UNIVERSITEIT
GENT

biblio.ugent.be

The UGent Institutional Repository is the electronic archiving and dissemination platform for all UGent research publications. Ghent University has implemented a mandate stipulating that all academic publications of UGent researchers should be deposited and archived in this repository. Except for items where current copyright restrictions apply, these papers are available in Open Access.

This item is the archived peer-reviewed author-version of:

Title: Region-Adaptive Probability Model Selection for the Arithmetic Coding of Video Texture

Authors: K. Vermeirsch, J. Barbarien, P. Lambert, and R. Van de Walle

In: Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 1537 - 1540, May 2011.

To refer to or to cite this work, please use the citation to the published version:

K. Vermeirsch, J. Barbarien, P. Lambert, and R. Van de Walle (2011). Region-Adaptive Probability Model Selection for the Arithmetic Coding of Video Texture. *. International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 1537-1540.

REGION-ADAPTIVE PROBABILITY MODEL SELECTION FOR THE ARITHMETIC CODING OF VIDEO TEXTURE

K. Vermeirsch, J. Barbarien, P. Lambert, and R. Van de Walle

K. Vermeirsch, P. Lambert, and R. Van de Walle are with the Department of
Electronics and Information Systems, Faculty of Engineering, Ghent University, Belgium.
J. Barbarien is with the Department of
Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Belgium.

ABSTRACT

In video coding systems using adaptive arithmetic coding to compress texture information, the employed symbol probability models need to be retrained every time the coding process moves into an area with different texture. To avoid this inefficiency, we propose to replace the probability models used in the original coder with multiple switchable sets of probability models. We determine the model set to use in each spatial region in an optimal manner, taking into account the additional signaling overhead. Experimental results show that this approach, when applied to H.264/AVC's context-based adaptive binary arithmetic coder (CABAC), yields significant bit-rate savings, which are comparable to or higher than those obtained using alternative improvements to CABAC previously proposed in the literature.

Index Terms— Video coding, CABAC, arithmetic coding, adaptivity, context modeling

1. INTRODUCTION

H.264/AVC [1], which represents the current state-of-the-art in video coding technology, is typically configured to use Context-based Adaptive Binary Arithmetic Coding (CABAC) [2] for the entropy coding of texture and motion information in application scenarios demanding optimal compression performance.

In the literature, improvements to CABAC have focused on a number of areas. Several authors [3, 4, 5] have proposed to replace the simple CABAC data modeling by the more sophisticated Context-Tree Weighting (CTW), reporting about 1 to 3 % bit-rate savings for typical quality settings, at the cost of a significant increase in computational complexity, as reported in [5]. Belyaev et al. [6] use a virtual sliding win-

dow (VSW) for probability model estimation and report rate savings between 0.3 and 0.8 % for realistic quality settings.

In this paper, we improve the efficiency of CABAC entropy coding by introducing the concept of region-adaptive probability model selection. This improvement is not only applicable to CABAC in H.264/AVC but can be adapted to any video or image coder, present or future, which employs adaptive probability models.

2. REGION-ADAPTIVE CODING METHOD

Typical video scenes comprise a number of distinct regions, each having unique texture characteristics, resulting in different symbol statistics for the quantized transform coefficients representing the texture information, even if the transform is preceded by a spatial or temporal prediction step. As the adaptive arithmetic coder visits the blocks of a video frame in the typical raster scan order, the boundaries between the different regions are often crossed multiple times. Each time such an event occurs, the coder's probability models for texture, which had been trained to the characteristics of the previous region, need to be retrained to the characteristics of the newly entered region. Before the model reconverges to the statistics of the symbols in the newly entered region, many symbols may have been coded suboptimally, resulting in a penalty in bit rate.

As mentioned, this retraining of the probability models occurs more often than necessary because of the raster block scanning order universally used in video coders. One solution is to remove this underlying cause, abandoning the raster scan, instead coding each image region or object in turn. This poses several practical problems: it requires an image segmentation process to be performed prior to encoding, and it causes unpredictable non-sequential access to the frame memory, making this solution hard to implement efficiently in either software or hardware.

