HD-VideoBench. A Benchmark for Evaluating High Definition Digital Video Applications

Mauricio Alvarez*, Esther Salamí*, Alex Ramírez*† and Mateo Valero*†
*Department of Computer Architecture. Universitat Politècnica de Catalunya (UPC), Spain
HiPEAC European Network of Excellence
Email: {alvarez,esalami}@ac.upc.edu
†Barcelona Supercomputing Center-CNS, Spain
Email: {alex.ramirez,mateo.valero}@bsc.es

Abstract—`HD-VideoBench` is a benchmark devoted to High Definition (HD) digital video processing. It includes a set of video encoders and decoders (Codecs) for the MPEG-2, MPEG-4 and H.264 video standards. The applications were carefully selected taken into account the quality and portability of the code, the representativeness of the video application domain, the availability of high performance optimizations and the distribution under a free license. Additionally, HD-VideoBench defines a set of input sequences and configuration parameters of the video Codecs which are appropriate for the HD video domain.

I. INTRODUCTION

Video applications are becoming a very important workload in multiple computing environments, ranging from mobile media players to Internet servers. In order to deliver the increasing levels of quality and compression efficiency that new multimedia applications are demanding, in the recent years a new generation of video coding standards have been defined [1], [2]. Furthermore, the trend towards high quality video systems has pushed the adoption of High Definition (HD) digital video [3]. The combination of the complexity of new video Codecs and the higher quality of HD systems has resulted in an important increase in the computational requirements of the emerging video applications [4], [5], [6]. As a result, new architectures are being proposed with the objective of delivering the required performance of HD video applications [7], [8]. The design and evaluation of these architectures requires a representative benchmark with a well defined operation environment.

Although there are several multimedia benchmarks, such as Mediabench [9], Berkeley Multimedia Workload [10] or EEMBC [11], none of them fulfills all the requirements for a complete HD video benchmark. Some of them use the reference versions of the applications that were written with the purpose of validating the standards but not for high performance. Furthermore, these reference codes usually do not include machine specific optimizations like SIMD instructions. Additionally, most of the existing benchmarks focus on the MPEG-2 (or MPEG-4 at the most), but only a few of them include recent video Codecs like H.264 that incorporates the most recent techniques in video compression technology. Even in the case of including H.264, none of them addresses HD applications, which requires a particular and careful selection

of the coding options and input sequences that, in turn, results in different computational and memory requirements.

In this document, we present `HD-VideoBench`, a benchmark devoted to HD video processing in which all the applications were carefully selected taken into account their representativeness of the video application domain, the availability of high performance optimizations, the portability and quality of the code and the distribution under a free license. This paper is organized as follows. Section II describes the desired conditions in a benchmark for HD video applications and overviews the existing benchmarks for multimedia. Next, Section III presents the set of video encoders and decoders selected for `HD-VideoBench`, Section IV shows the configuration parameters and input sets, and Section V describes the installation and running instructions. Section VI provides some experimental results in terms of coding efficiency and execution time. Finally, Section VII summarizes the main conclusions and guidelines for future work.

II. RELATED WORK

A. Benchmarking Video Codecs

The performance of a video Codec is a function of the available video coding tools (the coding algorithm itself), the actual implementation of these algorithms, the characteristics of the input sequences, and the architecture in which the Codec is implemented. Based on that, we believe that in order to make a comprehensive analysis of video applications, a video Codec benchmark have to meet the following conditions: First, the benchmark should include complete applications (not only kernels) that implement all the features defined in the standard for a given application domain. Second, the benchmarks have to be optimized for high performance. The implementations of the video standards that are designed with verification purposes could produce misleading results in complexity or architecture evaluations. Optimizations can be platform independent (like fast algorithms for motion compensation) and platform dependent (like SIMD optimizations). Third, a complete set of inputs with different resolution, motion characteristics and spatial details have to be provided. Having only one sequence can lead to confusing results in performance evaluations. Fourth, a detailed list of the coding parameters have to be provided. Those parameters have to be

