

# Distortion-Energy Analysis of an OMAP-Based H.264/SVC Decoder

Eduardo Juárez\*, Fernando Pescador, Pedro J. Lobo,  
Angel Groba, and César Sanz

Grupo de Diseño Electrónico y Microelectrónico (GDEM)  
Universidad Politécnica de Madrid (UPM)  
Ctra de Valencia Km 7. 28031 Madrid, Spain  
{ejuares, pescador, pjlobo, amgroba, cesar}@sec.upm.es

**Abstract.** This paper describes a Pareto frontier estimate of a scalable video decoder embedded in an OMAP-based multimedia terminal within the distortion-energy optimization space. A metric to estimate video distortion has been introduced. In addition, energy consumption estimates are obtained from real-time measurements of the computational load. Finally, test-bench operation is successfully demonstrated with different H.264/SVC-compliant sets of sequences.

**Keywords:** Distortion-Energy optimization, Scalable Video Coding, Video Quality Estimation.

## 1 Introduction

System-level energy optimization of battery-powered multimedia embedded systems has recently become a design goal. The poor operational time of multimedia terminals makes computationally demanding applications impractical in real scenarios. For instance, the so-called smart-phones are currently unable to remain in operation longer than several hours [1]. Moreover, because no step change in energy densities of lithium-based batteries is predicted in the near future [2], storage technology improvements alone will achieve no significant increase in terminal operational time. System-level solutions to maximize operational time have already been proposed in the literature [3][4][5][6]. However, despite the fact that degradations of perceived multimedia quality prevent technology user adoption [7], this performance parameter has usually been discarded as an optimization goal.

A multi-objective optimization that simultaneously considers perceived video quality and global energy consumption has already been envisaged [8]. The aim is to vary the perceived multimedia quality to achieve the maximum operational time. Generally speaking, both quality and system energy consumption depend on parameters such as power amplifier transmission power, linearity and video

---

\* This work was supported by the Spanish Ministry of Science and Technology under grants TEC2009-14672-C02-01 and TEC2006-13599-C02-01.

distortion, among others [9]. The set of possible values of these control parameters can be viewed as a set of points in a multidimensional control space. Depending on the battery state-of-charge, efficient multimedia terminals can search the set of system control points that simultaneously optimize operational time and multimedia quality [9].

A Pareto-optimum system control point [10] is found when an increase in the terminal operational time is only accomplished at the expense of multimedia quality or, conversely, when further improvement in quality is only realized with a simultaneous decrease in operational time. The set of Pareto-optimum system control points is known as the Pareto frontier [11].

To avoid the excessive overhead of finding the optimum system control points at runtime, the Pareto frontier can first be characterized at design time based on a scenario definition; later, at runtime, this can be used to select an optimum system control point as a function of the battery state-of-charge [8]. It is worth noting that achieving the maximum quality is equivalent to reaching the minimum distortion; similarly, accomplishing the maximum operational time is equivalent to reaching the minimum global energy consumption.

The methodology to obtain the Pareto frontier at design time could be summarized as follows. A scenario in which a multimedia terminal receives and decodes a Transport Stream (TS) is assumed. In this context, network- and multimedia-related features, especially real-time video decoding, will be the main greedy energy consumers. For instance, Neuvo [12] indicates that two-thirds of the 3W power budget of a third-generation (3G) mobile phone in a 384 Kbps video streaming scenario accounts for network- and multimedia-related features. Next, the quality and consumption behavior of the multimedia terminal are estimated or measured for a defined scenario based on the system control points. Finally, Pareto-optimum system control points belonging to the Pareto frontier are identified to be used at runtime.

Although multi-objective terminal characterizations have already been proposed in the literature [9], to the best of our knowledge, no detailed results have been presented on terminals with current state-of-the-art video decoder implementations. This paper describes an OMAP-based [13] Pareto frontier estimate of a video decoder embedded in a multimedia terminal within the distortion-energy optimization space.

The rest of the paper is organized as follows. In section II, a controllable energy consumption video decoder is described. Section III details the test-bench used to characterize the Pareto frontier of a multimedia terminal. In section IV, the results achieved for different sequences are discussed. Finally, in section V, conclusions are drawn, and future work is proposed.

## **2 Solution Based on Scalable Video Coding**

### **2.1 Introduction**

To control the energy consumption of a video decoder embedded in a battery-powered portable device, the H.264/SVC (Scalable Video Coding) standard [14], [15] is an appropriate choice. In this standard, the video compression is performed by

generating a unique hierarchical bit-stream structured in several levels or layers of information, consisting of a base layer and several enhancement layers. The base layer provides basic quality. The enhancement layers provide improved quality at increased computational cost and energy consumption. Because the energy consumption depends on the particular layer to decode, an H.264/SVC decoder is a very well-suited solution for managing the energy consumption by selecting the appropriate layer. H.264/SVC was standardized as an annex of H.264/AVC standard in 2007 to cover the needs of scalability.

