We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Effective Video Encoding in Lossless and Near-lossless Modes

Grzegorz Ulacha

West Pomeranian University of Technology,
Faculty of Computer Science and Information Technology
Poland

1. Introduction

The main goal of the research is to develop an algorithm appropriate for hardware realization of a lossless and near-lossless video sequences. The importance of compression is unquestionable in contemporary digital video signal processing systems, with an extensive requirement on a high storage and bandwidth imposed on data storage and transmission. For example, a 60 minutes long uncompressed movie with the TV PAL quality (i.e., 25 frames per second, 720x576 pixels) requires 104.28 GB hard disk space and 9.89 Mpixels/s of bandwidth in the 4:4:4 profile. One of the major requirements for the system is its work in the real-time. This is caused by one of the primary applications of the lossless video compression, i.e., edition of television programs, movies etc. (Andriani et al., 2004). In this situation it is not recommended to use lossy compression methods, such as MPEG2, MPEG4 etc.

Other important applications of the lossless video sequence compression is a storage of medical 2D and 3D images (in general, multi-planar 3D objects) (Xie et al., 2007), as well as astronomical images, and satellite photos compression (photographs of the Earth, Mars and other celestial bodies made by satellites and spaceships should be sent and stored in a lossless form) (Chen et al., 2004; CCSDS, 2007). A novel kind of application is the slow motion video technique for recording from scientific experiments up to dozen of thousands frames per second (fps). For example, recording of one second long video sequence (lumination signal, 8 bit/pixel) using 1000 fps and the SD (720×576 pixel) resolution needs 3955 MB of memory. Nowadays video cameras in the slow motion mode stores the images without any compression and the time of the recording is limited with the volume of the storage card in the device. Recording and archiving files of a larger size becomes an important issue that may be solved by introducing fast, hardware-based, parallel compression realization, e.g., in the near-lossless mode (see Section 3). In combination with the matrix of the effective SSD discs, it may lead to the increase of the functionality of the scientific site using the slow-motion camera.

2. Lossless static images and video sequences compression

This section presents the basic features and the motivation of the choice of the prediction blending technique. There are also described the blocks for contextual removing the

constant coefficient and for adaptive arithmetic encoding. The proposed method is compared, in terms of effectiveness, with other techniques known from literature.

2.1 Existing solutions in the field of lossless image compression

The CALIC method, up to today perceived as one of the most efficient techniques, was presented in 1996 (Wu & Memon, 1996). In that time this method was too computational complex, especially in comparison with the LOCO-I, whose modification has become the JPEG-LS standard (Weinberger et al., 2000). Among the algorithms with high implementation complexity three methods are worth to be mentioned: the TMW method (Meyer & Tischer, 1997) and its further extension, TMW[LEGO] (Meyer & Tischer, 2001b), WAVE-WLS (Ye, 2002) and MRP 0.5 (Matsuda et al., 2005). An encoding of a single image with any of these methods required a few hours in the time of their proposals (a dozen of minutes using the most efficient contemporary processor). Apart from the mentioned above methods utilizing predictive modelling, there are lossless versions of wavelet codecs used for encoding (e.g. JPEG-2000 (Marcellin et al., 2000)). However, the obtained results are inferior in comparison with the best predictive methods.

In (Andriani et al., 2004), it was presented an analysis of lossles video sequences compression, where methods LOPT-3D, GLICBAWLS-3D, JPEG-LS and JPEG-2000 have been compared. Based on the complexity analysis for the real-time applications, it was proposed to encode each video sequence frame independently with a reduced version of the JPEG-2000 standard (Andriani et al., 2004). It allows us to compress in the visually lossless mode (i.e., with low loss, unvisible for humans). Such the proposals as GLICBAWLS-3D, LOPT-3D (and its extension to the LPOSTC method (Andriani et al., 2005) belong to the solutions of high efficiency, but are too demanding in terms of computational complexity to be applied as real-time implementations.

Having looked for a solution being a compromise between efficiency and complexity, we decided to apply the blending predictors method, which is characterized with the highest flexibility (Seemann & Tisher, 1997a).

2.2 Subpredictor blending method

In the case of using contemporary compression methods, two stages must be devised: data modeling, and then compression with one of the efficient entropy method, among whom the most effective are arithmetic and Huffman encoding (Sayood, 2002). Data modeling in such cases aims at reducing maximum possible mutual information between neighboring pixels.

In modeling stage, an r-th order linear predictor computes an estimated value $\hat{x}$ of the n-th pixel taking into account values of the previously r neighboring pixels. Then, only the estimation errors is encoded, i.e., the difference between the actual and the predicted values (rounded up to the closest integer), which is usually small values close to zero:

$$e = P(0) - \hat{x} . \tag{1}$$

The obtained error values of the prediction are considerably lower than the initial values of the variance. Moreover, their distribution resembles the Laplace one. Both these features result in decreasing the value of unconditional entropy.

From the hardware realization point of view, one of the most important factor is time proportion between encoding and decoding. Time symmetric methods are of similar computational complexity of encoder and decoder. It is the reason why we resigned from

the methods for selecting an individual static predictor (or a set of such the predictors) based on the criteria of minimizing a minimum mean square error, MMSE (Sayood, 2002; Wu & Memon, 1996) (which requires introducing delays and solving sets of linear equations using floating numbers), and other methods needed an introductory preparation of modeling and compression parameters.

The works of G. Deng (Deng & Ye, 1999, 2003; Ye et al., 2000) show high efficiency of the blending prediction method. Having a given set a simple, constant predictors (named further subpredictors), one may conclude that the unconditional entropy (0-order) value obtained by an encoding with each of these subpredictors individually are far from being efficient; however, using even 7 simple subpredictors together leads to significant decrease of the entropy value (Seemann & Tisher, 1997a) in the majority of benchmark images. It is the main motivation of our interest in this approach. Another benefit of this method is the possibility of performing a number of computation in parallel for a single pixel. In order to evaluate our results and compare them with other authors' and choose the most appropriate set of subpredictors, we made usage of the fact that every author of algorithms utilizing methods of subpredictors blending has presented his own set of subpredictors. Taking into account the problems of implementation complexity, it was usually simple constant subpredictors of the range form 1 to 3. In case of the predictors of range 1, it was proposed to use the closest, neighbour pixels $P(i)$, where i is the index of the neighbour pixels (see Fig. 1). To these simplest subpredictors, there were added also constant predictors, such as planar (known also as Plane or JPEG4), Pirsh, etc.

