ENTROPY BASED MERGING OF CONTEXT MODELS FOR EFFICIENT ARITHMETIC CODING

Tilo Strutz

Deutsche Telekom, Leipzig University of Telecommunications
Institute of Communications Engineering
Gustav-Freytag-Str. 43–45, 04277 Leipzig, Germany

ABSTRACT

The contextual coding of data requires in general a step which reduces the vast variety of possible contexts down to a feasible number. This paper presents a new method for non-uniform quantisation of contexts, which adaptively merges adjacent intervals as long as the increase of the contextual entropy is negligible. This method is incorporated in a framework for lossless image compression. In combination with an automatic determination of model sizes for histogram-tail truncation, the proposed approach leads to a significant gain in compression performance for a wide range of different natural images.

Index Terms— context quantisation, modelling, arithmetic coding, image compression

1. INTRODUCTION

Let X be a source producing K different symbols s_i $(i = 1, \dots, K)$. Then the source entropy is computed based on the probabilities $p(s_i)$ of these symbols by

$$H(X) = -\sum_{i=1}^{K} p(s_i) \cdot \log_2(p(s_i)) \; . \tag{1}$$

Under the assumption that there are no dependencies between the symbols, at least $H(X)$ bits must be spent for one symbol on average for transmitting or storing a symbol sequence produced by this source. In many applications, however, symbol probabilities depend on some conditions. These conditions form a context C and the probabilities become *contextual probabilities* $p(s_i|C)$. The entropy for each subcontext C_j is given by

$$H(X|C_j) = -\sum_{i=1}^{K} p(s_i|C_j) \cdot \log_2(p(s_i|C_j)) \tag{2}$$

and the *contextual entropy* of the source X is the weighted average of these single, subcontext-related, entropies

$$H(X|C) = -\sum_{j} p(C_j) \cdot H(X|C_j) \; , \tag{3}$$

with $p(C_j)$ being the probability of subcontext C_j. As long as the symbol probabilities are affected by the context C, $H(X|C) < H(X)$ holds and fewer bits are required for storing or transmitting.[1]

[1] In the case when the context of the current symbol is simply constituted by the sequence of its predecessors, i.e. $p(s^m|C) = p(s^m|s^{m-1}s^{m-2}\cdots s^0)$, the contextual probabilities are called *conditional probabilities*, $H(X|C) = H(X|X^{(m)})$ is called *conditional entropy*, and the source could be described by a Markovian model of mth order.

The problem of context modelling was addressed in a rigorous manner for the first time in [1]. It turned out, however, that in applications with large symbol alphabets, and especially if the symbols have a physical meaning, this tree-based method causes too high costs compared to methods utilising some prior knowledge [2]. This prior knowledge should be part of the context C or can, as for example in lossless image compression, be used for a preprocessing step (e.g. prediction of signal values).

The context C can have an arbitrarily high complexity. The crucial task in practical application of data compression is to reduce this complexity down to a feasible order. This process of reduction is called context quantisation. Let $Q(C)$ be the quantised version of C, then $H(X|Q(C)) \geq H(X|C)$ holds. The aim is to find a practicable, finite set of subcontexts C_j making $H(X|Q(C))$ as small as possible. The knowledge about the contextual probabilities $p(s_i|C_j)$ can be used for efficient arithmetic coding.

In principle, context quantisation requires two steps. The first step identifies dependencies between accessible information and the symbols to be encoded, which can then be utilised for the context constitution. This task is mainly application-driven and will be called modelling in the sequel. The second step must limit the number of different contexts C_j avoiding the problem of context dilution, which appears when count statistics are spread over too many contexts [3]. In the process of symbol coding, the arithmetic coder ideally selects the distribution based on the context C_j.

In dependence on how many conditions are involved in context formation, the modelling is typically a multidimensional problem and the reduction of the number of subcontexts must be solved by any kind of vector quantisation [4]. In image coding, these conditions (mostly prediction errors in the causal neighbourhood) are often combined leading to a scalar value [5, 6, 7, 8]. This combination typically simplifies the process to non-uniform (scalar) quantisation. In addition to the prediction errors, also textural information can be exploited [9]. The approaches in [10, 11] even use merely a uniform quantiser. Other approaches convert the vector quantisation into a combination of several non-uniform scalar quantisers [3]. No quantisation of the scalar value is required at all when the conditions are solely used to find online the distribution of the symbols the arithmetic coder has to work with [12, 13].[2] In [14], the context quantisation is discussed with respect to the zero coding in the framework of JPEG2000.

