

**H.264 DELIVERS TWICE-AS-GOOD COMPRESSION AND ENHANCED QUALITY.**

# Video compression's quantum leap

AS VIDEO GOES DIGITAL, content producers, distributors, and users are all demanding ever-higher quality and ever-larger display screens—in other words, more megapixels per second. Consider a typical TV-broadcast video stream characterized by 24-bit color and 720×480-pixel resolution refreshing at 30 frames/sec. Uncompressed, it would require a bandwidth of greater than 248 Mbps. High-definition TV requires five times the bandwidth of standard-definition TV. Because the carrying capacity of most communication channels cannot keep up with pixel demand, video compression has been the only option, particularly in the high-definition era.

For more than a decade, standards from the ISO MPEG (International Standards Organization Moving Picture Experts Group) and the ITU-T (International Telecommunications Union Telecommunications Committee) have successfully addressed the demand for high compression ratios.

MPEG-2 has been the most successful to date, achieving mass-market acceptance in applications such as DVD players, cable- and satellite-digital TV, and set-top boxes. However, operators are continuously reducing the operating point, affecting image quality, and consumers are becoming increasingly aware of compression artifacts, such as blocking, ringing, and drifting. More recently, MPEG and ITU have developed MPEG-4 (Advanced Simple Profile) and H.263, respectively, but in the face of new market demands, they are ready for a successor.

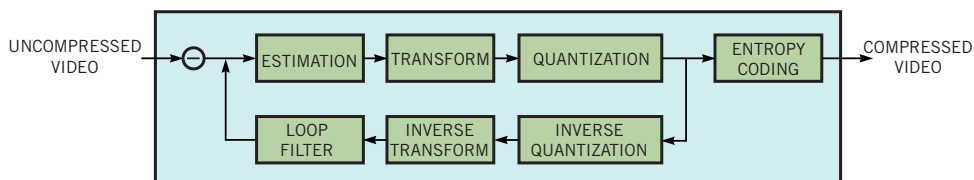
HDTV (high-definition TV) and video over IP (Internet Protocol) using an ADSL (asymmetrical-digital-subscriber-line) connection represent a set of bandwidth-hungry terrestrial-broadcast and wired applications. In the broadcast world, the cost of satellite transmission is increasing. It is becoming increasingly evident that two-times-better compression than MPEG-2 is the most cost-effective way to provide a sufficient number of local channels and to transmit HDTV. The

same arguments are true for cable and even more imperative in Internet-content distribution.

By all accounts, the HD-DVD will take over where DVD leaves off and start a multibillion-dollar market for players as long as video-compression technology keeps pace with bandwidth demands. Meanwhile, IEEE 802.11e WLAN (wireless-LAN) “hot spots” and in-home wireless networks, in which multiple users share bandwidth, present an even more daunting engineering challenge. Engineers will meet that challenge when they adopt an ITU-T- and ISO MPEG-approved standard. H.264/MPEG-4 AVC (Advanced Video Coding) will deliver a twofold improvement in compression ratio and improved quality. As such, it represents the most significant improvement in coding efficiency and quality since MPEG-2/H.262 (Reference 1).

## COMPRESSION BASICS

Compression essentially identifies and eliminates redundancies in a signal and provides instructions for reconstituting the bit stream into a picture when the bits are uncompressed. The basic types of redundancy are *spatial*, *temporal*, *psycho-visual*, and *statistical*. “Spatial redundancy” refers to the correlation between neighboring pixels in, for example, a flat background. “Temporal redundancy” refers to the correlation of a pixel’s position between video frames. Typically, the background of a scene remains static in the absence of camera movement, so that you need not code and decode those pixels for every frame. Psycho-visual redundancy takes advantage of the varying sensitivities of the human visual system. The human eye is much more discriminating regarding changes in luminance than chrominance, for example, so a system with this feature can discard



**Figure 1** H.264 computes the differences between actual incoming video and estimated/transformed video, using either motion estimation or intra-frame estimation. So, only the video and the difference appear in the compressed-video stream.

some color-depth information, and viewers do not recognize the difference. Statistical redundancy uses a more compact representation for elements that frequently recur in a video, thus reducing the overall size of the compressed signal.

