

Video Compression

Yu-Han Chen and Liang-Gee Chen

Abstract In this chapter, we show the demands of video compression and introduce video coding systems with state-of-the-art signal processing techniques. In the first section, we show the evolution of video coding standards. The coding standards are developed to overcome the problems of limited storage capacity and limited communication bandwidth for video applications. In the second section, the basic components of a video coding system are introduced. The redundant information in a video sequence is explored and removed to achieve data compression. In the third section, we will introduce several emergent video applications (including High Definition TeleVision (HDTV), streaming, surveillance, and multiview videos) and the corresponding video coding systems. People will not stop pursuing move vivid video services. Video coding systems with better coding performance and visual quality will be continuously developed in the future.

1 Evolution of Video Coding Standards

Digital video compression techniques play an important role to enable video applications in our daily life. The evolution of video coding standards is shown in Fig. 1. One demand of video compression is storage. Without compression, a raw video in the YCbYr4:2:0 color format [12], CIF (352×288) spatial resolution, and 30 Frame Per Second (FPS) temporal resolution generates about 36.5 Mbps of data rate. For a 700 MB CD-ROM, only 153 seconds of the CIF video can be stored. In order to store 1 h video in a CD-ROM, 25 times of compression ratio is required to achieve 1.5 Mbps of data rate. In the early 1990s, Motion Picture Experts Group (MPEG) of International Standard Organization/International Electrotechnical Commission

Y.-H. Chen • L.-G. Chen (✉)

Graduate Institute of Electronics Engineering and Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C

e-mail: doliama@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw

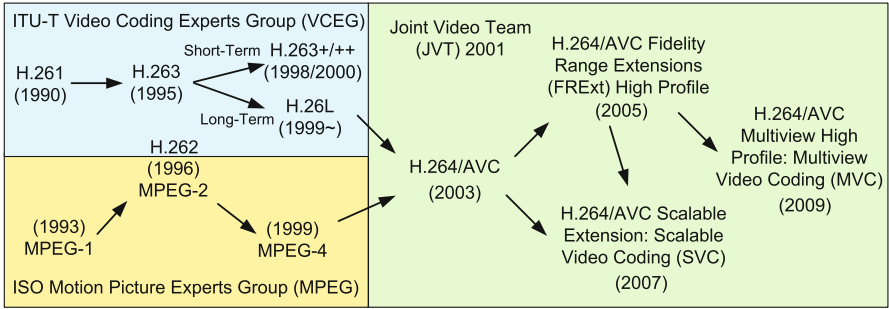


Fig. 1 Evolution of video coding standards

(ISO/IEC) started to develop multimedia coding standards for storage of audio and video. MPEG-1 [2] is the first successful standard which is applied to the VCD industry. MPEG-2 [4] (a.k.a H.262) is another successful story in the DVD industry. Without compression, a raw video in the YCbCr4:2:0 color format, D1 (720×480) spatial resolution, and 30 FPS temporal resolution generates 124.4 Mbps of data rate. For a 4.7 GB DVD-ROM, only 310 s of the D1 video can be stored. MPEG-2 is developed to provide about 2 h D1 video in a DVD-ROM with around 5 Mbps of data rate.

Video Coding Expert Group (VCEG) of International Telecommunication Union (ITU) also developed a series of coding standards, like H.261 [1] and H.263 [3], targeting at realtime video applications through communication network, e.g. video conferences and video phones. H.261 was designed for data rates which are multiples of 64 kbps. It is also referred as $p \times 64$ kbps (p is in the range of 1–30). These data rates are suitable for ISDN (Integrated Service Digital Network) lines. On the other hand, H.263 was designed for data rates less than 64 kbps. Its targeting applications are multimedia terminals providing low bitrate visual telephone services over PSNT (Public Switched Telephone Network). Later, ISO/IEC developed MPEG-4 [5] to extend video applications to mobile devices with wireless channels. Realtime video transmission through complicated communication network is more challenging than storage because of lower data rates and unpredicted network conditions. To provide a video phone service in the YCbCr4:2:0 color format, QCIF (176 × 144) spatial resolution, and 10 FPS temporal resolution under a 64 kbps channel, about 50 times of compression ratio is required. The video coder needs to maintain acceptable video quality under very limited channel bandwidth. In addition, rate control, i.e. adaptively adjusting the data rate according to channel conditions, and error concealment, i.e. recovering corrupted videos induced by packet loss, is also crucial in these environments.

In 2001, experts from ITU-T VCEG and ISO/IEC MPEG formed the Joint Video Team (JVT) to develop a video coding standard called H.264/AVC (Advanced Video Coding) [6]. The first version of H.264/AVC is finalized in 2003. The coding standard is famous with its outstanding coding performance. Compared with MPEG-4, H.263, and MPEG-2, H.264/AVC can achieve 39%, 49%, and 64%

of bit-rate saving under the same visual quality, respectively [11]. The advantage makes H.264/AVC potential for wide range of applications. In recent years, new profiles were developed to target at the emergent applications like HDTV, streaming, surveillance, and multiview videos. The above applications and the corresponding video coding systems are introduced in Sect. 3.

2 Basic Components of Video Coding Systems

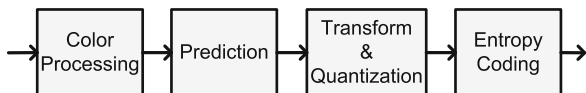
In this section, we introduce the basic components of a video coding system as shown in Fig. 2. In the evolution of various video coding standards, every new standard tries to improve its coding efficiency by further exploring all kinds of redundancy. In general, the redundancy in a video sequence can be categorized into perceptual, temporal, spatial, and statistical redundancy. Perceptual redundancy is the detailed information that human cannot perceive well. Removal of perceptually redundant information will not lead to severe perceptual quality degradation. Temporal and spatial redundancy comes from the pixels that are similar to the neighboring pixels temporally or spatially. We can predict the current pixel with the neighboring pixels and encode the different value between them. The different value is close to zero with high probability and good to be encoded with less bits by entropy coding. Statistical redundancy comes from the information occurred with high probability if we encode all information with the same number of bits. Short codewords (with fewer bits) for frequently occurred symbols and long codewords for infrequently occurred ones can reduce the total data size of the bitstream.

