



An Adaptive Block-Based Histogram Packing for Improving the Compression Performance of JPEG-LS for Images with Sparse and Locally Sparse Histograms

Souha Jallouli^(✉), Sonia Zouari, Nouri Masmoudi, and Atef Masmoudi

Laboratory of Electronics and Technologies of Information, National Engineering School of Sfax, University of Sfax, 3038 Sfax, Tunisia
souha.jallouli@enis.tn, sonia.zouarikrichen@gmail.com,
masmoudi123@gmail.com, masatef@gmail.com

Abstract. In this paper, we propose a new adaptive block-based histogram packing for lossless compression of images with sparse and locally sparse histograms. JPEG-LS provides an efficient lossless compression at a reasonable complexity. However, its efficiency is severely affected when encoding images or blocks containing only a subset of the possible values from the nominal alphabet. The aim of this work is to improve the compression performance of JPEG-LS by introducing a preprocessing technique that leads to improvements. Results of its effectiveness are presented.

Keywords: Adaptive block-based histogram packing
Sparse and locally sparse histograms · Lossless compression
JPEG-LS · Offline histogram packing · Alphabet reduction scheme
Preprocessing technique

1 Introduction

To obtain practical imaging systems, image compression is an important tool for transmission and storage requirements. It reduces bandwidth on a transmission channel and provides less disk space for storage. Although for most image processing applications some loss of information is acceptable, in several areas such as astronomical, medical and high dynamic range imaging, lossless compression techniques are more convenient. For example, the lossy compression of medical images can involve distortions which may yield to wrong diagnosis. Hence, lossless compression techniques are required. Unlike lossy compression, lossless compression techniques allow to reconstruct perfectly the original data from the compressed data. With lossless compression, no loss of information is involved.

JPEG-LS [1] is among the recent and well known lossless image standards and algorithms for continuous-tone images. The main characteristic of JPEG-LS

is its ability to provide an efficient lossless compression at a low complexity. It is based on the LOCO-I (LOW COMplexity LOSSless COMpression for Images) algorithm which includes also a near-lossless compression [2].

In this paper, we propose a new preprocessing technique to improve the lossless compression performance of JPEG-LS, particularly for images that do not fall in the class of continuous-tone images.

2 Motivation

Several techniques were developed to supply additional content options. Effectively, besides natural content, images may include also text, graphics and computer generated items. The number of images which do not belong to the continuous-tone natural category has been continuously increasing. Generally, histogram sparseness occurs when the actual number of pixels' intensities is limited by comparison with the one implied by the nominal alphabet. The set of possible intensities is not therefore all used and active levels are distributed all over the entire nominal intensity range. Images of high bit depths tend also to have sparse histograms, e.g. medical and HDR images due to the large range of available intensity values.

JPEG-LS was designed to compress continuous-tone natural images. For this kind of images, pixel values do not vary randomly except near edges and highly textured areas. Hence, there is a strong correlation among neighboring pixels, which can be used to predict a pixel value from its neighbors. Thus, JPEG-LS processes by de-correlating the pixels. Direct observation of the histogram obtained with JPEG-LS for a continuous-tone natural image 'lena' (see Fig. 1) illustrates the distribution of the prediction error. The narrow concentration close to zero confirms the hypothesis of local uniformity of natural images. Unfortunately, this assumption is not true for sparse histogram images. Generally, this kind of images cannot be well handled by image compression algorithms since the inter-pixel correlation is weaker. The impact of histogram sparseness on JPEG-LS can be shown through the Fig. 1 where the histogram of prediction error of the image "Books" is presented. Prediction residual values spread over the entire range. As a consequence, it is not easy to predict the exact pixel's value or even a good approximation. Therefore, the entropy is increased and the compression efficiency is severely affected [3].

Several approaches, aiming at improving the performance of the compression of sparse histogram images, have been proposed. This confirms the importance of the issue. Some of those approaches have been designed for a particular encoder, such as "e extended prediction mode". It has been integrated in the 2nd part of the JPEG-LS standard to deal with images having sparse histograms [4]. Other approaches which, rather than modifying the basics of lossless encoders, proceed as preprocessing techniques detached from the particular image encoder [5, 6].

Among these preprocessing techniques, histogram packing cuts down the approximation error by decreasing the image variation [7]. The simplest method of histogram packing is known as offline histogram packing [8–10]. It simply

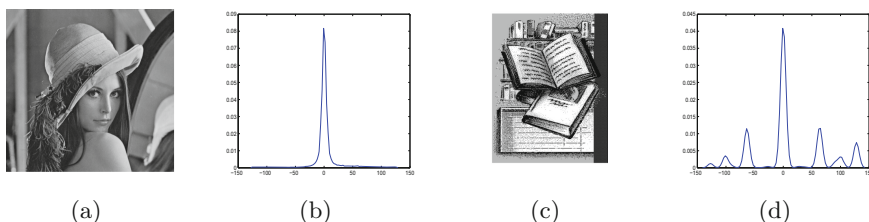


Fig. 1. Impact of histogram sparseness on JPEG-LS: (a) An example of a continuous-tone natural image ‘Lena’, (b) Prediction error histogram of ‘Lena’, (c) An example of an image with sparse histogram ‘Books’, (d) Prediction error histogram of ‘Books’.

corresponds to an order-preserving one-to-one mapping which makes the sparse histogram condensed to the lowest part of the nominal intensity interval with maintaining the general order.

