SCALABLE VIDEO CODING BASED ON MOTION-COMPENSATED TEMPORAL FILTERING: COMPLEXITY AND FUNCTIONALITY ANALYSIS

Fabio Verdicchio[*]*, Yiannis Andreopoulos, Tom Clerckx, Joeri Barbarien, Adrian Munteanu, Jan Cornelis and Peter Schelkens*

Vrije Universiteit Brussel (VUB)
Department of Electronics and Information Processing (ETRO-IRIS)
Pleinlaan 2, B-1050, Brussels, Belgium

[*]E-mail: fverdicc@etro.vub.ac.be

ABSTRACT

Video coding techniques yielding state-of-the-art compression performance require large amount of computational resources, hence practical implementations, which target a broad market, often tend to trade-off coding efficiency and flexibility for reduced complexity. Scalable video coding instead, not only provides seamless adaptation to bit-rate variation, but also allows the end user to trim down the resources he needs to perform real-time decoding by limiting the process to a subset of the original content. Hence, by choosing the quality, frame-rate and/or resolution of the reconstructed sequence, each decoder can meet its hardware limitations without affecting the encoding process of the media provider. This paper proposes a preliminary analysis of the memory-access behavior of a fully scalable video decoder and investigates the capability of selecting the operational settings in order to adapt to the available hardware resources on the target device.

1. INTRODUCTION

Streaming of multimedia content over heterogeneous networks, e.g. the Internet, where a variety of end-users may request the same material while experiencing different available bandwidths, is the natural environment for scalable video coding (SVC). The media provider, using SVC techniques, generates a single compressed bit-stream, from which appropriate subsets, providing different visual quality, frame-rate and resolution, can be extracted to meet the bit-rate requirements of a broad range of clients without the necessity for a low-level transcoding (i.e. full decoding and re-encoding). In case of SVC solely, the usage of a code-stream parser will suffice.

Previous works [1] have focused on standardized non-scalable video codecs, such as MPEG4-AVC, and analyzed the application under the *data transfer and storage* perspective, hence measuring complexity in terms of access frequency along with execution time. The authors of [1] profiled both encoder and decoder and reported the tradeoff between coding performance and complexity of a number of possible configurations supported by the standard and suggested an a-priori selection of the tools employed at encoding time, as a way to control complexity. Once a set of features is retained and the algorithm is fully specified, source code transformation and data flow optimization techniques [2] can be applied to achieve significant speedups of the application and/or reduction of the resources required, such as memory and clock frequency, thus decreasing power consumption.

The main contribution of this paper is to provide initial insights in the requirements of the fully scalable coding scheme proposed in [3] and report evidence of its capability to steer the computational and memory complexity of the decoder by simply adjusting client settings (frame-rate, resolution, and quality) independently of the encoding process. We believe this to be a necessary kick-off step of a broader feasibility study, targeting efficient implementations of fully scalable video coding methods, which recently raised the attention of the MPEG-committee.

The remainder of this paper is structured as follows: in section 2 we give a brief survey of motion compensated temporal filtering (MCTF) video coding scheme. In section 3 we describe the detailed configuration used to perform the experiments, while the results are reported and commented in section 4. Conclusions are drawn in section 5.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

2. SPATIAL DOMAIN MOTION COMPENSATED TEMPORAL FILTERING

The basic architecture analyzed in this paper is the open-loop scheme depicted in Figure 1, which performs a temporal and then a spatial decomposition (T+2D) of the video prior to embedded compression [4] thus alleviating the closed prediction loop used in conventional hybrid-coding schemes.

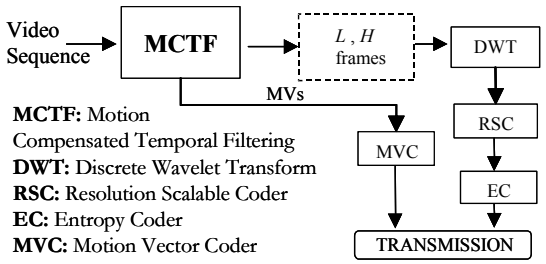


Figure 1. Spatial-domain (SD) MCTF encoder architecture

The core of the encoding procedure is the initial removal of temporal correlation within the sequence (MCTF); this task is basically accomplished by a low and high-pass filtering of the input frames along the temporal direction. To achieve efficient decorrelation, the input samples need to be aligned along the motion trajectories [5]. Hence two additional stages are incorporated into temporal filtering: first a motion estimation step (ME) determines the displacement information (i.e. the motion vectors (MV)), identifying the corresponding blocks in the temporal direction. In the motion compensation phase (MC) each macroblock is positioned according to the corresponding MVs before temporal filtering is performed. MCTF is efficiently implemented using the lifting framework [6], as depicted in Figure 2.

