

VIDEO COMPRESSION SLIMS DOWN FOR SPRING

EVER-IMPROVING CODECS SATISFY YOUR HUNGER FOR HIGH QUALITY WITHOUT BLOATING YOUR COST, MEMORY, POWER, AND PROCESSING BUDGETS. BUT WHICH OF THE SMORGASBORD OF CONTENDERS SHOULD BE THE MAIN COURSE? BEFORE FILLING YOUR PLATE, SAMPLE THE SPREAD TO FIGURE OUT WHAT YOU HAVE A TASTE FOR AND WHY.

At a glance.....60

An eye opener.....61

By Brian Dipert, Technical Editor

SEVERAL OF EDN'S RECENT MULTIMEDIA articles have pointed out the bloated sizes of high-fidelity audio files and the consequent appeal of compressing them in lossless or lossy ways for storage and transmission. In comparison to

video, though, uncompressed audio seems downright diminutive. Consider the following uncompressed-video bit-rate examples. (Multiply by 60 and divide by 8 to get the required per-minute storage capacity in bytes.):

- Cellular phones: 15 frames/sec, 8-bit color, QCIF (176×144-pixel) resolution=3.1 Mbps;
- PDAs: 30 frames/sec, 16-bit color, CIF (352×288-pixel) resolution=48.7 Mbps;
- PCs: 30 frames/sec, 24-bit color, VGA (640×480-pixel) resolution=221.2 Mbps;
- HDTV: 60 frames/sec, 24-bit color, 720P (1280×720-pixel, progressive-scan) resolution=1.4 Gbps (1000 times higher than the audio-CD bit rate);
- Digital cinema: 24 frames/sec, 30-bit color, 1080P (1920×1080-pixel, progressive-scan) resolution=1.5 Gbps.

To hold 1.5 Gbps of data over the duration of a typical two-hour movie, you would need slightly more than 1.3 Tbytes of storage capacity, which translates to 72 20-Gbyte drives or 80 high-density, dual-layer, double-sided DVDs. Add an alpha transparency channel to the red, green, and blue channels' data, and you bump up the bit-rate an additional 33%. Keep in mind, too, that these specs don't include audio, which is necessary unless all you want to watch are old Charlie Chaplin silent movies. They also don't include the bit overhead of error detection and correction, multistream synchronization, media identification and security, and other control-data structures. Mind-boggling, isn't it?

Now that you're convinced of your system's need for compression, which

For extra information on video compression, including a vendor table, article references, and a sidebar, go to the Web version of this article at www.ednmag.com.

of the multitude of algorithm options should you select? (See the table and a sidebar, “Quest for clarity,” on the Web version of this article at www.ednmag.com.) Instead of immediately picking a product, first identify the problem you’re trying to solve and the environment under which your system is operating. Is the network topology client/server or peer-to-peer? In other words, are you building a one-to-many unidirectional broadcast, a one-to-one bidirectional link, or an arrangement between these two extremes? Your answer determines the degree of source-to-destination symmetry in encoding and decoding complexity that the application demands and whether encoders and decoders must reside at both ends of the link.

Is the broadcast live, or can the broadcaster compress the material at non-real-time speeds and then later transmit it offline? This differentiator determines the acceptable complexity of the encoding algorithm or, from a different perspective, the amount of memory and processing horsepower you’ll have to throw at the algorithm you select. On one extreme, you might have, for example, a live broadcast of a sporting event. On the other extreme might be a DVD that the movie studio masters and mass-produces and consumers then buy or rent.

Next, analyze the transmission link between sender and recipient, which can take the form of either a wired- or a wireless-network connection or as a mass-storage device, such as an optical disc. What are its peak and typical bandwidth capabilities? What is its latency? How likely is it that encoded data packets will arrive at their destination out of order and that some won’t ever arrive? Will recipients expect to immediately begin viewing the video after pressing a computer mouse or remote control button, or can they tolerate a delayed response? A slow, error-filled link may not be a problem if the system is downloading

AT A GLANCE

- ▶ Video applications have different needs and priorities, but all are opportunities for lossless or lossy compression.

- ▶ Color subsampling and frequency-domain transformation can reduce file size without visible degradation.

- ▶ Quantization represents a tightrope act of balancing between beauty—that is, quality—and bit rate.

- ▶ Processing before compression and after decompression is often a key piece of the codec puzzle.

- ▶ The never-ending tug of war between proprietary and industry-standard algorithms advances the state of the art for everyone.

movies in the middle of the night to a set-top-box hard drive. It’s a bigger potential problem if you’re a teenager who wants to watch a high-quality presentation of the latest Britney Spears music video *right now*. In so-called progressive download, playback begins after the incoming data stream fills a buffer whose size corresponds to the amount of time it takes to download the remainder of the stream in the background. This downloading is an intermediate step between streaming video on one end and full download-and-then-play on the other.

How much processing power and how much volatile and nonvolatile memory are available at the receiving-video appliance? Is battery life a concern, or is the widget plugged into a wall outlet? What’s more important for a fixed-density video-storage medium: image quality or playback time? How big is the display, what is its maximum perceivable resolution, and how far away will the average viewer be from it? What are its refresh rate, color depth, and color gamut? Can you develop a single recipient description

that will apply to all of the video content’s potential viewers, such as in a closed cable television or cellular network, or will you have to satisfy viewers with a range of hardware and software at their disposals, such as cellular phones, PDAs, and PCs, all accessing the Internet?

KEEPING IT PRISTINE

Lossless video compression, which reduces the number of bits but doesn’t degrade or otherwise alter the images, is useful when you’re preparing video material for archiving. Although a lossy perceptual codec might not produce artifacts visible to your eyes, those alterations may still wreak havoc on encoding or other transformation algorithms you might want to later apply to the archived material. Huffvuv is perhaps the best known lossless codec. By taking advantage of the minimal pixel-to-pixel variations that exist in most parts of most video frames, it works in a conceptually similar fashion to lossless audio-tuned codecs, such as Shorten.

