A Performance Analysis of the ITU-T Draft H.26L Video Coding Standard

Anthony Joch, Faouzi Kossentini and Panos Nasiopoulos

Department of Electrical and Computer Engineering
University of British Columbia, Canada
e-mail: {anthonyj, faouzi, panos}@ece.ubc.ca

Abstract

The ITU-T draft H.26L video coding standard has been designed with the goal of enabling significantly improved compression performance relative to all existing video coding standards, as well as other important features, such as error resilience and network adaptability. While the basic coding framework of the standard is similar to that of currently popular video standards, H.26L includes many coding features not present in earlier standards. In this paper, we first illustrate the improved coding efficiency enabled by H.26L relative to existing video coding standards. Then, we provide a detailed analysis of the individual coding gains provided by several key features of the H.26L standard. These results can be used to motivate further improvements to the standard, assist in choosing the features that will comprise each of its profiles, and facilitate the development efficient encoding algorithms that find an optimal balance between coding performance and complexity.

1 Introduction

The development and acceptance of international standards for video coding, such as MPEG-2 and H.263, have been the driving force behind a diverse range of multimedia applications, including DVD systems and videoconferencing. The draft H.26L standard [1] is currently being developed by the ITU-T Video Coding Experts' Group (VCEG)[1] for coding of natural video content over a broad range of bit rates. The technical content of the standard is still under development, and this paper refers to the Test Model Long Term Number 8 (TML-8)[2].

The significantly improved coding performance offered by this draft standard relative to the most complex profiles of the H.263 and MPEG-4 standards has been demonstrated for high-latency applications [2, 3]. While H.26L uses the same hybrid block-based motion compensation and transform-coding model as these existing standards, improvements are achieved through the inclusion of a number of new features and capabilities. The goal of this paper is to systematically analyze the incremental coding gains that are provided by several of the key features of H.26L over a wide variety of video content. For more detailed results, see [4]. However, we first provide a brief introduction to the coding algorithms of the H.26L standard and illustrate the improved coding performance that this draft standard can provide relative to H.263 and MPEG-4 in conversational applications, where minimal latency is a key requirement.

---

[1] VCEG and ISO/IEC MPEG have recently formed a Joint Video Team (JVT) that will take over the H.26L project to create a solution for a next generation of standard video coding.

[2] Although technical work on H.26L is mature and nearing completion, we emphasize that the coding performance results presented in this work reflect only the current draft of the standard and are subject to change as the draft standard continues to evolve.

2 Overview of the Draft H.26L Standard

An important concept in the design of H.26L is the separation of the standard into two distinct layers: a video coding layer (VCL), which is responsible for generating an efficient representation of the video data; and a network adaptation layer (NAL) [5] which is responsible for packaging the coded data in an appropriate manner based on the characteristics of the network upon which the data will be used. The paper addresses the performance of the VCL.

The underlying coding scheme defined by H.26L is superficially similar to that successfully employed in prior video coding standards, such as H.263 and MPEG-2. This includes the use of translational block-based motion compensation, DCT-based residual coding, scalar quantization with an adjustable step size for bit rate control, zigzag scanning and run-length VLC coding of quantized transform coefficients. However, specific details within this structure and some key additional features differentiate H.26L from all other standards.

The motion compensation model used in H.26L is more flexible and efficient than those found in earlier standards. Support for the use of multiple previous reference pictures for prediction is included in the core of the standard (it was previously available only in the newest high-capability version of H.263). A much larger number of different motion compensation block sizes are available for performing motion compensation on each 16x16 macroblock (H.263 supported two such block sizes, while H.26L supports seven, as illustrated in Figure 1). Motion vectors can be specified with higher spatial accuracy than in earlier standards, with quarter-pixel accuracy as the default lower-complexity method and eighth-pixel accuracy available as a higher-complexity option (a form of quarter-pixel motion is found in prior standards only in the higher-capability newest versions of MPEG-4). In addition to intra-coded (I-) and predicted (P-) pictures, bidirectionally predicted (B-) pictures and a new type of inter-stream transitional picture known as SP-pictures are also supported in H.26L. The use of a deblocking filter within the motion compensation loop is specified in order to reduce visual artefacts and improve prediction (in-loop deblocking only appeared before in a high-capability optional mode of H.263).

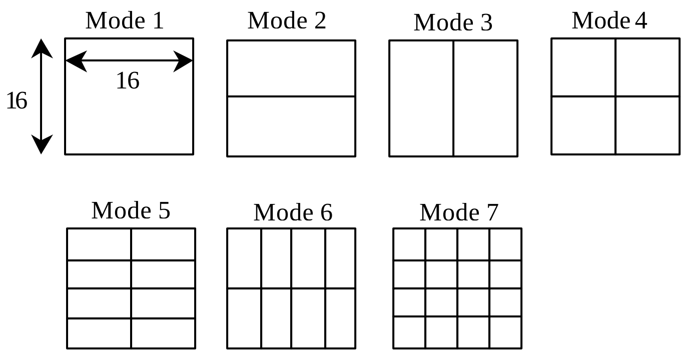

Figure 1: The seven possible modes for motion compensation of each 16x16 macroblock. H.26L supports motion compensation blocks as small as 4x4 pixels.

