# Image and Video Coding—Emerging Standards and Beyond

Barry G. Haskell, *Fellow, IEEE*, Paul G. Howard, Yann A. LeCun, *Member, IEEE*, Atul Puri, *Member, IEEE*, Jöern Ostermann, *Member, IEEE*, M. Reha Civanlar, *Member, IEEE*, Lawrence Rabiner, *Fellow, IEEE*, Leon Bottou, and Patrick Haffner

***Abstract*— In this paper, we make a short foray through coding standards for still images and motion video. We first briefly discuss standards already in use, including: Group 3 and Group 4 for bilevel fax images; JPEG for still color images; and H.261, H.263, MPEG-1, and MPEG-2 for motion video. We then cover newly emerging standards such as JBIG1 and JBIG2 for bilevel fax images, JPEG-2000 for still color images, and H.263+ and MPEG-4 for motion video. Finally, we describe some directions currently beyond the standards such as hybrid coding of graphics/photo images, MPEG-7 for multimedia metadata, and possible new technologies.**

***Index Terms*—Image, standards, video.**

## I. Introduction

IT is by now well agreed that among the necessary ingredients for the widespread deployment of image and video communication services are standards for the coding, compression, representation, and transport of the visual information. Without standards, real-time services suffer because encoders and decoders may not be able to communicate with each other. Nonreal-time services using stored bit streams may also be disadvantaged because of either service providers' unwillingness to encode their content in a variety of formats to match customer capabilities, or the reluctance of customers themselves to install a large number of decoder types to be able to handle a plethora of data formats.

As new technologies offer ever greater functionality and performance, the need for standards to reduce the enormous number of possible permutations and combinations becomes increasingly important. In this paper, we summarize recent activities and possible future needs in establishing image and video coding standards. Some of the material was exerpted (with permission) from a larger companion paper [1], which is a broad tutorial on multimedia.

## II. Compression and Coding of Image[1] Signals

Image coding generally involves compressing and coding a wide range of still images, including so-called bilevel or fax images, photographs (continuous tone color or monochrome images), and document images containing text, handwriting, graphics, and photographs. In order to appreciate the need for compression and coding, Table I shows the uncompressed size needed for bilevel (fax) and color still images.

TABLE I
Characteristics and Uncompressed Bit Rates of Image Signals

| Image Type | Pixels per Frame | Bits/Pixel | Uncompressed Size |
|---|---|---|---|
| FAX (200 dpi) | 1700 × 2200 | 1 | 3.74 Mb |
| VGA | 640 × 480 | 8 | 2.46 Mb |
| XVGA | 1024 × 768 | 24 | 18.87 Mb |

Unlike speech signals, which can take advantage of a well-understood and highly accurate physiological model of signal production, image signals have no such model to rely on. As such, in order to compress and code image signals [2], it is essential to take advantage of any observable redundancy in the signal. The two most important forms of signal redundancy in image signals are *statistical redundancy* and *subjective redundancy*, also known as irrelevance.

Statistical redundancy takes a variety of different forms in an image, including correlations in the background (e.g., a repeated pattern in a background wallpaper of a scene), correlations across an image (e.g., repeated occurrences of base shapes, colors, patterns, etc.), and spatial correlations that occur between nearby pixels.

Subjective redundancy takes advantage of the human visual system that is used to view the decompressed and decoded images. Through various psychophysical testing experiments, we have learned a great deal about the perception of images, and have learned several ways to exploit the human's inability to see various types of image distortion as a function of image intensity, texture, edges, etc.

### A. Coding of Bilevel (Fax) Images [3]

The concepts behind fax coding of bilevel images have been well understood for more than 100 years. However, until a set of standards was created and became well established, fax machines were primarily an office curiosity that were restricted to a few environments that could afford the costs of proprietary methods that could only communicate with like proprietary machines. Eventually, the industry realized that standards-based fax machines were the only way in which widespread acceptance and use of fax would occur, and a set

Manuscript received July 30, 1998. This paper was recommended by Associate Editor T. Sikora.

The authors are with the Speech and Image Processing Services Research Laboratory, AT&T Laboratories, Red Bank, NJ 07701 USA.

Publisher Item Identifier S 1051-8215(98)08400-6.

[1] We, along with others, use the term *image* for still pictures and *video* for motion pictures.

of analog (Group 1 and Group 2) and digital standards (Group 3 and Group 4) were created and widely used. In this section, we briefly consider the characteristics of digital fax [4], along with the more recent JBIG1 standard and the newly proposed JBIG2 standard.

To appreciate the need for compression of fax documents, consider the uncompressed bit rate of a scanned page (8.5 by 11 in) at both 100 and 200 dots/in. At 100 dots/in, the single page requires 935 000 bits for transmission, and at 200 dots/in, the single page requires 3 740 000 bits for transmission. Since most of the information on the scanned page is highly correlated across scan lines (as well as between scan lines), and since the scanning process generally proceeds sequentially from top to bottom (a line at a time), the digital fax coding standards process the document image line by line (or pairs of lines at a time) in a left to right fashion. For MH[2] fax coding, the algorithm emphasizes speed and simplicity, namely, performing a one-dimensional run length coding of the 1728 pixels on each line, with the expedient of providing clever codes for EOL (end-of-line), EOP (end-of-page), and for regular synchronization between the encoder and decoder. The resulting MH algorithm provides, on average, a 20-to-1 compression on simple text documents.

The MREAD[3] fax algorithm provides an improvement over MH fax by using a two-dimensional coding scheme to take advantage of vertical spatial redundancy as well as the horizontal spatial redundancy. In particular, the MREAD algorithm uses the previous scan line as a reference when coding the current scan line. When the vertical correlation falls below a threshold, MREAD encoding becomes identical to MH encoding. Otherwise, the MREAD encoding codes the scan line in either a vertical mode (coding based on the previous scan line), a horizontal mode (locally along the scan line), or a pass mode (which essentially defers the encoding decision until more of the scan line is examined). The simple expedient of allowing the encoding to be based on the previous scan line increases the average compression that is obtained on simple text documents to 25-to-1, a 25% improvement over MH encoding.

MREAD fax coding has proven adequate for text-based documents, but does not provide good compression or quality for documents with handwritten text or continuous tone images. As a consequence, a new set of fax standards was created in the late 1980's, including the JBIG1 (Joint Bilevel Image Experts Group) standard [5], and work began on the more recent JBIG2 standard. The key idea here is that for binary halftone images (i.e., continuous-tone images that are converted to dot patterns, as in newspapers), neither MR nor MREAD fax coding is adequate since each image pixel needs a significantly larger region of support for prediction than that needed for text images. The JBIG1 standard provides this larger region and, moreover, uses an arithmetic coder that dynamically adapts to the statistics for each pixel context. There are two prediction modes that can be used for encoding. The first is a sequential mode in which the pixel to be coded is predicted based on nine adjacent and previously coded pixels plus one adaptive pixel that can be spatially separated from the others, as shown in Fig. 1. Since each previously encoded pixel is a single bit, the previous pixels used in the sequential coding mode form a 10-bit context index that is used to arithmetically encode the current pixel.

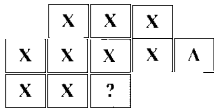


Fig. 1. JBIG1 sequential template where the "?" marks the pixel to be coded based on the previously coded pixels marked $X$, plus an adaptively located pixel marked $A$, which for each page can be moved to a different location.

The second coding mode of JBIG1 is a progressive mode that provides for successive resolution increases in successive encodings. This could be useful in browsing applications where a low-resolution image can be received fairly quickly, with higher resolution arriving later if the user wishes to wait. However, so far, it has not been widely used.

The key behind JBIG1 coding is that binary halftone images have statistical properties that are very different from binary text, and therefore need a significantly different coding algorithm to provide high-quality encoding at significant compression rates. The JBIG1 standard provides compression rates that are slightly better than MREAD fax coding for text sequences, and an improvement in compression by a factor of up to 8-to-1 for binary halftone images.

Although JBIG1 compression works quite well, it has become clear over the past few years that there exists a need to provide optimal compression capabilities for both lossless and lossy compression of arbitrary scanned images (containing both text and halftone images) with scanning rates of from 100 to 800 dots/in. This need was the basis for the JBIG2 method, which is being proposed as a standard for bilevel document coding. The key to the JBIG2 compression method is *soft pattern matching* [6], [7], which is a method for making use of the information in previously encountered characters without risking the introduction of character substitution errors that is inherent in the use of optical character recognition (OCR) methods [8].

The basic ideas of the JBIG2 standard are as follows [9].

- The basic image is first segmented into individual marks (connected components of black pixels).
- The resulting set of marks is partitioned into equivalence classes, with each class ideally containing all occurrences of a single letter, digit, or punctuation symbol.
- The image is then coded by coding a representative *token* mark from each class, the position of each mark (relative to the position of the previous mark), the index of the matching class, and finally the resulting error signal between each mark and its class token.
- The classes and the representative tokens are adaptively updated as the marks in the image are determined and coded.
- Each class token is compressed using a statistically based, arithmetic coding model that can code classes independently of each other

[2] Modified Huffman.

[3] Modified relative address. A modified MREAD algorithm (MMR) is also sometimes used. MMR increases compression somewhat, but at the expense of less resiliency to errors.

The key novelty with JBIG2 coding is the solution to the problem of substitution errors in which an imperfectly scanned symbol (due to noise, irregularities in scanning, etc.) is improperly matched and treated as a totally different symbol. Typical examples of this type occur frequently in OCR representations of scanned documents where symbols like "o" are often represented as "c" when a complete loop is not obtained in the scanned document, or a "t" is changed to an "l" when the upper cross in the "t" is not detected properly. By coding the bitmap of each mark, rather than simply sending the matched class index, the JBIG2 method is robust to small errors in the matching of the marks to class tokens. Furthermore, in the case when a good match is not found for the current mark, that mark becomes a token for a new class. This new token is then coded using JBIG1 with a fixed template of previous pixels around the current mark. By doing a small amount of preprocessing, such as elimination of very small marks that represent noise introduced in the scanning process, or smoothing of marks before compression, the JBIG2 method can be made highly robust to small distortions of the scanning process used to create the bilevel input image.

The JBIG2 method has proven itself to be about 20% more efficient that the JBIG1 standard for lossless compression of bilevel images. By running the algorithm in a controlled lossy mode (by preprocessing and decreasing the threshold for an acceptable match to an existing mark), the JBIG2 method provides compression ratios about two–four times that of the JBIG1 method for a wide range of documents with various combinations of text and continuous-tone images with imperceptible loss in image quality.

### B. Coding of Continuous Images—JPEG Methods [10]

In this section, we discuss standards that have been created for compressing and coding continuous-tone still images—both gray scale (monochrome) and color images, of any size and any sampling rate. We assume that the uncompressed images are available in a digital format, with a known pixel count in each dimension (e.g., the rates shown in Table I), and an assumed quantization of 8 bits/pixel for gray-scale images and 24 bits/pixel for color images.

The most widely used standard algorithm for compression of still images is called the JPEG (Joint Photographic Experts Group) algorithm [11], and it has the properties that it is of reasonably low computational complexity, is capable of producing compressed images of high quality, and can provide both lossless and lossy compression of arbitrary sized gray-scale and color images [12]. A block diagram of the JPEG encoder and decoder is shown in Fig. 2. The image to be compressed is first converted into a series of 8 (pixel) × 8 (pixel) blocks which are then processed in a raster scan sequence, from left to right, and from top to bottom. Each such 8 × 8 block of pixels is first spectrally analyzed using a forward discrete cosine transform (FDCT) algorithm, and the resulting DCT coefficients are scalar quantized based on a psychophysically based table of quantization levels. Separate quantization tables are used for the luminance component

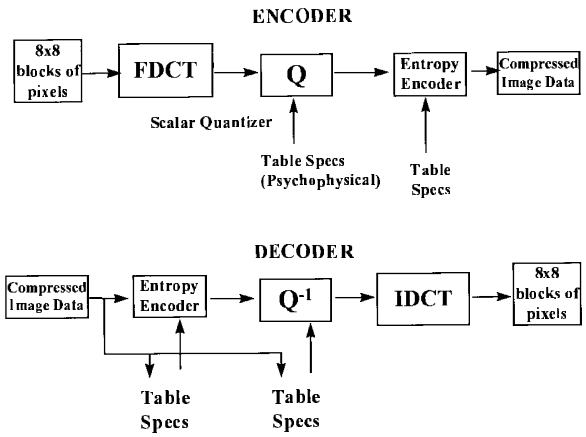


Fig. 2. Block diagram of JPEG encoder and decoder.

(the image intensity) and the chrominance component (the color). The entries in the quantization tables are based on eye-masking experiments, and are essentially an approximation to the best estimate of levels which provide just-noticeable distortion in the image. The 8 × 8 blocks are then processed in a zig-zag order following quantization. An entropy encoder performs run-length coding on the resulting DCT sequences of coefficients (based on a Huffman coder), with the dc coefficients being represented in terms of their difference between adjacent blocks. Finally, the compressed image data are transmitted over the channel to the image receiver. The decoder performs the inverse operations of the encoder.

*1) Progressive Encoding:* Progressive encoding modes are provided within the JPEG syntax in order to provide a layering or progressive encoding capability to be applied to an image, i.e., to provide for an image to be transmitted at a low rate (and a low quality) and then progressively improved by subsequent transmissions. This capability is convenient for browsing applications where a low-quality or low-resolution image is more than adequate for things like scanning through the pages of a catalog.

Progressive encoding depends on being able to store the quantized DCT coefficients for an entire image. There are two forms of progressive encoding for JPEG, namely, spectral selection and successive approximation. For spectral selection, the initial transmission sends low-frequency DCT coefficients, followed progressively by the higher frequency coefficients, according to the zig-zag scan used to order the DCT coefficients. Thus, the first transmission might send the lowest three DCT coefficients for all of the 8 × 8 blocks of the image, followed by the next higher three DCT coefficients, and so forth until all of the DCT coefficients have been transmitted. The resulting scheme is simple to implement, but each image lacks the high-frequency components until the end layers are transmitted; hence, the reconstructed images from the early scans are blurred.

With the successive approximation method,[4] all of the DCT coefficients for each 8 × 8 block are sent in each scan. However, instead of sending them at full resolution, only the

[4]Sometimes called SNR scalability.

most significant bits of each coefficient are sent in the first scan, followed by the next most significant bits, and so on until all of the bits are sent. The resulting reconstructed images are of reasonably good quality, even for the very early scans, since the high-frequency components of the image are preserved in all scans.