According to the developer’s Web site, “Huffvuv’s algorithm...predicts each sample and Huffman-encodes the error. The predictor functions are “left,” which predicts the previous sample from the same channel; “gradient,” which predicts Left + Above – AboveLeft; and “median” which predicts the median of Left, Above, and the gradient predictor. The error signal in each channel is encoded with its own Huffman table. On compression, Huffvuv picks appropriate tables from its built-in collection. These tables are then stored in the output file and used when decompressing. This way future versions of Huffvuv can decompress old files without my having to explicitly support old tables. A Huffvuv-savvy application can also specify the Huffman tables to be used for compression instead of accepting the defaults.”

Compressing a file to half or even one-fourth of its original size is great. But

TABLE 1—MPEG-x AND H.26x CODEC COMPARISONS

	MPEG-1	MPEG-2	MPEG-4	H.263	H.261
Primary application	CD-ROM	DVD, HDTV, broadcast	Portable multimedia	Desktop videoconferencing	Wire-line videoconferencing
Video-data rates	1.5 Mbps	4.6 Mbps	20 kbps to 6 Mbps	20 to 384 kbps	128 to 384 kbps
Typical video-frame size (pixels)	352×240	720×480	176×144 (QCIF), 352×288 (CIF)	176×144 (QCIF), 352×288 (CIF)	176×144 (QCIF), 352×288 (CIF)
Typical audio quality	Stereo CD	Surround sound	Speech, stereo CD, surround sound	Speech	Speech
Error resilience	Basic	Basic	Advanced	Adequate	Basic

(Table courtesy Toshiba)

what if you need to put your bit stream on an even more extreme diet? You now enter the realm of “visually lossless” compression. For example, you might decide to subsample the pixels’ chrominance information, because luminance accuracy is more critical to the human visual system. Component video first transforms red, green, and blue into Y (luminance); Cr (red minus Y); and Cb (blue minus Y) or undertakes some related transform function. Its 4:2:2 format then stores and transmits only half as many Cr and Cb samples as Y samples on each video-scan line. So-called 4:2:0 video takes the next step; there are only half as many scan lines

of Cr and Cb data as there are Y lines. Other combinations, such as 4:1:1, are possible. These digital transformations are analogous to the color-bandwidth-limiting that occurs in the analog domain before video storage and transmission.

Next step? Take another page from the audio-compression book of tricks, in which the algorithm transforms a group of temporally adjacent sound samples into the frequency domain. Image compression undertakes a conceptually similar transformation for a group of spatially adjacent pixels. As long as you don’t alter the frequency-domain data, a subsequent retransformation back to the

spatial domain theoretically reconstructs the original information, and, to the limits of arithmetic precision, algorithms such as lossless JPEG accomplish this goal. The advantage of such frequency transformations is that they output data that, by virtue of its reduced randomness, more efficiently compresses. Note that JPEG-based algorithms convert RGB to component video before frequency transformation and thus aren’t truly lossless.

DOING THE MATH

Still need more compression? At this point, the algorithms need to take that frequency-transformed data and quantize it;

AN EYE OPENER

Thanks again to all of you who gave me feedback on last year’s digital-audio-analysis hands-on project. This year, I wanted to do something similar for video compression, expanding beyond “how it works” to also include “how it looks.” I briefly considered a subjective study, but unless I assembled a diverse set of test subjects, the results wouldn’t be broadly applicable to anyone but me. Anyway, plenty of other excellent work already exists in this area (**references A to E**).

Instead, I’ve partnered with Sarnoff Labs, employing the company’s JNDmetrix video-analysis tool. Conventional peak-SNR tools simply calculate the error differences between the pixels in a reference image and those in lossy-compressed versions of the image. JNDmetrix, in contrast, accounts for perceptual factors in determining when error is perceptible or not to the human visual system. To use JNDmetrix, you need to be able to capture 10 consecutive frames that the lossy-compressed video’s decoder outputs, a capability that not all codecs provide.

The test clips come from ftp.tnt.uni-hannover.de, hosted by the University of Hannover, Germany. The test clips’ specifications include YUV (luminance, bandwidth, chrominance)

encoding, 12-bit samples, 352×288-pixel resolution and 30-frame/sec frame rate. E-Testing Labs also used these clips in its subjective study (**Reference D**).

The E-Testing Labs report states: “ISO/ITU clips like these are used industrywide to compare the capabilities of compression technologies. ISO/ITU clips are classified as Class A through E and represent varying degrees of content complexity. Class C clips are the most difficult clips to encode for testing MPEG-4 and are described as clips ‘that contain high amounts of spatial data and a medium amount of movement’ and represent a wide variety of content delivered today.”

In addition to measuring perceptual quality via JNDmetrix, I’ll also be logging the encoding and decoding times and the corresponding CPU and memory usage. I have several desktop PCs at my disposal, including a 2-GHz Pentium 4 system running Windows XP, a 1.8-GHz Pentium 4 system running Windows 2000, and an 800-MHz Pentium III dual-processor system running Windows 98SE. After encoding, I’ll attempt to “stream” the video across WAN connections, including both ADSL and 56-kbps analog modem, but not wireless,

because CDPD offers too low bandwidth, and Ricochet is still offline. I’ll also stream the video across 10- and 100-Mbps Ethernet and 802.11b LAN connections, not only to other desktop PCs, but also to several notebook computers and to a Casio E-125 Pocket PC. Target encoding bit rates are 30, 200, and 800 kbps. At the 800-kbps rate, I will use a variable bit rate if the codec supports it. These bit rates correspond to my best guesses of common settings for narrowband and broadband Internet broadcasts and for storage on optical media, such as CDs.

Lossy-video encoding and decoding products that were available at press time included Adobe (www.adobe.com) Premiere, Apple QuickTime Pro, Casio Mobile Video, Discreet (www.discreet.com) Cleaner, DivXNetworks 5 and Brandon Wirtz’s optimized DivX, Microsoft Windows Media Encoder and Portrait tool set, On2 Technologies’ VP3.2 (open source) and VP4, RealNetworks’ RealProducer, Sorenson Video 3, Streambox ACT-L2, and ZygoVideo’s Pro. I also hope to be able to test MPEG-4 from DynaPel, as well as many of the other lossy codecs that are listed in the table on the Web version of this article at www.ednmag.com. In addition, I will bench-

mark the Huffvuv lossless codec.

After I take my best stab at highest quality compression, I’ll turn over my results to the vendors to see whether they can improve them. This ongoing project on the *EDN* Web site will continue far beyond this issue’s publishing date, so as vendors upgrade and add tools, I can incorporate them into the testing. Your suggestions are welcome on how I should prioritize this work and on codecs I may have overlooked.

