

COMPLEXITY-DISTORTION ANALYSIS OF H.264/JVT DECODERS ON MOBILE DEVICES

Alan Ray, Hayder Radha

Michigan State University

ABSTRACT

Operational complexity-distortion curves for H.264/JVT decoding are generated and analyzed for low-complexity mobile devices under a variety of bitrate constraints. The focus of our study is on achieving optimum complexity-distortion operational points by evaluating different combinations of Group of Picture (GoP) types and varying the Quantization Parameter (QP) and entropy encoder (arithmetic or universal) to meet the desired rate constraints. Using a 400Mhz Intel PXA255 platform (found in popular iPaq devices), complexity-distortion curves are developed for common GoP structures and a wide range of QP values. The curves, based on extensive operational experimentation, indicate that under typical conditions for mobile platforms (low to mid computational complexity), I or I & P-frame combinations outperform the more compressed streams that include B-frames. In the 25-45dB range, the optimum complexity-distortion I or I & P structures outperformed B-frame structures by up to 21% in complexity for the equivalent distortion level and under the *same* bitrate constraints. Further, under the *same* complexity *and* bitrate constraints, selecting the optimum GoP structure achieves as much as a 10dB PSNR improvement.

1. INTRODUCTION

The emerging H.264/JVT video standard [1] has received a great deal of attention due to its coding efficiency when compared with previous standards such as baseline MPEG-4, MPEG-2, and H.263. The coding-efficiency advantages of H.264, however, come at the expense of higher computational complexity. For example, the study in [2] showed that H.264 decoders could exhibit more than double the complexity of H.263 decoders. Furthermore, previous studies have shown that fractional-pixel motion-compensation interpolation and the loop filtering consume a significant amount of computational power in emerging

H.264 decoders [2,3]. Since these operations are part of the baseline (required) part of H.264, there is a need to evaluate new ways for minimizing both complexity and distortion for H.264 decoders on low-complexity devices. In particular, new wireless handhelds have both complexity and bitrate constraints, yet the range of these constraints differ from traditional systems (e.g., powerful PCs that are networked over the best-effort Internet). Under common operational scenarios, a low-complexity wireless device may have significantly greater complexity/power constraints than bitrate limitation (e.g., over a wireless access LAN).

In this paper, we evaluate the feasibility of achieving optimum complexity-distortion operational points for low-complexity H.264 decoders by adapting the picture types (I, P, and B) under different bitrate constraints. We show that, over a wide range of QP values, adapting different Group of Picture (GoP) structures provides the H.264 coding system the option for realizing optimum complexity-distortion points while adhering to a certain rate requirement. Based on extensive operational complexity-distortion analysis, we show that for low- to mid-complexity constraints, and under the *same* bitrate constraint, an all I-frame or hybrid I and P frame H.264 structure could provide the optimum complexity-distortion curve compared with compressed H.264 streams that include one or more B-frames in their GoP structure.

The remainder of this paper is organized as follows. Section 2 describes the overall experimental set-up for our study. Section 3 describes our results and the implications for the H.264 decoder, and section 4 summarizes the analysis and future questions of interest.

2. EXPERIMENT SETUP

Our study uses three main components, which are discussed below: A mobile device as a testbed, an extensive selection of encoded video clips, and an optimized version of a publicly available H.264 decoder.

2.1 Experimental Platform

We used a standard HP iPaq 5550 with 128MB RAM and a 400Mhz Intel PXA255 processor running Microsoft PocketPC 2003 as our test platform. During the experiments, all extraneous background processes were terminated, including wireless communication. We eliminated network performance issues by storing encoded sequences in files and downloading them for each experiment. No additional effort was made to prioritize the decoding thread, so the operating system handled memory management and scheduling in its normal manner. A simple DOS shell was used on the iPaq in lieu of developing a graphical user interface for the decoder, further simplifying the operating system calls. Shell output was limited to non-timed portions of the tests.

