

Performance and Complexity Co-evaluation of the Advanced Video Coding Standard for Cost-Effective Multimedia Communications

Sergio Saponara

*Interuniversity Microelectronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium
Email: s.saponara@iet.unipi.it*

Kristof Denolf

*Interuniversity Microelectronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium
Email: denolf@imec.be*

Gauthier Lafruit

*Interuniversity Microelectronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium
Email: lafruit@imec.be*

Carolina Blanch

*Interuniversity Microelectronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium
Email: blanch@imec.be*

Jan Bormans

*Interuniversity Microelectronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium
Email: bormans@imec.be*

Received 22 November 2002; Revised 26 September 2003

The advanced video codec (AVC) standard, recently defined by a joint video team (JVT) of ITU-T and ISO/IEC, is introduced in this paper together with its performance and complexity co-evaluation. While the basic framework is similar to the motion-compensated hybrid scheme of previous video coding standards, additional tools improve the compression efficiency at the expense of an increased implementation cost. As a first step to bridge the gap between the algorithmic design of a complex multimedia system and its cost-effective realization, a high-level co-evaluation approach is proposed and applied to a real-life AVC design. An exhaustive analysis of the codec compression efficiency versus complexity (memory and computational costs) project space is carried out at the early algorithmic design phase. If all new coding features are used, the improved AVC compression efficiency (up to 50% compared to current video coding technology) comes with a complexity increase of a factor 2 for the decoder and larger than one order of magnitude for the encoder. This represents a challenge for resource-constrained multimedia systems such as wireless devices or high-volume consumer electronics. The analysis also highlights important properties of the AVC framework allowing for complexity reduction at the high system level: when combining the new coding features, the implementation complexity accumulates, while the global compression efficiency becomes saturated. Thus, a proper use of the AVC tools maintains the same performance as the most complex configuration while considerably reducing complexity. The reported results provide inputs to assist the profile definition in the standard, highlight the AVC bottlenecks, and select optimal trade-offs between algorithmic performance and complexity.

Keywords and phrases: video compression standards (MPEG AVC, H.26L, H.264), power-optimized multimedia coding, cost-effective design of multimedia systems, complexity analysis.

1. INTRODUCTION

New applications and services in the communication and computing technology mainly focus on the processing and transmission of multimedia contents with portable and per-

sonal access to the information. While the enabling technologies for speech, data, text, and audio are available today (allowing the widespread diffusion of mobile phones, MP3 music players, global positioning systems to name but a few), the management of video information represents a remaining

design challenge for its inherent high data rates and storage burdens. To cope with this issue, the advanced video codec (AVC), recently defined in a standardization effort of the ITU-T and ISO/IEC joint video team (JVT) [1, 2, 3, 4], promises both enhanced compression efficiency over existing video coding standards (H.263 [5], MPEG-4 Part 2 [6, 7]) and network friendly video streaming. The codec aims at both conversational (bidirectional and real-time videotelephony, videoconferencing) and nonconversational (storage, broadcasting, streaming) applications for a wide range of bit-rates over wireless and wired transmission networks.

Like previous video coding standards [5, 6, 7], AVC is based on a hybrid block-based motion compensation and transform-coding model. Additional features improve the compression efficiency and the error robustness at the expense of an increased implementation complexity. This directly affects the possibility for cost-effective development of AVC-based multimedia systems and hence the final success of the standard. The scope of this paper is the exploration of the compression efficiency versus implementation cost design space to provide early feedbacks on the AVC bottlenecks, select the optimal use of the coding features, and assist the definition of profiles in the standard. The complexity analysis focuses on the data transfer and storage, as these are the dominant cost factors in multimedia system design for both software- and hardware-based architectures [8, 9, 10, 11, 12, 13, 14, 15, 16]. Memory metrics are completed by computational burden measures. A comparison of the new codec with respect to current video coding technology, in terms of both compression efficiency and implementation cost, is also provided.

The paper is organized as follows. After a review of known profiling methodologies for multimedia system design, Section 2 defines and motivates the analysis approach adopted throughout the paper. A description of the upcoming standard including both encoder and decoder architectures is addressed in Section 3. Section 4 describes the testbench environment. Section 5 presents the global results obtained for the codec in terms of compression efficiency, memory cost, and computational burden. Section 6 exploits a multiobjective analysis to select the optimal trade-off between algorithmic performance and implementation cost. Section 7 deals with the definition of profiles in the standard. Conclusions are drawn in Section 8.

2. PERFORMANCE AND COMPLEXITY EVALUATION METHODOLOGY

As sketched in Figure 1, the design flow of complex multimedia systems such as video codecs typically features two main steps: an algorithmic development phase followed by a system implementation process. The first step focuses on algorithmic performance (peak signal-to-noise ratio (PSNR), visual appearance, and bit rate). The algorithmic specification is typically released as a paper description plus a software verification model (often in C). Usually, the software model is not optimized for a cost-effective realization since its scope is mainly a functional algorithmic verification and the tar-

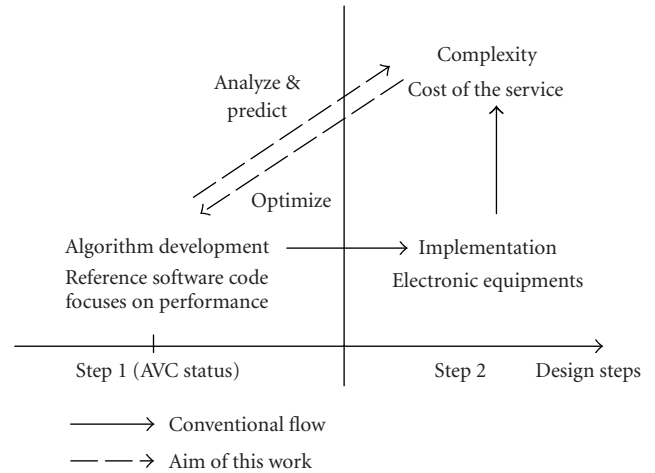


FIGURE 1: Algorithmic performance and complexity co-evaluation for cost-effective system design.

get platform is unknown. Moreover, in the case of multimedia standards such as ITU-T and ISO/IEC video codecs, the verification software models (up to 100.000 C-code lines [10]) are written in different code styles since they are the results of the combined effort of multiple teams. The second step of the design flow deals with the actual system realization starting from the paper and software specification. Only at this late stage, the true implementation complexity of the algorithm is known, which will determine the cost of the user's terminal and hence its success and widespread diffusion or not. If the initial cost specifications are not reached, the gained complexity information is used to re-enter the design flow making new actions at algorithmic and then implementation levels. This time-consuming loop ends only when the complexity meets the user's requirements.

To bridge the gap between the algorithmic development of a new multimedia application and its cost-effective realization, we propose to explore the performance versus implementation cost design space at the early algorithmic design phase. The goal of this co-evaluation approach is twofold: (i) to assess the performance and implementation cost of a new multimedia system presenting also a comparison with current technology ("Analyze & Predict" arrow in Figure 1); (ii) to provide feedback on the realization bottlenecks and highlight the properties of the system allowing for complexity reduction at the early algorithmic design phase ("Optimize" arrow in Figure 1). This way, the time-consuming iterations of the conventional design flow can be avoided. Particularly, this paper focuses on the design of the AVC video coding standard for which a committee draft specification and a verification software C-model have been recently defined [1, 2]. The huge C-code complexity of multimedia systems makes an implementation cost analysis without additional help time consuming and error prone. Hence, a supporting framework for automated analysis of the executable software specification is essential

to apply the co-evaluation approach to a complex real-life design such as AVC. To this aim, the C-in-C-out ATOMIUM/Analysis environment [10, 15, 17] has been developed. It consists of a set of kernels providing functionality for data transfer and storage analysis and pruning. Using ATOMIUM involves three steps [10]: instrumenting the program, generating complexity data by executing the instrumented code with representative test stimuli, and postprocessing of this data.

High-level profiling analyses have been addressed in the past for previous ITU-T (H.263+ in [5]) and ISO/IEC (MPEG-4 Part 2 in [6, 18], MPEG-1/-2 decoder in [19]) video codecs. However, the above approaches focus mainly on computational complexity (processing time [5] or instruction-level [6, 19] profiling on a specific platform: typically general purpose CISC processors, e.g., Pentium in [5], or RISC processors, e.g., UltraSPARC in [6, 19]), while the actual implementation of H.263 and MPEG-4 codecs clearly demonstrates that multimedia applications are data dominated. As a consequence, data transfer and storage have a dominant impact on the cost-effective realization of multimedia systems for both hardware- and software-based platforms [8, 9, 10, 11, 12, 13, 14, 15, 16]. Application specific hardware implementations have the freedom to match the memory and communication architectures to the application. Thus, an efficient design flow exploits this to reduce area and power [8, 11, 12]. On the other hand, programmable processors rely on the memory hierarchy and on the communication bus architecture that come with them. Efficient use of these resources is crucial to obtain the required speeds as the performance gap between CPU and DRAM is growing every year [9, 13, 14, 15, 20]. This high-level analysis is also essential for an efficient hardware/software system partitioning. In [18], a complexity evaluation methodology based on the extraction of execution frequencies of core tasks is proposed. Combining this data with complexity figures for the core tasks on a specific platform, a performance estimate of the whole system on that platform is obtained. This approach relies on implementation cost measures already available for the single tasks (provided as benchmarks of a specific platform). Therefore, it is not suitable to analyze systems, such as AVC, featuring new algorithms for which complexity results are not available.

In this paper, the coding performance analysis is reported in terms of PSNR and bit rate, while the complexity metrics are the memory access frequency (total number of data transfers from/to memory per second) and the peak memory usage (maximum memory amount allocated by the source code) as counted within the ATOMIUM environment. These figures give a platform independent measure of the memory cost (storage and communication of data) and are completed with the processing time as a measure of the computational burden (processing time figures are measured on a Pentium IV at 1.7 GHz with Windows 2000). The software models used as input for this paper are the AVC JM2.1 [2] and the MPEG-4 Part 2 [7] (simple profile in [21]), both nonoptimized source codes.