REFERENCES

A. Waggoner, Ben, “Web video codecs compared,” *DV* magazine, November 2001, pg 22.

B. Schiavon, Francesco, “Quality comparison: RealVideo, Windows Media Video and Sorenson Video,” *Streaming Media*, Nov 20, 2001, www.streamingmedia.com/r/printerfriendly.asp?id=8058.

C. Ozer, Jan, “MPEG-4: looks great!,” *ExtremeTech*, Dec 20, 2001, www.extremetech.com/article/0,3396,s=1022&a=3780,0.asp.

D. eTesting Labs, “Microsoft: video quality comparison study,” *EDN*, May 1, 2001, <http://www.etestinglabs.com/main/reports/msvideo.pdf>.

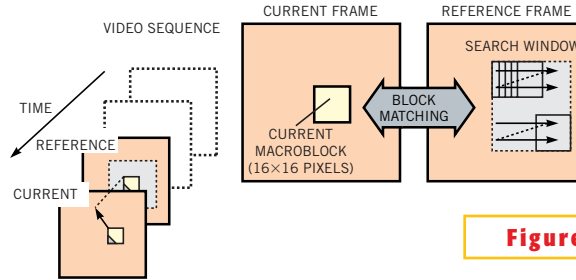
E. Alpha Video, *Codec Shootout*, www.codecshootout.com.

that is, permanently discard information that they've decided is unimportant given your target bit stream and file sizes, image resolution, color depth, and frame rate. Many of the lossy codecs employ conceptually similar high-level approaches: Transform a group of pixels either within a scan line or spanning multiple scan lines into the frequency domain, make intelligent decisions about what to

delete, and then untransform at the decoder. (See the table on the Web version of this article at www.edn.mag.com.) The specifics create the codec-versus-codec differences, because developers tune their techniques for dissimilar source material, delivery methods, assumptions of available processing power and memory at the encoder and decoder, and displays (see sidebar "An eye opener").

Although audio samples aren't highly random over short periods, they become increasingly uncorrelated as the time span between them lengthens (with the possible exception of repetitive Top 40 radio). Video differs from audio in that, unless the scene itself or the camera angle within the scene changes, one video frame is much the same as the next. Objects and, therefore, the pixels representing them might move around from one frame to another. But if the encoding algorithm can identify that movement, it can store the full set of pixel data the first time and store the much smaller set of vector data indicating where the pixels moved for subsequent frames (Figure 1).

Pixel movement is at the heart of the compression enhancements that marked the transition from MJPEG (motion JPEG) to MPEG. MJPEG is a series of JPEG-compressed still images—one JPEG image per video frame. MPEG calls these complete images I pictures. (Others generally also call them key pictures.) In MPEG-1, motion-estimation algorithms compare consecutive frames, find the closest matches if they exist for their respective pixel groups, and encode direction vectors between them. When the direction vectors for a compressed frame reference the frame before it, the compressed frame is called a P (predictive) picture. When they derive from the previous frame, the *next* frame, or a combination of the two frames, that frame is



Motion estimation takes advantage of frame-to-frame pixel redundancy and comprehends movement of those pixels (courtesy Fujitsu).

called a B (bidirectionally predictive) picture. B and P pictures are collectively also known as *delta pictures*.

NOW, I KNOW MY Bs AND Ps...

Often-optional B pictures are much more computationally intensive than P pictures, but they theoretically improve the compression efficiency and resultant quality by providing twice as many potential matches for each pixel cluster. To fine-tune the required performance and memory versus the resultant quality and compression ratio, your algorithm can adjust the match search radius around each original pixel group's location, as well as the acceptable match-detection threshold. Motion estimation sends not only direction vectors, but also as much of the prediction error as the bit rate allows. (The prediction error is the difference between the closest-match derived pixels and the actual pixels being compressed.) MPEG-2's enhancements beyond MPEG-1 include the ability to calculate motion estimation between interlaced fields, not just between frames, as well as support for higher frame resolutions (Table 1).

Once you've encoded an initial key picture, all you need to do is provide delta pictures for the remainder of the video stream, right? Not quite. Significant picture-to-picture transitions, such as scene changes, require key pictures. Because vector prediction is the best-possible—not necessarily perfect—pixel cluster-to-cluster match and because a series of delta pictures build on each other, eventually the video quality will degrade unless you occasionally "reset" it with a key picture. Errors that enter the bit stream during storage, transmission, and playback also affect subsequent delta pictures until the next key picture. Delta pictures are difficult to impossible to edit in a stand-alone fashion; this fact, along with

quality, is another reason that MJPEG, particularly the lossless variant, finds use as an archival codec. And periodic key pictures enable video playback to begin from multiple points within the stream, not just at the beginning.

MPEG-4, which the Motion Picture Experts Group first ratified in 1999, means different things to different people: a series of audio

codecs, video codecs, 2- and 3-D graphics formats, and—most notably, compared with past MPEG standards—a robust high-level file format "wrapper" derived from the format Apple uses for QuickTime. As with past MPEG standards, MPEG-4 comes in multiple profiles, each with an associated set of levels. These profiles and levels tailor the codec suite and the corresponding encoder and decoder complexity for various application needs. Although MPEG-2 offered numerous profiles and levels, only a few combinations achieved widespread use, such as on DVDs and in DTV (Table 2). A similar scenario will probably play out to some extent with MPEG-4 in standardized distribution formats, such as, perhaps, high-definition DVD, and with "closed" networks. But an increasing potential for consumer confusion comes with the emergence of high-bandwidth wireless as a burgeoning variety of devices access diverse Internet-housed content.

Several possible scenarios could occur. One unlikely scenario would have each media client containing a robust, costly, and high-power-consuming, multiprofile- and multilevel-compatible MPEG-4 decoder. In a second and viewer-unfriendly scenario, each media server contains multiple versions of each piece of content at various profiles and levels, and the consumer has to select a variant that matches the capabilities of his or her device and communication channel. In an ungainly third scenario, the client and server negotiate respective capabilities and determine an optimum profile-and-level combination.