19	11	8	6	9	12	22	
15	7	3	2	4	10	18	28
13	5	1	0				

Fig. 1. Numbering of the neighboring pixels

In this section we present the proposed flow for modeling and lossless image and video sequences compression, named Blend-V. This method uses 14 subpredictors: $x_1 = \text{GAP}_+$ (Wang & Zhang, 2004) (an improved version of the non-linear prediction used in the CALIC algorithm, which is a modification of GAP_+ described in (Wu & Memon, 1996), in our previous research, we also analysed a MED_+ modification from (Jiang & Grecos, 2002), but later we concluded that better results are obtained when this subpredictor is excluded), $x_2 = \text{GradWest} = 2P(1) - P(5)$, $x_3 = \text{GradNorth} = 2P(2) - P(6)$, $x_4 = \text{Plane} = P(1) + P(2) - P(3)$, $x_5 = \text{Plane2} = P(1) - P(2) + P(4)$ (Seemann, et al. 1997b), $x_6 = P(1)$, $x_7 = P(2)$, $x_8 = P(3)$, $x_9 = P(4)$, $x_{10} = P(5)$, $x_{11} = P(10)$, $x_{12} = P(18)$, $x_{13} = P(28)$, $x_{14} = P_{j-1}(0)$, where $P(i)$ denotes the pixel with index i from the closest neighborhood - see Fig. 1, and $P_{j-1}(0)$ denotes the pixel from the previous frame of the same coordinates that the currently encoded pixel, $P(0)$. Each of these proposals is suitable for encoding without any delay, they do not require any preliminary calculations referring to data from the entire image. Due to the encoding order (the consecutive rows are encoded top-down, and each of them left-right), both encoder and decoder have an access to the pixels placed above and on the left on the pixel being encoded (decoded). It is then required and access to the 3 previous rows (the row currently encoded and the two previous ones). In the proposed technique, we assigned numbers to neighbor pixels according to the Euclidean metric. The assignment of the numbers with the same distance is performed clock-wise.

A prediction error, $e_i(0) = P(0) - x_i$, is associated with each subpredictor x_i. A significance of a particular subpredictor is inversely proportional to prediction errors obtained in the closest neighborhood. The problem to be solved was the fact, that for the analyzed images the optimal neighborhood (common for all subpredictors) were situated in the whole analyzed range of k, i.e., from 3 to 30. Seemann and Tisher (Seemann & Tisher, 1997a) have proposed $k = 3$, Deng in his works has used $k = 4$ (Deng & Ye, 1999, 2003; Ye et al., 2000). As a result of our works, for a set of 45 various benchmark images, the best average results have been obtained for $k = 10$. The total error value, E_i, of the neighborhood associated with an i-th subpredictor is calculated with formula:

$$E_i = 1 + 2(e_i^2(1) + e_i^2(2)) + \sum_{j=3}^{k} e_i^2(j) . \qquad (2)$$

where $e_i(j)$ denotes the value of the prediction error obtained with the i-th subpredictor in the neighborhood with relative number j. To ensure that the value of the subpredictor weight, w_i, belongs to range 0 to α_i, it is necessary to determine it with the following equation:

$$w_i = \frac{\alpha_i}{E_i} . \qquad (3)$$

setting the importance parameter $\alpha_i = 1$ for each subpredictor but Plane2, for which $\alpha_i = 1.5$, and $\alpha_i = 2$ for the GradNorth and GradWest subpredictor. Taking into consideration a set of m subpredictors, we may determine its positive prediction coefficients, a_i, using their normalization (i.e., a sum of all the coefficients is to be equal to 1):

$$a_i = \frac{w_i}{\sum_{j=1}^{m} w_j} . \qquad (4)$$

Finally, the estimated value is computed in the following way:

$$\hat{x} = \sum_{i=1}^{m} a_i \cdot \hat{x}_i . \qquad (5)$$

2.3 Context-based error correction

Some particular properties of coded pixel neighborhoods may induce (transient, but long lasting) DC components in prediction errors associated with contexts. Each context is characterised with the individual properties of the closest neighbourhood of the pixels being encoded, taking into account mutual dependences existing between subsequent pixels, and often also their variance value. Context-based error correction methods consist in using occurrence number and cumulated error for each context for correcting current prediction error (Wu & Memon, 1996). Therefore, in our method the next stage is bias cancelation, i.e., usage of an adaptive method to remove the constant component $C_{(i)}$, which determines the error correction of the prediction associated with the appropriate context of the index i. This

method is used in both the CALIC and JPEG-LS algorithms. In our algorithm we use 1024 contexts and the arithmetic average from the results of these two methods:

$$\dot{x} = \hat{x} + \frac{C_{CALIC(i)} + C_{JPEG\text{-}LS(i)}}{2} . \tag{6}$$

Then, only the estimation errors is encoded, i.e., the difference between the actual and the predicted values (rounded up to the closest integer):

$$e = P(0) - \dot{x} . \tag{7}$$

The simplest method of calculating the number of context is to determine the weighted average of n constant subpredictors:

$$\bar{x} = \frac{102}{1024}\left(3 \cdot (P(1) + P(2)) + \sum_{j=3}^{6} P(j)\right), \tag{8}$$

e.g., for $n = 8$, the following subpredictors are utilised: $P(1)$, $P(2)$, $P(3)$, $P(4)$, $P(5)$, $P(6)$, GradNorth and GradWest. Next, the value of each of them is compared with the weighted average. If the value of the i-th subpredictor is higher than the average, bit flag z_i is set to 1, otherwise it is set to 0. From these flags, an n-bit number $z_{n-1}...z_3z_2z_1z_0$ is assembled. This number is the number of the context. In case of $n = 8$ we obtain number of the contexts equal to 256. Moreover, it is possible to determine the measure of the deviation from the average, measuring the variance level of these n subpredictors. This value can be quantized into Q partitions, which results in $Q \cdot 2^n = 1024$ contexts. The variance level (multiplied by 8) can be determined with the formula:

$$\sigma^2 = (\bar{x} - \text{GradNorth})^2 + (\bar{x} - \text{GradWest})^2 + \sum_{j=1}^{6} (\bar{x} - P(j))^2 . \tag{9}$$