This standard specifies three types of scalabilities: spatial, temporal and quality. The three types of scalability can be combined into a bit-stream. As an example, consider a coded video sequence that has three temporal layers, three spatial layers and three quality layers. A mobile device that has a medium charged battery may decode, for instance, the third spatial layer to get full spatial resolution, the second temporal layer to get half temporal resolution and the first quality layer to get a low-quality level. A device that has a fully charged battery might decode the entire bit-stream to get the full temporal and spatial resolution as well as the higher quality.

## 2.2 Open SVC Decoder

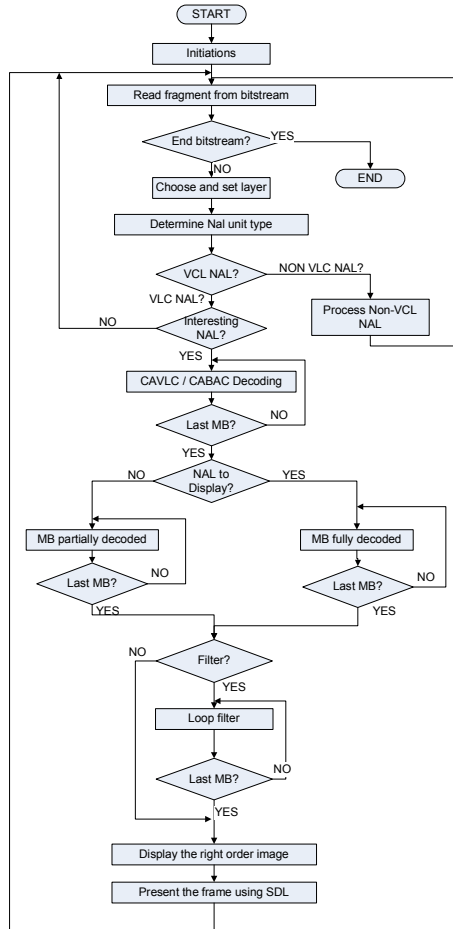
IETR has developed the Open SVC Decoder [16], a C language Baseline Profile SVC decoder supporting all tools to deal with spatial, temporal and quality scalabilities. It is based on a fully compliant H.264 baseline decoder with most of the tools of the Main Profile. Only interlaced coding and the weighted prediction are not supported because of their complexity for embedded systems.

The Open SVC Decoder has been developed in the framework of Scalim@ges [17] project. This project aimed to promote H.264/SVC standard in order to reduce the number of formats manipulated in production, distribution, and use of video compatible with existing solutions. Currently, others French and international projects like SVC4QoE [18] and ScalNet [19] are using Open SVC Decoder.

The Open SVC Decoder, contrary to the JSVM which decodes all layers in a bit-stream, can decode a specific layer with a specific scalability. This particularity provides an adaptability of the decoder over different platforms by selecting the right layer in order to have a real-time decoding. The changing of layer can be also done during the decoding process “on the fly”.

In Fig. 1, a simplified flow diagram of the decoding process for a H.264/SVC compliant stream is shown. The decoder reads the H.264/SVC stream from an input buffer and decodes the NAL units in sequence. After decoding the NAL header, the NAL unit content is identified as a slice header or another syntax element (e.g. an SPS or a PPS). When the NAL unit contains an interesting slice for the selected layer, the decoder extracts all the syntactical elements from the stream and stores them in an intermediate buffer. If the processed NAL must be displayed, each macroblock (MB) is completely decoded, however, if the NAL must not be displayed the MB is partially decoded.

In the next step, if a frame has been completely decoded, the deblocking filter is applied. Finally, the decoded pictures are stored in images buffers and presented in the right order using the PC Simple Direct Media Layer (SDL) library [20].



**Fig. 1.** Simplified Open SVC Decoder Flow Chart

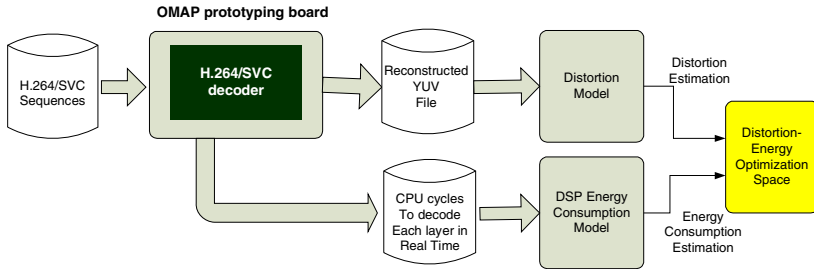
The Open SVC Decoder has been compared to the JSVM 9.16 to benchmark and to test the conformance of the decoder. The performance of the Open SVC decoder is up to 50 times faster than the JVSM decoder [21], making this decoder a good starting point in the development of an embedded H.264/SVC decoder.

### 3 Test-Bench

To find the Pareto-optimum system control points that minimize the distortion of the decoded sequence and the decoder energy consumption, the test-bench shown in Fig. 2 has been implemented.

The test-bench consists of a decoder that implements the standard H.264/SVC using a development platform based on an OMAP [13] processor from Texas Instruments. The test-bench decodes a layer of a sequence and measures the number of CPU clock cycles required to decode each frame.