The DivX ;-) compression format, a tongue-in-cheek reference to the now-defunct "enhanced" DVD that Circuit City promoted a few years ago, was, in its original Version 3.11 variant, a hacked version of Microsoft's MPEG4v3 codec.

In the developers' own grammatically challenged words, "This is a Hack of selected version of the M\$ MPEG4 codec. FOURCC and CLSID code have been hacked so you can make ALWAYS WORKING AVIs. it can coexist with retail version of the media-encoder tools and newers or hacked version of M\$ MPEG4 codec." In other words, by altering the FOURCC (four-character video-format identification) and CLSID (class-identifier) strings, the hackers made Microsoft's codec not only compatible with the proprietary ASF (Advanced Streaming Format), but also with the

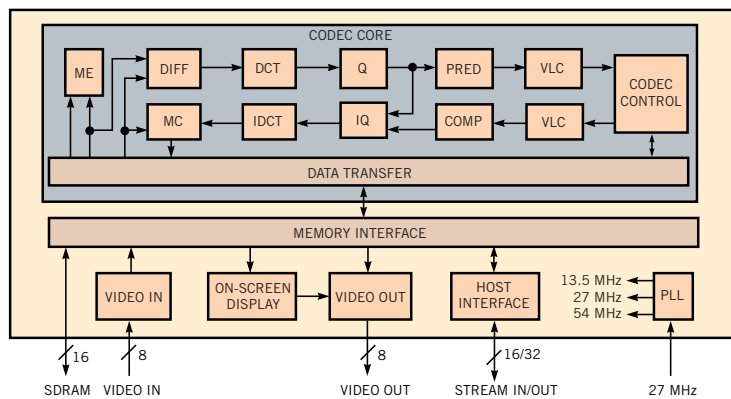
now-de facto industry-standard AVI (Audio Video Interleaved) "wrapper." Today's most common application of DivX, in conjunction with DeCSS and open-source MPEG-2 decoders, is "ripping" audio and video from DVDs, compressing the video to DivX, burning them onto CDs, and sharing them with others over peer-to-peer networks.

Remember how much concern MP3 gave—and still gives—recording studios? DivX is creating similar uproar within Hollywood movie studios even though, practically speaking, the sizes of DivX-encoded files make for much less wide-

spread video piracy even with broadband-network connections. Ironically, Microsoft released MPEG4v3 before the standard's approval, and DivX 3.11 therefore is both incompatible with MPEG-4 compliant decoders and lower quality than Microsoft's subsequent MPEG-4-compliant codec and the company's even higher quality Windows Media Video codec. The developers of DivX 4.xx and newly released 5.xx claim that their creations, also known as DivX Deux, are rewrites of the codec from the ground up, containing no Microsoft-sourced code and being fully MPEG-4-compliant. They come as open-source Open DivX, which the Project Mayo group maintains, and an enhanced, proprietary codec that DivXNetworks offers.

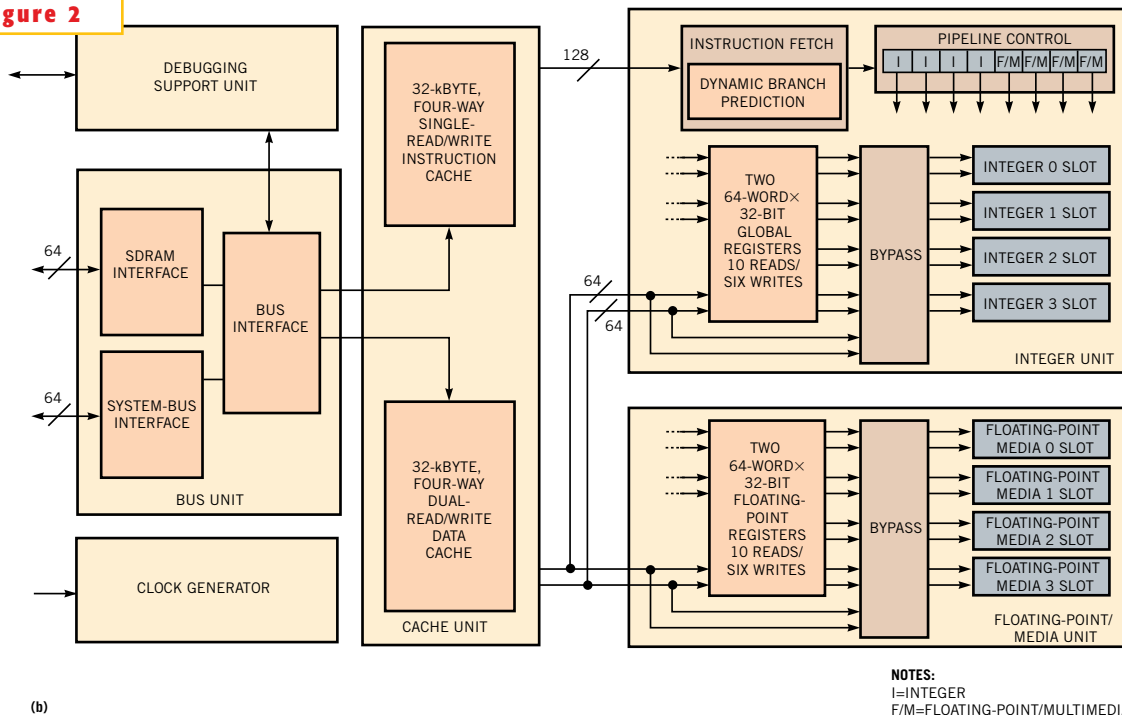
CATCHING A WAVE

At extreme compression levels, the aggressive quantization of pixel group-based transformations becomes visible as "blocky" artifacts. An alternative approach, wavelet compression, simultaneously transforms the entire frame of pixels and can sometimes produce a more visually pleasing result. Analog Devices is leading the wavelet charge from a silicon standpoint, targeting both still-image and video compression. QuVIS applies wavelet compression to digital cinema,



- (a)
- NOTES:
 ME=MOTION ESTIMATION.
 DIFF=DIFFERENCING.
 MC=MOTION COMPENSATION.
 DCT=DISCRETE-COSINE TRANSFORM.
 IDCT=INVERSE DISCRETE-COSINE TRANSFORM.
 Q=QUANTIZATION.
 IQ=INVERSE QUANTIZATION.
 PRED=PREDICTION.
 COMP=COMPENSATION.
 VLC=VARIABLE-LENGTH CODING.
 VLD=VARIABLE-LENGTH DECODING.