H.26L is unique in that it employs a purely integer spatial transform (an approximation of the DCT) which is primarily 4x4 in shape, as opposed to the usual floating-point 8x8 DCT specified with rounding-error tolerances that is used in earlier standards. The small size helps to reduce blocking and ringing artefacts, while the precise integer specification eliminates any mismatch between the encoder and decoder in the inverse transform. Extensive spatial prediction within frames is used for improved decorrelation in areas not using temporal prediction. Block- or macroblock-based prediction is used for intra-coded blocks, with all pixels being predicted in the spatial domain using either DC prediction or one of a number of directional prediction modes.

The current draft of the standard specifies two different methods for entropy coding: a simple universal variable length coding (UVLC) method that uses a single reversible VLC table for all syntax elements, and a more complex and efficient context-based arithmetic coding (CABAC) method. One benefit of the UVLC method is that it can provide increased robustness to bit errors.

3 H.26L vs. H.263 and MPEG-4 in Conversational Applications

Before looking at the performance of individual features within H.26L, we will illustrate the gains in coding efficiency that can provide relative to the currently popular video coding standards ITU-T H.263 and ISO/IEC MPEG-4. Our comparison focuses on video coding for conversational video applications, where minimal delay in the encoder-decoder loop and low to medium data rates are key requirements. To fulfill the low delay requirement, no B-pictures are employed by any of the encoders in this comparison. The following standards and profiles are included in this comparison: H.263 Baseline; the H.263 Conversational High Compression (CHC) Profile; the MPEG-4 Simple Profile; the MPEG-4 Advanced Simple Profile, disabling B-pictures and Global Motion Compensation; and H.26L using CABAC for entropy coding. Both H.263 CHC and H.26L used 5 previous reference frame for motion compensated prediction.

The public H.26L TML-8.5 encoder [6], UBC's H.263 Research Library version 0.3 [7], and UB Video's RD-optimized MPEG-4 encoder [8] were used to generate the results. In order to achieve a fair comparison between standards, the motion estimation and coding mode decision processes of all of the encoders were similarly optimized using Lagrangian techniques based on the methods described in [9]. All encoders used full search motion estimation with a search range of 32 pixels. For all sequences, a single I-picture was encoded, followed by a series of P-pictures.

The sequences chosen for this test are popular test sequences that represent a range of typical conversational video content. These sequences are Akiyo and Paris at 15 Hz, and Carphone, Foreman, and Silent Voice at 30 Hz. Average bit rate and PSNR results were collected for each sequence using quantization parameter values of 16, 20, 24 and 28 for H.26L, and 5, 8, 13, and 21 for H.263 and MPEG-4. For comparisons between different standards, a weighted PSNR measure that considers both the luminance and chrominance components is used in order to account for differences in the relative bit usage for luminance and chrominance components between standards. Through objective and subjective testing, we have determined that a weighted PSNR measure in which luminance is given 8 times the weight of each chrominance component is suitable for such comparisons.

The average coding gains provided by H.26L relative to the other standards have been summarized using the method proposed by Bjøntegaard and adopted by VCEG for measuring the average difference between two RD-curves [10,11]. Using this measure, we present results in terms of the average bit rate savings and the average weighted PSNR gain that is achieved by H.26L. Note that these two values are not additive, but are two different expressions of the same coding gain.

Rate-distortion curves illustrating the performance of the various standards are given for the Foreman and Paris sequences in Figure 2. The average coding gain for H.26L relative to all other encoders over the entire test set are shown in Table 1. The results in this table illustrate that – even at this draft stage – H.26L can provide equivalent objective quality at a data rate 24% lower than that required by the most complex profile of H.263, and 42% lower than that of the popular H.263 Baseline.

Average coding gain for H.26L				
	H.263 CHC	**MPEG-4 ASP**	**MPEG-4 SP**	**H.263 Baseline**
% Bit Savings	24.08	28.34	33.48	42.14
PSNR Gain (dB)	+1.20	+1.41	+1.66	+2.25