3. ADVANCED VIDEO CODEC

3.1. Standard overview

An important concept of AVC is the separation of the system into two layers: a video coding layer (VCL), providing the high-compressed representation of data, and a network adaptation layer (NAL), packaging the coded data in an appropriate manner based on the characteristics of the transmission network. This study focuses on the VCL. For a description of NAL features, the reader is referred to [22, 23]. Figures 2 and 3 show the block diagram of the AVC decoder and encoder, respectively. In analogy with previous coding standards, the AVC final committee draft [1] does not explicitly define the architecture of the codec but rather it defines the syntax of an encoded video bitstream together with the decoding method. In practice, according to the structure of the AVC reference software [2], a compliant encoder and decoder are likely to include the functional tasks sketched in Figures 2 and 3. Nevertheless, particularly at the encoder side, there is space for variations in the sketched architecture to meet the requirements of the target application with the desired trade-off between algorithmic performance and cost. At the decoder side, the final architecture depends on the encoder profiles (i.e., combination of coding tools and syntax of the relevant bitstream) supported for decoding.

The framework defined in Figures 2 and 3 is similar to the one of previous standards: translational block-based motion estimation and compensation, residual coding in a transformed domain, and entropy coding of quantized transform coefficients. Basically, rectangular pictures can be coded in intra (I), inter (P), or bidirectional (B) modes. Both progressive and interlaced 4 : 2 : 0 YUV sequences are supported. Additional tools improve the compression efficiency, albeit at an increased implementation cost. The motion estimation and compensation schemes (ME and MC in Figures 2 and 3) support multiple previous reference pictures (up to 5) and a large number of different block sizes (from 16×16 up to 7 modes including 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , and 4×4 pixel blocks). The motion vector field can be specified with a higher spatial accuracy, quarter- or eighth-pixel¹ resolution instead of half pixel. Pixel interpolation is based on a finite impulse response (FIR) filtering operation: 6 taps for the quarter resolution and 8 taps for the eighth one. A rate-distortion (RD) Lagrangian technique [24] optimizes both motion estimation and coding mode decisions. Since the residual coding is in a transformed domain, a Hadamard transform can be used to improve the performances of conventional error cost functions such as the sum of absolute differences (SAD). Moreover, a deblocking filter within the motion compensation loop aims at improving prediction and reducing visual artifacts. AVC adopts spatial prediction for intracoding, being the pixels

¹The eighth-pixel resolution, available in AVC JM 2.1 [1, 2], is no longer supported in the last release [3]. However, this tool is kept during the proposed analysis since it is useful for high-rate video applications (Section 7).

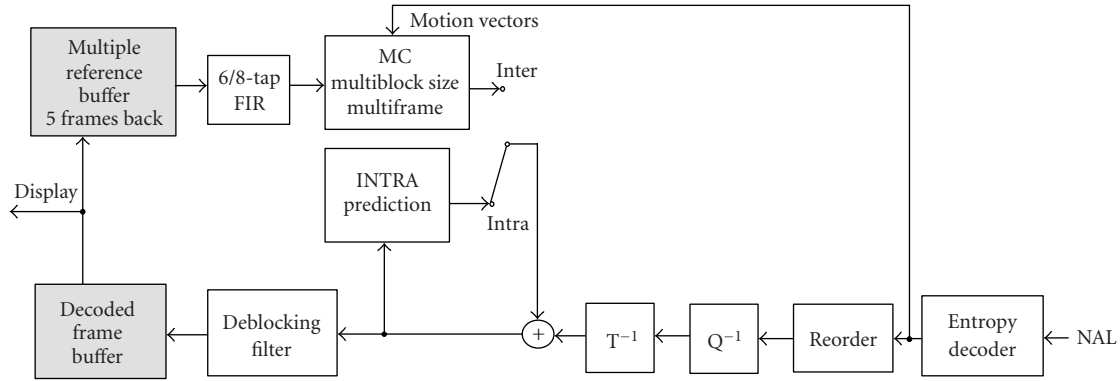


FIGURE 2: AVC decoder block diagram.

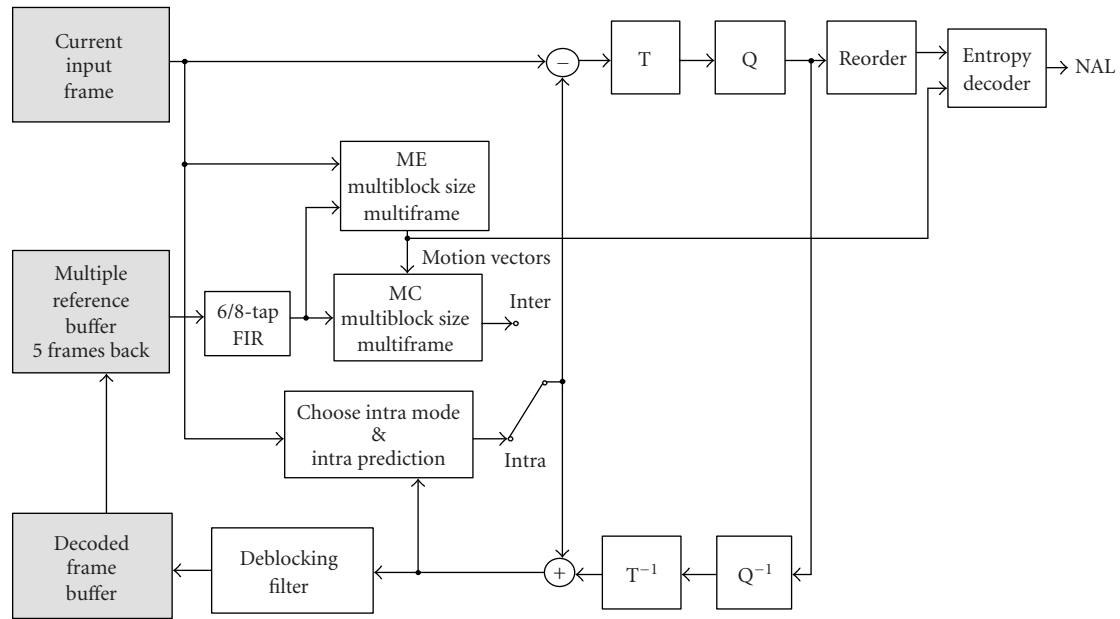


FIGURE 3: AVC encoder block diagram.

predicted from the neighbouring samples of already coded blocks. To this aim, the standard provides a DC plus 8 directional modes involving linear combinations of the samples. The conventional 8×8 floating-point discrete cosine transform, specified with rounding error margins, is replaced by a purely integer spatial transform (T and T^{-1} in Figures 2 and 3), basically working on 4×4 shapes. The small sizes help to reduce blocking artifacts while the integer specification prevents any mismatch between the encoder and the decoder. Finally, two methods are specified for entropy coding: a universal variable-length coder (UVLC) that uses a single reversible VLC table for all syntax elements and a more sophisticated context adaptive binary arithmetic coder (CABAC) [25].

3.2. Related work

Several contributions have recently been proposed to assess the coding efficiency of the AVC/H.26L scheme [3, 22, 25, 26, 27, 28] (H.26L is the original ITU-T project used as a starting point for the AVC standard, released as ITU-T H.264 and ISO/IEC MPEG-4 Part 10). Although this analysis covers all tools, the new features are typically tested independently comparing the performance of a basic configuration to the same configuration plus the tool under evaluation. In this way, the intertool dependencies and their impact on the trade-off between coding gain and complexity are not fully explored yet. Indeed, the achievable coding gain is greater for basic configurations, where the other tools are off and video data still feature a high correlation. For codec configurations

TABLE 1: Characteristics of the test sequences with their targeted bit rate.

Test case	Video sequence	Format (pixels)	Frame rate (Hz)	QP	Bit rate (Kbps)
MD	Mother & Daughter	QCIF (176×144)	30	18	40
FOR1	Foreman	QCIF (176×144)	25	17	150
FOR2	Foreman	CIF (352×288)	25	17	450
CM	Calendar & Mobile	CIF (352×288)	15	12	2000

in which a certain number of tools are already on, the residual correlation is lower and the further achievable gain is less noticeable.

Complexity assessment contributions have been proposed in [26, 27, 29, 30]. However, these works do not exhaustively address the problem since just one side of the codec is considered (the encoder in [26, 27] and the decoder in [29, 30]) and/or the analysis of the complete tool-set provided by the upcoming standard is not presented. Typically, the use of B-frames, CABAC, multireference frames, and eighth resolution is not considered. Consequently, the focus is mostly on a baseline implementation suitable for low-complexity and low-bit-rate applications (e.g., video conversation), while AVC aims at both conversational and nonconversational applications in which these discarded tools play an important role [3, 25, 28]. Furthermore, the complexity evaluation is mainly based on computational cost figures, while data transfer and storage exploration proved to be mandatory for efficient implementation of video systems [8, 9, 10, 11, 12, 13, 14, 15, 16] (see Section 2). Access frequency figures are reported in [29] for a H.26L decoder, but the analysis focuses on the communication between an ARM 9 CPU and the RAM, being a platform-dependent measure of the bus bandwidth rather than a platform-independent exploration of the system.

4. TEST ENVIRONMENT

4.1. Test sequences

The proposed testbench consists of 4 sequences with different grades of dynamism, formats, and target bit rates. Their characteristics are sketched in Table 1. Mother & Daughter 30 Hz QCIF (MD) is a typical head and shoulder sequence occurring in very low-bit-rate applications (tens of Kbps). Foreman 25 Hz QCIF (FOR1) has a medium complexity, being a good test for low bit rate applications ranging from tens to few hundreds of Kbps. The CIF version of Foreman (FOR2) is a useful test case for middle-rate applications. Finally, Calendar & Mobile 15 Hz CIF (CM) is a high-complexity sequence with lot of movements including rotation and is a good test for high-rate applications (thousands of Kbps). Since the current standard description does not provide online rate controls, the test sequences in Section 5.1 are coded with a fixed quantization parameter (QP in Table 2) to achieve the target bit rate. The dependency of the proposed analysis on the QP value is addressed in Section 5.2.