Figure 2



- (b)
- NOTES:
 I=INTEGER
 F/M=FLOATING-POINT/MULTIMEDIA

Dedicated hardware (a) and multimedia-enhanced CPUs (b) represent some of the many algorithm "vehicle" alternatives (courtesy Fujitsu).

and numerous companies offer wavelet-based compression software. Video-targeted wavelet formats remain proprietary, but the JPEG2000 specification has brought long-needed standardization to wavelets for still imaging. Just as MJPEG was one of the earliest widespread video-compression schemes, Motion JPEG2000 could conceivably bring standards-based wavelet compression to video. MPEG-4 supports wavelets, albeit for still images.

Two often-related techniques find common use in optimizing video quality. VBR (variable-bit-rate) encoding intelligently allocates bits among pixel groups and frames on an as-needed basis, mini-

versus video streaming. (DVD players with display-bit-rate modes allow you to view the statistics as you watch a movie, which can be educational.)

In traditional one-pass encoding, the algorithm must decide on the fly where to use key and delta pictures and, with VBR, where to allocate the bits. *Multipass encoding*, as the name implies, gives the encoder several opportunities to examine the video source material during the compression process. This more in-depth analysis impacts performance, memory, and power consumption, but it also might produce a higher quality result than with single-pass encoding at the same bit rate.

Increased pixel-to-pixel randomness reduces compression efficiency with lossy codecs, just as it does with lossless algorithms. For this reason, you should select as “clean” a video source as possible, free from “blockiness” and other compression artifacts, as well as with minimal analog and digital noise. For similar reasons, many codec vendors suggest the use of a preprocessing, “smoothing” low-pass filter to reduce the abruptness of object edges. Because eyes are more sensitive to brightness than to color, encoding algorithms often compress dimly illuminated frames and portions of frames more than they do brightly lit alternatives. And if the original video source is film, preprocessing the video with an inverse-telecine filter before compression eliminates redundant data.

WRAPPING IT UP

It’s interesting to analyze the tug of war between proprietary codecs, such as WMV and RealVideo, and industry-standard codecs, such as MPEG-4. On the one hand, you might argue that a proprietary codec, unencumbered by the slothlike politics of standardization, can more quickly advance the state of the art. But, in that case, you’re also handing over format control to a single company. You’ll find no easy answers regarding cost, either. Licensing fees for recently announced and hotly contested simple- and core-profile MPEG-4 Visual are 25 cents per encoder, 25 cents per decoder (therefore 50 cents per codec), and 2 cents per hour for content. The content fee, not part of the earlier MPEG-1 and -2 licensing agreements, is the most dif-

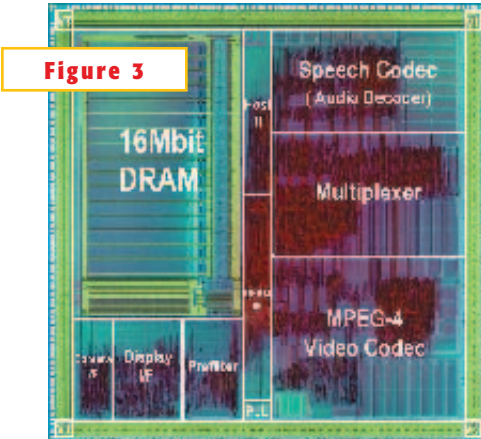


Figure 3

By embedding the memory and thereby reducing the amount of chip-to-chip interaction, you can further reduce power consumption at the system level (courtesy Toshiba).

mizing bits when possible and maximizing them when necessary. The average bit rate is no higher than that of the more common and less processing- and memory-intensive CBR (constant-bit-rate) approach. But the peak bit rate you see for any one video frame with VBR may be significantly higher than the average.

The variance between average and peak bit rate has several important implications. First, the video decoder needs to include buffer memory, which may be cost-prohibitive with price-sensitive consumer-electronics gear. Second, if the encoder overwhelms the peak processing capability of the decoder, the viewer sees distracting dropped frames. Finally, the encoder must not exceed the peak bandwidth of the network connection or the transfer rate of the mass-storage device. Therefore, VBR more commonly finds use in download-and-play environments

difficult to enforce, and companies and consortiums, such as the Internet Streaming Media Alliance, also claim that this fee is commercially inappropriate. Also note that MPEG LA has yet to announce licensing fees for MPEG-4 Audio or Systems. In contrast, Microsoft provides free encoding and decoding tools as long as the content ends up residing on hardware that runs on a Windows operating system. If you want to use Windows Media Technologies with a non-Microsoft OS, you have to negotiate a license fee.

Companies such as DivX Networks and Microsoft straddle both sides of the proprietary-versus-standardized fence. Both claim MPEG-4 compatibility at the video-codec level. But DivX Networks delivers that codec in a proprietary DivX wrapper-file format, and Microsoft delivers it in ASF and WMV, versus MPEG-4's MP4. Microsoft's wrapper also houses the non-MPEG-4-compliant WMA audio codec. Microsoft's support of MPEG-4 might surprise those familiar with its proprietary tendencies. In fact, the company was one of the first to deliver a standards-compliant codec. Microsoft's MPEG-4 embrace is a pragmatic move: Although the company prefers that you use its Windows Media Video codec, it knows that it must support MPEG-4, too, and would rather be on the receiving than the giving end of the royalty stream. Plus, its presence on the standards board gives it some amount of control over the standard and visibility of its competitors' activities.

These wrappers provide identification information for the potentially numerous audio, video, and other streams they store within them. They direct the decoder's playback of those streams, such as keeping video and corresponding audio in synchronization and presenting graphics and HTML content at appropriate places in the presentation. In the case of MPEG-4, they even suggest which audio and visual objects within the file the server and decoder can discard in

bandwidth- and processing-constrained environments. Format wrappers also embed media-access rights-control bits.

Network congestion, interference-filled transmission media, and other real-world phenomena can result in out-of-order and lost-data packets, and wrapper formats enable the decoder to appropriately respond to these packets. If there's sufficient time, the receiver might request that the transmitter resend the missing packet. Otherwise, it uses interpolation techniques to construct a semblance of the missing information or, more simply, redisplay previously received intact data. This dynamic per-frame quality-scaling technique is amenable to wavelet compression, which progressively "builds" the entire image as the decoder receives additional data about it.

