Available online at www.sciencedirect.com
SCIENCE DIRECT®
Signal Processing: *Image Communication* 19 (2004) 851–875
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Windows Media Video 9: overview and applications

Sridhar Srinivasan, Pohsiang (John) Hsu, Tom Holcomb, Kunal Mukerjee, Shankar L. Regunathan, Bruce Lin, Jie Liang, Ming-Chieh Lee, Jordi Ribas-Corbera*

Windows Digital Media Division, Microsoft Corporation, Redmond, WA 98052, USA

Abstract

Microsoft® Windows Media 9 Series is a set of technologies that enables rich digital media experiences across many types of networks and devices. These technologies are widely used in the industry for media delivery over the internet and other media, and are also applied to broadcast, high definition DVDs, and digital projection in theaters.

At the core of these technologies is a state-of-the-art video codec called Windows Media Video 9 (WMV-9), which provides highly competitive video quality for reasonable computational complexity. WMV-9 is currently under standardization by the Society of Motion Picture and Television Engineers (SMPTE) and the spec is at the CD (Committee Draft) stage.

This paper includes a brief introduction to Windows Media technologies and their applications, with a focus on the compression algorithms used in WMV-9. We present analysis, experimental results, and independent studies that demonstrate quality benefits of WMV-9 over a variety of codecs, including optimized implementations of MPEG-2, MPEG-4, and H.264/AVC. We also discuss the complexity advantages of WMV-9 over H.264/AVC.

Keywords: Video coding; Windows media; Codec standards

1. Introduction to windows media

Windows Media 9 Series is the latest generation of digital media technologies developed by Microsoft [13]. Although the origins of Windows Media focused on streaming compressed audio and video over the Internet to personal computers, the vision moving forward is to enable effective delivery of digital media through any network to any device.

1.1. A wide range of applications

Fig. 1 illustrates a variety of examples of how Windows Media technology is being used today. In addition to Internet-based applications (e.g., subscription services, video on demand over IP, web broadcast, etc.), content compressed with Windows Media codecs is being consumed by a wide range of wired and wireless consumer electronic devices (e.g., mobile phones, DVD players, portable music players, car stereos, etc.) [27]. Windows Media content can also be delivered to consumers in physical formats—for instance,

*Corresponding author.
E-mail address: jordir@microsoft.com (J. Ribas-Corbera).

0923-5965/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2004.06.005

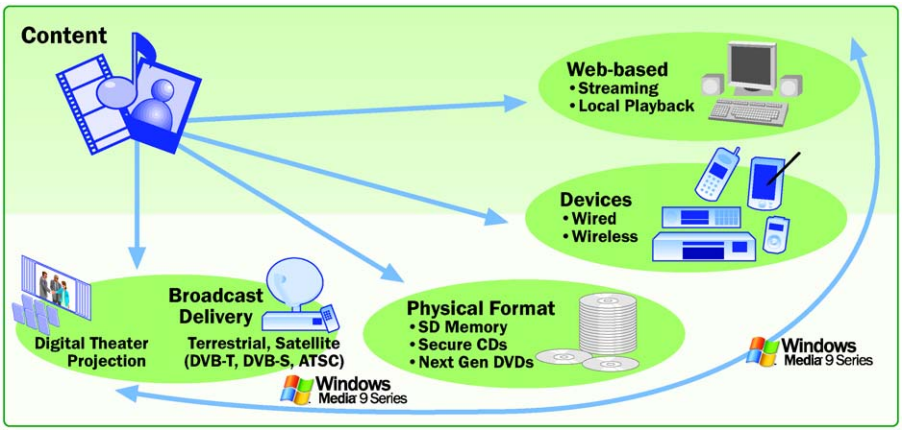

Fig. 1. Examples of current Windows Media technology applications.

using the secure digital (SD) memory card [12], or on CD or DVD using the HighMAT™ format [24]. Recently high definition (HD) movies (such as "Terminator 2" and "Step into liquid") have been released commercially on DVDs using Windows Media, and the DVD Forum has adopted the Windows Media Video 9 (WMV-9) codec as mandatory for the next generation HD DVD format.

In the terrestrial and satellite broadcast space, a recent project at the International Broadcasting Convention (IBC) demonstrated how to deliver Windows Media 9 Series content via DVB-T and DVB-S [21]. As another example, Windows Media technology is also used to compress movies in HD and multi-channel audio for projection in commercial theaters. For example, the popular "BMW Films" and independent films from the "Sundance Festival" were compressed and encrypted in Windows Media, distributed electronically, and projected digitally in numerous public theaters in the US.

1.2. End-to-end delivery

All of the applications mentioned thus far require a set of building blocks or components that permit the deployment of complete end-to-end solutions. The fundamental components of Windows Media 9 Series are illustrated in Fig. 2 and can be classified in three steps: authoring, distribution, and playback. In addition, digital rights management (DRM) is a key component that is distributed across the entire media data path.

1.2.1. Authoring

Authoring is the process of creating and encoding digital media. The basic encoding software provided by Microsoft is called Windows Media Encoder 9 Series. It is a flexible software encoder that can compress audio and video sources for live or on-demand streaming by using the Windows Media codecs.

At the same time, there are alternative encoding solutions provided by third parties that are built on top of the Windows Media porting kits (e.g., hardware encoders from companies such as Harmonic, Optibase, Tandberg Television, Texas Instruments, etc.) or the Windows Media software development kits (SDKs) (e.g., software encoders from companies like Accom, Adobe, Avid, Discreet, Quantel, etc.).

1.2.2. Distribution

The distribution of content compressed with Windows Media codecs over the Internet is generally done by a Windows Media Services server. Windows Media Services version 4.1 is an optional component in Windows 2000 Server, and Windows Media Services 9 Series is an optional component in Windows 2003 Server. The new server supports more features for advertising and corporate scenarios and is twice as scalable—it

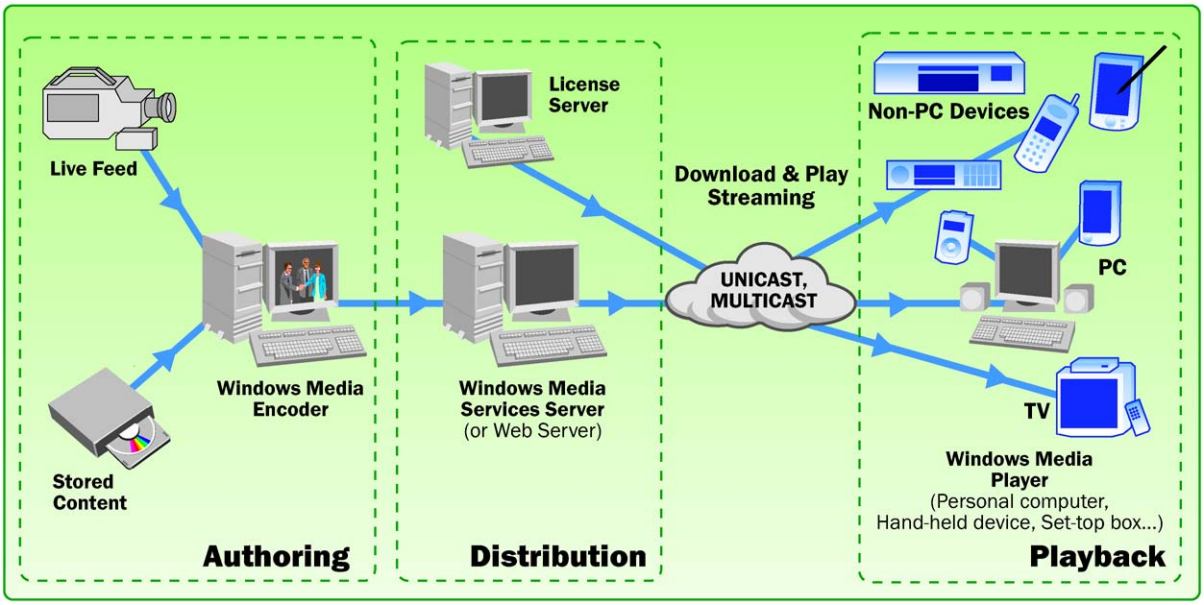

Fig. 2. End-to-end delivery of Windows Media content: authoring, distribution, and playback. The DRM system protects content based on policies set by the content owner.

doubles the number of customers that can receive a media clip at the same time.

A server can either stream the clip (transmit it with as little delay as possible) or download it (transmit and store it) over the internet into a user's playback device. The transmission of the clip can be performed live (for news, sports, concerts, or similar events) or on demand (for music videos, movies on demand, and so on). When streaming or transmitting a media clip, the server adapts its throughput and re-transmits lost packets intelligently, using feedback from the network quality metrics. For on-demand streaming, the latest server takes advantage of the additional bandwidth available (above the average bit rate of the clip) to reduce the start-up delay. In addition, such server reduces the likelihood of losing the connection (which manifests as playback glitches and re-buffering to the viewer) by sending more data to the playback device, so that the device can continue playing even when there is network congestion.

A robust and scalable server is essential for Internet delivery, but obviously having a solid network connection is also critical. The latter is addressed by content delivery networks (CDNs) such as Akamai, Digital Fountain, and SMC. The combination of robust servers and networks result in TV-like experiences that are remarkably superior to those of internet streaming in the recent past.

As mentioned earlier, one can also deliver Windows Media content through other means as in DVB-based transmission systems, or using physical formats such as CD or DVD.

1.2.3. Playback

The final step of the end-to-end delivery process is playback, which consists of decoding and rendering the compressed data in the user's device. On a personal computer, Windows Media Player and a variety of other players built by third parties (such as MusicMatch Jukebox or RealOne Player) can decode and play Windows Media streams and files.

As illustrated in Fig. 2, a wide variety of consumer electronic devices (over 600 devices, such as DVD players, CD players, personal digital

assistants, portable music players, car stereos, and so on) and chips (from manufacturers like ARM, ATI, Cirrus Logic, Equator, ESS Technology, LSI Logic, Texas Instruments, ST Microelectronics, Zoran, etc.) can decode Windows Media content as well [27]. As in the encoder case, third parties can build such hardware playback devices on any platform, without or with any operating system (such as Linux, Mac, Windows, …), by using Windows Media porting kits.

1.2.4. Digital rights management

A critical component of the end-to-end delivery is DRM, which we represent by the license server in Fig. 2. DRM crosses boundaries between the three steps of the media delivery system, so we will discuss it separately in this subsection.

The DRM technology used by Windows Media lets owners encrypt their media products and services and specify the usage rules and policies. For example, the owner may decide that the user can play the digital media until a certain date, or can play it a given number of times, or the owner can let the user copy the digital media to a certain number (and type) of devices.

In the typical Internet scenario, the content owner encrypts the (compressed) digital media stream using DRM. When a viewer selects the stream, the playback device connects to the license server, which offers a license for the content. The viewer then decides whether to accept the terms and price of the license and, if so, the license is downloaded into the viewer's device. The viewer is then able to decrypt and play the content according to the terms of the license.