4.2. Test cases

The paper reports for each test video 18 different AVC configurations whose descriptions are shown in Table 2.

For each test case (identified by a number from 0 to 17), Table 2 details the activation status of the optional video tools with respect to a basic AVC configuration (case 0) characterized by a search range of 8, 1 reference frame, quarter-pixel resolution, intracoding by 9 prediction modes, in-loop deblocking, UVLC entropy coder, and a first I picture followed by all P pictures. The tools which are changing between two successive test cases are highlighted in bold style in Tables 2 and 3. Comparisons with MPEG-4 Part 2 [7], simple profile in [21] with a 16 search size, half-pixel resolution, and I and P pictures (referred to as test case M4 in the next sections) are provided in Sections 5 and 6.

The 18 reported AVC configurations are selected, for sake of space, as representatives of more than 50 considered test cases. The first two cases represent a “simple” AVC implementation with all new video tools off (with search displacements of 8 for case 0 and of 16 for case 1). Then, in cases 2 to 9 (“accumulative video tool enabling” in Table 2), the new AVC features are added one by one up to “complex” configurations, with all tools on (including B pictures with search displacements of 16 for cases 10 and 12, and of 32 for case 11), reaching the best coding performance although at maximum complexity. Comparing the test cases from 3 to 12 with the basic configurations 0 and 1 gives feedbacks about the coding efficiency versus complexity trade-off of the new AVC video tools. As it will be explained further, cases 13 to 17 in Table 2 have been properly selected to achieve roughly the same coding efficiency as the complex cases, while considerably reducing the complexity overhead by discarding some tools and reducing the number of reference frames and the search area. The overall set of AVC configurations (roughly 50) is the same for all the considered test sequences. As it will be detailed in Sections 5 and 6, the performance and usefulness of the different video tools depend on the considered bit rate and hence on the considered sequence (MD for tens of Kbps, FOR1 and FOR2 from tens to hundreds of Kbps, and CM for thousands of Kbps). During the selection, among the set of 50 configurations, of the 18 more representative test cases to be reported in this paper, the configurations from 0 to 12 (“simple,” “accumulative video tool enabling,” and “complex”) have been chosen identical for all the video sequences, while the configurations from 13 to 17 (cost-efficient) feature some differences. Table 2 refers to FOR1 and FOR2, while Table 3 reports the cost-efficient configurations for MD and CM.

TABLE 2: AVC configurations cases—FOR1 and FOR2 test sequences (Y = tool on, N = tool off).

Case number	Simple		Accumulative video tool enabling								Complex			Cost-efficient				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Search range	8	16	16	16	16	16	16	16	16	16	16	32	16	16	8	8	8	8
Block sizes ^a	1	1	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Ref. frames	1	1	1	1	3	5	5	5	5	5	5	5	5	5	5	3	2	1
RD-Lagrange	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Hadamard	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N	N	N	N	N
Eighth resolution	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	N	N	N	N
CABAC	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y
B-frames	N	N	N	N	N	N	N	N	N	N	1	1	2	1	1	1	1	1

TABLE 3: AVC “cost-efficient” configurations cases—MD and CM test sequences (Y = tool on, N = tool off).

Case number	MD					CM				
	13	14	15	16	17	13	14	15	16	17
Search range	16	16	16	16	8	8	8	8	8	8
Block sizes ^a	7	7	7	7	7	7	7	7	7	7
Ref. frames	2	1	1	1	1	5	3	2	3	2
RD-Lagrange	Y	Y	Y	Y	Y	Y	Y	Y	N	N
Hadamard	Y	Y	Y	N	N	Y	N	N	N	N
Eighth resolution	Y	Y	N	N	N	Y	Y	Y	Y	Y
CABAC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
B-frames	N	N	N	N	N	1	1	1	1	1

^aIf block sizes = 1, only the 16×16 mode is on; if block sizes = 4, then 16×16 , 16×8 , 8×16 , and 8×8 modes are on; otherwise, all modes are on.

5. AVC PERFORMANCE AND COMPLEXITY ASSESSMENT

5.1. Codec analysis

An overview of the encoder and decoder results (PSNR-Y, bit rate, peak memory usage, memory access frequency, processing time) for the 18 AVC test cases and the M4 one is summarized in Figures 4, 5, 6, 7, 8, 9, 10, and 11 and Tables 4, 5, and 6.

Coding performance results

Figures 4, 5, 6, and 7 list the rate-distortion results for all the video inputs using the fixed QP values reported in Table 1. For the sake of clarity, a rhombus represents the simple AVC configurations (cases 0 and 1), a cross identifies test cases from 2 to 9, a square represents complex AVC configurations (cases 10 to 12), a triangle indicates the cost-efficient configurations (cases 13 to 17 in Tables 2 and 3), and a circle refers to M4 results.

Clearly, AVC is a new codec generation featuring an outstanding coding efficiency: if all the novel video tools are used, AVC leads to an average 40% bit saving plus a 1–2 dB PSNR gain compared to previous M4 video coding standard (see results for test cases 10, 11, and M4 in Figures 4, 5, 6, and 7).

Complexity results

Figures 8, 9, 10, and 11 deal with processing time and memory access frequency costs for both AVC encoder (Figures 8

and 9) and decoder (Figures 10 and 11). In these figures, for all video inputs, the reported values are normalized with respect to the ones of the relevant test cases 0. A close similarity between the processing time and the memory access frequency curves emerges from the comparison of Figures 8 and 9 at the encoder and Figures 10 and 11 at the decoder. Moreover, the analysis of the performance and complexity metrics shows that the new coding scheme acts likewise for all input sequences, particularly at middle and low bit rates (see the behaviors of MD, FOR1, and FOR2 in Figures 4, 5, 6, 8, 9, 10, and 11). Small differences arise for high rate video applications (CM) as emerges from Figures 10 and 11.

Absolute complexity values are reported in Table 4 listing the range achieved by the different AVC configurations (rows Min and Max) and the complexity results of M4 as a reference. The processing time values in Table 4 are expressed in a relative way: they refer to the time needed to encode/decode on a Pentium IV at 1.7 GHz, 1 second of the original test sequence, that is to say, (see Table 1) 25 frames of FOR1 and FOR2, 30 frames of MD, and 15 frames of CM. As a consequence, meting real-time constraints entails a relative processing time smaller than 1.

The encoder peak memory usage depends on the video format and linearly on the number of reference frames and the search size. The influence of the other coding tools and the input video characteristics is negligible. At the decoder side, the peak memory usage depends only on the video format and on the maximum number of reference frames to

TABLE 4: Codec complexity for MPEG-4 Part 2 (M4) and AVC codec (ranges Min, Max).

Test cases	Encoder			Decoder		
	Relative time	Accesses ($10^9/s$)	Peak memory (MB)	Relative time	Accesses ($10^6/s$)	Peak memory (MB)
MD	AVC Min	6.48	1.40	2.19	0.30	61.10
	AVC Max	409.12	79.92	15.60	0.69	104.41
	M4	4.15	1.32	2.97	0.49	102.79
FOR1	AVC Min	5.40	1.18	2.19	0.58	90.16
	AVC Max	330.87	65.78	15.60	1.30	153.88
	M4	3.55	1.15	2.97	0.61	107.51
FOR2	AVC Min	21.70	4.60	7.31	3.04	385.21
	AVC Max	1117.48	258.01	26.87	5.51	636.90
	M4	14.64	4.96	9.88	2.25	411.4
CM	AVC Min	12.98	2.75	7.31	2.33	287.10
	AVC Max	567.37	134.26	26.87	4.26	492.54
	M4	11.32	3.19	9.88	1.76	278.30

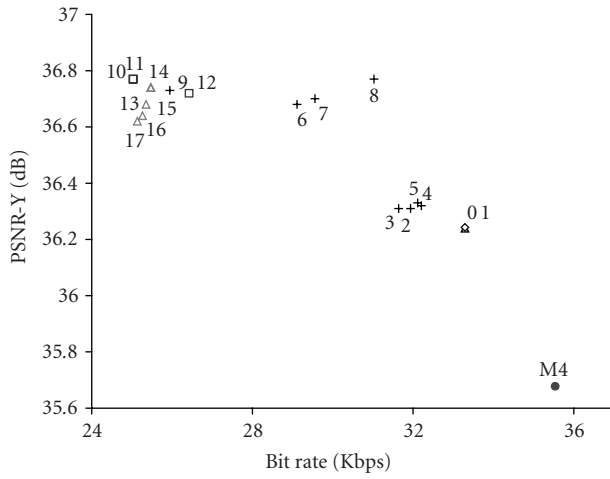


FIGURE 4: Rate-distortion results for the MD test cases.

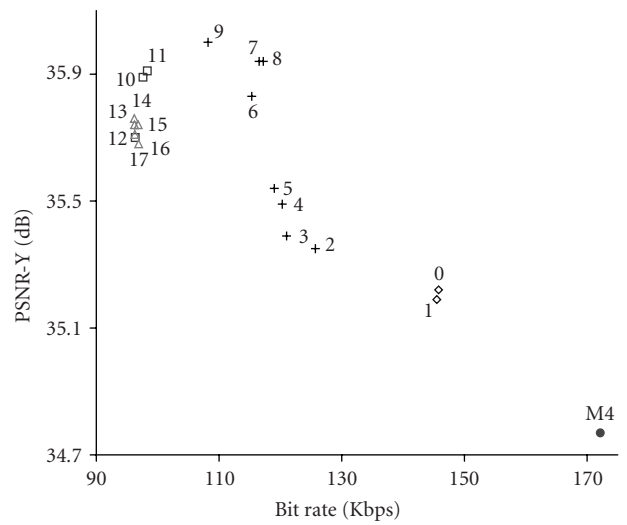


FIGURE 5: Rate-distortion results for the FOR1 test cases.

TABLE 5: Decoder memory usage (MB) versus video format and maximum number of decodable reference frames.

	1 frame	2 frames	3 frames	4 frames	5 frames
QCIF	1.06	1.14	1.22	1.30	1.38
CIF	2.91	3.22	3.53	3.84	4.15

decode. Peak memory usage dependencies for the decoder and the encoder are detailed in Tables 5 and 6.