Postprocessing at the decoder before displaying the data works in conjunction with "hints" that the encoded bit stream embeds; this postprocessing can boost viewers' perceived video quality. If aggressive quantization creates distracting blocking artifacts, the decoder can apply a smoothing algorithm to blend the blocks. Conversely, if the encoder applies a smoothing preprocessing filter to reduce randomness, the decoder can apply compensating edge sharpening. The decoder can also interpolate between frames and between scan lines within a frame to improve the quality, the apparent resolution, and the apparent frame rate of the presentation.

ABUNDANT OPTIONS

In selecting video-compression and -decompression algorithms, you also need to decide which silicon platforms you want to run them on. The historical trend has been that first-generation hard-wired circuits gave way to more flexible DSPs, followed eventually by a move to general-purpose CPUs with multimedia-tuned functions such as Intel's MMX (multimedia-extensions) and SSE (streaming-SIMD-extensions) instruction sets. De-

TABLE 2—MPEG-2 PROFILES, LEVELS, AND CHARACTERISTICS

Level	Simple	Main	4:2:2
High		4:2:0; 1920×1152 pixels; 80 Mbps; I, P, B	
High-1440		4:2:0; 1440×1152 pixels; 70 Mbps; I, P, B	
Main	4:2:0; 720×576 pixels; 15 Mbps; I, P	4:2:0; 720×576 pixels; 15 Mbps; I, P, B	4:2:2; 720×608 pixels; 50 Mbps; I, P, B
Low		4:2:0; 352×288 pixels; 4 Mbps; I, P, B	

coders tend to perform this migration first, and encoders follow. However, the sequence depends on the algorithm's degree of asymmetry. Battery-powered or otherwise-power-conscious devices also tend to stay with fixed-function hardware longer than do devices that you plug into a wall. For example, notebook PCs until recently included MPEG-2-decoder chips, even if their CPUs could handle decoding in software; this approach maximizes battery life during movie playback. As graphics accelerators incorporated an increasing percentage of the video-decoding pipeline, the need waned for redundant functions in separate video-decoding chips.

Today's hardware options are fixed-function circuits and DSPs. In both cases, you also need to include a separate CPU or a microcontroller in your system design to handle other system-level functions (Figure 2a). Other options include dual-processor combinations of a DSP and a CPU on one piece of silicon, and CPUs with SIMD (single-instruction multiple-data)-integer and floating-point support (Figure 2b). The more versatile the silicon foundation, the more easily the system can respond to codec upgrades and diversification. Conversely, the more hard-wired the foundation, the higher its per-clock performance is and the lower its power consumption and cost are. Embedding the buffer and scratchpad memory further reduces system-level power consumption (Figure 3). FPGAs, which deliver many of the fixed-function benefits of ASICs but are in-system-reprogrammable-like processors, offer an interesting middle-ground option, and companies such as QuickSilver Technology and Xilinx are increasingly focusing their resources on video compression and decompression.

Codec innovations and introductions are accelerating. This trend is especially desirable if vendors maintain compatibility with previous-gener-



ation decoders, and the lack of this compatibility increasingly isn't an issue, thanks to easily upgradable system firmware. Microsoft's Corona codec, which the company unveiled at 2001's Streaming Media East show, represents the next generation of Windows Media Technology. It combines audio, video, and server enhancements and targets high-definition television and surround sound.

Industry-standards bodies acknowledge that proprietary codecs, such as RealVideo, Sorenson, and Windows Media, are rapidly improving in quality. Their response, the H.26L, also known as MPEG-4 JVT (Joint Video Team), development group (<http://standard.pictel.com/ftp/video-site/>), may not be backward-compatible with MPEG-4's profiles and levels. (Note that MPEG-4's developers based much of the video technology on H.263.) As was the case with MPEG-4's AAC audio-compression algorithm versus previous-generation MP3 and with Corona's Windows Media Pro versus today's Windows Media 8, vendors sometimes need to approach development with a clean slate to ensure sufficient advancement in the state of the art. □

ACKNOWLEDGMENTS

Special thanks to Ben Waggoner from Interframe Media (www.interframe.com) and to Brandon Wirtz from Griffin Digital Solutions (www.griffin-digital.com) for their instructive dialogue on various video-compression discussion lists and Wirtz's review of an early draft of this article. Thanks to Doug Dixon and peers at Sarnoff Labs for their assistance with JNDMetrix.

AUTHOR'S BIOGRAPHY

One of these days, Technical Editor Brian Dipert will plug his DV camcorder into the PC's 1394 card and begin exploring the brave new world of video editing. You can reach Brian at 1-916-454-5242, fax 1-916-454-5101, bdipert@pacbell.net, and www.bdipert.com.

SNR	Spatial	High
	4:2:0; 1440×1152 pixels; 60 Mbps; I, P, B	4:2:0, 4:2:2; 1920×1152 pixels; 100 Mbps; I, P, B
4:2:0; 720×576 pixels; 15 Mbps; I, P, B		4:2:0, 4:2:2; 1440×1152 pixels; 80 Mbps; I, P, B
4:2:0; 352×288 pixels; 4 Mbps; I, P, B		4:2:0, 4:2:2; 720×576 pixels; 20 Mbps; I, P, B