Table 1: Average coding gain for H.26L versus other encoders

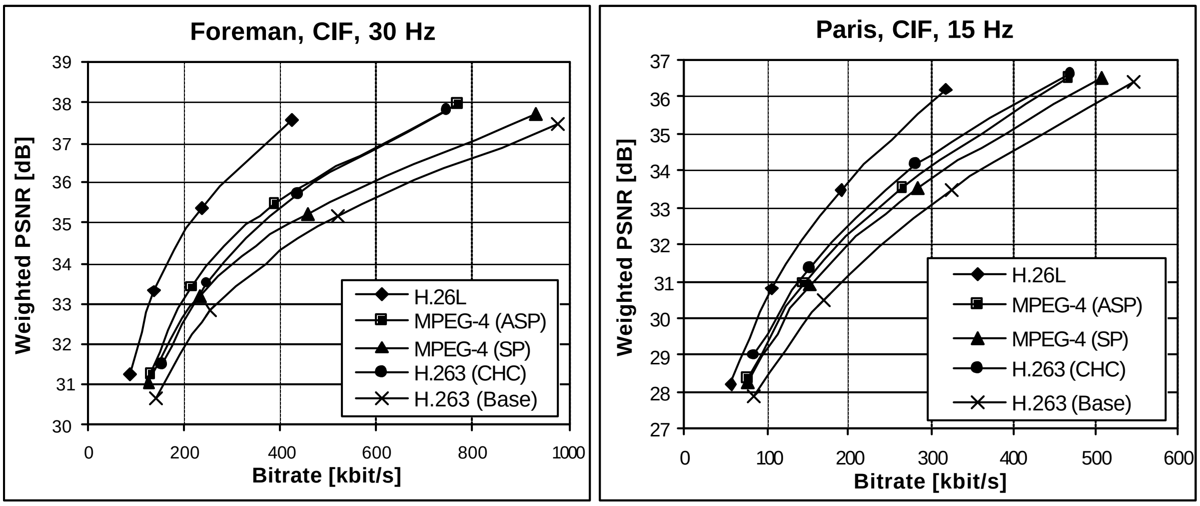

Figure 2: Sample rate-distortion curves for inter-standard coding efficiency comparison

4 Analysis of H.26L Coding Features

Next, we proceed with the analysis of the coding performance of several individual features of H.26L. We compare the performance of the two entropy coding methods available in the current draft and evaluate several components of the H.26L motion compensation model. These components include various combinations of block sizes for motion compensation, fractional pixel motion compensation accuracy and multiple reference frame prediction. Additionally, the coding gain from using multiple reference frames for prediction in H.26L is compared to the gain provided by enabling similar functionality in H.263 using Annex U of that standard.

As in the previous section, results have been generated the rate-distortion optimized mode of the TML-8.5 software and summarized using the Bjøntegaard method. The default configuration for these experiments includes 5 reference frames for prediction, UVLC coding, quarter-pixel accurate motion compensation, all 7 motion compensation block types, and no B-pictures. A full search range of 32 pixels from the predictor was only used for the experiment that evaluates multi-frame prediction, where a smaller search range might limit the use of temporally distant frames. In the other experiments, a search range of only 16 pixels was used, since the smaller search range should not provide any significant advantage or disadvantage to the features being tested.

For this analysis, we use a relatively large set of content with the hope of producing results that are representative of general video content at CIF resolution and lower. The large number of sequences can also help to establish any significant dependence of the performance of coding features on the characteristics of the content and the source resolution. Details of the sequences used are given in Table 2. The majority of the sequences are popular test sequences used in video standardization. The PI sequence contains two people talking and gesticulating relatively close to the camera. Trailblazers is a basketball scene containing fast camera panning and heavy action.

Sequence	Format	Frame Rate	Frames Coded
Container Ship	QCIF	10 fps	100
Foreman	QCIF	10 fps	100
News	QCIF	10 fps	100
Silent Voice	QCIF	15 fps	150
PI	320x240	15 fps	400
Paris	CIF	15 fps	150
Silent Voice	CIF	15 fps	150
Bus	CIF	30 fps	150
Coastguard	CIF	30 fps	300
Flowergarden	CIF	30 fps	250
Foreman	CIF	30 fps	300
Mobile and Calendar	CIF	30 fps	300
Tempete	CIF	30 fps	260
Trailblazers	CIF	30 fps	300

Table 2: Test sequences used in H.26L analysis.

Motion Compensation Block Size

For the purpose of motion compensation, each 16x16 macroblock can be partitioned in one of seven ways, from using a single motion vector for the entire block to using sixteen individual motion vectors for each of the 4x4 blocks that compose a macroblock as illustrated in Figure 1. Smaller blocks are intended to improve motion representation, especially in areas with fine motion detail, at the cost of larger overhead for transmitting motion vectors in the bitstream.

Supporting a large number of different block sizes, particularly smaller blocks, increases algorithmic complexity in both the encoder and decoder. Modern processors can deal more efficiently with larger blocks because they allow for a larger amount of data to be processed in the same way, eliminating additional overhead for looping and branching. Additionally, support for a larger number of block sizes directly increases the computational complexity of the motion estimation process in the encoder by increasing the number of possible options that can be tested when searching for the best match for macroblock.

In our comparisons, smaller blocks are added incrementally in logical groupings. We also include the combination of 16x16 and 8x8, which are the two sizes found in the currently popular H.263 and MPEG-4 standards. Details of the block size combinations used are shown in. Other encoder settings for this experiment included use of five reference frames, UVLC entropy coding, quarter-pixel motion vector accuracy and a search range of 16 pixels. Average coding gain results comparing the benefit of the various combinations of block types versus using only 16x16 blocks for motion compensation only are presented in. A typical rate-distortion plot that illustrates the effect of using different combinations of block sizes for the Tempete sequence is provided in Figure 3.