To better highlight the intertools dependencies, the complexity results of Figures 8, 9, 10, and 11 and Tables 4, 5, and 6 refer to the whole AVC coder and decoder. A functional access and time distribution over the different components (e.g., motion estimator, intra predictor, etc.) have already been addressed by the authors in [31] for simple and complex configurations. At the encoder side, up to 90% of the complexity is due to motion estimation. The decoder's main bottlenecks are the motion compensation (up to 30% and

60% for simple and complex configurations, respectively) and the intrareconstruction (nearly 20% and 15% for simple and complex configurations, respectively). With respect to previous ITU-T and ISO/IEC standards, another important component of the AVC decoder is the in-loop deblocking filter (see further details in Section 5.3) whose implementation entails an overhead up to 6% for the access frequency and 10% for the processing time.

Analysis of coding performance and complexity results

AVC is a new codec generation featuring an outstanding coding efficiency, but its cost-effective realization is a big challenge. If all the novel coding features are used, AVC leads to an average 40% bit saving plus a 1–2 dB PSNR gain compared to previous video coding standards (see results for test cases 10, 11, and M4 in Figures 4, 5, 6, and 7). In this way, it represents the enabling technology for the widespread diffusion of

TABLE 6: Encoder memory usage (MB) versus reference frame number, search size, and video format.

Search size	QCIF			CIF		
	1 frame	3 frames	5 frames	1 frame	3 frames	5 frames
32	5.76	10.69	15.60	10.87	18.87	26.87
16	2.92	5.09	7.14	8.03	13.06	18.41
8	2.19	3.64	4.98	7.31	11.94	16.25

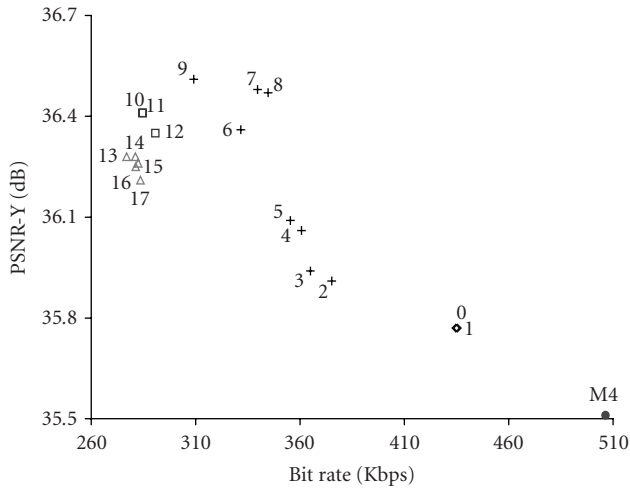


FIGURE 6: Rate-distortion results for the FOR2 test cases.

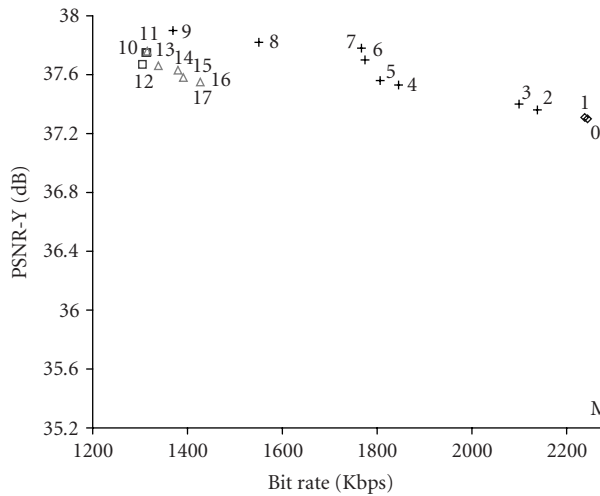


FIGURE 7: Rate-distortion results for the CM test cases.

multimedia communication over wired and wireless transmission networks such as xDSL, 3G mobile phones, and WLAN. However, these figures come with a memory and computational complexity increase of more than one order of magnitude at the encoder. The decoder's complexity increase amounts to a factor 1.5–2 (see results for test cases 0, 10, and 11 in Figures 8, 9, 10, and 11 and those for the AVC Max and M4 rows in Table 4). These increase factors are higher for the lower bit rate video as it emerges from the

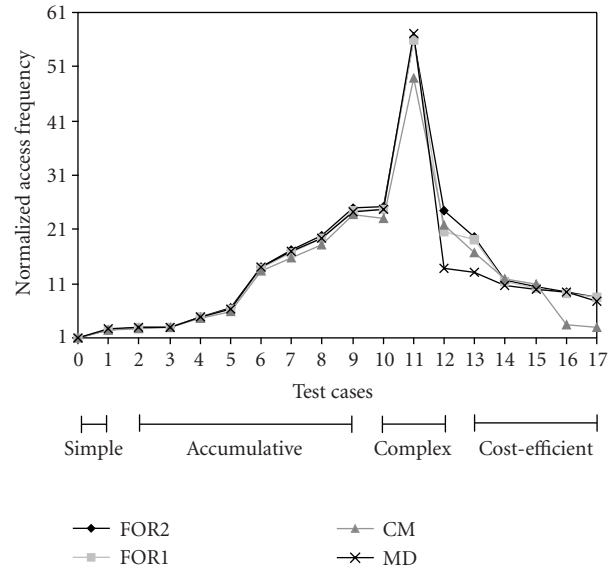


FIGURE 8: Normalized access frequency in case of AVC encoder.

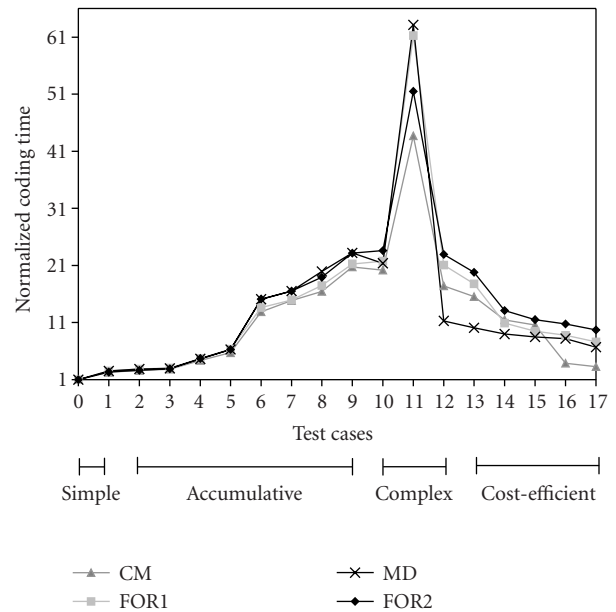


FIGURE 9: Normalized AVC coding time.

comparison of cases 0 and 11 in Figures 8, 9, 10, and 11 for the different video inputs. Case 11, for which the maximum

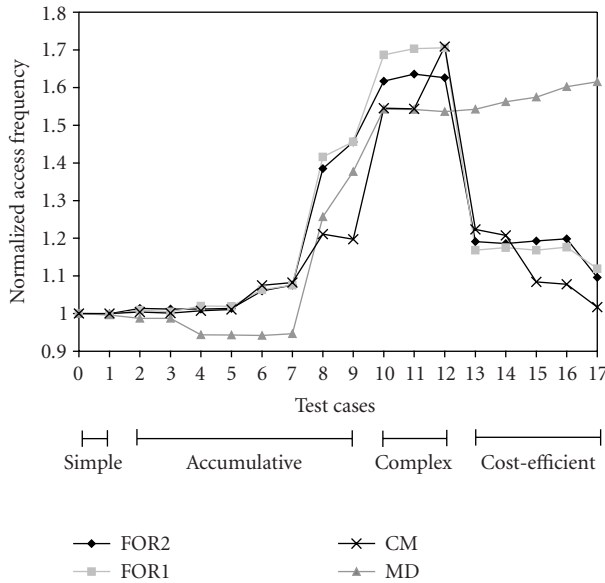


FIGURE 10: Normalized access frequency in case of AVC decoder.

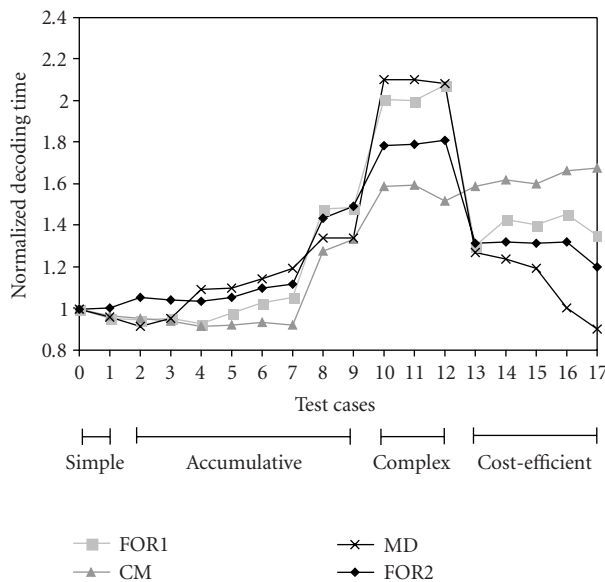


FIGURE 11: Normalized AVC decoding time.

complexity is measured, is the configuration used in [3] to show the AVC compression efficiency with respect to previous video coding standards. Finally, the complexity ratio between the encoder and the decoder further highlights the AVC bottleneck, particularly for conversational applications (e.g., videotelephony), where both the encoder and the decoder capabilities must be integrated in the user's terminal. For a simple profile, Min rows in Table 4, the encoder requires an access frequency and coding time at least 10 times that of the decoder and uses 2 times more memory space. For complex profiles, Max rows in Table 4, the encoder access frequency is two orders of magnitude larger than the decoder

one, while the peak memory usage is one order of magnitude higher.

