A Hardware Entropy Decoder for Scalable Video

Hendrik Eeckhaut

Supervisor(s): Dirk Stroobandt, Jan Van Campenhout

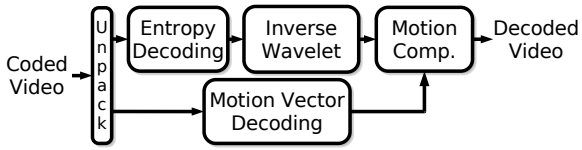

Fig. 1. High-level overview of the video decoder

Abstract—**In the RESUME project (Reconfigurable Embedded Systems for Use in Multimedia Environments) we explore the benefits of an implementation of scalable multimedia applications using reconfigurable hardware by building an FPGA implementation of a scalable wavelet-based video decoder.**

In this article we present the results of our investigation into the hardware implementation of such a scalable video codec. In particular we found that the implementation of the entropy codec is a significant bottleneck. We present an alternative, hardware-friendly algorithm for entropy coding with superior data locality (both temporal and spatial), streaming capabilities, a high degree of parallelism, a small memory footprint and superior compression while maintaining all required scalability properties.

These claims are supported with a full fledged hardware (FPGA) implementation. A really compact implementation was developed which can easily decode the required 50 million symbols per second.

Keywords— **Reconfigurable hardware, scalable video, wavelet entropy coding**

I. Introduction

SCALABLE VIDEO is encoded in such a way that it allows to easily change the Quality of Service (QoS) i.e. the frame rate, resolution, color depth and image quality of the decoded video, without having to change the video stream used by the decoder (except for skipping unnecessary blocks of data without decoding) or without having to decode the whole video stream if only a part of it is required.

The internal structure of a scalable decoder is shown in Figure 1 and was described in [1], [2]. In this paper we focus on the Wavelet Entropy Decoder (WED). In this part of the algorithm encoded data is decompressed into wavelet frames. It consists of two parts: the *Model Selector* (MS) and the *Arithmetic Decoder* (AD). The MS provides the AD with continuous guidance about what type of data is to be decoded by selecting an appropriate statistical model for the symbol (a bit) that has to be decoded next. It exploits the correlation between neighboring wavelet coefficients in different contexts. Finally the AD performs the actual decompression of the symbol stream.

The difference with conventional arithmetic decoders is that this WED supports quality scalability. I.e. the decoder can choose how many bits it decodes; the more decoded bits, the better the quality. Quality scalability is achieved through encoding the wavelet frame bitlayer per bitlayer (from top to bottom), yielding progressive accuracy of the wavelet coefficients (Figure 2).

H. Eeckhaut is with the Parallel Information Systems (PARIS) group, Department of Electronics and Information Systems, Ghent University (UGent), Gent, Belgium. E-mail: Hendrik.Eeckhaut@Elis.UGent.be .

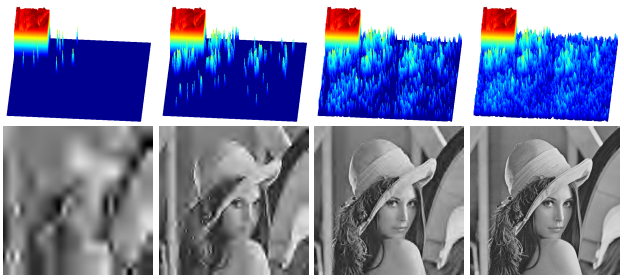

Fig. 2. Quality scalability: decoding more bitlayers gives a more accurate wavelet transformed frame.

Conventional video codecs don't offer quality scalability. Quantization is performed at encoding time and all (DCT) coefficients are encoded as a whole. In this new (scalable) approach all bits of the non-quantized (wavelet) coefficients are processed separately. This implies that every context model has to be calculated per bit instead of per coefficient. So the pressure on the Model Selector is much higher. The new approach also has to decode more bits. We start decoding every bit from the highest bitlayer, in which only the largest coefficients are non-zero. So we have to process the same number of bitlayers for the largest as for the smallest coefficients (which are often zero). This results in a larger total number of bits that the arithmetic decoder has to process for the same quality settings. In conclusion the actual question is whether the bitlayer approach for quality scalability is practically feasible.

II. A hardware-friendly scalable WED

In [3] we proposed a new algorithm for entropy coding which fully supports the scalability of the presented video codec. It offers quality scalability by encoding the wavelet image bitlayer by bitlayer and resolution scalability by encoding data from the different resolution layers independently. It is really economical with memory, both in memory footprint and in required bandwidth. Moreover, despite the sequential nature of entropy coding algorithms, it supports a high degree of parallelism on the subband level. The algorithm was designed for simplicity to stimulate an elegant implementation. Finally, it gives state-of-the-art compression results.

This new algorithm encodes (and decodes) all subbands of the wavelet transformed channel independently. Therefore it is possible to process all subbands of the wavelet transformed frame in parallel. The subbands are processed bitlayer per bitlayer from top to bottom. The top is the bitlayer that contains the most significant bit of the largest absolute value of all coefficients and

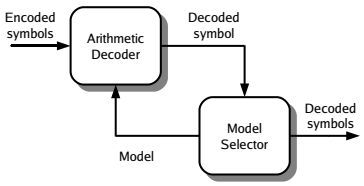

Fig. 3. The critical feedback loop in the entropy decoder: The model selector has to provide the arithmetic decoder with a new model that depends on the output of the arithmetic decoder.

the bottom is the bitlayer containing the least significant bits. The bitlayers are processed in scanline order. This greatly benefits the memory accesses, since this is the order in which the data is stored in memory. It also enables us to stream data and use burst mode features of slower memories. All data from one subband is processed sequentially since all bits are now encoded based on information of previously encoded bits.