This paper presents a novel technique for the reduction of the context number, mapping the vector-quantisation problem into non-uniform scalar quantisation. The application of lossless image com-

[2] Strictly speaking, the quantisation is performed at that moment, when the continuous distribution model is mapped to the discrete representation of the arithmetic coder intervals.

pression is addressed. As in [8], the proposed method does not use fixed thresholds for the determination of the coding contexts, but computes them adaptively for each single image. This requires an initial pass at the encoder and the thresholds have to be transmitted as side information (overhead). In addition, also the model sizes of the symbol distributions are adaptively defined improving the updating process in the arithmetic coding stage.

The paper is organised as follows: Section 2 describes the application-specific context modelling and the novel process of reducing the number of contexts. Section 3 discusses how the compression scheme utilises the context information and describes some coding details. Section 4 presents the results showing the influence of the proposed methods also in comparison to the state-of-the-art methods, and a summary is given in Section 5 .

2. MODELLING AND CONTEXT QUANTISATION

2.1. Application-specific context modelling

The application in mind is lossless image compression using a context-based linear prediction similar to CoBaLP proposed in [15]. In contrast to CoBaLP, we use a two-pass scheme and adaptively determine the prediction contexts based on the texture in the image.

As large prediction-error magnitudes tend to cluster in certain image regions (and small ones in other regions), the magnitudes of errors in the causal neighbourhood of the current position are correlated to the magnitude of actual prediction error. In [9], it was already mentioned that the current value also depends on the texture of the original image data surrounding the current position. That is why we combine both types of information. The estimate of the current prediction-error magnitude is computed as

$$|\hat{e}_0| = \frac{\sum_{i \in T} w_i \cdot |e_i| + w_{px} \cdot \overline{|e_{px}|}}{\sum_{i \in T} w_i + w_{px}} \tag{4}$$

The index px corresponds to the prediction context under which the current prediction was made. It should be remarked that the prediction contexts are built based on the texture of the original image data and should not be mixed with the coding context, which will be determined based on equation (4).

The last summand in (4) expresses the textural dependency and corresponds to the average of absolute errors occurring in context px up to the actual position

$$\overline{|e_{px}|} = \frac{1}{\text{count}(px)} \cdot \sum_{j \in px} |e_j| \,. \tag{5}$$

The magnitudes of prediction errors $|e_i|$ are taken from the causal neighbourhood defined by the template $T = \{A, B, \ldots, R\} = \{|e_i|\}$ (**Fig. 1**). The weights w_i are empirically set to the values shown in **Fig. 2** and turned out to be superior compared to other settings as in [7] and [8], where the neighbours are weighted based on their Euclidean distances to the current position. The influence of the closest neighbours is triggered by a kind of directional information based on the absolute gradients $|C - B|$ and $|C - A|$.

The textural information is not only involved via $w_{px} \cdot \overline{|e_{px}|}$ with an empirical value of $w_{px} = 0.3 \cdot \sum_{i \in T} w_i$, but is also taken into account in a second manner. In such a case when the prediction context at the current position is identical to the context at other positions within the template T, the weights w_i of the corresponding

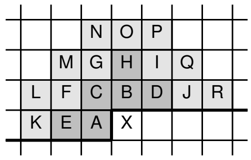

Fig. 1. Template T of prediction-errors magnitudes, which are used for the coding-context determination

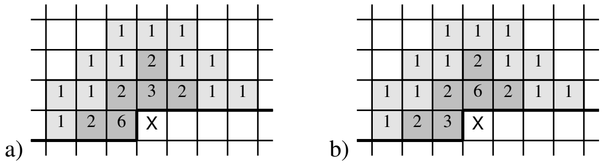

Fig. 2. Weights w_i for prediction-error magnitudes in template T, a) if $|C - B| < |C - A|$, b) if $|C - B| > |C - A|$. If $A == B$, then $w_\text{A} = w_\text{B} = 6$.

magnitudes are increased, by two for $A, B, C,$ and D and by one for all other.

Using (4), the multidimensional problem based on the elements $|e_i|, \forall i$ and $|e_{px}|$ is mapped to a one-dimensional problem based on the scalar value $|\hat{e}_0|$.