The JPEG algorithm also supports a hierarchical or pyramid mode in which the image can be sent in one of several resolution modes to accommodate different types of displays. The way this is achieved is by filtering and downsampling the image, in multiples of two in each dimension. The resulting decoded image is upsampled and subtracted from the next level, which is then coded and transmitted as the next layer. This process is repeated until all layers have been coded and transmitted.

The lossless mode of JPEG coding is different from the lossy mode shown in Fig. 2. Fundamentally, the image pixels are handled separately (i.e., the $8 \times 8$ block structure is not used), and each pixel is predicted based on three adjacent pixels using one of eight possible predictor modes. An entropy encoder is then used to losslessly encode the predicted pixels.

*2) JPEG Performance:* If we assume that the pixels of an arbitrary color image are digitized to 8 bits for luminance, and 16 bits for chrominance (where the chrominance signals are sampled at one-half the rate[5] of the luminance signal), then effectively there are 16 bits/pixel that are used to represent an arbitrary color image. Using JPEG compression on a wide variety of such color images, the following image qualities have been measured subjectively:

| Bits/Pixel | Quality | Compression Ratio |
|---|---|---|
| $\geq 2$ | Indistinguishable | 8-to-1 |
| 1.5 | Excellent | 10.7-to-1 |
| 0.75 | Very Good | 21.4-to-1 |
| 0.50 | Good | 32-to-1 |
| 0.25 | Fair | 64-to-1 |

The bottom line is that, for many images, good quality can be obtained with about 0.5 bits/pixel with JPEG providing a 32-to-1 compression.

*3) JPEG-2000 Color Still Image Coding [13]:* Much research has been undertaken on still image coding since the JPEG standards were established in the early 1990's. JPEG-2000 is an attempt to focus these research efforts into a new standard for coding still color images.

The scope of JPEG-2000 includes not only potential new compression algorithms, but also flexible compression architectures and formats. It is anticipated that an architecturally based standard has the potential of allowing the JPEG-2000 standard to integrate new algorithm components through downloaded software without requiring yet another new standards definition.

[5]The so-called 4:2:2 color sampling.

Some examples of the application areas for JPEG-2000 include the following.

| | |
|---|---|
| Document imaging | Medical imagery |
| Facsimile | Security cameras |
| Internet/WWW imagery | Client–server |
| Remote sensing | Scanner/digital copiers |
| Video component frames | Prepress |
| Photo and art digital libraries | Electronic photography |

JPEG-2000 is intended to provide low bit-rate operation with subjective image quality performance superior to existing standards, without sacrificing performance at higher bit rates. It should be completed by the year 2000, and offer state-of-the-art compression for many years beyond.

JPEG-2000 will serve still image compression needs that are currently not served. It will also provide access to markets that currently do not consider compression as useful for their applications. Specifically, it will address areas where current standards fail to produce the best quality or performance including the following.

- *Low bit-rate compression performance:* Current JPEG offers excellent compression performance in the mid- and high bit rates above about 0.5 bits/pixel. However, at low bit rates (e.g., below 0.25 bits/pixel for highly detailed images), the distortion, especially when judged subjectively, becomes unacceptable compared with more modern algorithms such as wavelet subband coding [14].
- *Large images:* Currently, the JPEG image compression algorithm does not allow for images greater then 64K $\times$ 64K without tiling (i.e., processing the image in sections).
- *Continuous-tone and bilevel compression:* JPEG-2000 should be capable of compressing images containing both continuous-tone and bilevel images. It should also compress and decompress images with various dynamic ranges (e.g., 1–16 bits) for each color component. Applications using these features include: compound documents with images and text, medical images with annotation overlays, graphic and computer-generated images with binary and near to binary regions, alpha and transparency planes, and of course bilevel facsimile.
- *Lossless and lossy compression:* It is desired to provide lossless compression naturally in the course of progressive decoding (difference image encoding, or any other technique, which allows for the lossless reconstruction is valid). Applications that can use this feature include medical images, image archival applications where the highest quality is vital for preservation but not necessary for display, network applications that supply devices with different capabilities and resources, and prepress imagery.
- *Progressive transmission by pixel accuracy and resolution:* Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution as more bits are received is essential for many applications. This feature allows the reconstruction of images with different resolutions and pixel accuracy, as needed or desired, for different target devices. Examples of applications include the World Wide Web, image archival applications, printers, etc.

- *Robustness to bit errors:* JPEG-2000 must be robust to bit errors. One application where this is important is wireless communication channels. Some portions of the bit stream may be more important than others in determining decoded image quality. Proper design of the bit stream can aid subsequent error-correction systems in alleviating catastrophic decoding failures. Usage of error confinement, error concealment, restart capabilities, or source-channel coding schemes can help minimize the effects of bit errors.
- *Open architecture:* It is desirable to allow open architecture to optimize the system for different image types and applications. This may be done either by the development of a highly flexible coding tool or adoption of a syntactic description language which should allow the dissemination and integration of new compression tools. Work being done in MPEG-4 (see Section V) on the development of downloadable software capability may be of use. With this capability, the user can select tools appropriate to the application and provide for future growth. With this feature, the decoder is only required to implement the core tool set plus a parser that understands and executes downloadable software in the bit stream. If necessary, unknown tools are requested by the decoder and sent from the source.
- *Sequential one-pass decoding capability (real-time coding):* JPEG-2000 should be capable of compressing and decompressing images with a single sequential pass. It should also be capable of processing an image using either component interleaved order or noninterleaved order. However, there is no requirement of optimal compression performance during sequential one-pass operation.
- *Content-based description:* Finding an image in a large database of images is an important problem in image processing. This could have major application in medicine, law enforcement, environment, and for image archival applications. A content-based description of images might be available as a part of the compression system. JPEG-2000 should strive to provide the opportunity for solutions to this problem. (However, see Section VI on MPEG-7.)
- *Image security:* Protection of the property rights of a digital image can be achieved by means of watermarking, labeling, stamping, encryption, etc. Watermarking is an invisible mark inside the image content. Labeling is already implemented in SPIFF,[6] and must be easy to transfer back and forth to JPEG-2000 image files. Stamping is a very visible and annoying mark overlayed onto a displayed image that can only be removed by a specific process. Encryption can be applied on the whole image file or limited to part of it (header, directory, image data) in order to avoid unauthorized use of the image.
- *Side channel spatial information (transparency):* Side channel spatial information, such as alpha planes and transparency planes, are useful for transmitting information for processing the image for display, print, or editing, etc. An example of this is the transparency plane used in World Wide Web applications.

For JPEG-2000, a prime candidate for the base signal processing is wavelet subband coding [15]. Compared with the discrete cosine transform (DCT) as used in JPEG coding, wavelet coding is able to achieve the advantages of low bit-rate coding with large block size, while at the same time providing progressive transmission and scalability features. However, the low-pass wavelet filter may not be optimum in terms of picture quality versus bandwidth. Thus, another candidate might be MPEG intra coding with the pyramid style progressive transmission found in JPEG. With pyramid coding, the filtering can be optimized since it is independent of the coding.

### C. Hybrid Coding of Bilevel/Continuous Document Images—DJVU

In this section, we describe the DJVU [16] format (pronounced "Déjà vu") for compressing high-quality document images in color. Traditional color image compression standards such as JPEG are inappropriate for document images. JPEG's usage of local cosine transforms relies on the assumption that the high spatial frequency components in images can be essentially removed (or heavily quantized) without too much quality degradation. While this assumption holds for most pictures of natural scenes, it does not hold for document images. The sharp edges of character images require a special coding technique so as to maximize readability.

It is clear that different elements in the color image of a typical page have different perceptual characteristics. First, the text with sharp edges is usually highly distinct from the background. The text must be rendered at high resolution, 300 dpi in bilevel or 100 dpi in color, if reading the page is to be a pleasant experience. The second type of element in a document image is natural pictures. Rendering natural pictures at 50–100 dpi is typically sufficient for acceptable quality. The third element is the background color and paper texture. Background colors can usually be presented with resolutions less than 25 dpi for adequate quality.

The main idea of DJVU is to decompose the document image into three constituent images [17] from which the original document image can be reconstructed. The constituent images are: the background image, the foreground image, and the mask image. The first two are low-resolution (100 and 25 dpi, respectively) color images, and the latter is a high-resolution bilevel image (300 dpi). A pixel in the decoded document image is constructed as follows.

> If the corresponding pixel in the mask image is 0, the output pixel takes the value of the background image.
>
> If the mask pixel is 1, the pixel color is taken from the foreground image. The foreground and background images can be encoded with any suitable means, such as JPEG. The mask image can be encoded using JBIG2.

Consider the color histogram of a bicolor document image such as the old text image in Fig. 3. Both the foreground and background colors are represented by peaks in this histogram. There may also be a small ridge between the peaks representing the intermediate colors of the pixels located

[6] Still Picture Interchange File Format, formally known as ITU-T Recommendation T.84|ISO/IEC 10918-4.


Fig. 3. Document images with file sizes after compression in the DJVU format at 300 dots/in.

near the character boundaries. Extracting uniform foreground and background colors is achieved by running a clustering algorithm on the colors of all of the pixels.

*1) Multiscale Block Bicolor Clustering:* Typical document images are seldom limited to two colors. The document design and the lighting conditions induce changes in both the background and foreground colors over the image regions.

An obvious extension of bicolor image clustering consists of dividing the document image using a regular grid to delimit small rectangular blocks of pixels. Running the bicolor clustering algorithm within each block produces a pair of colors for each block. We can therefore build two low-resolution images whose pixels correspond to the cells of the grid. The pixels of the first image (or the second image) are painted with the foreground (or background) color of the corresponding block.

This block bicolor clustering algorithm is affected by several factors involving the choice of a block size, and the selection of which peak of each block color histogram represents the background (or foreground) color. Blocks should be small enough to capture the foreground color change, for instance, of a red word in a line of otherwise black text. However, such a small block size increases the number of blocks located outside the text area. Such blocks contain only background pixels. Blocks may also be entirely located inside the ink of a big character. Such blocks contain only foreground pixels. In both cases, the clustering algorithm fails to determine a pair of well-contrasted foreground and background colors.

Therefore, instead of considering a single block size, we consider now several grids of increasing resolution. Each successive grid employs blocks whose size is a fraction of the size of the blocks in the previous grid. By applying the bicolor clustering algorithm on the blocks of the first grid (the grid with the largest block size), we obtain a foreground and background color for each block in this grid. The blocks of the next smaller grid are then processed with a slightly modified color clustering algorithm.

This modification biases the clustering algorithm toward choosing foreground and background colors for the small blocks that are close to the foreground and background colors found for the larger block at the same location.

This operation alleviates the small block size problem discussed above. If the current block contains only background pixels, for instance, the foreground and background colors of the larger block will play a significant role. The resulting background color will be the average color of the pixels of the block. The resulting foreground color will be the foreground color of the larger block. If, however, the current block contains pixels representing two nicely contrasted colors, the colors identified for the larger block will have a negligible impact on the resulting clusters.

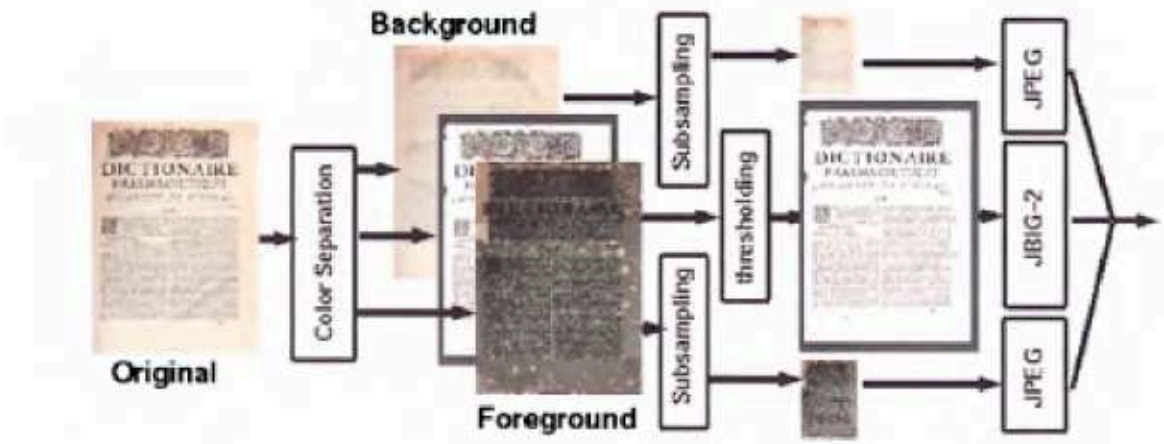


Fig. 4. DJVU compression algorithm first runs the foreground/background separation. Both the foreground and background images are compressed using a sparse wavelet encoding dubbed IW44, while the binarized text and drawings are compressed using AT&T's proposal to the JBIG2 standards committee.

*2) Implementation:* A fast version of this multiscale foreground–background color identification, as shown in Fig. 4, has been implemented and tested on a variety of high-resolution color images scanned at 24 bits/pixel and 300 pixels/in. The sequence of grids of decreasing block sizes is built by first constructing the smallest grid using a block size of 12 × 12 pixels. This block size generates foreground and background images at 100 and 25 pixels/in, respectively. Successive grids with increasing block size are built by multiplying the block width and height by four until either the block width or height exceeds the page size.

This coding implementation runs in about 20 s on a 175 MHz Mips R10000 CPU for a 3200 × 2200 pixel image representing a typical magazine page scanned at 300 pixels/in. Decoding time is a few seconds, most of which is taken up by I/O. When viewing less than the full document, decoding is usually fast enough to keep up with scrolling.

Once the foreground and background colors have been identified with the above algorithm, a gray-level image with the same resolution as the original is computed as follows. The gray level assigned to a pixel is computed from the position of the projection of the original pixel color onto a line in $RGB$ space that joins the corresponding foreground and background colors. If the projection is near the foreground, the pixel is assigned to black, if it is near the background, the pixel is white. This gray-level image will contain the text and all of the areas with sharp local contrast, such as line art and drawings. This gray-level image is transformed into a bilevel mask image by an appropriate thresholding.

The document is now represented by three elements. The first two elements are the 25 dpi color images that represent the foreground and background color for each 12 × 12 pixel block. Those images contain a large number of neighboring pixels with almost identical colors. The third element is the 300 dpi bilevel mask image whose pixels indicate if the corresponding pixel in the document image should take the foreground or the background color. This mask image acts as a "switch" or stencil for the other two images.

