PROCEEDINGS OF SPIE

Visual Communications and Image Processing 2005

Shipeng Li
Fernando Pereira
Heung-Yeung Shum
Andrew G. Tescher
Editors

12–15 July 2005
Beijing, China

Co-hosted by
Microsoft Research Asia
Tsinghua University
National Natural Science Foundation of China
Institute of Computing Technology, Chinese Academy of Sciences

Technical Co-sponsorship by
SPIE—The International Society for Optical Engineering
IEEE Circuits and Systems Society

Sponsored by
Microsoft Research Asia
National Natural Science Foundation of China
DoCoMo Communications Laboratories USA, Inc.
Ricoh Company Ltd.

Published by
SPIE—The International Society for Optical Engineering

Volume 5960
Part Two of Four Parts

The International Society for Optical Engineering

SPIE is an international technical society dedicated to advancing engineering and scientific applications of optical, photonic, imaging, electronic, and optoelectronic technologies.

The papers included in this volume were part of the technical conference cited on the cover and title page. Papers were selected and subject to review by the editors and conference program committee. Some conference presentations may not be available for publication. The papers published in these proceedings reflect the work and thoughts of the authors and are published herein as submitted. The publisher is not responsible for the validity of the information or for any outcomes resulting from reliance thereon.

Please use the following format to cite material from this book:

Author(s), "Title of Paper," in *Visual Communications and Image Processing 2005*, edited by Shipeng Li, Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proceedings of SPIE Vol. 5960 (SPIE, Bellingham, WA, 2005) page numbers.

ISSN 0277-786X
ISBN 0-8194-5976-3

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 1 360/676-3290 (Pacific Time) · Fax 1 360/647-1445
http://www.spie.org

Printed in the United States of America.

Contents

Part One

WLB-2: VIDEO CONTENT ANALYSIS

WLC-2: NOISE REDUCTION AND ARTIFACT REMOVAL

WLA-3: ADVANCES IN SCALABLE VIDEO CODING (SPECIAL SESSION)

Part Two

TLC-1: VIDEO CODING ALGORITHMS II

TLA-2: CODING STANDARDS AND APPLICATIONS (SPECIAL SESSION)

TLB-2: MEDIA PROCESSING FOR DIGITAL ENTERTAINMENT (SPECIAL SESSION)

TLB-3: TARGET DETECTION AND IDENTIFICATION

TLC-3: MEDIA DRM AND SECURITY

An Overview of VC-1

Sridhar Srinivasan, Shankar L. Regunathan

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

ABSTRACT

VC-1 is a state-of-the-art video compression format that provides highly competitive video quality, from very low through very high bit rates, at reasonable computational complexity. The technology includes several innovations to improve the rate-distortion performance envelope of the format and to provide additional tools for perceptual quality improvement. This paper describes some of the key innovations of VC-1 that contributes to its quality and complexity advantages. The focus of this paper is on those tools which are unique to VC-1, and differentiate it from other video compression standards.

Keywords: Video coding, standards, VC-1, compression.

1. INTRODUCTION

VC-1 is a state-of-the-art video compression format that provides highly competitive video quality, from very low through very high bit rates, at reasonable computational complexity. VC-1 is under standardization by the Society of Motion Picture and Television Engineers (SMPTE), and is currently at the threshold of the Final Committee Draft (FCD) ballot.

This paper provides an introduction to the compression algorithms used in VC-1. We discuss in detail the key innovations of the codec that are critical for its good performance. In addition, VC-1 uses many techniques and approaches for entropy coding, interlace specific tools, low-bit rate tools which are not covered in this paper.

VC-1 uses a block based motion compensation and spatial transformation scheme which, at a high level, is similar to all popular video compression standards starting with MPEG-1 through MPEG-4/AVC. At a high level, all of these standards perform block-by-block motion compensation from the previous reconstructed frame. A two dimensional quantity called the motion vector (MV) is used to signal spatial displacement. The 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. 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. Needless to say, the above description only provides a high level overview and does not discuss implementation details.

VC-1 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 VC-1 are not used as a reference for subsequent frames. Thus, B frames are outside of the decoding loop, and the decoding of B frames can be omitted without drift or long-term visual artifacts. The above definition of I, P and B frames holds for both progressive and interlaced sequences.