Using an energy consumption model of the OMAP processor and measuring the processor computational load, it is possible to estimate the energy consumption of the decoder for each layer. On the other hand, the distortion of the decoded sequences is estimated by assigning to each of the layers a value as a function of the spatial, temporal and quality resolutions. Both estimations define a two-dimensional optimization space in which it is possible to select the combinations that simultaneously minimize both parameters.



**Fig. 2.** Test-bench Implemented to Find the Distortion-Energy Optimization Points

The following subsections describe the selected DSP core, the implementation of a H.264/SVC decoder, the generated sequences for the tests and the estimation models of energy consumption and distortion.

#### 3.1 Prototyping Platform

The OMAP3530 [13] processor basically consists of two processing cores, a General Purpose Processor (GPP) and a Digital Signal Processor (DSP). The former, an ARM Cortex-A8 processor [22], is aimed to run a generic Operating System (OS) while the latter, a DSP core based on the C64+ [23], has an architecture optimized for video processing.

The ARM Cortex processor has two levels of cache (L1 and L2). The program (L1P) and data (L1D) caches, within the Microprocessor Unit (MPU) Subsystem, consist of a 16 KB memory space. The L2 cache consists of a 256 KB memory space, shared between program and data. In addition, the MPU integrates a NEON coprocessor, optimized for multimedia applications, with its own multiplication-accumulation unit (MAC) and support for floating-point operations.

The fixed-point DSP core has two levels of internal memory (L1 and L2). The L1P memory/cache consists of a 32 KB memory space and the L1D memory consists of an 80 KB memory space. Both memories can be configured as cache memories, general-purpose memories or a combination of both. Finally, the L2 memory/cache consists of a 64 KB memory space, shared between the program and data. L2 memory can be configured as a general-purpose mapped memory, a cache memory, or a combination of both.

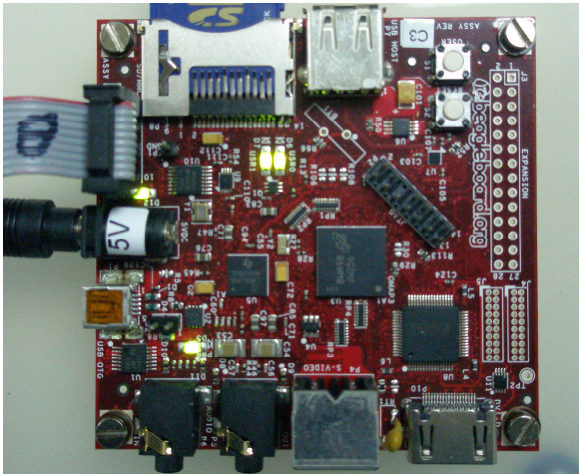
In the test-bench, the C64+ DSP L1P and L1D internal memories have been configured as 32 KB level-1 cache memories, and the L2 internal memory, as a 64 KB of level-2 cache memory.

The *BeagleBoard* [24], a commercial prototyping board (Fig. 3) based on the OMAP processor, has been used to test the Open SVC Decoder and measure its performance as the number of CPU cycles spent to decode each frame of a specific layer.

The board has 128 MB of SDRAM external memory, 256 MB of Flash external memory and several interfaces. Note that the clock frequency of the ARM and DSP OMAP cores is 600 MHz and 520 MHz, respectively.

The decoder has been migrated to the *BeagleBoard* and a specific test application has been developed to measure the decoder performance. In this application, a test stream is read from a file and written into a stream buffer allocated in external memory. Subsequently, the DSP reads the stream from this memory, decodes it on a picture basis and writes the decoded picture into a buffer. The picture is also written into a file. Moreover, a file with real-time performance measurements is generated that includes the number of CPU cycles spent to decode each picture.

To execute automatic tests, a Perl script [25] has been generated using the script language available in Code Composer Studio [26]. This automatic test decodes all of the sequences and extracts profiling results in Excel file format.



**Fig. 3.** The BeagleBoard Prototyping System

### 3.2 Open SVC Decoder Migration into the Test-Bench

The Open SVC Decoder has been developed for a PC-based platform. Same changes have been done to integrate the decoder in the OMAP-based test-bench:

- The decoder has been encapsulated into a DSP-BIOS [27] task. The size of the stack associated to this task has been adjusted to 1 MB and has been allocated in external memory.
- Internal memory has been configured as follows: L1D is divided in 32 KB for cache memory and 48 KB for general purpose data; L1P is configured as a 32 KB cache program memory and L2 is splitted between level-2 cache memory and general purpose memory. Currently, neither the code nor data are allocated in internal memory but these memories have available space for future optimizations.
- The maximum size of the decoded pictures has been reduced from HD (1920×1080) to SD (720×576) to limit the amount of memory of the DSP implementation.
- The decoder output interface has been modified. In the original code, the decoded pictures are displayed on screen using the SDL library. In the DSP code, the decoded pictures are written in a YUV file.
- Functions used to access the stream files have been adapted to the functions available in the real-time support DSP libraries.
- The way to select the layer to be decoded has been modified. In the original code, the layer was selected using the command line arguments while in the DSP version these parameters are introduced through a configuration file that is parsed at the begging of the decoding process.