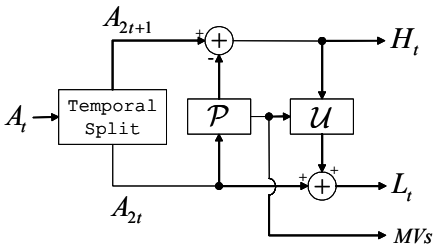


Figure 2. MCTF: one level decomposition with lifting.

The $\mathcal{P}$ operator generates a prediction of the each odd frame (current frame) based upon the even frames (reference frames) using the above-mentioned ME/MC stages. The output of $\mathcal{P}$ is then subtracted from the current frame to obtain the residual error frame (high-pass temporal frame) or *H-frame*. This information is also added back to the reference frame by the $\mathcal{U}$ operator, which performs an additional MC stage using the reversed motion field, to generate a set of *L-frames* (or low-pass temporal frames). These frames represent a temporally smooth version of the input sequence, sampled at half of the original frame-rate. The temporal filtering process continues recursively to obtain subsequent temporal levels.

Finally, as shown in Figure 1, a 2D discrete wavelet transform (DWT) is performed on the MCTF output, yielding a multiresolution spatio-temporal representation of the input sequence. The result of a three-level decomposition performed both in the temporal and spatial direction is illustrated in Figure 3. Each wavelet frame is subsequently encoded using an embedded intra-band compression algorithm [7], which progressively encodes the coefficients of each spatial resolution, enabling the video reconstruction at a dyadic set of resolution levels.

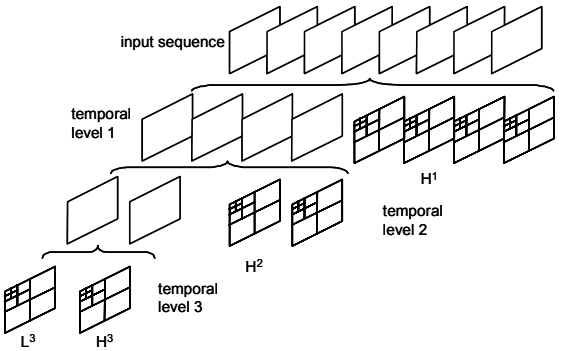


Figure 3. Output of three-level T+2D decomposition.

The decoding process basically consists of the inverse encoding steps. Due to embedded encoding, the user receives the data belonging to those temporal or spatial levels that are necessary to reach the requested frame-rate or spatial-resolution, thus saving bandwidth and computations. Moreover, due to embedded coding, the visual quality of the decoded video sequence can be progressively refined for any target resolution-level or frame-rate. Hence, the user can trade-off the overall visual quality for bandwidth and computational cost.

3. SELECTED CONFIGURATION AND EXPERIMENTAL SETTINGS

Since multimedia application such as video coding are clearly identified as data dominated applications, the complexity metric used in this paper as main indicator is memory access frequency, i.e. the number of accesses to the memory the application has to perform each second in order to operate in real time. A thorough study of the typical size of the necessary memory buffers for scalable MCTF-based codecs (see e.g. [8]) is deferred to a later stage, in which the possible exploitation of multi-hierarchical memory architectures will be investigated. To evaluate the complexity of the decoder in the scalable

framework, we select a single set of features and trim scalable parameters in the given range. Thereafter we use the ATOMIUM Analyzer tool [9] to measure for each setting the access rate required to achieve real-time decoding. Experiments are carried out using 3 sequences "Bus", "Canoa" and "Football", which have the same spatial resolution (CIF) and frame rate (30 fps). These sequences are considered to typically contain medium to complex motion scenes. Given the original frame-rate and resolution, we limit our analysis to video reconstruction at full or half-frame-rate, and full or half resolution as well.

3.1. Temporal filtering

With respect to the general scheme of Figure 2 we disable the $\mathcal{U}$ operator hence $L_t^i \equiv A_{2^i \cdot t}$ at any temporal level i, where A_t is the input sequence. This choice is motivated by the presence of visual artifacts in the updated *L-frames*. Thus, to preserve visual quality when a lower frame-rate is targeted, we choose to sacrifice the performance gain provided by the update step. On the other hand this choice yields a reduction in the complexity of the decoder (MC stages are halved) and more important, allows for parallel reconstruction of different temporal levels. In fact, since $L_t^i \equiv L_{2^j \cdot t}^{i-j}$ with $j>0$, once a frame has been compensated and corrected, it can be immediately used as a reference in the reconstruction of any upper temporal level. In all experiments, we use 4 temporal levels, and the sequence is decoded at 30 fps or at 15 fps when data from level 1 is not received / processed. No attempt is made to replace the dropped frames using temporal interpolation.