This experiment reveals that using all 7 motion compensation block types provides approximately 16% bit savings versus using 16x16 blocks only. However, we observe that by allowing only blocks that are 8x8 and larger (modes 1-4), greater than 80% of the bit savings realized allowing all of the block sizes can be captured. Moreover, block sizes smaller than 8x8 tend to be useful only at relatively high bit rates. The 4x4 block size provides minimal PSNR improvement for all of the tested content. Lastly, our results suggest that smaller block sizes provide less benefit for higher resolution sequences. In particular, the benefit of adding more motion compensation block types for the Silent Voice sequence is smaller at CIF resolution than at QCIF.

Code	Block types used
1	16x16
1, 4	16x16, 8x8
1 to 3	16x16, 16x8, 8x16
1 to 4	16x16, 16x8, 8x16, 8x8
1 to 6	16x16, 16x8, 8x16, 8x8, 8x4, 4x8
1 to 7	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4

Table 3: Key to combinations of motion compensation block sizes tested.

			1, 4		1 to 3		1 to 4	
Res.	**Hz**	**Sequence**	**%Bits**	**PSNR**	**%Bits**	**PSNR**	**%Bits**	**PSNR**
QCIF	10	Container	13.60	0.699	15.76	0.807	16.79	0.872
		Foreman	8.94	0.505	12.22	0.682	13.49	0.774
		News	10.05	0.586	10.98	0.631	13.33	0.786
	15	Silent Voice	7.19	0.347	9.48	0.454	11.38	0.561
320x240	15	PI	12.44	0.703	13.61	0.758	16.27	0.925
CIF	15	Paris	12.75	0.677	13.44	0.700	15.61	0.831
		Silent Voice	4.56	0.191	7.73	0.325	8.34	0.354
	30	Bus	12.11	0.622	15.56	0.800	17.05	0.886
		Coastguard	6.72	0.243	9.67	0.351	10.67	0.391
		Flowergarden	14.39	0.804	16.26	0.906	17.43	0.980
		Foreman	7.69	0.374	12.00	0.589	12.50	0.620
		Mobile	10.28	0.500	11.55	0.561	13.04	0.641
		Tempete	9.20	0.395	12.19	0.529	12.91	0.565
		Trailblazers	8.39	0.400	11.13	0.522	12.75	0.605
		MEAN	**9.88**	**0.503**	**12.26**	**0.615**	**13.68**	**0.699**
		MIN	**4.56**	**0.191**	**7.73**	**0.325**	**8.34**	**0.354**
		MAX	**14.39**	**0.804**	**16.26**	**0.906**	**17.43**	**0.980**

			1 to 6		1 to 7	
Res.	**Hz**	**Sequence**	**%Bits**	**PSNR**	**%Bits**	**PSNR**
QCIF	10	Container	19.19	1.017	19.59	1.039
		Foreman	16.08	0.950	16.38	0.970
		News	16.12	0.984	16.42	1.013
	15	Silent Voice	12.71	0.641	12.74	0.645
320x240	15	PI	19.21	1.130	19.51	1.153
CIF	15	Paris	19.02	1.048	19.21	1.069
		Silent Voice	9.28	0.402	9.12	0.396
	30	Bus	19.80	1.056	20.00	1.072
		Coastguard	11.67	0.434	11.90	0.443
		Flowergarden	23.31	1.359	23.59	1.381
		Foreman	14.89	0.756	14.85	0.755
		Mobile	16.13	0.814	16.38	0.830
		Tempete	15.07	0.673	15.26	0.683
		Trailblazers	14.69	0.710	14.88	0.722
		MEAN	**16.23**	**0.855**	**16.42**	**0.869**
		MIN	**9.28**	**0.402**	**9.12**	**0.396**
		MAX	**23.31**	**1.359**	**23.59**	**1.381**

Table 4: Coding gain summary for motion compensation block sizes.