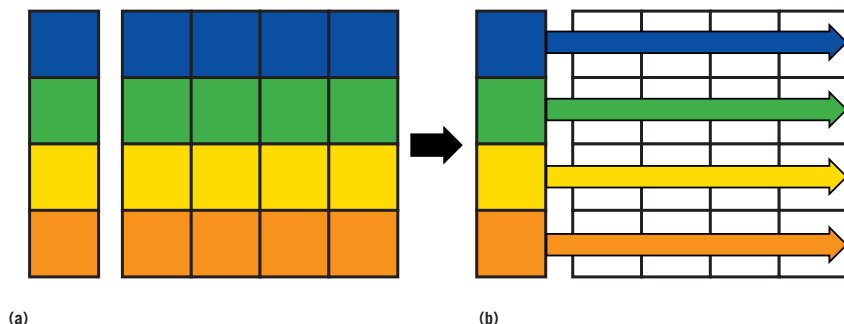
Removing temporal redundancies is responsible for a significant percentage of all the video compression that you can achieve. Although H.264 makes advances in removing temporal redundancies, it is also better across the board, thanks to the adoption of innovative techniques.

### INTO THE FUTURE WITH H.264

Video-compression schemes today follow a common set of interactive operations. First, they segment the video frame into blocks of pixels. Also, the schemes estimate frame-to-frame motion of each block to identify temporal or spatial redundancy, within the frame. In another operation, an algorithmic DCT (discrete cosine transform) decorrelates the motion-compensated data to produce an expression with the lowest number of coefficients, reducing spatial redundancy. The video-compression scheme then quantizes the DCT coefficients based on a psycho-visual redundancy model. Entropy coding then removes statistical redundancy, reducing the average number of bits necessary to represent the compressed video. Coding, or rate, control—also known as mode decision—comes into play to select the most efficient mode of operation. **Figure 1** provides an overview of coding.

### MOTION ESTIMATION

Estimating the movement of blocks of pixels from frame to frame and coding the displacement vector—not the details of the blocks themselves—reduce or eliminate temporal redundancy. To start, the compression scheme divides the video frame into blocks. Whereas MPEG-2 uses only 16×16-pixel motion-compensated blocks, or *macroblocks*, H.264 provides the option of motion compensating 16×16-, 16×8-, 8×16-, 8×8-, 8×4-, 4×8-, or 4×4-pixel blocks within



**Figure 2** Intraframe estimation operates at the pixel-block level and attempts to predict the current block by extrapolating the neighboring pixels from adjacent blocks in a defined set of different directions (a). It then codes the difference between the predicted block and the actual block (b).

each macroblock. The scheme accomplishes motion estimation by searching for a good match for a block from the current frame in a previously coded frame. The resulting coded picture is a *P-frame*.

The estimate may also involve combining pixels resulting from the search of two frames. In this case, the coded picture, or B-frame. (In MPEG-2, these two frames must be one temporally previous frame and one temporally future frame, whereas H.264 generalizes B-frames, thus removing this restriction.) Searching is an important aspect of the process because it must try to ascertain the best match for where the block has moved from one frame to the next.

To substantially improve the process, you can use subpixel motion estimation, which defines fractional pixels. Unlike MPEG-2, which offers half-pixel accuracy, H.264 uses quarter-pixel accuracy for both the horizontal and the vertical components of the motion vectors.

H.264 uses P- and B-frames to detect and code periodic motion. Although B-macroblocks often give better performance than P-macroblocks, using them in a traditional manner delays decoding. This delay occurs because H.264 must decode the future P-frames before temporally decoding preceding B-frames. By using multiple frames, H.264 delivers superior performance for translational motion and occlusions.

For blocks that are poorly represented

in previously decoded frames, due to such actions as camera panning or moving objects uncovering previously unseen background, motion compensation yields little significant compression benefit. In these instances, H.264 capitalizes on intraframe estimation to eliminate spatial redundancies. By also removing spatial redundancy in the pixel domain instead of exclusively in the frequency domain, as its predecessors do, H.264 achieves significantly better compression that is comparable to that of the JPEG-2000 still-image compression standard.

Intraframe estimation operates at the pixel-block level and attempts to predict the current block by extrapolating the neighboring pixels from adjacent blocks in a defined set of directions. The method then codes the difference between the predicted block and the actual block. Intraframe estimation is particularly useful in coding flat backgrounds (**Figure 2**).