Designing a complete DRM service is a challenging project. The system needs to be secure (with the capability of upgrading quickly), flexible (while accommodating the desires of end users, content owners as well as device manufacturers), and user-friendly. Windows Media DRM addresses the requirements of many scenarios involving DRM, and thus offers a competitive solution that is widely used by major Hollywood studios (e.g., in movie services such as MovieLink, CinemaNow, etc.) and music labels (e.g., in music services like Napster, MusicMatch, etc.) to secure their content.

2. Windows Media Video 9: overview

Windows Media 9 Series includes a variety of audio and video codecs, which are key components for authoring and playback of digital media. The Windows Media Video 9 (WMV-9) codec is the latest video codec in this suite and is based on technology that can achieve state-of-the-art compressed video quality from very low bit rates (such as 160×120 at 10 Kbps for modem applications) through very high bit rates (1280×720/1920×1080 at 4–8 Mbps for high-definition video, and even higher bit rates for mastering). This section describes in detail the overall structure of the WMV-9 codec, and covers the key innovations critical for its good performance.

2.1. Structure of the codec

The internal color format for WMV-9 is 8-bit 4:2:0. The codec uses a block-based motion compensation and spatial transform scheme which, at a high level, is similar to all popular video compression standards since MPEG-1 and H.261. Broadly speaking, these standards—as well as WMV-9—perform block-by-block motion compensation from the previous reconstructed frame using a two-dimensional quantity called the motion vector (MV) to signal spatial displacement. A prediction of the current block is formed by looking up a same-sized block in the previous reconstructed frame that is displaced from the current position by the motion vector. Subsequently, the displaced frame difference, or residual error, is computed as the difference between the actual block and its motion-compensated prediction. This residual error is transformed using a linear energy-compacting transform then quantized and entropy coded.

On the decoder side, quantized transform coefficients are entropy decoded, dequantized and inverse transformed to produce an approximation of the residual error, which is then added to the motion-compensated prediction to generate the reconstruction. The high level description of the codec is shown in Fig. 3. Needless to say, the above description only provides a high level overview and does not discuss implementation

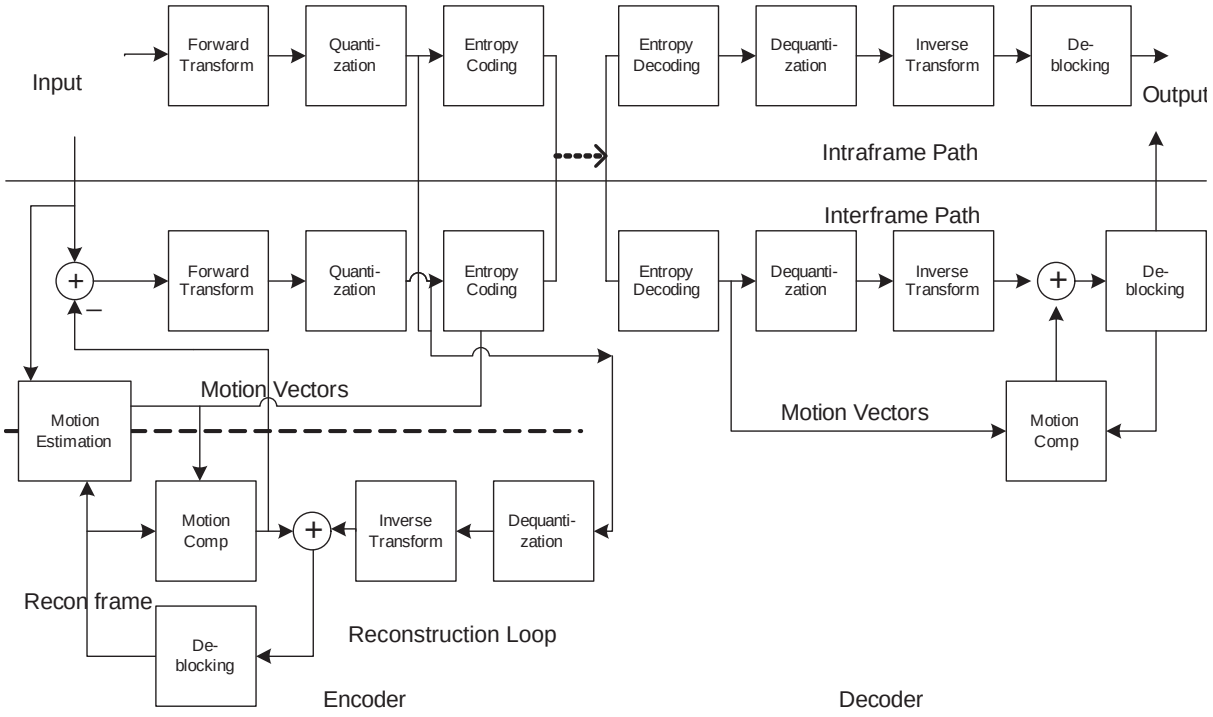

Fig. 3. Block diagram of the WMV-9 codec.

details. The remainder of this section describes, in depth, the innovations in WMV-9 that distinguish it from other competing video coding solutions such as the MPEG standards [6,7,8].

WMV-9 has intra (I), predicted (P) and bidirectionally predicted (B) frames. Intra frames are those which are coded independently and have no dependence on other frames. Predicted frames are frames that depend on one frame in the past. Decoding of a predicted frame can occur only after all reference frames prior to the current frame starting from the most-recent I frame have been decoded. B frames are frames that have two references—one in the temporal past and one in the temporal future. B frames are transmitted subsequent to their references, which means that B frames are sent out of order to ensure that their references are available at the time of decoding. B frames in WMV-9 are not used as a reference for subsequent frames. This places B frames outside of the decoding loop, allowing shortcuts to be taken during the decoding of B frames without drift or long-term visual artifacts. The above definition of I, P and B frames holds for both progressive and interlaced sequences.

2.2. Metrics and design issues

The key performance metric that video codecs are measured against is the rate–distortion (R–D) plot. The R–D plot is a two-dimensional curve that indicates the distortion suffered by the process of compression, at a certain target bitrate. Alternatively, it indicates the required bitrate necessary to code a sequence at a certain level of distortion. The most widely used distortion metric is the peak signal-to-noise ratio (PSNR), mainly because it is simple to compute. PSNR is not an accurate measure of subjective quality though [19], which is why subjective evaluation of a video codec is a very important stage in its development. Understanding of codec metrics is central to a discussion of codec tools, because some tools used in WMV-9

are more critical to subjective improvement of quality than to the R–D plot.

Apart from quality of compression, the computational complexity required to achieve this level of compression is a key factor that can determine the success or failure of a video codec. WMV-9 requires a bit exact reconstruction at the decoder in order to prevent any drift between the decoder and the reconstruction loop of the encoder. Since frames are predicted from prior reconstructions, any small drift or mismatch between encoder and decoder has the potential to grow into a visually annoying artifact in a matter of a few seconds of video. Moreover, non-linear operations such as loop filtering tend to accelerate the buildup of error, further increasing the susceptibility of the system to mismatch between the encoder and decoder, and hastening the production of visible errors. As a result, it is required that all decoders must produce the same result which is the result produced by the encoder reconstruction loop, shown in the lower section of Fig. 3. This places a minimum computational requirement on the decoder—it is no longer possible for decoders to perform any approximations or use reduced precision in I and P frames.

Floating point arithmetic is ruled out on the decoder side for several reasons, the important ones being the need to minimize decoder complexity, and the need to implement decoders that precisely match the specification so as to avoid mismatch. Floating point operations are not very portable across processors—their definitions usually involve some measure of tolerance, making them unsuitable for perfectly matching implementations. It is largely accepted that low-precision integer arithmetic (usually 16-bit word size) is a desirable feature. Wherever possible, single instruction multiple data (SIMD) type operations are preferred since these can be carried out in parallel both on programmable processors as well as on application specific hardware. Conditional statements are not very implementation friendly and it is preferred that their use be minimized.

Broadly speaking, the operations in a video codec can be classified as *signal processing* operations and *entropy coding* operations. The former include motion compensation, transform, loop filtering and other steps which directly operate on the pixel values. The latter category covers the coding technique used to lay out the encoded bits, and also related operations such as zigzag scanning, motion vector prediction, etc. Any inefficiency in signal processing operations tends to have a big impact on the R–D plot at high rates, whereas any inefficiency in entropy coding operations usually has more of a significant impact on the low rate R–D plot. These are guidelines that help in designing a good video codec. In order to operate well across a variety of rates (down from "internet rate" of less than 1 bit/pixel/s to "backhaul rate" of well over 50 bits/pixel/s) it can be seen that equal attention must be paid to all operations.

2.3. Innovations

WMV-9 addresses R–D quality and visual performance using a variety of techniques. The important ones that distinguish WMV-9 from MPEG standards are:

1. Adaptive block size transform,
2. Limited precision transform set,
3. Motion compensation,
4. Quantization and dequantization,
5. Advanced entropy coding,
6. Loop filtering,
7. Advanced B frame coding,
8. Interlace coding,
9. Overlap smoothing,
10. Low-rate tools,
11. Fading compensation.

In addition, WMV-9 uses many techniques and approaches for encoding side information, including motion vectors, B frame related quantities, fading parameters etc. A discussion of all of these is outside the scope of this paper.

Each innovation included in WMV-9 is a result of research, controlled experiments to determine performance and a thorough complexity analysis. It is not possible, within the scope of this paper, to fully explore experimental results and theoretical justifications for all the innovations. In the following section, we describe each innovation at

the level of detail necessary to get an understanding of the fundamentals, without deluging the reader with excessive information. Since the blocks of a video codec are tuned as a whole, experimental results showing the performance comparison of each individual component of this codec with that of similar individual components of another codec may not be that meaningful. The only truly meaningful result is an overall performance analysis of WMV-9 with respect to existing codec standards. This is provided in Section 4.

3. Windows Media Video 9: detailed description

This section describes, in some detail, the salient innovations in WMV-9 that are listed in Section 2.3.

3.1. Adaptive Block Size Transform

Traditionally, 8×8 transforms have been used for image and video coding [6,14,19]. The 8×8 size has the advantages of being dyadic, large enough to capture trends and periodicities while being small enough to minimize spreading effects due to local transients over the transform area. There is no clear evidence pointing to the advantage of going to a transform with larger support (except possibly in a multi-resolution framework). However, H.264/AVC uses a 4×4 transform and has the claimed advantage of reducing ringing artifacts at edges and discontinuities [8].

It is known that smaller transforms are better in areas with discontinuities because they produce fewer ringing artifacts [5,15,17]. However, trends and textures, especially periodic textures, are better preserved with the transforms having a larger support. WMV-9 takes the approach of allowing 8×8 blocks to be encoded using either one 8×8, two horizontally stacked 8×4 s, two vertically stacked 4×8 s, or four 4×4 block transforms, as shown in Fig. 4. This allows WMV-9 to use the transform size and shape that is best suited for the underlying data. The specific transform configuration used must be signaled as part of the bitstream. This signaling is performed in an efficient manner as well, as outlined below.

