TOWARDS OPTIMISED CONTEXT SELECTION IN SCALABLE WAVELET BASED VIDEO CODING

Toni Zgaljic, Marta Mrak and Ebroul Izquierdo

Multimedia and Vision Research Group, Queen Mary, University of London
Mile End Road, E1 4NS, London, UK
phone: + 44 20 7882 7880, fax: + 44 20 7552 7997, email: {toni.zgaljic, marta.mrak, ebroul.izquierdo}@elec.qmul.ac.uk
web: www.elec.qmul.ac.uk/mmv/

ABSTRACT

The effectiveness of arithmetic coding in image and video compression depends on probability estimation of symbols to be encoded. Considering context in which these symbols occur can lead to improved compression. In wavelet-based coding same contexts are usually used for all wavelet subbands of same type across different scales of wavelet transform. Efficiency of different strategies for context selection has not yet been fully analysed for application in scalable wavelet-based video coding. In this paper an algorithm for context modelling based on tree structure is adapted for optimisation of contexts for symbols generated during Embedded ZeroBlock Coding (EZBC) of wavelet coefficients. With the proposed technique optimised contexts have been adaptively obtained for different wavelet subbands and EZBC quadtree levels. Comparison with predefined context models, which are used in common approach, shows that context models based on tree structure together with advanced modelling strategy significantly improve overall compression.

1. INTRODUCTION

As a final compression step, entropy coding in video coding plays an important role in overall system performance. Recently, advanced probability estimation techniques have become inevitable tools in highly optimised image and video compression systems. Entropy coding modules in H.264 / AVC video coding standard [1], JPEG2000 still image compression standard [2], as well as other compression systems that are used in wavelet-based image and video coding, such as Embedded ZeroBlock Codec (EZBC) [3], all use advanced probability estimation techniques. Underlying algorithms use contexts in which symbols occur in order to achieve better prediction, and thus better compression. All these strategies use predefined contexts which is still a requirement for real-time systems since adaptive context selection generally has high computational requirements. However, it has been shown that application of contexts adaptively selected for each encoding unit can improve overall compression [4].

Context modelling is a process of designing contexts for symbols to be encoded based on the values of neighbouring symbols (so-called *context elements*) with a target that the resulting length of the compressed data is minimised. While techniques that adaptively choose contexts according to the statistics of underlying source add complexity that is required by context modelling, they can still be used as offline methods for design of models that will be used as predefined contexts. Currently, two common strategies are used for context optimisation. In the first strategy contexts are quantised by so-called context quantiser which is obtained using different optimisation methods targeting minimisation of the code length. Examples of this strategy are described in [6] and [7]. In [6], context elements are mapped to a common random variable which is quantised by a scalar. Different approach is used in [7] where it is shown how contexts can be quantised using similar approach as in vector quantisation. On the other hand, in the second strategy, context optimisation based on tree structures shapes the contexts trees according to underlying source statistics.

In this paper we adapt a strategy for adaptive context modelling based on context tree representation in order to build new contexts for application in wavelet-based video coding systems. The proposed approach builds contexts in a training phase and then uses optimised models in real coding. Algorithm is based on growing and pruning of context tree. In the growing phase of the algorithm, each node of the context tree is assigned with context element which produces minimal length of the code in the current tree node. In the pruning phase of the algorithm, context tree is pruned in the way so that its leaves correspond to nodes of the full tree which give minimal overall code length. In this way the algorithm produces optimal context models for the used tree structure. The presented algorithm is used to obtain optimised contexts for symbols generated by EZBC bit-plane encoder, for each wavelet subband and quadtree level. Code length obtained by using contexts resulting from this algorithm is compared with the code length obtained when original contexts of EZBC algorithm are used. Results show improvement in compression efficiency when new contexts are used.

The remainder of this paper is organised as follows. Section 2 provides overview of methods used for texture compression in wavelet based scalable video coding with an emphasis on EZBC algorithm which is used in the presented work. In section 3, a method that optimises contexts for symbols generated by EZBC bit-plane encoder is presented.

This research was partly supported by the European Commission under contract FP6-001765 aceMedia.

Selected experimental results are shown in section 4. Section 5 concludes this paper.

2. ENTROPY CODING WITH CONTEXT MODELS

In wavelet-based scalable coding entropy coding generally consists of bit-plane coding and arithmetic coding of symbols generated by bit-plane encoder. Although these symbols are already highly uncorrelated, the remaining redundancy can be further exploited by a careful selection of context models that drive probability estimation at the arithmetic coder. However, a context selection mechanism depends on the data to be encoded and the first step is to provide an efficient binary representation of wavelet coefficients which is done by bit-plane encoder. In the following subsections the bit-plane coding and arithmetic coding used in popular EZBC codec are described.