VIDEO COMPRESSION VENDOR TABLE

Company/consortium	Product	Web site
ActiveSky	Lossy software encoders and decoders	www.activesky.com
Advanced RISC Machines (ARM)	Lossy hardware (cores) encoders and decoders	www.arm.com
Alphamosaic	Lossy hardware encoders and decoders	www.alphamosaic.com
Amphion	Lossy hardware (cores) encoders and decoders	www.amphion.com
Analog Devices	Lossy hardware encoders and decoders	www.analog.com
Anark	Lossy software encoders and decoders	www.anark.com
Apple Computer	Lossy software encoders and decoders	www.apple.com , www.quicktime.com
ARC Cores	Lossy hardware (cores) encoders and decoders	www.arccores.com
ATI Technologies	Lossy hardware decoders	www.ati.com
Avipix	Lossy software encoders and decoders	www.avipix.com
Axeda (Ravisent Technologies)	Lossy software encoders and decoders	www.axeda.com
BOPS	Lossy hardware (cores) encoders and decoders	www.bops.com
Casio	Lossy software encoders and decoders	www.casio.com
Cirrus Logic	Lossy hardware encoders and decoders	www.cirrus.com
Compression Science	Lossy hardware and software encoders and decoders	www.compressionscience.com
Compression Technologies	Lossy software encoders and decoders	www.cinepak.com
Custom Technology	Lossy software encoders and decoders	www.cinecraft.com
Cyberlink	Lossy software encoders and decoders	www.gocyberlink.com
DemoGraFx	Lossy hardware encoders and decoders	www.demografx.com
DiamondBack Vision	Lossy software encoders and decoders	www.objectvideo.com
Divio	Lossy hardware encoders and decoders	www.divio.com
DivX Networks (and Project Mayo)	Lossy software encoders and decoders	www.divx.com , www.projectmayo.com
DSP Group	Lossy hardware encoders and decoders	www.dspg.com
DynaPel	Lossy software encoders and decoders	www.dynapel.com
Emblaze Systems	Lossy hardware and software encoders and decoders	www.emblaze.com
Envivio	Lossy software encoders and decoders	www.envivio.com
Equator Technology	Lossy hardware encoders and decoders	www.equator.com
ESS Technology	Lossy hardware encoders and decoders	www.esstech.com
E-Vue		www.e-vue.com
FAST Search and Transfer	Lossy software encoders and decoders	www.fastsearch.com
Firepad	Lossy software encoders and decoders	www.firepad.com
Flask	Lossy software decoders	www.flaskmpeg.net
Flavor Software	Lossy software encoders and decoders	www.flavorsoftware.com
Fujitsu	Lossy hardware encoders and decoders	www.fujitsu.com
Generic Media	Lossy software encoders and decoders	www.genericmedia.com
Globespan	Lossy hardware encoders and decoders	www.globespan.net
Grass Valley Group	Lossy hardware and software encoders and decoders	www.grassvalleygroup.com
Hantro Products	Lossy hardware encoders	www.hantro.com
Happy Machines	Lossy software encoders and decoders	www.3ivx.com
Image Power	Lossy hardware (cores) and software encoders and decoders	www.imagepower.com
Improv Systems	Lossy hardware encoders and decoders	www.improvsys.com
IndigoVision	Lossy software encoders and decoders	www.indigovision.com
InSilicon	Lossy hardware (cores) encoders and decoders	www.insilicon.com
Intel	Lossy hardware and software encoders and decoders	www.intel.com
Intervideo	Lossy software encoders and decoders	www.intervideo.com
ISMA (Internet Streaming Media Alliance)	Consortium	www.ism-alliance.com
KT-Tech	Lossy software encoders and decoders	www.kttech.com
Lexra	Lossy hardware (cores) encoders and decoders	www.lexra.com
Ligos	Lossy software encoders and decoders	www.ligos.com
LizardTech	Lossy software encoders and decoders	www.lizardtech.com
Loronix	Lossy software encoders and decoders	www.loronix.com
LSI Logic	Lossy hardware encoders and decoders	www.lsil.com
LuraTech	Lossy software encoders and decoders	www.luratech.com
Luxxon	Lossy hardware and software encoders and decoders	www.luxxon.com
Matsushita	Lossy hardware encoders and decoders	www.panasonic.com
Media Excel	Lossy software encoders	www.mediaexcel.com
Media Metastasis	Lossy software encoders and decoders	www.zygovideo.com
MediaQ	Lossy hardware decoders	www.mediaq.com
MetaVideo	Lossy hardware (cores) encoders and decoders	www.metavideo.com
M4IF (MPEG-4 Industry Forum)	Consortium	www.m4if.org
Microsoft	Lossy software encoders and decoders	www.microsoft.com/windowsmedia , http://research.microsoft.com/~jiangli/portrait
MoMuSys	Lossy software encoders and decoders	www.tnt.uni-hannover.de/project/eu/momusys
MPEG4IP	Lossy software encoders and decoders	http://mpeg4ip.sourceforge.net
MPEG-LA	Consortium	www.mpeg-la.com

