A Performance Evaluation of High Definition Digital Video Decoding Using the H.264/AVC Standard

Mauricio Alvarez[*,1], Esther Salamí[*],
Alex Ramírez[*], Mateo Valero[*]

[*] *DAC, Univesitat Politècnica de Catalunya, Campus Nord, 08034 Barcelona, España*

ABSTRACT

H.264/AVC is a new international video coding standard that provides higher coding efficiency with respect to previous standards at the expense of a higher computational complexity. The computational complexity is even higher when H.264/AVC is used in applications with high bandwidth and high quality like high definition (HD) video. In this paper, we analyze the computational requirements of H.264 decoder with a special emphasis in HD video and we compare it with previous standards and lower resolutions. The analysis was done with both the reference and a SIMD optimized decoder. The main objective is to identify the application bottlenecks and suggest the necessary support in the architecture for processing HD video efficiently.

KEYWORDS: Multimedia; H.264/AVC; SIMD extensions; Altivec

1 Introduction

In recent years multimedia applications have become one of the most important workload of microprocessors, between multimedia applications digital video processing is one of the most computing demanding tasks due to its high bandwidth and its real time requirements [DD97]. Also in the last years the compression efficiency and quality of video systems have been improving as can be seen with the evolution of MPEG and H.26X video standards. The recent step of this evolution is the H.264/AVC, a new international video coding standard that provides higher coding efficiency with respect to previous standards [WSBA03]. The higher compressing ratio and quality of H.264 comes from the inclusion of advanced compression techniques that in turn demand more computational power. In applications

[1]E-mail: {alvarez,esalami,aramirez,mateo}@ac.upc.edu

with high bandwidth and high quality, like high definition video, the computational requirements are even bigger.

The decoding of high definition video using H.264/AVC is a big challenge for current processor architectures, in special in embedded systems in which the processor not only need to provide the required performance for real time operation but also maintain a low power consumption.

In this document we analyze the computational requirements of H.264 decoder using for HD video and compare the results with other video standards and with lower resolutions. With such a performance characterization we are trying to find the necessary architecture support for decoding H.264 HD sequences.

2 Overview of H.264/AVC

H.264/AVC is based on the same block-based motion compensation and transform-based coding framework of prior video coding standards like MPEG-2 and MPEG-4. It provides higher coding efficiency through added features and functionality that entail additional complexity. Some of these new features are an enhanced motion-prediction capability with variable size blocks; the use of a small block-size integer transform; the inclusion of an adaptive in-loop de-blocking filter, and the use of new enhanced entropy coding methods. H.264/AVC has been defined to allow its use in different contexts and applications: it can be implemented in applications like video conferencing, Internet streaming and high definition DVD [WSBA03].

3 Methodology and Tools

The study has been made on a 1.6 GHz PowerPC970 machine using the performance monitoring counters and the Apple CHUD tools. The processor has a 64KB L1 instruction cache, a 32KB data cache and a 512KB combined L2 cache. We have used the H.264/AVC, MPEG-4 and MPEG-2 reference decoders, and as input to them we have selected various test sequences with different motion characteristics. All the applications have been compiled with gcc-3.3.3 under Mac OS-X.

4 Profiling

The profiling of the JM-9.5 reference decoder is showed in Figure 1 for the 4 input sequences (blue_sky, pedestrian, riverbed ans rush_hour) and three different resolutions (720×576, 1280×720, 1920×1088). The application was subdivided into several functional sections: Luma Interpolation (LI), Intra-prediction (IP), Chroma interpolation (CI), Inverse transform and quantization (IT), De-blocking filter (DB), Entropy decoding and bit-stream processing (ED), Spatial Compensation (SPC), Picture Buffer management (PB) and miscellaneous operations (OT). The analysis shows that decoder spent most of the time in the interpolation process (both Luma and Chroma 47%) and the de-blocking filter (44.79%). Other significant funtions are the entropy decoding (11%) and the inverse transform (5.74%).

The interpolation process and the de-blocking-filter are the most time consuming kernels. Interpolation is used for generating sub-sample positions by means of FIR filters and/or