2.1 Bit-plane encoding of wavelet coefficients

Bit-plane encoding is performed by applying a successive approximation quantisation to wavelet coefficients, i.e. by encoding magnitudes of wavelet coefficients in a bit-plane by bit-plane fashion from the highest bit-plane containing the most significant bits to the lowest bit-plane containing the least significant bits. In this way quantization step is halved by each encoded bit-plane and output bit-stream is embedded which provides quality scalability feature of the output bit-stream. In order to improve rate-distortion embedding of the final bit-stream, each bit-plane coding pass generally consists of at least two fractional bit-plane passes. These are the encoding of significance information and the encoding of refinement information. During the significant pass all coefficients that have not been found significant until the beginning of the current bit-plane pass are visited (i.e. those with magnitude lower than 2^{bp+1} if the index of the current bit-plane is denoted as *bp*) and the information if they have become significant in the current bit-plane are encoded. During the refinement pass, the refinement bits of all coefficients found significant in the previous bit-planes are encoded.

To efficiently exploit information redundancy inherent to wavelet coefficients, EZBC establishes a quadtree representation of individual wavelet subbands. A quadtree is built in a way so that each quadtree node at quadtree level *ql* contains maximum value of its four children nodes at the level $ql - 1$. The lowest level ($ql = 0$) corresponds to the pixel value level and contains magnitudes of wavelet coefficients. All nodes at all lower levels corresponding to the same spatial location of a node at a higher quadtree level are said to be its descendants, therefore a quadtree node represent the highest magnitude of all of its descendants. Quadtree nodes are encoded from the highest bit-plane to the lowest one starting from the nodes at the lowest quadtree level to the highest quadtree level in each bit-plane. During processing of a level *ql* of a quadtree only insignificant nodes with significant parents from previous bit-planes are visited. If a node has been found significant, its descendants are tested for significance in a recursive way until they are all found insignificant or the pixel level of the quadtree has been reached. Therefore, processing of each quadtree level *ql* consists of encoding nodes at the level *ql* and encoding nodes at lower quadtree levels which spatially correspond to significant nodes at the level *ql*. To differentiate these two types of level processing, the encoding of nodes at some level invoked by significance of nodes at a higher quadtree level in the same bit-plane will be referred to as encoding in *descendant mode*, otherwise as encoding in *parent mode*. After the processing of all quadtree levels in a single bit-plane, refinement information for coefficients found significant in the previous bit-planes is encoded.

2.2 Arithmetic coding and probability estimation using contextual information

Bit-stream generated by a bit-plane encoder can further be compressed by a binary arithmetic encoder. In order to exploit correlation between neighbouring symbols, arithmetic encoder is driven by conditional probabilities $p(x \mid CTX(x))$ where x is value of a symbol to be encoded and $CTX(x)$ is context in which x appears. Positions of symbols whose values are considered in creation of context define so-called context template. Thus, context template defines which neighbouring symbols (so-called context elements) are going to be considered for conditional probability estimation of the current symbol. Context template for encoding significance information of quadtree nodes in EZBC consists of eight neighbouring nodes as shown in Figure 1. For encoding significance information of nodes in the descendant node some additional contextual information is used, specifically the position of the node in the group of four adjacent nodes corresponding to the node at the next higher level and significance information of these nodes. EZBC contexts described in this paper are valid for intra-band modelling. In inter-band modelling contextual information from parent wavelet subband is also used.

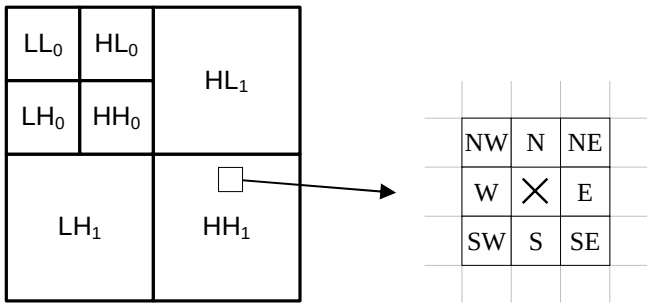

Figure 1 - Context template used in EZBC; × represents a position of the symbol to encode.

Although application of contexts can significantly improve compression, using too many contexts can result with context dilution meaning that probability estimations within contexts are inefficient since the number of symbols in each context is too low to obtain a good probability estimate. In such case data outputted by arithmetic encoder will be expanded. On the other hand, if too few contexts are used redundancies between symbols are not efficiently exploited resulting in less efficient compression. Thus, contexts have to be carefully selected. Considering these

facts EZBC uses linear combination of significance bits at positions of predefined context elements and quantises them into contexts.