2.1 Color Processing

The first functional block is color processing. Color processing includes color transform and chroma (chrominance) subsampling [12]. Image data from sensors are usually in *RGB* (Red, Green, and Blue) color format. Each pixel in an image with the *RGB* format is first transformed to another color space with one luma (luminance) component and 2 chroma components. The most frequently used color space is *YCbCr* and defined as

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.169R - 0.331G + 0.500B \\ Cr &= 0.500R - 0.419G - 0.081B \end{aligned} \quad (1)$$

Fig. 2 Basic components of a video coding system



Y is the luma component. Cb and Cr are the chroma components correlated to blue color difference ($Y - B$) and red color difference ($Y - R$), respectively. All the luma components forms a grayscale image. Human visual system is less sensitive to the chroma parts in an image. If we sub-sample the chroma parts by 2 horizontally and vertically (e.g. average the neighboring 2×2 chroma pixels), there is almost no perceptual quality degradation. With removal of this kind of perceptual redundancy by chroma subsampling, information is reduced by 2 (from $1(Y) + 1(Cb) + 1(Cr) = 3$ to $1(Y) + \frac{1}{4}(Cb) + \frac{1}{4}(Cr) = 1.5$). Chroma subsampling is widely adopted in video coding standards. However, in emergent HDTV applications, people try to reserve all the color information to provide more vivid videos.

2.2 Prediction

The second functional block is prediction. Prediction is usually the most computation-intensive part in current video coding standards. This unit tries to find the similarity inside a video sequence. Prediction techniques can be categorized into temporal prediction and spatial prediction. Temporal prediction explores the temporal similarity between consecutive frames. Without scene change, the objects in a video are almost the same. The consecutive images are only a little different due to object movement. Therefore, we can predict the current frame with the previous frame well. Temporal prediction is also called inter prediction or inter-frame prediction. Spatial prediction explores the spatial similarity between neighboring pixels. For a region with a smooth texture, neighboring pixels are very similar. Therefore, each pixel can be well predicted by the surrounding ones. Spatial prediction is also called intra prediction or intra-frame prediction.

Simple examples of prediction are shown in Fig. 3. With prediction, each pixel in the current frame is subtracted from its corresponding predictor. For temporal prediction in Fig. 3b, the predictor for each pixel in the current frame is the pixel at the same location of the previous frame as shown in Eq. (2).

$$Predictor_{x, y, t} = Pixel_{x, y, t-1} \quad (2)$$

$Pixel_{x, y, t}$ and $Predictor_{x, y, t}$ are respectively the values of original and predicted pixels in the coordinate (x, y) of the frame at time t . The coordinate $(0, 0)$ is set as the upper-left corner of an image. For spatial prediction in Fig. 3c, the predictor is the average of the upper and left pixels as shown in Eq. (3).

$$Predictor_{x, y, t} = \frac{Pixel_{x-1, y, t} + Pixel_{x, y-1, t}}{2} \quad (3)$$

As we can see, most of the pixel difference values (also called residues) after prediction are close to zero. Spatial prediction is better for videos with less texture like the *Foreman* sequence. On the other hand, temporal prediction is better for videos with less motion like the *Weather* sequence. If prediction is accurate, the residues are almost zeros. In this condition, the entropy of the image is low and thus entropy coding can achieve good coding performance.

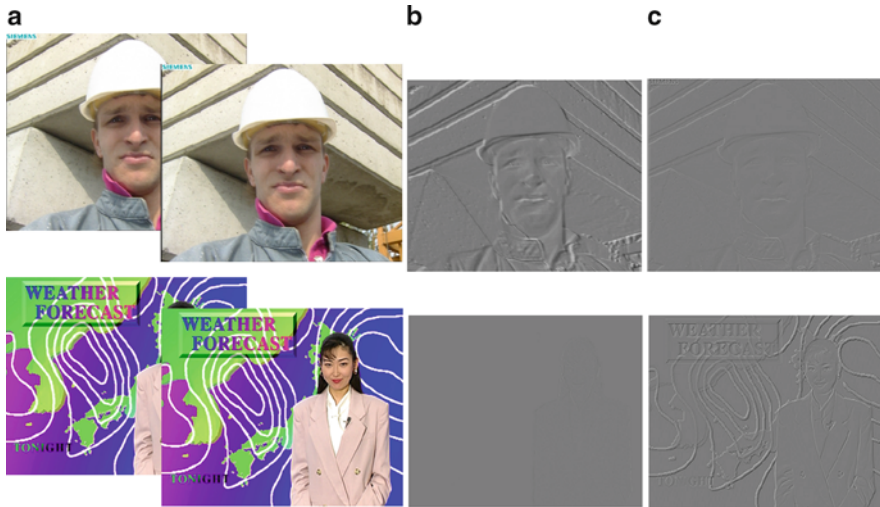
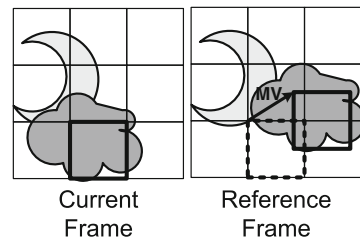


Fig. 3 Illustration of prediction. After prediction, the whiter/blacker pixels mean more positive/negative residues. The gray pixels are residues close to zero. (a) Video sequences “Foreman” (above) and “Weather” (below); (b) Temporal prediction; (c) Spatial prediction

Fig. 4 Illustration of motion estimation



2.2.1 Temporal Prediction

The above temporal prediction scheme is simple but not accurate, especially for videos with moving objects. The most frequently used technique for temporal prediction is Motion Estimation (ME). Illustration of ME is shown in Fig. 4. In most of the video coding standards, an image is first partitioned into MacroBlocks (MBs) which are usually with the size of 16×16 pixels. Each MB in the current frame searches for a best matching block in the reference frame. Then, the pixel values of the current MB is subtracted by the corresponding pixel values of the best matching block to generate the residual block. The process is called motion compensation. The Motion Vector (MV) information is also needed to code. In the decoder side, MV information is required to get the prediction data in the reference frame. Then, the prediction data are added with the residual data to generate the reconstructed image. There are many kinds of motion estimation algorithms, please read [10] for the details.