2.2. Adaptive reduction of the number of contexts

The computed value of $|\hat{e_0}|$ (see eq.4), gives a good estimation of the true magnitude of the current prediction error. The estimate could be used as a parameter for a pre-defined distribution-model function. This approach is followed in [13], for example, and no quantisation of $|\hat{e_0}|$ would be required. The main disadvantage therein is that this distribution might be suitable for a certain class of images but might not be for others. This could be taken into account with additional parameters making the distribution model more flexible [8]. The alternative is to adaptively create the distribution of the true e_0 based on the samples which have been already processed. This is realised with a histogram h_{C_j}, containing the counts of all sample values which occurred for a context C_j up to the current position. As the distribution has to be dependent on the estimates $|\hat{e}_0|$, we need a function which maps the estimate to one of a limited number of different distributions, or more precisely to a histogram h_{C_j}. With respect to the derivations in Section 1, $|\hat{e}_0|$ has to control the selection of a certain subcontext C_j in such a manner that the context-related entropies $H(X|C_j)$ remain small.

This paper proposes a two-step method. At first, $|\hat{e}_0|$ is uniformly quantised into $K = 10 \cdot \text{range}(|e_i|)$ intervals. The value $\text{range}(|e_i|)$ is equal to $(x_\text{max}+1)/2+1$ and the factor of 10 merely guarantees that the initial granularity is fine enough.[3]

The second step merges adjacent intervals based on an entropy criterion. Let $q = 0, 1, 2, \ldots, K-1$ be the interval numbers. Based on the count statistics of the absolute prediction errors $|\hat{e}_0|$ in each of the K intervals, the entropies $H(X|q)$ are calculated. Then those entropies $H(X|q, q+1)$ are computed which result after merging the count statistics of adjacent intervals q and $q + 1$. The cost of

[3]In case that there are $x_\text{max} + 1$ different values in the image signal, the prediction errors $e[n] = x[n] - \hat{x}$ can be mapped into the range of $-(x_\text{max} + 1)/2 \leq e[n] \leq x_\text{max}/2$. Taking the absolute value of $e[n]$ results to a range of $0 \leq |e_i| \leq (x_\text{max} + 1)/2$.

Table 1. Examples of adaptive context quantisation and model sizes

	RNAi_dna_12bit		kodim01		woman_G	
	thresh.	size	thresh.	size	thresh.	size
cx	for $\|\hat{e}_0\|$	$M[cx]$	for $\|\hat{e}_0\|$	$M[cx]$	for $\|\hat{e}_0\|$	$M[cx]$
0	13.7	16	1.2	2	0.7	2
1	16.7	21	1.5	2	1.0	2
2	45.2	37	2.0	3	1.2	2
3	78.0	98	2.4	3	1.5	3
4	121.4	180	3.0	4	1.8	3
5	158.8	259	3.6	5	2.1	3
6	202.6	309	4.3	6	2.3	3
7	–	412	5.3	8	2.5	4
8	–	–	6.4	10	2.7	4
⋮	–	–	⋮	⋮	⋮	⋮
18	–	–	–	54	10.9	18
⋮	–	–	⋮	⋮	⋮	⋮
22	–	–	–	–	26.7	33
23	–	–	–	–	–	43

merging the two intervals is

$$J(q, q+1) = H(X|q, q+1) \cdot (n_q + n_{q+1}) - \\ [H(X|q) \cdot n_q + H(X|q+1) \cdot n_{q+1}] \ , \tag{6}$$

with n_q and n_{q+1} being the numbers of samples belonging to interval q or $q+1$, respectively. The two intervals with the smallest merging costs are finally combined, the total number K of intervals is decremented, and this process is iteratively continued until the smallest cost $J^{(l)}$ in iteration number l exceeds a threshold, i.e. the iteration stops if

$$J^{(l)} \geq J^{(l-1)} \cdot 1.2 \ . \tag{7}$$

A safe-guard procedure ensures that the iteration is neither stopped too early ($J^{(l-1)}$ must be larger than a certain value and the number of remaining contexts has fallen below a limit) nor to late (at least two intervals must survive). The maximum number of coding context is limited to 40. The factor of 1.2 in (7) balances between increasing contextual entropy and coding costs caused by (i) the successive update of the internal distributions of the arithmetic coder and (ii) the side information (the interval borders), which has to be transmitted.