For significance encoding of quadtree nodes contained in specific type of wavelet subbands (LL, HL, LH or HH), EZBC uses different contexts for 0-th level of quadtree, first and second level of quadtree and quadree levels higher than level two. For the latter set of levels a number of contexts is set to one, i.e. contexts are not used. Additional contexts are used for encoding of nodes in the descendant mode within quadtree levels. However, same contexts are used across different spatial resolutions for same types of wavelet subbands. It is expected that using different contexts for different resolutions of same wavelet subband types may further improve compression efficiency. Following this observation, in the work presented in this paper, contexts for symbols created by EZBC bit-plane encoder are optimised separately for different resolutions using the method described in the following section. Additionally contexts are optimised for each level of quadtree since it is expected that intra-band redundancy decreases with increasing quadtree level.

3. APPLICATION OF CONTEXT TREES IN WAVELET-BASED VIDEO CODING

In addition to the context definition using linear combination of values of context elements, an alternative approach for describing context is by arranging context elements into context trees. Using context tree representation, optimised contexts can be selected for a given context template knowing the statistics of values that are to be encoded. Also, in this approach, the optimisation takes into account the probability models in each context for used arithmetic coding. It has been shown that application of Growing, by Reordering and Selection by Pruning (GRASP) algorithm, [4], for context modelling using context trees in video coding, optimised context models can be found for various elements of compressed video syntax, improving the overall compression. So far this approach has been tested only in H.264 / AVC video coding standard. However, it can be used in other compression schemes such as wavelet based-video coding. Starting from GRASP approach, we optimise contexts models for EZBC used in video coding.

Context modelling and actual coding phases in our approach take the following order:

1. collecting of information on binary symbols and corresponding values of EZBC context elements,
2. processing of colleted data using GRASP algorithm in order to obtain optimised context models,
3. encoding with obtained context models.

In the first step a training set of video sequences or frames is used. For each element of the training set, data containing symbols to be encoded with their context element values are collected in the order of their occurrence. Data is collected separately for each so-called *syntax element* s_i. Thus, each s_i represents the set of symbols for which context optimisation is going to be performed separately, where

$$i = f(tl, sl, cp, sbb, ql, dsc) . \quad (1)$$

In (1) *tl* and *sl* represent temporal and spatial resolution level respectively for which data is collected, *cp* denotes index of the colour component, *sbb* represents index of subband type created by the wavelet transform, *ql* represents processed quadtree level, $dsc = 1$ if the symbols are created by encoding of quadtree nodes in the descendant mode and $dsc = 0$ if quadtree nodes are encoded in parent mode. For symbols created by encoding of refinement information the value of *ql* is set to the number of the quadtree levels for observed spatial resolution level. Maximum temporal and spatial resolution levels are determined by a number of wavelet decompositions applied to the input frames in temporal and spatial directions and index of subband type *sbb* is defined as

$$sbb = \begin{cases} 0 \text{ for } sl = 0 \text{ or } (sl > 0 \text{ and } type = HL) \\ 1 \text{ for } sl > 0 \text{ and } type = LH \\ 2 \text{ for } sl > 0 \text{ and } type = HH \end{cases} . \quad (2)$$

Context elements used in the optimisation process are the same as in EZBC, i.e. for the example in Figure 1 the context elements are from set {NW, N, NE, W, E, SW, S, SE}. Already in this step a different approach than in conventional EZBC is used. For instance, instead of treating different resolution levels with the same context model, here different models will be optimised for those syntax elements.

In the second step, for each syntax element a context tree is optimised for data from training set. Here, a tree-building algorithm is described for one syntax element that occurs in F frames. As in GRASP algorithm, starting from an empty tree root at tree depth $d = 0$, an index j of context element is assigned to the current node. y_j^d denotes assignment of value of the context element, y_j, with index j to a tree node at depth $d < D$, where D is the maximal tree depth and number of context elements.

For a current node at depth d, j is chosen from set $\{0, ..., D - 1\} \setminus \{j_0, ..., j_{d-1}\}$, so that the code length L_j, obtained by encoding using sequence of context assignments $Z_j = \left\{ y_{j_0}^0, y_{j_1}^1, ..., y_{j_{d-1}}^{d-1}, y_j^d = l \right\}$, $l \in \{0, 1\}$, to reach the node at the depth $d + 1$ from the root node is minimised.