The important innovations that distinguish VC-1 from prior standards are:

a) adaptive block size transform,

b) 16-bit implementation of the transform,

Visual Communications and Image Processing 2005, edited by Shipeng Li, Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proc. of SPIE Vol. 5960 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.631574

c) multiple precision modes for motion compensation,

d) uniform and non-uniform quantization,

e) loop-filtering, and

f) overlap smoothing.

These techniques are discussed in the subsequent sections.

2. ADAPTIVE BLOCK SIZE TRANSFORM

Traditionally, 8x8 transforms have been used for image and video coding[6-8]. The 8x8 size has the advantage of being dyadic, and large enough to capture and preserve trends and periodic structures. However, it is known that smaller transforms are better in areas with discontinuities because they produce fewer ringing artifacts[1,2].

VC-1 takes the approach of allowing 8x8 blocks to be encoded using either 8x8, 8x4, 4x8 or 4x4 transforms as shown in Figure 1. This allows the VC-1 codec to choose 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.

The transform type can be signaled at the frame, or at the macroblock or block level to optimize overhead for the bitrate and content type. 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 8x8 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 8x8 transforms. The use of adaptive block transform achieves significant rate-distortion benefits. More importantly, from the point of subtle texture preservation (such as keeping film details and grain noise), adaptive transform also provide significant subjective quality benefits[3,9].

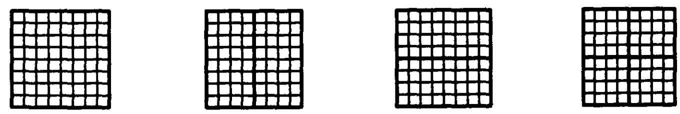

Figure 1. Transform sizes allowed in VC-1

3. 16 BIT PRECISION TRANSFORM

The important motivation for the transform design is to minimize the computational complexity of the decoder while preserving the compression efficiency. The VC-1 inverse transforms are designed to be implemented in 16-bit fixed point arithmetic which facilitates software as well as hardware realizations. It is well known that 32 bit operations are expensive to implement on 16 bit processors which form a large percentage of commonly used DSPs. Further, when 32 bit arithmetic or SIMD operations such as those defined by MMX® is available, multiple 16-bit operations of the transform can be computed in parallel. Thus, the computational complexity of the decoder is significantly reduced, if the inverse transform can be implemented exactly in 16-bit fixed point arithmetic, when compared to an inverse transform requiring 32-bit integer or floating point arithmetic. At the same time, the transform approximates a DCT, and thus retains the favorable energy compaction properties of the DCT on intra and residual video data.

The VC-1 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:

a) Transform coefficients are small integers.

b) The transform is a 16 bit operation – where both sums and products of two 16 bit values produce results within 16 bits.

c) Forward and inverse transforms form an orthogonal pair.

d) The transform approximates a DCT.

e) 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.

f) 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 the use of same quantization parameter between various transform types while maximizing the rate-distortion performance.

It is impossible to find am 8x8 transform that meets all of these constraints simultaneously. The key innovation in VC-1 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.

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.

4. MOTION COMPENSATION

Efficiency of a video codec is closely related to the ability of the motion compensator to generate a good set of predictors. The quality of motion compensation is determined by three factors:

1) sub-pixel resolution,

2) size of predicted area and

3) filters used for interpolation.

While sub-pixel resolution improves the quality of prediction, the benefits of going to finer pixel resolutions are offset by the increased cost of coding motion vectors to higher degrees of precision[4]. At low rates, higher precision in motion vectors is a liability since the percentage of bits used in coding motion vectors is significant. Motion vector resolution is either ½ or ¼ pixel in existing profiles in standard video codecs. VC-1 allows a maximum resolution of ¼ pixels. However, VC-1 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 16x16 pixel area in the luminance plane. MPEG-4 allows definition of motion vectors for 16x16 or 8x8 blocks; this choice is made for each macroblock being coded. H.264/AVC permits motion vectors to reference areas as small as 4x4. While this level of freedom can be useful at high bit rates, smaller areas impose higher computational overhead on the codec. Smaller blocks with randomly distributed motion vectors cause increased cache access, and need more filtering steps on a per-pixel basis, and thus increase the computational complexity of the decoder. VC-1 uses 16x16 motion vectors by default, but allows for 8x8 motion vectors in 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. VC-1 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, VC-1 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 (16x16 and 8x8), ¼ pixel, bicubic
2) 16x16, ¼ pixel, bicubic
3) 16x16, ½ pixel, bicubic
4) 16x16, ½ pixel, bilinear