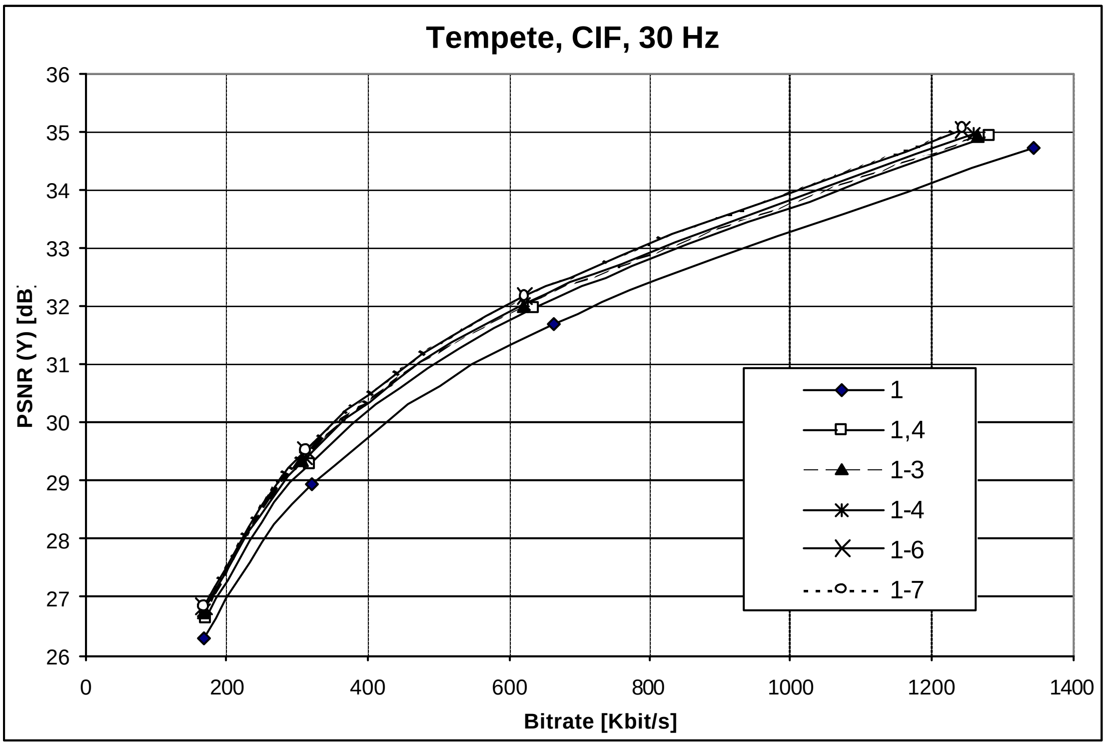

Figure 3: Rate-distortion performance of various combinations of block types for Tempete.

Multiple Reference Frame Prediction

Allowing motion compensated predictions to come from more than only the most recent temporally previous reference frame has been shown to enable improved coding efficiency for many types of content. The idea is to add a temporal component to the motion vectors, permitting the selection of one of several reference frames at the macroblock level. Memory requirements are increased in both the encoder and decoder in order to keep a buffer of reference frames. Furthermore, the computational complexity of the encoder's motion estimation process may increase significantly as additional reference frames are included in the search space.

In these results, we only consider the performance of a sliding window reference frame buffer, in which only the N most recent reference frames are stored and available for use in motion compensated prediction. Support for an adaptive buffering mechanism, in which specific reference frames could be stored for an arbitrary amount of time (with a constraint on the total number of frames in the buffer) has been adopted in principle by VCEG for inclusion in H.26L, but is not a part of the current draft.

In this experiment, the sequences were coded once using only a single reference frame, and once with five reference frames. UVLC coding and quarter-pixel motion compensation were used and the search range was set to 32 pixels from the predictor.

This experiment also provides an opportunity to compare the benefit of using multiple reference frames in H.26L and H.263. UBC's RD-optimized H.263 encoder was used with Annexes D, F, I, J, T, and optionally Annex U. A full search range of 32 pixels was used with quantization parameter values of 5, 8, 13 and 21. Summary coding gain results for multi-frame prediction are presented in Table 5. Sample RD-curves are given in Figure 4.

			H.26L		H.263		H.26L/H.263	
Res.	**Hz**	**Sequence**	**%Bits**	**PSNR**	**%Bits**	**PSNR**	**%Bits**	**PSNR**
QCIF	10	Container	2.73	0.135	16.83	0.841	0.162	0.161
		Foreman	4.34	0.246	16.31	0.932	0.266	0.264
		News	0.95	0.055	2.06	0.107	0.461	0.514
	15	Silent Voice	4.07	0.199	7.58	0.362	0.537	0.550
320x240	15	PI	1.65	0.090	2.85	0.146	0.579	0.616
CIF	15	Paris	3.36	0.173	7.29	0.372	0.461	0.465
		Silent Voice	5.69	0.240	6.36	0.331	0.895	0.725
	30	Bus	6.76	0.336	11.82	0.628	0.572	0.535
		Coastguard	0.53	0.019	2.29	0.090	0.231	0.211
		Flowergarden	6.32	0.329	10.10	0.610	0.626	0.539
		Foreman	3.19	0.153	11.82	0.541	0.270	0.283
		Mobile	20.05	1.027	30.78	1.935	0.651	0.531
		Tempete	18.64	0.845	28.90	1.360	0.645	0.621
		Trailblazers	1.15	0.050	1.55	0.070	0.742	0.714
		MEAN	**5.67**	**0.278**	**11.18**	**0.595**	**0.507**	**0.481**
		MIN	**0.53**	**0.019**	**1.55**	**0.070**	**0.162**	**0.161**
		MAX	**20.05**	**1.027**	**30.78**	**1.935**	**0.895**	**0.725**

Table 5: Average coding gain summary for multi-frame prediction in H.26L and H.263. The two rightmost columns show the relative coding gain provided in H.26L compared to H.263.