The above measurements refer to nonoptimized source code and hence the future application of algorithmic and architectural design optimizations will lead to a decrease of the absolute complexity values, as it is the case in implementations of previous ITU-T and ISO/IEC standards [5, 11, 15, 16]. For instance, [32] recently proposed a fast motion estimation technique exploiting the new features of AVC, such as multireference frames and variable block sizes. The authors report a complexity reduction of a factor 5–6 with respect to a nonoptimized encoder realization based on the full search. However, the large complexity ratio between the reference codes of AVC and M4 (one order of magnitude at the encoder and a factor 2 at the decoder) presents a serious challenge requiring an exhaustive system exploration starting from the early standard design phase. Indeed, the performance growth rate predicted by Moore's law for the CPU amounts roughly to a factor 2 every 18 months. If we assume the same optimization factor as previously achieved for M4 to the current AVC code, but without any further system-level investigation, a cost-effective implementation could still not be scheduled before 2007 (i.e., the algorithmic complexity increase at the encoder would be covered in about four years and a half by the silicon technology improvements). Taking into account the lower performance growth rate of memories compared to CPU [20], the above time figure would even be worse.

The results in Figures 4, 5, 6, 7, 8, 9, 10, and 11 also provide useful hints for the selection of the optimal trade-off between coding efficiency and implementation complexity in order to maximize the utility for the final user. Indeed, the analysis of the above data clearly demonstrates a property of the AVC scheme: when combining the new coding features, the relevant implementation complexity accumulates (see the waveforms in Figures 8, 9, 10, and 11 for the test cases 0 to 11), while the global compression efficiency saturates (see the clusters in Figures 4, 5, 6, and 7 for the test cases 9 to 17). As a matter of fact, the achievable coding gain when enabling one of the new AVC features is greater for basic codec configurations, where the other tools are off and video data still feature a high correlation. For codec configurations in which a certain number of tools are already on, the residual data correlation is lower and hence, the further achievable gain is less noticeable, that is, the global compression efficiency saturates.

As a consequence, a "smart" selection of the new coding features can allow for roughly the same performances as a complex one (all tools on) but with a considerable complexity reduction. The coding efficiency (Figures 4, 5, 6, and 7) of test cases 13 to 17 is similar to that of cases 10 and 11, but their implementation cost (Figures 8, 9, 10, and 11) is closer to the basic cases 0 and 1. The achievable saving factor is at least 6.5 for the encoder. At the decoder side, the range of variation among simple and complex configurations is smaller, therefore, the saving is less noticeable than for the encoder. No complexity reduction is achieved for high rate video (CM), while saving factors of roughly 1.5 for both time

and memory metrics can be achieved for low bit rate videos. A single AVC configuration able to maximize coding efficiency, while minimizing memory and computational costs, does not exist. However, different configurations leading to several performance/cost trade-offs exist. To find these configurations, and hence to highlight the bottlenecks of AVC, a multiobjective optimization problem (solved, as it will be explained further, through a Pareto curve analysis) is addressed in Section 6 to explore the five-dimensional design space of PSNR, bit rate, computational burden, memory access frequency, and storage.

5.2. Performance and complexity analysis versus QP

Typically, video codecs incorporate a rate control scheme to target a given bit rate by adapting the quantization level. Since the standard description used in this paper does not yet provide such regulator, this section details the impact of different QP values on the analysis addressed in Section 5.1. All measurements described above are repeated on the 4 test sequences using several QP values (12, 16, 20, 24, 28) next to the fixed ones set in Table 1. To be noted that this analysis refers to the QP range defined in the JM2.1 implementation of the standard. Figure 12 sketches rate-distortion results: the points with higher PSNR and bit rate values are obtained with lower QP values. Figures 13, and 14 present the encoder and decoder complexity metrics expressed in terms of memory access frequency and processing time (expressed as relative time like in Section 5.1). In Figures 13 and 14, an arrow indicates the direction of growing QP values and hence decreasing bit rates. All these figures refer to the FOR2 video, covering a range from 100 to 1100 Kbps. Four representative AVC configurations are considered: cases 0, 10, 11, and 17.

The rate-distortion results (Figure 12) show a typical logarithmic behavior. For all bit rates, the complex configurations (cases 10 and 11) achieve at least a 2 dB PSNR increment versus the simple one (case 0). As expected from Section 5.1, the coding performances with a search size of 16 and 32 are practically the same. With respect to the M4 standard, the same PSNR results are achieved with a 50% reduced bit rate enabling full-rate video communication over today wireless and wired networks. For instance, according to the results of Figure 12, a complex CIF video like Foreman can be transmitted at 25 Hz and 36 dB with less than 300 Kbps being compatible with 3G wireless network capabilities. The analysis of Figure 12 for the whole bandwidth range further highlights the importance of AVC: even in case of broadband networks (e.g., xDSL and WLAN), nowadays multimedia communication terminals, based on MPEG-4 Part 2 and H.263 technologies, lead to a video coding and transmission 3 dB poorer on the same bit rate or they double the bandwidth (thus increasing the cost of the service) required to reach a certain PSNR level. Moreover, the high coding efficiency of AVC allows the insertion of some redundancy in the source coder to improve the transmission robustness in error-prone channels [22, 23, 33].

As already shown in Section 5.1, a proper use of the AVC tools allows for nearly the same efficiency as the com-

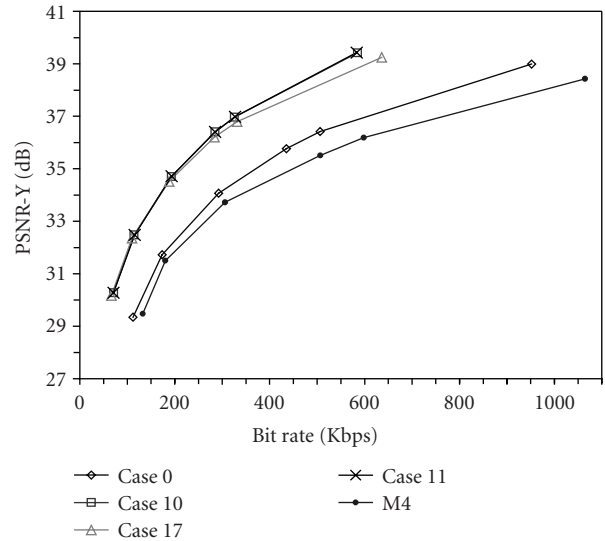


FIGURE 12: PSNR-Y (dB) versus bit rate in case of FOR2.

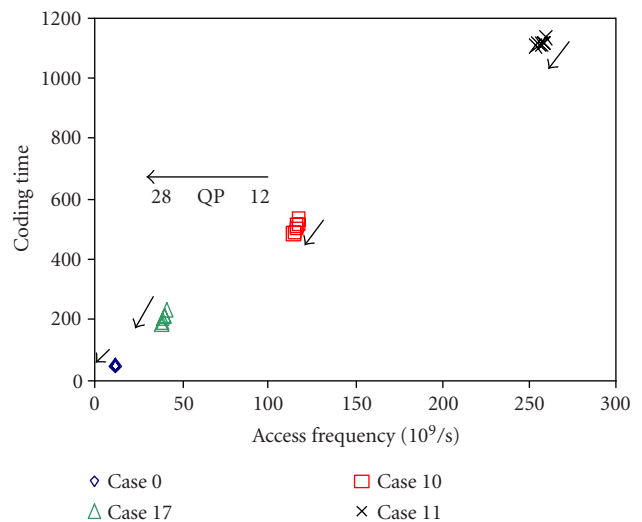


FIGURE 13: Encoder complexity for various QP values.

plex cases with a considerable complexity reduction. Indeed, while in complexity figures (Figures 13 and 14) the implementation cost of case 17 is close to the simple one, in Figure 12 the relevant coding efficiency results are close to the complex ones (the difference between case 17 and complex curves is below 0.4 dB and for $QP \geq 16$ the same results are achieved). The encoder data transfer and processing time practically do not depend on QP: indeed for each test case in Figure 13, the points with different QP values show nearly the same coding time and access frequency. At the decoder side (Figure 14), this dependency is more noticeable: as expected from literature [30], the higher the QP value (and hence the lower the bit rate), the lower the complexity. Finally, the storage requirements at both the encoder and the decoder (Figure 15) sides are not affected by the selected QP.

The dependency of AVC performance and complexity on the QP value has also been analyzed for the other test videos at lower (MD, FOR1) and higher (CM) rates. The achieved results are similar to those presented for FOR2.

5.3. Low-complexity AVC configuration

To test some low-complexity configurations not included in the basic scheme (half-pixel accuracy instead of quarter one and without in-loop deblocking), the JM2.1 code was suitably modified. Results for the same test cases of previous sections prove that restricting to half-pixel resolution decreases the compression efficiency (up to 30%, particularly for complex video inputs). Reducing the pixel accuracy can be useful only for very low rate video (MD) coded with a complex AVC profile. In this case, the lower pixel accuracy does not affect the coding efficiency and allows for a complexity reduction (both access frequency and processing time) of 10 and 15% for the encoder and decoder, respectively. As concerns deblocking, its use leads to PSNR (up to 0.7 dB) and bit rate (up to 6% saving) improvements. The complexity overhead is negligible at the encoder side and is up to 6% (access frequency) and 10% (processing time) at the decoder side. As proved in literature [13], the PSNR analysis is not enough for a fair assessment of the deblocking tool since a subjective analysis is also required. The latter, in addition to the above rate-distortion gain, confirms the effectiveness of the insertion of deblocking within the basic standard profile [1].

6. PERFORMANCE VERSUS COST TRADE-OFF USING PARETO ANALYSIS

As shown in Section 5.1, achieving a good balance between algorithmic performance (coding efficiency) and cost (memory and computational complexity) is the first step to address the challenge of a cost-effective AVC realization. A Pareto curve [9, 34, 35] is a powerful instrument to select the right trade-off between these conflicting issues at system level. In a search space with multiple axes, it only represents the potentially interesting points and excludes all the others having an equally good or worse solution for all the axes.

