

Adaptive Golomb Codes For Level Binarization In The H.264/AVC FRExt Lossless Mode

Davide Bardone, Elias S. G. Carotti, Juan Carlos De Martin

Dipartimento di Automatica e Informatica

Politecnico di Torino

c.so Duca degli Abruzzi, 24 — 10129, Torino, Italy

phone: +(39) 011 090 7036, fax: + (39) 011 090 7138,

email: [davide.bardone|carotti|demartin]@polito.it

Abstract— *Fidelity Range Extensions (FRExt) is an H.264/AVC amendment which provides enhanced coding tools and the possibility to perform high resolution and lossless video encoding. However, most of the efforts for lossless coding in the H.264/AVC framework have been concentrated on improving the prediction step while leaving the entropy coder, CABAC, originally designed for lossy coding, unaltered. However, if transformation and quantization of the corresponding coefficients are not performed, as is the case of the lossless coding mode for FRExt, CABAC becomes sub-optimal. In this paper we show how considerable improvements in compression ratios can be achieved with simple modifications of the CABAC engine. The proposed technique was tested on a set of 4:4:4 test sequences, achieving gains of up to 12.80% with respect to the original unmodified H.264/AVC algorithm.*

Keywords— *Lossless, Video Coding.*

I. INTRODUCTION

Lossless image and video coding has received much attention in recent years, because of its importance in many different fields ranging from medical imaging to professional video editing. Often, in fact, it is very important to preserve images to the best quality possible especially when further processing is needed or where images have to be used for diagnostic purposes.

The well-known standard for lossy video coding H.264/AVC[1][2] was designed by the Video Coding Experts Group (VCEG) of the ITU-T and the Moving Picture Experts Group (MPEG) of ISO/IEC to be capable of lower bitrates than previous standards (nearly half or less of the H.262/MPEG-2 bitrate, given the same video quality), without excessively increase complexity (also considering progresses in VLSI design technology). H.264/AVC was also conceived as an ensemble of state-of-the-art coding tools able to cover a wide range of applications, providing high robustness and the maximum flexibility between low and high bitrates, from video telephony to high-definition DVD storage.

The original standard H.264/AVC could already perform lossless coding, although not very efficiently, by means of a special “PCM” macroblock mode, where pixel sample values were directly coded without any form of prediction, transformation and quantization. However, the “PCM” mode had not been designed either for high quality or lossless coding. Thus, the standard was extended in July 2004 with an amendment

called *Fidelity Range Extensions (FRExt)*[3][4], to improve coding efficiency for some applications as well as to provide new coding tools for the highest video resolutions, going beyond the entertainment-quality videos for which the original standard had been designed. These extensions were motivated by the increasing demand for highest video resolutions, for professional applications such as film and video post production, studio editing and high definition television or DVD, also including specifications for lossless coding, were the highest possible quality has to be preserved.

Thus, the FRExt amendment introduced a suite of new profiles collectively called the *High* profiles, among which the *High 4:4:4* profile (H444P), supporting up to 4:4:4 chroma sampling, integer color transform for RGB video sequences to avoid rounding errors during color-space transformations, higher bit-depths, and, last but not least, lossless coding.

More recently, researchers proposed some further improvements on the FRExt lossless mode as in [5], approved for standardization in a new amendment[6], where the authors applied sample-by-sample differential pulse code modulation (DPCM) to the H.264 multidirectional spatial Intra prediction residuals. Previously, Takamura *et al.* in [7] proposed to use a reversible, adaptive transform as an alternative to the FRExt lossless compression scheme.

H.264/AVC comes with two different entropy coding engines, one using variable length instantaneous codes (CAVLC) and the second (more complex but more efficient) based on a context-adaptive binary arithmetic coder (CABAC)[8].

CABAC is typically used for demanding applications at high resolutions, gaining additional performance ($\sim 10\%$) over CAVLC. On the other hand, this greater efficiency is obtained at the price of significantly higher computational complexity.

If CABAC is used, non-binary values are first converted to binary strings and then entropy coded by means of an adaptive binary arithmetic coder. However, despite the very efficient performance for lossy coding, the current binarization and entropy coding scheme still lags behind for lossless coding and can be improved to achieve better packing of the data if lossless coding is desired.