Signaling of transform type in a WMV-9 stream is possible at the frame, macroblock or block level. If the signal is sent at the frame level, all blocks within the frame use the same transform type. Likewise, if the signal is sent at the macroblock level, all blocks within a macroblock (there are six 8×8 blocks in all, including four luminance and two chrominance blocks) use the same transform type. Block level signaling is specific to the current block. Macroblock and block level signaling can be mixed across macroblocks within one frame. This allows for coarse and fine level specification of the transform type, which is useful when the data is non-stationary. Frame level signaling helps in low-rate situations where the transform type coding overhead may be excessive, if transform type is sent at the macroblock or block level.

When macroblock or block level signaling is used there is a possibility of saving a few bits in static or perfectly predicted areas. When for the chosen transform type all quantized transform coefficients over a macroblock or block are zero, there is no need to send the transform type information, as all varieties of inverse transform will produce all-zero blocks for an all-zero input. This allows the overhead to be reduced for static areas or areas that can be generated purely by motion compensation, and is a key factor for improved performance at low rates or for low-motion sequences such as talking heads.

Intra frames and intra blocks/macroblocks in predicted frames use 8×8 transforms.

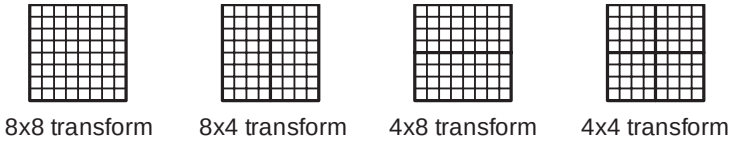

Fig. 4. Transform sizes allowed in WMV-9.

In the following paragraph, we present an informal proof to demonstrate the ability of larger block sizes to better retain texture information. Consider a one-dimensional signal with a periodic texture component $a(n) = (-1)^n$, which is a signal that alternates between $+1$ and -1. A 4-point DCT applied to this signal produces the odd frequency components [0.77 1.85] (even frequency components are identically zero). An 8-point DCT applied to the same signal produces the odd frequency components [0.51 0.60 0.90 2.56]. The largest coefficient of the 8-point DCT is around $\sqrt{2}$ times that of the 4-point DCT. In two dimensions, this factor gets squared. Subtle textures that get quantized out in smaller transform sizes have a higher chance of survival when a larger transform is used. A similar argument can be made in favor of larger transforms for accurately preserving flat areas and long-term trends.

Purely on the rate-distortion side, the advantage of allowing multiple transform sizes is significant though not huge. However, from the point of subtle texture preservation (such as keeping film details and grain noise), adaptive transforms provide major subjective quality benefits, as demonstrated recently by work of other researchers [4,25]. In fact, we believe that our adaptive transform is among the technologies responsible for the apparent subjective superiority of WMV-9 over H.264/AVC and other fixed-transform codecs in the independent tests discussed in Section 4.

3.2. 16 Bit transforms

The four transform sizes used by the adaptive block transform described above makes the task of designing the WMV-9 transform a challenge. The most important motivation for the transform design is to minimize computational complexity of the decoder. An inverse transform that can be implemented in 16-bit fixed point arithmetic significantly reduces the computational complexity of the decoder, compared to a transform requiring 32-bit or floating point arithmetic. First, 32-bit operations are expensive to implement on 16-bit processors which form a large percentage of commonly used DSPs. Second, if the result of two 16-bit multiplies can be restricted to be in the 16-bit range, additional speedup can be obtained by simultaneously computing two 16-bit operations on a 32-bit register in parallel with clever programming, when 32-bit arithmetic is available. Finally, twice the number of 16-bit operations can be performed per cycle over 32-bit operations for SIMD operations such as those defined by MMX® on the Pentium®.

The WMV9 transforms are designed to meet a list of constraints enumerated below. The transforms are separable, which allows the constraints to be defined for each one dimensional transform stage. The constraints for both the one-dimensional 4- and 8-point transforms are

1. Transform coefficients are small integers.
2. The transform is a 16-bit operation—where both sums and products of two 16-bit values produce results within 16-bits.
3. Forward and inverse transforms form an orthogonal pair. This is defined as follows: let U and V be the forward and inverse transform matrices, then U and V are biorthogonal if V U = diag(D). In addition, V and U are orthogonal when $V = U'$. It is not required for their product to be an identity matrix, any diagonal matrix is a valid product for orthogonality or biorthogonality—any scaling introduced by the non-unity diagonal elements of the product can be accounted for during quantization.
4. The transform approximates a DCT, which is known to have favorable energy compaction properties on intra and residual video data.
5. Norms of basis functions within one transform type are identical so as to eliminate the need for any coefficient-indexed renormalization in the dequantization process.
6. Norms of basis functions between transform types are identical. In other words the 4- and 8-point basis functions have the same norm. This allows us to use the same quantization parameter between various transform types while maximizing the rate-distortion performance.

The WMV9 inverse transform consists of an 8×8 transform, a 4×8 transform, an 8×4 transform, and a 4×4 transform. The range expansion associated with any N point DCT-like transform $\sqrt{N}$, which is larger than that of a M point transform when $N > M$. Among all these transforms, the 8×8 inverse transform places the tightest constraint on range of integer transform coefficients, and it is impossible to find an 8×8 transform that satisfies all of these constraints simultaneously.

The key innovation in WMV-9 is to mildly relax constraints 5 and 6, allowing the basis of the transform coefficients to be very close, though not identical, in norm. These small discrepancies between basis function norms are accounted for entirely on the encoder side with no loss in compression efficiency.

The transform matrices for a one-dimensional 8-point inverse transformation and a one-dimensional 4-point inverse transformation are presented in Figs. 5 and 6. The norms of basis functions in one dimension, in ascending numbers, are in the ratio 288:289:292, which satisfies constraints 5 and 6 to within 1%.

Although the 4- and 8-point transforms are essentially described by the transform matrices, their two-dimensional implementations have additional restrictions. First, the rows of the dequantized transform coefficients are inverse transformed. This is followed by a rounding operation which is followed by inverse transformation of the columns, followed by another rounding operation. In order to operate within 16-bits with sufficient headroom and maximum accuracy, rounding is distributed between the horizontal and vertical transforms. In addition, the second transform stage exploits the specific transform matrix entries to achieve an extra bit of precision. The output of the inverse transform is in the range 10 bits which allows for headroom for quantization error beyond the theoretically possible 9 bits.

$$T_8 = \begin{bmatrix} 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 16 & 15 & 9 & 4 & -4 & -9 & -15 & -16 \\ 16 & 6 & -6 & -16 & -16 & -6 & 6 & 16 \\ 15 & -4 & -16 & -9 & 9 & 16 & 4 & -15 \\ 12 & -12 & -12 & 12 & 12 & -12 & -12 & 12 \\ 9 & -16 & 4 & 15 & -15 & -4 & 16 & -9 \\ 6 & -16 & 16 & -6 & -6 & 16 & -16 & 6 \\ 4 & -9 & 15 & -16 & 16 & -15 & 9 & -4 \end{bmatrix}$$

Fig. 5. Matrix for one-dimesional 8-point inverse transform.

$$T_4 = \begin{bmatrix} 17 & 17 & 17 & 17 \\ 22 & 10 & -10 & -22 \\ 17 & -17 & -17 & 17 \\ 10 & -22 & 22 & -10 \end{bmatrix}$$

Fig. 6. Matrix for one-dimesional 4-point inverse transform.

3.3. Motion compensation

Motion compensation is the process of generating a prediction of a frame of video by displacing the reference frame. Typically, the prediction is formed for a block (defined as an 8×8 tile) or macroblock (16×16 tile) of data. Also, the displacement is typically rectilinear and constant over the entire tile being predicted. Such a displacement is defined by the motion vector (MV) with two components corresponding to the shift along the X and Y directions. MV components are usually specified in terms of pixel displacements, often with sub-pixel accuracy. Sub-pixel displacements are realized by filtering the reference frame using appropriately defined interpolation filters.

Efficiency of a video codec is closely related to the ability of the motion compensator to generate a good set of predictors. It has been shown that sub-pixel resolution plays a substantial role in this process although the gains of going to finer pixel resolutions are offset by the increased cost of coding motion vectors to higher degrees of precision [18]. Motion vector resolution is either $\frac{1}{2}$ or $\frac{1}{4}$ pixel in existing profiles in standard video codecs. WMV-9 allows a maximum resolution of $\frac{1}{4}$ pixels. In other words, fractional shifts or motion vectors of $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{3}{4}$ pixels are allowed. At low rates, higher precision in motion vectors is a liability since the percentage of bits used in coding motion vectors is significant. Thus, WMV-9 allows

for a lower precision motion vector resolution that is signaled at the frame level.

The second factor influencing the ability to generate good predictors is the size of the predicted area. Typically in the older formats, a single motion vector is used for a macroblock, which is a 16×16 pixel area in the luminance plane. MPEG-4 allows definition of motion vectors for 16×16 or 8×8 blocks; this choice is made for each macroblock being coded. In other words, a macroblock is predicted from four motion vectors. H.264/AVC permits motion vectors to reference areas as small as 4×4. While this level of freedom can be useful at high bitrates, smaller areas impose higher computational overhead on the codec. On the decoder side, motion compensation is usually the most significant computational component. Smaller blocks with randomly distributed motion vectors cause increased cache access, and need more filtering steps on a per-pixel basis. WMV-9 follows a middle of the road approach. 16×16 motion vectors are used by default, but 8×8 motion vectors are permitted for frames which are signaled as containing mixed motion vector resolution.

Finally, the filter used for generating sub-pixel predictors is the third key determinant of the quality of motion compensation. Shorter filters are computationally simpler but have poor frequency response and are adversely influenced by noise. Longer filters referencing more pixels are computationally more difficult to implement. Moreover, images have strong local and transient characteristics that tend to get blurred with long filters. WMV-9 trades off these considerations by using two sets of filters for motion compensation. The first is an approximate bicubic filter with four taps, and the second is a bilinear filter with two taps.

Further, WMV-9 combines the motion vector modes derived from the three criteria (MV resolution, size of predicted area and filter type) into a single mode. This combined mode is one of the following:

1. Mixed block size (16×16 and 8×8), $\frac{1}{4}$ pixel, bicubic,
2. 16×16, $\frac{1}{4}$ pixel, bicubic,
3. 16×16, $\frac{1}{2}$ pixel, bicubic,
4. 16×16, $\frac{1}{2}$ pixel, bilinear.

The combined motion vector mode is signaled at the frame level. In general, higher bitrates tend to use the modes at the top of the list and vice versa. Only when mode 1 is chosen is the predicted block size indicated on a macroblock basis. The consolidation of these three criteria into one leads to a more compact decoder implementation, with no significant performance loss.

3.3.1. Sub-pixel filters

In the existing codec standards, sub-pixel interpolation in two dimensions is performed by filtering in one dimension, rounding and clamping the intermediate value back to the input range of 8 bits, followed by filtering in the second direction, rounding and clamping. It is possible to achieve additional accuracy by retaining a higher precision result after the first stage of filtering. Another advantage is that since the clamping operation is non-linear it may be more difficult to implement on certain processors (though clamping can be done easily on hardware and specific DSPs).