2.2 Encoder and Sequence Selection

The standard JVT reference decoder version JM 6.1e [4] was used, with slight modifications to the interface to facilitate data gathering. Table I shows the standard encoder settings. Three video test sequences with different characteristics were selected: Akiyo for its high compression rates, Foreman for its higher motion and panning, and Mobile because of its coding difficulty. Four different GoP structures were tested; in each of them every twelfth frame was an I frame to refresh the sequence. The first structure was an all I sequence. The second was an all P sequence. The third alternated between P and B frames (I-B-P-B-P...). The final sequence used two B frames between P frames (I-B-B-P-B-B-P...). In the results, they are referred to as I, P, B and T sequences respectively. For a given sequence, the same QP value was used for I, P, and B frames. Each sequence was encoded and timed using both entropy modes: arithmetic (CABAC) and context-based adaptive variable-length coding (CAVLC). The CAVLC entropy coding mode used in the current H.264 standard is different from a universal variable length coding mode used in earlier draft versions.

Parameter	Value
QP Value	0-51; each trial used a constant QP for all frames.
Frame Rates	5, 10, 15 fps
Format	QCIF (176x144)
I-Frame Frequency	Every 12 frames
Hadamard Transform	On
Max. Search Range	16
Num. Reference Frames	1
Forced Intra-Macroblocks	None
Block Search Restrictions	None
Slices	Unused
SP Frames	Unused
Entropy Coding	CAVLC & CABAC
Loop Filter Parameters	Default

Table I: Encoder Options

Sequence	Complexity @ 25dB	% Diff.	Complexity @ 45dB	% Diff.
Akiyo (CABAC)	P: 0.385 B: 0.44	14.2%	P: 0.44 B: 0.49	11.4%
Foreman (CABAC)	I: 0.35 B: 0.425	21.4%	I: 0.55 B: 0.58	5.4%
Mobile (CABAC)	I & P: 0.31 B: 0.34	9.7%	P: 0.52 B: 0.55	5.8%
Akiyo (CAVLC)	P: 0.32 B: 0.36	11.1%	P: 0.41 B: 0.45	9.8%
Foreman (CAVLC)	I: 0.36 B: 0.40	10.0%	I: 0.67 B: 0.73	9.0%
Mobile (CAVLC)	P: 0.405 B: 0.42	3.7%	I: 0.82 B: 0.91	11.0%

Table II: Relative Complexity of best I or P vs. best B or T sequence at 25 and 45dB, shown for 10 fps

2.3 Decoder Optimizations

The JM 6.1e decoder [4] was used as the baseline code for the decoder. Many modifications were made to the decoder to streamline the code and improve performance. The changes included using circular buffers instead of memory copying, streamlining bit-oriented processing, and reducing calls to the most frequent functions. The results presented later for CAVLC entropy decoding significant improvements to the reference software's algorithm. The primary improvements involved additional caching and optimization of frequently called functions. Despite the generally greater complexity of CAVLC when compared to CABAC, the CAVLC decoding has been improved an additional 10-15% compared to the CAVLC algorithms in the reference software.