tuned for the resolutions under study because the performance of the Codecs could change dramatically depending on the selected coding options. Fifth, the programs should be free (as in freedom) in order to be able to access the source code, analyze it, perform changes, and be able to distribute them. The same apply for the input sequences. Sixth, the code has to be easy to port between different processor architectures, compilers and operating systems. Finally, the programs must be representative enough of real life multimedia applications, for example as part of multimedia players used in desktop operating systems. The desired characteristics for a video benchmark can be summarized as follows:

- The benchmarks should be complete applications and implement all the features defined in the standards.
- The Codecs should be optimized for high performance.
- A complete set of input sequences must be provided.
- A detailed description of the coding parameters must be provided.
- Programs and input sequences need to be free.
- The code must be portable.
- Programs must be representative of the multimedia application domain.

B. Multimedia Benchmarks

Table I provides a summary of the existing benchmarks for multimedia. Only the applications related to video processing are detailed.

Mediabench [9] is the most popular multimedia benchmark. For the video domain it includes a MPEG-2 encoder and decoder based on the implementation of the MPEG Software Simulation Group (MSSG) with short input videos in low resolution (352x240 pixels). The MSSG Codec does not implement SIMD optimizations and, in general, it has low performance. An extension of the Mediabench called Mediabench+ [12] tried to solve the limitations of Mediabench by including MPEG-4 and H.263 video Codecs, but they selected the reference implementations (MoMusys and Telenor respectively) and they do not address high definition. Recently, a new version of the Mediabench (called Mediabench II [13]) has been released in which more video Codec applications have been added: it includes Codecs for MPEG-2, MPEG-4, H.263 and H.264. The MPEG-2 Codec is the same MSSG implementation, the MPEG-4 is taken from the `FFmpeg` Codec library, the H.263 Codec is the Telenor implementation, and the H.264 is taken from the reference software (called JM). The main problem with this selection is the combination of reference implementations for some of the Codecs (MSSG for MPEG-2 and JM for H.264) with highly optimized version for others (`FFmpeg` for MPEG-4). On the other hand, although they have increased the resolution compared to the original Mediabench, they do not address HD applications and remains on Standard Resolution (SD). Additionally, Mediabench II provides only one short input sequence (10 frames) and the coding options are not tuned for HD applications.

The Berkeley Multimedia Workload [10] solved the problem of the low resolution of the input sequences by including

Application	Description
libmpeg2	MPEG-2 video decoding
ffmpeg-mpeg2	MPEG-2 video encoding
Xvid	MPEG-4 video decoding
Xvid	MPEG-4 video encoding
ffmpeg-h264	H.264 video decoding
x264	H.264 video encoding

TABLE II
SUMMARY OF HD-VIDEOBENCH APPLICATIONS

inputs with higher resolutions, but they have selected only the MSSG implementation of the MPEG-2 Codec. The EEMBC Digital Entertainment [11] benchmark includes Codecs for MPEG-2 and MPEG-4 video standards using the MSSG and `Xvid` implementations respectively, they address low and standard resolutions and provide a different set of input sequences. Nevertheless, they do not have recent Codecs like H.264 and the coding options and input sequences are not publicly available. Finally, the BDTI Video Encoder and Decoder Benchmark [14] is a set of applications representative of modern video Codecs, but they are not complete video Codec applications. The Codecs seems to be similar to H.264 but the details of the Codec, its sources, the coding parameters, and input sequences are not publicly available.

Thus, none of the available benchmarks for multimedia includes all the desired characteristics for a complete benchmark for emerging video Codec applications and for HD environments. `HD-VideoBench` try to solve all the before mentioned limitations by providing different a set of different video Codec applications optimized for high performance, and providing a complete set of input sequences and coding options tuned for HD applications.