In fact, CABAC currently expects to deal with runs of quantized coefficients whose values usually tend to be small,

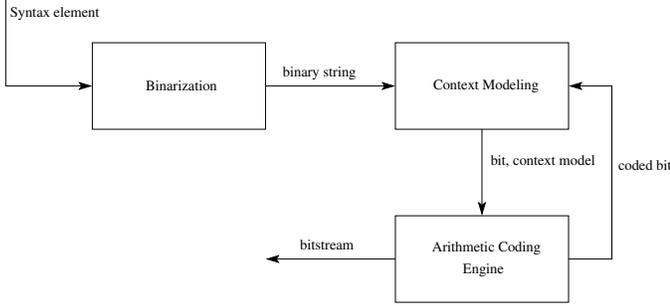


Fig. 1. High level block diagram of the CABAC entropy encoder for non-binary valued symbols.

and moreover high frequency transform coefficients are often either zero or very small after quantization. This is not true if transform is not performed, especially for high motion scenes where prediction, in general, fails to remove all the redundancy, or for intra-coded frames.

In this paper we propose a binarization scheme for the CABAC entropy coder based on low-complexity adaptive Golomb-Rice codes, which is more suited to lossless coding, in order to achieve better performance with respect to the binarization scheme used for run-lengths of pixel values currently used by H.264/AVC.

Paper organization is as follows: first, the FReXt lossless mode is described and the CABAC encoder is briefly reviewed in Section II, then, in Section III the proposed technique is presented. Experimental results are discussed in Section IV, and, finally, conclusions are drawn in Section V.

II. THE FREXT LOSSLESS MODE

The FReXt amendment introduces a transform-bypass lossless mode in the High 4:4:4 profile which immediately performs entropy coding after prediction, without any spatial transform and quantization steps. In fact, CABAC is directly fed with the prediction residuals yielded either by intra prediction on I-frames or motion compensation on P- and B-frames. However, neither the binarization engine, nor CABAC have been changed to better reflect the different type of data, i.e. the untransformed and unquantized residuals. In fact, although bypassing the transformation step leads, in general, to more efficient lossless video coding, the current binarization scheme still expects runs of small or zero-valued coefficients and it is not optimized for high valued prediction errors such as those usually generated on Intra-coded frames or when prediction fails to remove all the redundancy, e.g. along edges or in high motion scenes in inter-prediction modes.

Since CABAC is based on arithmetic coding, it can assign a non-integer number of bits to each symbol according to an adaptive probability model.

Fig. 1 shows a high-level block diagram of the CABAC entropy coder scheme. Initially each non binary-valued syntax element is converted into a binary sequence called *binary*

```

exp_golomb_encode(symbol, k)
{
    while(1) {
        if (symbol >= (1<<k)) {
            emit(1); // Unary part
            symbol = symbol - (1<<k);
            k++;
        } else {
            /* First terminates unary part */
            emit(0);

            /* Remaining binary part */
            while (k-->0)
                emit((x >> k) & 0x01);
            break;
        }
    }
    return;
}
  
```

Fig. 2. Pseudo-code describing the construction of the k^{th} -order EG k code, for a given unsigned integer symbol.

string. For each bit a statistical model is selected and a proper context is determined to achieve better compression through context conditioning. The context is meant to select the proper probability model, and is adaptively updated after each binary symbol has been coded.

Despite the well known efficiency of arithmetic coding, the binarization step is very important to achieve good compression performance.

The CABAC framework provides for four types of binarization and their concatenations. These four basic types are:

- the *unary code*;
- the *truncated unary code*;
- the *k th order Exp-golomb code*;
- the *fixed-length code*.

The *unary* scheme maps an integer unsigned value $x \geq 0$ on a sequence of x '1' with a terminating '0'. Its *truncated* version is defined only for x , with $0 \leq x \leq M$; if $x = M$, the terminating '0' is omitted.