To obtain the split into 4 quantisation levels, it was determined experimentally 3 thresholds of σ^2 values equal to 400, 2500, 8000, respectively. More details on the update component $C_{(i)}$ are described in (Ulacha & Stasiński, 2008).

2.4 Adaptive arithmetic encoder

In the developed system, an adaptive encoder of prediction errors has been designed as a separate module. Two main assumptions are the encoding without delays (i.e., in the real-time) and no feed-back between the modeling and the encoding/decoding blocks.

As we encode absolute values of the prediction errors, the distribution is close to the Laplace one. The initial distribution, i.e., the instance vector $n_e(i)$ of value i (the values of the absolute values of the prediction errors), can be treated as its approximation utilizing the simplified formula (Meyer & Tischer, 2001a):

$$n_e(i) = \left\lfloor A \cdot 0.8^i \right\rfloor + 1 . \tag{10}$$

Good results are obtained for $A = 10$. After reading and encoding each subsequent value of $|e|$, it is necessary to actualise the instance vector by increasing the $|e|$ index by 1.

Additionally, one can introduce the forgetting effect, which may lead to decreasign the weights of the instances which have appeared rarelier (or have not appeared at all) among the recently encoded values in comparison with the earlier encoding stage (Gallager, 1978). It is the reason why the total number of encoded values (or, more precisely, the counter value) is increased by 1 after every encoded number $|e|$.

The adaptive encoder is capable of adjusting to the distribution in long-term, but it is possible to use also the presence of short-term dependences between subsequently encoded data utilizing the closest neighbourhood of the two-dimensional error prediction signal. The properties of the neighbouring prediction errors (without considering the knowledge of the properties of the pixels neighbourhood) are capable of determining with better precision the distribution type of the currently encoded value $|e_n|$. Consequently, it is possible to design a context-based arithmetic encoder including K probability distributions (instead of one), associated with the context numbers from 0 to $K - 1$. Theoretically, one can assume the increase of compression efficiency with the growing number of contexts, but the problem of too slow adaptiveness of their distribution may appear. The adaptativeness of the distribution construction requires fast stabilisation of the target characteristics of each from the K distributions, so a certain trade-off between the number of contexts and the time of their adaptiveness has to be made. One often use 8 (Wu & Memon, 1996), 16, or even 20 contexts (Deng & Ye, 1999).

The errors, e, are compressed using a contextual, adaptive arithmetic encoder. We use seven universal and two 16-contextual adaptive arithmetic encoders ($K = 16$). In order to increase the adaptation speed of the distributions associated with the contextes, the quantisation of $|e|$ values is often applied. It is possible thanks to scaling down the number range from 0 to 255 to a lower range, e.g., from 0 to 17. This idea is applied in numerous encoding methods, e.g., in the JPEG standard. The first 16-contextual encoder compresses the absolute value of the error, $|e|$, quantized according to rule $\mathbf{T}(k) \leq |e| < \mathbf{T}(k + 1)$, where $\mathbf{T} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 32, 64, 128\}$. The value of k is send to the arithmetic encoder. The quantization error, $e_q = |e| - \mathbf{T}(k)$, is treated as $\mathbf{q}(k)$-bit number, where $\mathbf{q}(k)$ is the k-th value from vector $\mathbf{q} = \{0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 5, 6, 7\}$. If $\mathbf{q}(k) > 0$, the value of e_q is encoded using one of the 7 universal adaptive arithmetic encoders (with index $\mathbf{q}(k)$).

Due to the symmetry of the prediction error distribution, it is more convenient to encode their absolute values $|e|$, which results in faster adaptation of the distribution of each context of the arithmetic encoder. The second contextual encoder encodes the sign bit of the e value. Both these encoders use context switching based on the error level from the closest neighborhood, which allows us to determine the individual probability distribution type (one of the 16 contexts) for the currently encoded pixel. More details on the selection of the context number are described in (Ulacha & Dziurzański, 2008).

2.5 Experimental results

In Table 1, it is presented a comparison of the bitrates of the method proposed in this paper, Blend-V (the intraframe mode), with a few software-based, but fast and effective methods known from literature: JPEG-LS (Strutz, 2002), HBB (Seemann, et al. 1997b), CALIC (Ye et al., 2000), P13 (Deng & Ye, 1999). Among the compared methods there is also a bit slower, but effective method LAT-RLMS (Marusic & Deng, 2002), which shows the high efficiency of the proposed method, which is characterized with the lowest bitrate. The experiment has been performed for the set of 9 grayscale benchmark images of the resolution equal to

720×576 pixels. Relatively few proposals of hardware realizations of the lossless compression systems for video sequences have been presented. The system proposed in (Drost & Bourbakis, 2001) is one of the solutions well-known from literature. For example, for that proposal the bit average for the Lenna image of the 512×512 pixel size in 8-bit grayscale is equal to 4.609 bit/pixel. Applying our system, we are obtaining the average 4.006 bit/pixel, which is about 13 % efficiency improvement.