The solution we propose, and which we call region-adaptive probability model selection, instead retains the raster scanning order of the coding process. The original

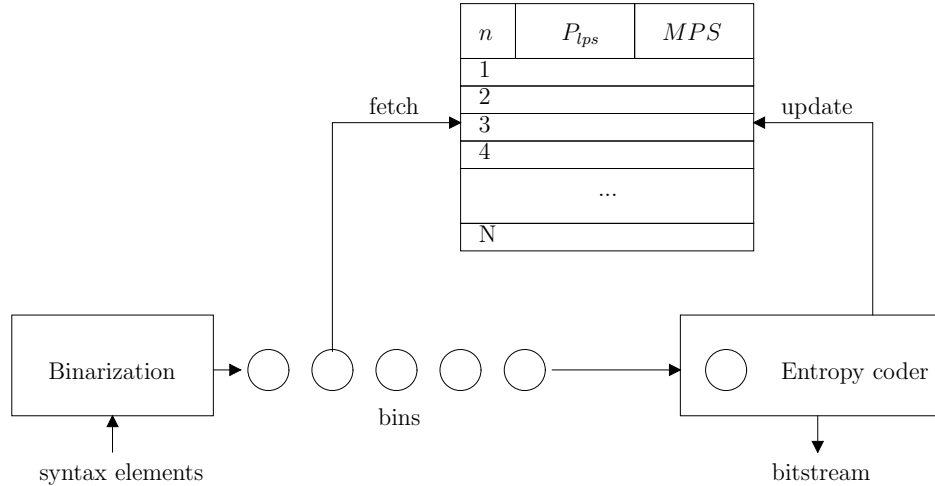| n	P_{lps}	MPS
1		
2		
3		
4		
...		
N		

Fig. 1. Schematic overview of CABAC.

set of probability models used to code texture information is replaced by several switchable sets of probability models. Each frame is then partitioned into square tiles, and one of the available sets of probability models is associated with each tile. Within a tile, only the active set of models is being trained and adapted; the other sets remain untouched, ready to be activated in other regions.

3. SUMMARY DESCRIPTION OF CABAC

Before giving a detailed description of our region-adaptive probability selection approach, a brief overview of CABAC is given. For a full description, the reader is referred to [2]. A schematic representation of the data flow in CABAC is presented in Figure 1.

Binarization: For reasons of computational complexity and uniformity, CABAC is based on an arithmetic coder, which only accepts binary input symbols. As a result, symbols which can take on more than two values need to be binarized first, i.e., they are converted to two or more binary subsymbols called bins prior to arithmetic coding. CABAC employs many different binarization schemes, including Exponential Golomb and unary binarization. The type of binarization used depends on the source of the symbols being coded and has been determined based on the typical probability distribution of the symbols generated by this source.

Bin model selection and bin model update: Each bin is assigned a binary probability model (bin model), and the state of this model is represented by a most probable symbol (MPS), which is a binary value, and the probability of the least probable symbol occurring. The model restricts to one of 64 discrete values, which implies that the state of a bin model can be represented using only 7 bits.

The assignment of a probability model to a bin is determined by the source of the symbols, the associated binariza-

tion scheme and, for some bins, by local context. For example, the model assignment for transform coefficients depends on the coding mode and on the position of the coefficient within the transform block. Consequently, there are many hundreds of bin models employed in CABAC.

The symbols composing the texture information in 4:2:0 progressive video sequences include flags indicating the positions of the nonzero transforms coefficients; and symbols representing the magnitudes of the nonzero coefficients; and their signs. The coding of these symbols involves 225 different bin models. This collection of probability models is, in the context of this paper, collectively called a *texture model*.