*3) Results and Comparison with Other Methods:* Table II gives a full comparison of JPEG and the DJVU method on various documents. Compressing 300 dpi documents using JPEG with a quality[7] comparable to DJVU yields compressed images that are typically five–ten times larger. For the sake of comparison, we subsampled the document images to 100 dpi (with local averaging) and applied JPEG compression with a moderate quality[8] so as to produce files with similar sizes as with the DJVU format. It can be seen in Fig. 5 that JPEG causes many "ringing" artifacts that impede the readability.

## III. Compression and Coding of Video Signals [18], [19]

In order to appreciate the need for compression and coding of the video signals that constitute the multimedia experience, Table III shows the necessary bitrates for several video types. For standard television, including the North American NTSC standard and the European PAL standard, the uncompressed bit rates are 111.2 (NTSC) and 132.7 Mbits/s (PAL). For videoconferencing and videophone applications, smaller format pictures with lower frame rates are standard, leading to the CIF (Common Intermediate Format) and QCIF (Quarter CIF) standards, which have uncompressed bit rates of 18.2 and 3.0 Mbits/s, respectively. Finally, the digital standard for HDTV (in two standard formats) has requirements for an uncompressed bitrate of between 662.9 and 745.7 Mbits/s. We will see that modern signal compression technology leads to compression rates of over 100-to-1.

There have been several major initiatives in video coding that have led to a range of video standards.

- Video coding for video teleconferencing, which has led to the ITU standards called H.261 for ISDN videoconferencing [20], H.263 for POTS[9] videoconferencing [21], and H.262 for ATM/broadband videoconferencing.[10]

[7] Quality factor 25% using the independent JPEG Group's JPEG implementation.

[8] Quality factor 30% using the independent JPEG Group's JPEG implementation.

[9] Plain Old telephone Service, i.e., analog phone line.

[10] H.262 is the same as MPEG-2.

TABLE II
COMPRESSION RESULTS FOR TEN SELECTED IMAGES

| File | Description | Raw | DJVU | Ratio | JPEG-300 | JPEG-100 |
|---|---|---|---|---|---|---|
| metric.tif | (Text on various backgrounds) | 20,000 | 56 | 350 | 496 | 89 |
| hobby002.tif | (Mail order catalog page) | 24,000 | 80 | 300 | 412 | 82 |
| plugin.tif | (Weekly magazine page) | 21,000 | 66 | 318 | 527 | 102 |
| pharm1.tif | (XVIIe century book page) | 6,000 | 88 | 181 | 630 | 85 |
| encyc2347.tif | (Dictionary page) | 23,000 | 98 | 234 | 950 | 150 |
| amend.tif | (US first amendment) | 61,000 | 212 | 287 | 456 | 78 |
| carte.tif | (XVIIIe century map) | 32,000 | 146 | 220 | 586 | 100 |
| wpost2.tif | (Newspaper page section) | 21,000 | 96 | 218 | 435 | 64 |
| curry239.tif | (Book page with picture) | 13,000 | 76 | 171 | 490 | 53 |
| curry242.tif | (Book page without picture) | 14,000 | 40 | 350 | 410 | 73 |

Column "Raw" reports the size in kbytes of an uncompressed TIFF file at 24 bits/pixel and 300 pixels/in. Column "DJVU" reports the total size in kbytes of the DJVU-encoded file. Column "JPEG-300" reports the size in kbytes of 300 pixel/in images JPEG-encoded with quality comparable to DJVU. For the sake of comparison, column "JPEG-100" reports the size in kbytes of 100 pixel/in images JPEG encoded with medium quality. However, the quality of those images is unacceptable for most applications.

- Video coding for storing movies on CD-ROM, with on the order of 1.2 Mbits/s allocated to video coding and 256 kbits/s allocated to audio coding, which led to the initial ISO MPEG-1 (Moving Picture Experts Group) standard.
- Video coding for broadcast and for storing video on DVD (digital video disks), with on the order of 2–15 Mbits/s allocated to video and audio coding, which led to the ISO MPEG-2 standard [22].
- Video coding for low-bit-rate video telephony over POTS networks, with as little as 10 kbits/s allocated to video and as little as 5.3 kbits/s allocated to voice coding, which led to the H.324 standard [23].
- Coding of separate audio–visual objects, both natural and synthetic, which will lead to the ISO MPEG-4 standard.
- Coding of multimedia *Metadata*, i.e., data describing the features of the multimedia data, which will ultimately lead to the MPEG-7 standard.
- Video coding using MPEG-2 for advanced HDTV (high definition TV) with from 15 to 400 Mbits/s allocated to the video coding.

In the following sections, we provide brief summaries of each of these video coders, with the goal of describing the basic coding algorithm as well as the features that support use of the video coding in multimedia applications.

### A. H.26X

The ITU-T H-series of video codecs has evolved for a variety of applications. The H.261 video codec, initially intended for ISDN teleconferencing, is the baseline video mode for most multimedia conferencing systems. The H.262 video codec is essentially the high-bit-rate MPEG-2 standard, and will be described later in this paper. The H.263 low-bit-rate video codec is intended for use in POTS teleconferencing at modem rates of from 14.4 to 56 kbits/s, where the modem rate includes video coding, speech coding, control information, and other logical channels for data.

*1) H.261:* The H.261 codec codes video frames using a discrete cosine transform (DCT) on blocks of size $8 \times 8$ pixels, much the same as used for the JPEG coder described previously. An initial frame (called an intra frame) is coded and transmitted as an independent frame. Subsequent frames, which are modeled as changing slowly due to small motions of objects in the scene, are coded efficiently in the inter mode using motion compensation (MC) in which the displacement of groups of pixels from their position in the previous frame (as represented by motion vectors) are transmitted together with the DCT-coded difference between the predicted and original images [24].

Since H.261 is intended for conferencing applications with only small, controlled amounts of motion in a scene, and with rather limited views consisting mainly of head-and-shoulders views of people along with the background, the video formats that are supported include both the CIF and the QCIF format. All H.261 video is noninterlaced, using a simple progressive scanning pattern.

A unique feature of H.261 is that it specifies a standard coded video syntax and decoding procedure, but most choices in the encoding methods, such as allocation of bits to different parts of the picture, are left open and can be changed by the encoder at will. The result is that the quality of H.261 video, even at a given bit rate, depends greatly on the encoder implementation. This explains why some H.261 systems appear to work better than others.

Since motion compensation is a key element in most video coders, it is worthwhile understanding the basic concepts in this processing step. Fig. 6 shows a block diagram of a motion-compensated image coder. The key idea is to combine transform coding (in the form of the DCT of $8 \times 8$ pixel blocks) with predictive coding (in the form of differential PCM) in


Fig. 5. Comparison of DJVU at 300 dpi (top three pictures) and JPEG at 100 dpi (bottom). The pictures are cut from encyc2347.tif (top), pharm1.tif (bottom), and hobby002.tif (right). The file sizes are given in Table II. The two methods give files of similar sizes, but very different qualities.

TABLE III
CHARACTERISTICS AND UNCOMPRESSED BIT RATES OF VIDEO SIGNALS

| Video Type | Pixels per Frame | Image Aspect Ratio | Frames per Second | Bits/Pixel | Uncompressed Bitrate |
|---|---|---|---|---|---|
| NTSC | 480 × 483 | 4:3 | 29.97 | 16* | 111.2 Mb/s |
| PAL | 576 × 576 | 4:3 | 25 | 16 | 132.7 Mb/s |
| CIF | 352 × 288 | 4:3 | 14.98 | 12† | 18.2 Mb/s |
| QCIF | 176 × 144 | 4:3 | 9.99 | 12 | 3.0 Mb/s |
| HDTV | 1280 × 720 | 16:9 | 59.94 | 12 | 622.9 Mb/s |
| HDTV | 1920 × 1080 | 16:9 | 29.97 | 12 | 745.7 Mb/s |

*Based on the so-called 4:2:2 color subsampling format with two chrominance samples $C_b$ and $C_r$ for every four luminance samples.
†Based on the so-called 4:1:1 color subsampling format with one chrominance sample $C_b$ and $C_r$ for every four luminance samples.

order to give a high degree of compression. Since motion compensation is difficult to perform in the transform domain, the first step in the interframe coder is to create a motion-compensated prediction error. This computation requires only a single frame store in the receiver. The resulting error signal is transformed using a DCT, quantized by an adaptive quantizer, entropy encoded using a variable-length coder (VLC) and buffered for transmission over a fixed rate channel. Motion compensation uses 16 × 16 pixel macroblocks with integer pixel displacement.

*2) H.263 Coding:* The H.263 video codec is based on the same DCT and motion compensation techniques as used in H.261. Several incremental improvements in video coding were added to the H.263 standard for use in POTS conferencing. These included the following.

- Half-pixel motion compensation in order to reduce the roughness in measuring best matching blocks with coarse time quantization. This feature significantly improves the prediction capability of the motion compensation algorithm in cases where there is object motion that needs fine spatial resolution for accurate modeling.
- Improved variable-length coding.
- Reduced overhead.
- Optional modes including unrestricted motion vectors that are allowed to point outside the picture.

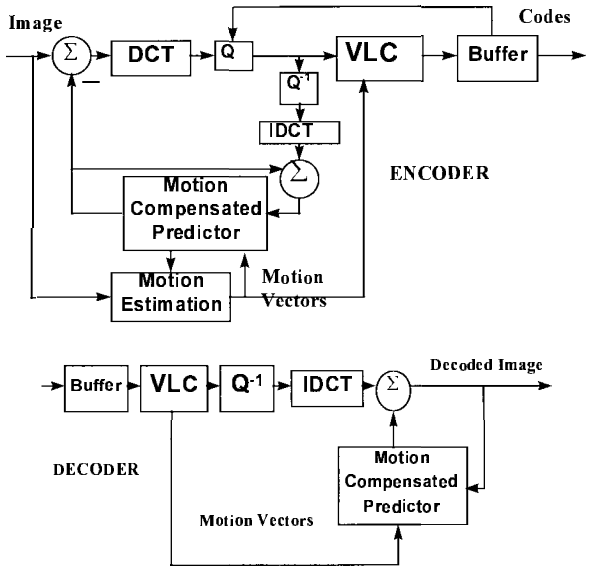


Fig. 6. Motion compensated codec for interframe coding.

- Arithmetic coding in place of the variable length (Huffman) coding.
- Advanced motion prediction mode including overlapped block motion compensation.
- A mode that combines a bidirectionally predicted picture with a normal forward predicted picture.

In addition, H.263 supports a wider range of picture formats including 4CIF (704 × 576 pixels) and 16CIF (1408 × 1152 pixels) to provide a high resolution mode picture capability. Comparisons have shown that H.263 coders can achieve the same quality as H.261 coders at about half the bit rate.

*3) Summary of New "H.263+" Extension Features:* These revisions add optional features to Recommendation H.263 in order to broaden its range of useful application and to improve its compression performance. The additional optional feature set can be summarized in terms of the new types of pictures it can use, the new coding modes which can be applied to those pictures, and the definition of backward-compatible supplemental enhancement information which can be added into the video bit stream.

*a) New types of pictures:*

*Scalability pictures:* Scalable video coding has the potential for improving the delivery of video in error-prone, packet-loss-ridden, and heterogenous networks. It allows the video bit stream to be separated into multiple logical channels, so that some data can be lost or discarded without irreparable harm to the video representation. Three types of scalability pictures were added in this revision, one which provides temporal scalability ($B$) and two which provide SNR or spatial scalability ($EI$ and $EP$):

1) $B$: a picture having two reference pictures, one of which temporally precedes the $B$ picture and one of which is temporally subsequent to the $B$ picture (this is taken from MPEG);
2) $EI$: a picture having a temporally simultaneous reference picture; and
3) $EP$: a picture having two reference pictures, one of which temporally precedes the $EP$ picture and one of which is temporally simultaneous.

*Improved PB frames:* The existing Recommendation H.263 contains a special frame type called a "$PB$ frame," which enables an increase in perceived frame rate with only a moderate increase in bit rate. However, recent investigations have indicated that the Improved $PB$-frame as it exists is not sufficiently robust for continual use. Encoders wishing to use $PB$ frames are limited in the types of prediction that a $PB$ frame can use, which results in a lack of usefulness for $PB$ frames in some situations. An improved, more robust type of $PB$ frame has been added to enable heavier, higher performance use of the $PB$ frame design. It is a small modification of the existing $PB$ frames mode.

*Custom source formats:* The existing Recommendation H.263 is very limited in the types of video input to which it can be applied. It allows only five video source formats defining the picture size, picture shape, and picture clock frequency. The new H.263+ feature set allows a wide range of optional custom source formats in order to make the standard apply to a much wider class of video scenes. These modifications help make the standard respond to the new world of resizable computer window-based displays, high refresh rates, and wide format viewing screens.

*b) New coding modes:* This set of optional extensions of the H.263 video coding syntax also includes nine new coding modes that can be applied to the pictures in the bit stream.

1) Advanced INTRA coding (AIC): A mode which improves the compression efficiency for INTRA macroblock encoding by using spatial prediction of DCT coefficient values.
2) Deblocking filter (DF): A mode which reduces the amount of block artifacts in the final image by filtering across block boundaries using an adaptive filter.
3) Slice structured (SS): A mode which allows a functional grouping of a number of macroblocks in the picture, enabling improved error resilience, improved transport over packet networks, and reduced delay.
4) Reference picture selection (RPS): A mode which improves error resilience by allowing a temporally previous reference picture to be selected which is not the most recent encoded picture that can be syntactically referenced.
5) Reference picture resampling (RPR): A mode which allows a resampling of a temporally previous reference picture prior to its use as a reference for encoding, enabling global motion compensation, predictive dynamic resolution conversion, predictive picture area alteration and registration, and special-effect warping.
6) Reduced-resolution update (RRU): A mode which allows an encoder to maintain a high frame rate during heavy motion by encoding a low-resolution update to a higher resolution picture while maintaining high resolution in stationary areas.
7) Independent segment decoding (ISD): A mode which enhances error resilience by ensuring that corrupted

data from some region of the picture cannot cause propagation of error into other regions.

8) Alternative inter VLC (AIV): A mode which reduces the number of bits needed for encoding predictively coded blocks when there are many large coefficients the block.
9) Modified quantization (MQ): A mode which improves the control of the bit rate by changing the method for controlling the quantizer step size on a macroblock basis, reduces the prevalence of chrominance artifacts by reducing the step size for chrominance quantization, increases the range of representable coefficient values for use with small quantizer step sizes, and increases error detection performance and reduces decoding complexity by prohibiting certain unreasonable coefficient representations.