The multi objective design space exploration is reported in this section for the FOR2 (see Figures 16, 17, 18, and 19) and CM (see Figures 20, 21, 22, and 23) video inputs. The algorithmic performance is measured as the required bit rate achieving a fixed PSNR (36 dB for the target FOR2 video, covering a range from 250 to 500 Kbps, and 37.6 dB for the CM video, covering a range from 1000 to 3000 Kbps). The Pareto analysis has also been applied to the other video inputs at very low bit-rates (MD, covering a 20–50 Kbps range) and low bit rates (FOR1, covering a 80–200 Kbps range) achieving similar results to those obtained for the FOR2 video.

Figures 16 and 17 sketch the FOR2 sequence Pareto curves for the encoder using as cost metrics the memory access frequency and the peak memory usage. Figures 18 and 19 show the same analysis for the decoder. A rhombus represents the simple AVC configurations (cases 0 and 1), a cross refers to test cases from 2 to 9, a square represents complex AVC configurations (cases 10 to 12), a triangle indicates the

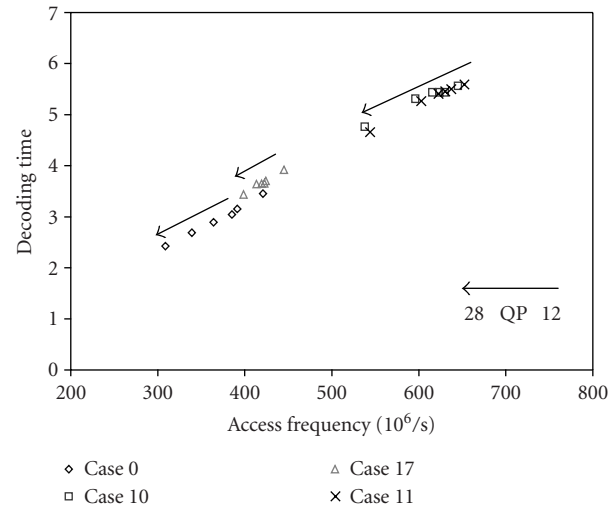


FIGURE 14: Decoder complexity for various QP values.

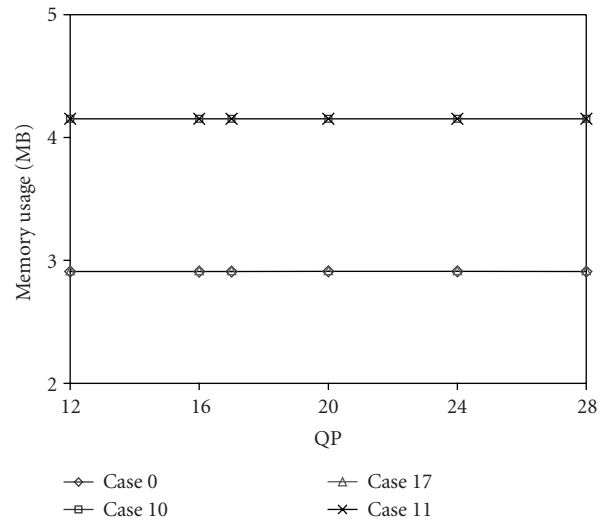


FIGURE 15: Decoder memory usage versus QP.

cost-efficient configurations (cases 13 to 17), and finally a circle identifies the M4 results. A Pareto analysis for the processing time is not presented since it is linear with the access frequency (see Figures 8, 9, 10, 11, 13, and 14), leading to the same conclusions.

A simple AVC configuration (case 0) outperforms M4 since a lower bit rate (greater performance) is achieved for the same costs. Among the 18 tests, cases 4, 5, 6, 7, 8, 9, 10, 11, 12 are not interesting (namely, the above Pareto curves) since they offer a certain coding performance at a higher cost with respect to points 0, 1, 2, 3, 13, 14, 15, 16, 17 near the Pareto curves. The latter points offer different optimal trade-offs. Case 0 is the less complex and 13 is the most performing in coding efficiency. The results at low (FOR1 test video) and very low (MD test video) bit rates lead to similar observations as the middle-rate ones achieved in the FOR2 analysis.

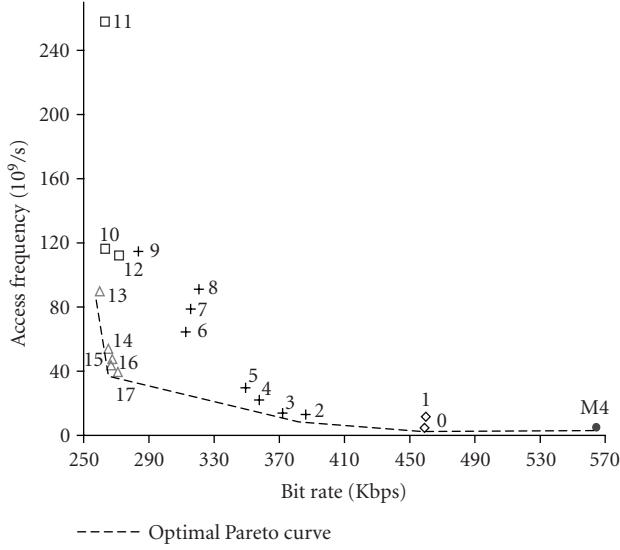


FIGURE 16: Encoder analysis, data transfer as cost in case of FOR2.

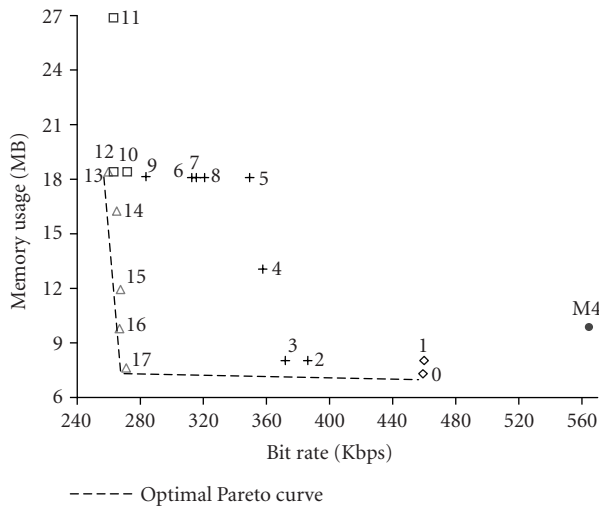


FIGURE 17: Encoder analysis, data storage as cost in case of FOR2.

The above analysis presents some differences when applied to high-rate video applications. With reference to the CM test, Figures 20 and 21 sketch the Pareto curves for the encoder using as cost metrics the memory access frequency and the peak memory usage. Figures 22 and 23 show the same analysis for the decoder. Differently from the results of Figures 16, 17, 18, and 19, in Figures 20, 21, 22, and 23, complex configurations such as cases 9, 10, and 12 are near the Pareto optimal curves.

From the combined analysis of the Pareto plots (Figures 16, 17, 18, 19, 20, 21, 22, and 23) and their description in Tables 2 and 3 and Section 5.1, the following considerations can be derived, valid for all kind of sequences (both low and high bit rates).

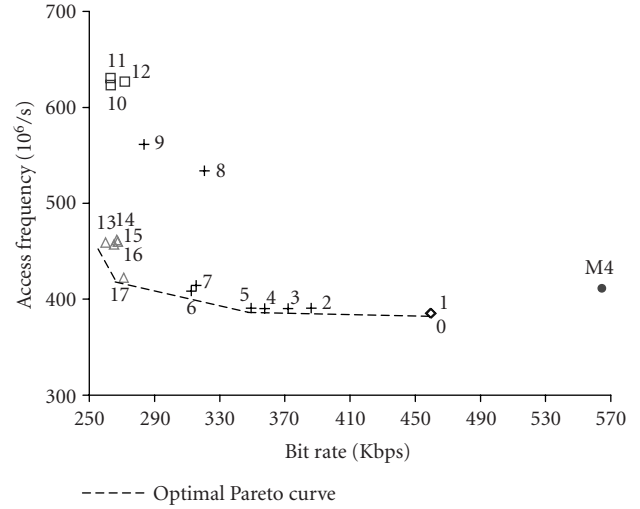


FIGURE 18: Decoder analysis, data transfer as cost in case of FOR2.

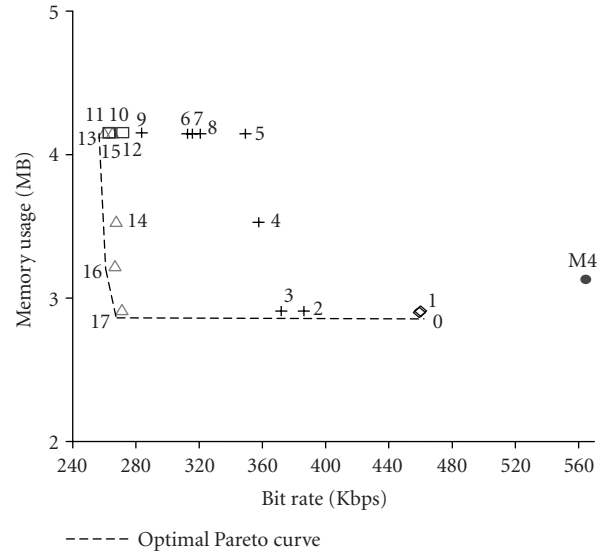


FIGURE 19: Decoder analysis, data storage as cost in case of FOR2.

- (i) The main AVC bottleneck is the combination of multiple reference frames and large search sizes. These tools have a limited impact on coding efficiency but a great one on complexity (up to a factor 60).
- (ii) The use of Hadamard should be avoided since complexity is increased without any coding efficiency gain.
- (iii) The adoption of multiple block sizes results in higher coding efficiency. While the complexity increases linearly with the number of block sizes, the major part of the gain is already achieved with the first 4 block sizes (16×6 , 16×8 , 8×16 , and 8×8 pixels).
- (iv) The CABAC entropy coder provides a consistent bit saving (10%) at the expense of a computational and memory increase (up to 30%) compared to UVLC.