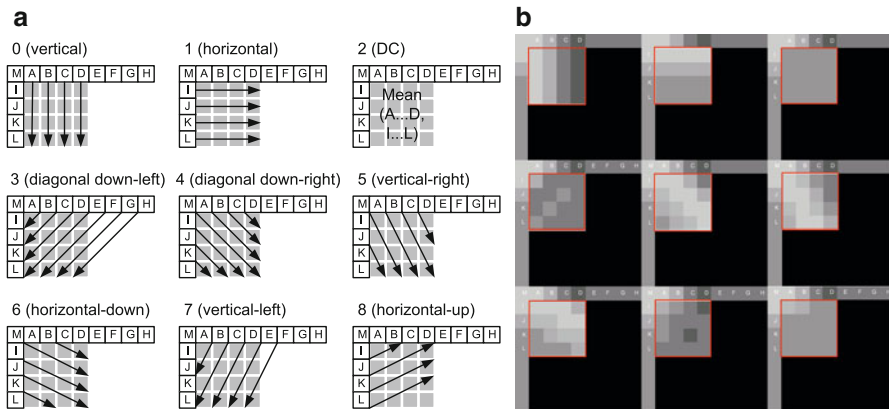


Fig. 5 Illustration of nine kinds of Intra 4×4 modes for luma blocks in H.264/AVC (a) 13 boundary pixels and the predictive directions for different modes; (b) Examples of the prediction data with real images

2.2.2 Spatial Prediction

Here we introduce a spatial prediction technique called intra prediction which is adopted in H.264/AVC. Intra prediction explores the spatial similarity in a region with a regular texture. It utilizes the boundary pixels in the neighboring pre-coded blocks to predict the current block. Illustration of Intra 4×4 modes in H.264/AVC is shown in Fig. 5. Each MB is partitioned into 16 4×4 blocks. Each 4×4 block is predicted by the 13 left and top boundary pixels. There are 9 Intra 4×4 modes with different predictive directions from the neighboring pixels. The mode with the prediction data most similar to the current block will be chosen. There is another scheme call Intra 16×16 as shown in Fig. 6. In general, Intra $4 \times 4/16 \times 16$ modes are suitable for regions with more/less detailed texture. Please refer to [25] for the detailed introduction of intra prediction in H.264/AVC.

2.2.3 Coding Structure

In a video coding system, a video sequence is partitioned into Group-Of-Pictures (GOPs). Each GOP contains three kinds of frames—I-frames (Intra Coded Frames), P-frames (Predictive Coded Frames), and B-frames (Bidirectionally Predictive Coded Frames). For I-frames, only spatial prediction is supported. For P-frames and B-frames, both spatial and temporal prediction is supported. P-frames can only use the preceding frames for temporal prediction (a.k.a forward prediction). B-fames can use both the preceding and the following (backward prediction) frames for temporal prediction. In general, B-frames can achieve the best coding performance because the most data are utilized for prediction. However, B-frames require more

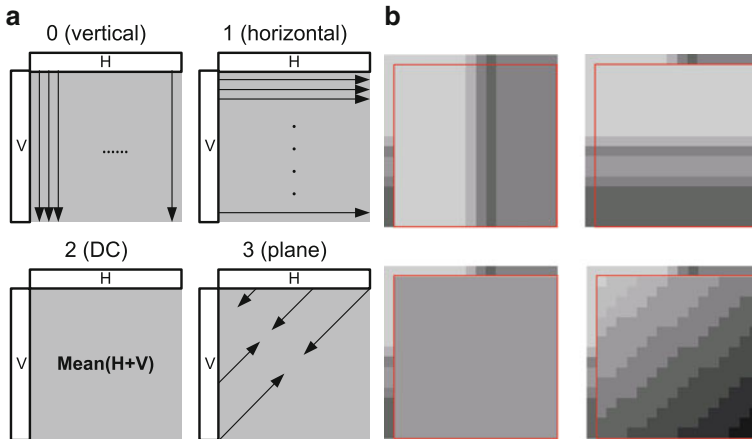


Fig. 6 Illustration of four Intra 16×16 modes in H.264/AVC (a) The predictive directions; (b) Examples of the prediction data with real images

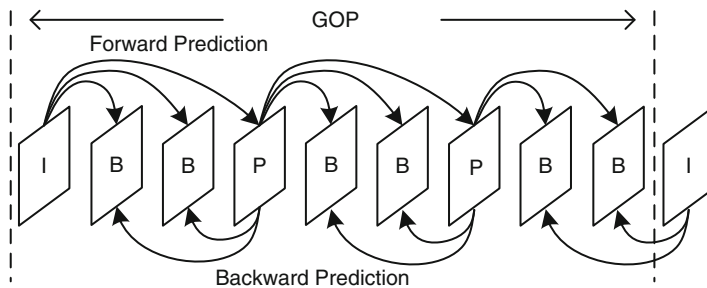


Fig. 7 Illustration of Groups-Of-Picture (GOP)

memory to store the reference frames and more computation to find the best prediction mode. In a GOP, the first frame is always an I-frame. We can use two parameters N and M to specify the coding structure of a GOP. N is the GOP size. It represents the distance between two I-frames. M represents the distance between two anchor frames (I-frames or P-frames). It also means there are $(M - 1)$ B-frames between two anchor frames. In Fig. 7, a GOP with parameters $N=9$ and $M=3$ is shown. When a bitstream error occurs, the error will propagate through the temporal prediction paths in a GOP. An I-frame do not require other reference frames to reconstruct the image. Therefore, error propagation will be stopped after I-frames.