This 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 lower bit rates use modes towards the bottom of the list. Only when mode 1 is chosen, the predicted block size is allowed to be either 16x16 or 8x8, and the size is signaled on a macroblock basis. In all other modes, only 16x16 block sizes are used. The consolidation of these three criteria into one leads to a more compact decoder implementation, with no significant performance loss.

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).

VC-1 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 VC-1 for ¼ and ½ pixel shifts are: [-4 53 18 -3]/64 and [-1 9 9 -1]/16.

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. VC-1 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 ¼ 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 ½ 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.

5. QUANTIZATION:

Quantization and dequantization of transform coefficients are key steps that can critically affect the rate-distortion performance of a video codec. Earlier standards such as MPEG-2 use quantization with a dead-zone, where all quantization intervals except the dead-zone are of the same size—the dead-zone being typically larger. The quantization intervals and reconstruction levels of a dead-zone quantizer are shown in Figure 2(a). The use of a dead-zone leads to substantial bit savings at low bit rates.

In regular uniform dequantization, the reconstruction levels are all equally spaced. The quantization intervals and reconstruction levels of a regular uniform quantizer are shown in Figure 2(a). This kind of a quantizer performs better at higher bit rates.

VC-1 allows for the use of both dead-zone and regular uniform quantization. The type of quantizer is signaled at the frame level, and the appropriate dequantization rule is applied to all coefficients within the frame by the decoder. The encoder typically uses dead-zone quantizers at larger step sizes, and uses uniform quantizer at lower step sizes. 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 the choice of quantizer type. The flexibility to switch between dead-zone and regular uniform quantization is a key factor in the superior performance of VC-1 at both low and high bit rates.

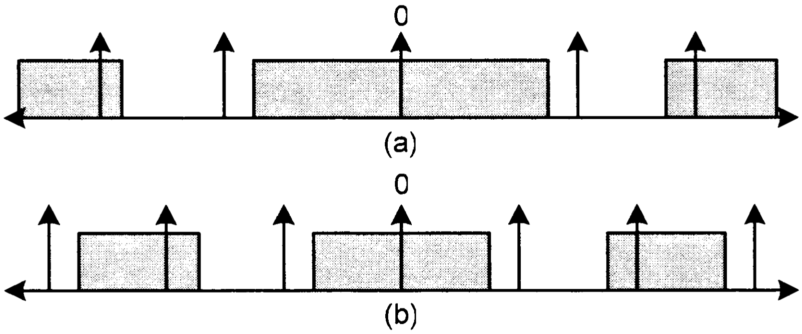

Figure 2. VC-1 quantization and dequantization rules showing (a) dead-zone and (b) regular uniform quantization – arrows are reconstruction levels, and gray boxes are recommended quantization bins (for alternate intervals)

6. LOOP-FILTERING

Compression can induce discontinuities at block boundaries, and these discontinuities show up as visible 'blocky' artifacts. Moreover, these discontinues adversely affect the quality of the reconstructed frame as a predictor for future frames. To mitigate these effects, the VC-1 scheme uses an in-loop deblocking filter. 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.

Figure 3 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 a discontinuity measurement involving pixels P1 through P8 which seeks to detect whether the discontinuity is above a certain threshold. This measurement is 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.

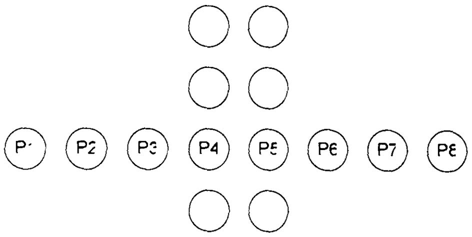

Figure 3. Pixels used in loop filtering operation- only pixels P4 and P5 may be modified

In P pictures, the boundary between a block and a neighboring block is not filtered if both blocks have the same motion vector and if both blocks have no residual error. This prevents over-smoothing of block boundaries where quantization or motion compensation induced discontinuities are unlikely to occur. This constraint also lowers the complexity of the loop-filtering process in P pictures.