### DOMAIN TRANSFORMATION

Perhaps the best known aspect of previous MPEG and H.26x standards is the use of DCTs to transform the video information that results from motion and intraframe estimation into the frequency domain in preparation for quantization. The widely used 8×8 DCT assumes a numerically accurate implementation, akin to floating point. This implementation leads to problems when you use mismatched inverse-DCT implementations

**TABLE 1—COMPARISON OF H.264-ENTROPY-CODING APPROACHES**

Characteristics	VLC	CABAC
Where it is used	MPEG-2, MPEG-4, ASP	H.264/MPEG-4 AVC (high-efficiency option)
Probability distribution	Static: probabilities never change	Adaptive: adjusts probabilities based on actual data
Leverages correlation between symbols	No: conditional probabilities ignored	Yes: exploits symbol correlations by using "contexts"
Noninteger code words	No: Low coding efficiency for high-probability symbols	Yes: exploits "arithmetic coding," which generates noninteger code words for higher efficiency

in the encoder and the decoder. This mismatch causes “drifting,” which results in visible degradations particularly apparent at low bit rates in streaming applications.

In a significant innovation, H.264 uses a DCT-like 4×4 integer transform to translate the motion-compensated data into the frequency domain. A key advantage of switching to the new algorithm is that the smaller block reduces blocking and ringing artifacts. In addition, integer coefficients eliminate the rounding errors inherent with floating-point coefficients. Rounding errors can cause drifting artifacts in MPEG-2 and MPEG-4 ASP.

### QUANTIZATION

Psycho-visual redundancy comes about because of the human eye’s acute sensitivity to slow and linear changes (constant low-frequency information) and its relative insensitivity to high-frequency information, such as busy textures. The human eye has a lower sensitivity to spatial resolution in the chrominance signal; in consumer-video applications, therefore, the system commonly subsamples chrominance signal by a factor of two, both horizontally and vertically. The output of the transform step completely represents information for all frequency levels, providing another opportunity for compression.

The quantization process eliminates high-frequency information by mapping, or quantizing, each DCT coefficient to a discrete set of levels. In H.264, the 4×4 transform expands to an 8×8 transform for chroma-predicted blocks and to a 16×16 transform for some luma-predicted blocks by applying second-level 2×2 and 4×4 integer transforms, respectively, to the lowest frequency information to which the human eye is most sensitive. You use the smaller transforms for the quantization of chroma samples because the chroma samples are already decimated at a 2-to-1 ratio. The larger transform for the luma samples reduces fidelity, but the human eye cannot discern the result. Quantization is also useful

in controlling the bit rate by selectively eliminating visual information.

### ENTROPY CODING

The entropy-coding stage maps symbols representing motion vectors, quantized coefficients, and macroblock headers into actual bits (Figure 3). In compression standards, all entropy coding shares a common goal: to reduce the average number of bits necessary to represent the compressed video. Previous standards accomplished this task by using VLC (variable-length-code) tables. The goal of VLC tables is to ensure that you use shorter code words for more frequently occurring symbols, such as small coefficient values. But you can also use arithmetic coding instead of VLC tables, and H.264 introduced this concept for the first time for video compression, even though it was an option in the JPEG standard for still images.

In either scheme, before entropy coding can begin, the system serializes the quantized DCT coefficients into a 1-D array by scanning them in zigzag order. The resulting serialization places the dc coefficient first, and the ac coefficients follow in low- to high-frequency order. Because higher frequency coefficients tend to be zero (the result of the quantization process), the system uses run-length encoding to group adjacent zeros, which re-

sults in more efficient entropy coding.

