torino, Torino, Italy

Synonyms

► Automatic video editing; ► Categorical documentaries; ► Rhetorical documentaries

Definition

Using repositories of semantically annotated video material, automatic video generation approaches are able to automatically edit documentaries according to the subject specified by the viewer/user of the system. Documentaries become dynamic documents instead of static artifacts that offer no interaction to users.

Introduction

Traditional documentaries are fixed static artifacts. Viewers do not have any influence on how a documentary looks like. Automatic video generation lets the viewer take the seat of the director and allows the documentarist to provide viewers with documentaries dynamically generated according to their interests. A video generation system can help the documentarist by automatically presenting the material. Different documentaries can then be generated from the same footage, facilitating reuse of the media asset, and allowing new footage to be added at a later stage. Automatic video generation can make a documentary an evolving up-to-date video document rather than a static final product. The price to pay is that the video material needs to be annotated in order for the generation process to use it.

Different Types of Documentary Generation

If we look at how documentaries present information, Bordwell distinguishes three types of form: the *narrative form*, the *categorical form* and the *rhetorical form* ([1] p. 132). The narrative form presents information using stories. In automatic generation, systems mostly use the categorical and the rhetorical forms, and in the following we present those two forms, together with the most representative systems that use them.

Categorical Documentaries

Categorical documentaries organize information in categories. Categories are groupings that individuals or societies create to organize their knowledge of the world. Categories can be strictly defined, as in science (e.g., for plants and animals), or more based on common sense. Categories and subcategories provide a form for a documentarist to use in order to organize the information she wants to convey.

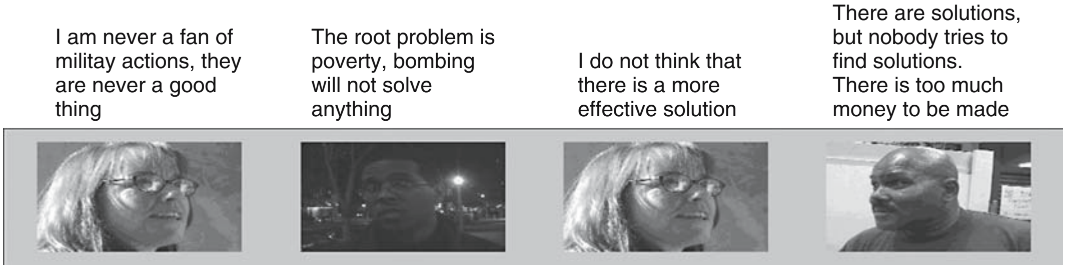
ConTour [2] was developed to support evolving documentaries, i.e., documentaries that could incorporate new media items as soon as they were made. The underlying philosophy was that some stories keep evolving, and so should the documentaries describing them. The system was used to support an evolving documentary about an urban project in Boston. ConTour has a twofold aim: for the author, to provide a framework for gathering content and making it available without having to specify explicitly how (and in what order) the user should view the material; for the user, to support visual navigation of the content. ConTour allows the author to create and expand the repository by adding material to it. The author is required to attach keywords (called descriptors) to each media item. The goal of the descriptors is to capture abstract ideas or elements relevant for the documentary story (i.e., the categories), e.g., names of people or places. Keywords in ConTour relieve authors from the process of defining explicit relationships or links between units of content. Instead, the author connects media items only to keywords. By doing so, the author defines a potential connection between a media item and other media items that share that keyword. Since there are no explicit links between the clips, sequencing decisions are made during viewing, based on the implicit connections via the keywords. Deferring sequencing decisions in this way has as a consequence that the base of content is extensible. Every new media item is simply described by keywords, rather than hardwired to every other relevant media item in the system. In this way, the potentially exponentially-complex task of adding content is managed and requires a constant effort[1].

Rhetorical Documentaries

In using the rhetorical form, a documentary aims at persuading the audience to adopt an opinion about the subject, usually a matter-of-opinion issue. In a rhetorical documentary, the documentarist tries to make her point of view seem the most plausible by presenting different types of arguments and evidence ([1], p. 140).