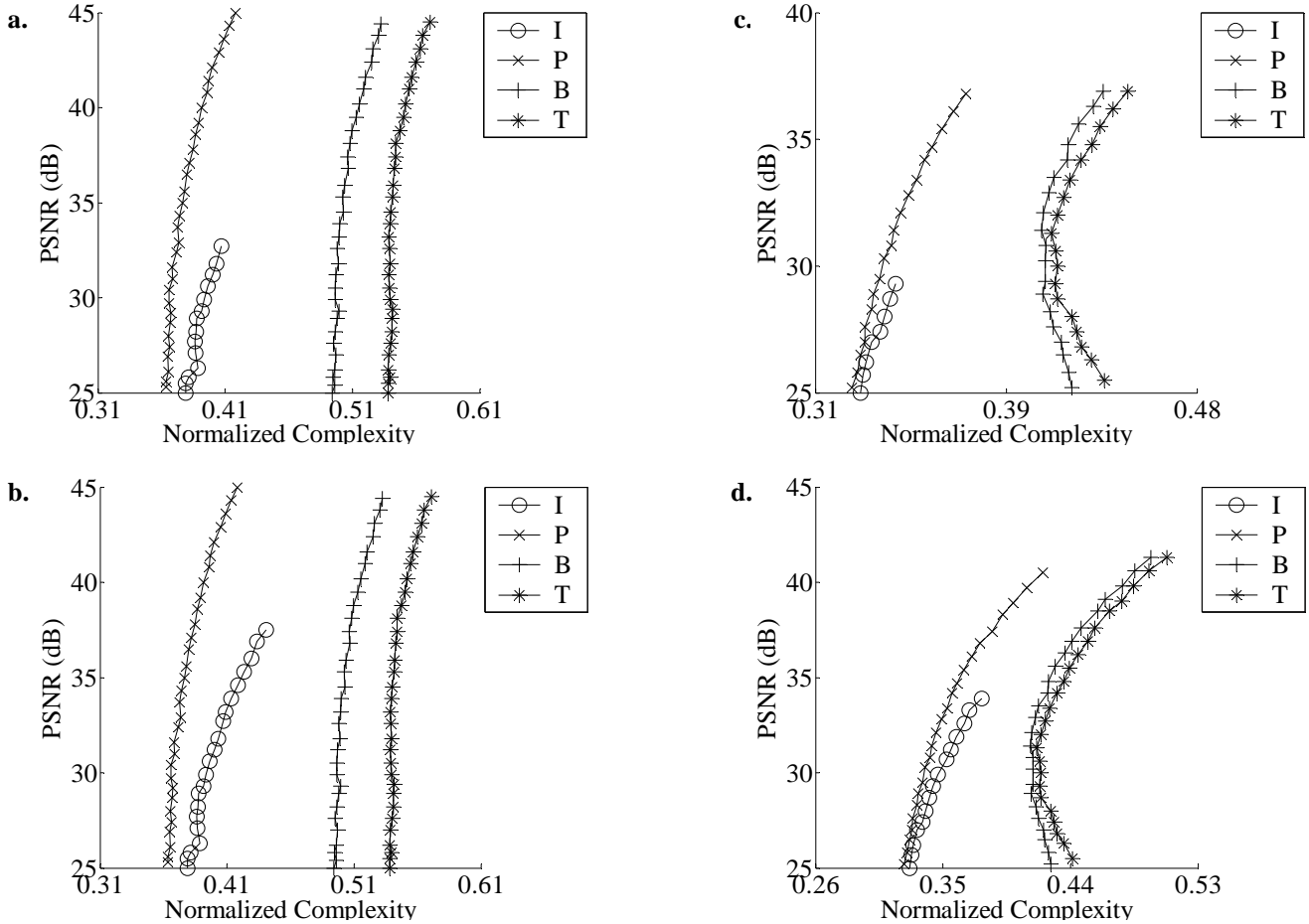


Figure I: 15fps CABAC Complexity-Distortion Curves
(a) Akiyo <1KBpf, (b) Akiyo <2KBpf, (c) Foreman <100Kbps, (d) Foreman <200Kbps

Changes did not affect the numerical accuracy of the decoder. Due to the lack of WindowsCE profiling tools, a limited number of timing statements were introduced into the decoder, but frequency of these was negligible compared to the overall decoder complexity. Our configuration used the encoder-generated parameters for the leaky bucket parameters, but no additional rate control was implemented. Decoder timings ignored the program initialization time and simple timed from beginning of the first frame to the end of the last. The code was built with Intel's compiler for WinCE and Microsoft's Embedded Visual C++ 4.0 linker. The compiler flags were optimized for speed; file size was ignored.

2.4 Performance Metrics

The aforementioned four different GoP structures (I, P, B, T) were compared for the three different video clips (Akiyo, Foreman, Mobile) using two different entropy codings (CAVLC, CABAC). Different frame rates (5, 10, and 15 fps) were also tested to explore whether the

more temporally related streams made a significant difference in the complexity-distortion curves. Each video clip/sequence pairing was timed for each quantization parameter (0-51). However, results were only thoroughly examined for quantization (QP) values that provided the more practical range of 25-45dB. Since no rate control was used, bitrates are based on the total number of bits generated for the given QP value used. Results are shown for single consecutive runs as experiments indicate that identical runs tend to vary by only 1-2%. Certain specific sequences seem to have unusual complexities and break the smooth complexity curves that characterize most of the data. Once timing data had been gathered for a given sequence and set of GoP structures, various bit rate limits were selected and plotted. All complexity data has been normalized as a fraction of the time it takes to decode a 0-QP all-I frame sequence of the video sequence in question.