WMV-9 exploits this observation by defining the two-dimensional filtering process as follows. First, filtering is performed in the vertical direction. Then, a rounding factor is added to the result and some bits are shifted out. The intermediate result is possibly outside of the [0 255] range. Next, the intermediate result is filtered in the horizontal direction. Finally, a second rounding parameter is added to the result which is shifted and clamped to within the range [0 255]. The two shifts are chosen so as to (a) add up to the required shift for normalizing the filters and (b) to allow for a 16-bit implementation—where the intermediate values in the second filtering operation are within 16-bits.

The four-tap bicubic filters used in WMV-9 for $\frac{1}{4}$ and $\frac{1}{2}$ pixel shifts are: $[-4\ 53\ 18\ -3]/64$ and $[-1\ 9\ 9\ -1]/16$.

3.3.2. Chrominance channel

Since chrominance motion vectors are implicitly derived from co-located luminance motion vectors, their accuracy is limited and offers scope for simplification. Also, the chroma channels have a

strong low-pass component. WMV-9 uses bilinear filters for chroma motion interpolation. In general, chroma motion vectors are obtained by dividing the co-located luminance motion vectors by 2 and rounding the result to a $\frac{1}{4}$ pixel position. In addition, there is a sequence level 1 bit field that controls the rounding of chroma motion vectors. If this bit is set, then the chroma motion vectors that are at quarter pixel offsets are rounded to the nearest full pixel positions—in effect only allowing $\frac{1}{2}$ and full pixel locations for chroma motion vectors. The purpose of this mode is speed optimization of the decoder.

The motivation for this optimization is the significant difference between the complexities of interpolating pixel offsets that are at (a) integer pixel; (b) half pel; (c) at least one coordinate (of x and y) at a quarter pel; and (d) both coordinates at quarter pel positions. The ratio of a:b:c:d is roughly 1:4:4.7:6.6. By applying this mode one can favor (a) and (b), thus cutting down on decoding time. Since this is done only for chroma interpolation, the coding and quality loss (especially subjective quality) are both negligible.

3.4. Quantization and dequantization

Quantization and dequantization of transform coefficients are key steps that can critically affect the rate–distortion performance of a video codec. Strictly speaking, a codec standard only defines the dequantization process. For the purpose of this section, however, we will interchangeably refer to both dequantization and quantization with the understanding that although quantization is an encoder choice, any reference to quantization is a reference to the "intended" quantization that is matched to the specific dequantization rule. WMV-9 uses multiple transform sizes, but the same quantization rules apply to all coefficients. Scalar quantization, whereby each transform coefficient is independently quantized and coded, is used. At a high level, this process is similar to the corresponding process in MPEG standards.

The earlier standards such as MPEG-2 use uniform quantization with a "dead-zone". In uniform quantization, quantization intervals are identical. A dead-zone is an interval on the number line around zero, such that unquantized coefficients lying in the interval are quantized to zero. All quantization intervals except the dead-zone are of the same size—the dead-zone being typically larger. The use of a dead-zone leads to substantial bit savings at low bitrates. From the decoder side, the reconstruction levels are given by the set $\{-kQ - D, 0, kQ + D\}$, where k, $D > 0$ (D is $Q/2$ for MPEG-4). The quantization intervals and dequantization points (i.e. reconstruction levels) are shown in Fig. 7(a). In regular uniform dequantization, the reconstruction levels are all equally spaced, i.e. the reconstruction set is $\{kQ\}$, where k is an integer. In practice, a dead-zone may be used on the encoder side, but this is not revealed by the reconstruction set. This type of quantization, shown in Fig. 7(b), performs very well at high bitrates.

WMV-9 allows both dead-zone and regular uniform quantization, i.e. both variations shown in Fig. 7 are possible. The specific type of quantization is signaled at the frame level and the appropriate dequantization rule is applied to all coefficients within the frame by the decoder. On the encoder side, a quantization parameter based rule allows for an automatic switch from regular uniform quantization to dead-zone uniform quantization as the parameter increases. Although this rule works well across a variety of sequences, other factors such as the noise within a sequence and rate control parameters may be used in more sophisticated encoders to fine-tune this change-over. Allowing for the rule to switch between dead-zone and regular uniform quantization is a key factor in the superior performance of WMV-9 at both low and high bitrates.

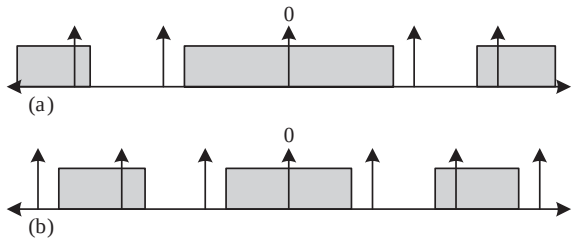

Fig. 7. WMV-9 quantization and dequantization rules showing (a) dead-zone and (b) regular uniform quantization—arrows are reconstruction levels, and gray boxes are recommended quantization bins (for alternate intervals).

Subsequent to quantization, the two-dimensional transform coefficients are reordered into a linear array. This is done by means of a scanning process. Multiple scan arrays are used for this reindexing process. These arrays are tuned, based on edge orientations, to cluster non-zero coefficients efficiently in macroblocks of the frame. The scan array used is inferred from causal data.

3.5. Advanced entropy coding

WMV-9 uses multiple tools for the efficient encoding of symbols into the bitstream, or entropy coding. Across a wide range of rates, the bulk of the bit usage is taken up in the coding of quantized transform coefficients. Although WMV-9 uses simple variable length codes for encoding this information, a high degree of efficiency is achieved by allowing the use of multiple code tables for encoding each alphabet. One out of a possible set of code tables is chosen for a particular alphabet and this is signaled in the bitstream. This set is itself determined by the frame level quantization parameter. A finer level switch making use of causal contextual information is possible in theory, but considerations of computational ease have ruled out this possibility in WMV-9.

Information different from the quantized transform coefficients may often account for a significant portion of the total bit usage. This information includes motion vectors, coded block and subblock patterns, intra/predicted macroblock switch, macroblock coding modes including transform type, motion vector resolution etc. Motion vectors and coded block patterns are themselves entropy coded using one of multiple code tables. Some information such as the motion vector resolution, skip macroblock and frame/field switch is represented as a bitplane with each bit signifying the corresponding value for a macroblock. An efficient method of encoding such bitplanes is also included in the syntax of WMV-9. These, and other innovations are used for efficient entropy encoding in WMV-9.

3.6. Loop filtering

Quantization error in the intra-coded blocks, and in the residuals of the motion-compensated (inter-coded) blocks can induce discontinuities at the block boundaries. These discontinuities result in two undesirable consequences: (1) The discontinuities can show up as visible 'blocky' artifacts in the decompressed video frames, especially in smoothly textured regions and (2) The quality of the reconstructed frame as a predictor for future frames is reduced. To mitigate these effects, the WMV-9 scheme uses an in-loop deblocking filter to attempt to remove the block-boundary discontinuities. The filtering is performed on the reconstructed frame prior to its use as a reference frame for the subsequent frame(s). Therefore, the encoder and decoder must perform the same filtering operation.

Since the intent of loop filtering is to smooth out the discontinuities at block boundaries, the filtering process operates on the pixels that border neighboring blocks. For P and B pictures, the block boundaries can occur at every 4th, 8th, 12th, etc pixel row or column depending on whether an 8×8, 8×4, 4×8 or 4×4 transform is used. For I pictures filtering occurs at every 8th, 16th, 24th, etc pixel row and column since only 8×8 transforms are used.

Fig. 8 shows the pixels that are involved in a filtering operation. In this case, a vertical boundary is being filtered. The columns containing P4 and P5 represent the boundary between two adjacent transform blocks. The figure shows the pixels that are involved in filtering one set of boundary pixels along the vertical boundary. As the figure shows, eight pixels are involved in the filter computation, four on each side of the block boundary. The filtering process involves

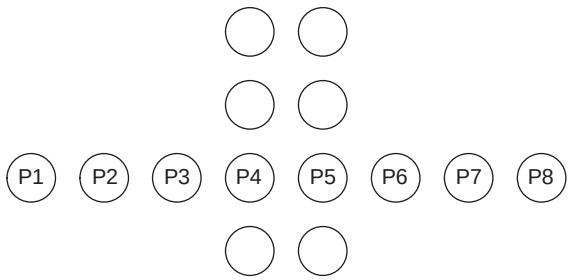

Fig. 8. Pixels used in loop filtering operation—only pixels P4 and P5 may be modified.

a discontinuity measurement involving pixels P1 through P8 which seeks to detect whether the discontinuity is above a certain threshold. This measurement attempts to be more sensitive to discontinuities that occur in smoothly textured regions since these are more visually objectionable and because these tend to have a more adverse effect on the predictive quality of the reconstructed frame. If a discontinuity is detected then an averaging operation is performed on pixels P4 and P5 in an attempt to smooth the discontinuity.

For P pictures, blocks may be intra or inter-coded. Intra-coded blocks always use an 8×8 transform to transform the samples and the 8×8 block boundaries are always filtered. Inter-coded blocks may use an 8×8, 8×4, 4×8 or 4×4 inverse transform to construct the samples that represent the residual error. Depending on the status of the neighboring blocks, the boundary between the current and neighboring blocks may or may not be filtered. The decision of whether to filter a block or subblock border is as follows:

1. The boundaries between coded (at least one non-zero coefficient) subblocks (8×4, 4×8 or 4×4) within an 8×8 block are always filtered.
2. The boundary between a block or subblock and a neighboring block or subblock is not filtered if both have the same motion vector and both have no residual error (no transform coefficients). Otherwise it is filtered.

This prevents over-smoothing block boundaries where quantization or motion compensation induced discontinuities are unlikely to occur.

Due to the various condition checks involved, loop filtering is a computationally expensive process. WMV-9 uses a shortcut to reduce computations. Determination of whether to smooth across an edge or not is made only once every four pixels. For instance, in Fig. 8, this step is performed only for the numbered pixels. The four pixels above P4 and P5, and the two pixels below P4 and P5 use the same determination thus avoiding multiple computations. This shortcut helps speed up the loop filtering process with little rate–distortion or visual detriment.

3.7. Interlace coding

Interlaced video content is prevalent in the television broadcasting industry. This type of content has a special structure that a video codec can take advantage of, to improve compression. Each interlaced frame contains data from two different time instants where all of the even lines (top field) are from one time instant and all of the odd lines (bottom field) are from another time instant that is different from the time instant of the top field lines. Fig. 9 shows the relationship between the chrominance and luminance positions for a 4:2:0 interlace frame.