Code length L_j is for each available context element computed as:

$$L_j = -\sum_{l=0}^{1} \sum_{f=1}^{F} \sum_{t=0}^{N(f, Z_j)-1} \log_2 \frac{c_{b(t,f)}^{Zj}(t-1, f)+1}{c_0^{Zj}(t-1, f) + c_1^{Zj}(t-1, f) + 2} , \quad (3)$$

where $N(f, Z_j)$ is a number of symbols that occur in the current node for frame f in sequence of context assignments

Z_j. c_i are counters for binary symbols, $i \in \{0, 1\}$, and $b(t, f)$ is a symbol at the time instance t. Initial values of counters, $c_i(-1, f)$ are set to 0. With each occurrence of symbol $b(t, f)$, counter $c_{b(t,f)}$ is updated. The evaluation and selection of available context elements is performed recursively until the maximal tree depth is reached. In contrast to original GRASP algorithm, which considers only counts of binary symbols regardless of the time of their occurrence, this mechanism allows using additional features of probability modelling, such as scaling of counts. It this way it simulates the steps of the encoder producing the code exactly as the encoder would produce it.

Optimal tree selection is performed in the opposite way: from context tree leaves towards the root using tree-pruning algorithm. The code length expected in each node, as given in (3) for selected context element, is compared to the code lengths of its child nodes. If the code length of current node is smaller then the sum of the code lengths of its child nodes, the branch below current node is removed from the tree. Otherwise the current node is associated with the sum of the code lengths of its child nodes. This algorithm is also recursively performed and as a result an optimised context tree is found.

After the second step has been performed, optimised context models are built for each syntax element in the training set. Those context models are then implemented in encoder and decoder and used for actual coding of video content.

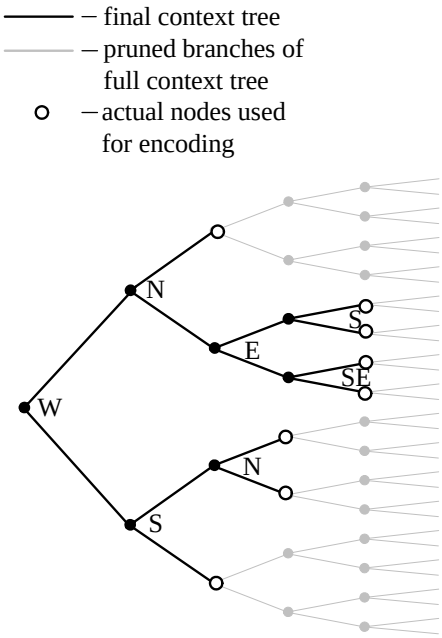

Figure 2 - An example of context tree obtained by GRASP optimisation on context template from Figure 1.

An example of context tree obtained with modified GRASP on EZBC context template is shown in Figure 2. Context elements considered for this context tree correspond to the context elements of EZBC (Figure 1). Since the application of full context tree in actual compression can lead to context dilution, the optimal context tree in this case is obtained by pruning of branches that do not increase overall compression. In Figure 2 the pruned tree nodes are labelled with symbols for context elements. The actual encoding is performed from the leaves of pruned context tree.

4. EXPERIMENTAL RESULTS

Experiments were performed in aceSVC scalable video coder [7], [8] on two test sequences: “City” and “Crew”. Resolution and frame rate of both sequences were 1280×720 and 60 fps allowing context optimisation for subbands at different resolutions. Sequences were encoded in intra mode, i.e. without temporal decomposition. Five levels of spatial wavelet transform were applied using biorthogonal CDF 9/7 wavelet. On each sequence, the training phase was firstly performed in order to obtain optimised contexts for the whole sequence. For the optimisation, all bit-planes that contribute to the high quality (> 45 dB) were taken into account. For each sequence, optimised contexts tailored to that sequence were used during the encoding of each frame.

Obtained results for syntax elements corresponding to encoding of significance information for nodes in parent mode of the 0-th, first and second quadtree level of the luminance component are shown in Table 1. Results show comparison of relative bit-rate savings for those syntax elements when EZBC contexts and contexts obtained by modified GRASP are used. Relative bit-rate saving is computed as

$$rbs = \frac{L_{wc} - L_c}{L_{wc}} \cdot 100\ [\%], \qquad (4)$$

where L_{wc} denotes code length of all frames obtained without using contexts and L_c denotes code length of all frames when context are used. In Table 1, *bp* represents the final decoded bit-plane of adapted bit-stream. Note that all results for one sequence have been obtained by adaptation of a single scalable compressed bit-stream.