Images	JPEG-LS	HBB	CALIC	P13	LAT- RLMS	Blend-V
Balloon	2.889	2.80	2.78	2.74	2.75	2.746
Barb	4.690	4.28	4.31	4.29	4.15	4.172
Barb2	4.684	4.48	4.46	4.47	4.45	4.441
Board	3.674	3.54	3.51	3.48	3.48	3.465
Boats	3.930	3.80	3.78	3.75	3.74	3.716
Girl	3.922	3.74	3.72	3.67	3.68	3.640
Gold	4.475	4.37	4.35	4.33	4.34	4.323
Hotel	4.378	4.27	4.18	4.19	4.21	4.189
Zelda	3.884	3.72	3.69	3.68	3.61	3.673
Average	4.058	3.889	3.864	3.844	3.823	3.818

Table 1. Average bitrates for standard benchmark images

3. Near-lossless mode and color mode

In order to obtain higher compression ratio, it is possible to use an irreversible encoding. There exist a large number of lossy compression methods based on discrete cosines transform, DCT (e.g. JPEG), or wavelet transform (e.g. JPEG-2000) (Sayood, 2002). These methods allows us to adjust the compression level, but their basic implementations do not offer the capabilities of selecting the maximal possible error. This possibility is offered by the *near-lossless* mode, where it is possible to determine the highest acceptable value of the difference module, d, between the original and the decoded images. This mode is usually based on the predictive encoding, where the set of prediction error, e, is quantized in the following way (Carotti et al., 2004):

$$\bar{e} = \begin{cases} \left\lfloor \dfrac{e+d}{2d+1} \right\rfloor, & \text{for } e \geq 0 \\ \left\lfloor \dfrac{e-d}{2d+1} \right\rfloor, & \text{for } e < 0 \end{cases}, \tag{11}$$

and the original color value can be restored with the accuracy $\pm d$. When $d = 0$, we obtain the lossless mode.

The near-lossless method may be used for medical images compression, where experts should determine the error tolerance d that does not influence the diagnosis of the medical images of the given type (that compression kind is sometimes referred to as visually lossless (Andriani et al., 2004)), and in the situation when there is recording of video sequences with very large number of frames and only a slight error level can be acceptable. Often used quality measure, PSNR, is treated as not utterly capable of considering human visual

perception. The quality measure Mean Structural SIMilarity Index (MSSIM), developed in (Wang Z. et al., 2004), takes into consideration the comparison of the structural features of the source and the encoded images, as well as the features connected with luminance and contrast. This measure is a value ranged from -1 to 1; the higher value, the higher the images resemblance.

Blend-V				JPEG2000			
d	bitrate	PSNR	MSSIM	d	bitrate	PSNR	MSSIM
1	2.49144	49.89429	0.99473	7	2.49145	45.09458	0.98501
2	1.85465	45.15079	0.98462	8	1.85480	43.15705	0.97573
3	1.46825	42.26534	0.97124	11	1.46783	41.74485	0.96705
4	1.19771	40.26459	0.95682	12	1.20132	40.74678	0.95868
5	0.99264	38.75684	0.94257	17	0.99277	39.81996	0.95150
6	0.84358	37.51494	0.92906	17	0.84363	39.10160	0.94507
7	0.72745	36.45099	0.91631	24	0.72781	38.60727	0.93805
8	0.64068	35.47136	0.90388	27	0.64258	38.06484	0.93290
9	0.57260	34.61310	0.89212	27	0.57275	37.49548	0.92723
10	0.51554	33.86520	0.88133	29	0.51157	37.12479	0.92410

Table 2. Comparison of Blend-V in the near-lossless mode with JPEG2000 using the Lennagrey image
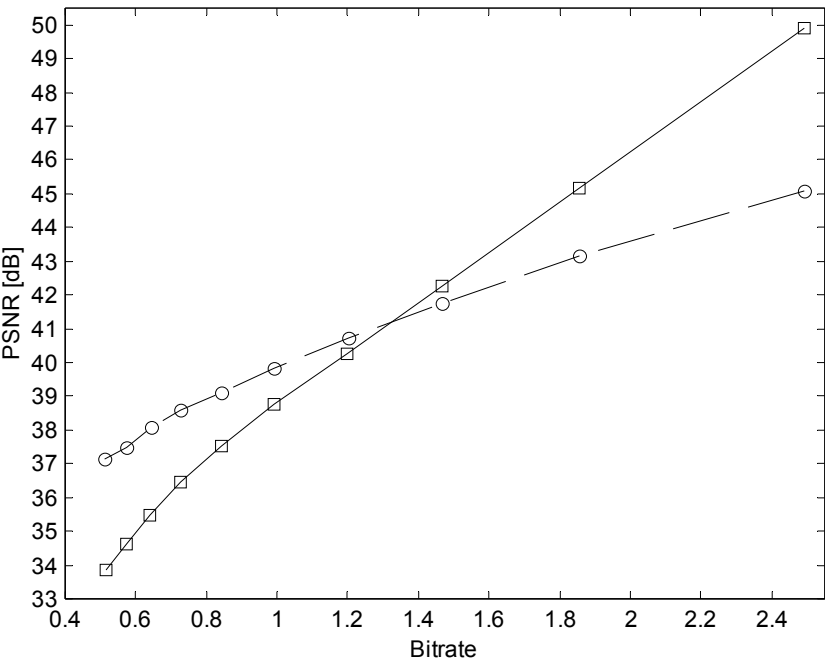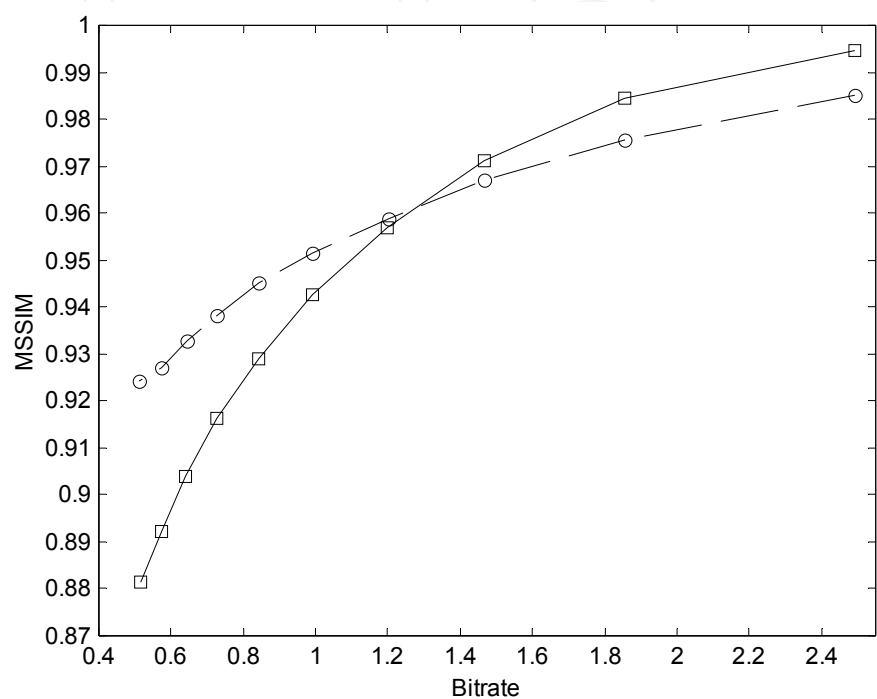