3.7.1. Field picture coding

In field picture coding, the two fields that make up a frame are coded separately. A field is divided into macroblocks that can be either intra- or inter-coded. An intra-coded macroblock in a field is coded in the same manner as progressive picture intra-macroblock coding. Inter-coded macroblocks may contain either one (16×16) or four (8×8) motion vectors. Each motion vector can refer to either one of two previously encoded fields

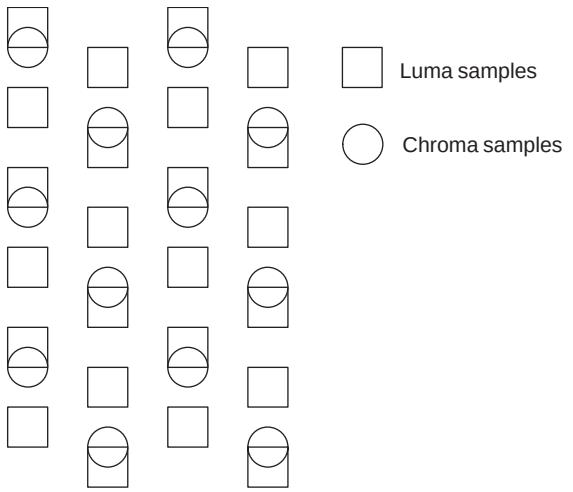

Fig. 9. There are two types of interlace picture coding mode supported by WMV-9 namely field picture coding, and frame picture coding. 4:2:0 Luma and chroma temporal and vertical sample positions (where from left to right is shown a top field, bottom field, top field, and bottom field).

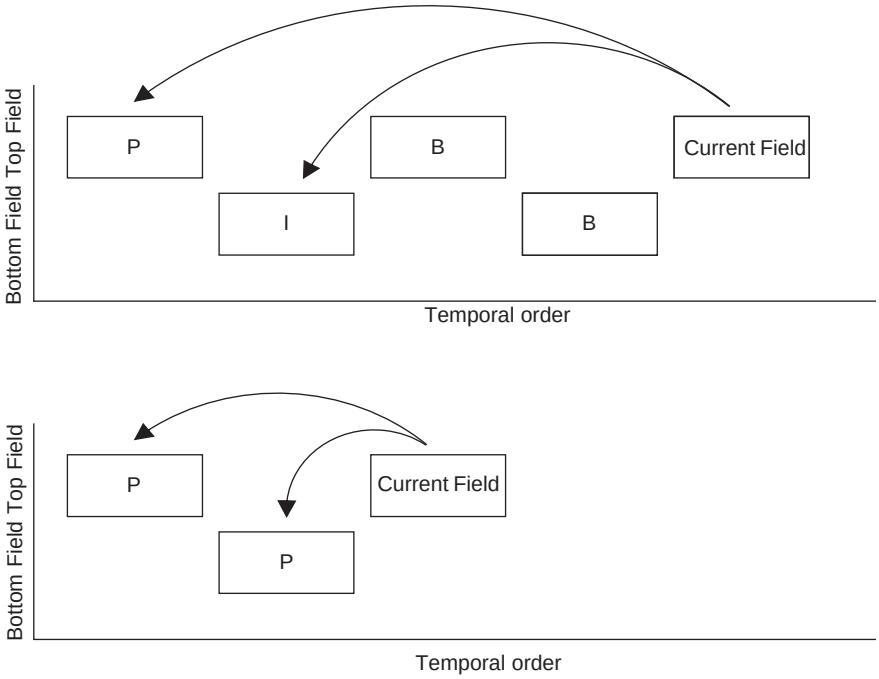

Fig. 10. Field picture reference.

for motion compensation as shown in Fig. 10. When one motion vector is used, the entire macroblock is compensated from one previous field. On the other hand, when four motion vectors are used, each of the four luminance blocks may be compensated from different fields.

3.7.2. Frame picture coding mode

In frame picture coding, both fields in an interlaced frame are coded jointly. A frame is divided into macroblocks that may be either intra-coded or inter-coded. Each macroblock contains samples from two time instants (i.e. 8 lines of top field and 8 lines of bottom field). For an intra-coded macroblock, the encoder has the option to re-order the luminance portion according to fields to try to increase the spatial correlation prior to transform coding. Fig. 11 shows the luminance portion of a re-ordered macroblock.

An inter-coded macroblock may be motion compensated in one of two modes. In frame motion compensation mode, each macroblock is motion compensated without regards to the field structure and it is similar to progressive coding where the macroblock is compensated using either one motion vectors (16×16) or four motion vectors (8×8). In field motion compensation mode, the fields inside a macroblock are compensated separately using two field motion vectors or four field motion vectors. When two field motion vectors are used, the top field lines inside the macroblock are compensated using one motion vector and the bottom field lines inside the macroblock are compensated using the other motion vector. On the other hand, when there are four field motion vectors, each field inside the macroblock is compensated using two motion vectors. More specifically, each field is further sub-divided into two 8×8 blocks and each block has its own motion vectors. After motion compensation, the residual can also be re-ordered in the same way as the intra macroblock coding prior to transform, and the re-ordering process is independent of the type of the motion compensation used.

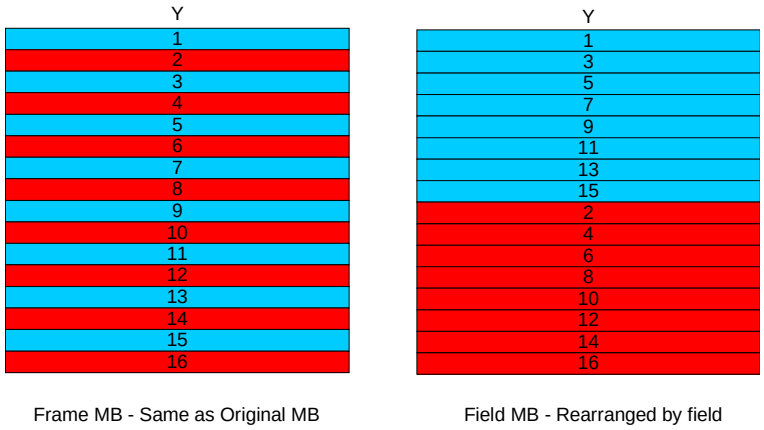

Fig. 11. Luminance macroblock structure for frame picture.

3.8. Advanced B-frame coding

Video codecs use bi-directional or B-frames, which employ motion estimation and compensation from both forward and backward directions. Broadly speaking, the operation of B-frames in WMV-9 is broadly similar to that in MPEG in the sense that it includes multiple prediction modes. The key innovations that differentiate WMV-9 are motion vector prediction for higher efficiency, explicit inclusion of timing for scaling of motion vectors, and prediction of B-fields in interlaced video coding. WMV-9 employs some new algorithms to make B-frames more efficient at reducing the bit rate by means of the following:

1. Explicit coding of the B frame's temporal position relative to its two reference frames. This need not be the true or time-stamped temporal position, but any fraction that the encoder chooses so as to pick the best scaling of motion vectors with the direct mode. This effectively removes time-stamp dependency, and generalizes the direct mode prediction model from a constant velocity model to variable velocity model, thus aligning it more closely with the kind of motion that is actually present in a typical frame sequence.
2. Intra coded B frames (known as B/I frames) are allowed in WMV-9. These frames typically occur at scene changes where it is more economical to code the data as intra rather than P or B. Nevertheless, we distinguish the frame type from true I or key frames, by disallowing them to be referenced by other frames. This allows the decoder (e.g. on a small device) to omit decoding them, as well as allows the encoder to choose a lower quality setting or quantization step size to encode them.
3. Improved motion vector (MV) coding efficiency, due to MV prediction based on "forward predicts forward, backward predicts backward" rule—this involves buffering the forward and backward components of each MV separately, including those corresponding to the direct mode. If we send a forward or a backward MV, then so as not to lose the chain of predictions, we fill in the "hole" in the MV buffer of the opposite type with either the direct mode's MV, or the MV that would be predicted at that position (using the normal rules of MV prediction).
4. Allowing bottom B-fields (in interlace coding) to refer to top fields from the same picture. The bottom field references the top field from the current frame (i.e. top B field) as the "opposite polarity" and the bottom field of the previous picture ("same polarity"), plus top and bottom fields of the next picture. This is shown in Fig. 12. In this way the bottom B field is unique because it references

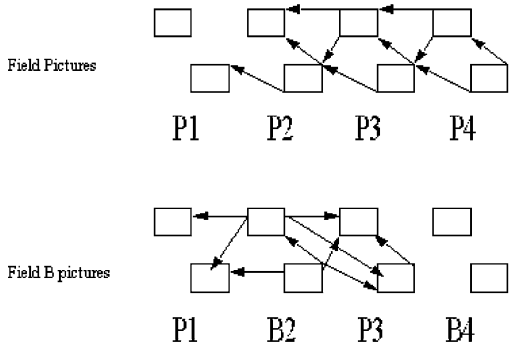

Fig. 12. Coding of B pictures.

part of the same picture—this may be viewed as a slight deviation from the norm of not using any parts of B pictures to predict other pictures although the distinction is academic.

3.9. Overlap smoothing

Overlap smoothing is a technique used to reduce blocking artifacts in intra data—both intra frames as well as intra areas in predicted (P and B) frames. Intra coding in WMV-9 and prior standards such as MPEG-2 and MPEG-4 is carried out by partitioning the image into tiles, performing a linear transform operation on the tiles, quantizing the transform coefficients and entropy coding the non-zero coefficients. At higher levels of quantization, fewer coefficients are quantized to non-zero values. Since the spatial support of the transform basis is restricted to the block (typically 8×8 pixels), the influence of any given non-zero coefficients is circumscribed by this support. This causes apparent edges at the block boundaries.

Using a deblocking filter is one way of reducing blocking artifacts. The downside with the deblocking filter operation is that it is purely a decoder (and reconstruction loop on the encoder) process—there is no accounting or compensation for the fact that a deblocking filter is being applied to the forward process of encoding. As a result, the deblocking filter operates equally on both block-aligned true edges and apparent block edges. Moreover, the deblocking filter is a conditional non-linear operation which may be disabled in the

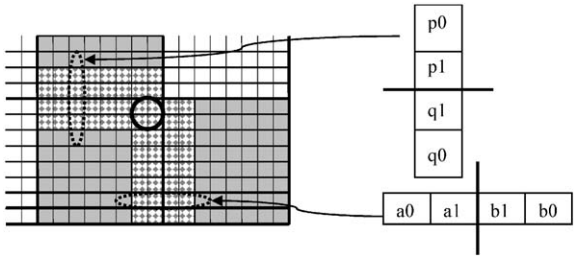

Fig. 13. PSNR vs. Bitrate for WMV-9 Main (without B frames)

less complex profiles. It is desirable to achieve block edge smoothing using a simpler operation.

In order to minimize the blocking effect, cross block correlations can be exploited by means of a lapped transform [20]. A lapped transform is a transform whose input spans, besides the data elements in the current block, a few adjacent elements in neighboring blocks. On the reconstruction side the inverse transform influences all data points in the current block as well as a few data points in neighboring blocks. In two dimensions, the lapped transform is a function of the current block, together with select elements of blocks to the left, top, right, bottom and possibly top-left, top-right, bottom-left and bottom-right. The number of data points in neighboring blocks that are used to compute the current transform is referred to as the overlap.

Fig. 13 shows the pixels to which overlap smoothing is applied. Intra blocks are marked in gray, and the pixels being smoothed are patterned. In the example shown, the overlap is 2 and two pixels each from the two adjacent blocks shown are pre-processed. The encode side operation compensates for the decode side smoothing. The key advantage of the spatial domain realization of the lapped transform is that an existing block-based codec can be retrofitted with a pre- and post-processing stage to derive the benefits of the lapped transform—using the existing codec framework. The post-processing step is a linear smoothing filter applied to the inverse transform reconstruction, within the decode loop. The pre-processing step is the inverse of post-processing.