Due to the various condition checks involved, loop filtering is a computationally expensive process. VC-1 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 Figure 3, 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.

7. OVERLAP SMOOTHING

While loop-filtering can mitigate artifacts due to boundary discontinuities, it cannot distinguish between both block-aligned true edges and apparent (quantization induced) block edges. This drawback is the result of loop-filtering being purely a decoder (and reconstruction loop on the encoder) process as there is no accounting for the fact that a deblocking filter is being applied to the forward process of encoding. Moreover, the loop-filter is a conditional non-linear operation, and hence may be disabled in the less complex profiles due to its computational requirements.

A lapped transform is a simpler technique to minimize block boundary artifacts in intra-coded macroblocks, while avoiding the drawbacks of loop-filtering[5]. The use of lapped transform is motivated by the following observation: Intra coding in video codecs is performed by partitioning the picture into tiles, performing a linear transform, and quantizing the transform 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 8x8 pixels), the influence of any given non-zero coefficients is circumscribed by this support. This causes apparent edge artifacts at the block boundaries.

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.

Figure 4 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.

A spatial domain realization of the lapped transform is used in VC-1. 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. 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. VC-1 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 prevent underflow or overflow.

Further, VC-1 allows for the signaling of the overlap smoothing at the macroblock level for an intermediate range of bit rates. Thus, overlap can be enabled in high texture areas and disabled in smooth regions providing additional perceptual benefits.

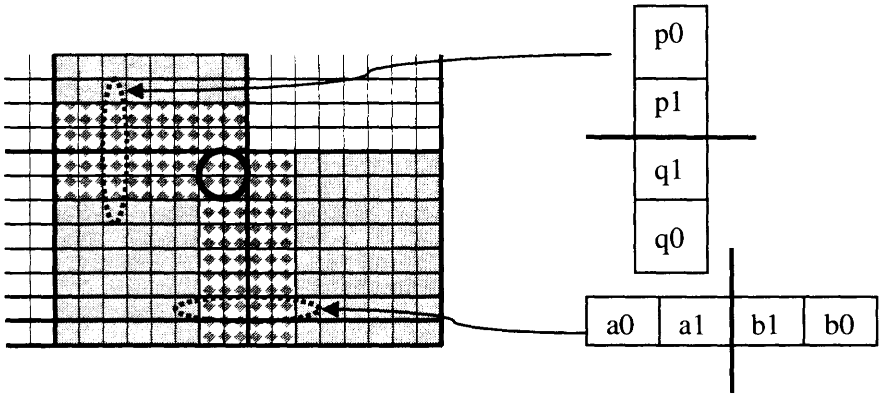

Figure 4. Overlap smoothing applied to block edges

8. OTHER TOOLS

VC-1 specifies some tools and algorithms that are specifically geared to accommodate low bit rate (LBR) scenarios, e.g. sub-100 kbps for CIF size video. 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.

VC-1 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.

Fading compensation is used by VC-1 to improve the performance of motion compensation on video sequences that include fading effects such as fade-to-black, fade-from-black and cross-fades. This tool helps in the generation of better predictors since luminance change cannot be modeled by 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 VC-1 on sequences with fading, or other global illumination changes.

9. RESULTS

It is well understood that the newer video bit-stream formats such as VC-1 and H.264 allow for superior compression performance (in terms of rate-distortion plots) compared with the older formats such as MPEG-2 and MPEG-4 [10]. The more interesting question that remains to be answered is the relative performance of VC-1 vs. H.264. In general, video formats are difficult to compare because of several reasons. Primarily, formats define syntax, and not a specific implementation. Any experimental comparison therefore compares individual encoder implementations rather than the format itself. Secondly, the most meaningful comparison is a subjective one. However, subjective tests are difficult to conduct in a statistically meaningful manner. Finally, objective comparisons such as PSNR etc collapse the time-varying encoder performance into a single data point. This glosses over aspects of rate control (which may be very different between encoder implementations), error patterns and temporal artifacts.