National Semiconductor (Mediamatics division)	Lossy software encoders and decoders	www.national.com
NeoMagic	Lossy hardware decoders	www.neomagic.com
NuCore Technology	Lossy hardware encoders and decoders	www.nucoretech.com
Nvidia	Lossy hardware decoders	www.nvidia.com
Office Noa	Lossy software encoders and decoders	www.nancy.co.jp
On2 Technologies	Lossy software encoders and decoders	www.on2.com, www.vp3.com
Oplayo	Lossy software encoders and decoders	www.oplayo.com
PacketVideo	Lossy software encoders and decoders	www.pv.com
Pegasus Imaging	Lossy software encoders and decoders	www.jpg.com
Philips	Lossy hardware and software encoders and decoders	www.mpeg-4.philips.com
Piranha	Lossy software encoders and decoders	www.piranha.com
Qualcomm	Lossy hardware encoders and decoders	www.qualcomm.com, www.technicalordigital.com
QuickSilver Technology	Lossy hardware encoders and decoders	www.quicksilvertech.com
QuVIS	Lossy hardware encoders and decoders	www.quvis.com
RAD Game Tools	Lossy software encoders and decoders	www.radgametools.com
RealNetworks	Lossy hardware encoders and decoders	www.reálnetworks.com
Rudiak-Gould (Huffyuv)	Lossless software encoders and decoders	http://math.berkeley.edu/~benrg
Sarnoff	Lossy software encoders, decoders, and analysis tools	www.sarnoff.com
Sci-Worx	Lossy hardware encoders and decoders	www.sci-worx.com
Sigma Designs	Lossy hardware encoders and decoders	www.sigmadesigns.com
Silicon Magic	Lossy hardware encoders and decoders	www.simagic.com
Silicon Motion	Lossy hardware decoders	www.siliconmotion.com
SMPTE (Society of Motion Picture and Television Engineers)	Consortium	www.smpte.org
SolidStreaming	Lossy software encoders and decoders	www.solidstreaming.com
Sonic Foundry	Lossy software encoders and decoders	www.sonicfoundry.com
Sony	Lossy software encoders and decoders	www.sony.com
Sorenson (and partner Apple)	Lossy software encoders and decoders	www.sorenson.com
SoundVision	Lossy hardware encoders and decoders	www.soundvision.com
STMicroelectronics	Lossy hardware encoders and decoders	www.st.com
Streambox	Lossy software encoders and decoders	www.streambox.com
Summus	Lossy software encoders and decoders	www.summus.com
TealPoint Software	Lossy software encoders and decoders	www.tealpoint.com
Teralogic	Lossy hardware encoders and decoders	www.teralogic-inc.com
Texas Instruments	Lossy hardware encoders and decoders	www.ti.com
3DSP	Lossy hardware encoders and decoders	www.3dsp.com
TMPGEnc	Lossy software encoders	www.tmpgenc.com
Toshiba	Lossy hardware encoders and decoders	www.toshiba.com
Total Multimedia	Lossy software encoders and decoders	www.tmmgroup.com
Trident Microsystems	Lossy hardware decoders	www.tridentmicro.com
TriMedia Technologies	Lossy hardware encoders and decoders	www.trimedia.com
Vianet	Lossy software encoders and decoders	www.vianet.com
ViDE (Video Development Initiative) MPEG-4 Working Group	Consortium	www.vide.net
VirtualDub	Lossy software decoders	www.virtualdub.org
Vweb	Lossy hardware encoders and decoders	www.vwebcorp.com
Wildform	Lossy software encoders and decoders	www.wildform.com
WIS Technologies	Lossy hardware (cores) and software encoders and decoders	www.wis-tech.com
WMF (Wireless Multimedia Forum)	Consortium	www.wmmforum.com
Xilinx	Lossy hardware encoders and decoders	www.xilinx.com
XiWave	Lossy software encoders and decoders	www.xiwave.com
XviD	Lossy software encoders and decoders	www.videocoding.de
Zeros & Ones	Lossy software encoders and decoders	www.zerosones.com
Zoran	Lossy hardware encoders and decoders	www.zoran.com
ZygoVideo		www.zygovideo.com

REFERENCES

1. Dipert, Brian, "Now hear this," *EDN*, Feb 3, 2000, pg 50.
2. Dipert, Brian, "Digital audio breaks the sound barrier," *EDN*, June 20, 2000, pg 71.
3. Dipert, Brian, "Digital audio gets an audition, Part 1: lossless compression," *EDN*, Jan 4, 2001, pg 48.
4. Dipert, Brian, "Digital audio gets an audition, Part 2: lossy compression," *EDN*, Jan 18, 2001, pg 87.
5. Dipert, Brian, "Hot & streamin'," *Commverge*, April 2000, pg 28.
6. Dipert, Brian, "A crash course in color conversion," *EDN*, June 7, 2001, pg 46.
7. Dipert, Brian, "Compression puts images on a diet," *EDN*, June 18, 1998, pg 71.
8. Dipert, Brian, "Video improvements obviate big bit streams," *EDN*, March 15, 2001, pg 83.
9. Dipert, Brian, "Video quality: a hands-on view," *EDN*, June 7, 2001, pg 83.
10. Dipert, Brian, "Industry standard or standstill?," *EDN*, July 5, 2001, pg 37.
11. Dipert, Brian, "Media security thwarts temptation, permits prosecution," *EDN*, June 20, 2000, pg 101.

QUEST FOR CLARITY

Usually about half of the research I do for an article ends up on the cutting-room floor. But this time it feels like I wasn't able to share 95% of what I read and heard. This situation is not necessarily bad: It gives me plenty of things to cover in future projects. But for those of you whose appetites are whetted, it might feel a bit like being served an appetizer and then having to leave before the main course arrives.

For the eager among you, then, here are some suggestions for continuing that education in a more leisurely fashion.

References A through E are some of the many excellent technical manuals I've reviewed over the years. **Reference A** covers the human-visual-system characteristics that perceptual-compression algorithms exploit. **Reference B** covers the codecs themselves in sufficient detail to

satisfy most enthusiasts. And for you deeply curious individuals, **references C through E** give all the nitty-gritty details of the algorithms, including the equations and the probability diagrams. I highly recommend *DV* magazine; it's well worth the subscription price.

For more information on the QuickTime and Windows Media codecs, see **references F and G**, respectively. Turning to the Internet, you can access a host of Web-site resources by performing a simple search on Google (www.google.com) using the keywords "video compression." Two highly recommended newsgroups are `comp.compression` and `comp.dsp`; you can peruse them the old-fashioned way with a newsgroup-reader program, such as Forté (www.fortinc.com) Agent, as I do, or through Google's Groups feature. Many of the vendor Web

sites that the **Web-only table** lists are also excellent sources of in-depth, albeit sometimes biased, information about video compression.

Much of my practical knowledge has come from e-mail discussion groups that Apple, Microsoft, and *Streaming Media* magazine host. Plenty of technology developers and implementers frequent these groups, and there's nothing like getting the real story from the folks in the trenches. To sign up for the Apple QuickTime list, visit www.lists.apple.com/mailman/listinfo/quicktime-talk. For Microsoft's WMTalk, go to <http://discuss.microsoft.com/archives/wmtalk.html>. And for the Streaming Media lists, head to www.streamingmedia.com/discussion.asp. (I particularly recommend the advanced tech list.)

REFERENCES

- A. Dipert, Brian, "Reference manual delivers on its vision," *EDN*, Dec 7, 2000, pg 32.
- B. Dipert, Brian, "Documentation demystifies 'd'video'," *EDN*, June 7, 2001, pg 26.
- C. Nelson, Mark and Jean-Loup Gailly, *The Data Compression Book*, M&T Books, 1996, ISBN 1-55851-434-1.
- D. Sayood, Khalid, *Introduction to data compression*, Morgan Kaufmann, 2000, ISBN 1-55860-558-4.
- E. Solari, Stephen J, *Digital video and audio compression*, McGraw-Hill, 1997, ISBN 0-07-059538-0.
- F. Apple Computer, *QuickTime for the Web*, Morgan Kaufmann, 2002, ISBN 1-55860-780-3.
- G. Microsoft, *Inside Windows Media*, Que, 1999, ISBN 0-7897-2225-9.