After a bin has been coded by the binary arithmetic coder, its associated bin model is updated. This is how the texture model adapts to the image region's nonstationary characteristics. If the bin value matches the model's MPS, then the MPS probability is increased by one step, ultimately converging to nearly one. If, however, the bin value is not equal to the MPS (but equal to the least probable symbol or LPS), the MPS probability is reduced toward 0.5 by multiple steps at once. When the MPS and LPS become equally probable, the MPS and LPS may be interchanged.

4. PROPOSED REGION-ADAPTIVE TEXTURE MODEL SELECTION

In the proposed method, the image is subdivided into square tiles of size S. Each tile may be coded using one of N texture models. We now describe the tile coding procedure, followed by the determination of the S and N parameters.

Tile coding procedure: At the start of a tile, the coder evaluates which is the best of the N texture models for coding the tile's syntax elements. For optimal compression efficiency, the N different texture models may be evaluated by effectively coding all syntax elements using each of the avail-
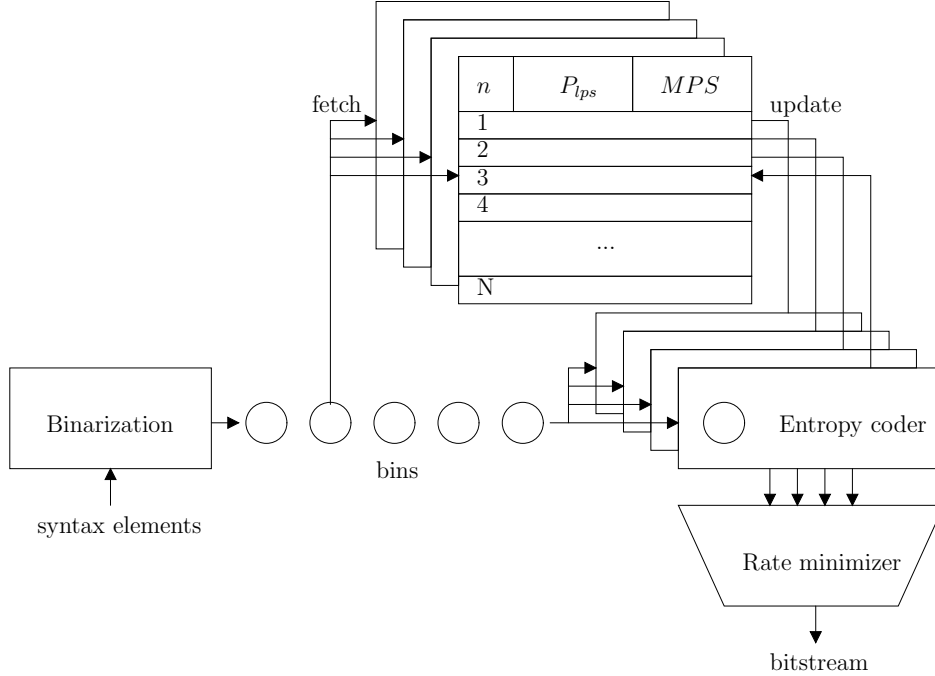
Fig. 2. Schematic overview of the proposed algorithm with exhaustive search.

able texture models and comparing the resulting bit-rates, as illustrated in Figure 2. The texture model yielding the smallest bit-rate is retained and all other texture models are reset to their previous states. Note that the time-consuming processes of binarization and bin model selection only need to be performed once, since their outcome does not depend on the chosen texture model. Furthermore, it should be observed that the effective encoding operations for different texture models can proceed in parallel, allowing for a computationally efficient implementation.

As a low-complexity alternative to this exhaustive approach, the coder can also determine the best texture model without actually coding N times, by counting, for each texture model, how many times a least-probable symbol would be encoded. This metric provides a reliable indication of bit-optimality, except in those cases where two or more texture models would lead to comparable coding efficiency, in which case a misprediction is of little consequence anyway.

Since only 7 bits are needed to represent a bin model, the additional memory requirements for encoder and decoder are quite modest: each texture model has a memory footprint of 197 bytes. When four texture models are used as in the experiments below, only 788 extra bytes are required.