Certain critical design issues emerge when the spatial domain lapped transform is implemented by means of pre and post-processing stages in an

existing codec. The key issues are range expansion, need for higher precision arithmetic, and reduced quality in high contrast regions. WMV-9 handles these issues by a combination of techniques. First, the lapped transform is used only for a certain quantization and higher, i.e. at the lower bitrates where blocking artifacts are more apparent. Second, the pre- and post-processing operations are not true inverses of each other—the post-processing smoothing operation is heavier than the pre-processing sharpening operation. Third, the range of intermediate data is clamped to 9 bits to ensure that the rest of the transform plumbing does not underflow or overflow. Finally, within an intermediate range of quantization parameters or bitrates, the application of overlap smoothing is signaled at the macroblock level. This allows for overlap smoothing to be switched on or off selectively, for instance switched off in high texture areas and on in smooth regions.

The lapped inverse transform used in WMV-9 is determined by the post-filter matrix which is shown below:

$$P_i = \begin{bmatrix} 7 & 0 & 0 & 1 \\ -1 & 7 & 1 & 1 \\ 1 & 1 & 7 & -1 \\ 1 & 0 & 0 & 7 \end{bmatrix} /8.$$

Two pixels on either side of a block edge, as shown in Fig. 13, are filtered by means of the above 4×4 matrix. The smoothing operation is apparent by noting the filter coefficients, which are the four rows of the matrix. Filter coefficients are small and the post-processing operation is easy to implement. The above matrix has an easily approximated inverse which is used in forward encoding.

3.10. Low-rate tools

The WMV-9 video codec specifies some tools and algorithms that are specifically geared to accommodate low bit rate (LBR) scenarios, e.g. sub-100 kbps. One of these is the ability to code frames at multiple resolutions, by scaling down the X, Y or both X and Y dimensions of each coded frame. The decoder is informed that these frames have been scaled down, and it will up-scale the decoded image in X and Y as appropriate before displaying them. By operating at a down-scaled level, we are able to effectively extend the range of quantization beyond the normal range (usually 1–31) by a factor of $\sqrt{2}$ each time we down-scale any dimension by a factor of 2. This is often sufficient to allow the WMV-9 encoder to operate at low bit rates.

WMV-9 allows for arbitrary resizing factors but the coded frame size is maintained from one I frame to, but not including, the next I frame. If the decoded frame is one of the subsampled resolutions, then it must be upsampled to full resolution prior to display. Since this upsampling process is outside the decoding loop, the implementer is free to use any upsampling filter. This allows hardware vendors to conveniently use pixel shaders or other off-the-shelf components to perform this operation. However, attention should be paid to the relative spatial positioning of the samples produced from the upsampling and downsampling processes.

3.11. Fading compensation

Video sequences with global illumination changes due to effects such as fade-to-black, fade-from-black and dissolves require relatively large amounts of bits to encode because standard motion-compensation techniques are ineffective on such frames. For example, consider a video sequence where a fade-to-black is occurring. During that time period in the video sequence, every luminance value in the current frame may have changed relative to the previous frame, preventing the motion-compensating algorithm from finding a good predictor for any block in that frame. The absence of good predictors implies that the entire frame will be coded as an intra-frame. Thus, fading causes a significant loss in compression efficiency. Natural illumination changes, and artificial transitioning effects such as blending, cross-fades, and morphing also reduce the effectiveness of straightforward motion compensation.

Fading compensation is used by WMV-9 to improve the performance of motion compensation on video sequences that include fading. The

WMV-9 encoder detects fading prior to motion compensation. Fading detection comprises computing an error measure for the current video image relative to the original reference video image, and comparing the error measure with a threshold. If fading is detected, the encoder computes the fading parameters which specify a pixel-wise linear first order transform of the reference image. The fading parameters are quantized and signaled to the decoder. The encoder and decoder use the quantized fading parameters to transform the original reference frame into a new reference frame, which is used for motion compensation. This process allows motion compensation to find better predictors for each block, and thus code more blocks as inter-blocks. Thus fading compensation improves the overall compression efficiency of WMV9 on sequences with fading, or other global illumination changes.

4. Experiments and observations

In this section, we present a consolidated picture of the coding efficiency improvements that WMV-9 achieves over other existing video codecs and formats. First, we look at performance of WMV-9 compared to MPEG-2 and MPEG-4. Next, we compare the compression efficiency of WMV-9 to that of H.264/AVC. We then present some results from the latest independent study by the DVD Forum, which evaluated the quality of WMV-9, and of optimized implementations of H.264/AVC, MPEG-4 ASP, and MPEG-2. Finally, we look at some numbers showing the clear computational advantage of WMV-9 over H.264/AVC.

4.1. WMV-9 compared to MPEG-2 and MPEG-4

Overall, WMV-9 achieves 15 to 50% compression improvements over its predecessor WMV-8, and the improvements tend to be greater at higher bit rates. Fig. 14 shows a peak signal-to-noise ratio (PSNR) quality plot versus bit rate for WMV 9, WMV 8 (for additional reference), and Microsoft's ISO MPEG-4 video codec (simple profile). The source consisted of 13 typical MPEG clips (i.e., "Stefan", "Akiyo", "Mother & Daughter", "Funfare", "Foreman", "Singer", "News", "Sean", "Children", "Mobile & Calendar", "Weather", "Coastguard", and "Hall"). We set a fixed quantization step size for all codecs and used the same mode selection strategy, as it is usually done in MPEG and ITU standard tests. Even though PSNR is by no means an exact measure of video quality, the plot conveys that the visual compression gains also translate to PSNR gains.

Higher gains are achieved by WMV-9 over MPEG-2 [1–3,23]. We estimate that for some sequences and coding bitrates, WMV-9 requires less than 50% of the bits used in MPEG-2. In

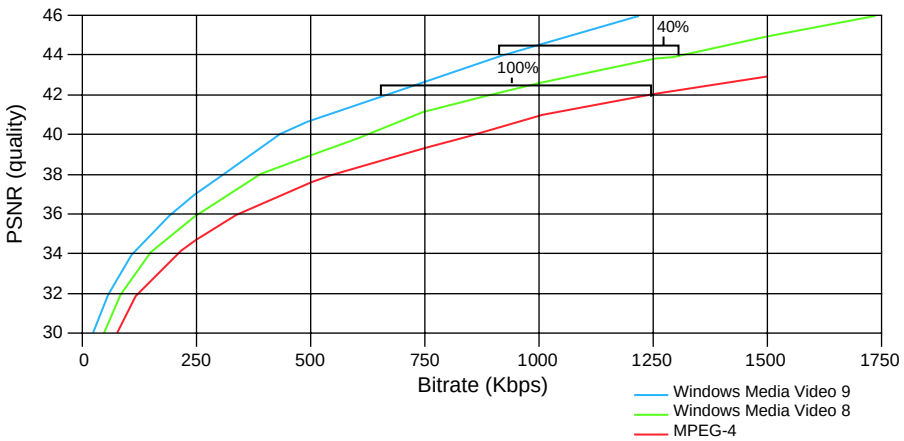

Fig. 14. Rate-distortion plot for WMV 9, WMV 8, and Microsoft's ISO MPEG-4 (based on the simple profile) video codec on MPEG clips. Note that the 100% legend refers to the fact that WMV9 uses 50% fewer bits than MPEG-2 to achieve the same PSNR.

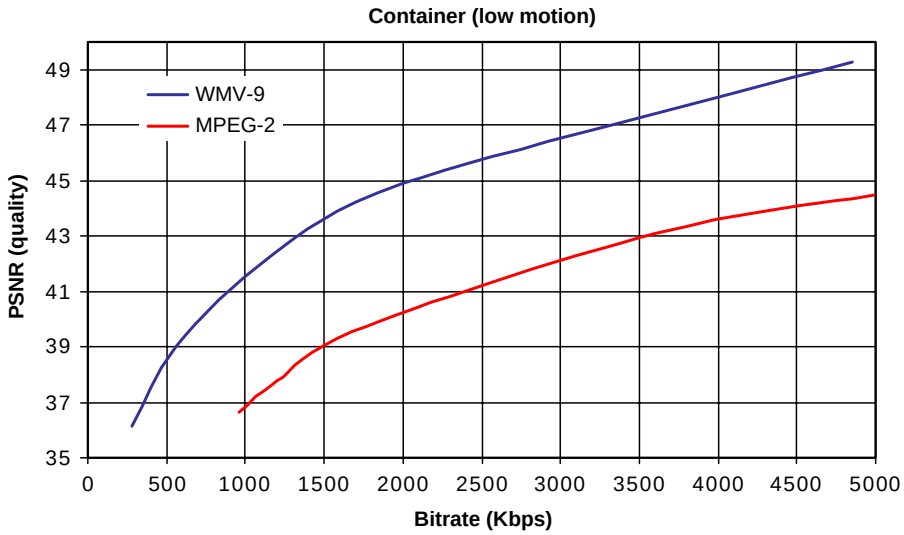

Fig. 15. Rate–distortion plot for WMV-9 vs. MPEG-2, low motion sequence (“Container”).

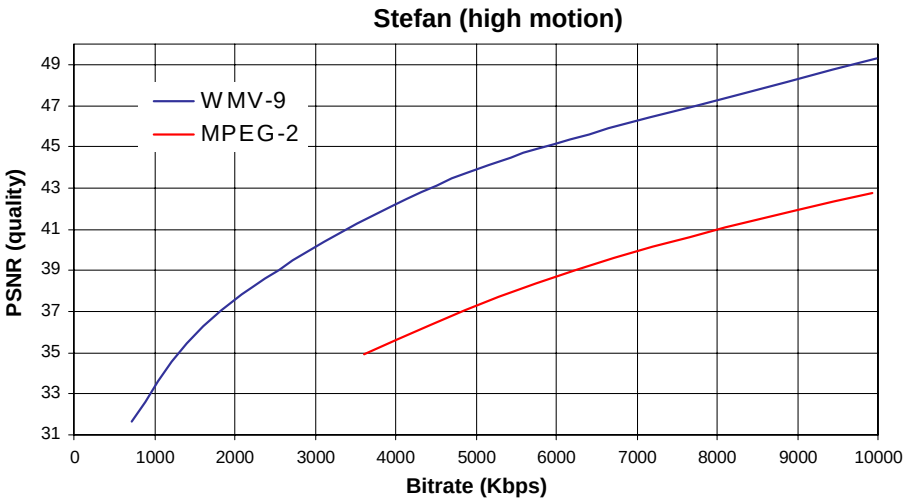

Fig. 16. Rate–distortion plot for WMV-9 vs. MPEG-2, high motion sequence (“Stefan”).

other words, MPEG-2 requires between twice to three times the bitrate of WMV-9 to achieve a given PSNR. As can be expected, the actual performance with WMV-9 in terms of bitrate savings depends on many factors and may be more or less than 50%. Sample results are shown in Figs. 15 and 16—for low motion (container) and high motion (stefan) sequences belonging to the MPEG set, respectively. Minerva’s popular (commercially available) encoder was used to generate the MPEG-2 main profile data points.