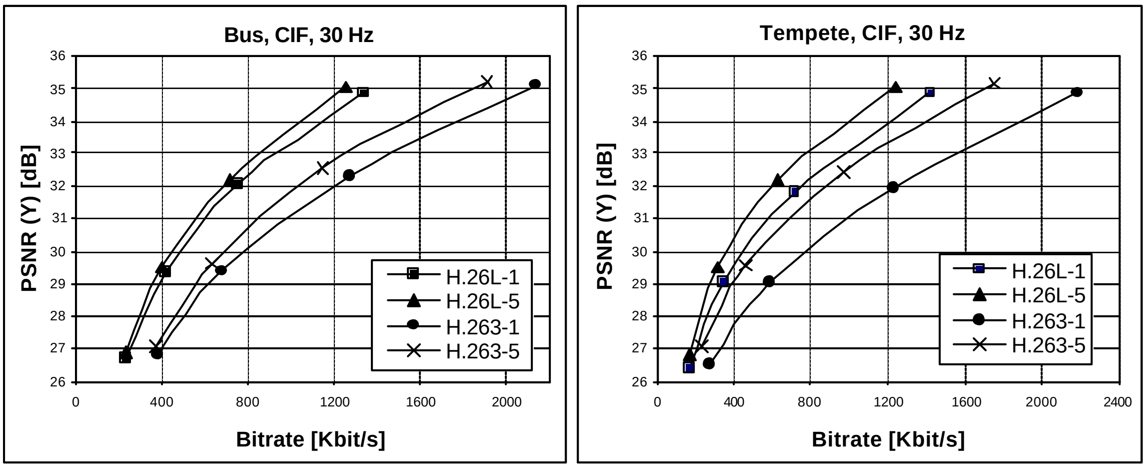

Figure 4: Rate-distortion performance of multiple reference frame prediction (1 versus 5 frames) in H.26L and H.263 for Bus and Tempete sequences.

These results illustrate that the performance of multi-frame prediction in H.26L is highly dependent on the source content. For the majority of sequences, average bit savings are less than 5%. Yet two sequences – Mobile and Tempete – show savings around 20%, nearly 3 times greater than the gains shown for any of the other 12 sequences. This suggests that an intelligent encoder might benefit from recognizing content for which multiple reference frames show little benefit by limiting the searching of multiple reference frames in order to save processor cycles could be better utilized for some other purpose. Or, perhaps a more complex adaptive buffering scheme should be implemented in place of the simple sliding window method in order to benefit from multiple reference frame prediction.

Comparing to H.263, we can observe that searching 5 frames in a sliding window buffer in H.26L provides approximately half of the gain that can be provided by searching a similar buffer using Annex U in an H.263 encoder. The main reason for this difference is that other features of H.26L that are not in H.263, such as quarter-pixel motion compensation and the availability of more block sizes, already capture some of the coding gain that Annex U can capture in H.263. Additionally, the macroblock level syntax for multi-frame prediction allows for less flexibility in H.26L, since the entire macroblock must be predicted from the same reference frame, whereas Annex U allows each 8x8 block to be predicted from a different frame. This result suggests that some improvements might be possible in the H.26L syntax to give more flexibility at the macroblock level.

Fractional Pixel Motion Compensation Accuracy

The current draft of H.26L specifies a default of quarter-pixel and includes an optional one-eighth-pixel accuracy mode for motion compensation. Eighth-pixel accuracy [12] increases the computational complexity of both the encoder and decoder due to the larger number of phases and taps necessary in the interpolation filter. Also, since motion vectors must be transmitted with higher spatial accuracy, the overhead in the bitstream is essentially doubled for eighth-pixel accuracy.

This experiment assesses the benefit of increasing the spatial accuracy of motion vectors from quarter-pixel to eighth-pixel. Five reference frames, UVLC entropy coding, and a search range of 16 pixels were used. Average coding gain results are presented in Table 6 and sample RD-curves in Figure 5.

Res.	Hz	Sequence	% Bits Savings	PSNR Gain
QCIF	10	Container Ship	0.57	0.032
		Foreman	-5.60	-0.314
		News	-3.47	-0.205
	15	Silent Voice	-6.05	-0.285
320x240	15	PI	-4.66	-0.252
CIF	15	Paris	-1.85	-0.097
		Silent Voice	-5.28	-0.217
	30	Bus	2.50	0.129
		Coastguard	-2.50	0.089
		Flowergarden	6.80	0.380
		Foreman	-5.22	-0.254
		Mobile	10.78	0.574
		Tempete	1.88	0.082
		Trailblazers	-5.99	-0.265
		MEAN	**-1.29**	**-0.043**
		MIN	**-6.05**	**-0.314**
		MAX	**10.78**	**0.574**

Table 6: Average coding gain summary for eighth-pixel motion compensation accuracy. Negative bit savings and PSNR gain values indicate that eighth-pixel offers worse RD-performance than quarter-pixel.