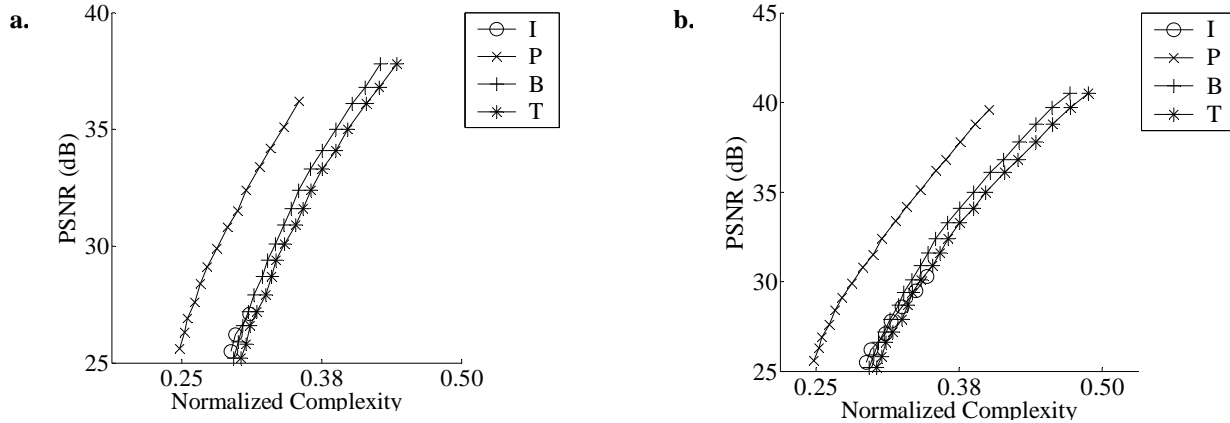


Figure II: 15fps Mobile CABAC Complexity-Distortion Curves: (a) <4KB/frame, (b) <6KB/frame

3. RESULTS & ANALYSIS

3.1 Arithmetic Coding (CABAC)

Regardless of the framerate selected, I, P, B, and T sequences performed similarly for a given video clip. The T (two B frames between every P frame) sequences were always slightly more computationally-complex for a *given distortion* than the B sequence. Likewise, the P sequence was less complex than the B sequence. The performance of the all I sequence varied greatly: It did very poorly on the highly compressible Akiyo, while compared similarly with the P GoP for portions of Foreman, and varied widely for Mobile. Table II shows the performance gain from selecting a larger, less compressed stream at a low quality setting (25dB) and a high quality setting (45dB).

Despite the higher bandwidth required for the I or P sequences, a significant performance advantage is seen in terms of complexity-distortion optimization under a given maximum bitrate. Figure 1b shows the CABAC encoded Akiyo complexity-PSNR graphs for all four sequences under 2KB per frame (15 frames per second). While the B and T sequences achieve better compression and thus provide higher quality pictures at low bitrates, the P sequence runs 10-14% faster at a given distortion level compared to the I GoP, closer to 30% compared to the B and T GoPs. (The I GoP's best achievable distortion is approximately 34dBs, due to the bandwidth limitation.) More importantly, for a given complexity constraint, selecting the P GoP structure achieves as much as a 10-15dB PSNR improvement over the I GoP, as well as the B and T structures (e.g. a complexity of 0.37 is 25dB for a I GoP, but 35dB for the P GoP).

In general, it's clear that the P GoP decodes more efficiently (in terms of optimum complexity-distortion) but requires higher bitrates relative to the B, and T GoP structures, as seen in the Foreman figures. In the Akiyo example, the highly correlated P frames are coded extremely efficiently so that the P GoP represents the optimum curve. Figure 1a shows results for Akiyo using a maximum bitrate of 25Kbps. Here we see that the rate is too low to support a high-quality pure I-frame sequence, but that the high compressibility of Akiyo allows the P GoP to almost match the distortion of the B and T sequences at a much lower complexity.

Figure 1c-d show the CABAC Foreman sequence with the same 1KB and 2KB per frame limits. Here, the I GoP is similar to the P GoP at high distortions. As the distortion decreases, the P GoP improves to offer a significant performance advantage over the other sequences. This option is attractive when computational power is at a premium and the bandwidth may be more flexible. For example, in Figure 1d, the P GoP shows as much as a 4dB improvement over the I GoP; and as much as 7dB over the B frames for a fixed complexity constraint. (The B and T GoPs in the Foreman deviate from the expectation that lower distortion causes greater complexity. The effect is greatly enhanced because of the focus on the 25-45dB range.)