Recently, in independent tests performed by DV Magazine [23] the compression quality of WMV-9 was compared (using subjective tests) to competitive MPEG-2 and MPEG-4 advanced simple profile (ASP) codecs (as well as RealVideo 9) for high definition content, and WMV-9 was found to outperform other codecs tested.

4.2. WMV-9 Compared to H.264/AVC

In this section, we compare the video quality of WMV-9 to that of H.264/AVC. First, we present some PSNR results briefly for completeness, once again keeping in mind that PSNR does not measure perceptual video quality accurately (c.f., [19]). Afterwards, we discuss the results of the subjective evaluations performed by independent entities, including the DVD Forum.

Figs. 17 and 18 compare the PSNR performance of WMV-9 and H.264/AVC for the video sequences "Glasglow" and "Stefan", respectively. These PSNR results are part of the contribution in [10] to 3GPP, in which we used the H.264/AVC JM reference encoder [9] with the same configuration of H.264/AVC that was proposed to 3GPP by Nokia. This feature set consisted of H.264/AVC Baseline without use of multiple reference frames, following Nokia's interpretation of the 3GPP application complexity constraints. Accordingly, we used WMV-9 Main profile without B frames. The PSNR of WMV-9 and H.264/AVC would be somewhat higher if their more complex features were used. (However, the configuration of H.264/AVC used in these experiments still has higher complexity than that used for WMV-9, as we will show in Section 4.4.) The results in Figs. 17 and 18 suggest that the PSNR performance of both codecs is data dependent and relatively similar, if one contrasts the differences here to those observed in Figs. 14–16.

When it comes to subjective quality, however, WMV-9 has equaled or outperformed optimized implementations of H.264/AVC. There have been a variety of studies that have independently evaluated the compression efficiency of WMV-9 and H.264/AVC. For example, Tandberg Television evaluated WMV-9 and the H.264/AVC (baseline and main profile) implementations (version 6.0a), and compared them to their optimized MPEG-2 codec and MPEG-4 ASP codecs [1,2]. They concluded that the visual quality achieved by H.264/AVC main profile and WMV-9 was comparable, and that these two codecs provide the best quality among the competing codecs in their comparison.

In another independent study [22] from C'T Magazine, Germany's premiere AV/Computer magazine, the performance of popular codecs such as WMV-9, RealVideo 9, Sorenson 3.1, On2's VP3.2, and codecs based on H.264/AVC and

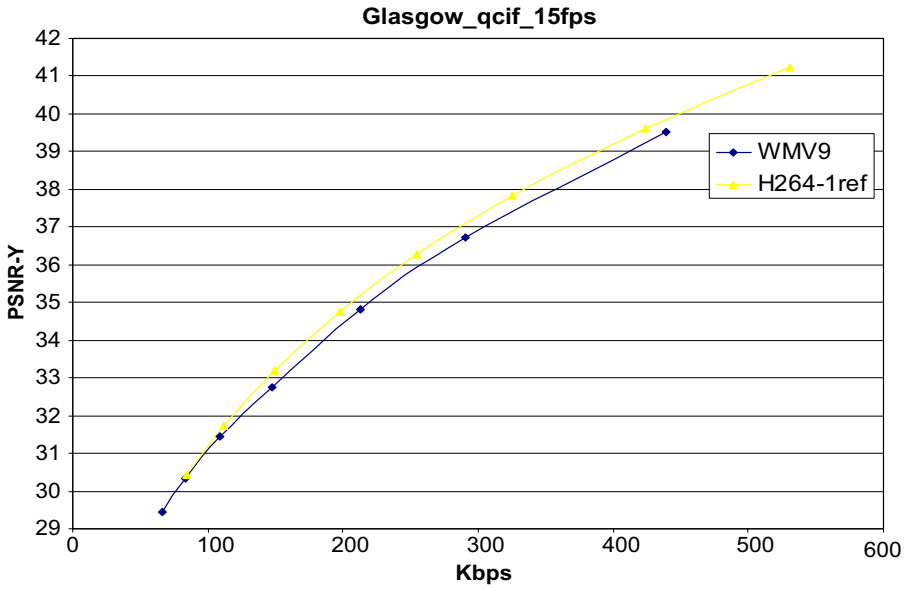

Fig. 17. PSNR vs. Bitrate for WMV-9 Main (without B frames) and H.264/AVC Baseline, in favor of the latter.

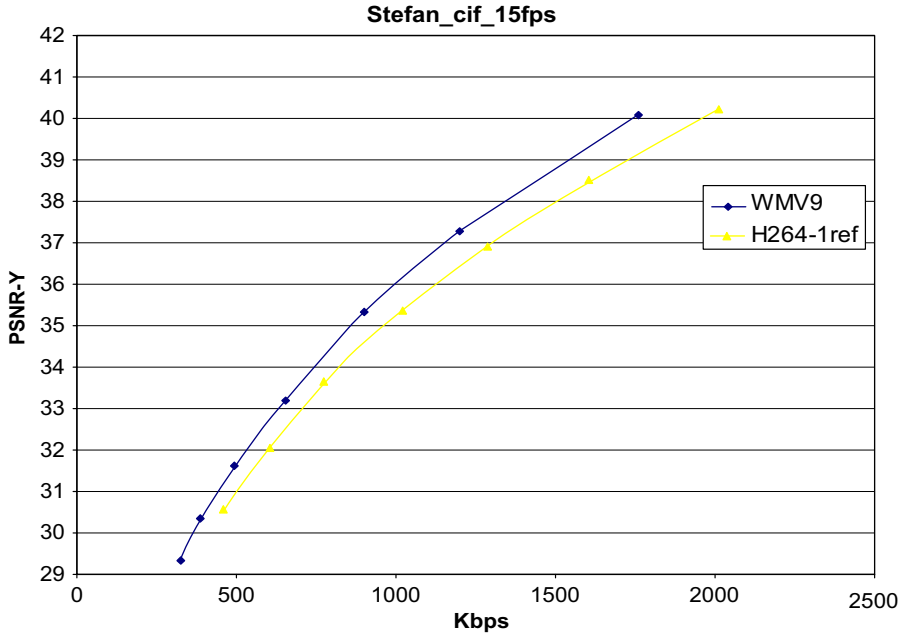

Fig. 18. PSNR vs. Bitrate for WMV-9 Main (without B frames) and H.264/AVC Baseline, in favor of the former.

MPEG-4 (e.g., Dicas, DivX, XVid, etc.) were evaluated subjectively and objectively (using Sarnoff's JNDmetrix, which is also integrated in Tektronix video analysis equipment [19]). In this study, WMV-9 was selected as producing the best subjective and objective quality. In addition, WMV-9 also achieved the highest perceptual quality in each of the clips tested at the latest independent test performed by the DVD Forum, which also evaluated optimized implementations of H.264/AVC main profile, MPEG-4 ASP, and MPEG-2. The DVD Forum results are discussed in some detail in Section 4.3.

The technical description of WMV-9 in section 3 provides sufficient reasons for the conclusions favoring WMV-9, especially for high definition content. For instance, adaptive block transforms used in WMV-9 retain the fine structure, textures and film grain of content more faithfully than H.264/AVC, which tends to smooth out fine details. This effect has been recently demonstrated by other researchers (c.f., [4,26]). As another example, the loop filter of WMV-9 is weaker than that of H.264/AVC and this is a design choice that affects both complexity and subjective performance in favor of WMV-9 likewise for high bitrate video.

4.3. Results from the DVD Forum video codec tests

The latest video codec tests of the DVD Forum provide further evidence of the strong capability of WMV-9 for compressing high definition content. The Forum tested the performance of multiple video codecs (e.g., MPEG-2, MPEG-4 ASP, H.264, WMV-9, etc.) in six film clips of time length 90 s and resolution 1920×1080. These clips were selected for their variety and complexity by major Hollywood Studios. To be more concrete, the sequences 1, 2, 3, 4, 5, and 6, were of segments from the movies "Dick Tracy", "Titan A.E", "Harry Potter and the Sorcerer's stone", "Stuart Little 2", "Seven", and "Monsters, Inc", respectively. Over 35 experts from the Studios and Consumer Electronic companies performed carefully crafted blind tests (e.g., using appropriate lighting conditions, randomizing the clips, displaying the results in a variety of high end monitors, etc.) and compared these codecs to the industry

reference D-VHS (MPEG-2 at 24 Mbps) and the orignal D5 master (near-lossless compression at 235 Mbps). Some key results from the DVD Forum tests in regards of WMV-9 are shown in Figs. 19 and 20. The values in Fig. 19 correspond to the "Overall Impression" perceptual scores for the final round of tests at Panasonic Hollywood Labs. The scoring method used the typical 1–5 scale, where 5 indicates that the given clip is perceived to be equal to the D5 reference. The Forum averaged and rounded the scores to the nearest one decimal point, and the D-VHS and D5 references were included at random in the testing set to eliminate bias, which is why not even the D5 clips achieved an average of 5. The D5 reference was always shown side-by-side with each of the compressed clips. The bitstreams were cross-verified by independent companies. Both pre-processing and post-processing were not allowed in this final round of tests. In Fig. 19, observe that, with only 7.7 Mbps, WMV-9 achieved similar perceptual scores as the references. In fact, WMV-9's score was even higher than that of D-VHS in one sequence, and tied with the scores of the original D5 and D-VHS in another.

Initially, there were nine video codecs participating in the DVD Forum codec tests, which included WMV-9 and several professional (optimized) implementations of MPEG-2, MPEG-4 ASP, and H.264/AVC. The table in Fig. 20 shows the final codec ranking per sequence for WMV-9 and the best H.264/AVC and MPEG-2 implementations. We focus only on these three codecs here, because they were the ones that achieved the highest scores. The codec ranking was obtained by averaging all the perceptual scores for that specific video sequence to the nearest one decimal point, and then ordering the codecs from higher to lower score. The numbers in parentheses indicate the difference in the average perceptual scores from WMV-9. Observe that WMV-9 ranked first for each of the video sequences, and hence received the highest scores and most consistent performance. H.264/AVC's scores were lower in sequences that included fine textures (e.g., Seq 4 is from a segment in "Stuart Little 2" that contains fine artificial textures), while MPEG-2 suffered with fast motion (e.g., Seq 6 is from a fast action scene in "Monsters, Inc").