Having provided the above disclaimer, we present a rate-distortion result comparing VC-1 with H.264. In this test, we used the high profile of H.264, with CABAC. The H.264/AVC JM 9.1 code was used in this experiment. The VC-1 comparison used the advanced profile. Both codecs did not use B Frames. The input was a 720x480 film clip with 300 progressive frames, encoded with no periodic I frames. Rate control effects were sidestepped by encoding with fixed quantization parameters for both formats. The rate distortion plot corresponding to this test is shown in Figure 5. H.264 is slightly superior to VC-1 at the higher rates, and loses this advantage as the rate falls. It must also be borne in mind that the complexity of H.264 is noticeably higher than that of VC-1[10].

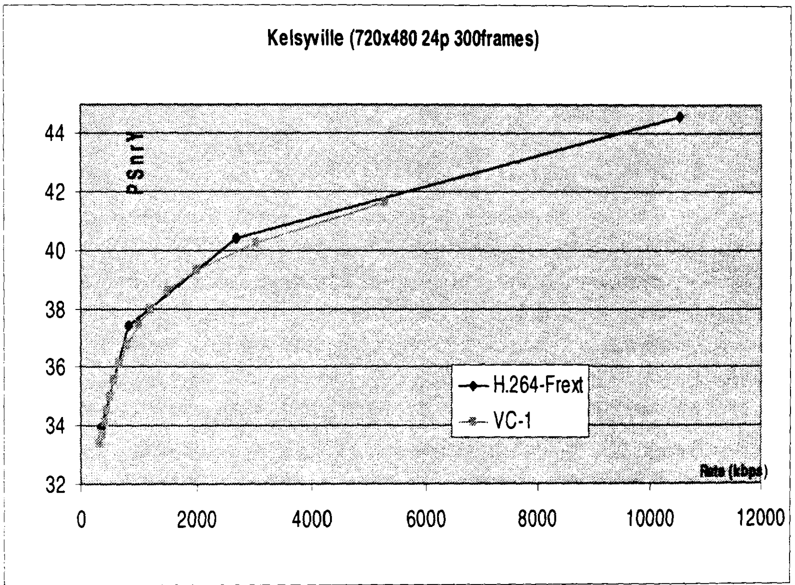

Figure 5. Rate-distortion plot comparing VC-1 with H.264

10. SUMMARY

This paper described some of the key innovations in VC-1. The focus was on describing some of the tools unique to VC-1, and the quality and performance benefits of these tools.

ACKNOWLEDGMENTS

The authors wish to acknowledge the work and assistance of Tom Holcomb, Pohsiang Hsu, Bruce Lin and Kunal Mukerjee in the preparation of this paper.

REFERENCES

1. 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.
2. J. Ribas-Corbera and D.L. Neuhoff, "Optimizing Block Size in Motion Compensation", Journal of Electronic Imaging, No.1, January 1998, pp.155-165.
3. M. Wien, Variable block size transforms for H.264/AVC, IEEE Trans. Circuits Systems Video Technol. 13 (7) (July 2003) 604–613.
4. J. Ribas-Corbera and D.L.Neuhoff, "Optimizing Motion Vector Accuracy in Block-Based Video Coding", IEEE Trans. On Circuits and Systems for Video Technology, pp. 497-511, Vol. 11, No. 4, April 2001.
5. T.D.Tran, J.Liang and C. Tu, "Lapped Transform via Time-Domain Pre- and Post-Filtering", IEEE Trans on Signal Processing, vol.51, no.6, pp. 1557-1571, June 2003.
6. W.B.Pennebaker and J.L.Mitchell, JPEG Still Image Data Compression Standard, van Nostrand Reinhold, 1993.
7. International Standards Organization, ISO/IEC 13818-7, MPEG-2 Standards Document.
8. International Standards Organization, ISO/IEC 14496-3, MPEG-4 Standards Document.
9. 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://ftp.imtc-files.org/jvt-experts/)
10. S. Srinivasan et al, "Windows Media Video 9: overview and applications", Signal Processing: Image Communication, vol. 19, no. 9, October 2004, pp. 851-875.