3.2. Motion Compensation

Motion estimation (ME) – performed only at encoding time – and MC employ block-based techniques with fractional pixel accuracy (1/4 pel in our tests), whose efficiency is enhanced using multi-hypothesis and bidirectional prediction with multiple references. Additionally, each macroblock (MB) can be split into 4 children blocks predicted independently. The splitting can be iteratively refined, but we restrict the experiment to MBs of 16x16 or 8x8 pixels. When the resolution is halved, the decoder reconstructs both the error-frames and the reference-frames at half resolution, and MC occurs using properly scaled motion vectors, achieving approximately a reduction of the memory accesses by a factor of four. This reduction needs to be evaluated experimentally since samples predicted at full resolution without interpolation (MB with integer displacement) may require interpolation when the displacement is halved, thus causing some additional operations. As for the frame-rate, no attempt is made to interpolate frames decoded at lower resolutions either. In fact, due to the use of orthogonal or bi-orthogonal wavelet filters, the information that is absent when a DWT level is not decoded cannot be recovered via trivial interpolation.

3.3. Motion vector coding

The MV coding engine used for these experiments does not support quality scalability; savings in accesses, then, only come when a temporal level is dropped. We did not profile this component.

3.4. Wavelet Transform

Literature conveys a large amount of 2D DWT related implementation research, (e.g. [10]). The spatial wavelet engine is not the focus of our profiling experiments and is not discussed in the next section. However, this codec component also benefits from the scalable approach, as its memory cost scales with the temporal and spatial scaling during decoding.

3.5. Texture Coding

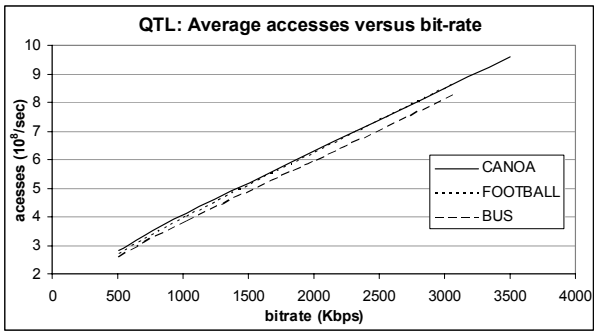
The wavelet coefficients of each resolution level are compressed in an embedded manner using the QT-L algorithm of [7]. This component is our main focus in the profiling stage, because its behavior is affected by each operational parameter (quality, resolution or frame-rate), as various spatial and temporal levels contribute differently to the compressed stream, hence to the amount of operations the decoder performs at any target bit-rate.

4. MEMORY PROFILING RESULTS

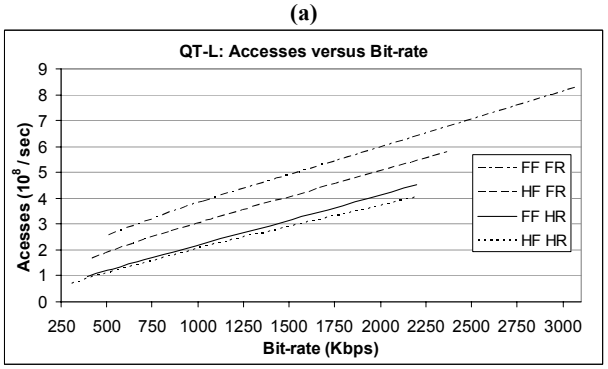
Figure 4 (a) reports the access rate caused by the sole QT-L module when decoding three different sequences at original frame-rate and resolution for a wide set of bit-rates. It is noticeable that the access rate is approximately linearly depending on the target bit-rate, and that this behavior is not strongly content-dependant. Notice that a similar linear behavior is observed by decoding to lower-resolutions and/or frame-rates, as shown in Figure 4 (b) for the "Bus" sequence.