2.3 Transform and Quantization

In an image, neighboring pixels are usually correlated. To further remove the spatial redundancy, transformation is performed after prediction to transfer a block of residual pixels to a more compact representation. Discrete Cosine Transform

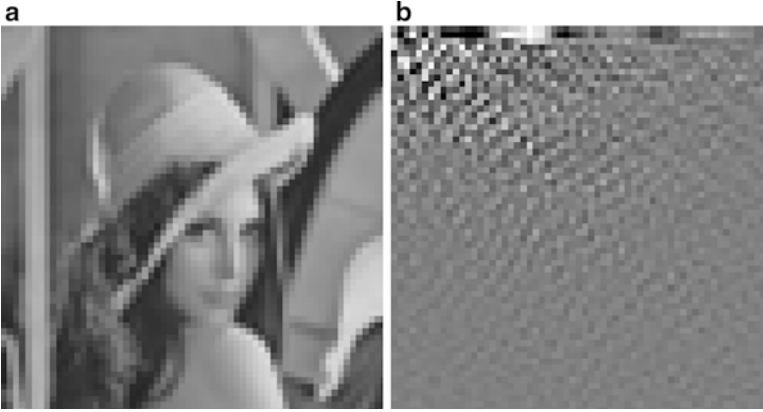


Fig. 8 Illustration of Discrete Cosine Transform (a) The original Lena image; (b) After transform, energy is compacted to low-frequency components (the upper-left side). White and black pixels are with high energy. Gray pixels are with low energy

(DCT) [18] is adopted in most of the video coding standards. The $N \times N$ 2-D DCT transform is defined as Eq. (4).

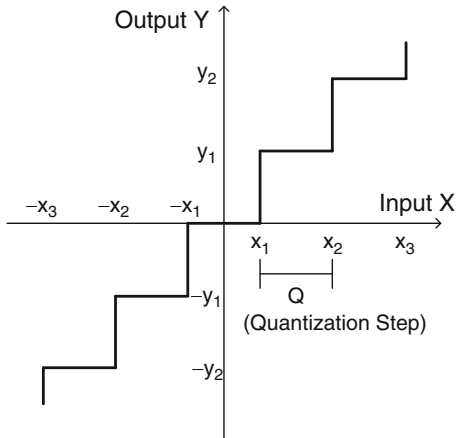
$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{2\Pi(2x+1)u}{4N} \cos \frac{2\Pi(2y+1)v}{4N}$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u, v = 0 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

$F(u, v)$ is the DCT value at the location (u, v) and $f(x, y)$ is the pixel value at the location (x, y) . The first and the second cosine terms are the vertical/horizontal cosine basis function. With DCT, energy will be gathered to the low-frequency parts as shown in Fig. 8. The entropy of a block is reduced and it leads to better coding efficiency in entropy coding.

Quantization [8] is performed after transformation. It discards some of the information in a block and leads to lossy coding. A simple example of uniform quantization is shown in Fig. 9. There are several decision boundaries like x_1 and x_2 . All input value between two decision boundaries are output as the same value. For example, the output value is y_1 if the input value is inside the interval $x_1 < X < x_2$. In this example, y_1 is set as the average value of x_1 and x_2 . Quantization step is the distance between two decision boundaries. In a video coding system, quantization step is adjusted to control the data rate. Larger quantization step leads to lower data rate but poorer visual quality, and vice versa. In addition, human visual system is less sensitive to high-frequency components. High-frequency components are usually quantized more heavily with less visual quality degradation. This is another example to remove perceptual redundancy for data compression.

Fig. 9 Illustration of one-to-one input to output mapping of uniform quantization



2.4 Entropy Coding

The last step of video coding is entropy coding. It translates symbols into bitstream. Symbols can be the transformed and quantized residual values, the MVs, or the prediction modes. From information theory, we know that the average length of the code is bounded by the entropy of the information source. It is why prediction, transformation, and quantization techniques are adopted in video coding systems to reduce the entropy of the encoded data. The entropy H is defined as

$$H = -\sum P(A_i) \log_2 P(A_i) \tag{5}$$

A_i is the i -th symbol and $P(A_i)$ is its probability. In addition, the optimal code length of a symbol is its own entropy $-\log_2 P$. Therefore, we have to build a probability model for each kind of symbol. For frequently appeared symbols, shorter codewords are assigned. As a result, the final bitstream is shorter.

Huffman coding is a famous entropy coding scheme. An example of Huffman coding is shown in Fig. 10. At the start, the symbols are sorted by their probability values from low to high. At each step, we choose the two symbols with lowest probability. In this example, symbol A and symbol D are chosen first and combined to a new symbol $\{A, D\}$ with probability 0.22. The symbol A with lower probability is put in the left side and assigned a bit 0. The symbol D with higher probability is put in the right side and assigned a bit 1. Then, symbol $\{A, D\}$ and symbol E are chosen to form a symbol $\{\{A, D\}, E\}$ with probability 0.45. Symbol $\{A, D\}$ is assigned a bit 0 and symbol E is assigned a bit 1. With the process, we can finally build a codeword tree as shown in Fig. 10b. The final codeword of each symbol is assigned by traversing the tree from the root to the leaf. For example, the codeword of symbol D is 001. To encode $BABCE$, the bitstream is 11000111001. For Huffman coding, the codeword length is always an integer number. Therefore,

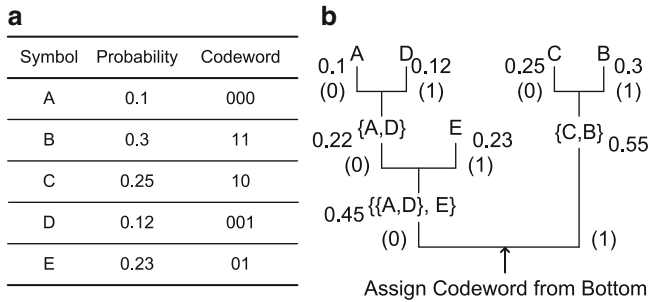


Fig. 10 Illustration of Huffman coding (a) List of the symbols, probability values, and the final codeword; (b) The process of Huffman code generation

Huffman coding cannot achieve the best coding performance in many conditions. For example, the optimal codeword length for symbol *B* is 1.737 bits according to Eq. (5) but not 2 bits assigned by Huffman coding as shown in Fig. 10a.