*c) Supplemental enhancement information:* This revision also allows the addition of supplemental enhancement information to the video bit stream. Although this information does not affect the semantics for decoding the bit stream, it can enable enhanced features for systems which understand the optional additional information (while being discarded harmlessly by systems that do not understand it). This allows a variety of picture freeze and release commands, as well as tagging information associated synchronously with the pictures in the bit stream for external use.

It also allows for video transparency by the use of *chroma key* information [25]. More specifically, a certain color is assigned to represent transparent pixels, and that color value is sent to the decoder. The motion video can then, for example, be superimposed onto a still image background by overlaying only the nontransparent pixels onto the still image.

### B. MPEG-1 Video Coding

The MPEG-1 standard is a true multimedia standard with specifications for coding, compression, and transmission of audio, video, and data streams in a series of synchronized, multiplexed packets. The driving focus of the standard was storage of multimedia content on a standard CDROM, which supported data transfer rates of 1.4 Mbits/s and a total storage capability of about 600 Mbytes. The picture format that was chosen was the SIF format (352 × 288 at 25 noninterlaced frames/s or 352 × 240 pixels at 30 noninterlaced frames/s) which was intended to provide VHS VCR-like video and audio quality, along with VCR-like controls.

The video coding in MPEG-1 is very similar to the video coding of the H.26X series described above, namely, spatial coding by taking the DCT of 8 × 8 pixel blocks, quantizing the DCT coefficients based on perceptual weighting criteria, storing the DCT coefficients for each block in a zig-zag scan, and doing a variable run-length coding of the resulting DCT coefficient stream. Temporal coding was achieved by using the ideas of uni- and bidirectional motion-compensated prediction, with three types of pictures resulting, namely,

- $I$ or intra pictures which were coded independently of all previous or future pictures;
- $P$ or predictive pictures which were coded based on previous $I$ or previous $P$ pictures;
- $B$ or bidirectionally predictive pictures which were coded based on either the next and/or the previous pictures.

High-quality audio coding also is an implicit part of the MPEG-1 standard, and therefore it included sampling rates of 32, 44.1, and 48 kHz, thereby providing provision for near-CD audio quality.

### C. MPEG-2 Coding

The MPEG-2 standard was designed to provide the capability for compressing, coding, and transmitting high-quality, multichannel, multimedia signals over terrestrial broadcast, satellite distribution, and broad-band networks, for example, using ATM (asynchronous transmission mode) protocols. The MPEG-2 standard specifies the requirements for video coding, audio coding, systems coding for combining coded audio and video with user-defined private data streams, conformance testing to verify that bit streams and decoders meet the requirements, and software simulation for encoding and decoding of both the program and the transport streams. Because MPEG-2 was designed as a transmission standard, it supports a variety of packet formats (including long and variable-length packets of from 1 up to 64 kbits), and provides error correction capability that is suitable for transmission over cable TV and satellite links.

*1) MPEG-2 Systems:* The MPEG-2 systems level defines two types of streams: the program stream and the transport stream. The program stream is similar to that used in MPEG-1, but with a modified syntax and new functions to support advanced functionalities. Program stream decoders typically use long and variable-length packets, which are well suited for software-based processing and error-free environments. The transport streams offer the robustness necessary for noisy channels, and also provide the ability to include multiple programs in a single stream. The transport stream uses fixed-length packets of size 188 bytes, and is well suited for delivering compressed video and audio over error-prone channels such as CATV networks and satellite transponders.

The basic data structure that is used for both the program stream and the transport stream data is called the packetized elementary stream (PES) packet. PES packets are generated by packetizing the compressed video and audio data, and a program stream is generated by interleaving PES packets from the various encoders with other data packets to generate a single bitstream. A transport stream consists of packets of fixed length, consisting of 4 bytes of header followed by 184 bytes of data obtained by chopping up the data in the PES packets. The key difference in the streams is that the program streams are intended for error-free environments, whereas the transport streams are intended for noisier environments where some type of error protection is required.

*2) MPEG-2 Video:* MPEG-2 video was originally designed for high-quality encoding of interlaced video from standard TV with bit rates on the order of 4–9 Mbits/s. As it evolved, however, MPEG-2 video was expanded to include high-resolution video, such as HDTV, as well as hierarchical or scalable video coding for a range of applications. Since MPEG-2 video does not standardize the encoding method,

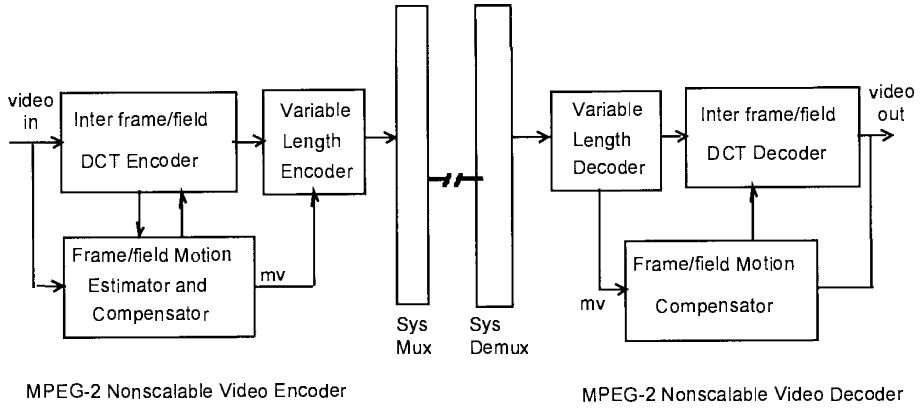


Fig. 7. Generalized codec for MPEG-2 nonscalable video coding.

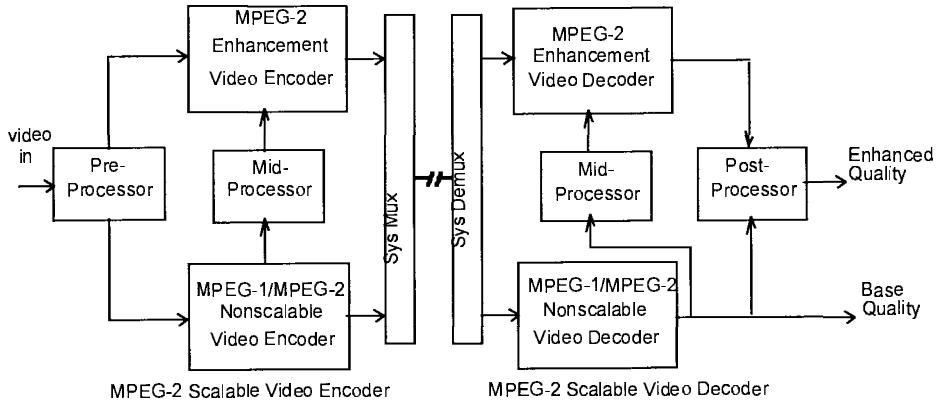


Fig. 8. Generalized codec for MPEG-2 scalable video coding.

but only the video bit stream syntax and decoding semantics, there have evolved two generalized video codecs, one for nonscalable video coding and one for scalable video coding. Fig. 7 shows a block diagram of the MPEG-2 nonscalable video coding algorithm. The video encoder consists of an interframe/field DCT encoder, a frame/field motion estimator and compensator, and a variable-length encoder (VLE). The frame/field DCT encoder exploits spatial redundancies in the video, and the frame/field motion compensator exploits temporal redundancies in the video signal. The coded video bit stream is sent to a systems multiplexer, Sys Mux, which outputs either a transport or a program stream.

The MPEG-2 decoder of Fig. 7 consists of a variable-length decoder (VLD), interframe/field DCT decoder, and the frame/field motion compensator. Sys Demux performs the complementary function of Sys Mux and presents the video bit stream to VLD for decoding of motion vectors and DCT coefficients. The frame/field motion compensator uses a motion vector decoded by VLD to generate motion-compensated prediction that is added back to a decoded prediction error signal to generate decoded video out. This type of coding produces nonscalable video bit streams since, normally, the full spatial and temporal resolution coded is the one that is expected to be decoded.

A block diagram of a generalized codec for MPEG-2 scalable video coding is shown in Fig. 8. Scalability is the property that allows decoders of various complexities to be able to decode video of resolution/quality commensurate with their complexity from the same bit stream. The generalized structure of Fig. 8 provides capability for both spatial and temporal resolution scalability in the following manner. The input video goes through a preprocessor that produces two video signals, one of which (called the base layer) is input to a standard MPEG-1 or MPEG-2 nonscalable video encoder, and the other (called the enhancement layer) is input to an MPEG-2 enhancement video encoder. The two bit streams, one from each encoder, are multiplexed in Sys Mux (along with coded audio and user data). In this manner, it becomes possible for two types of decoders to be able to decode a video signal of quality commensurate with their complexity from the same encoded bit stream.

### *D. HDTV—The Ultimate TV Experience*

High-definition television is designed to be the ultimate television viewing experience providing, in a cost-effective manner, a high-resolution and wide-screen television system with a more panoramic image aspect ratio (16:9 versus the conventional 4:3 ratio for NTSC or PAL), producing much better quality pictures and sound. HDTV systems will be one of the first systems that will not be backward compatible with NTSC or PAL. The key enablers of HDTV systems are the advanced video and audio compression capability, the availability of inexpensive and powerful VLSI chips to realize the system, and the availability of large displays.

The primary driving force behind HDTV is the high level of picture and sound quality that can be received by consumers in their homes. This is achieved by increasing the spatial resolution by about a factor of 2 in both the horizontal and vertical dimensions, providing a picture with about 1000 scan lines and with more than 1000 pixels per scan line. In addition, HDTV allows the use of progressive (noninterlaced) scanning at about 60 frames/s to allow better fast-action sports and far better interoperability with computers, while eliminating the artifacts associated with interlaced pictures.

Unfortunately, increasing the spatial and temporal resolution of the HDTV signal and adding multichannel sound greatly increases its analog bandwidth. Such an analog signal cannot be accommodated in a single channel of the currently allocated broadcast spectrum. Moreover, even if bandwidth were available, such an analog signal would suffer interference both to and from the existing TV transmissions. In fact, much of the available broadcast spectrum can be characterized as a fragile transmission channel. Many of the 6 MHz TV channels are kept unused because of interference considerations, and are designated as *taboo* channels.

Therefore, all of the current HDTV proposals employ digital compression, which reduces the bit rate from approximately 1 Gbit/s to about 20 Mbits/s which can be accommodated in a 6 MHz channel either in a terrestrial broadcast spectrum or a cable television channel. This digital signal is incompatible with the current television system, and therefore can be decoded only by a special decoder.

The result of all of these improvements is that the number of active pixels in an HDTV signal increases by about a factor of 5 (over NTSC signals), with a corresponding increase in the analog bandwidth or digital rate required to represent the uncompressed video signal. The quality of the audio associated with HDTV is also improved by means of multichannel, CD-quality, surround sound in which each channel is independently transmitted.

Since HDTV will be digital, different components of the information can be simply multiplexed in time, instead of frequency multiplexed on different carriers as in the case of analog TV. For example, each audio channel is independently compressed, and these compressed bits are multiplexed with compressed bits from video, as well as bits for closed captioning, teletext, encryption, addressing, program identification, and other data services in a layered fashion.

The U.S. HDTV standard [26] specifies MPEG-2 for the video coding. It uses hierarchical, subpixel motion compensation, with perceptual coding of interframe differences, a fast response to scene and channel changes, and graceful handling of transmission errors. Early versions of the system provide digital transmission of the video signal at 17 Mbits/s with five-channel surround sound CD-quality audio at 0.5 Mbits/s.

In summary, the basic features of the HDTV system that will be implemented are as follows:

- higher spatial resolution, with an increase in spatial resolution by at least a factor of 2 in both the horizontal and vertical directions;
- higher temporal resolution with an increase in temporal resolution by use of a progressive 60 Hz temporal rate;
- higher aspect ratio with an increase in the aspect ratio to 16 : 9 from 4 : 3 for standard TV providing a wider image;
- multichannel CD-quality surround sound with at least four–six channels of surround sound;
- reduced artifacts as compared to analog TV by removing the composite format artifacts as well as the interlace artifacts;
- bandwidth compression and channel coding to make better use of terrestrial spectrum using digital processing for efficient spectrum usage;
- interoperability with the evolving telecommunications and computing infrastructure through the use of digital compression and processing for ease of interworking.

## IV. Techniques for Improved Video Delivery over the Internet

Delivery of play-on-demand video over the Internet involves many problems that simple file transfers do not encounter. These include packet loss, variable delay, congestion, and many others. TCP/IP (transmission control protocol) avoids packet loss by retransmitting lost packets. However, this causes delay that in some cases may become excessive. UDP/IP (universal datagram packets) has no such delay because it does not retransmit packets. However, in that case, the receiving decoder must provide some error recovery mechanism to deal with lost packets.

### A. Streaming Issues for Video

The increases in the computing power and the network access bandwidth available to the general public on their desktop and home computers have resulted in a proliferation of applications using multimedia data delivery over the Internet. Early applications that were based on a first-download-then-play approach have been replaced by "*streaming*" applications [27], which start the playback after a short, initial segment of the multimedia data gets buffered at the user's computer. The success of these applications, however, is self limiting. As they get better, more people try to use them, thereby increasing the congestion on the Internet which, in turn, degrades the performance of such real-time applications in the form of lost and delayed packets.

Although several mechanisms, such as smart buffer management and the use of error-resilient coding techniques, can be employed at the end points to reduce the effects of these packet losses in a streaming application, these can only be effective below a certain packet loss level. For example, generally, the amount of data to be buffered at the beginning of the playback and, hence, the start-up delay is determined in real time based on the reception rate and the rate used to encode the particular material being played. If the congestion level is too high, the amount of data to be buffered initially can be as large as the entire material, effectively converting the streaming application into a download-and-play application. In addition, the time needed to download the material can become so long that users lose interest while waiting for the playback to begin.

The four elements of a streaming system are:

- the compressed (coded) information content, e.g., audio, video, multimedia data, etc.;
- the server;
- the clients;
- the data network (e.g., the Internet) and the connections of the server and the clients to the data network.

A successful streaming application requires a well-designed system that takes into account all of these elements.

Currently, streaming in data networks is implemented as part of the application layer protocols of the transmission, i.e., it uses UDP and TCP at the transport layer. Because of the known shortcomings of TCP, most streaming implementations are based on the inherently unreliable UDP protocol. Thus, whenever there is network congestion, packets are dropped. Also, since delay can be large (order of seconds) and often unpredictable on the Internet, some packets may arrive after their nominal presentation time, effectively turning them into lost packets. The extent of the losses is a function of the network congestion, which is highly correlated with the time of the day and the distance (in terms of the number of routers) between the client and the multimedia source. Thus, streaming itself does not inherently guarantee high quality or low delay playback of real-time mulitmedia material.