To confirm the DSP-based decoder conformance, an automatic test has been developed to compare pixel by pixel the sequences decoded by the Open SVC Decoder running in a PC with the sequences generated by the DSP. The results of this comparison demonstrate that the sequences decoded by the PC version are identical to the sequences decoded by the DSP.

After the migration process, some optimizations have been done to improve the decoder performance. The methodologies presented in **¡Error! No se encuentra el origen de la referencia.** have been applied to reduce the number of CPU cycles needed to decode a layer. Currently, the CABAC entropy decoding, the up-sampling filter, the luminance and chrominance interpolation, the deblocking filter and the qpel interpolation functions have been rewritten using assembler instructions.

### 3.3 Distortion Model

Generally speaking, each type of scalability affects the distortion of decoded scalable streams differently. Spatial scalability influences the size of decoded frames; temporal

scalability has an impact on video motion feeling, and quality scalability concerns Signal-to-Noise Ratio (SNR).

To have an objective measure of the global distortion of a decoded layer,  $L$ , the definition of a distortion parameter that includes the three kinds of scalabilities is needed. With this idea in mind, a normalized layer distortion parameter,  $\delta(L)$ , has been defined as the complement of the distance between the layer,  $L$ , and the maximum-distortion layer,  $L_B$ .

$$\delta(L) = 1 - d(L, L_B) . \quad (1)$$

To measure the layer separation, the weighted L1 (Manhattan) distance has been selected:

$$d(X, Y) = WL_1(X, Y) = \sum_{i=1}^3 w_i |x_i - y_i| . \quad (2)$$

The  $x_i$  and  $y_i$  are the scalability values of layers  $X$  and  $Y$ , and  $w_i$  are the weights for the spatial, temporal and quality dimensions. Frame size (Fs), frame-rate (Fr) and bit-rate (Br) have been used as metrics of the spatial, temporal and quality scalabilities, respectively. Note that the metric used for quality scalability is the sequence bit-rate instead of the Peak-SNR (PSNR). Bit-rate and PSNR parameters are directly related and usually the decoders can be configured to obtain a specific bit-rate but not a PSNR.

Given the previous definition, the normalized layer distortion parameter can be calculated as indicated in equation (3)

$$\delta(L) = 1 - \{DS(L) + TS(L) + QS(L)\} . \quad (3)$$

$DS(L)$ ,  $TS(L)$ , and  $QS(L)$  are the spatial, temporal and quality scalability components of the interlayer distance, defined in equations (4-6).

$$DS(L) = \frac{C_1}{Fs_{\max} - Fs_{\min}} |Fs(L) - Fs_{\min}| . \quad (4)$$

$$TS(L) = \frac{C_2}{Fr_{\max} - Fr_{\min}} |Fr(L) - Fr_{\min}| . \quad (5)$$

$$QS(L) = \frac{C_3}{Br_{\max} - Br_{\min}} |Br(L) - Br_{\min}| . \quad (6)$$

- $C_i$  are the scale coefficients,

$$\sum_{i=1}^3 C_i = 1 . \quad (7)$$

- $L$  is the selected layer.



- $F_s(L)$  is the frame size of layer  $L$ .
- $Fr(L)$  is the frame-rate of layer  $L$ .
- $Br(L)$  is the bit-rate of layer  $L$ .
- $F_{s_{\max}}$  and  $F_{s_{\min}}$  are the greatest and smallest layer frame sizes, respectively.
- $Fr_{\max}$  and  $Fr_{\min}$  are the greatest and smallest layer frame-rates, respectively.
- $Br_{\max}$  and  $Br_{\min}$  are the greatest and smallest layer bit-rates, respectively.

The scale coefficients,  $C_i$ , adjust the weight of each distance scalability component, DS, TS, and QS to fit the normalized layer distortion to subjective quality measurements. Their values vary between zero and one and verify equation (7).

### 3.4 Test Sequences

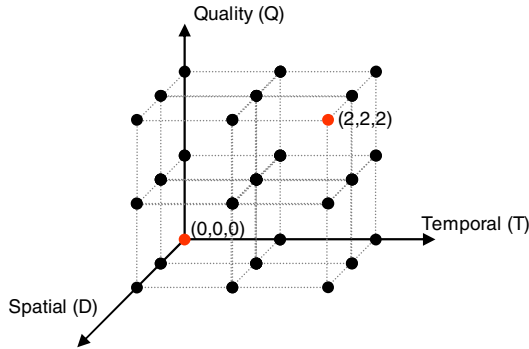
In order to assess the test-bench depicted in Fig. 2, the well-known video sequence foreman (50 frames, YUV 4:2:0) has been encoded using a commercial H.264/SVC codec [27]. Three 9-layer sequences have been generated. The layer structure of each sequence consists of all the possible combinations among three spatial resolutions (CIF, QCIF, and subQCIF) and three frame-rates (24 fps, 12 fps, and 6 fps). In addition, each sequence has been encoded with different bit-rate (0.5 Mbps, 1.0 Mbps, and 2.0 Mbps).