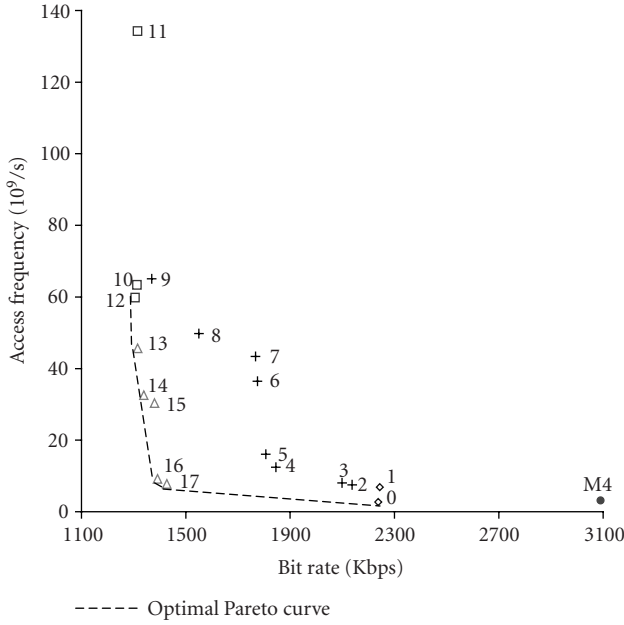


FIGURE 20: Encoder analysis, data transfer as cost in case of CM.

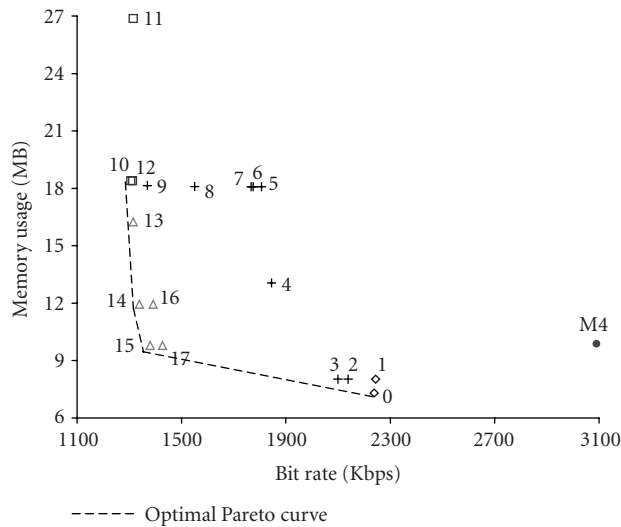


FIGURE 21: Encoder analysis, data storage as cost in case of CM.

- (v) RD-Lagrangian techniques give a substantial compression efficiency improvement, but the complexity doubles when the codec configuration entails a lot of coding modes and motion estimation decisions.
- (vi) For all video inputs, the use of B-frames reduces the bit rate (10% in average for the considered tests) for a complexity increase particularly noticeable at the decoder (from 20 to 40% extra cost specially at low rates). A larger number of B pictures increases the latency of the system (B frames need backward and forward reference pictures to be reconstructed). Thus, this tool is not used in low-latency constrained video applica-

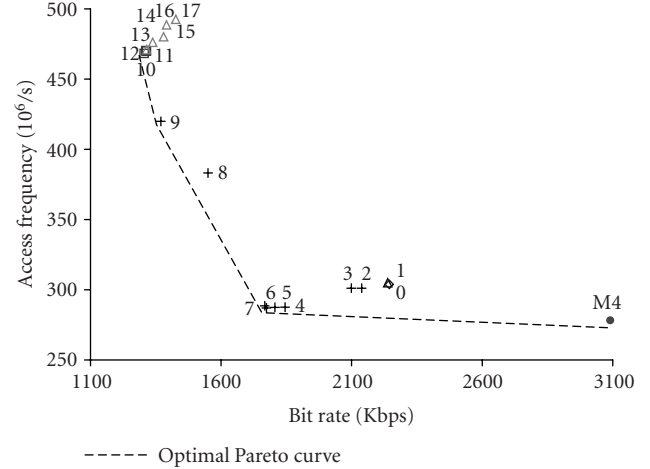


FIGURE 22: Decoder analysis, data transfer as cost in case of CM.

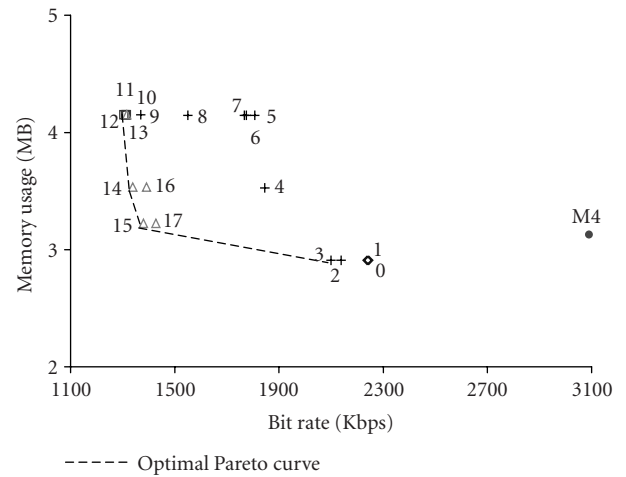


FIGURE 23: Decoder analysis, data storage as cost in case of CM.

tions and is typically not supported in baseline standard profiles [1, 5].

The effect of some tools differs when applied to different sequences. Comparing the nonoptimal Pareto points in Figures 16, 17, 18, and 19 with their description in Table 2 provides useful hints on the AVC video tools for *video applications at low and middle bit rates (i.e., few tens up to hundreds of Kbps)*:

- (i) the use of the eighth-pixel resolution leads to a complexity increase without any coding efficiency gain;
- (ii) the use of B-frames for very low-bit-rate sequences as MD provides a low improvement in compression efficiency for the complexity increase it involves.

Different results emerge (see Figures 20, 21, 22, and 23 for the CM video test) when a similar analysis is applied to high-bit-rate video applications (thousands of Kbps):

- (i) a higher pixel accuracy, eighth pixel instead of the basic quarter one, is a useful tool since it allows the same PSNR performance for at least 12% bit rate reduction (compare point 8 to point 7). The complexity increase is the same as for middle and low rates: roughly 15% for the encoder and 30% for the decoder as concerns data transfer and processing time. The impact on peak memory usage is negligible,
- (ii) multiple reference frames are more useful (e.g., 5 reference frames lead to roughly 15% bit saving), where most of the bit saving is already achieved with 3 reference frames.

The above analysis is a static evaluation of the algorithmic performance and the required complexity to assess the efficiency of the video coding tools. It provides a basis for automatic tool selection and gives pointers for the development of a resource manager in future work.

7. AVC PROFILES

The results of the performance versus cost Pareto analysis in Section 6 provide inputs to assist the profile definition in the standard. A profile defines a set of coding tools that can be used for generating a conforming bitstream. All decoders conforming to a specific profile must support all features in that profile. Encoders are not required to make use of any particular set of features supported in a profile but they have to provide bitstreams decodable by conforming decoders. In AVC/H.264, three profiles are defined: the baseline, the extended, and the main profile [3]. With reference to the VCL video tools presented in Section 3.1,² the Baseline profile supports all new features in AVC (multireference frames, variable block sizes, quarter-pixel accuracy, in-loop deblocking, integer spatial transform, and spatial prediction for intracoding) except 1/8-pixel accuracy, CABAC and B pictures. The Extended profile supports all features of the Baseline profile plus B frames and some tools for error resiliency (e.g., switching pictures SP/SI and data partitioning [3]). The Main profile supports all VCL features described in Section 3.1 except eighth-pixel accuracy.³ Baseline and Extended profiles are tailored for conversational services (typically operating below 1 Mbps) and streaming services (typically operating in the range 50–1500 Kbps), while entertainment video applications (several Mbps) would probably utilize the main profile.

The results of the VCL analysis presented in this paper are aligned with the profile considerations made by the standards body with the exception of the eighth-pixel accuracy which is no longer included in the last AVC release [3]. According to the results of Section 6, this choice is suitable for

applications not targeting a high-rate, high-quality video scenario or when the low cost is the main issue (e.g., wireless video). In high-rate multimedia applications (e.g., thousands of Kbps for the test CM in Section 6), an increased pixel accuracy should be adopted since it leads to a noticeable coding efficiency gain. This consideration suggests that future extensions of the standard to high-quality video scenario, currently being considered by AVC, could/should envisage a pixel accuracy higher than quarter.

8. CONCLUSIONS

The advanced video codec (AVC) is recently defined in a joint standardization effort of ITU-T and ISO/IEC. This paper introduces this new video codec together with its performance and complexity co-evaluation. First, a description of the upcoming standard including both the encoder and the decoder architectures is addressed. Then, an exhaustive analysis of the coding efficiency versus complexity design space is carried out over a wide variety of video contents at the early algorithmic design phase. Since the increasing complexity of multimedia applications makes high-level system exploration time consuming and error prone, the co-evaluation approach is supported by a framework for automated analysis of the C-level specification. Different from known profiling methodologies, focusing mainly on PSNR, bit rate, and computational burden, the proposed approach also investigates memory metrics (data transfer and storage). Real-life implementations of H.263 and MPEG-4 systems demonstrate that multimedia applications are data dominated: data transfer and storage are the dominant cost factors for both hardware- and software-based architectures.

The simulation results show that AVC outperforms current video coding standards (up to 50% bit saving for the same PSNR) offering the enabling technology for a widespread diffusion of multimedia communication over wired and wireless transmission networks. However, this outstanding performance comes with an implementation complexity increase of a factor 2 for the decoder. At the encoder side, the cost increase is larger than one order of magnitude. This represents a design challenge for resource constrained multimedia systems such as wireless and/or wearable devices and high-volume consumer electronics, particularly for conversational applications (e.g., video telephony), where both the encoder and the decoder functionalities must be integrated in the user's terminal.

The analysis also highlights important properties of the AVC framework allowing for complexity reduction in the early algorithmic design phase. When combining the new coding features, the relevant implementation complexity accumulates, while the global compression efficiency saturates. As a consequence, a proper use of the AVC tools maintains roughly the same coding performance as the most complex configuration (all tools on) while considerably reducing complexity (up to a factor 6.5 for the encoder and 1.5 at the decoder side). A single AVC configuration able to maximize algorithmic performance while minimizing memory and computational burdens does not exist. However,

²A detailed analysis of AVC/H.264 profiles, including tools for error resiliency and network friendly data packaging, can be found in [3].