Fig. 2. Comparison of the PSNR of the Lennagrey image with respect to the average bitrate for JPEG2000 (the dashed line) and the Blend-V method in the near-lossless mode with $d \leq 10$ (the solid line)

In Fig. 2 and 3, there is a comparison of PSNR and MSSIM, respectively, between Blend-V and JPEG2000. For majority of the benchmark images the values of PSNR and MSSIM

obtained with the Blend-V method with $d \le 3$ is higher than with the JPEG2000 standard. It is worth stressing that for the wavelet method it is impossible to define the maximal error value d a priori. Under the same average bitrate, the maximum error d is considerably higher in the case of JPEG2000 (see Table 2). Considering the average bitrate and better values of PSNR and MSSIM, Blend-V should be the method of choice when the average bitrate is higher than 1.5 bit per pixel.

Fig. 3. Comparison of the MSSIM of the Lennagrey image with respect to the average bitrate for JPEG2000 (the dashed line) and the Blend-V method in the near-lossless mode with $d \le 10$ (the solid line)

Table 3 includes the experimental results of grayscale images for $d = \{2, 10\}$. It is compared the proposed Blend-V method (in the intraframe mode) with two other techniques, known from literature: LOCO-I (Weinberger et al., 2000) and TMW (Meyer & Tischer, 1997). The obtained results show the high efficiency of the Blend-V method not only in the lossless mode, but also in the near-lossless one, where it turns out to be competitive even with TMW, one of the most efficient techniques, but characterized with high computational complexity.

In the case of encoding color images in Blend-V in the lossless mode, the YD_bD_r transform, known from the JPEG2000 standard, is used. This transformation also requires only the addition, subtraction, and bit shifting operation. The equations for this transformation are as follows:

$$
\begin{aligned}
Y &= \left\lfloor \frac{R + 2G + B}{4} \right\rfloor \\
D_b &= B - G \\
D_r &= R - G
\end{aligned}
\qquad . \tag{12}
$$

Image	$d = 2$			$d = 10$		
	LOCO-I v.0.90N	TMW (mode 3)	Blend-V	LOCO-I v.0.90N	TMW (mode 4)	Blend-V
Balloon	1.242	0.90	0.905	0.49	0.32	0.198
Boats	1.902	1.65	1.646	0.78	0.66	0.546
Gold	2.333	2.19	2.166	0.99	0.81	0.710
Airplane	1.84	1.64	1.660	0.72	0.57	0.517
Baboon	3.72	3.49	3.481	1.91	1.68	1.656
Lennagrey	2.09	1.83	1.855	0.93	0.55	0.516
Peppers	2.29	2.09	2.096	0.93	0.64	0.558
Shapes	0.79	0.75	0.709	0.47	0.58	0.320
Bridge256	3.49	3.38	3.325	1.73	1.63	1.570
Camera256	2.28	2.08	2.024	0.96	0.90	0.795
Couple256	1.82	1.60	1.606	0.86	0.70	0.559
Average	2.163	1.964	1.952	0.979	0.822	0.722

Table 3. Average bitrates in the near-lossless mode with error value $d = \{2, 10\}$

The reverse transform is of the following form:

$$G = Y - \left\lfloor \frac{D_b + D_r}{4} \right\rfloor$$
$$B = D_b + G$$
$$R = D_r + G$$

. (13)

However, in the near-lossless mode, there is no color transformation. In Table 4, it is presented an efficiency comparison between the proposed Blend-V method with the hardware compression realization system Enhanced Lossless Image Compression (ELIC) from GEMAC (Dittrich, 2005). In table, there is the compression ratio measurement of five color images in the lossless ($d = 0$) and near-lossless mode.

Image	Blend-V $d = 0$	ELIC $d = 0$	Blend-V $d = 1$	ELIC $d = 1$	Blend-V $d = 2$	ELIC $d = 2$	Blend-V $d = 4$	ELIC $d = 4$	Blend-V $d = 8$	ELIC $d = 8$
lena	1.869	1.33	2.884	1.95	3.788	2.40	5.634	3.14	10.364	4.34
monarch	2.741	1.84	3.841	2.76	5.298	3.34	8.383	4.20	13.515	5.68
peppers	2.502	1.69	3.560	2.57	4.826	3.18	7.530	4.03	14.432	5.44
sail	2.322	1.50	2.444	2.16	3.072	2.58	4.187	3.16	6.233	4.03
tulips	2.472	1.63	3.420	2.44	4.519	2.97	6.572	3.77	10.716	4.89
Average	2.381	1.598	3.230	2.376	4.300	2.894	6.461	3.660	11.052	4.876

Table 4. Compression ratio of color images in the near-lossless mode with error value $d = \{0, 1, 2, 4, 8\}$

4. Video sequence encoding

Although there exist a few non-linear editing systems (NLEs) offering lossless video compression, some of them do not work in real-time, whereas others split video data in long

sequences and are not capable of decoding the n-th frame in a sequence without decompressing of n-1 previous frames (Ohanian, 1998). These requirements are covered in the proposed approach. The basic rule of a lossless video sequences compression is exploiting both the spatial dependencies (the intraframe mode - in the currently encoded image) and the temporal dependencies (the interframe mode), taking into account the neighboring frames of the sequence. The most popular is encoding of so-called Group of Pictures (GoPs), where the first frame is encoded in the intraframe mode, and the remaining N - 1 frames - in the interframe mode. This technique facilitates a quick access to any video sequences fragment with the accuracy up to N frames. When the whole sequence is encoded in the interframe mode, the correct reconstruction of the last frame is possible only after decoding all the previous frames.