4.4. Computational complexity

WMV-9 is more complex to decode than MPEG-2 and MPEG-4 simple profile. However, a comparison of computational complexity is only meaningful between codecs that achieve similar levels of compression, which means that the only other codec of interest here is H.264/AVC. A concern with H.264/AVC is the high computational complexity required for encoding and decoding. For example, a preliminary study shows that the decoder complexity of H.264/AVC

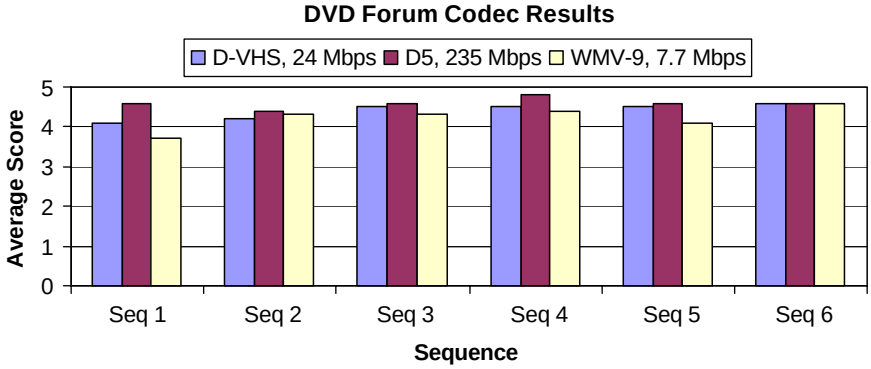

Fig. 19. Average perceptual scores of WMV-9 in comparison to D5 and D-VHS references, as performed by experts in the DVD Forum on six (complex) Film sequences. Observe that, with only 7.7 Mbps, WMV-9's final score was even higher than that for D-VHS (24 Mbps) in one sequence, and tied with D5 (235 Mbps) and D-VHS in another.

Codec Ranking	WMV-9	H.264/AVC	MPEG-2
Seq 1	1	1 (0)	4 (-0.4)
Seq 2	1	2 (-0.3)	3 (-0.4)
Seq 3	1	2 (-0.2)	2 (-0.2)
Seq 4	1	3 (-0.4)	2 (-0.1)
Seq 5	1	3 (-0.1)	1 (0)
Seq 6	1	2 (-0.1)	3 (-0.6)

Fig. 20. Codec ranking for the three codecs that performed best in the DVD Forum final set of tests. The values in parentheses indicate the difference in perceptual scores from WMV-9. WMV-9 ranked first for each of the video sequences, and hence achieved the highest scores and most consistent performance.

(main profile) is about three times higher than MPEG-4 simple profile [16].

On the other hand, the decoding complexity of WMV 9 (main profile) is relatively close to that of the MPEG-4 simple profile codec. To be more specific, decoding with WMV 9 is about 1.4 times slower, which can be verified easily by using both codecs in the Windows Media Player (or other MPEG-4 simple profile decoders). Even though one cannot derive strong conclusions or draw parallels on such complexity analyses, this information suggests that H.264/AVC decoding complexity is likely to be twice that of the WMV-9 codec, or at least that there is a significant computational benefit of WMV-9 over H.264/AVC on the decoder side.

Finally, our superficial computational complexity analysis is backed up with some experimental results. We compare the decode performance of WMV-9 and H.264/AVC on a non-PC processor, the ARM (clock cycles measured through the Armulator). Fig. 21 shows a table comparing the clock cycles required for WMV-9 and H.264/AVC decode on 10 s long clips encoded with the main profile of WMV-9 and baseline profile of H.264/AVC. The numbers for H.264/AVC decode are as optimized and reported by Nokia [11]. It is clear that there is a significant computational advantage with WMV-9, i.e., WMV-9 Main profile decoding requires 2–3 times less computation than H.264/AVC Baseline. Observe that H.264/AVC Main decoding is quite more complex than baseline (e.g., Main includes arithmetic coding).

Sequence	Millions of ARM cycles/second	
	WMV9	H.264/AVC
Foreman	27	70.0
News	17	45.9
Container	19	45.5
Silent	18	50.8
Glasgow	25	48.5
Average	**21.2**	**52.14**

Fig. 21. Comparison of WMV-9 main and H.264/AVC baseline decoder complexity in an ARM chip. Observe that WMV-9 decoding is 2–3 times less complex than H.264/AVC baseline.

4.5. Inferences

Although we draw inferences in the comparison of WMV-9 with MPEG-2, MPEG-4 and H.264/AVC in the above paragraphs, it must be pointed out that standards only define the bit stream syntax and decoder semantics. Hence, different encoder implementations of the same standard can produce different quality results, all of which are compatible with the standard. This holds for WMV-9 as well.

Although today's independent data suggests that the performance of the currently available WMV-9 is as good as or better than that of any other codec, there are plenty of opinions on which video codec produces the best video quality. As a result, it is often recommended that experts do their own tests and reach their own conclusions. The easy availability of the WMV-9 codec, including encoder and decoder, allows interested readers to perform their own experiments on their choice of data and codec parameters. Such readers are warned that some of the options available in the commercially available version of WMV-9 (and quite possibly with other encoders) are designed for visual quality improvement and as such may be detrimental to PSNR. Rate control is another factor that can mask more subtle

differences between codecs, even different implementations of the same underlying format.

It can be appreciated that WMV-9 contains several innovations and design breakthroughs. For a thorough analysis, each innovation must be evaluated in a carefully controlled test setup. However, this level of detail is outside the scope of this paper. Moreover, it is widely known that the performance of video codecs is highly data dependent.

With the compression efficiency of WMV-9, one can achieve broadcast-quality BT.601 video at about 1–2 Mbps, and high-quality, high-definition video (e.g., 1280×720, 1920×1080) at high-end broadcast or DVD rates (e.g., around 4–8 Mbps). All the broadcast formats are supported, including the high-definition 720p and 1080i variants. This makes WMV-9 a significant improvement over today's commercial MPEG-2 broadcast technology.

5. Standardization of WMV-9

Given the large acceptance of and interest in WMV-9 by the industry, Microsoft has decided in September 2003 to propose the decoder syntax and semantics for standardization under the aegis of SMPTE (Society of Motion Picture and Television Engineers). SMPTE accepted the work item, and the group C-24 has been working on the standardization of the WMV-9 codec for several months, and recently promoted the spec to Committee Draft (CD) stage. The standardization process is expected to take about a year, and it will result in a standard known by the brand-agnostic label VC-9 (for Video Codec 9). This term is expected to be used by other companies that independently implement the codec based on the standards specification, while WMV-9 will be the Microsoft-based implementation of VC-9.

6. Summary

The vision of Windows Media 9 Series is to deliver compressed digital media content to any device over any network. Solutions and services are provided by the ecosystem of partners that either use the basic components in Windows Media 9 Series (Windows Media Encoder for authoring, Windows Media Services for distribution, Windows Media DRM technology for usage rights assignment, and Windows Media Player for playback), or build their own hardware or software components using the Windows Media hardware porting kits or SDKs.

Windows Media 9 Series provides a variety of state-of-the-art audio and video codecs for different applications. The Windows Media Video 9 (WMV-9) codec is a block-based, hybrid codec that incorporates advanced compression technology in each of its components. Experimental results and analysis show that the WMV-9 quality is competitive with H.264/AVC, and arguably superior based on current independent tests, with significantly lower computational complexity. We explain why some of the tools unique to WMV-9 provide an intrinsic quality benefit over H.264/AVC. For example, WMV-9's adaptive block transform is better suited for higher resolution content than the fixed 4×4 transform of H.264/AVC. At the time of writing this paper, the H.264/AVC committee is aware of this weakness and is planning to add an additional transform to future extensions (e.g., for 4:2:2 or 4:4:4 video) of H.264/AVC.

Acknowledgements

The authors would like to thank their colleagues in the Microsoft Windows Digital Media Division for their contributions and feedback to this paper. We are also grateful to the anonymous reviewers for their insightful comments and suggestions.

References

[1] J. Bennet, A. Bock, Comparison of MPEG and Windows Media Video and audio encoding, White Paper, Tandberg Television, September. 10, 2002.

[2] J. Bennett, A. Bock, In-depth review of advanced coding technologies for low bit rate broadcast applications, TANDBERG Television, UK (presented at IBC' 03).

[3] T. deBie, Show report—video software dealers association, Home Theater Hi-Fi, July 2003.

[4] S. Gordon, Adaptive block transform for film grain reproduction in high definition sequences, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, doc. JVT-H029, Geneva, Switzerland, 23–27 May, 2003 (available via anonymous ftp from ftp://fttp.imtc-files.org/jvt-experts/).
[5] Y. Huh, K. Panusopone, K.R. Rao, Variable block size coding of images with hybrid quantization, IEEE Trans. Circuits Systems Video Technol. 6 (December 1996) 679–685.
[6] International Standards Organization, ISO/IEC 13818-7, MPEG-2 Standards Document.
[7] International Standards Organization, ISO/IEC 14496-3, MPEG-4 Standards Document.
[8] International Standards Organization, ISO/IEC 14496-10, H.264/AVC/MPEG-4 Part 10 Standards Document.
[9] Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, Joint Model number 6.1, Pattaya II, Thailand, 7–14 March, 2003.
[10] Microsoft, WMV-9—an Advanced Video Codec for 3GPP, 3GPP SA4 Meeting #18, document S4-030613. (available from http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_28/Docs/)
[11] Nokia, Proposal to support MPEG-4 AVC / H.264/AVC in Rel-6, 3GPP SA4 Meeting #27, document S4-030478. (available from http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_27/Docs/)
[12] Official Web site of the SD association, http://www.sdcard.org/.
[13] Official Windows Media Web site, http://www.microsoft.com/windows/windowsmedia/default.asp.
[14] W.B. Pennebaker, J.L. Mitchell, JPEG Still Image Data Compression Standard, van Nostrand Reinhold, New York, 1993.
[15] N. Ranganthan, S.G. Romaniuk, K.R. Namuduri, A lossless image compression algorithm using variable size block segmentation, IEEE Trans. Image Proccess. 4 (10) (1995) 1396–1407.
[16] M. Ravassi, M. Mattavelli, C. Clerc, JVT/H.26L decoder complexity analysis, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, doc. JVT-D153, Klagenfurt, Austria, 22–26 July, 2002 (available via anonymous ftp from ftp://ftp.imtc-files.org/jvt-experts/).
[17] J. Ribas-Corbera, D.L. Neuhoff, Optimizing block size in motion compensation, J. Electron. Imaging 1 (January 1998) 155–165.
[18] J. Ribas-Corbera, D.L. Neuhoff, Optimizing motion vector accuracy in block-based video coding'', IEEE Trans. Circuits Systems Video Technol. 11 (4) (April 2001) 497–511.
[19] Tektronics Picture Quality Analyzer PQA 300 (http://www.tek.com/site/ps/0,,25-11735-INTRO_EN,00.html).
[20] T.D. Tran, J. Liang, C. Tu, Lapped transform via time-domain pre- and post-filtering, IEEE Trans Signal Process. 51 (6) (June 2003) 1557–1571.
[21] Video on the move, the IBC Daily, Saturday, 14 September 2002, p. 1 (electronic version under IBC Daily News in http://www.ibc.org).
[22] Z. Volta, Kompressionisten, C'T Mag. 10 (2003) 146-159, (in German, summary at http://www.heise.de/ct/03/10/146/).
[23] B. Waggoner, HD delivery codecs, DV Mag. (2003) 34–40.
[24] Web site for HighMAT CD and DVD format, http://www.microsoft.com/windows/windowsmedia/Consumelectronics/highmat.asp.
[25] M. Wien, Variable block size transforms for H.264/AVC, IEEE Trans. Circuits Systems Video Technol. 13 (7) (July 2003) 604–613.
[26] Windows Media partners, http://www.microsoft.com/windows/windowsmedia/partner.asp.
[27] Windows Media Web site for Consumer Electronic devices, http://www.microsoft.com/windows/windowsmedia/conselec.asp.