This procedure typically results in 10–30 intervals (i.e., coding contexts C_j) for natural (non-synthetic) images. The thresholds between the intervals, however, are quite different, **Table 1**. Non-photographic natural images can show different characteristics leading to different settings, as RNAi_dna_12bit, for example.

3. CODING ASPECTS

Let us assume that the entire process of application-specific modelling and context quantisation led to the contexts $C_j = cx = 0, 1, \ldots, K-1$. The mapping $|\hat{e}_0| \mapsto cx$ typically results in sorted contexts, i.e., the higher cx, the higher the variance of the true prediction errors e_0 which are assigned to cx. The theoretical range of possible prediction errors in each context is $-(x_{\max}+1)/2 \leq e_0 \leq x_{\max}/2$, i.e., the alphabet of symbols comprises $x_{\max}+1$ different symbols.

As already mentioned above, the distribution model h_{cx} (the histogram) is built up based on the counts of all samples e_0 already processed. As the counts of all symbols must be initialised to a value of at least one in the beginning of arithmetic coding, the histogram can be significantly distorted reducing the coding efficiency, especially for narrow distributions without long tails. Instead of using the same alphabet for each context, we therefore propose to reduce the range to $-M[cx] \leq e_0 \leq M[cx]$ with $M[cx]$ chosen in such a manner that at least $13/16$ of all samples of cx are included. Assuming a Laplacian distribution, this corresponds to a range of $\pm\sigma$. The technique of limiting the alphabet size is known as tail truncation [9]. In contrast to former approaches (e.g. [9, 16]), which use fixed truncation thresholds, the proposed scheme adaptively determines the thresholds $M[cx]$ as described above for each single image and they have to be transmitted to the decoder.

The symbols to be encoded are $s = e_0 + M[cx] + 1$ with an alphabet size of $2 \cdot (M[cx]+1)$. If the magnitude of e_0 is too large ($s < 1$ or $s > 2 \cdot M[cx]+1$, $s = 0$ is transmitted instead, signalling an exception handling.

Symbols which do not fit the selected distribution model show different statistics. That is why a second set of distribution models $g[cx]$ with same sizes $M[cx]$ is prepared. As soon as the exception handling is activated, the context number is incremented $cx' = cx + 1$ (resulting in a model with possibly larger alphabet) and the corresponding distribution model $g[cx']$ is used for encoding the symbol. As the prediction error did not fit the previous alphabet, its magnitude can be decreased to $|e_0'| = |e_0| - M[cx]$ in advance and the symbol is newly determined with $s' = e_0' + M[cx'] + 1$. The process is iterated until the symbol fits the alphabet of the selected model.

Dependent on the context px used in the prediction stage, the distribution of the prediction errors might be skewed to one or the other side. The final distribution within the coding context cx is a mixture of these skewed distributions. However, it can be narrowed if all contributions are skewed to the same side. This is realised by conditional flipping of the sign of e_0 before it is mapped to the symbol s. While former approaches uses the mean of the prediction errors (i.e., $\{\overline{e_0}|px\}$), we found that the counts of positive and negative values of e_0 within each px lead to better compression results.

4. INVESTIGATIONS

The influence of the new context quantisation and some coding aspects is listed in **Table 2**. In the left part, it shows the entropies of the prediction-error signals for some selected test images [17][4] and the bitrates when using a basic coding scheme. Then single techniques (tail truncation, usage of $g[cx]$, adaptive context quantisation, and sign flipping) are successively added. The last column shows the overhead, which has to be transmitted in the proposed scheme. It consists of a fixed part (15 bytes), 10-962 bytes for the adaptively generated prediction contexts, and 2-33 bytes for coding-context thresholds and model sizes $M[cx]$. When fixed (i.e., non-adaptive) context quantisation and/or full alphabet sizes are used, the corresponding overhead bytes need not be sent. For checkker_bw, xray_10bit, and RNAi_dna_12bit, an offline histogram packing ([19]) is adaptively activated leading to an additional overhead of 4, 112, and 490 bytes, respectively. The side information is generally transmitted using adaptive Rice coding.

[4] The RNAi image is taken from www.broadinstitute.org/bbbc/BBBC017 and was already used in [18]. However, there the data were scaled down to 8bpp.