K th-order *Exp-Golomb* codes, constitute a parametrized family of codes derived from the Golomb codes, first proposed in the context of run-length coding schemes[9]. The Exp-Golomb codes are two parts codes, constructed by the concatenation of a unary prefix part and a binary suffix code. The prefix part consist of a sequence of $n(x)$ '1', where x is the unsigned integer value to be coded and $n(x)$ is defined as

$$n(x) = \lfloor \log_2 \left(\frac{x}{2^k} + 1 \right) \rfloor.$$

The suffix part corresponds to the binary representation of

$$x + 2^k(1 - 2^{n(x)}),$$

using k significant bits. Fig. 2 shows the pseudo-code to perform the Exp-Golomb binarization of an unsigned integer symbol given parameter k .

Value	Truncated Unary Prefix	Zero order Exp-Golomb Suffix
1	0	
2	1 0	
3	1 1 0	
4	1 1 1 0	
...	...	
13	1 1 1 1 1 1 1 1 1 1 1 1 0	
14	1 1 1 1 1 1 1 1 1 1 1 1 1 0	
15	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0
16	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 0 0
17	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 0 1
18	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 0 0 0
...

Table 1. UEG0 Binarization for some absolute values of transform coefficient levels.

Value	Unary Prefix	Binary Suffix
1	0	0 0 1
2	0	0 1 0
3	0	0 1 1
4	0	1 0 0
...
13	1	0 1 0 1
14	1	0 1 1 0
15	1	0 1 1 1
16	1 1	0 0 0 0
17	1 1	0 0 0 1
18	1 1	0 0 1 0
...

Table 2. Golomb-power-of-2 binarization for some absolute values of prediction residuals assuming an estimated $k = 3$.

Finally, the *fixed-length* binarization scheme simply assumes a finite alphabet of values, with a fixed maximum S , thus coding every syntax element x with $0 \leq x \leq S$ on $\lceil \log_2 S \rceil$ bits.

Motion vectors differences and absolute values of transform coefficient levels are binarized by CABAC using a combination of *truncated unary* codes and *k-th order Exp-Golomb* codes. This binarization scheme is referred to as *Unary/kth order Exp-Golomb*, or simply as UEGk.

The absolute values of the transform coefficients are binarized with a UEG0 binarization, with a cutoff value $M=14$ and a fixed order $k = 0$. Table 1 shows the corresponding bin strings for some values of this kind of syntax elements. This scheme leads to a simple and low complexity binarization and a fast adaptation of each symbol probability in the context modeling stage, particularly for small values. Indeed, for each bin belonging to the truncated unary prefix a simple context based on the number of *trailing 1's* already coded and on the accumulated number of coded levels with absolute value greater than 1 is computed. For the suffix part a uniform distribution of the bins is assumed to speed up encoding.

III. PROPOSED BINARIZATION TECHNIQUE

When lossless coding is performed CABAC is fed with a sequence of unquantized prediction residuals, typically characterized by an absolute value greater than quantized transform

coefficients levels. Indeed, for lossy coding, the quantization step is usually tuned in order to coarsely represent the less perceptually important high frequency coefficients and more faithfully represent lower frequencies. Since the UEG0 binarization scheme was designed assuming that most coefficients had small magnitude, it is not efficient when this assumption does not hold, instead leading to poor coding performance.

To overcome this problem we propose to perform binarization by means of a family of Golomb codes which are optimal for one-sided geometric distributions of non-negative integers, which is a good approximation of prediction residuals for image and video sequence data. More specifically, binarization is achieved by means of a family of *power-of-2* Golomb codes similarly to the lossless image compression algorithm LOCO-I[10], core of the JPEG-LS[11] standard, allowing for very low-complexity encoding and decoding. This also implies that both the encoder and the decoder have to be changed to properly decode the bitstream.

Golomb codes have been proven in [12] to be optimal for *one-sided geometric distributions* of non-negative integers x , that can be described, for some arbitrary θ , $0 < \theta < 1$, as

$$(1 - \theta)\theta^x,$$

which are known to be a good approximation for prediction residuals on images and video sequences.

For every θ there exists an optimal parameter l which leads to the shortest binary representation, satisfying:

$$\theta^l + \theta^{l+1} \leq 1 < \theta^l + \theta^{l-1}.$$

In the *power-of-2* Golomb codes family $l = 2^k$, allowing for an extremely simplified parameter estimation.