Arithmetic coding is another famous entropy coding scheme. Unlike the Huffman coding, arithmetic coding does not assign codeword to each symbol but represents the whole data into a single fractional number n between 0 and 1. Arithmetic coding divides the solution space into intervals according to the probability values of the symbols as shown in Fig. 11a. The process of arithmetic coding for *BABCE* is shown in Fig. 11b. At the start, the current interval is $[0.00, 1.00)$. The first symbol is *B*. The sub-interval $[0.10, 0.40)$ representing *B* becomes the next current interval. Then, the current interval becomes $[0.10, 0.13)$ for symbol *A* and go on. At last, the interval becomes $[0.1083325, 0.10885)$. The lower and upper bounds of the final interval are $0.00011011101\dots$ and $0.00011011110\dots$ with binary representations. The final chosen fractional number to encode *BABCE* is 0.0001101111 (0.10839844). The encoded bitstream with arithmetic coding is 0001101111 .

3 Emergent Video Applications and Corresponding Coding Systems

3.1 HDTV Applications and H.264/AVC

With the progress in video sensors, storage devices, communication systems, and display devices, the style of multimedia applications changes. The prevalence of full HD (High Definition) camera coders and digital full HD TVs shows the demands of video services with high resolution and high quality. In order to provide better visual quality, the video coding standards for HDTV applications should be optimized for coding performance. Video coding standards with better coding performance

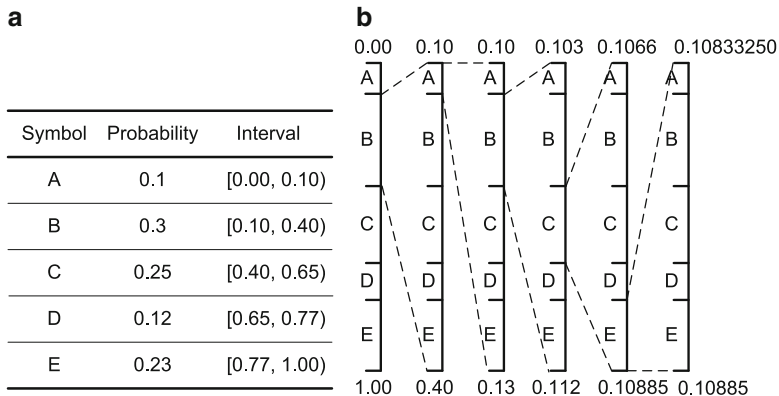


Fig. 11 Illustration of arithmetic coding (a) List of the symbols, probability values, and the partitioned intervals; (b) The process of arithmetic coding for *BABCE*

can provide better visual quality under the same transmitted data rate. In 2003, H.264/AVC was standardized by JVT to combine the state-of-the-art coding tools for optimization of coding performance. Then, H.264/AVC High Profile [13,21] was developed for the incoming HDTV applications in 2005. H.264/AVC High Profile has been adopted into HD-DVD and Blu-ray Disc. Compared to MPEG-2 which is adopted in DVD, H.264/AVC High Profile can provide comparable subjective visual quality with only one-third of data rate [21].

Here, we show some advanced coding tools of H.264/AVC. For inter (temporal) prediction, Multiple Reference Frames (MRF) [23] are supported. As shown in Fig. 12, more than one prior reconstructed frames can be used for ME. Each MB can choose its own best matching block in different reference frames. In addition, B-Frame is also adopted. That is to say not only forward prediction but also backward prediction can utilize more than one reference frames. The above tools can improve inter prediction accuracy when some parts of the current image are unseen in the first preceding image (e.g. uncovered background). Variable Block Sizes is another useful technique. It is helpful for the MBs which include multiple objects moving in different directions. Each MB is first split into four kinds of partitions— 16×16 , 16×8 , 8×16 , and 8×8 . If the 8×8 partition is selected, each block is further split into 4 kinds of sub-partitions— 8×8 , 8×4 , 4×8 , and 4×4 . With this scheme, each partition can have its own MV for the best matching block. Smaller partitions can get lower prediction error but need higher bitrate to transmitted MV information. Therefore, there is trade-off between rate and distortion for mode decision. Please refer to [22, 24] for the issue of rate-distortion optimization. The last coding tool for inter prediction is Fractional Motion Estimation (FME). Sometimes, the moving distance of an object is not on the integer-pixel grid of a frame. Therefore, H.264/AVC provides FME up to quarter-pixel resolution. The half-pixel and quarter-pixel prediction data are interpolated from the neighboring integer-pixels in the reference frame. Please refer to [25] for the detailed.

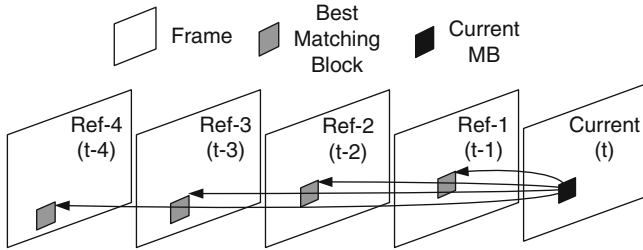


Fig. 12 Illustration of Multiple Reference Frame (MRF) scheme

For intra (spatial) prediction, there are two modes defined in H.264/AVC Baseline Profile as introduced in Sect. 2.2. Intra 4×4 mode is suitable for regions with detailed texture. On the contrary, Intra 16×16 mode is suitable for regions with smooth texture. In H.264/AVC High Profile, another intra prediction mode Intra 8×8 is added for the texture with the medium size. Each MB is partitioned into 8×8 blocks. There are also nine modes provided and the predictive directions are the same with those of Intra 4×4 modes.

For entropy coding, H.264/AVC adopts Context-based Adaptive Variable Length Coding (CAVLC) and Context-based Adaptive Binary Arithmetic Coding (CABAC) [14]. Compared to the previous standards, CAVLC and CABAC adaptively adjust probability models according to the context information. Accurate probability models lead to better coding performance for entropy coding. In addition, CABAC can save 10% bitrate compared to CAVLC. It is because arithmetic coding can achieve better coding performance than variable length coding. There are still some other coding tools in H.264/AVC High Profile like Adaptive Transform Block-size and Perceptual Quantization Scaling Matrices. Please refer to [13,21] for the details.