³The Main profile does not support some tools for error resiliency, such as flexible macroblock ordering [3], which are supported by the baseline profile. Thus, only a subset of the coded video sequences decodable by a baseline profile decoder can be decoded by a main profile decoder.

different configurations leading to several performance/cost trade-offs exist. To find these optimal configurations, and hence to highlight the bottlenecks of AVC, a Pareto multiobjective analysis is presented to explore the five-dimensional design space of PSNR, bit rate, computational burden, and memory access frequency and storage. The reported results provide inputs to assist the definition of profiles in the standard and represent the first step for a cost-effective implementation of the new AVC.

REFERENCES

- [1] T. Wiegand, "Joint Final Committee Draft (JFCD) of joint video specification (ITU-T Rec. H.264, ISO/IEC 14496-10 AVC)," Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-D157, Klagenfurt, Austria, July 2002.
- [2] ftp://ftp.imtc-files.org/jvt-experts/reference_software
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [4] D. Alfonso, D. Bagni, L. Celetto, and L. Pezzoni, "Detailed rate-distortion analysis of H.264 video coding standard and comparison to MPEG-2/4," in *Proc. of SPIE Conference on Visual Communications and Image Processing*, Lugano, Switzerland, July 2003.
- [5] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 849–866, 1998.
- [6] P. Kuhn and W. Stechele, "Complexity analysis of the emerging MPEG-4 standard as a basis for VLSI implementation," in *Visual Communications and Image Processing*, vol. 3309 of *Proc. of SPIE*, pp. 498–509, San Jose, Calif, USA, January 1998.
- [7] ISO/IEC 14496-2, "Information technology—Generic coding of audio-visual objects," N 3056, Maui, Hawaii, USA, December 1999.
- [8] F. Catthoor, S. Wuytack, E. De Greef, F. Balasa, L. Nachtergaele, and A. Vandecapelle, *Custom Memory Management Methodology: Exploration of Memory Organisation for Embedded Multimedia System Design*, Kluwer Academic, Boston, Mass, USA, 1998.
- [9] F. Catthoor, S. Wuytack, E. De Greef, F. Balasa, L. Nachtergaele, and A. Vandecapelle, *Data Access and Storage Management for Embedded Programmable Processors*, Kluwer Academic, Boston, Mass, USA, 2002.
- [10] J. Bormans, K. Denolf, S. Wuytack, L. Nachtergaele, and I. Bolsens, "Integrating system-level low power methodologies into a real-life design flow," in *IEEE Proc. 9th Int. Workshop Power and Timing Modeling Optimization and Simulation*, pp. 19–28, Kos, Greece, October 1999.
- [11] A. Chimienti, L. Fanucci, R. Locatelli, and S. Saponara, "VLSI architecture for a low-power video codec system," *Microelectronics Journal*, vol. 33, no. 5-6, pp. 417–427, 2002.
- [12] T. Meng, B. Gordon, E. Tsern, and A. Hung, "Portable video-on-demand in wireless communication," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 659–680, 1995.
- [13] J. Jung, E. Lesellier, Y. Le Maguet, C. Miro, and J. Gobert, "Philips deblocking solution (PDS), a low complexity deblocking for JVT," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-B037, Geneva, February 2002.
- [14] L. Nachtergaele, D. Moolenaar, B. Vanhoof, F. Catthoor, and H. De Man, "System-level power optimization of video codecs on embedded cores: a systematic approach," *Journal of VLSI Signal Processing*, vol. 18, no. 2, pp. 89–111, 1998.
- [15] K. Denolf, P. Vos, J. Bormans, and I. Bolsens, "Cost-efficient C-level design of an MPEG-4 video decoder," in *Proc. IEEE Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation: 10th International Workshop*, D. Soudris, P. Pirsch, and E. Barke, Eds., vol. 1918 of *Lecture Notes in Computer Science*, pp. 233–242, Springer-Verlag, Heidelberg, September 2000.
- [16] M. Takahashi et al., "A 60-MHz 240-mW MPEG-4 video-phone LSI with 16-Mb embedded DRAM," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1713–1721, 2000.
- [17] <http://www.imec.be/atomium>.
- [18] H.-J. Stolberg, M. Berekovic, and P. Pirsch, "A platform-independent methodology for performance estimation of streaming media applications," in *Proc. IEEE International Conference on Multimedia and Expo*, pp. 105–108, Lausanne, Switzerland, August 2002.
- [19] C.-G. Zhou, L. Kohn, D. Rice, I. Kabir, A. Jabbi, and X.-P. Hu, "MPEG video decoding with the UltraSPARC visual instruction set," in *40th IEEE Computer Society International Conference*, pp. 470–475, San Francisco, Calif, USA, March 1995.
- [20] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, San Francisco, Calif, USA, 1996.
- [21] ISO/IEC 14496-5, "Information technology—Generic coding of audio-visual objects—Part 5: Amd1: Simulation software," N 3508, Beijing, July 2000.
- [22] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, 2003.
- [23] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
- [24] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.
- [25] D. Marpe, G. Blättermann, G. Heising, and T. Wiegand, "Video compression using context-based adaptive arithmetic coding," in *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 558–561, Thessaloniki, Greece, October 2001.
- [26] V. Lappalainen, A. Hallapuro, and T. Hämäläinen, "Optimization of emerging H.26L video encoder," in *Proc. IEEE Workshop on Signal Processing Systems*, pp. 406–415, Antwerp, Belgium, September 2001.
- [27] V. Lappalainen, T. Hämäläinen, and A. Hallapuro, "Performance analysis of low bit rate H.26L video encoder," in *IEEE Trans. Acoustics, Speech, and Signal Processing*, pp. 1129–1132, May 2001.
- [28] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of the ITU-T draft H.26L video coding standard," in *Proc. 12th International Packet Video Workshop*, Pittsburgh, Pa, USA, April 2002.
- [29] M. Hagai, K. Abe, S. Kadono, S. Kajita, and M. Schlockerman, "H.26L implementation evaluation," ISO/IEC MPEG, document M7666, Pattaya, December 2001.
- [30] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 704–716, 2003.
- [31] K. Denolf et al., "Initial memory complexity analysis of the AVC codec," in *Proc. IEEE Workshop on Signal Processing Systems*, pp. 222–227, San Diego, Calif, USA, October 2002.
- [32] H.-Y. Cheong and A. Tourapis, "Fast motion estimation within the H.264 codec," in *Proc. IEEE International Conference on Multimedia and Expo*, pp. 517–520, Baltimore, Md, USA, July 2003.

- [33] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, 2000.
- [34] V. Pareto, *Cours D'Economie Politique*, vol. I-II, Lausanne, Switzerland, 1896.
- [35] E. Brockmeyer, A. Vandecappelle, and F. Catthoor, "Systematic cycle budget versus system power trade-off: a new perspective on system exploration of real-time data-dominated applications," in *Proc. International Symposium on Low Power Electronics and Design*, pp. 137–142, ACM Press, New York, 2000.

Sergio Saponara received his M.S. degree in electronics and his Ph.D. degree in information engineering, both from the University of Pisa, Italy, in 1999 and 2003. In 2001, he collaborated with "Consorzio Pisa Ricerche" on a MEDEA+ project related to the low-power design of xDSL modem. In 2002, he was with MICS group at IMEC, Leuven, Belgium, under a Marie Curie research scheme working on the complexity analysis of advanced video coding standards. Currently, he is a Researcher at Pisa University, working on algorithms and VLSI architecture design for multimedia and low-power CMOS design methodologies.



Kristof Denolf received the Industrial Engineering degree in electronics from the Katholieke Hogeschool, Brugge-Oostende, Belgium, in 1998, and the M.S. degree in electronic system design from Leeds Metropolitan University, Leeds, UK, in 2000. He joined the Multimedia Image Compression Systems (MICS) group at the Interuniversity Microelectronics Centre (IMEC), Leuven, Belgium, in 1998. His current research interests are the cost-efficient design of advanced video processing systems and the end-to-end quality of experience.



Gauthier Lafruit was a Research Scientist with the Belgian National Foundation for Scientific Research from 1989 to 1994, being mainly active in the area of wavelet image compression implementations. Subsequently, he was a Research Assistant in the Vrije Universiteit Brussel (VUB), Belgium. In 1996, he became the recipient of the Scientific Barco Award and joined IMEC, where he was involved as a Senior Scientist with the design of low-power VLSI for combined JPEG/wavelet compression engines. He is currently the Principal Scientist in the MICS group with IMEC. His main interests include progressive transmission in still image, video and 3D object coding, as well as scalability and resource monitoring for advanced, scalable video, and 3D coding applications. He is the author/coauthor of around sixty scientific publications, fifty MPEG standardization contributions, and five patent(s) (applications) and has participated (and has been appointed as Evaluator) in several national and international project(s) (proposals).



Carolina Blanch was born in Pamplona, Spain. She received her M.S. degree in telecommunications engineering in January 2000 from the University of Navarra, Spain. She received a scholarship from the Flemish community to continue her studies in Belgium, and obtained an M.S. in artificial intelligence from the Katholieke Universiteit Leuven in 2001. Since December 2001, she is working as a Research Engineer in the MICS group at IMEC, Leuven, Belgium. She has been involved in the complexity assessment of video codecs and has several related articles. Currently, she is involved in cross layer optimization and network adaptation of video codecs.



Jan Bormans has been a Researcher at the Information Retrieval and Interpretation Sciences laboratory, the Vrije Universiteit Brussel (VUB), Belgium, in 1992 and 1993. In 1994, he joined the VLSI Systems and Design Methodologies (VSDM) division of the IMEC research center in Leuven, Belgium. Since 1996, he is heading IMEC's Multimedia Image Compression Systems group. This group focuses on the efficient design and implementation of embedded systems for advanced multimedia applications. He is the Belgian head of the delegation for ISO/IEC's MPEG and SC29 standardization committees. He is also MPEG-21 requirements Editor and Chairman of the MPEG liaison group.