From the presented results it can be seen that the application of contexts improves compression efficiency in all points. Contexts obtained by modified GRASP provide bit-rate saving in 90 % points up to 8.08 % when compared to the case when EZBC contexts are used. Also it can be observed that in 78 % points relative bit-rate savings increase when final bit-plane is higher. In these cases a number of symbols to be encoded is not large and efficient context modelling is desired. Results for lower resolution (640×360) were obtained by removing HL, LH and HH subbands resulting from the first level of wavelet transform. Note that in some cases relative bit-savings are exactly the same for both resolutions. These are the cases when bit-rate saving is observed for higher bit-planes and none significant coefficient has been found in the highest spatial subbands for those bit-planes.

Resolution	Seq	Contexts	Relative bit-rate saving by using contexts (*rbs*) [%]									
			bp = 0	*bp* = 1	*bp* = 2	*bp* = 3	*bp* = 4	*bp* = 5	*bp* = 6	*bp* = 7	*bp* = 8	*bp* = 9
1280 × 720	Crew	GRASP	4.26	5.48	7.31	9.42	11.09	12.87	14.14	15.40	16.88	13.08
		EZBC	3.66	4.75	6.45	8.41	9.85	11.08	11.59	11.65	8.80	7.82
	City	GRASP	3.87	4.37	4.82	4.98	4.67	4.24	3.65	2.55	4.42	2.65
		EZBC	3.21	3.62	4.08	4.43	4.55	4.56	3.96	1.77	2.05	1.03
640 × 360	Crew	GRASP	3.91	4.89	6.47	8.63	10.73	12.70	14.17	15.40	16.88	13.08
		EZBC	3.60	4.52	5.99	7.84	9.47	10.94	11.61	11.65	8.80	7.82
	City	GRASP	2.66	2.87	3.16	3.48	3.82	3.99	3.65	2.55	4.42	2.65
		EZBC	2.52	2.74	3.04	3.40	3.82	4.23	3.96	1.77	2.05	1.03

Table 1 - Experimental results

5. CONCLUSION AND FUTURE WORK

A strategy for context optimisation for encoding of symbols generated by EZBC bit-plane encoder has been presented. In EZBC, wavelet coefficients are classified according to subbands and encoded using a quadree structure. Since it is expected that using of different contexts for different quadtree levels and wavelet subbands improves compression, application of different context models for each of those combinations needs to be supported by the modelling technique. Chosen optimisation approach is based on GRASP algorithm for context modelling which is capable of selecting context models for various types of data in video coding. In the proposed approach the contexts are built according to data from a training set. These are then used as predefined contexts for encoding. In contrast to predefined contexts used in EZBC, contexts obtained by presented approach are arranged in context tree structures and are separately designed for each subband resolution and quadtree level. Experiments measuring compression efficiency when optimised contexts are implemented have been performed for high resolution sequences with intra video coding. Presented results show that with application of sfficiently designed contexts the bit-rate savings are achieved in 90% points for observed syntax elements comparing to the case when EZBC contexts are used. Additionally, it has been observed how optimised contexts obtained from encoding of training set at the high quality are also efficient at lower qualities. In the future the presented context optimisation method will be adapted to scalable motion-compensated video coding and optimisation will be performed on a large training set.

REFERENCES

[1] D. Marpe, H. Schwarz, Thomas Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Trans. on Circuits and Systems for Video Techn*., Vol. 13, No. 7, pp. 620-636, July 2003.

[2] D. Taubman, M. W. Marcellin, *JPEG2000 image compression: fundamentals, standards and practice*, Kluwer Academic Publishers, 2002.

[3] S.-T. Hsiang, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," in *Proc. Data Compression Conf.* March 2001, pp 83-92.

[4] M. Mrak, D. Marpe, T. Wiegand, "A Context Modeling Algorithm and its Application in Video Compression," in *Proc. Intl Conf. on Image Proc*., ICIP 2003, September 2003.

[5] X. Wu "Lossless Compression of Continuous-Tone Images via Context Selection, Quantization, and Modeling" *IEEE Transactions on Image Processing*, Vol. 6, No., pp. 656-6645, May 1997.

[6] J. Chen "Context Modeling Based on Context Quantization With Application in Wavelet Image Coding", *IEEE Transactions on Image Processing*, Vol. 13, No. 1, pp. 26-32, January 2004.

[7] N. Sprljan, M. Mrak, T. Zgaljic, E. Izquierdo, Software proposal for Wavelet Video Coding Exploration group, ISO/IEC JTC1/SC29/WG11/MPEG2005, no. M12941, 75th MPEG Meeting, January 2006.

[8] T. Zgaljic, N. Sprljan, E. Izquierdo, "Bit-stream allocation methods for scalable video coding supporting wireless communications," *Signal Processing: Image Communication*, Vol. 22, No. 3, pp. 298-316, March 2007.