3.2 Streaming and Surveillance Applications and Scalable Video Coding

There are various of multimedia platforms with different communication channels and display sizes. The variety of multimedia systems makes scalability become important for video standards to support various demands. Scalability of video coding means one encoded bitstream can be partially or fully decoded to generate several kinds of videos. More decoding data contribute higher spatial/temporal resolution or better visual quality. For a streaming service, its clients may be a mobile phone, a PC, or an HD TV. Without scalability, we need to encode a video into different bitstreams for different requirements. It is not efficient for storage. Another key application for scalable video coding is surveillance. Due to the limited storage size, less important data should be removed from the bitstream. The importance of the information in a surveillance video is decayed with time. We

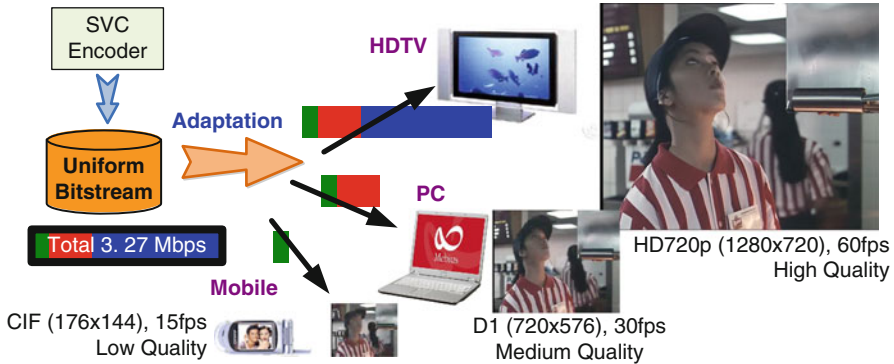


Fig. 13 The concept and framework of Scalable Video Coding

can truncate the bitstream to reduce the resolution or visual quality of the videos in the longer past under scalable video coding rather than just dump them under the conventional video coding systems. Scalability of video coding also means more truncation points are provided for the users. For these requirements, H.264/AVC Scalable Extension (a.k.a Scalable Video Coding, SVC) [19] is established and finalized in 2007. The new video coding framework can be depicted as Fig. 13. An unified scalable bitstream is only encoded once, but it can be adapted to support various multimedia applications with different specifications from mobile phone, personal computer to HDTV.

In general, there are three kinds of video scalability should be included—temporal scalability, spatial scalability, and quality scalability. Temporal scalability is to provide different frame rates (temporal resolution). In SVC, temporal scalability is achieved by Hierarchical B-frame (HB) as shown in Fig. 14. The decoding order of frames in a GOP is from A , B^1 , B^2 , to B^3 . More decoded frames lead to higher frame rate. In fact, HB is first proposed in H.264/AVC High Profile as an optional coding tools to improve coding efficiency because it can explore more temporal redundancy in the whole GOP. This scheme will induce large encoding and decoding delay and thus is more suitable for storage applications.

SVC provides spatial scalability (different frame resolutions) with a multi-layer coding structure as shown in Fig. 15. If the pictures of different layers are independently coded, there is redundancy between different spatial layers. Therefore, three techniques of inter-layer prediction included in SVC. Inter-layer motion prediction utilizes the MB partition and MVs of the base layer to predict the information in the enhancement layer as shown in Fig. 16. Inter-layer residual prediction upsamples the co-located residues of the base layer as the predictors for the residues of the current MB as depicted in Fig. 17. In this figure, the absolute values of the residues are shown. Pixels with black color is close to zero. Inter-layer intra prediction upsamples the co-located intra-coded signals as the prediction data for the enhancement layer.

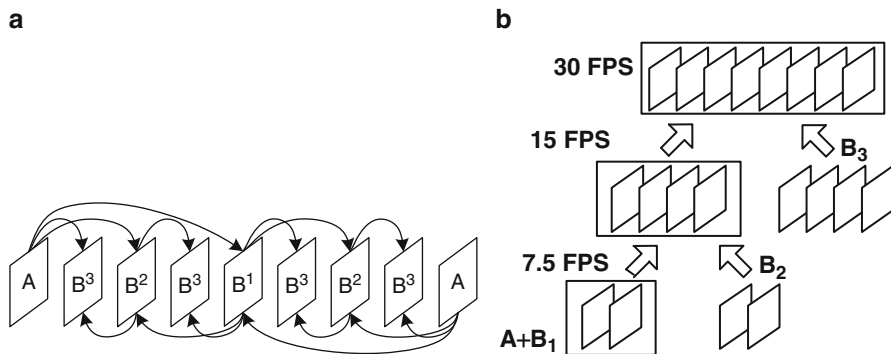


Fig. 14 Illustration of Hierarchical B-frame technique. *A* is the anchor frame (I-frame or P-frame). The superscript number of a B-frame represents the encoding order. (a) Three-level Hierarchical B-frame (HB) scheme which GOP size is 8 frames. The B^1 frame utilizes the anchor frames as its reference frames. The B^2 frames utilize the anchor and B^1 frames as the reference frames. At last, the reference frames of the B^3 frames are the anchor, B^1 , and B^2 frames. (b) The decomposition of HB for temporal scalability