According to (Andriani et al., 2005), only a low number of the papers about lossless video compression is available. Consequently, it is difficult to make some comparison in a solid way. Thus the efficiency of the proposed method has been earlier assessed mainly in the intraframe mode.

In this section, we present the further compression efficiency improvement thanks to the interframe mode introduction.

Among lossy video sequence compression methods, the most popular technique for finding the dependencies between the neighboring frames is to split the image into square blocks (usually 8×8 pixels), for which individual movement vectors are determined, and then the information about shifting according to the previous frame is added to each encoded block. This technique used for a lossless encoding is presented in (Matsuda et al., 2004), but this method is characterized with rather high computational complexity. In (Carotti et al., 2004), it is described a method for expecting a value of the current pixel based on the pixel from the previous frame, whose coordinates are calculated using the movement vector computed from the two previous frames, whereas five reference frames are used in (Maeda et al., 2006).

Each of these methods requires an extra computational expenditure due to the determining the movement vectors. Moreover, one has to consider also a quite complex mechanism for detecting a scene change, which should activate an entrance into the intraframe mode while encoding the first frame from each new scene (Yang & Faryar, 2000). In (Andriani & Calvagno, 2007), the authors resigned from the movement compensation obtaining quite promising results at the scene changes due to rather computationally complex technique named Octopus.

In our proposal, there is also no movement vector analysis; the encoding is performed with the usage of 13 subpredictors operating in the intraframe mode and one referring to the previous frame. As the predictor of the interframe mode, it is used a value of the pixel $P_{j-1}(0)$ from the previous frame of the coordinates the same as in the encoded pixel, $P_j(0)$. The blending prediction method automatically reduces the negative impact of the interframe mode predictor in the situation of the scene change. It is worth stressing that the usage of a simple predictor, which does not take into consideration the movement vector, leads to surprisingly good results. It results from the fact that in numerous situations big image fragments are stable (usually a background of the stage). On the basis of the analysis of the video sequences set, a compromise value of the importance of the interframe predictor is α = 6. The influence of the number of frames forming the GoP, N, on the average bitrate of the sequence for the Y component is presented in Fig. 4.

In Table 5, it is presented an example of the compression efficiency comparison between the proposed system, denoted here as Blend-V, with the results of the JPEG-LS and CPC encoders (Yang & Faryar, 2000). The results consider 30 frames of the *Y* luminance signal of the Salesman sequence. The experiment has been conducted for both the lossless and near-lossless modes with parameter d = {1, 2, 3, 4}. The measurement of the Blend-V method is taken in the intraframe mode (N = 1), and form the interframe mode with parameters N = 10 and N_{max} = 30, where N_{max} denotes that in the whole sequence only the first frame is to be encoded in the intrafame mode.

When designing a hardware module it is necessary to determine the trade-off between the compression and the module efficiency so that it can operate in real-time. Considering the dependencies between frames during decoding, we have decided to introduce a pipeline of length N = 10 for encoding/decoding separate frames in an image group. Larger value of N does not improve the efficiency significantly, whereas higher number of pipeline stages would result in further increase of hardware resource utilization and a more difficult access to any fragment of a decoded sequence.

d	JPEG-LS	CPC	Blend-V intraframe	Blend-V $N = 10$	Blend-V N_{max}
0	4.377	3.760	4.102	3.686	3.656
1	2.872	2.321	2.623	2.225	2.197
2	2.243	1.765	2.011	1.647	1.620
3	1.868	1.453	1.644	1.309	1.286
4	1.617	1.252	1.393	1.071	1.048

Table 5. Average bitrates for the Salesman video sequence (30 frames, 352x288 resolution) for various maximal error parameter, *d*, using the near-lossless mode
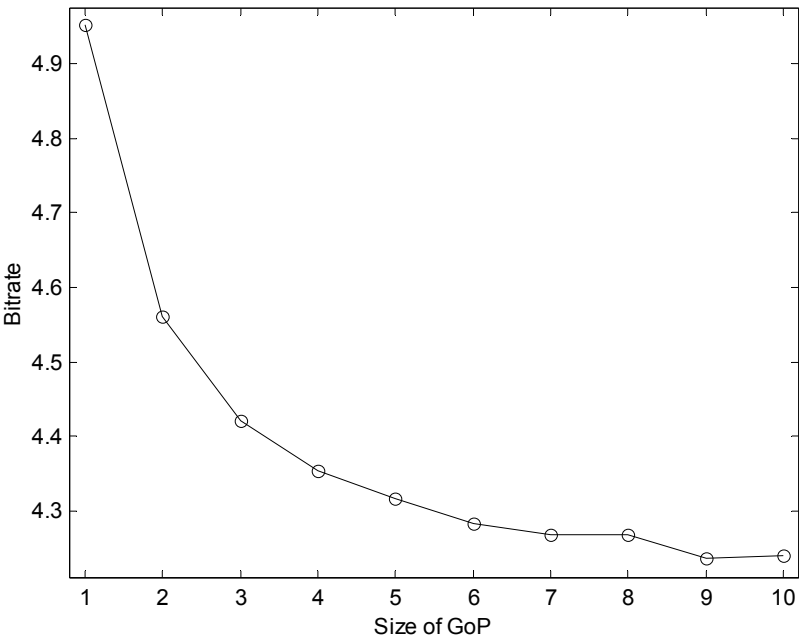

Fig. 4. Average bitrate of the Tennis video sequence with respect to the number of frames, *N*, in GoP

5. System-level hardware model

One of disadvantages of the prediction-based methods is a sequential image decoding that makes it impossible to perform computation in parallel (i.e., simultaneously decode more than one pixel). This property is caused with maintaining the principle of causality, where to decode pixel $P(0)$ it is required to have the value of its predecessor $P(1)$ (in general, values of the pixels situated directly on the left side and the rows above the $P(0)$ pixel).