Note that three independent test sequences have been encoded instead of only one because neither the encoder nor the Open SVC Decoder supports bit-streams with 27 layers.

As far as the codec parameters to generate the three test sequences concern, the GOP size equals 16 progressive frames, the Context-based Adaptive Binary Arithmetic Coding (CABAC) tool is used for entropy coding, the deblocking filter is active, all possible macroblock partitions are enable for intra- and inter-prediction, a maximum of three reference frames is allowed, and, at last, 3 B-frames are coded for each I- or P-frame.

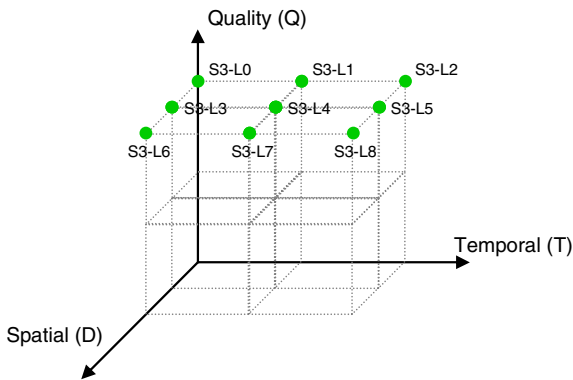
For each test sequence layer, the values of the metrics defined in the previous section have been ordered and mapped into three indexes: D, T and Q. The D index designates the spatial resolution, the T index symbolizes the frame rate and the Q index denotes the bit-rate level. Each combination of D, T and Q values, i.e., a layer, defines a control point.

Since the quality, space and temporal scalabilities of the test sequences includes three possible values, the triplet (D, T, Q) defines a three-dimensional global control space with 27 control points (See Fig. 4). For instance, the control point associated in Fig. 4 with the base layer of the first encoded sequence is (0, 0, 0), i.e. (subQCIF, 6 fps, 0.5 Mbps), and the maximum quality layer control point of the third encoded sequence is (2, 2, 2), i.e. (CIF, 24 fps, 2.0 Mbps).



**Fig. 4.** Test Sequences (D, T, Q)-Triplet Control Space

Each test sequence includes three spatial resolutions ( $D = 0, 1$  and  $2$ ) and three frame rates ( $T = 0, 1$  and  $2$ ). As can be seen in Fig. 5, the 9 control points of a test sequence define a subset of the global control space. In particular, Fig. 5 shows the subset of the third sequence for  $Q = 2$ , i.e. 2 Mbps of bit-rate. The layers within the sequence are labeled from L0 to L8.



**Fig. 5.** Control Point Set of the 2.0 Mbps Test Sequence ( $Q = 2$ )

The example generated sequences have been used with the test-bech shown in the Fig. 2. The results obtained are presented in the section 4.

### 3.5 Energy Consumption Model

The OMAP processor energy consumption model provided by the manufacturer [30] has been used. The model is basically divided in two energy components. The former, the baseline core energy, describes the energy consumption that is independent of any chip activity. The latter, the module active energy, describes the energy consumed by the active modules depending on resource usage.

The energy consumption estimates have been obtained setting the frequency value at 520 MHz and considering that only the Display Subsystem and the SDRAM memory controller are the active modules. It is worth noting that to obtain only DSP-based energy results, the ARM core of the OMAP processor has been considered as a non-active module.

## 4 Results

The test-bench depicted in Fig. 2 has been employed to measure the average computational load. This performance metric has been derived as the ratio of the average number of clock cycles per second to the DSP operational frequency. The active energy needed to process the 50-frame *foreman* test sequences defined in section 3.4 has been estimated from the average computational load with the energy consumption model described in section 3.5. Table 1 summarizes the consumed active energy, measured in millijoules (mJ), for each layer of the *foreman* sequences. Since the current Open SVC Decoder OMAP implementation does not achieved real-time performance for the layer (CIF, 24 fps, 2 Mbps), the corresponding active energy is not shown in Table 1. As can be seen from Table 1, frame size modifications account for most of the active energy change (more than a 10-time increase when the (SUB QCIF, 24 fps, 0.5 Mbps) and (CIF, 24 fps, 0.5 Mps) layers are considered). In contrast, a smaller increase (almost 70%) in active energy consumption is achieved varying the layer bit-rate from (SUB QCIF, 24 fps, 0.5 Mbps) to (SUB QCIF, 24 fps, 2.0 Mbps)

**Table 1.** Estimated Active Energy Consumption (mJ) for the *Foreman* Test Sequences

		0.5 Mbps	1 Mbps	2 Mbps
6 fps	SUB QCIF	23.5	29.2	39.8
	QCIF	64.4	74.7	95.0
	CIF	187.5	208.7	242.9
12 fps	SUB QCIF	37.7	46.2	62.0
	QCIF	126.7	143.7	179.3
	CIF	373.8	405.3	462.3
24 fps	SUB QCIF	63.6	77.1	102.7
	QCIF	238.7	268.5	332.2
	CIF	721.7	771.8	-