III. THE HD-VIDEOBENCH APPLICATIONS

In this section, a description of the applications included in HD-VideoBench is provided. A description of the reference implementations of the video standards is included for comparison purposes. Table II shows a summary of the HD-VideoBench applications.

A. MPEG-2 Applications

1) MSSG: MPEG Software Simulation Group: The MPEG-2 Reference Video Codec [15] is a widely MPEG-2 Codec used for benchmarking. Nevertheless, it was designed for the verification of the standard, but not for high performance. Because of that, we have not included it in `HD-VideoBench`.

2) FFmpeg MPEG-2 Encoder: `FFmpeg` [16] is a free solution to record, convert and stream audio and video. It includes `libavcodec`, a very complete audio/video Codec library that is capable of encoding and decoding streams in many audio and video Codecs. It is optimized for high performance with fast algorithms and SIMD extensions for X86, PowerPC and other architectures. It is a widely used library for video and audio encoding and decoding in many free software projects like `MPlayer`, `Xine`, `VideoLAN` and others. As a part of the `FFmpeg`, there is a very fast MPEG-2 encoder which includes SIMD optimizations, parallelization

Benchmark	Release Date	License	Video Applications	Input Sequences
Mediabench I	1997	Free	MPEG-2 decoder (MSSG) MPEG-2 encoder (MSSG)	mei16v2.m2v: 352x240 pixels, 30 fps 4 frames YUV sequence: 352x240 pixels
Mediabench+	1999	Free	MPEG-2 decoder (MSSG) MPEG-2 encoder (MSSG) H.263 encoder (Telenor) H.263 encoder (Telenor)	n.a.
Mediabench II	2006	Free	MPEG-2 decoder (MSSG) MPEG-2 encoder (MSSG) MPEG-4 decoder (FFmpeg) MPEG-4 encoder (FFmpeg) H.263 decoder (Telenor) H.263 encoder (Telenor) H.264 decoder (JM 10.2) H.264 encoder (JM 10.2)	704x576, 10 frames, 25fps
Berkeley Multimedia Workload	2000	Free	MPEG-2 encoder (MSSG) MPEG-2 decoder (MSSG)	720x576p, 1280x720p, 1920x1080p (16 frames)
EEMBC Digital Entertainment	2005	Closed	MPEG-2 decoder (MSSG) MPEG-2 encoder (MSSG) MPEG-4 decoder (Xvid) MPEG-4 encoder (Xvid)	Graphic: 720x480p30 (50 frames) Ralgrind: 320x240p25 (30 frames) Sign: 352x240p25 (30 frames) Zoom: 320x240p30 (30 frames) Marsface: 192x192p25 (49 frames)
BDTI Video Benchmarks		Closed	H.264 like decoder H.264 like encoder	n.a.

TABLE I
DESCRIPTION OF EXISTING MULTIMEDIA BENCHMARKS

Test Sequence	Resolution	Frames / second	No. frames	Comments
Blue_sky	720x576 1280x720 1920x1088	25	100	Top of two trees against blue sky. High contrast, small color differences in the sky. Many details. Camera rotation.
Pedestrian_area	720x576 1280x720 1920x1088	25	100	Shot of a pedestrian area. Low camera position, people pass by very close to the camera. High depth of field. Static camera.
Riverbed	720x576 1280x720 1920x1088	25	100	Riverbed seen through the water. Very hard to code.
Rush_hour	720x576 1280x720 1920x1088	25	100	Rush-hour in Munich city. Many cars moving slowly, high depth of focus. Fixed camera.

TABLE III
INPUT SEQUENCES OF HD-VIDEOBENCH

at slice level, and provides very fast algorithms for motion estimation.