Table 2. Entropy of prediction error images and final bitrates in bit per pixel for different settings. See text for details.

image	size	Entropy of $\{e[n]\}$	Bitrates basic coding	+ tail truncation	+ usage of g_{cx}	+ adaptive context quant.	+ signflip =proposed	Overhead [bytes]
barbara.y	720×576	4.236	4.022	3.997	3.998	3.976	**3.976**	315
kodim01	768×512	5.338	5.011	4.988	4.989	4.975	**4.973**	327
woman_G	2048×2560	4.540	4.163	4.161	4.161	4.148	**4.147**	1001
k05_Y	3072×2048	3.288	3.213	3.211	3.211	3.188	**3.188**	853
roebuck	1936×1288	3.577	3.225	3.219	3.219	3.190	**3.186**	614
checker	880×560	0.470	0.353	0.343	0.338	0.320	**0.314**	406
checker_bw	440×440	0.005	0.007	0.007	0.007	0.006	**0.006**	31
tree_flowers	1420×930	5.151	5.054	5.047	5.046	5.039	**5.039**	508
k05_U_9bit	3072×2048	2.490	2.381	2.379	2.378	2.334	**2.333**	606
xray_10bit	1576×1976	3.499	3.176	3.165	3.165	3.137	**3.135**	687
RNAi_dna_12bit	512×512	6.047	6.122	5.933	5.927	5.888	**5.888**	699

Table 3. Bitrates in bits per pixel for different approaches. (CALIC, MRP, and Blend-24 cannot process images with more than 8bpp.)

image	Bitrate in bpp CoBaLP2 proposed	Glicbawls [13]	CALIC [9]	MRP [8]	Blend-24 [21]
barbara.y	3.976	3.915	4.339	3.844	3.685
kodim01	4.973	5.082	5.091	4.961	4.890
woman_G	4.147	4.220	4.295	4.133	4.093
k05_Y	3.188	3.159	3.318	3.146	3.111
roebuck	3.186	3.155	3.343	3.083	3.070
chekker	0.314	1.984	0.163	0.164	0.232
chekker_bw	0.006	0.397	0.007	0.010	0.042
tree_flowers	5.039	5.085	5.276	4.971	4.938
average	**3.104**	**3.375**	**3.229**	**3.039**	**3.008**
k05_U_9bit	2.333	2.343	–	–	–
xray_10bpp	3.135	3.382	–	–	–
RNAi_dna_12bit	5.888	5.979	–	–	–

Table 4. Total coding times in seconds for the image 'barbara.y'

	CoBaLP2 proposed	Glicbawls [13]	CALIC [9]	MRP [8]	Blend-24 [21]
Enc.	37	9	< 1	214	659
Dec.	36	9	< 1	1	629

If $g[cx]$ is not used, the symbols s which do not fit the model size $M[cx]$ also are coded using the distribution $h[cx]$. The fixed non-uniform context quantisation uses eight intervals as in [9] instead of using the proposed adaptive merging of context intervals. If the tail truncation is disabled, all distributions $h[cx]$ use the full range of possible prediction errors. This implies automatically that the distributions $g[cx]$ are meaningless. The influence of $g[cx]$ is negligible, when a fixed context quantisation is used. In combination with the the adaptive mode, however, it becomes essential.

Table 3 shows the compression results in comparison to other state-of-the-art codecs. The Blend-24 codec is an advanced version of the compression scheme presented in [20].

The encoding and decoding times of all investigated compression schemes are listed in **Table 4**. They were measured on an Intel(R) Pentium(R) CPU G620 2.60GHz. MRP is the only non-symmetric compressor, which determines the prediction parameters on the encoder side and transmit them to the decoder along with the compressed data.

5. SUMMARY AND DISCUSSION

We have presented a new technique for the adaptive merging of coding contexts based on an entropy criterion with application to lossless image coding. Based on the complexity reduction via mapping the multi-dimensional problem to a one-dimensional, it has been verified that the data-specific determination of the context borders leads to an improvement in coding performance.

The context quantisation is accompanied by an adaptive determination of the alphabet size (tail truncation), which both together significantly benefit from the second set of distribution models $g[cx]$, which is used for symbols outside the truncated range of $h[cx]$. The sign flipping based on the counts of signs (instead of mean values) is another contribution of the presented work. The investigations have generally shown that a image-content adaptive processing increases the coding performance compared to fixed settings.

The proposed approach closes the gap between fast methods like CALIC and brute-force approaches as MRP and Blend-24.