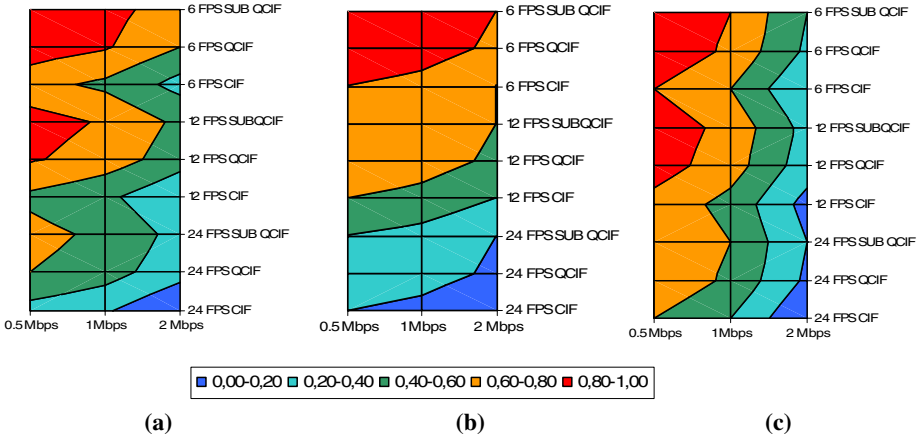
Table 2 provides the values of the normalized layer distortion,  $\delta(L)$ , for three different instances (named  $\delta_1$ ,  $\delta_2$  and  $\delta_3$ , respectively) of the model described in section 3.3. These model instances have been selected as three possible fittings of the proposed distortion model to subjective quality measurements conducted on the test sequences. The scale factors of the  $\delta_1$  model instance have been selected to provide

equal significance to the spatial, temporal and quality scalability components of the interlayer distance ( $C_1 = 1/3$ ,  $C_2 = 1/3$  and  $C_3 = 1/3$ ). Furthermore, the  $\delta_2$  model instance emphasizes the temporal scalability against the spatial and quality components ( $C_1 = 1/5$ ,  $C_2 = 3/5$  and  $C_3 = 1/5$ ) and the  $\delta_3$  model instance gives the prominence to the quality scalability component ( $C_1 = 1/5$ ,  $C_2 = 1/5$  and  $C_3 = 3/5$ ).

**Table 2.** Normalized Layer Distortion ( $\delta_i$ ) for Three Different Instances of the Distortion Model

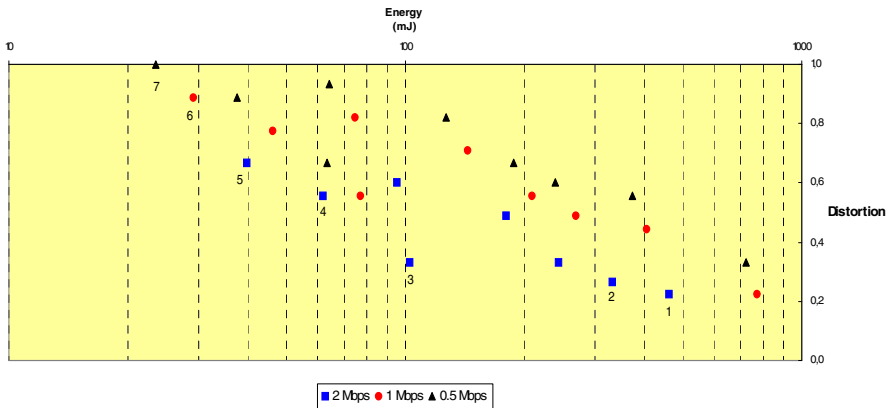
		$\delta_1$			$\delta_2$			$\delta_3$		
		0.5 Mbps	1 Mbps	2 Mbps	0.5 Mbps	1 Mbps	2 Mbps	0.5 Mbps	1 Mbps	2 Mbps
6 fps	SUB QCIF	1.00	0.89	0.67	1.00	0.93	0.80	1.00	0.80	0.40
	QCIF	0.93	0.82	0.60	0.96	0.89	0.76	0.96	0.76	0.36
	CIF	0.67	0.56	0.33	0.80	0.73	0.60	0.80	0.60	0.20
12 fps	SUB QCIF	0.89	0.78	0.56	0.80	0.73	0.60	0.93	0.73	0.33
	QCIF	0.82	0.71	0.49	0.76	0.69	0.56	0.89	0.69	0.29
	CIF	0.56	0.44	0.22	0.60	0.53	0.40	0.73	0.53	0.13
24 fps	SUB QCIF	0.67	0.56	0.33	0.40	0.33	0.20	0.80	0.60	0.20
	QCIF	0.60	0.49	0.27	0.36	0.29	0.16	0.76	0.56	0.16
	CIF	0.33	0.22	0.00	0.20	0.13	0.00	0.60	0.40	0.00