3) Libmpeg2: Although `FFmpeg` includes a MPEG-2 decoder, there is another library called `Libmpeg2` [17] that is faster than the `FFmpeg` implementation. `Libmpeg2` is a free library for decoding MPEG-2 and MPEG-1 video streams. It is highly optimized for high performance and include SIMD optimization of the motion compensation and inverse cosine transform routines. Due to its high performance, `Libmpeg2` is a very popular Codec used in many free multimedia players, such as `MPlayer`, `Xine` and `VideoLAN`.

B. MPEG-4 benchmarks

1) Reference code: An ISO reference code of the MPEG-4 video coding standard exists, but it is not convenient for benchmarking due to the same performance reasons mentioned before for other reference implementations.

2) Xvid: `Xvid` [18] is a free implementation of the MPEG-4 video coding standard that supports the MPEG-4 Advanced Simple Profile (ASP). It has algorithmic optimizations for

motion estimation and SIMD optimizations of the most complex kernels. `FFmpeg` also includes a MPEG-4 encoder that has a similar performance than `Xvid`, but `Xvid` provides a higher coding efficiency. `Xvid` is part of other multimedia benchmarks like EEMBC and Berkeley Multimedia Workload, and it is widely used in free multimedia players and transcoder applications.

C. H264 benchmarks

1) JM Reference Codec: Joint Model (JM) [19] is the reference Codec of the H.264 standardization bodies. It is designed for describing and verifying the standard, and it exhibits very low performance; in fact, it is at least one order of magnitude slower than other FFmpeg implementation [20]. Although being included in Mediabench II, it is not recommended for performance evaluations.

2) X264 encoder: `x264` [21] is a free H.264 encoder. It implements most of the standard features and has a lot of algorithmic optimizations for motion estimation, SIMD optimizations, and allows parallel encoding at slice and frame

levels. It is widely used in free encoding applications like `MEncoder`, `GordianKnot` and `VideoLAN`.

3) FFmpeg H.264 decoder: `FFmpeg` includes a H.264 decoder that implements most of the features of the standard. The code is very optimized and include SIMD instructions for the most time consuming kernels. It is widely used in free multimedia players.

IV. HD-VIDEOBENCH INPUT SEQUENCES AND CODING OPTIONS

We have selected three resolutions that are useful for performance analysis in HD video: DVD (720x576), HD-720 (1280x720) and HD-1088 (1920x1088). The original sequences are available from TU Munchen [22] and were taken with a Sony HDW-F900 digital camera at 1920x1080 pixels resolution, 25 frames per second, progressive scan, and using a 4:2:0 chroma subsampling scheme. Table III summarizes the main characteristics of the input sequences.

The rate control mechanism used by the encoders is based on one-pass constant quality (QP) variable bit rate scheme. We do not use multiple pass or constant bit rate mechanisms because HD-VideoBench is for benchmarking the video Codecs, not the rate control algorithms. The equivalence between the quantization parameter of MPEG-2/-4 and H.264 has derived empirically (see Equation 1).

$$H264\_QP = 12 + 6 \cdot \log_2 (MPEG\_QP) \qquad (1)$$

The selected sequence of frames is I-P-B-B. Adaptive placement of B frames is disabled. The only intra frame is the first one. The Motion estimation algorithms used are EPZS (Enhanced Predictive Zonal Search) [23] for MPEG-2 and MPEG-4 and hexagonal search [24] for H.264.

V. RUNNING HD-VIDEOBENCH

At the HD-VideoBench web page[1], we provide a complete description of the benchmark, a link for downloading the source code and input sequences, and a script for automating the installation and execution processes. Furthermore, in order to provide a single front end to execute all the video Codecs, we have selected the `MPlayer` multimedia application. `MPlayer` is a free media player that includes support for multiple video Codecs by using `FFmpeg`, `libmpeg2`, `Xvid` and other multimedia libraries. `MEncoder` is a companion application of `MPlayer` that can encode audio and video in multiple formats. `MPlayer` simplifies the process of installing and running multiple video libraries because MPlayer selects the appropriate Codec and uses it to encode or decode the input video. By default, we have disabled the output of the video to the screen because we are interested in benchmarking the video Codecs not the displaying process. Table IV presents a summary of the commands for running the `HD-VideoBench` applications.