As mentioned in the Introduction, symbols are encoded with different models depending on their contexts to exploit statistical characteristics, e.g. the fact that pixels become significant in clusters. For optimal compression, storing all information about previously processed data would be ideal but since this excludes an efficient hardware implementation only the most relevant information is stored. Our algorithm limits this information to the sign and the significance of each coefficient. This information can easily be organized as two bitmaps with dimensions equal to the subband's. From these bitmaps the number of significant (or negative) neighbors of the current coefficient are counted to determine the context model.

III. Hardware implementation

Entropy coding is a very sequential process by nature. Its purpose is to exploit statistical redundancy in the wavelet frames. Therefore it has to take as much information as possible about previously coded symbols into account to achieve optimal compression. This results in a critical feedback loop in the algorithm of the arithmetic decoder (Figure 3). Both the model and result of the currently decoded bit depend on the result of the previously decoded bit.

If we want to speed up the entropy decoding with a hardware implementation we have to squeeze every drop of possible parallelism out of the algorithm. Through advanced pipelining and speculative execution and data fetching, we managed to develop a decoder that is able to decode one symbol per clock cycle.

The entire architecture of the complete Wavelet Entropy Decoder is illustrated in Figure 4. This design was implemented in VHDL, co-simulated in a System-C testbench and synthesized for an Altera Stratix S25 (speed grade 5). The results are summarized in Table I. The one DSP multiplier block is used for calculating the size of the sign and significance bitmaps ($rows \times cols$) for the ModelSelector and could easily be avoided.

The entire design works for clock rates up to 52 MHz. This is fast enough to decode lossless CIF sequences with one decoder. The implementation is also very compact, less than 10% of the available resources are used. Moreover only 5% of the available memory is used, of which about 70% is needed to store the two bitmaps which are large enough to support CIF frames. Larger

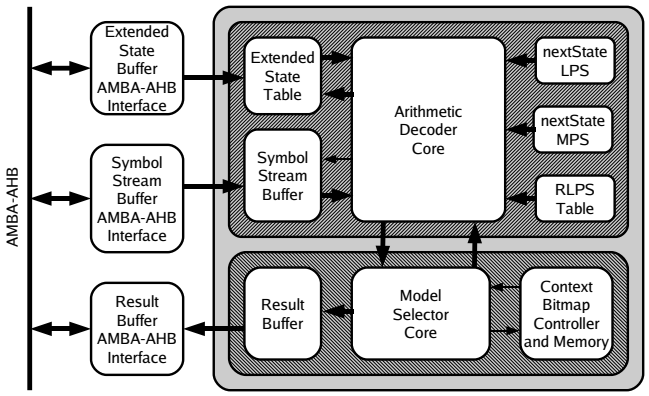

Fig. 4. Architecture of the Wavelet Entropy Decoder.

TABLE I
Synthesis results for an Altera Stratix S25.

Block	LEs	Memory bits	Clock (MHz)
Arithmetic Decoder	1169	32640	53.53
Core	691	0	60.18
SymbolStreamBuffer	394	1024	218.72
ROMs	2	18432	300
ExtendStateTable	0	4992	300
ExtendStateInputBuf.	0	8192	300
ModelSelector	869	68608	89.38
BitmapController	181	66560	166.67
ResultBuffer	143	2048	214.18
WED	2038	101248	53.53

resolutions are possible by simply changing some design parameters. The targeted FPGA is clearly large enough to contain multiple instantiations of our decoder.

IV. Conclusions

In this paper we demonstrated that the bitlayer approach to offer quality scalability is indeed feasible. We presented a small and fast implementation of a scalable Wavelet Entropy Decoder, which is able to produce more than 50 million decoded bits per second. This is sufficient for decoding lossless CIF sequences. Larger resolutions are possible since the design is small enough to allow multiple parallel instantiations.

Acknowledgments

This research is supported by I.W.T. grant 020174, F.W.O. grant G.0021.03, GOA project 12.51B.02 of Ghent University and by the Altera University Program.

References

[1] A. Munteanu, *Wavelet Image Coding and Multiscale Edge Detection - Algorithms and Applications*, Ph.D. thesis, Vrije Universiteit Brussel, 2003.

[2] D. Stroobandt, H. Eeckhaut, H. Devos, M. Christiaens, F. Verdicchio, and P. Schelkens, "Reconfigurable hardware for a scalable wavelet video decoder and its performance requirements," *Computer Systems: Architectures, Modeling, and Simulation*, vol. 3133, pp. 203–212, July 2004.

[3] Hendrik Eeckhaut, Harald Devos, Benjamin Schrauwen, Mark Christiaens, and Dirk Stroobandt, "A hardware-friendly wavelet entropy codec for scalable video," in *DATE 2005 Designers' Forum Proceedings*, March 2005, pp. 14–19.