In order to confirm the benefits of hardware realization of the proposed technique, we prepared its system level model; the individual stages of the algorithm are implemented as an MPSoC cores connected with a regular 2D mesh. As the communication infrastructure we utilize the Network on Chip (NoC) paradigm where the cores follow the GALS (Globally Asynchronous, Locally Synchronous) synchronization scheme, i.e., each core is equipped with its individual clock and the clocks communicate each other in the asynchronous manner. Besides, each core is equipped with a router for determining the next-hop port for the data to be sent. We decided to use the wormhole routing scheme, where data is sent in packages split into smaller portion of equal length, flits (flow control digits). The first flit of each package is used by the routers to select the route; the remaining packages follows the same path. For more details about our technique, the reader is referred to (Ulacha & Dziurzański, 2009).

We decomposed the algorithm in the following way. The subpredictor values are computed by separate cores in the parallel mode. Having computed these values, each core sends the obtained data to the core realizing the blending procedure. Then the data is transmitted into the core realizing arithmetic encoding.

As in the mesh architecture each core (except the boundary ones) is connected with its four neighbours only, it is crucial to map the functionalities into cores in a meticulous way so that the cores sending each other vast amount of data are located close to each other. Since in our case the amount of cores is relatively low, we managed to check all the possible permutation of the cores with regards to their NoC node mappings and determined the mappings leading to the lowest traffic in the network. After choosing the NoC router architecture, the routing type and the core mapping into mesh nodes, we could start with preparing a system-level model.

The model has been written in the SystemC language at the bus cycle accurate (BCA) level of abstraction and tested with CoCentric® System Studio by Synopsys™ [1]. According to the simulation, the system operates in real-time, confirming our assumptions described in section 1.

6. Conclusions

In this paper, a motivation for lossless and near-lossless compression of images and video sequences has been provided. The state of the art of modern compression methods considering their implementation complexity has been described. A technique for blending predictors as an new effective modeling method, which in combination with an adaptive arithmetic encoder allows us to obtain high compression ratio of video sequences. The proposed method leads also to high efficiency in the near-lossless mode.

The Blend-V benefits from various aspects connected with its hardware realization in a novel architecture based on Network on Chip (NoC) (Ulacha & Dziurzański, 2008).

---

[1] Synopsys and the Synopsys product names described herein are trademarks of Synopsys, Inc.

7. Acknowledgment

## 8. References

Andriani, S.; Calvagno, G.; Erseghe, T.; Mian, G.A.; Durigon, M.; Rinaldo, R.; Knee, M.; Walland, P. & Koppetz, M. (2004). Comparison of lossy to lossless compression techniques for digital cinema, *Proc. International Conference on Image Processing*, pp.513-516, ISBN 0-7803-8554-3

Andriani, S.; Calvagno, G. & Mian, G.A. (2005). Lossless Video Compression using a Spatio-Temporal Optimal Predictor, *Proc. 13th European Signal Processing Conference*, on CD.

Andriani, S. & Calvagno, G. (2007). Lossless Compression of Colour Video Sequence using Optimal Prediction Theory - Octopus, *Proc. Data Compression Conference*, pp. 375-375, ISBN 0-7695-2791-4

Carotti, E.S.G.; De Martin, J.C. & Meo, A.R. (2004). Backward-Adaptive Lossless Compression of Video Sequences, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol.4, pp. 3417-3420, ISBN 0-7803-7402-9

Chen, X.; Canagarajah, C.; N., Vitulli, R. & Nunez-Yanez, J. L. (2008). Lossless Compression for Space Imagery in a Dynamically Reconfigurable Architecture, *Proc. International Workshop on Applied Reconfigurable Computing*, LNCS Vol.4943, pp. 336-341, DOI 10.1007/978-3-540-78610-8_38

Deng, G. & Ye, H. (1999). Lossless image compression using adaptive predictor combination, symbol mapping and context filtering, *Proc. IEEE 1999 International Conference on Image Processing*, Vol.4, pp. 63-67, ISBN 0-7803-5467-2

Deng, G. & Ye, H. (2003). A general framework for the second-level adaptive prediction, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol.3, pp. 237-240.

Dittrich, C. (2005). FPGAs for lossless image/video and universal data compression, *ECE Magazine*, Vol.4, pp. 42-44.

Drost G. W. & Bourbakis N. G. (2001). A hybrid system for real-time lossless image compression, *Microprocessors and Microsystems*, Vol.25, No.1, pp. 19-31, DOI 10.1016/S0141-9331(00)00102-2

Gallager R. G. (1978). Variations on a theme by Huffman, *IEEE Transactions on Information Theory*, Vol.24, No.6, pp. 668-674, ISSN: 0018-9448

Jiang, J. & Grecos, C. (2002). Towards an improvement on prediction accuracy in JPEG-LS, *Optical Engineering, SPIE*, Vol.41, No.2, pp. 335-341, ISSN 0091-3286

Maeda, H.; Minezawa, A.; Matsuda, I. & Itoh, S. (2006). Lossless Video Coding Using Multi-Frame MC and 3D Bi-Prediction Optimized for Each Frame, *Proc. 14th European Signal Processing Conference*, on CD.

Marcellin, M.; Gormish, M.; Bilgin, A. & Boliek, M. (2000). An Overview of JPEG-2000. *Proc. Data Compression Conference*, pp. 523-541.