[1]http://people.ac.upc.edu/alvarez/hdvideobench

VI. HD-VIDEOBENCH PERFORMANCE ANALYSIS

The most important metric for analyzing a video Codec is its ability to compress video efficiently with good quality. Table V shows the resultant quality (in terms of the Peak Signal to Noise Ratio PSNR) and the bitrate of the resultant compressed video (in Kbits per second). All the videos have almost the same quality because they have been coded with a constant quantization parameter. At this equal quality, and taken MPEG-2 as the baseline, the MPEG-4 Codec achieves, on average for the four input sequences, a 39,4%, 36,7% and 34,1% compression gains at the 576p25, 720p25 and 1088p25 resolution respectively. H.264 results in bigger compression ratios 48,2%, 49,5% and 51,8% compared to MPEG-2, and 19,9%, 19,4% and 26,4% compared to MPEG-4 for the three resolutions respectively.

The coding capabilities of each video Codec are directly related to their computational complexity. Figure 1 shows the execution time of all the applications for all the sequences and for all the resolutions under study. We have evaluated two versions of each benchmark: a scalar version (plain C code) and a version which includes SIMD optimizations. The results were collected in an Intel IA32 Xeon processor at 2.4 GHz with 512KB of L2 cache with five runs of each application. The benchmark was compiled with gcc-4.1.1 under Fedora Core Linux 5 with kernel 2.6.17. The decoding time is expressed in terms of frames per second. The value of 25 frames per second is showed as an indication of the performance required for real time execution.

Figure 1(a) shows the decoding performance of the three Codecs for the scalar version. In the case of MPEG-2, the performance without SIMD optimization allows for real time decoding at the 576p25 (88 fps) and 720p25 (43 fps) resolutions, but in the case of 1088p25 is below the real time limit (19 fps). For the MPEG-4 case it is not possible to achieve real time for the 1088p25 resolution (9 fps), and for H.264 it is not possible at 720p25 (18 fps) and 1088p25 (8 fps) resolutions. With SIMD optimizations (Figure1(b)) the decoding process achieves an average 2.13X, 1,88X and 1,55X speed-up for the MPEG-2, MPEG4, and H.264 applications. This allows 1088p25 MPEG-2 (41 fps) and 720p25 H.264 (28 fps) resolutions to achieve real time operation. HD MPEG-4 and H.264 however maintain below the real time limit (19 and 13 fps respectively).

Figure 1(c) show the encoding performance for the scalar version. Without SIMD optimizations all the three Codecs encode less than 25 frames per second. For 1088p25 resolution the resultant performance is 3.8, 0.5 and 0.3 fps for MPEG-2, MPEG-4 and H.264 respectively. With SIMD optimizations the applications obtain speed-ups of 2.46X, 2,42X and 2,31X for MPEG-2, MPEG-4 and H.264 respectively. With these optimizations the MPEG-2 encoder is able to perform at real time for the lower resolution (576p25), for the 720p25 and the 1088p25 resolutions the average frame rate is 22.4 and 9.8 fps respectively. For MPEG-4 and H.264, the coding rate is below ten frames per second for all the resolutions; for the