The practical techniques that have evolved for improving the performance of streaming-based real-time signal delivery can be classified into four broad areas, namely, the following.

- Client side buffer management—determining how much data needs to be buffered both prior to the start of the streaming playback as well as during the playback, and determining a strategy for changing the buffer size as a function of the network congestion and delay and the load on the media server.
- Error-resilient transmission techniques—increasing client-side resilience to packet losses through intelligent transport techniques, such as using higher priority for transmitting more important parts (headers, etc.) of a stream and/or establishing appropriate retransmission mechanisms (where possible).
- Error-resilient coding techniques—using source (and perhaps combined source and channel) coding techniques that have built-in resilience to packet losses.
- Media control mechanisms—using efficient implementations of VCR-type controls when serving multiple clients.

None of these techniques is sufficient to guarantee high-quality streaming, but in combination, they serve to reduce the problems to manageable levels for most practical systems.

### *B. Quality-of-Service (QOS) Considerations*

The ultimate test of any multimedia system is whether it can deliver the quality of service that is required by the user of the system. The ability to guarantee the QOS of any signal transmitted over the POTS network is one of the key strengths of that network. For reasons discussed throughout this paper, the packet network does not yet have the structure to guarantee QOS for real-time signals transmitted over the packet network. Using the standard data protocol of TCP/IP, the packet network can provide guaranteed eventual delivery for data (through the use of the retransmission protocol in TCP). However, for high-quality real-time signals, there is less possibility of retransmission. Using the of UDP/IP protocol, the packet network delivers most of the packets in a timely fashion. However, some packets may be lost. Using the new real time protocol [28] (RTP), a receiver can feed back to the transmitter the state and quality of the transmission, after which the transmitter can adjust its bit rate and/or error resilience to accommodate.

The ultimate solution to this problem lies in one of three directions, namely, the following.

- Significantly increased bandwidth connections between all data servers and all clients, thereby making the issues involved with traffic on the network irrelevant. This solution is highly impractical and somewhat opposed to the entire spirit of sharing the packet network across a range of traffic sites so that it is efficiently and statistically multiplexed by a wide range of traffic. However, such overengineering is often done.
- Provide *virtual circuit* capability for different grades of traffic on the packet network. If real-time signals were able to specify a virtual circuit between the server and the client so that all subsequent packets of a specified type could use the virtual circuit without having to determine a new path for each packet, the time to determine the packet routing path would be reduced to essentially a table lookup instead of a complex calculation. This would reduce the routing delay significantly.
- Provide *grades of service* for different types of packets, so that real-time packets would have a higher grade of service than a data packet. This would enable the highest grade of service packets (hopefully reserved for real-time traffic) to bypass the queue at each router and be moved with essentially no delay through routers and switches in the data network.

Guaranteed, high-quality delivery of real-time information over the Internet requires some form of resource reservations. Although there is ongoing work to bring such functionality to the Internet (via RSVP [29] and IPv6 [30]), its global implementation is still a few years down the line. This is due to the fact that the network bandwidth is bound to stay as a limited resource needed by many in the foreseeable future and, therefore, a value structure must be imposed on data exchanges with any defined QOS. Establishing such a structure may be easy for a corporate intranet; however, a nationwide implementation requires new laws, and a global implementation requires international agreements!

A key problem in the delivery of "on-demand" multimedia communications over the Internet is that the Internet today cannot guarantee the quality of real-time signals such as speech, audio, and video because of lost and delayed packets due to congestion and traffic on the Internet. An alternative to this, which can be implemented immediately, is to use the existing POTS telephone network, in particular the ISDN, together with the Internet to provide a high QOS connection when needed.

The FusionNet [31] service overcomes this problem by using the Internet only to browse (in order to find the multimedia material that is desired), to request the video and audio, and to control the signal delivery (e.g., via VCR-like controls). FusionNet uses either POTS or ISDN to actually deliver guaranteed quality of service (QOS) for real-time transmission of audio and video.

FusionNet service can be provided over a single ISDN $B$ channel. This is possible because the ISP (Internet service provider) provides ISDN access equipment that seamlessly merges the guaranteed QOS audio/video signal with normal Internet traffic to and from the user via PPP (point-to-point protocol) over dialed-up ISDN connections. Unless the traffic at the local ISP is very high, this method provides high-quality FusionNet service with a single ISDN $B$ channel. Of course, additional $B$ channels can always be ganged together for higher quality service.

## V. MPEG-4

Most recently, the focus of video coding has shifted to *object-based coding* at rates as low as 8 kbits/s or lower and as high as 1 Mbit/s or higher. Key aspects of this newly proposed MPEG standard [32] include independent coding of objects in a picture; the ability to interactively composite these objects into a scene at the display; the ability to combine graphics, animated objects, and natural objects in the scene; and finally, the ability to transmit scenes in higher dimension formats (e.g., 3-D). Also inherent in the MPEG-4 standard is the concept of video scalability, both in the temporal and spatial domains, in order to effectively control the video bit rate at the transmitter, in the network, and at the receiver so as to match the available transmission and processing resources. MPEG-4 is scheduled to be finished essentially in 1998.

MPEG-4 builds on and combines elements from three fields: digital television, interactive graphics, and the World Wide Web. It aims to provide a merging of the production, distribution, and display elements of these three fields. In particular, it is expected that MPEG-4 will provide:

- multimedia content in a form that is reusable, with the capability and flexibility of incorporating on-the-fly piece parts from anywhere and at any time the application desires;
- protection mechanisms for intellectual property rights associated with that content;
- content transportation with a quality of service (QOS) custom tailored to each component;
- high levels of user interaction, with some control features being provided by the multimedia itself and others available locally at the receiving terminal.

The design of MPEG-4 is centered around a basic unit of content called the *audio-visual object* (AVO). Examples of AVO's are a musician (in motion) in an orchestra, the sound generated by that musician, the chair she is sitting on, the (possibly moving) background behind the orchestra, explanatory text for the current passage, etc. In MPEG-4, each AVO is represented separately, and becomes the basis for an independent stream.

In order for a viewer to receive a selection that can be seen on a display and heard through loudspeakers, the AVO's must be transmitted from a storage (or live) site. Since some AVO's may have an extremely long duration, it is usually undesirable to send each one separately in its entirety one after the other. Instead, some AVO's are *multiplexed* together and sent simultaneously so that replay can commence shortly after transmission begins. Other AVO's needing a different QOS can be multiplexed and sent on another transmission path that is able to provide that QOS.

Upon arrival of the AVO's, they must be assembled or *composed* into an audio–visual *scene*. In general, the scene may be three dimensional. Since some of the AVO's, such as moving persons or music, involve real-time portrayal, proper time synchronization must also be provided.

AVO's can be arbitrarily composed. For example, the musicians could be moved around to achieve special effects, e.g., one could choose to see and hear only the trumpet section. Alternatively, it would be possible to delete the drummer (and the resulting drum audio component), leaving the rest of the band so that the viewer could play along on his own drums. Fig. 9 illustrates scene composition in MPEG-4.

Following composition of a 3-D scene, the visual AVO's must be projected onto a viewer plane for display, and the audio AVO's must be combined for playing through loudspeakers or headphones. This process is called *rendering*. In principle, rendering does not require standardization. All that is needed is a view point and a window size.

### A. MPEG-4 Multiplex [33]

The transmission of coded real-time AVO's from one or more sources to a destination is accomplished through the two-layer multiplex shown in Fig. 10. Each coded *elementary* AVO is assigned to an *elementary stream*. The FlexMux layer then groups together elementary streams (ES's) having similar QOS requirements to form *FlexMux streams*. The TransMux layer then provides transport services matched to the required QOS of each FlexMux stream. TransMux can be any of the existing transport protocols such as (UDP)/IP, (AAL5)/ATM, MPEG-2 Transport Stream, etc. It is not standardized by MPEG-4 (as yet).

### B. MPEG-4 Systems

The systems part of MPEG-4 specifies the overall architecture of a general receiving terminal. Fig. 11 shows the major elements. FlexMux streams coming from the network are passed to appropriate FlexMux demultiplexers that produce elementary streams (ES). The ES's are then syntactically decoded into intermediate data such as motion vectors and DCT coefficients and then passed to appropriate decompressors that produce the final AVO's, which are composed and rendered into the final display.

To place the AVO's into a scene (composition), their spatial and temporal relationships (the scene structure) must be known. For example, the scene structure may be defined by a multimedia author or interactively by the end viewer. Alternatively, it could be defined by one or more network elements

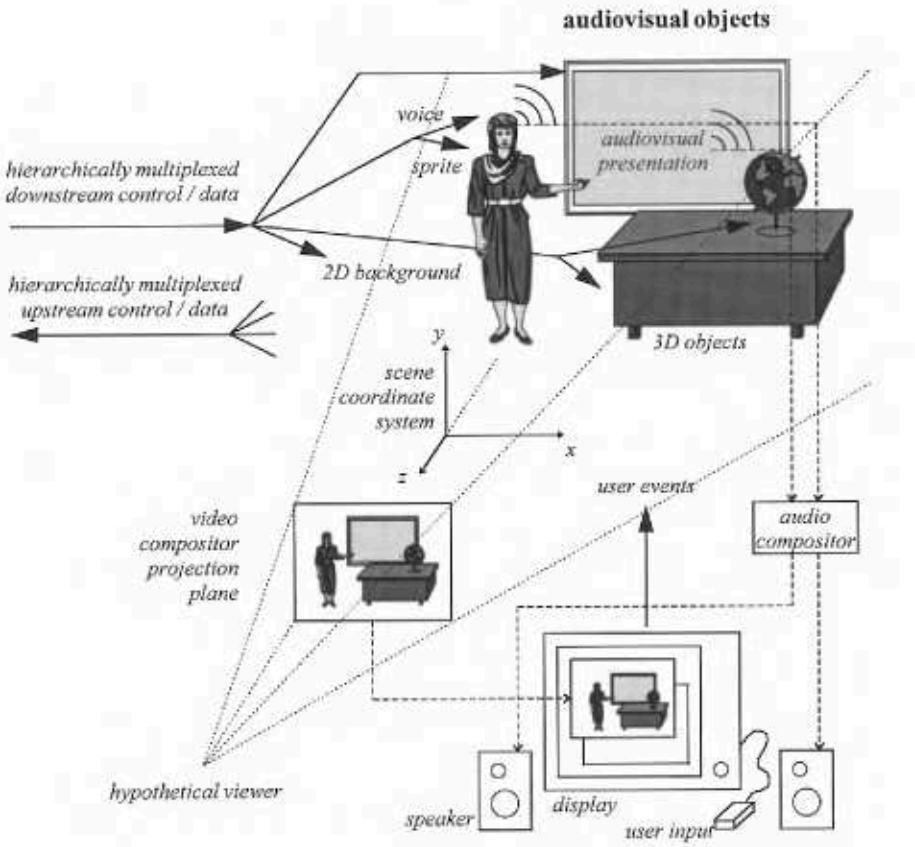


Fig. 9. Example of an MPEG-4 scene (courtesy of MPEG Systems Group).

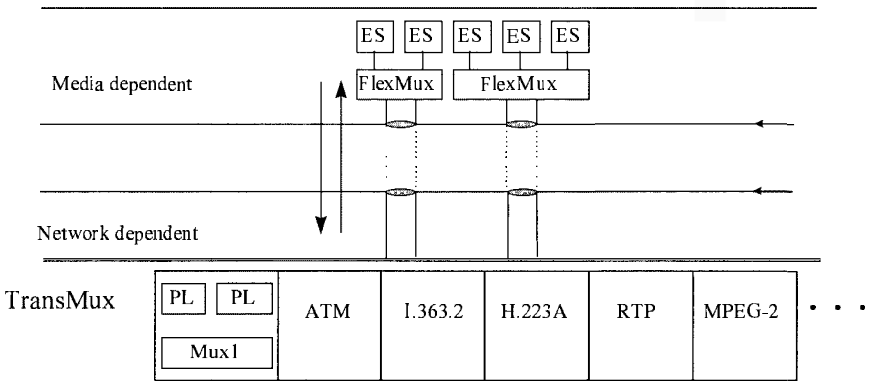


Fig. 10. MPEG-4 two-layer multiplex (courtesy of MPEG Systems Group).

that manage multiple sources and multipoint communication between them. In any event, the composition part of MPEG-4 systems specifies the methodology for defining this structure.

Temporally, all AVO's have a single time dimension. For real-time, high-quality operation, end-to-end delay from the encoder input to the decoder output should be constant. However, at low bit rates or operation over lossy networks, the ideal of constant delay may have to be sacrificed. This delay is the sum of encoding (including video frame dropping), encoder buffering, multiplexing, communication or storage, demultiplexing, decoder buffering, decoding (including frame repeating), and presentation delays.

The transmitted data streams must contain either implicit or explicit timing information. As in MPEG-1 and MPEG-2, there are two kinds of timing information. One indicates periodic values of the encoder clock, while the other tells the desired presentation timing for each AVO. Either one is optional, and if missing, must be provided by the receiver compositor.

Spatially, each AVO has its own *local coordinate system*, which serves to describe local behavior independent of the scene or any other AVO's. AVO's are placed in a scene by specifying (possibly dynamic) coordinate transformations from the local coordinate systems into a common *scene coordinate system*, as shown in Fig. 9. Note that the coordinate transformations, which position AVO's in a scene, are part of the scene structure, not the AVO. Thus, object motion in the scene is the motion specified locally by the AVO plus the motion specified by the dynamic coordinate transformations.