Marusic, S. & Deng, G. (2002). Adaptive prediction for lossless image compression, *Signal Processing: Image Communications*, Vol.17, pp. 363-372, DOI 10.1016/S0923-5965(02)00006-1

Matsuda, I.; Shiodera, T. & Itoh, S. (2004). Lossless Video Coding Using Variable Block-Size MC and 3D Prediction Optimized for Each Frame, *Proc. European Signal Processing Conference*, pp. 1967-1970, ISSN 0913-5685

Matsuda, I.; Ozaki, N.; Umezu, Y. & Itoh S. (2005). Lossless coding using Variable Blok-Size adaptive prediction optimized for each image, *Proc. 13th European Signal Processing Conference*, on CD.

Meyer, B. & Tischer, P. (1997). TMW – a new method for lossless image compression, *Proc. International Picture Coding Symposium*, pp. 533-538, Berlin, Germany.

Meyer B. & Tischer P. (2001a). Glicbawls, Grey Level Image Compression by Adaptive Weighted Least Squares, *Proc. of Data Compression Conference*, p. 503, ISBN 0-7695-1031-0

Meyer, B. & Tischer, P. (2001b). TMWLego - An Object Oriented Image Modelling Framework, *Proc. Data Compression Conference*, p. 504, ISBN 0-7695-1031-0

Ohanian T. (1998). *Digital Nonlinear Editing: Editing Film and Video on the Desktop*, Focal Press, ISBN 978-0240802251

Sayood, K. (2002). *Introduction to Data Compression*, 2nd ed., Morgan Kaufmann Publishers, ISBN 978-1558605589

Seemann, T. & Tisher, P. (1997a). *Generalized locally adaptive DPCM*, Department of Computer Science Technical Report CS97/301, pp. 1-15, Monash University, Australia.

Seemann, T.; Tisher, P. & Meyer, B. (1997b). History-Based Blending of Image Sub-Predictors, *Proc. Picture Coding Symposium*, pp. 147-151.

Strutz, T. (2002). Context-Based Adaptive Linear Prediction for Lossless Image Coding, *Proc. 4th International ITG Conference on Source and Channel Coding*, pp. 105-109.

Ulacha, G. & Stasiński, R. (2008). Predictor Blending for Real-Time Lossless Coding System, *50th International Symposium ELMAR'08*, pp. 61-64, 10-13 September 2008, Zadar, Croatia.

Ulacha, G. & Dziurzański, P. (2008). Blending-prediction-based approach for lossless image compression, *Proc. 1st International Conference on Information Technology*, pp. 471-474.

Ulacha, G. & Dziurzański, P. (2009). Lossless and near-lossless image compression scheme utilizing blending-prediction-based approach, *Proc. International Conference on Computer Vision and Graphics*, LNCS Vol.5337, pp. 208-217, ISSN 0302-9743

Wang, H. & Zhang, D. (2004). A linear edge model and its application in lossless image coding, *Signal Processing: Image Communication*, Vol.19, pp. 955-958, DOI 10.1016/j.image.2004.04.006

Wang, Z.; Bovik, A. C.; Sheikh, H. R. & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing*, Vol.13, No.4, pp. 600-612, DOI 10.1109/TIP.2003.819861

Weinberger, M. J.; Seroussi, G. & Sapiro, G. (2000). LOCO-I: Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS, *IEEE Trans. on Image Processing*, Vol.9, No.8, pp. 1309-1324, ISSN 1057-7149

Wu, X. & Memon, N. D. (1996). CALIC – A Context Based Adaptive Lossless Image Coding Scheme, *IEEE Trans. on Communications*, Vol.45, pp. 437-444, ISSN 0916-8516

Xie, X.; Li, G.L. & Wang, Z.H. (2007). A Near-Lossless Image Compression Algorithm Suitable for Hardware Design in Wireless Endoscopy System, *EURASIP Journal on Advances in Signal Processing*, (2007) pp. 48-61, DOI 10.1155/2007/82160

Yang, K. H. & Faryar, A. F. (2000). A contex-based predictive coder for lossless and near-lossless compression of video, *Proc. International Conference on Image Processing*, Vol.1, pp. 144-147, ISBN 978-1-4244-7994-8

Ye, H. (2002). *A study on lossless compression of greyscale images*, Ph. D thesis, Department of Electronic Engineering, La Trobe University.

Ye, H.; Deng, G. & Devlin, J. C. (2000). Adaptive linear prediction for lossless coding of greyscale images, *Proc. IEEE International Conference on Image Processing*, on CD.

CCSDS (2007). *Lossless Data Compression. Recommendation for Space Data System Standards, CCSDS 120.1-G-1* Green Book, 1.
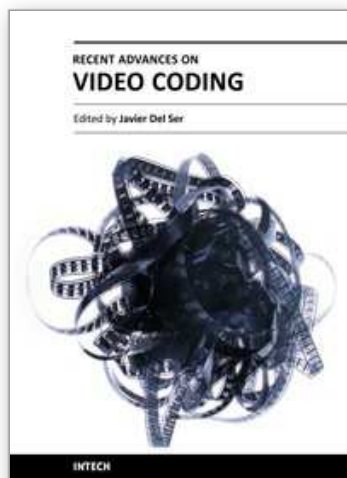
Recent Advances on Video Coding

Edited by Dr. Javier Del Ser Lorente

ISBN 978-953-307-181-7

Hard cover, 398 pages

Publisher InTech

Published online 24, June, 2011

Published in print edition June, 2011

This book is intended to attract the attention of practitioners and researchers from industry and academia interested in challenging paradigms of multimedia video coding, with an emphasis on recent technical developments, cross-disciplinary tools and implementations. Given its instructional purpose, the book also overviews recently published video coding standards such as H.264/AVC and SVC from a simulational standpoint. Novel rate control schemes and cross-disciplinary tools for the optimization of diverse aspects related to video coding are also addressed in detail, along with implementation architectures specially tailored for video processing and encoding. The book concludes by exposing new advances in semantic video coding. In summary: this book serves as a technically sounding start point for early-stage researchers and developers willing to join leading-edge research on video coding, processing and multimedia transmission.