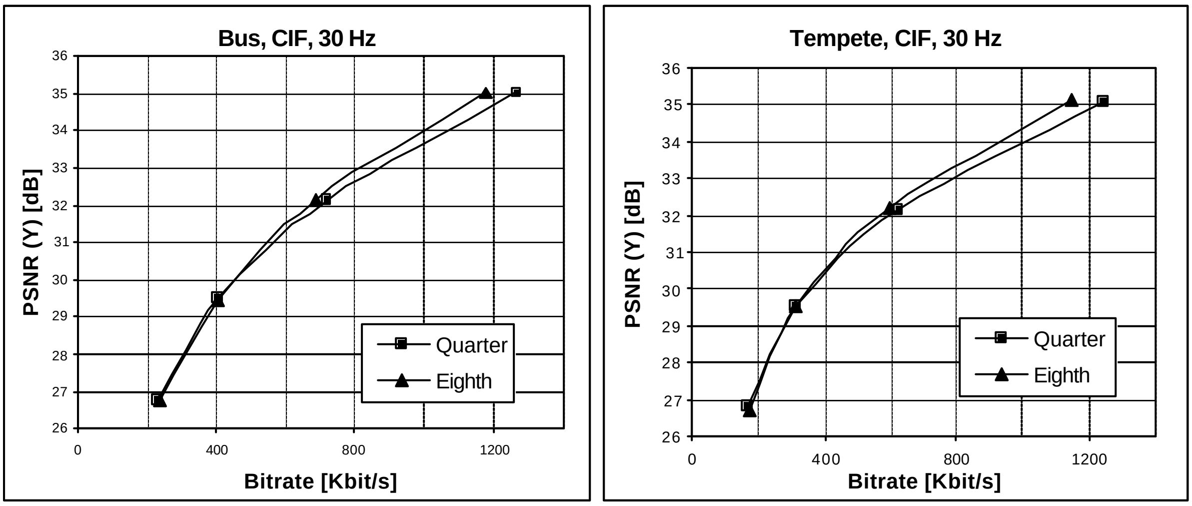

Figure 5: Rate-distortion performance of quarter-pixel and eighth-pixel motion compensation for Bus and Tempete.

From the average coding gain summary, it seems that eighth-pixel accuracy provides little benefit considering its added complexity, particularly for the QCIF sequences. However, from the rate-distortion curves, it is clear that the benefit provided by using eighth-pixel accuracy increases as the fidelity of the coded video is increased. This occurs because the motion vector overhead occupies a smaller fraction of the total bit rate as the rate is increased. Therefore, eighth-pixel would likely be beneficial in profiles meant to target applications that require high-quality video, digital cinema for example.

Entropy Coding Methods

In our final experiment, we compare the rate-distortion performance of the two entropy coding methods that are available in H.26L. The UVLC method offers ease of implementation and computational simplicity by using a single, reversible variable-length code table for all syntax elements. This method provides improved robustness to bit errors. The more efficient CABAC method is adaptive to content and offers near-optimal entropy coding over all different types of content and bitrates, but with complexity drawbacks. For further details and results, see [13].

In this final experiment, quarter-pixel motion compensation and 5 reference frames were used. The search range was 16 pixels from the predicted motion vector. Average coding gain results that summarize the coding gain provided by CABAC relative to UVLC are presented in Table 7 and sample RD-curves are in Figure 6.

Res.	Hz	Sequence	% Bits Savings	PSNR Gain
QCIF	10	Container Ship	4.96	0.251
		Foreman	7.56	0.436
		News	4.98	0.295
	15	Silent	4.81	0.238
320x240	15	PI	6.35	0.355
CIF	15	Paris	5.47	0.288
		Silent	6.94	0.298
	30	Bus	9.49	0.485
		Coastguard	9.50	0.376
		Flowergarden	9.98	0.536
		Foreman	13.07	0.686
		Mobile	9.39	0.467
		Tempete	7.54	0.329
		Trailblazers	7.48	0.349
		MEAN	**7.68**	**0.385**
		MIN	**4.81**	**0.238**
		MAX	**13.07**	**0.686**

Table 7: Average coding gain summary for CABAC entropy coding versus UVLC coding.