Tile size selection: To ensure decodeability, a certain amount of overhead bits need to be transmitted to indicate which texture model was used in each tile. This overhead is signaled inside the bit stream using CABAC and truncated Exponential Golomb binarization. The amount of overhead bits increases for larger number of models N and smaller tile

sizes S. A balance thus needs to be struck between flexibility and signaling overhead. A straightforward approach to this problem would be to encode each frame using multiple values of S and N. This was deemed to be computationally prohibitive. Instead, S and N are determined heuristically based on an extensive set of experiments using different values for S and N while coding different sequences with varying resolutions and targeting different quality levels. These experiments have shown that tile sizes exceeding 16x16 luma samples do not provide sufficient granularity to adapt to the different regions, while 4x4 sample tiles represent the lower bound as it is the smallest transform size in the H.264/AVC standard. The following rules are used to determine S:

$$S = \begin{cases} 8 \times 8 \text{ if } N_{mb} \leq 396 \text{ and } QP \geq 28 \\ 16 \times 16 \text{ if } N_{mb} \leq 396 \text{ and } QP < 28 \\ 4 \times 4 \text{ if } N_{mb} > 396. \end{cases} \quad (1)$$

with N_{mb} denoting the number of macroblocks in the frame and QP denoting the average quantization parameter in the previous slice of the same type.

Our experiments also show that a value of $N = 4$ is universally optimal except in the case of low-quality HD content where it may be reduced to 3. N is thus selected as:

$$N = \begin{cases} 3 \text{ if } N_{mb} > 396 \text{ and } QP \geq 28 \\ 4 \text{ otherwise.} \end{cases} \quad (2)$$

Table 1. Bit rate savings for texture data of the proposed region-adaptive extension to CABAC compared to regular CABAC, per cent.

Type	Sequence name	Proposed vs. CABAC
CIF	bus	0.84
	football	1.34
	foreman	0.71
	mobile	1.06
	paris	1.59
	stefan	0.65
	table	1.49
	tempete	0.73
720p HD	big ships	4.54
	city	3.21
	crew	3.79
	harbour	3.89
	night	4.01

5. EXPERIMENTAL RESULTS

Table 1 presents bit savings, compared to classical CABAC, for a number of CIF (352×288 pixels) and 1280×720 resolution video sequences. The sequences are coded with I and P pictures and a GOP size of 8. Quality parameter (QP) settings are 20 for the CIF sequences and 30 for the 720p sequences. Bit savings for the CIF resolution test set measure up to 1.5 %. For the higher-resolution 720p sequences, particularly good results of 4 % on average are reported.

6. CONCLUSION

We presented a region-adaptive arithmetic coding method for video texture information. It reduces the impact on coding performance of probability mis-estimation when transiting between image regions having different texture characteristics while not dramatically increasing computational complexity. Bit rate savings of 0.75 to 1.6 % for CIF sequences and 3.5 to 4.5 % for 720p resolution sequences are obtained.

7. REFERENCES

[1] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[2] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand, "Context-based Adaptive Binary Arithmetic Coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, July 2003.

[3] D. Hong, M. Van der Schaar, and B. Pesquet-Popescu, "Arithmetic coding with adaptive context-tree weighting for the H.264 video coders," in *SPIE Visual Comm. and Image Proc. (VCIP)*, 2004.

[4] M. H. Firooz and M. S. Sadri, "Improving H.264/AVC entropy coding engine using CTW method," in *25th Picture Coding Symposium*, 2006.

[5] D. Karwowski and M. Domanski, "Improved context-adaptive arithmetic coding in H.264/AVC," in *17th European Signal Proc. Conf (EUSIPCO)*, 2009.

[6] E. Belyaev, M. Gilmutdinov, and A. Turlikov, "Binary arithmetic coding system with adaptive probability estimation by virtual sliding window," in *Proc. Intl. Symp. for Consumer Electron. (ISCE)*, 2006.