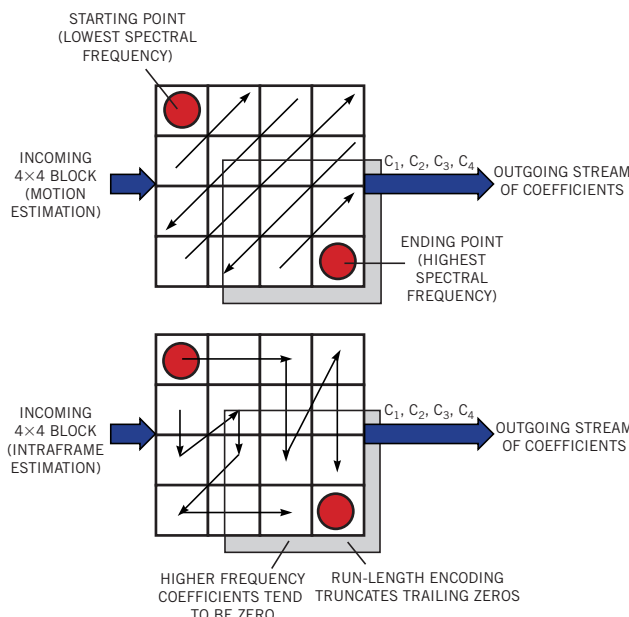
In MPEG-2, serialization depends on whether the coefficients originate from a motion-estimated or an intraframe-estimated macroblock. In H.264, serialization depends only on whether the coefficients originate from samples coded from the same video field (a field macroblock) or from a frame containing both the top and bottom video fields (a frame macroblock).

H.264 introduced CABAC (context-adaptive binary-arithmetic coding), which is more efficient than VLC for symbol probabilities greater than 50% because it allows you to represent a symbol with less than one bit. CABAC manages this task by adapting to the changing probability distribution of symbols and by exploiting correlations between symbols. Table 1 illustrates the differences between VLC and CABAC. H.264 also supports CAVLC (context-adaptive variable-length coding), which is superior to VLC without the full cost of CABAC.

### DEBLOCKING LOOP FILTER

Introducing more options and modes into the encoding algorithm also introduces more opportunities for discontinuities. Artifacts occur, for example, when you code adjacent macroblocks using different modes. The encoder may make independent decisions to compress macroblocks using the motion-estimation (interframe) mode, the spatial (intraframe) mode, or skip mode (skipping the macroblocks altogether). As a result, the pixels adjacent to two blocks compressed in different modes can have different values, even though they should be similar. Artifacts can also occur around block boundaries by the transformation/quantization process and motion-vector differences between blocks.

To eliminate these artifacts, H.264 defines a deblocking filter that operates on both 16×16-macroblocks and 4×4-block boundaries. In the case of the macroblocks, the filter eliminates artifacts resulting from motion or intraframe estimation or different quantizer scales. In the case of the smaller blocks, the



**Figure 3** The entropy-coding stage maps symbols representing motion vectors, quantized coefficients, and macroblock headers into actual bits.

filter removes artifacts that transformation/quantization and motion-vector differences between adjacent blocks cause. Generally, the loop filter modifies the two pixels on either side of the boundary using a content-adaptive, nonlinear filter. (Both the decoder and the decoding loop that replicates within each encoder use the deblocking filter—hence, the term “loop filter.”) The result is not only improved visual quality, but also, because motion compensation uses deblocked decoded frames, improved coding efficiency.

### MODEST COMPLEXITY INCREASE

As mentioned, H.264 achieves compression ratios that are two to three times better than those of its immediate predecessors. You must balance these gains, however, against the increase in the complexity of the algorithm as well as its implementation in silicon. The result is a manageable increase when you compare it to MPEG-2; so the trade-offs have been positive for the industry. Nevertheless, implementation is far from trivial and demands a thorough understanding of the standard and the silicon-design and -fabrication process. Striking examples are the new options available for compression coding, which requires a way to fit together macroblocks that you have compressed using different modes. Implementing this feature and other H.264 features in silicon requires years of experience to achieve the most cost-effective silicon approach. □

---

### REFERENCE

1. Côté, Guy, and Lowell Winger, “Recent Advances in Video Compression Standards,” *IEEE Canadian Review*, Spring 2002.

---

### AUTHOR'S BIOGRAPHY

*Didier LeGall is vice president of engineering and business development for LSI Logic's Broadband Entertainment Division. He is in charge of the engineering and business-development activities for digital-video products, including DVD, video peripheral, PVR/DVR, and video production and broadcasting. He has been involved with ISO's MPEG-standardization effort since its inception and served as chairman of the MPEG-Video group until 1995. He holds a doctorate in electrical engineering from the University of California—Los Angeles.*