Fig. 3 depicts the high-level block diagram of the optimal k parameter estimation for each pixel in a sub-block of the macroblock to be coded. First, k is estimated based on local, per-block statistics, then estimated k is sent to the coder along with the value to be coded, lastly, statistics are updated with the newly coded symbol.

An accumulator A and a counter N are used to estimate the proper k , which is thus computed on the expected magnitude of the symbol to be coded. The estimation for the next value is then performed as

$$k = \min\{k' | 2^{k'} N \geq A\}.$$

This operation can be performed in C programming language with a "one-liner"

$$\text{for } (k = 0; (N \ll k) < A; k + +);$$

as in the LOCO-I implementation.

The sequential, backward adaptive nature of this operation allows to perform the same estimation in a symmetrical way both at encoder and decoder sides, without introducing any loss of information.

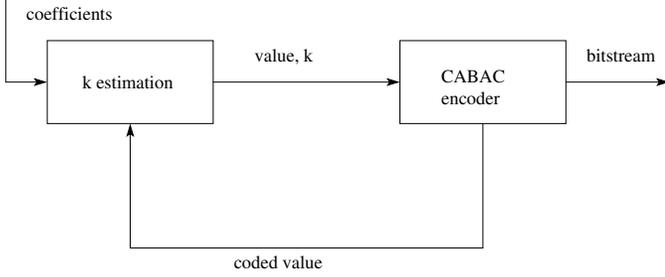


Fig. 3. K parameter estimation at the encoder side for the proposed binarization scheme.

Sequence	Total frames (Hz)	Coding Options
CrowdRun	101 (50Hz)	Rate-Distortion
ParkJoy	101 (50Hz)	Optimization,
DucksTakeOff	101 (50Hz)	CABAC,
IntoTree	101 (50Hz)	GOP: IBPB,
OldTownCross	101 (50Hz)	FRExt Lossless mode

Table 3. Test sequences and coding conditions.

This binarization, given the value to be coded y uses a unary representation of $q = \lfloor y/2^k \rfloor$, a 0 bit as a delimiter, and a binary representation of $r = y \bmod 2^k$ instead of a fixed k for the Exp-Golomb part and a truncated unary prefix as in regular CABAC.

This family of Golomb codes allows for low complexity parameter estimation since all the operations involved can be efficiently performed with only shift-add arithmetic.

Table 2 shows the corresponding bin strings for some values processed with this technique, assuming an estimated $k = 3$.

Our goal is to estimate k as accurately as possible, thus minimizing the length of the unary prefix, obtaining the exact binary representation of the value, without adding further complexity to the original design. This kind of approach aims to reduce the quantity of binary symbols to be encoded, especially for large residual values for which the original binarization design was not optimized.

IV. RESULTS

The proposed binarization technique was implemented into the JM H.264/MPEG-4 AVC Reference Software [13] version 13.2, and tested on a set of five High-Definition color video sequences[14]. We converted the sequences of SGI (Silicon Graphic Image format[15]) frames to 8bit 4:4:4 YUV video sequences using the sgi2yuv tool implemented by the Lehrstuhl für Datenverarbeitung[16]. Table 3 describes test sequences and coding conditions used to verify the validity of the proposed technique.

Tables 4-5-6 show compression results in terms of average bit per pixel, and for any frame type. Performances of the standard encoder with the FRExt lossless option enabled are compared with those of the encoder with the same configuration but using the proposed binarization scheme. It can be seen that better results are achieved with sequences characterized

Sequence	H.264/AVC	Proposed Technique	Gain
CrowdRun	15.19	13.73	9.62%
ParkJoy	15.69	13.88	11.56%
DucksTakeOff	14.84	14.08	5.13%
IntoTree	12.54	12.68	-1.09%
OldTownCross	12.97	13.04	-0.54%

Table 4. Compression results in terms of bit per pixel averaged over 26 1280×720 I-Frames, using the standard H.264/AVC binarization and the proposed binarization scheme.