As presented in section 3.3, the normalized layer distortion,  $\delta(L)$ , is a relative measurement. Effectively, in each of the above model instances, the distortion varies from a minimum distortion ( $\delta = 0.00$ ) for the (CIF, 24 fps, 2.0 Mbps) layer to a maximum one ( $\delta = 1.00$ ) for the (SUB QCIF, 6 fps, 0.5 Mbps) layer. The distortion distribution of the rest of the layers is a function of the importance given to each of the interlayer distance components. Fig. 6 presents the distortion maps of the instances. As can be seen in Fig. 6(b), when the temporal scalability component is emphasized the map regions trend to be horizontally distributed, with the maximum distortion region located around the layers with the smallest frame-rate. On the other hand, as illustrated in Fig. 6(c), in case the quality scalability component is given more prominence, the tendency changes and the map regions are distributed vertically. Fig. 6(a) provides a scenario in which map regions are positioned between the horizontal ( $\delta_2$ ) and vertical distribution ( $\delta_3$ ).



**Fig. 6.** Normalized Layer Distortion Maps for Three Model Instances (a)  $\delta_1$  ( $C_1 = 1/3$ ,  $C_2 = 1/3$  and  $C_3 = 1/3$ ) (b)  $\delta_2$  ( $C_1 = 1/3$ ,  $C_2 = 3/5$  and  $C_3 = 1/3$ ) (c)  $\delta_3$  ( $C_1 = 1/3$ ,  $C_2 = 1/3$  and  $C_3 = 3/5$ )

As defined in section 1, a Pareto-optimum system control point is found when a decrease in active energy consumption is only achieved at the expense of a distortion increase or, conversely, when further distortion reduction is only realized with a simultaneous increase in active energy consumption. In order to estimate the Pareto frontier for the *foreman* test sequences, Fig. 7 plots the active energy consumption against the normalized layer distortion for the  $\delta_1$  model instances. It goes without saying that layer (CIF, 24 fps, 2 Mbps) has not been depicted in Fig. 7.



**Fig. 7.** Active Energy vs. Distortion Plots of the Foreman Test Sequences for the Model Instance  $\delta_1$  ( $C_1 = 1/3$ ,  $C_2 = 1/3$  and  $C_3 = 1/3$ )

The Pareto-optimum system control points, in increasing distortion order, are shown in Table 3. Layers marked with a dash do not belong to the Pareto frontier. As can be seen in Fig. 7, each of the Pareto-optimum system control points corresponds to a point (with the same order number) in the optimization space.

**Table 3.** Pareto-Optimum System Control Points Ordered in Increasing Distortion for the Model Instance  $\delta_1$  ( $C_1 = 1/3$ ,  $C_2 = 1/3$  and  $C_3 = 1/3$ )

		0.5 Mbps	1 Mbps	2 Mbps
6 fps	SUB QCIF	7	6	5
	QCIF	-	-	-
	CIF	-	-	-
12 fps	SUB QCIF	-	-	4
	QCIF	-	-	-
	CIF	-	-	1
24 fps	SUB QCIF	-	-	3
	QCIF	-	-	2
	CIF	-	-	-

For the  $\delta_1$  model instance, Table 3 shows that the optimal path from the minimum to the maximum distortion layer remains within the 2 Mbps layers until the minimum frame-rate and frame-size is achieved. Then, the (SUB QCIF, 6 fps, 1 Mbps) and (SUB QCIF, 6 fps, 0.5 Mbps) are consecutively selected. In fact, different instances of the distortion model results in different optimal paths.

## 5 Conclusions and Future Work

This paper describes an OMAP-based Pareto frontier estimate of a video decoder embedded in a multimedia terminal within the distortion-energy optimization space. To find the Pareto-optimum system control points that minimize the distortion of the decoded sequence and the decoder energy consumption, a test-bench based on the OMAP processor and the Open SVC Decoder has been implemented. Estimates of the active energy needed to process the test sequences have been obtained from the ratio of the average number of clock cycles per second to the operational frequency. In addition, a distortion model to fit subjective quality measurements has been proposed. The results have shown that frame size differences account for most of the active energy change to decode the test sequences. Furthermore, distortion map region distributions are aligned to the emphasis given to the scalability component of the interlayer distance employed as distortion metric. At last, different instances of the distortion model results in different optimal paths or Pareto frontiers from the minimum to the maximum distortion layer.

In near future the work will be focused in two lines. The former consists in the design of an OMAP-based embedded system to measure the energy consumption in real time to validate the model estimates while the latter will concentrate in fitting the scale factors of the distortion model to subjective quality measurements.

**Acknowledgments.** The authors would like to thank David Samper and Ernesto Seisdedos from Grupo de Diseño Electrónico y Microelectrónico (UPM) and Mickaël Raulet and Mederic Blester from IETR/Image Group Lab for their contributions to this work.