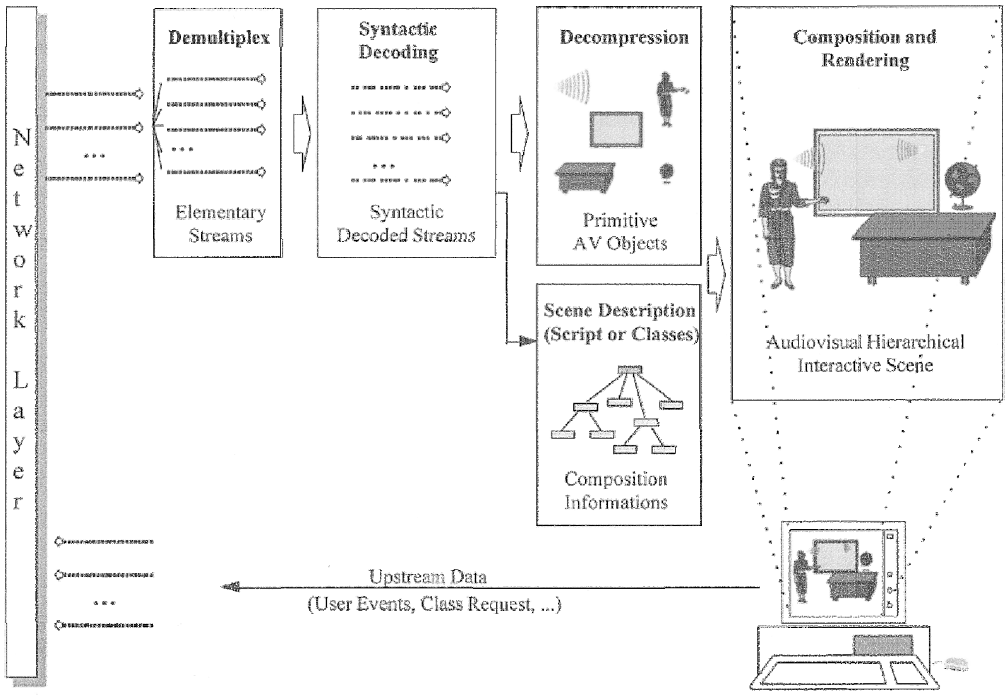


Fig. 11. Major components of an MPEG-4 terminal (receiver side) (courtesy of MPEG Systems Group).

The scene description is sent as a separate elementary stream. This allows for relatively simple bit-stream editing, one of the central functionalities in MPEG-4. In bit-stream editing, we want to be able to change the composition and scene structure without decoding the AVO bit streams and changing their content.

In order to increase the power of editing and scene manipulation even further, the MPEG-4 scene structure may be defined hierarchically and represented as a tree. Each node of the tree is an AVO, as illustrated in Fig. 12. Nodes at the leaves of the tree are *primitive nodes*. Nodes that are parents of one or more other nodes are *compound nodes*. Primitive nodes may have elementary streams assigned to them, whereas compound nodes are of use mainly in editing and compositing.

In the tree, each AVO is positioned in the local coordinate system of its parent AVO. The tree structure may be dynamic, i.e., the positions can change with time, and nodes may be added or deleted. The information describing the relationships between parent nodes and children nodes is sent in the elementary stream assigned to the scene description.

### C. Natural 2-D Motion Video [34]

MPEG-4 coding for natural video will, of course, perform efficient compression of traditional video camera signals for storage and transmission in multimedia environments. However, it will also provide tools that enable a number of other functionalities such as object scalability, spatial and temporal scalability, sprite overlays, error resilience, etc. MPEG-4 video will be capable of coding both conventional rectangular video as well as well as arbitrarily shaped 2-D objects in a video scene. The MPEG-4 video standard will be able to code video ranging from very low spatial and temporal resolutions in progressive scanning format up to very high spatial and temporal resolutions for professional studio applications, including

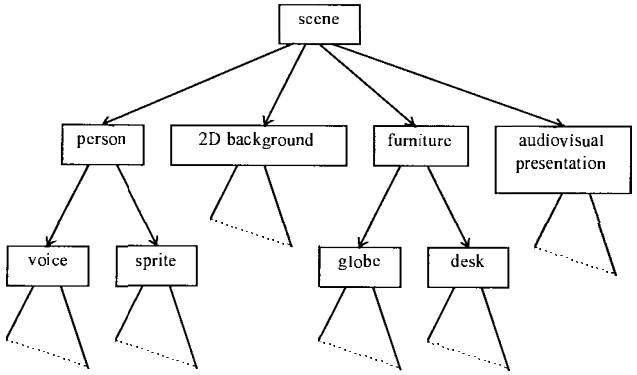


Fig. 12. Logical structure of the scene (courtesy of MPEG Systems Group).

interlaced video. The input frame rate can be nonuniform, including single picture input, which is considered as a special case.

The basic video AVO is called a *video object* (VO). If the VO is *scalable*, it may be split up, coded, and sent in two or more *video object layers* (VOL). One of these VOL's is called the *base layer*, which all terminals must receive in order to display any kind of video. The remaining VOL's are called *enhancement layers*, which may be expendable in case of transmission errors or restricted transmission capacity. For example, in a broadcast application transmitting to a variety of terminals having different processing capabilities or whose connections to the network are at different bit rates, some of the receiving terminals might receive all of the VOL's, while others may receive only a few, while still others may receive only the base layer VOL.

In a scalable video object, the VO is a compound AVO that is the parent of two or more child VOL's. Each VOL is a primitive AVO, and is carried by a separate elementary stream.

A snapshot in time of a video object layer is called a *video object plane* (VOP). For rectangular video, this corresponds to a *picture* in MPEG-1 and MPEG-2 or a *frame* in other standards. However, in general, the VOP can have an arbitrary shape.

The VOP is the basic unit of coding, and is made up of luminance ($Y$), and chrominance ($Cb$, $Cr$) components plus shape information. The shape and location of VOP's may vary from one VOP to the next. The shape may be conveyed either implicitly or explicitly. With implicit shape coding, the irregularly shaped object is simply placed in front of a (say, blue–green) colored background known to the receiver, and a rectangular VOP containing both object and background is coded and transmitted. The decoder retrieves the object by simple chroma keying, as in H.263+.

Explicit shape is represented by a rectangular *alpha plane* that covers the object to be coded [35]. An alpha plane may be *binary* (0 for transparent, 1 for object) if only the shape is of interest, or it may be *gray level* (up to 8 bits/pixel) to indicate various levels of partial transparency for the object. If the alpha plane has a constant gray value inside the object area, that value can be sent separately and the alpha plane coded as a binary alpha plane. Blending the alpha map near the object boundary is not supported by the video decoder since this is a composition issue.

Binary alpha planes are coded as a bitmap one macroblock at a time using a context-based arithmetic encoder with motion compensation. In the case of arbitrary gray-level alpha planes, the outline of the object is coded as a binary shape as above, while the gray levels are coded using DCT and motion compensation.

Coding of texture for an arbitrarily shaped region whose shape is described with an alpha map is different from traditional methods. Techniques are borrowed from both H.263 and earlier MPEG standards. For example, intraframe coding, forward prediction motion compensation, and bidirectional motion compensation are used. This gives rise to the definitions of I-VOP's, P-VOP's, and B-VOP's for VOP's that are intra coded, forward predicted, or bidirectionally predicted, respectively. For boundary blocks, i.e., blocks that are only partially covered by the arbitrarily shaped VOP, the texture of the object is extrapolated to cover the background part of the block. This process is called *padding*, and is used for efficient temporal prediction from boundary blocks in temporally adjacent pictures.

Fig. 13 shows the block diagram of this object-based video coder. In contrast to the block diagram shown in the MPEG-4 standard, this diagram focuses on the object-based mode in order to allow a better understanding of how shape coding influences the encoder and decoder. Image analysis creates the bounding box for the current VOP $S_k$ and estimates texture and shape motion of $S_k$ with respect to the reference VOP $S'_{k-1}$. Shape motion vectors of transparent macroblocks are set to 0. Parameter coding encodes the parameters predictively. The parameters are transmitted, decoded, and the new reference VOP is stored in the VOP memory and also handed to the compositor of the receiver for display.

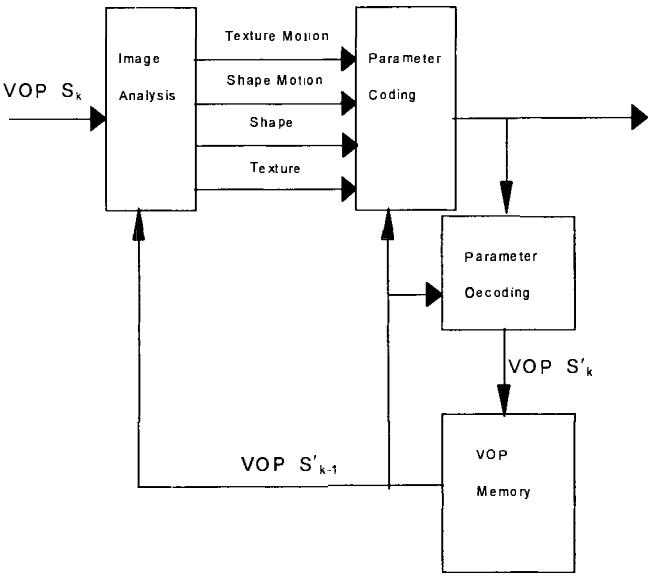


Fig. 13. Block diagram of MPEG-4 object-based coder for arbitrary-shaped video objects.

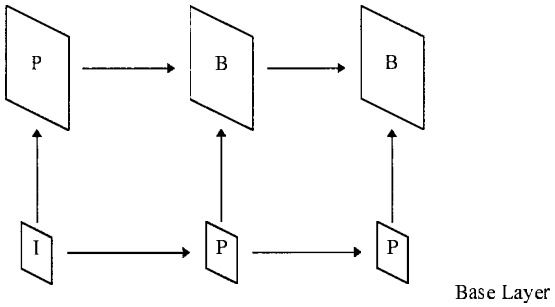


Fig. 14. Spatial scalability with two layers (courtesy MPEG Video Group).

The parameter coder encodes first the shape of the boundary blocks using shape and texture motion vectors for prediction. Then shape motion vectors are coded. The shape motion coder knows which motion vectors to code by analyzing the possibly lossily encoded shape parameters. For texture prediction, the reference VOP is padded as described above. The prediction error is then padded using the original shape parameters to determine the area to be padded. Using the original shape as a reference for padding is again an encoder choice. Finally, the texture of each macroblock is encoded using DCT.

*1) Multifunctional Coding Tools and Algorithms:* Multifunctional coding refers to features other than coding efficiency. For example, object based spatial and temporal scalabilities are provided to enable broad-based access over a variety of networks and facilities. This can be useful for Internet and database applications. Also, for mobile multimedia applications, spatial and temporal scalabilities are extremely useful for channel bandwidth scaling for robust delivery. Spatial scalability with two layers is shown in Fig. 14. Temporal scalability with two layers is shown in Fig. 15.

Multifunctional coding also addresses multiview and stereoscopic applications, as well as representations that enable simultaneous coding and tracking of objects for surveillance

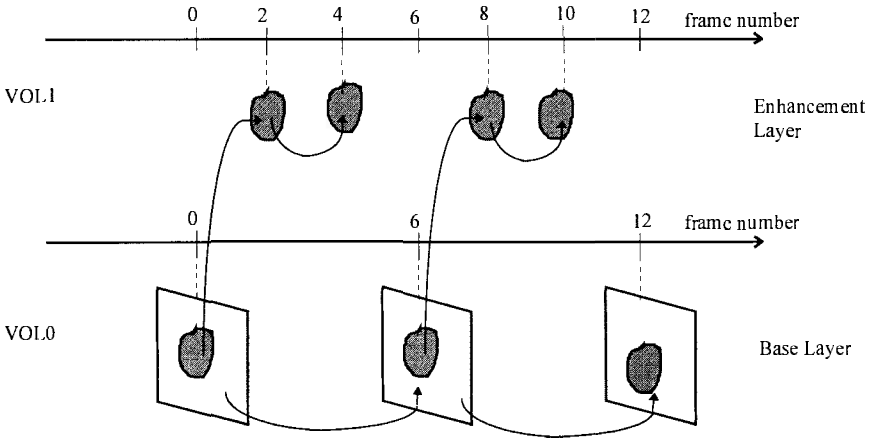


Fig. 15. Temporal scalability with two layers (courtesy MPEG Video Group).

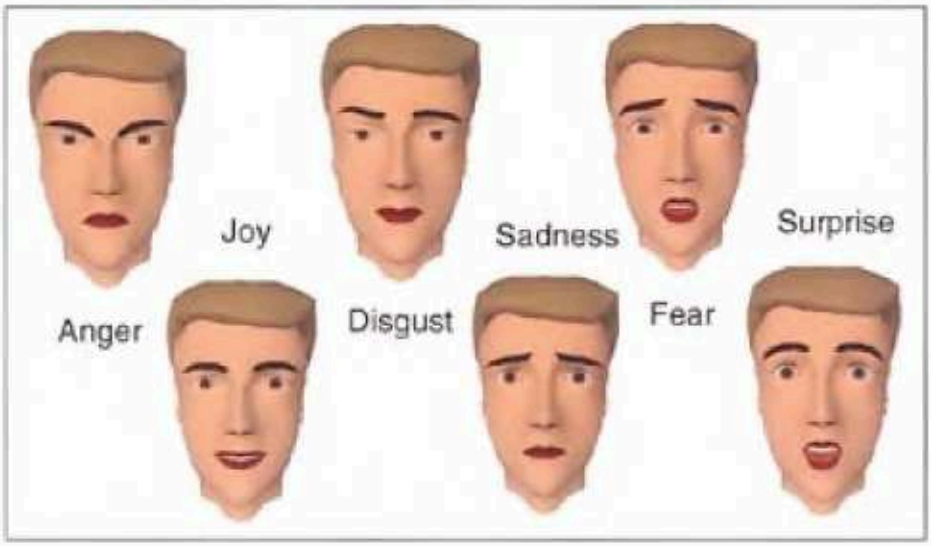


Fig. 16. Examples of facial expressions as defined by MPEG-4.

and other applications. Besides the aforementioned applications, a number of tools are being developed for segmentation of a video scene into objects and for coding noise suppression.

*2) Error Resilience:* Error resilience is needed, to some extent, in all transmission media. In particular, due to the rapid growth of mobile communications, it is extremely important that audio and video information is sent successfully via wireless networks. These networks are typically error prone and usually operate at relatively low bit rates., e.g., less than 64 kbits/s. Both MPEG and the ITU-T are working on error resilience methods, including forward error correction (FEC), automatic request for retransmission (ARQ), scalable coding, slice-based bit-stream partitioning, and motion-compensated error correction [36].

### *D. Synthetic Images*

Several efforts are underway to provide synthetic image capabilities in MPEG-4. There is no wish to reinvent existing graphics standards. Thus, MPEG-4 uses the Virtual Reality Modeling Language (VRML) as a starting point for its synthetic image specification. MPEG-4 will add a number of additional capabilities plus predictable performance at the receiving terminal.