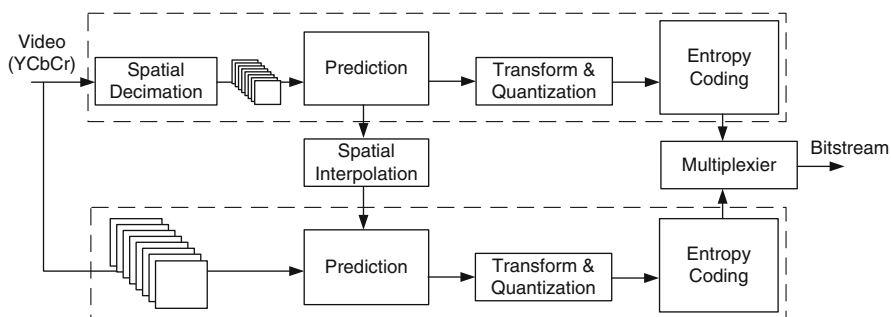


Fig. 15 The video coding flow of multi-layer coding scheme for SVC

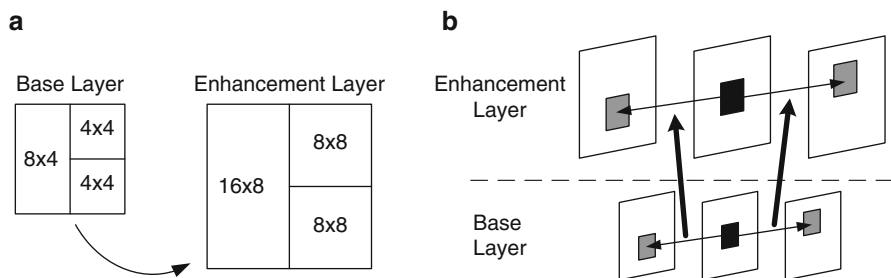
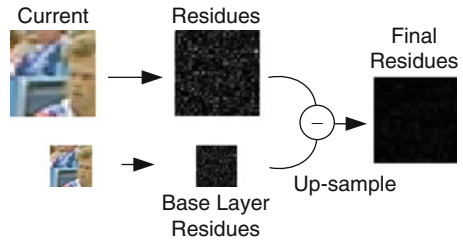


Fig. 16 Inter-layer motion prediction (a) MB partition scaling; (b) motion vector scaling

Fig. 17 Illustration of inter-layer residual prediction



Quality (SNR) scalability is to provide various decoded bit rate truncation points so that users can choose different visual quality. It can be realized by three main strategies in SVC—Coarse Grain Scalability (CGS), Medium Grain Scalability (MGS), Fine Grain Scalability (FGS). CGS is to add a layer with a smaller quantization parameter, like an additional spatial layer with better quality and with the same frame resolution. But CGS can only provide few pre-defined quality points. FGS focuses on providing arbitrary quality levels according to users' channel capacity. Each FGS enhancement layer represents refinement of residual signals. It is designed through multiple scans in the entire frame so that the quality improvement can be distributed to the whole frame. MGS is recently proposed in H.264/AVC Scalable Extension to provide quality scalability between CGS and FGS. Please refer to [20] for the details.

3.3 3D Video Applications and Multiview Video Coding

The TV evolves from the monochrome one to the color one, and then to the HDTV in the past 80 years. People still want to get a more vivid perceptual experience from the TV. Stereo vision is the natural feeling for human beings. Stereo parallax provides us depth perception for scene understanding. In the recent years, three-dimensional TV (3DTV) becomes popular. It is because more and more advanced 3D capturing devices (e.g. camera arrays) [26, 27] and 3D displays [9] are introduced. Multiview video as an extension of 2D video technology is one of the potential 3D applications. At the same time, multiple 2D views are displayed as shown in Fig. 18. Viewers can perceive different stereo videos at different viewpoints to reconstruct the 3D scene. In order to display multiple views at the same time, the required transmitted data are much more than those in the traditional single view video sequence. For example, an 8-view video requires 8 times of bandwidth compared to the single view video. Therefore, an efficient data compression system for multiview video is required. The Multiview Video Coding (MVC) system is shown in Fig. 19.

MPEG-2 Multiview Profile (MVP) [16] is the first developed standard which describes the bitstream conformance of stereoscopic video. Figure 20 illustrates the two-layer coding structure. In the coding structure, the “temporal scalability”



Fig. 18 Multiview video data set is composed of two to several view channels

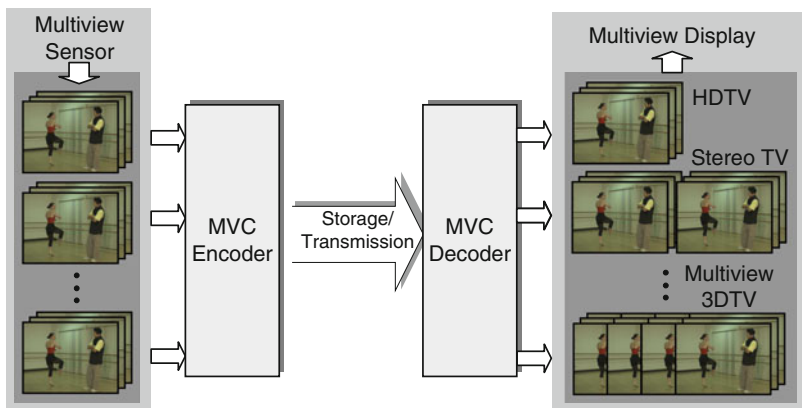
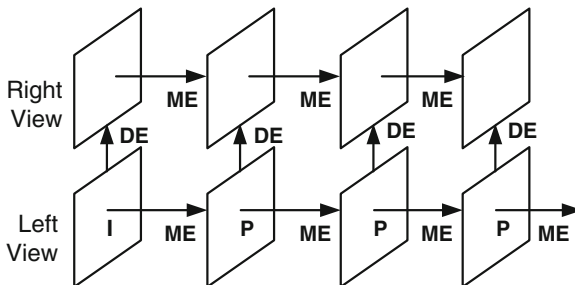


Fig. 19 Overview of an MVC system. Redrawn from [15]

Fig. 20 Two-layer coding structure of MPEG-2 multi-view profile



supported by MPEG-2 Main Profile is utilized in MPEG-2 MVP. That is, a view is encoded as the base layer, and the other one is encoded as the enhancement layer. The frames in the enhancement layer are additionally predicted via Disparity Estimation (DE). The process of DE is similar to ME. The only difference is that DE searches for the best matching block in the neighboring views at the same time slot. On the contrary, ME searches for the best matching block in the preceding or