Terminal Time [3] constructs ideologically-biased historical video documentaries in response to an audience's feedback. Multiple choice questions are posed periodically on the projection screen, and the audience chooses the answers by clapping. The answer generating the loudest clapping wins. After each question, Terminal Time manipulates the presentation of the historical facts in the documentary to mirror and exaggerate the ideological position implied in the audience's answers. Terminal Time is modeled as a rhetorical narrator, who has a rhetorical goal (for example an anti-religious rationalist narrator would have the "show religion is bad" goal) and "spins" the events narrated to support her position. The system follows a top-down approach. It first starts with the rhetorical goal, then it creates a version supporting this goal and finally it presents the version to the public with a generated voice-over narration plus selected historical footage. In Terminal Time, several rhetorical goals are modeled. For example, the goal "show religion is bad" with subgoal "show-thinkers-persecuted-by-religion." A goal is satisfied if one of its subgoals is. A test is associated with each subgoal to determine which events contained in a knowledge-base can be used to make the point. Once a goal is chosen by the audience, Terminal Time runs the associated tests to select the events. Each rhetorical goal also has an associated rhetorical plan which is designed to present events so that they create an argument supporting the goal. Rhetorical plans spin the events by selecting only the details that serve a rhetorical purpose. After the sequencing is done, a Natural Language Generation engine generates the narration. As a last step, video and audio tracks are selected to illustrate the story segment. This search is based on weighted keyword indexing on each video. The clips are shown together with the generated narration.

[1] Other systems that use the same principles as ConTour are the Korsakow System (http://www.korsakow.com/ksy/) and Lev Manovich's Soft Cinema (http://www.softcinema.net/).


I am never a fan of militay actions, they are never a good thing
The root problem is poverty, bombing will not solve anything
I do not think that there is a more effective solution
There are solutions, but nobody tries to find solutions. There is too much money to be made

Automatic Generation of Video Documentaries. Figure 1. A documentary generated with subject = War in Afghanistan, opinion: For, point of view: Attack, and opponents: black males.

Vox Populi [4] is a system capable of generating short documentaries using rhetorical patterns. Its domain is matter-of-opinion documentaries based on interviews. The system was tested on a repository containing video footage shot in the aftermath of 9–11, 8 h of man-on-the street interviews about themes like the war in Afghanistan, the role of the media, anthrax, etc.

Videos in the repository need to be semantically annotated. The annotations encode in a formal manner the verbal message contained in the video and are used to build arguments in favor or against a particular user-chosen subject. Annotations are composed by:

1. Statements, i.e., short sentences that capture the sense of what the speaker says, such as "War is not effective," or "Diplomacy cannot be used."
2. A thesaurus containing the terms to be used to compose the statements, plus the relations between them, for example "war" *opposite* "diplomacy" or "military actions" *similar* "bombing."
3. An argumentation model that assigns a role to each statement used by the interviewee when building an argument, such as the *claim* or the *data* supporting the claim.

Using this information, the system can dynamically compose arguments that support or attack a particular position expressed in an interview. The system requests the user to specify how the documentary should start, by choosing the subject of an interview and the opinion expressed in it, for example "War in Afghanistan – For" (see Fig. 1). It then asks the point of view the documentary should express, which can be to support or attack the chosen interview. By choosing the statements of other interviewees, the arguments in the initial interview can be made to look weak or strong, using also factors as ethos (the social status) and pathos (the emotional impact) of the chosen interviewees.

References

1. D. Bordwell and K. Thompson, "Film Art: An Introduction (7th edn)," McGraw-Hill, New York, 2003.
2. M. Murtaugh, "The Automatist Storytelling System," PhD thesis, Massachusetts Institute of Technology, 1996.
3. M. Mateas, "Generation of Ideologically-Biased Historical Documentaries," Proceedings of AAAI'2000, pp. 36–42, July 2000.
4. S. Bocconi, et al. "Automatic Generation of Matter-of-Opinion Video Documentaries," Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 6, No. 2, 2008, pp. 139–150.