The first addition is a synthetic *face and body* (FAB) animation capability, which is a model-independent definition of artificial face and body animation parameters. With these parameters, one can represent facial expressions, body positions, mouth shapes, etc. Fig. 16 shows examples of facial expressions defined by MPEG-4, and Fig. 17 shows the feature points of a face that can be animated using MPEG-4 face animation parameters. The FAB model to be animated can be either resident in the receiver or a model that is completely downloaded by the transmitting application.

Capabilities include 3-D feature point positions, 3-D head and body control meshes for animation, texture mapping of face and body, and personal characteristics. MPEG-4 employs a text-driven mouth animation combined with a text-to-speech synthesizer for a complete text-to-talking-head implementation.

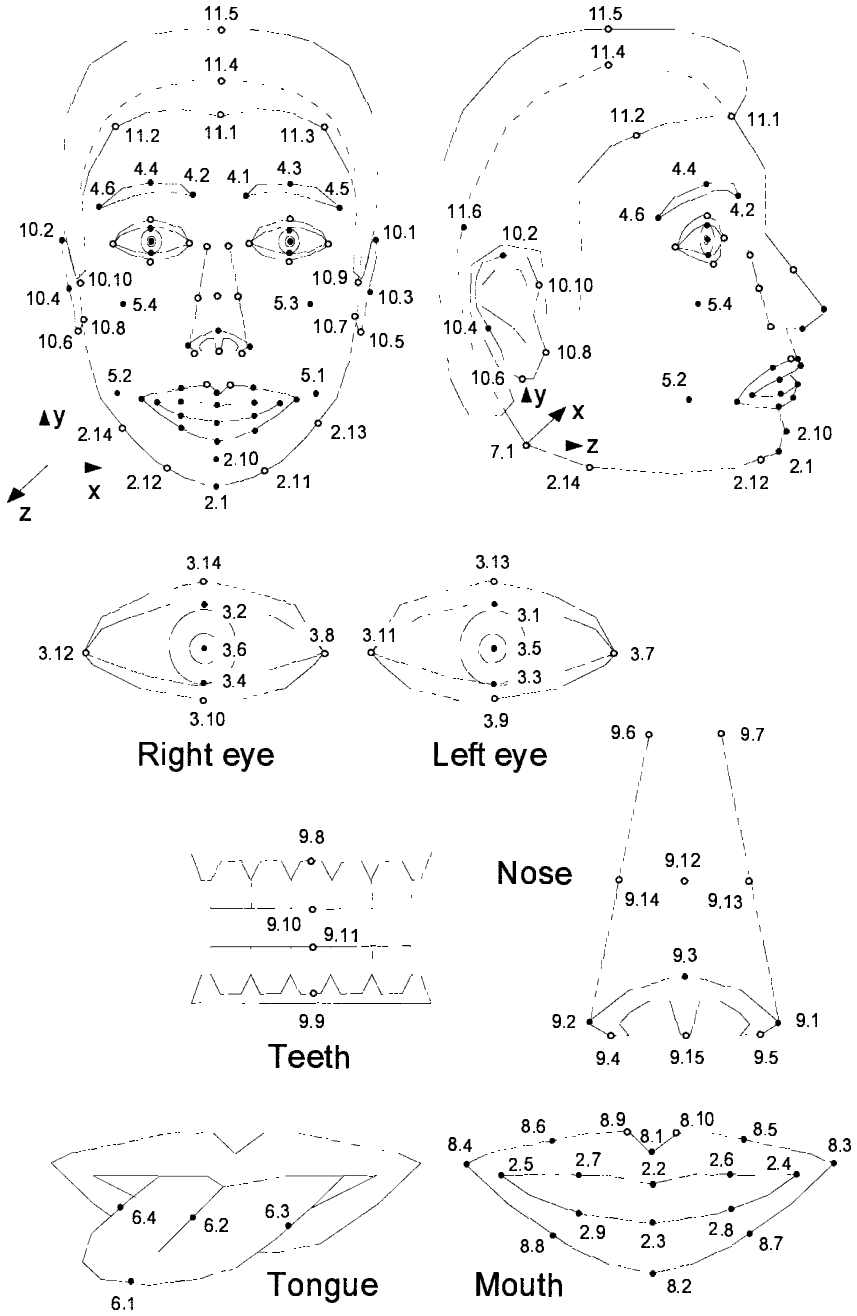


Fig. 17. Head control points for face and body animation (courtesy of MPEG SNHC Group).

Another capability is for texture mapping of real image information onto artificial models such as the FAB model. For this, wavelet-based texture coding is provided. An advantage of wavelet based coding is the relative ease of adjusting the resolution of the visual information to match the requirements of the rendering. For example, if an object is being composed at a distance far from the viewer, then it is not necessary to send the object with high resolution.

Associated with this is a triangular mesh-modeling capability to handle any type of 2-D or 3-D synthetic or natural shape. This also facilitates integration of text and graphics onto synthetic and natural imagery. For example, putting text onto a moving natural object requires a tracking of features on the natural object, which is made easier by a mesh-based representation of the object.

## VI. Conclusion of MPEG-4 and Launching of MPEG-7

In summary, MPEG-4 will integrate most of the capabilities and features of multimedia into one standard, including live audio/video, synthetic objects, and text, all of which can be combined on-the-fly. Multipoint conversations can be facilitated by displays tailored to each viewer of group of viewers. Multimedia presentations can be sent to auditoriums, offices, homes, and mobiles with delivery scaled to the capabilities of the various receivers.

However, MPEG-4 is not the end. Plans are now underway to begin MPEG-7, which is aimed at the definition of audiovisual (AV) multimedia features for purposes such as searching large databases, identification, authentication, cataloging, etc. This phase of MPEG [37] is scheduled for completion essentially in 2000.

MPEG-7 is aiming at the description of content. Some of the ideas and applications are still a little vague, but at its simplest, the content will be described simply by text. For example, "dozens of trees," "six horses," "two cowboys," or "the Queen of England." This would allow classification and searching either by category or by specific naming.

A little more complicated could be the notes of a musical selection. For example, B flat, G sharp, or, in fact, the audio waveform itself. Then you could search for a song you wanted to purchase by singing a few notes. Or businesses could search for copyright infringements by comparing parts of songs with each other.

Now, with a lot more computational complexity comes the hope to be able to specify pictorial features in some parametric way. For example, much work has gone into describing human faces by the positions of eyes, nostrils, chins, cheeks, ears, etc. And for purposes of recognition, at least, we need a representation that is independent of hair color, beards, and mustaches. For children, we would like a representation that is useful even as the child grows.

For marketing applications, we would like to be able find the dress that has a collar that is "sort of shaped like this" and shoulders that "sort of look like this." I would like to be able to stand in front of a camera, and request a necktie that goes with the shirt and sweater I am wearing. I could show a picture of my house. Then I could ask for some flower seeds that would grow into flowers with just the right shape and just the right color.

We also expect MPEG-7 to make it much easier to create multimedia content. For example, if you need a video clip of a talking bird, we want it to be easy to find. Then you simply type or speak the words you want the bird to say, and the talking bird will be instantly added to your program. The possibilities for this technology are limited only by our imagination.

### A. MPEG-7 Requirements and Terminology

MPEG-7 aims to standardize a set of multimedia description schemes and descriptors, as well as ways of specifying new description schemes and descriptors. MPEG-7 will also address the coding of these descriptors and description schemes. Some of the terminology (not finalized) needed by MPEG-7 includes the following.

*Data*—AV information that will be described using MPEG-7, regardless of the storage, coding, display, transmission, medium, or technology. This definition is intended to be sufficiently broad to encompass text, film, and any other medium. For example, an MPEG-4 stream, a videotape, or a paper book.

*Feature*—A feature is a distinctive part or characteristic of the data which stands for something to somebody in some respect or capacity. Examples might be the color of an image, but also its style, or the author.

*Descriptor*—A descriptor is a formal semantic for associating a representation value to one or more features. Note: it is possible to have multiple representations for a single feature. Examples might be a time code for representing duration, or both color moments and histograms for representing color.

*Description Scheme*—The description scheme defines a structure of descriptors and their relationships.

*Description*—A description is the entity describing the AV data that consist of description schemes and descriptors.

*Coded description*—Coded description is a compressed description allowing easy indexing, efficient storage, and transmission.

MPEG-7 aims to support a number of audio and visual descriptions, including free text, $N$-dimensional spatiotemporal structure, statistical information, objective attributes, subjective attributes, production attributes, and composition information. For visual information, descriptions will include color, visual objects, texture, sketch, shape, volume, spatial relations, motion, and deformation.

MPEG-7 also aims to support a means to describe multimedia material hierarchically according to abstraction levels of information in order to efficiently represent a user's information need at different levels. It should allow queries based on visual descriptions to retrieve audio data and vice versa. It should support the prioritization of features in order that queries may be processed more efficiently. The priorities may denote a level of confidence, reliability, etc. MPEG-7 intends to support the association of descriptors to different temporal ranges, both hierarchically as well as sequentially.

MPEG-7 aims to be effective (you get what you are looking for and not other stuff) and efficient (you get what you are looking for, quickly) retrieval of multimedia data based on their contents, whatever the semantic involved.

### B. New Technology for Multimedia (MM)

In order to achieve to grand goals of MPEG-7, an enormous amount of new technology needs to be invented. For example, a quantum leap in image understanding algorithms will have to take place, including motion analysis, zoom and pan compensation, foreground–background segmentation, texture modeling, etc.

Some of this technology may be useful for compression coding as well. For example, corresponding to preprocessing and analysis at the transmitter is "synthesis" and "postprocessing" at the receiver.

We need to keep in mind that in our assumed applications, the viewer cannot see the original (MM) information. If a viewer can tell the difference between image information and noise, then so should a very smart postprocessor. For example, many objects have known shape and color. No picture should ever be displayed that has the wrong shape for known objects.

MPEG-7 will find its greatest use on the Internet. On the Internet itself, browsers have become a very popular human interface to deal with the intricacies of database searching.

Bulletin boards of many types offer a huge variety of material, most of it at no additional charge. Multimedia will soon follow.

One of the favorite applications for the global information infrastructure (GII) is home shopping. And from this, we can expect such benefits as virtual stores, search agents, automated reservations, complex services being turned into commodities, specialized services tailored to the individual, and so on, and so on. Commercials ought to be tuned to the user, not the other way around.

Being able to connect to everyone means we will have the usual crowd of (let us say) entrepreneurs trying to get your money. There is no way to tell what is true and what is false just by reading it. Security tools are available if users choose to use them. Anonymity ought to be possible. Impersonation ought to be impossible.

If all of this come to pass, there will be a frightening overload of MM information. We will all be inundated. Today's MM information handling is rather primitive, involving many manual operations for creation and assimilation.

We need automated ways to identify and describe the content of audio, video, text, graphics, numerical data, software, etc. This should involve as little human interaction as possible. Algorithms for MM understanding should disect the MM objects into their constituent parts, identify them, and provide labels. Required technologies include audio speech recognition, video scene analysis, text concept identification, software capability detection, automatic closed captioning, etc.

Given descriptions of MM objects, we want to create databases and catalogs with index information, automatically search for relationships between MM objects and include these in catalog, distribute catalog index information, check for obsolescence, and delete old objects.

We want to automatically determine the quality, technical level, timeliness, and relevance of each MM object. Authenticity should also be checked at this stage. We would like to produce metrics for novelty, utility, redundancy, boredom, etc., tailored to the individual user. We should give feedback to author/creators of an MM object as a straw poll of usefulness.

Database search engines should discover nonredundant, high-quality MM objects of a technical level matched appropriately to the user. These should be combined, using hyperwhatever "glue," into a meaningful, enjoyable, and exciting program.

We want machine-assisted or automatic authoring/creation of MM objects, including database search for MM piece parts useful for current creation, interaction with a quality checker to maintain utility and nonredundancy. Then check that all technical levels are covered.

We would like edutainment capabilities for motivation building customized to the user. This is especially needed for children. We have to compete with MM games and entertainment. We need interactive measures of attention. We need reward mechanisms for cooperation and success.

We would like tools for audio/video speed up or slow down, along with audio gap removal. This can turn a one hour slow talk into a half hour fast-paced talk.

There will be problems of intellectual property rights. The problem of copyrights and patents might be solved by downloadable software that can only be executed or watched once. After that, it self destructs or times out. Another solution is to uniquely label each piece of software so that if a violation is detected, you will know who made the original purchase. This is all reminiscent of the advent of the copying machine. In the early days, lots of copying was illegal, but you could only catch the blatant violators. The motto may be (as in many parts of life), "if you won't get caught, go ahead and do it."

Which brings us to our final topic, virtuous virtuosos. It may turn out that we will see some "unvirtuous" behavior on the GII. Now, should there be any limits on MM GII traffic? Well, the liberals will say no, it is just like a bookstore. The conservatives will say yes, it is like broadcast television. It is clearly a hard question. Violence has been in movies for a long time, the objective being to scare the daylights out of you. Some of the recent computer games are pretty violent too, and I guess the appeal there is to your aggressions... or hostility... or maybe even hatreds?? We can see the day when hate groups use MM games to spread their propaganda. Even now, such messages as, "your grandfather committed genocide" can be found.

Some networks today have censors, both unofficial and unofficial. On some networks, anyone can censor. If you see a message you do not like, you can delete it. Or you can issue a "cancelbot" that looks for a message from a certain individual or organization, and cancels all of them. This might be considered antisocial, but the current paradigm seems to allow it as long as you announce to everyone that you have done it. If there are only a few objections, then it seems to be okay.

Well, what of the future? It is not hard to predict the technical possibilities. Just read Jules Verne. But it is extremely hard to predict exactly when, and at what cost. Herein lies our problem and our opportunity. However, we *are* sure of one thing, and that is that communication standards will make the new technology happen sooner and at a lower cost.

## ACKNOWLEDGMENT

The original work described in this paper was done by a number of individuals at AT&T Laboratories as well as outside of AT&T. The authors would like to acknowledge both the strong contributions to the literature, as well as the fine work done by the following individuals, as reported on in this paper: video coding: T. Chen, A. Reibman, R. Schmidt; streaming: D. Gibbon; FusionNet: G. Cash; DJVU: P. Simard.

## REFERENCES

[1] R. Cox, B. Haskell, Y. LeCun, B. Shahraray, and L. Rabiner, "On the applications of multimedia processing to communications," *Proc. IEEE*, vol. 86, pp. 755–824, May 1998.

[2] A. N. Netravali and B. G. Haskell, *Digital Pictures-Representation, Compression, and Standards*, 2nd ed. New York: Plenum, 1995.

[3] K. R. McConnell, D. Bodson, and R. Schaphorst, *Fax, Digital Facsimile Technology and Applications*. Boston, MA: Artech House, 1992.