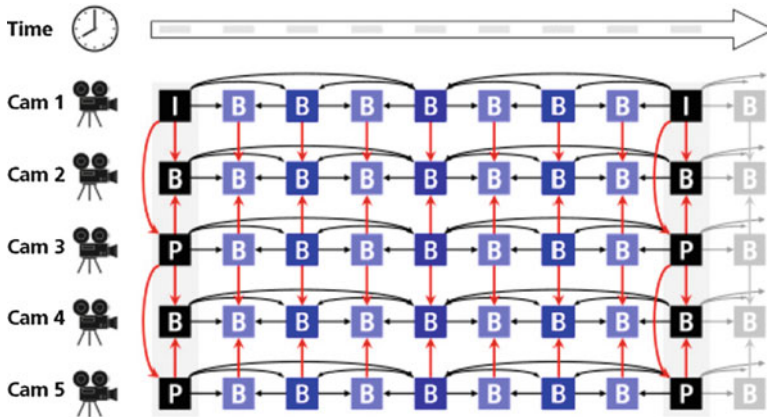


Fig. 21 Example of MVC hierarchical-B coding structure with the extension of inter-view prediction. The *horizontal direction* represents different time slots and the *vertical direction* represents different views. Redrawn from [15]

the following frames in the same view. The concept of MPEG-2 MVP is simple, and the only additional functional block is a processing unit of syntax element in entropy coding. However, MPEG-2 MVP can not provide the good coding efficiency because the prediction tool defined in MPEG-2 is poor compared with the state-of-the-art coding standard such as H.264/AVC [7].

In recent years, MPEG 3D Audio/Video (3DAV) group has worked toward the standardization for MVC [17] which adopts H.264/AVC as the base layer due to its outstanding coding performance. The hybrid ME and DE inter prediction technique is also adopted. To further improve the coding performance, MVC extends the original hierarchical-B coding structure to inter-view prediction for multiview video sequences as shown in Fig. 21. The bitrate can be saved up to 50%, but it depends on the characteristics of multiview video sequences [15].

4 Conclusion

With the improvement of video sensor and display technologies, more and more video applications are emergent. Due to the limited capacity of storage devices and communication channels, video compression plays a key role to drive the applications. H.264/AVC is the most important coding standard in recent years due to its outstanding coding performance. Some other coding standards like Scalable Video Coding (SVC) and Multiview Video Coding (MVC) are developed based on H.264/AVC and add some specific coding tools targeting at their own applications. The demands for video services with better visual quality will keep go on. Therefore, video coding systems for super high resolution, high frame rate, high dynamic range, and large color gamut will also be developed in the near future.

References

1. Video Codec for Audiovisual Services at $p \times 64$ Kbit/s. ITU-T Rec. H.261, ITU-T (1990)
2. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s—Part 2: Video. ISO/IEC 11172-2 (MPEG-1 Video), ISO/IEC JTC 1 (1993)
3. Video Coding for Low Bit Rate Communication. ITU-T Rec. H.263, ITU-T (1995)
4. Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video. ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1 (1996)
5. Coding of Audio-Visual Objects—Part 2: Visual. ISO/IEC 14496-2 (MPEG-4 Visual), ISO/IEC JTC 1 (1999)
6. Advanced Video Coding for Generic Audiovisual Services. ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1 (2003)
7. Chien, S.Y., Yu, S.H., Ding, L.F., Huang, Y.N., Chen, L.G.: Efficient stereo video coding system for immersive teleconference with two-stage hybrid disparity estimation algorithm. In: Proc. of IEEE International Conference on Image Processing (2003)
8. Gray, R.M., Neuhoff, D.L.: Quantization. *IEEE Transactions on Information Theory* **44**(6), 2325–2383 (1998)
9. Holliman, N.: 3d display systems. In: *Handbook of Optoelectronics*, chap. 3. Taylor and Francis (2006)
10. Huang, Y.W., Chen, C.Y., Tsai, C.H., Shen, C.F., Chen, L.G.: Survey on block matching motion estimation algorithms and architectures with new results. *Journal of VLSI Signal Processing* **42**(3), 297–320 (2006)
11. Joch, A., Kossentini, F., Schwarz, H., Wiegand, T., Sullivan, G.J.: Performance comparison of video coding standards using Lagrangian coder control. In: Proc. IEEE International Conference on Image Processing (ICIP), pp. 501–504 (2002)
12. Kerr, D.A.: Chrominance Subsampling in Digital Images. Available: <http://doug.kerr.home.att.net/pumpkin/Subsampling.pdf> (2005)
13. Marpe, D., et al.: H.264/MPEG4-AVC fidelity range extensions : Tools, profiles, performance, and application areas. In: Proc. IEEE International Conference on Image Processing (ICIP), vol. 1, pp. 593–596 (2005)
14. Marpe, D., Schwarz, H., Wiegand, T.: Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7), 620–644 (2003)
15. Merkle, P., abd Karsten Muller, A.S., Wiegand, T.: Efficient prediction structures for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology* **17**(11), 1461–1473 (2007)
16. N1088, I.J.S.W.: Proposed draft amendment No. 3 to 13818-2 (multi-view profile). MPEG-2 (1995)
17. N6501, I.J.: Requirements on multi-view video coding (2004)
18. Rao, K.R., Yip, P.: *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press (1990)
19. Reichel, J., Schwarz, H., Wien, M.: Working Draft 4 of ISO/IEC 14496-10:2005/AMD3 Scalable Video Coding. ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. N7555 (2005)
20. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology* **17**, 1103–1120 (2007)
21. Sullivan, G., Topiwala, P., Luthra, A.: The H.264 advanced video coding standard : Overview and introduction to the fidelity range extensions. In: Proc. SPIE Conference on Applications of Digital Image Processing XXVII (2004)
22. Sullivan, G.J., Wiegand, T.: Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine* **15**(6), 74–90 (1998)
23. Wiegand, T., Girod, B.: *Multi-Frame Motion-Compensated Prediction for Video Transmission*. Kluwer Academic Publishers (2001)

24. Wiegand, T., Schwarz, H., Joch, A., Kossentini, F., Sullivan, G.J.: Rate-constrained coder control and comparison of video coding standards. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7), 688–703 (2003)
25. Wiegand, T., Sullivan, G.J., Bjøntegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7), 560–576 (2003)
26. Wilburn, B.S., Smulski, M., Lee, H.H.K., Horowitz, M.A.: Light field video camera. In: *Proceedings of Media Processors, SPIE ElectronicImaging*, vol. 4674, pp. 29–36 (2002)
27. Zhang, C., Chen, T.: A self-reconfigurable camera array. In: *Eurographics symposium on Rendering* (2004)