Sequence	H.264/AVC	Proposed Technique	Gain
CrowdRun	14.71	13.45	8.55%
ParkJoy	15.68	13.67	12.80%
DucksTakeOff	14.83	14.08	5.06%
IntoTree	12.15	12.49	-2.83%
OldTownCross	12.85	12.99	-1.05%

Table 5. Compression results in terms of bit per pixel averaged over 25 1280×720 P-Frames, using the standard H.264/AVC binarization and the proposed binarization scheme.

Sequence	H.264/AVC	Proposed Technique	Gain
CrowdRun	14.58	13.37	8.31%
ParkJoy	15.42	13.52	12.32%
DucksTakeOff	14.82	14.08	4.97%
IntoTree	12.04	12.28	-2.04%
OldTownCross	12.80	12.94	-1.06%

Table 6. Compression results in terms of bit per pixel averaged over 50 1280×720 B-Frames, using the standard H.264/AVC binarization and the proposed binarization scheme.

by an higher bitrate, due to a more difficult pixel values predictability. Gains are lower when sequences are characterized by limited motion activity with slow panning cameras, allowing motion compensation to generate low prediction residuals, similar in magnitude to the transform coefficients absolute values for which the original binarization scheme was designed.

The proposed coding algorithm gains up to the 12.80% with respect to the standard H-264/MPEG-4 AVC binarization scheme with a modest increase of the overall architecture complexity.

V. CONCLUSIONS

Unquantized absolute values of prediction residuals are different in nature with respect to quantized absolute values of transform coefficients for which the H.264/AVC binarization design is conceived. In this paper we propose a binarization scheme for the H.264/AVC coder to be used in the FRExt lossless mode. The proposed scheme is based on the *Golomb-power-of-2* codes and is aimed at estimating the parameter which minimizes the bin string length, instead of using the original Truncated Unary/Zero order Exp-Golomb binarization. This technique, tested on a number of High Definition 4:4:4 video sequences, attains gains up to the 12.80% with respect to the original binarization scheme, with a modest increase in complexity of the coding-decoding process.

REFERENCES

- [1] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services," *ITU-T*, May 2003.
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.
- [3] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Lossless coding for professional extensions," July 2004.
- [4] D. Marpe, T. Wiegand, and S. Gordon, "H. 264/MPEG4-AVC fidelity range extensions: Tools, profiles, performance, and application areas," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 1, Sept. 2005, pp. 593–596.
- [5] Y.-L. Lee, K.-H. Han, and G. J. Sullivan, "Improved lossless intra coding for h.264/MPEG-4 AVC," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2610–2615, Sept. 2006.
- [6] H. Yu, "Draft Text of H.264/AVC Advanced 4:4:4 Profile Amendment to ITU-T Rec. H.264 ISO/IEC 14496-10 AVC, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16 Joint Video Team document JVT-Q209," Oct. 2005.
- [7] S. Takamura and Y. Yashima, "H.264-based lossless video coding using adaptive transforms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, Philadelphia, PA, USA, Mar. 2005, pp. 301–304.
- [8] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, July 2003.
- [9] J. Teuhola, "A compression method for clustered bit-vectors," *Inform. Processing Lett.*, vol. 7, pp. 308–311, Oct. 1978.
- [10] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: a low complexity, context-based, lossless image compression algorithm," in *Proceedings of Data Compression Conference*, March 1996, pp. 140–149.
- [11] I.-T. SG8, "Lossless and near-lossless compression of continuous-tone still images (ITU-T T.87—ISO/IEC 14495-1)," *ITU-T*, June 1998.
- [12] R. Gallager and D. van Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 228–230, Mar. 1975.
- [13] F. Heinrich-Hertz-Institut, "H.264/AVC Reference Software," <http://iphome.hhi.de/suehring/tml/download/>, 2008.
- [14] Ftp server managed by Video Quality Experts group (VQEG), "SVT Multi Format Test Set," <ftp://vqeg.its.bldrdoc.gov>.
- [15] S. G. C. Systems, "The SGI Image File Format (specification)," <ftp://ftp.sgi.com/graphics/SGIIMAGESPEC>.
- [16] Lehrstuhl für Datenverarbeitung, "sgi2yuv video tool," <http://www.ldv.ei.tum.de/Members/tobias/videotools/sgi2yuv.zip>.