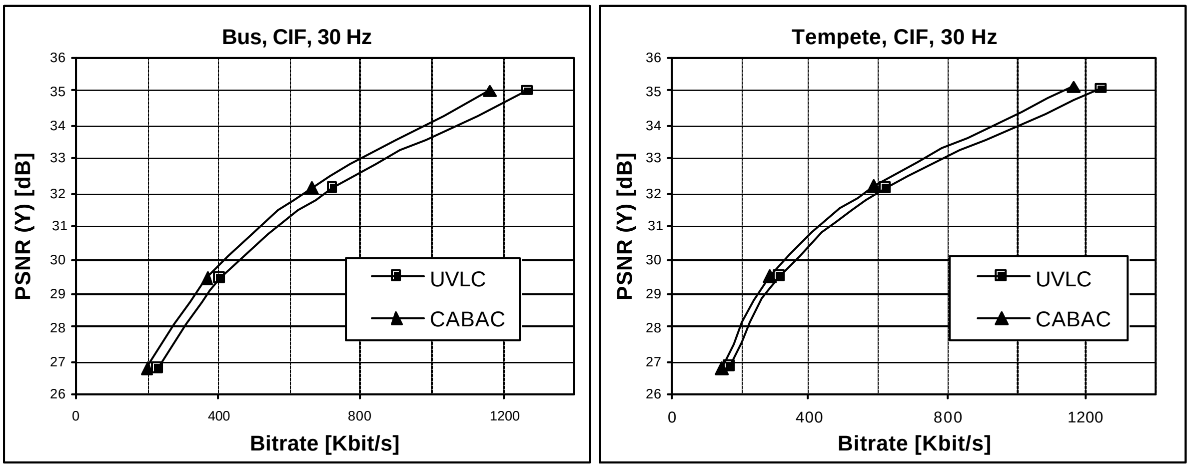

Figure 6: Rate-distortion performance of entropy coding methods for Bus and Tempete.

These results show that CABAC provides consistent gains in coding efficiency, typically between 5 and 10 percent, over a wide variety of content. CABAC tends to provide larger gains for the higher resolution sequences. From the RD-curves, we note that the coding efficiency gains are present at all bit rates, and tend to be largest at very low and very high bit rates.

5 Conclusions

In this work, we first illustrated that the draft H.26L standard can provide substantial improvements in coding efficiency relative to existing video coding standards in conversational video applications. We then presented a large set data that was generated using the rate-distortion optimized mode of the H.26L TML-8.5 software in order to analyze the performance of several features of the current H.26L draft standard. The results indicate that the CABAC mode provides a fairly consistent improvement in coding efficiency of between 5 and 10%. This seems to be largest for higher resolution sequences and at very low bit rates. Coding gains using multiple reference frame prediction seem to be highly dependent on source content. While large improvements are produced for 2 sequences, average bit savings are less than 5% for the majority of the 14 tested sequences. Our observations with regards to the benefit of eighth-pixel accuracy indicate that such is only beneficial at high resolutions and high bit rates, and also contain high spatial detail. Finally, the use of many different block sizes provides a consistent improvement, averaging 16% bit savings if all block types are used versus using the 16x16 mode only. However, using only 8x8 and larger blocks can capture most of the benefit of the different block sizes, although the smaller blocks become more beneficial as the bit rate is increased.

6 References

[1] ITU-T/SG16/VCEG (Q.6), H.26L Test Model Long-Term Number 8 (TML-8), Document VCEG-N10, Video Coding Experts Group (VCEG) 14th meeting, Santa Barbara, CA, USA, 24-27 September, 2001.

[2] P. Topiwala, G. Sullivan, A. Joch, F. Kossentini, "Performance Evaluation of H.26L TML-8 versus H.263++ and MPEG-4", Document VCEG-N18, VCEG 14th meeting, Santa Barbara, CA, USA, 24-27 September, 2001.

[3] P. Topiwala, G. Sullivan, A. Joch, F. Kossentini, "Overview and Performance Evaluation of the ITU-T Draft H.26L Video Coding Standard," Proc. SPIE, Appl. of Digital Image Processing, August 2001.

[4] A. Joch and F. Kossentini, "Performance analysis of H.26L coding features," Document VCEG-O42, VCEG 15th meeting, Pattaya, Thailand, 4-6 December, 2001.

[5] S. Wenger and T. Stockhammer, "An Overview of the H.26L NAL Concept," Document JVT-B028, JVT 2nd meeting, Geneva, Switzerland, 29 January - 1 February, 2002.

[6] ITU-T/SG16/VCEG (Q.6), H.26L TML-8.5 software, available online at ftp://standard.pictel.com/video-site/h26l/tml85.zip.

[7] Signal Processing & Multimedia Group, University of British Columbia, "ITU-T H.263 Research Library, version 0.3", April, 2001. http://spmg.ece.ubc.ca.

[8] http://www.ubvideo.com.

[9] G. Sullivan and T. Wiegand, “Rate-Distortion Optimization for Video Compression”, IEEE Signal Processing Magazine, pp. 74-90, November 1998.

[10] G. Sullivan, "Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material," Document VCEG-N81, VCEG 14th meeting, Santa Barbara, CA, USA, 24-27 September, 2001.

[11] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," Document VCEG-M33, VCEG 13th meeting, Austin, TX, USA, 2-4 April, 2001.

[12] Thomas Wedi, "1/8-pel Displacement Vector Resolution for TML-6," Document VCEG-M45, VCEG 13th meeting, Austin, TX, USA, 2-4 April, 2001.

[13] D. Marpe, G. Blättermann, G. Heising, T. Wiegand, "Further Results for CABAC entropy coding scheme," Document VCEG-M59, VCEG 13th meeting, Austin, TX, USA, 2-4 April, 2001.