Codec	Application	Execution Command
MPEG-2 decoder	`libmpeg2`	mplayer mpeg2/576p25_blue_sky.avi -vc mpeg12 -nosound -vo null -benchmark
MPEG-2 encoder	`FFmpeg-mpeg2`	mencoder yuv/576p25_blue_sky.yuv -demuxer rawvideo -rawvideo \ fps=25:w=720:h=576 -o out/576p25_blue_sky_mpeg2.avi -ofps 25 \ -ovc lavc -lavcopts vcodec=mpeg2video:vqscale=5:vmax_b_frames=2:subq=8:psnr
MPEG-4 decoder	`Xvid`	mplayer mpeg4/576p25_blue_sky.avi -vc xvid -nosound -vo null -benchmark
MPEG-4 encoder	`Xvid`	mencoder yuv/576p25_blue_sky.yuv -demuxer rawvideo -rawvideo \ fps=25:format=i420:w=176:h=144 -o out/576p25_blue_sky_mpeg4.avi \ -ofps 25 -ovc xvid -xvidencopts fixed_quant=5:max_bframes=2:qpel:psnr
H.264 decoder	`FFmpeg-h264`	mplayer h264/576p25_blue_sky.h264 -vc ffh264 -nosound -vo null -benchmark
H.264 encoder	`x264`	x264 –bframes 2 –no-b-adapt –b-bias=0 –ref 16 –qp=26 –analyse all \ –weightb –me hex –merange 24 –subme 7 –8x8dct -fps 25 –frames 101 \ –progress -o out/576p25_blue_sky.h264 yuv/576p25_blue_sky.yuv 720x576

TABLE IV
SUMMARY OF HD-VIDEOBENCH EXECUTION COMMANDS

Resolution	Input	MPEG-2		MPEG-4		H.264	
		PSNR	bitrate	PSNR	bitrate	PSNR	bitrate
576p25	blue_sky	39.82	3504	38.69	1146	39.248	1095
	pedestrian_area	41.28	2724	40.76	1715	41.141	1382
	riverbed	38.95	10688	39.27	9435	38.456	7783
	rush_hour	42.49	2085	41.41	1217	41.965	1092
720p25	blue_sky	40.97	5541	39.84	2154	40.198	1887
	pedestrian_area	41.89	4783	41.47	3093	41.700	2249
	riverbed	39.70	19729	40.15	17108	39.391	13716
	rush_hour	43.09	3647	42.16	2290	42.649	1872
1088p25	blue_sky	41.81	9462	40.71	4265	40.947	3490
	pedestrian_area	41.93	9360	41.69	6219	41.661	3961
	riverbed	40.07	36475	40.65	31063	39.933	24131
	rush_hour	42.73	7086	42.17	4722	42.496	3357

TABLE V
HD-VIDEOBENCH RATE DISTORTION PERFORMANCE COMPARISON

1088p25 case the frame rate is 1.25 and 0.66 fps respectively. For H.264, a 38X speed-up is required to achieve real time operation.

For a more detailed performance evaluation of the `HD-VideoBench` applications with a special emphasis in H.264 see Alvarez et al [20].

VII. CONCLUSIONS

We have presented `HD-VideoBench`, a benchmark devoted to video coding applications and specialized for High Definition. After a careful examination of existing benchmarks for multimedia applications, we have found that none of them have all the required characteristics for a complete benchmark for HD video coding. Some of them do not include recent video Codecs, others have only low resolution inputs, others include reference versions of the Codecs that do not have been optimized for high performance, and others are not publicly available restricting the use and examination of the source code, input videos and coding options.

In `HD-VideoBench`, we solve these problems by providing a key selection of applications for video processing. `HD-VideoBench` includes Codecs for MPEG-2, MPEG-4 and H.264 standards based on open source implementations that have been extensively optimized for high performance. These applications are part of real life programs used in desktop operating systems for coding, transcoding and playing multimedia content. By using this kind of applications, we are ensuring the representativeness of the benchmark and, at the same time, by selecting open source implementations, we allow the researchers to have full access to the source code in order to perform analysis and further optimizations. Additionally, we have carefully selected a set of input sequences at HD resolution with different motion and spatial details. We have also analyzed and provided the coding options that are best suited for HD applications. As a result, `HD-VideoBench` has all the required characteristics for detailed benchmarking of HD digital video applications.

Initial performance results show that the processing of High Definition digital video with emerging video Codecs require an important increase in the performance provided by the architectures, more than it can be provided with SIMD optimizations. Enhancing the performance of these applications would require the efficient exploitation of multiple levels of parallelism like data, function and thread level parallelism.