Although the presented mapping to a one-dimensional context-quantisation problem already integrates several single components, it is expected that there is still some room for improvements. Especially, the optimal size of the template comprising the local neighbourhood heavily depends on the data characteristics. For very noisy data, the template should be larger and vice versa. Also the weights could be set even more adaptively.

6. REFERENCES

[1] Rissanen, J.: A universal data compression system. *IEEE Transactions on Information Theory*, Vol.29, No.5, Sep. 1983, 656–664

[2] Weinberger, M.J.; Rissanen, J.; Arps, R.B.: Applicatkons of Universal Context Modeling to Lossless Compression of Gray-Scale Images. *IEEE Transactions on Image Processing*, Vol.5, No.4, April 1996, 575–586

[3] Weinberger, M.J.; Seroussi, G.; Sapiro, G.: LOCO-I: A Low Complexity, Context Based, Lossless Image Compression Algorithm. *Proceedings of DCC'96*, 140–149

[4] Ginesta, X. ; Kim, S.P.: Vector quantization of contextual information for lossless image compression. *Proc. of DCC'94*, Snowbird, Utah, USA, 1994, 390–399

[5] Deng, G.: Symbol Mapping and Context Filtering for Lossless Image Compression. *Proceedings of ICIP'98*, Vol.1, 1998, 526–529

[6] Li, X.; Orchard, M.T.: Edge-Directed Prediction for Lossless Compression of Natural Images. *IEEE Transactions on Image Processing*, Vol.10, No.6, June 2001, 813–817

[7] Aiazzi, B. ; Alparone, L. ; Baronti, S.: Context modeling for near-lossless image coding. *IEEE Signal Processing Letters*, Vol.9, No.3, 2002, 77 – 80

[8] Matsuda, I.; Ozaki, N.;Umezo, Y.; Itoh, S.: Lossless Coding using variable block-size adaptive prediction optimized for each image. *Proc. of EUSIPCO 2005*, WedAmPO3, Sep. 2005

[9] Wu, X.; Memon, N.: Context-based, Adaptive, Lossless Image Codec. *IEEE Transactions on Communications*, Vol.45, No.4, April 1997, 437–444

[10] Howard, P.G.; Vitter, J.S.: Fast and Efficient Lossless Image Compression. *Proc. of DCC'93*, Snowbird, Utah, USA, 1993, 351–360

[11] Lee, W. S.: Edge-adaptive prediction for lossless image coding. *Proc. of DCC'99*, Snowbird, Utah, USA, 29-31 March, 1999, 483–490

[12] Meyer, B.; Tischer, P.: TMW – a new method for lossless image compression. *Proc. of PCS'97*, 10-12 September, Berlin, Germany, 533–538

[13] Meyer, B.; Tischer, P.: Glicbawls – Greylevel Image Compression By Adaptive Weighted Least Squares. *Proc. of DCC 2001*, 27-29 March, Snowbird, Utah, USA, 2001

[14] Liu, Z.; Karam L.J.: Contex formation by mutual information maximation. *Proc. of ICIP 2002*, 22-25 Sept. 2002, Vol.3, 89–92

[15] Strutz, T.: Context-Based Adaptive Linear Prediction for Lossless Image Coding. *Proc. of 4th Int. ITG Conf. on Source and Channel Coding*, Berlin, Germany, 28-30 January, 2002, 105–109

[16] Kau, L.-J., Lin, Y.P.: Least-Squares-Based Switching Structure for Lossless Image Coding, *IEEE Trans. on Circuits and Systems*, Vol.54, No.7, July 2007, 1529–1541

[17] http://www1.hft-leipzig.de/strutz/Papers/Testimages/CoBaLP2/ last visited: 11.10.2013

[18] Karimi, N.; Samavi, S.; Shirani, S.: Lossless Compression of RNAi Fluorescence Images Using Regional Fluctuations of Pixels. *IEEE Journal of Biomedical and Health Informatics*, Vol.17, No.2, March 2013

[19] Pinho, A.J.: An online preprocessing technique for improving the lossless compression of images with sparse histograms. *IEEE Signal Processing Letters*, Vol.9, No.1, Jan. 2002, 5–7

[20] Ulacha, G.; Stasinski, R.: Performance optimized predictor blending technique for lossless image coding. *Proc. of ICASSP 2011*, 22-27 May 2011, Prague, 1541–1544

[21] Ulacha, G.: Blend-24 compression software. private correspondence, September 2013