[4] W. B. Pennebaker and J. L. Mitchell, "Other image compression standards," in *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993, ch. 20.

[5] ISO/IEC International Standard 11544, "Progressive bi-level image compression," 1993.

[6] K. Mohiudin, J. J. Rissanan, and R. Arps, "Lossless binary image compression based on pattern matching," in *Proc. Int. Conf. Comput., Syst., Signal Processing*, Bangalore, India, 1984.
[7] P. G. Howard, "Lossy and lossless compression of text images by soft pattern matching," in *Proc. Data Compression Conf.*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, IEEE Press, 1996, pp. 210–219.
[8] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Trans. Comput.*, pp. 1174–1179, Nov. 1974.
[9] P. G. Howard, "Text image compression using soft pattern matching," *Comput. J.*, 1997.
[10] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, Dec. 1991.
[11] ISO/IEC International Standard 10918-1, "Digital compression and coding of continuous-tone still images," 1991.
[12] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
[13] ISO/IEC JTC1/SC29/WG1 N505, "Call for contributions for JPEG-2000," 1997.
[14] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, June 3, 1996.
[15] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[16] P. Haffner, L. Bottou, P. G. Howard, P. Simard, Y. Bengio, and Y. L. Cun, "Browsing through high quality document images with DJVU," presented at the Advances in Digital Libraries Conf., IEEE ADL '98, U.C. Santa Barbara, Apr. 21–24, 1998.
[17] ITU-T Recommendation T.44, "Mixed raster content (MRC) colour mode."
[18] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York: Chapman & Hall, 1997.
[19] H.-M. Hang and J. W. Woods, *Handbook of Visual Communications*. New York: Academic, 1995.
[20] ITU-T Recommendation H.261, "Video codec for audiovisual services at p*64 kbits/sec."
[21] ITU-T Recommendation H.263, "Video coding for low bit rate communication."
[22] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*. New York: Chapman & Hall, 1997.
[23] ITU-T Recommendation H.324, "Line transmissions of non-telephone signals—Terminal for low bit rate multimedia communication."
[24] A. Puri, R. Aravind, and B. G. Haskell, "Adaptive frame/field motion compensated video coding," *Signal Processing Image Commun.*, vol. 1–5, pp. 39–58, Feb. 1993.
[25] V. G. Devereux, "Digital chroma-key," in *IBC 84, Int. Broadcasting Conv. (Proc. 240)*, Brighton, U.K., Sept. 21–25, 1984, pp. 148–152.
[26] U.S. Advanced Television Systems Committee, "Digital television standard for HDTV transmission," ATSC Standard Doc. A/53, Apr. 1995.
[27] H. Schulzrinne, A. Rao, and R. Lanphier, "Real-time streaming protocol (RTSP)," Internet Draft, draft-ietf-mmusic-rtsp-01.txt, Feb. 1997.
[28] H. Schulzeinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 1889, Jan. 1996.
[29] "Resource reservation protocol (RSVP)—Version 1, Functional specification," R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, Internet Draft, draft-ietf-rsvp-spec-14.txt, Nov. 1996.
[30] 30. C. Huitema, *IPv6: The New Internet Protocol*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[31] M. R. Civanlar, G. L. Cash, and B. G. Haskell, "FusionNet: Joining the internet and phone networks for multimedia applications," in *ACM Multimedia Proc.*, Boston, MA, Nov. 1996.
[32] Special Issue on MPEG-4, *Signal Processing: Image Commun.*, vol. 9, May 1997 and vol. 10, July 1997.
[33] International Organization for Standardization, Final Committee Draft ISO/IEC 14496-2, "Coding of audio-visual objects: Systems."
[34] International Organization for Standardization, Final Committee Draft ISO/IEC 14496-2, "Coding of audio-visual objects: Visual."
[35] G. Privat and I. Le-Hin, "Hardware support for shape decoding from 2D-region-based image representations," in *Multimedia Hardware Architectures 1997, Proc. SPIE*, San Jose, CA, vol. 3021, 1997, pp. 149–159.
[36] S.-L. Wee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Error resilience in video and multiplexing layers for very low bit-rate video coding systems," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1764–1774, Dec. 1997.
[37] MPEG Requirements Group, "MPEG-7: Context and objectives," Doc. ISO/MPEG N1733; also, "Applications for MPEG-7," Doc. ISO/MPEG N1735, MPEG Stockholm Meeting, July 1997.

**Barry G. Haskell** (S'65–M'68–SM'76–F'87) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Berkeley in 1964, 1965, and 1968, respectively.

From 1964 to 1968 he was a Research Assistant in the University of California Electronics Research Laboratory, with one summer being spent at the Lawrence Livermore Laboratory. From 1968 to 1996 he was at AT&T Bell Laboratories, Holmdel, NJ. Since 1996 he has been at AT&T Labs, Middletown, NJ, where he is presently Division Manager of the Image Processing and Software Technology Research Department. He has also served as Adjunct Professor of Electrical Engineering at Rutgers University, City College of New York and Columbia University.

Since 1984, Dr. Haskell has been very active in the establishment of International Video Communications Standards. These include International Telecommunications Union—Telecommunications Sector (ITU-T) for Video Conferencing Standards (H-series), ISO Joint Photographic Experts Group (JPEG) for still images, ISO Joint Bilevel Image Group (JBIG) for documents and ISO Moving Picture Experts Group (MPEG) for Digital Television. His research interests include digital transmission and coding of images, videotelephone, satellite television transmission, medical imaging as well as most other applications of digital image processing. He has published over 60 papers on these subjects and has over 40 patents either granted or pending. He is also coauthor of the books: *Image Transmission Techniques*, Academic Press, 1979; *Digital Pictures—Representation and Compression*, Plenum Press, 1988; *Digital Pictures—Representation, Compression and Standards*, Plenum Press 1995 and *Digital Video—An Introduction to MPEG-2*, Chapman and Hall, 1997.

In 1997 Dr. Haskell won (with Arun Netravali) Japan's prestigious C&C (Computer & Communications) Prize for his research in video data compression. In 1998 he received the Outstanding Alumnus Award from the University of California, Berkeley Department of Electrical Engineering and Computer Science. He is a member of Phi Beta Kappa and Sigma Xi.

**Paul G. Howard** received the B.S. degree in computer science from M.I.T. in 1977, and the M.S. and Ph.D. degrees from Brown University in 1989 and 1993, respectively.

He was briefly a Research Associate at Duke University before joining AT&T Labs (then known as AT&T Bell Laboratories) in 1993. He is a Principal Technical Staff Member at AT&T Labs-Research. He is a member of JBIG, the ISO Joint Bi-level Image Experts Group, and an editor of the emerging JBIG2 standard. His research interests are in data compression, including coding, still image modeling, and text modeling.

**Yann A. LeCun** (S'87–M'87) received the Diplome d'Ingenieur from the Ecole Superieure d'Ingenieur en Electrotechnique et Electronique, Paris in 1983, and the Ph.D. degree in computer science from the Université Pierre et Marie Curie, Paris, in 1987.

He then joined the Department of Computer Science at the University of Toronto as a Research Associate. In 1988, he joined the Adaptive Systems Research Department at AT&T Bell Laboratories in Holmdel, NJ, where he worked among other things on neural networks, machine learning, and handwriting recognition. In 1996, he became head of the Image Processing Services Research Department at AT&T Labs-Research. He has published over 70 technical papers and book chapters on neural networks, machine learning, pattern recognition, handwriting recognition, document understanding, image processing, image compression, VLSI design, and information theory. In addition to the above topics, his current interests include video-based user interfaces, document image compression, and content-based indexing of multimedia material.

Dr. LeCun is serving on the board of the *Machine Learning Journal*, and has served as Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is general chair of the "Machines that Learn" workshop. He has served as program co-chair of IJCNN 89, INNC 90, NIPS 90, 94, and 95. He is a member of the IEEE Neural Network for Signal Processing Technical Committee.

**Atul Puri** (S'87–M'85) received the B.S. degree in electrical engineering in 1980, the M.S. degree in electrical engineering from the City College of New York in 1982, and the Ph.D. degree, also in electrical engineering, from the City University of New York in 1988.

While working on his dissertation, he was a Consultant in Visual Communications Research Department of Bell labs and gained experience in developing algorithms, software and hardware for video communications. In 1988 he joined the same department at Bell labs as a Member of Technical staff. Since 1996, he has been a Principal Member of Technical Staff in Image Processing Research Department of AT&T Labs, Red Bank, NJ. He has represented AT&T at the Moving Pictures Experts Group Standard for past ten years and has actively contributed toward development of the MPEG-1, the MPEG-2, and the MPEG-4 audio–visual coding standards. Currently he is participating in Video and Systems part of the MPEG-4 standard and is one of its technical editors. He has been involved in research in video coding algorithms for a number of diverse applications such as videoconferencing, video on Digital Storage Media, HDTV, and 3D-TV. His current research interests are in the area of flexible multimedia systems and services for web/internet. He holds over 14 patents and has applied for another 8 patents. He has published over 30 technical papers in conferences and journals, including several invited papers. He is a coauthor of a book entitled *Digital Video: An Introduction to MPEG-2*. He is currently coediting a book on multimedia systems.

Dr. Puri has been the recipient of exceptional contribution and individual performance merit awards of AT&T. Furthermore, he has also received awards from the AT&T Communications Services and AT&T Technical Journal. He is a member of the IEEE Communication and Signal Processing societies and was recently appointed as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

**Jörn Ostermann** (S'86–M'88) studied electrical engineering and communications engineering at the University of Hannover and Imperial College London, respectively. He received the Dipl.-Ing. and Dr.-Ing. degrees from the University of Hannover in 1988 and 1994, respectively. From 1988 till 1994, he worked as a Research Assistant at the Institut für Theoretische Nachrichtentechnik conducting research in low bit-rate and object-based analysis-synthesis video coding. 1994 and 1995 he worked in the Visual Communications Research Department at Bell Labs. Since 1996, he is with Image Processing and Technology Research within AT&T Labs-Research.

From 1993 to 1994, he chaired the European COST 211 sim group coordinating research in low bitrate video coding. Within MPEG-4, he organized the evaluation of video tools to start defining the standard. Currently, he chairs the Adhoc Group on Coding of Arbitrarily-shaped Objects in MPEG-4 Video.

His current research interests are video coding, computer vision, 3D modeling, face animation, coarticulation of acoustic and visual speech.

**M. Reha Civanlar** (S'84–M'84) received the B.S. and M.S. degrees from Middle East Technical University, Turkey, and the Ph.D. degree from North Carolina State University, all in electrical engineering, in 1979, 1981, and 1984, respectively.

From 1984 to 1987, he was a Researcher in the Center for Communications and Signal Processing in NCSU where he worked on image processing, particularly restoration and reconstruction, image coding for low bit rate transmission, and data communications systems. In 1988, He joined Pixel Machines Department of AT&T Bell Laboratories, where he worked on parallel architectures and algorithms for image and volume processing and scientific visualization. Since 1991, he is a Member of Visual Communications Research Department of AT&T Bell Laboratories (renamed AT&T Laboratories-Research in 1996) working on various aspects of video communications. His current research interests include packet video systems, networked video and multimedia applications, video coding and digital data transmissions.

Dr. Civanlar is a Fulbright scholar and a member of Sigma Xi. He is an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS and a member of IMDSP and MMSP technical committees of the Signal Processing Society of IEEE. He has numerous publications and 16 patents either granted or pending. He is a recipient of the 1985 Senior Award of the ASSP Society of IEEE.

**Lawrence Rabiner** (S'62–M'67–SM'75–F'76) was born in Brooklyn, NY, on September 28, 1943. He received the S.B. and S.M. degrees simultaneously in June 1964 and the Ph.D. degree in electrical engineering in June 1967, all from the Massachusetts Institute of Technology, Cambridge.

From 1962 through 1964, he participated in the cooperative program in Electrical Engineering at AT&T Bell Laboratories, Whippany and Murray Hill, NJ. During this period he worked on digital circuitry, military communications problems, and problems in binaural hearing. He joined AT&T Bell Labs in 1967 as a Member of the Technical Staff. He was promoted to Supervisor in 1972, Department Head in 1985, Director in 1990, and Functional Vice President in 1995. His research focused primarily on problems in speech processing and digital signal processing. Presently he is Speech and Image Processing Services Research Vice President at AT&T Labs, and is engaged in managing research on speech and image processing, and the associated hardware and software systems which support services based on these technologies. He is coauthor of the books *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), *Multirate Digital Signal Processing* (Prentice-Hall, 1983), and *Fundamentals of Speech Recognition* (Prentice-Hall, 1993).

Dr. Rabiner is a member of Eta Kappa Nu, Sigma Xi, Tau Beta Pi, the National Academy of Engineering, the National Academy of Sciences, and a Fellow of the Acoustical Society of America, Bell Laboratories, and AT&T. He is a former President of the IEEE Acoustics, Speech, and Signal Processing Society, a former Vice-President of the Acoustical Society of America, a former editor of the ASSP Transactions, and a former member of the IEEE Proceedings Editorial Board.

**Leon Bottou** received the Diplome degree from Ecole Polytechnique, Paris in 1987, the Magistére en Mathematiques Fondamentales et Appliquees et Informatiques degree from Ecole Normale Superieure, Paris in 1988, and the Ph.D. degree in computer science from Universite de Paris-Sud in 1991, during which he worked on speech recognition and proposed a framework for stochastic gradient learning and global training.

He then joined the Adaptive Systems Research Department at AT&T Bell Laboratories where he worked on neural networks, statistical learning theory, and local learning algorithms. He returned to France in 1992 as a Research Engineer at ONERA. He then became Chairman of Neuristique S.A. He returned to AT&T Bell Laboratories in 1995 where he worked on graph transformer networks for optical character recognition. He is now a member of the Image Processing Services Research Department at AT&T Labs-Research. Besides learning algorithms, his current interests include arithmetic coding, image compression and indexing.

**Patrick Haffner** graduated from Ecole Polytechnique, Paris, France in 1987 and from Ecole Nationale Superieure des Telecommunications (ENST), Paris, France in 1989. He received the Ph.D. degree in speech and signal processing from ENST in 1994.

From 1989 to 1995, as a Research Scientist for CNET/France-Telecom in Lannion, France, he developed connectionist learning algorithms for telephone speech recognition. In 1995, he joined AT&T Bell Laboratories. Since 1997, he has been with the Image processing Services Research Department at AT&T Labs-Research. His research interests include statistical and connectionist models for sequence recognition, machine learning, speech and image recognition, and information theory.