## References

- [1] Pentikousis, K.: In search of Energy-Efficient Mobile Networking. *IEEE Communications Magazine* 48(1), 95–103 (2010)
- [2] Hall, P.J., Bain, E.J.: Energy-Storage Technologies and Electricity Generation. *Energy Policy* 36, 4352–4355 (2008)
- [3] Jejurikar, R., Gupta, R.: Dynamic Voltage Scaling for Systemwide Energy Minimization in Real-time Embedded Systems. In: *ISLPED 2004*, pp. 78–81 (August 2004)
- [4] Reason, J.M., Rabaey, J.M.: A Study of Energy Consumption and Reliability in a Multi-Hop Sensor Network. *ACM SIGMOBILE Mobile Computing and Communications Review* 8(1), 84–97 (2004)
- [5] Park, C., Liu, J., Chou, P.: Eco: an Ultra-Compact Low Power Wireless Sensor Node for Real-time Motion Monitoring. In: *Proceeding of the 4th International Symposium on Information Processing in Sensor Networks* (April 2005)
- [6] Zamora, N.H., Kao, J.-C., Marculescu, R.: Distributed Power-Management Techniques for Wireless Network Video Systems. In: *DATE 2007* (April 2007)
- [7] Wu, W., Arefin, A., Rivas, R., Nahrstedt, K., Sheppard, R., Yang, Z.: Quality of Experience in Distributed Interactive Multimedia Environments: Toward a Theoretical Framework. In: *Proceedings of the Seventeenth ACM International Conference on Multimedia*, pp. 481–490 (October 2009)
- [8] Eberle, W., Bougard, B., Pollin, S., Catthoor, F.: From Myth to Methodology: Cross-Layer Design for Energy-Efficient Wireless Communication. In: *Proceedings of the 42nd Annual Design Automation Conference (DAC)*, pp. 303–308 (June 2005)
- [9] Ji, X., Pollin, S., Lafruit, G., Moccagatta, I., Dejonghe, A., Catthoor, F.: Energy-Efficient Bandwidth Allocation for Multiuser Scalable Video Streaming over WLAN. *EURASIP Journal on Wireless Communications and Networking* 2008, Article ID 219570, 14 pages (2008)
- [10] Miettinen, K.M.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers (1999) ISBN 978-0-792-38278-1
- [11] Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): *Multiobjective Optimization*. LNCS, vol. 5252. Springer, Heidelberg (2008) ISBN 978-3-540-88907-6
- [12] Neuvo, Y.: Cellular Phones as Embedded Systems. In: *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 32–37 (February 2004)
- [13] Texas Instruments. OMAP DSPs, <http://focus.ti.com/docs/prod/folders/print/omap3530.html>
- [14] ISO/IEC 14496-10. Information technology. Coding of audio-visual objects. Part 10: Advanced Video Coding (2008)

- [15] Schwarz, H., Marple, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 17(9), 1003–1120 (2007)
- [16] Blestel, M., Raulet, M.: The Open SVC Decoder project. In: *ACM Multimedia 2009, Open Source Software Competition Program* (2009)
- [17] Scalim@ges project, <http://www.images-et-reseaux.com/en/les-projets/fiche-projets-finances.php?id=125>
- [18] SVC4QoE, <http://www.images-et-reseaux.com/en/les-projets/fiche-projets-finances.php?id=203>
- [19] ScalNet, <http://www.scalnet.info/system/web/default.aspx>
- [20] Simple Direct Media Layer (SDL), <http://www.libsdl.org/>
- [21] Joint Scalable Video Model JSVM-9.9, Available in CVS repository at Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen
- [22] A ARM, Cortex-A8 Technical Reference Manual, ARM DDI 0344J rev.: r3p2, [http://infocenter.arm.com/help/topic/com.arm.doc.ddi0344j/DDI0344J\\_cortex\\_a8\\_r3p2\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0344j/DDI0344J_cortex_a8_r3p2_trm.pdf)
- [23] Texas Instruments, TMS320C64x/C64x+ DSP CPU and Instruction Set, SPRU732H (October 2008), <http://focus.ti.com/lit/ug/spru732h/spru732h.pdf>
- [24] BeagleBoard System Reference Manual Rev. C4 (December 2009), [http://beagleboard.org/static/BBSRM\\_latest.pdf](http://beagleboard.org/static/BBSRM_latest.pdf)
- [25] The Perl Programming Language, <http://www.perl.org/>
- [26] Using the Scripting Utility in the Code Composer Studio IDE, <http://www.ti.com/libv/pdf/spra383a.pdf>
- [27] Texas Instruments. TMS320 DSP BIOS User's guide (SPRU303B – May 2000), <http://focus.ti.com/lit/ug/spru303b/spru303b.pdf>
- [28] Pescador, F., Sanz, C., Garrido, M.J., Juárez, E., Samper, D.: A DSP Based H.264 Decoder for a Multi-Format IP Set-Top Box. *IEEE Transactions on Consumer Electronics* 54(1), 145–153 (2008)
- [29] Mainconcept SVC Scaleble Video Coding, <http://www.mainconcept.com/site/developer-products-6/pc-based-sdks-20974/svc-tech-preview-22033/information-2036.html>
- [30] Texas Instruments. OMAP3530 Power Consumption Summary. SPRAB98 (January 2010), <http://focus.ti.com/lit/an/sprab98/sprab98.pdf>