Automatic Video Editing

► Automatic Generation of Video Documentaries

Availability and Scalability of LOMSS

Synonyms

► Two-level meta data management

Definition

Availability in large-scale object-based multimedia storage systems is sued to describe the systems to ensure that multimedia data continues to be available at a required level performance.

In the Large-Scale Object-Based Multimedia Storage System (LOMSS), *availability* is used to describe the systems to ensure that multimedia data continues to be available at a required level performance in situations ranging from normal through disastrous. Availability is a critical issue in the design of LOMSS, since low availability multimedia storage systems will result in unstable services in the Internet and thus, lead to losing customers from service providers. In LOMSS, we shall consider two levels of availabilities: system level and device level. The availabilities of Client, MDS and OSD belong to system level; whereas the availabilities of disks within an OSD belong to device level. In system level, the availability of MDS is critical to the entire system since that if MDS is out of service due to the failures of hardware, software, or networks, etc., the entire system cannot provide continuous data retrieval even though the data in the OSDs are intact [1]. In the literature, two-level metadata management has been proposed in order to achieve high availability of metadata, which consists of global and local metadata managements handled by MDS and OSD, respectively [1]. For OSDs, in an excellent design of scheduling and replication strategy, failure of one or more OSDs will allow the Clients to retrieve data continuously without any interruption so long as there is at least one OSD to cater. The availability of Clients, or in another word, multimedia servers, is inherently considered by multimedia applications, and interested readers can refer to [2] for more details. Further, in device level, LOMSS can adopt RAID to improve the availability of disks in a single OSD, e.g., RAID1 can allow one disk failure and RAID6 can still work well even if two disks fail at the same time.

Scalable storage systems allow for the addition or removal of storage devices to increase storage capacity and bandwidth or to retire older devices. Multimedia applications typically require ever-increasing storage capacity to meet the demands of expanding multimedia files, and traditional storage systems may not reserve a great amount of excess space for future growth. OSDs can be added to LOMSS to increase overall disk capacity or removed due to space conservation or storage reallocation. OSD removals are different from OSD failures since objects can be first moved off an OSD before removal, whereas if an OSD fails, the data are lost. Further, LOMSS shall be a highly scalable storage system that can balance the aggregate storage capacity and bandwidth for concurrently accessing large data objects even after more OSDs are added into the system. Moreover, the redistributed objects shall be retrieved efficiently during the normal mode of operation: in one I/O request and with low complexity computation [3]. In the literature, many redistribution methods have been proposed to deal with the scalability issue, such as *Psudorandom Placement, Linear Hashing (LH), Highest Random Weight (HRW), Distributed Hash Tables (DHTs), Random Disk Labeling (RDL), etc.* However, these algorithms focus on either file-by-file approaches or disk-to-disk approaches. In the domain of LOMSS, object-by-object approaches shall be investigated and implemented in the real-life situations.

Cross-References

► Design of OSD: Issues and Challenges
► Large – Scale Object-Based Multimedia Storage Systems

References

1. F. Wang, Y.L. Yue, D. Feng, J. Wang, and P. Xia, 'High Availability Storage System Based on Two-Level Metadata Management,' Frontier of Computer Science and Technology (FCST 2007), pp. 41–48, November 2007.
2. L.G. Dong, V. Bharadwaj, and C.C. Ko, 'Efficient Movie Retrieval Strategies for Movie-On-Demand Multimedia Services on Distributed Networks,' Journal of Multimedia Tools and Applications, Vol. 20, No. 2, 2003, pp. 99–133.
3. S.Y.D. Yao, C. Shahabi, and P.A. Larson, 'Hash-Based Labelling Techniques for Storage Scaling,' The VLDB Journal, Vol. 14, 2005, pp. 222–237.