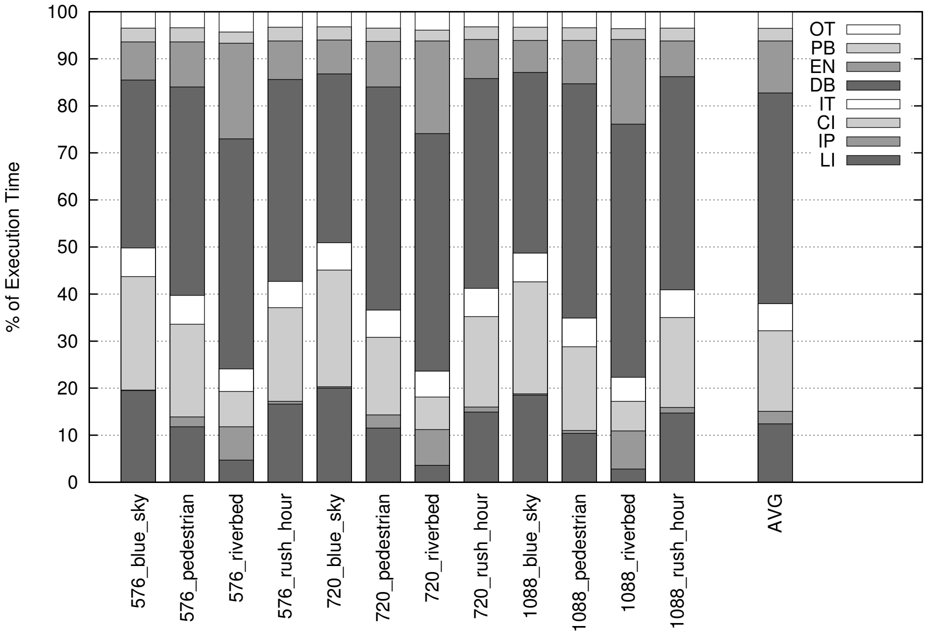

Figure 1: Profiling of H.264 Reference Decoder

Sequence	H.264-opt	MPEG-4	MPEG-2
	fps	fps	fps
576_blue_sky	3.43	58.21	237.53
576_pedestrian	3.41	78.55	299.40
576_riverbed	2.36	53.88	158.98
576_rush_hour	2.84	66.80	338.98
720_blue_sky	1.38	31.28	161.29
720_pedestrian	1.37	37.98	167.50
720_riverbed	1.04	27.09	80.84
720_rush_hour	0.86	31.72	179.53
1088_blue_sky	0.62	13.98	72.10
1088_pedestrian	0.61	15.52	67.34
1088_riverbed	0.50	12.73	37.01
1088_rush_hour	0.60	13.91	71.28

Table 1: Decoding performance in frames per second.

linear interpolation. It has a regular data pattern amenable for SIMD implementation. The de-blocking filter is a 4- or 5-tap FIR and need to perform a lot of decisions in the edges of 4×4 blocks that reduces the data level parallelism and make it difficult for SIMD optimization.

In order to improve the performance of the decoder we have implemented some kernels using the Altivec media extension to the PowerPC. In Figure 2 the profiling of the JM-9.5 reference decoder with Altivec optimizations is showed. With the Altivec optimization the luma interpolation kernel has passed from 12.43% to 3.58% of the total execution time. After that the execution time is dominated by the chroma interpolation (19.98%), the de-blocking filter (49.01%) and the entropy decoding (12.08%). In the first one there is potential for SIMD optimization, but the two later ones have a very limited data level parallelism so it is not possible to obtain a huge speed-up by aplying SIMD optimization.

Using the Altivec optimized version of the decoder we have measured the decoder performance in frames per second. Table 1 shows the average frames per second of H.264 decoder and compares it with MPEG-2 and MPEG-4 decoders. For all the resolutions the

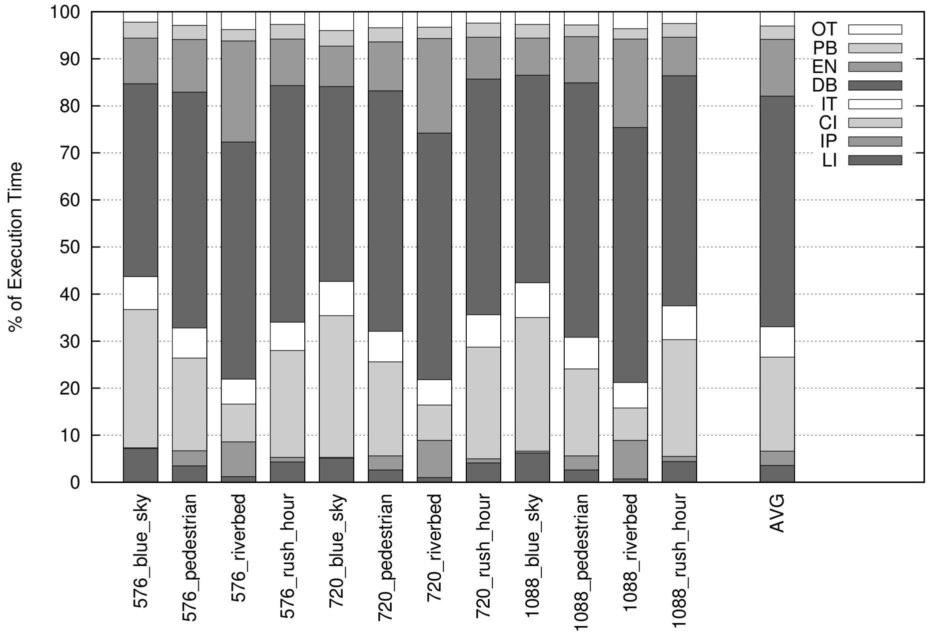

Figure 2: Profiling of H.264 Decoder with Altivec Optimizations.

MPEG-2 decoder can process the sequences in real time (25 frames per second). For MPEG-4 it is not possible to go at 25 fps for the 1920×1088 resolution. But for H.264/AVC the performance is far from real time in all the resolutions.

5 Conclusions

The decoding of High Definition video using the new standard H.264/AVC require a considerable amount of computational resources. Although HD video decoding is considered a data parallel application the new characteristics of the standard, mainly the reduced size of data blocks and the de-blocking filter, reduce the parallelism exploitable with current SIMD instructions. In order to decode HD video using H.264/AVC in real time current SIMD extensions need to be adapted for the new kernels and probably a multiprocessor system need to be adopted.

References

[DD97] K. Diefendorff and P.K. Dubey. How Multimedia Workloads Will Change Processor Design. *IEEE Micro*, 30(9):43–45, Sept 1997.

[WSBA03] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A.Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.