CABAC encoded Mobile, shown in Figure 2, shows the P GoP running 15 to 20% faster than the alternative options for a given distortion, and a 3-5dB PSNR improvement for a given complexity. Once again, the I GoP is limited by the lack of bandwidth. In addition, for the first time, the B GoP's performance is competitive with the I GoP

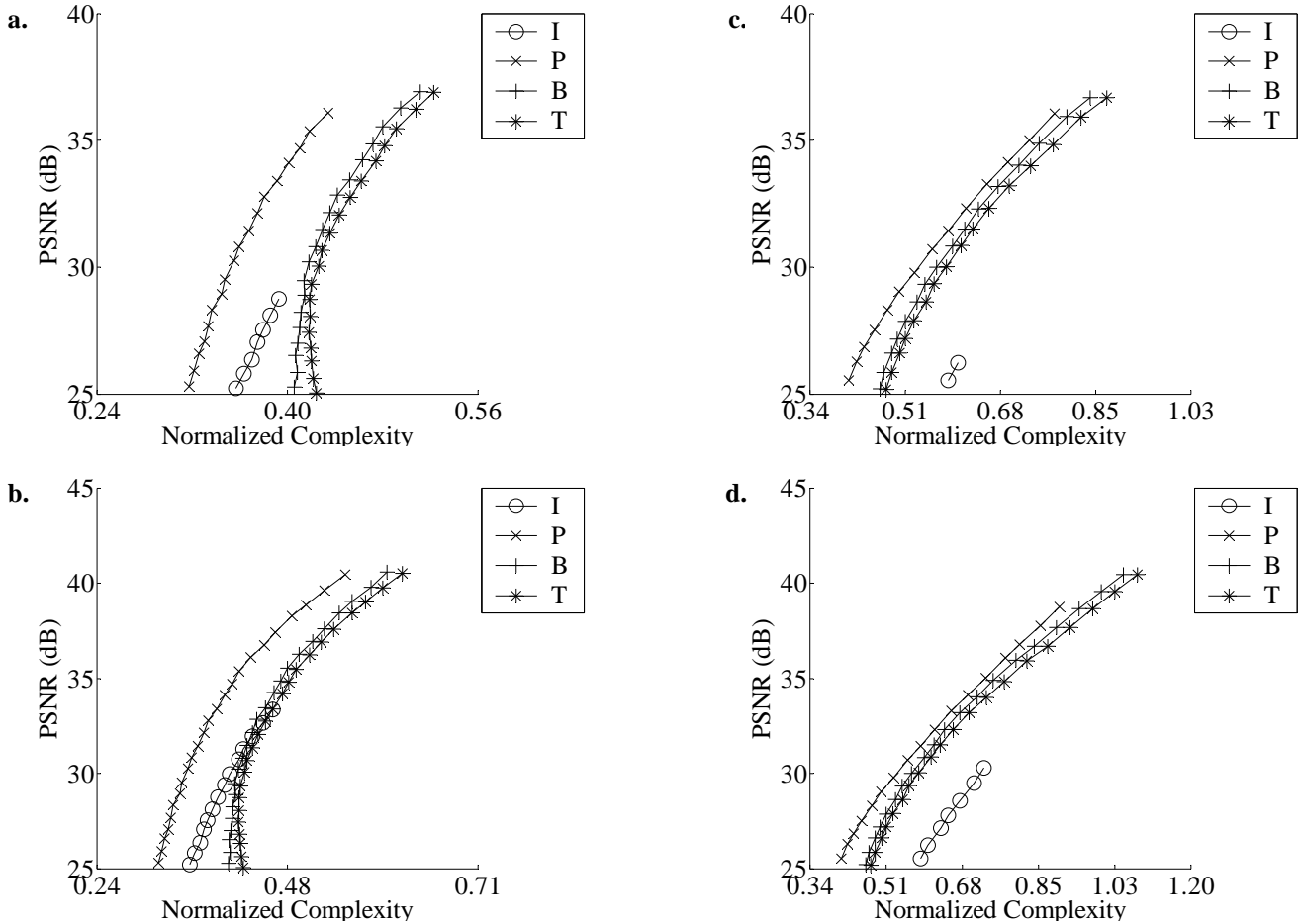


Figure III: 15fps CAVLC Complexity-Distortion Curves
(a) Foreman <1KBpf, (b) Foreman <2KBpf, (c) Mobile <4KBpf, (d) Mobile <6KBpf

The steep distortion improvement slope generated by increasing in bandwidth shows that in terms of complexity, higher bitrate limits are preferable to more compressed GoP structures. As pointed out by Horowitz *et al.* much of the interpolation and loop filtering complexity is fixed as long as a numerically correct decoder is required [3]. These experimental results indicate the extra parsing is much less complicated than additional motion prediction. The general trend shown in these figures continues as bitrate limits are increased: A much lower complexity option using I or P GoPs is available for a small increase in bandwidth.