From the access-rate perspective, the option providing the largest gain is resolution scaling: accesses performed by QT-L are significantly reduced, especially at high bit-rates, while those caused by the MC should diminish, independently of the bit-rate, to 25% of the amount needed at full resolution. With respect to the later, experiments report a figure of 33.2%, confirming the expected overhead due to MV scaling and additional interpolations. These results show that the access rate can be decreased in a fine grain manner by varying the target bit-rate until real-time decoding is achieved. To avoid reducing excessively the visual quality, the user can switch to a less demanding configuration, decoding lower-resolution versions of the input video, or fewer frames, as shown in Figure 4 (c). Thus, when one resolution or temporal level (or both) are not processed, the decoder switches to an operational

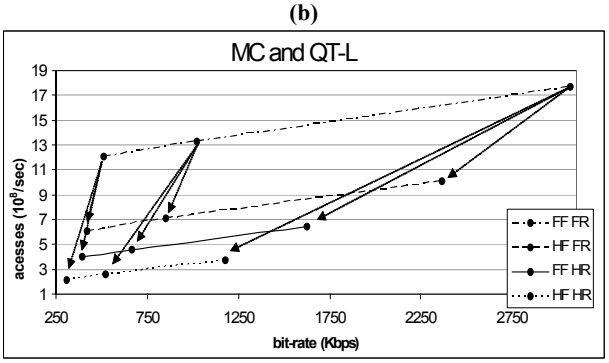
point which lays on a curve positioned below the one corresponding to full-resolution and frame-rate decoding.



(a)



(b)



(c)

Figure 4. QT-L access rate for (a) three different sequences, and (b) the "Bus" sequence, decoded at full/half frame-rate (FF/HF), and full/half resolution (FR/HR). (c) Access rates obtained for both QT-L and MC components on the "Bus" sequence.

These results show also that the relationship between the access rate and target bit-rate for decoding at different resolutions and frame-rates can be "learned" using appropriate training on large datasets. In this way, the decoder can estimate the optimum bit-rate, given a maximum memory-access rate, and user-specified resolution and frame-rate. Conversely, for bandwidth-limited applications, the decoder can estimate the optimum operational settings in terms of resolution and frame-rate, for given access rate and channel-bandwidth.

5. CONCLUSIONS

This paper proposes a preliminary analysis of the memory-access behavior of a fully scalable video decoder and investigates the impact of varying the operational settings on the memory-access rate. It is shown that by choosing the quality, frame-rate and/or resolution of the reconstructed sequence, each decoder can meet its hardware limitations without requiring transcoding, hence affecting the encoding process of the media provider.

6. ACKNOWLEDGMENTS

This work was supported by the Federal Office for Scientific, Technical and Cultural Affairs (IAP Phase V - Mobile Multimedia). P. Schelkens has a post-doctoral fellowship with the Fund for Scientific Research – Flanders (FWO), Egmontstraat 5, B-1000 Brussels, Belgium.

7. REFERENCES

[1] S. Saponara, C. Blanch, K. Denolf, J. Bormans, "The JVT Advanced Video Coding Standard: Complexity and Performance Analysis on a Tool-by-Tool Basis," *IEEE Workshop on Packet Video (PV'03)*, Nantes, France, April 2003.

[2] K. Denolf, P. Vos, J. Bormans, and I. Bolsens, "Cost-efficient C-Level Design of an MPEG-4 Video Decoder," *Workshop on Power and Timing Modeling, Optimization and Simulation,* Goettingen, Germany, Sept. 2000.

[3] I. Andreopoulos, J. Barbarien, F. Verdicchio, A. Munteanu, M. van der Schaar, J. Cornelis, and P. Schelkens, "Response to Call for Evidence on Scalable Video Coding," ISO/IEC JTC1/SC29/WG11, M9911, Trondheim, Norway, July 2003.

[4] J. R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 559-571, Sept. 1994.

[5] B. Girod, "The efficiency of motion-compensated prediction for hybrid video coding of video sequences," *IEEE J. Select. Areas Commun*., vol. SAC-5, pp. 1140-1154, Aug. 1987.

[6] M. Flierl and B. Girod "Video Coding with Motion-Compensated Lifted Wavelet Transforms," *J. Image Comm.*, Special Issue on Subband/Wavelet Video Coding, submitted.

[7] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. Medical Imaging*, vol. 22, no. 3, pp. 441-458, March 2003.

[8] H. Devos, H. Eeckhaut, M. Christiaens, F. Verdicchio, D. Stroobandt, and P. Schelkens, "Performance requirements for reconfigurable hardware for a scalable wavelet video decoder," *Proc. of IEEE ProRiSC 2003*, Veldhoven, The Netherlands, pp. 56-63, Nov. 2003.

[9] http://www.imec.be/atomium.

[10] Y. Andreopoulos, P. Schelkens, G. Lafruit, K. Masselos, J. Cornelis, "High-level cache modeling for 2-D discrete wavelet transform implementations," *VLSI Signal Proc. Systems*, vol. 34, no. 3, pp. 209-226, July 2003.