Currently, we are working on extending `HD-VideoBench` by including parallel versions of the video Codecs for multiprocessor architectures, specially for emerging chip multiprocessing architectures. Additionally, we are also planning to include new video Codecs like VC-1, and Motion-JPEG-2000.

REFERENCES

[1] G. J. Sullivan and T. Wiegand, "Video Compression–From Concepts to the H.264/AVC Standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, Jan 2005.

[2] I. Richardson, *H.264 and MPEG-4. Video Compression for Next-generation Multimedia*. Chichester, England: Wiley, 2004.

[3] T. Sikora, "Trends and Perspectives in Image and Video Coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, Jan 2005.

(a) Decoding Performance Scalar Version

(b) Decoding Performance with SIMD Optimizations

(c) Encoding Performance Scalar Version
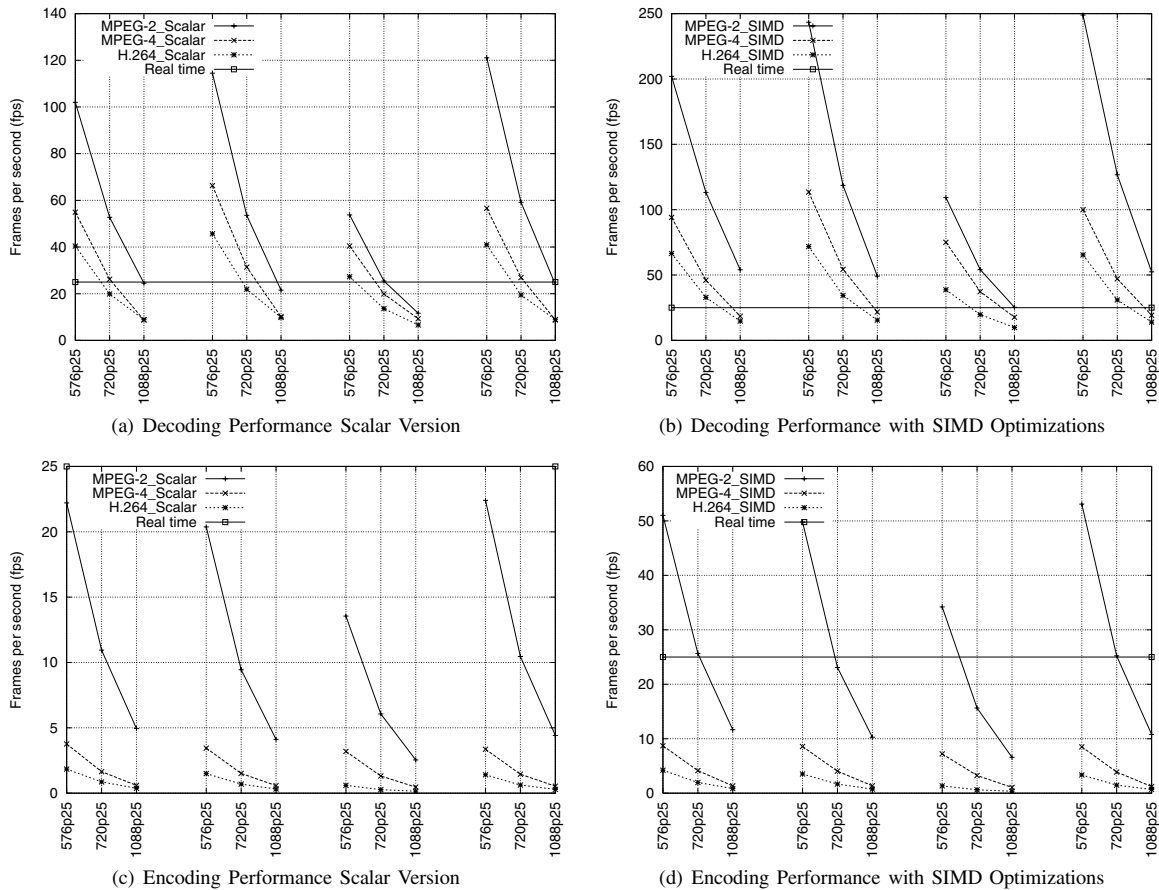
(d) Encoding Performance with SIMD Optimizations

Fig. 1.   HD-VideoBench Performance

[4] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, Jan 2004.