3.2 Context-based (CAVLC)

As mentioned previously, the context based adaptive variable length entropy coding mode is a relatively new addition to the standard. Much of the previous literature examines the earlier universal variable length coding

mode (UVLC). Horowitz *et al.*, for example, primarily examines the UVLC complexity while footnoting that experimentation suggested CAVLC was roughly twice as complex as UVLC. [3] The scope of our research focuses on the operational complexity-distortion curves of CAVLC and comparing them with the CABAC curves.

Figure III presents the Foreman and Mobile sequences with the same rate limits as shown in Figure Ic-d and Figure II. While the P GoP still performs the best for a given distortion (up to 20% faster) or a given complexity (up to 5dBs), the I GoP performance is significantly degraded. Not only does the I GoP's minimum distortion shrink due to the increased bandwidth used by CAVLC, but its complexity increases much more steeply as a function of distortion. In fact, all four GoPs increase more rapidly in complexity as distortion decreases (compared to CABAC sequences). However, only the I GoP radically

changes its performance relationship to the other GoPs. Figure IIIb suggests that at lower distortion (>32dB), the B and T GoPs' complexity is similar to the I GoP.

Figure IIIc-d, showing the performance of the CAVLC Mobile sequence under two different bitrate constraints, shows another change in the I GoP performance. Under CABAC, the I GoP's complexity was similar to the B or T GoPs for a given distortion. For CAVLC, the I GoP is the slowest for a given complexity. At 30dB, the I GoP is roughly 30% more complex than the B or P GoPs.

Compared to the CABAC curves (without normalization), the CAVLC curves have approximately 10% greater complexity at high distortions (around 25dB). The complexity difference grows, largely based upon bandwidth. For the Mobile sequences at low distortion (and high bitrate), the CAVLC sequences are approximately 50% more complex than the CABAC sequences of the same distortion.

The significant increase in the GoP complexity as distortion decreases, especially for the I GoP, indicates that the CAVLC entropy decoding mode is, in the current implementation, significantly more complex for a given distortion than the equivalent CABAC sequence. This complexity most likely reflects a combination of two factors: The capabilities and weaknesses of the iPAQ architecture and software and the specific implementation of the CAVLC algorithm in the reference software.

Overall, the operational complexity-distortion are very similar for the CAVLC sequences when compared to the CABAC ones in terms of optimal GoP structures. As shown in Table II and Figure III, the curves are very similar despite the slightly higher bitrates required for CAVLC encoded sequences. For this implementation, the CAVLC sequences generally require greater complexity to decode compared to the same sequence encoded using CABAC.

4. CONCLUSIONS

In this paper, we explored the complexity-distortion curves for a mobile H.264/JVT decoding environment, including the impact of rate limits upon the complexity curves. We showed that simpler sequences potentially achieve equal distortion with lower complexity than more compressed sequences. This suggests an efficient real-time encoding for mobile devices may use less computing power and compensate with faster network service. The highly compressible sequences (e.g. Akiyo) benefit greatly from P-frame compression; while more difficult sequences vary in their optimal GoP structures. Alternatively, sequences with quantization parameters for B-frames that are much smaller than I or P frame parameters may lead to better distortion rates without increasing the complexity. These results are also highly dependent upon the mobile network implementation as sufficient network processing will erase the computational savings. Our study suggests that maximizing the network bandwidth could provide a viable approach to achieving high video quality for mobile platforms while maintaining the low-complexity constraints of these platforms.

5. REFERENCES

- [1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Joint Video Team (JVT), Mar. 2003, Doc. JVT-G050.
- [2] V. Lappalainen, A. Hallapuro, and T.D. Hämäläinen, "Complexity of Optimized H.26L Video Decoder Implementation," IEEE CSVT, vol 13., pp. 717-725, July 2003
- [3] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis," IEEE CSVT, vol 13., pp. 704-7716, July 2003
- [4] JVT Reference Software version JM 6.1e via H.264/AVC Software Coordination webpage. Available: <http://bs.hhi.de/~suehring/tml/>