[5] M. Horowitz, A. Joch, and F. Kossentini, "H.264/AVC Baseline Profile Decoder Complexity Analyis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 704–716, July 2003.

[6] V. Lappalainen, A. Hallapuro, and T. D. Hamalainen, "Complexity of Optimized H.26L Video Decoder Implementation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 717–725, July 2003.

[7] M. Gschwind, H. P. Hofstee, B. Flachs, M. Hopkins, Y. Watanabe, and T. Yamazaki, "Synergistic processing in cell's multicore architecture," *IEEE Micro*, vol. 26, no. 2, pp. 10–24, 2006.

[8] J.-W. van de Waerdt, S. Vassiliadis, S. Das, S. Mirolo, C. Yen, B. Zhong, C. Basto, J.-P. van Itegem, D. Amirtharaj, K. Kalra, P. Rodriguez, and H. van Antwerpen, "The TM3270 Media-Processor," in *MICRO 38: Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture*, Nov 2005, pp. 331–342.

[9] C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communicatons Systems," in *30th International Symposium on Microarchitecture*, 1997, pp. 330–335.

[10] Nathan T. Slingerland and Alan Jay Smith, "Design and Characterization of the Berkeley Multimedia Workload," *Multimedia Systems*, vol. 8, no. 4, pp. 315–327, 2002.

[11] Markus Levy, "Evaluating Digital Entertainment System Performance," *IEEE Computer*, vol. 38, no. 7, pp. 68–72, 2005.

[12] J. Fritts, W. Wolf, and B. Liu, "Understanding Multimedia Application

Characteristics for Designing Programmable Media Processors," in *SPIE Photonics West, Media Processors '99*, 1999, pp. 2–13.

[13] J. E. Fritts, F. W. Steiling, and J. A. Tucek, "MediaBench II Video: Expediting the Next Generation of Video Systems Research," in *Proceedings of SPIE. Embedded Processors for Multimedia and Communications II*, 2005, pp. 79–93.

[14] "BDTI H.264 Solution Certification Benchmark," 2006, http://www.bdti.com.

[15] "MSSG: MPEG Software Simulation Group," 1994, http://www.mpeg.org/MPEG/MSSG/.

[16] "FFmpeg Multimedia System." 2005, http://ffmpeg.mplayerhq.hu/.

[17] "Libmpeg2. A Free MPEG-2 Video Stream Decoder," 2005, http://libmpeg2.sourceforge.net/.

[18] "XviD. An ISO MPEG-4 Compliant Video Codec," 2005, http://www.xvid.org.

[19] "H.264/AVC Software Coordination," 2005, http://iphome.hhi.de/suehring/tml/.

[20] M. Alvarez, E. Salami, A. Ramirez, and M. Valero, "A Performance Characterization of High Definition Digital Video Decoding Using H.264/AVC," in *IEEE International Symposium on Workload Characterization*, Oct 2005, pp. 24–33.

[21] "X264. A Free H.264/AVC Encoder," 2006, http://developers.videolan.org/x264.html.

[22] "MPEG-Test Sequences," 2005, http://www.ldv.ei.tum.de/liquid.php?page=70.

[23] A. M. Tourapis, "Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation," in *Proceedings of SPIE Visual Communications and Image Processing 2002*, Jan. 2002, pp. 1069–1079.

[24] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 5, pp. 349–355, May 2002.