Block-based histogram packing [11] supplies also an interesting alternative to the global histogram packing by dint of its ability to exploit the local characteristics. To explore these characteristics, the procedure consists basically on applying the offline histogram packing technique on consecutive image blocks of a predefined size. By comparison with the global offline histogram packing, the major difference, instead of one, block-based histogram packing generates many mapping tables (one per image block). Hence, the increase in the number of blocks produces a compression efficiency loss. The total number of overhead bits, depending on the number of image blocks and the size of the mapping table, becomes increasingly significant. Notwithstanding, as in [12–14], it has been shown that block-by-block encoding leads to improvements especially for images with locally sparse histograms.

Recently, Masmoudi et al. [15] proposed an efficient block-wise lossless image compression algorithm for images with sparse histograms. This method consists in encoding separately each image block with a suitable alphabet reduction scheme which takes advantage of the strength likelihood between active symbol sets of current image blocks and their neighbors. This alphabet reduction scheme provides, for each image block, the smallest set containing all the existing symbols with the minimum overhead bits.

In [16], Iwahashi et al. introduced a new preprocessing technique based on a simple quantization scheme. Unlike the classical uniform quantization, a non zero-bin condition is added. This scheme includes the histogram packing as a spacial case. Iwahashi et al. showed that the combination of the zero-skip quantization with any lossless encoder can outperform the existing methods for sparse histogram packing in both lossless and near lossless coding.

In this paper, we extend the technique of [15] to introduce a new compression technique (ABBHP) for images with sparse and locally sparse histograms. We propose to encode image block by block by using offline histogram packing with the alphabet reduction scheme in order to produce an adaptive block based histogram packing.

3 Proposed Approach

In this section, we present a full description of the algorithm of the proposed scheme (ABBHP).

3.1 Alphabet Reduction Scheme

The alphabet reduction scheme (ARS) takes advantage of the strength likelihood between neighboring blocks. The aim of this scheme is to attribute for each block, a set of symbols as close as possible to its corresponding active symbol set, with the minimum overhead bits as possible. In fact, the similarity between sets of active symbols can be measured with the Hamming distance. The Hamming Distance between two sets I and K, denoted by $D(I,K)$, is the sum of the newly appearing symbols number $D^+ = |I \setminus K|$ and the disappearing symbols number $D^- = |K \setminus I|$.

We consider \mathcal{I} as the original image to compress which is composed of N non-overlapping blocks, $\mathcal{I} = I^1, I^2, \dots, I^N$. Each pixel takes a value from the nominal alphabet of the image \mathcal{A} , where $\mathcal{A} = \{\alpha_1, \dots, \alpha_n\}$, e.g., for an 8-bit gray-scale image $\mathcal{A} = \{0, \dots, 255\}$. We denote by P the probability distribution of the symbols of the alphabet \mathcal{A} .

Step 1: Calculate the set of active symbols of the whole image X^0 which will be encoded by its binary representation B with $|\mathcal{A}|$ -bits length calculated as follows:

$$X^0 = \{\alpha_i \in \mathcal{A} / P(\alpha_i) \neq 0\}, \quad (1)$$

$$B(i) = \begin{cases} 1, & \text{if } \alpha_i \in X^0 \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

Step 2: Read the image block by block. For each block I^k , compute its active symbol set X^k , minimum Min^k , maximum Max^k , and the set of its neighbor block active symbol sets \mathcal{N}^k :

$$X^k = \left\{ \alpha_i \in \mathcal{A} / \exists t, I^k(t) = \alpha_i \right\}. \quad (3)$$

We consider only three neighboring blocks, then \mathcal{N}^1 of the first image block, which is located at the top left corner of the input image, is equal to $X_{Min-Max}^1$.

Otherwise, the \mathcal{N}^k of blocks belonging to the first block row is $\left\{ X_0^k, X_{Min-Max}^k \right\}$.

However, \mathcal{N}^k of blocks belonging to the first block column is $\left\{ X_1^k, X_{Min-Max}^k \right\}$.

For the remaining blocks we have:

$$\mathcal{N}^k = \left\{ X_0^k \cup X_1^k \cup X_2^k \cup X_{Min-Max}^k \right\}, \quad (4)$$

where X_j^k corresponds to the set of active symbols in I_j^k the j 'th neighbor block of I^k . $X_{Min-Max}^k$ is the symbol set containing all the elements of X^0 having a value between the minimum and the maximum symbols present in X^k :

$$X_{Min-Max}^k = \{Min^k, \dots, Max^k\}. \tag{5}$$

Step 3: Seek for the closest neighbor block set \hat{X}^k for which $D(X^k, X_j^k)$ is minimum where $X_j^k \in \mathcal{N}^k$. Let's denote by \hat{j} its position in \mathcal{N}^k , i.e., $\hat{j} \in \{0, 1, 2, 3\}$. This method needs to send \hat{j} value, of each block, as an additional information to the decoder, \hat{j} is encoded with $\lceil \log_2(3 + 1) \rceil$ bits.

Step 4: Instead of encoding all the active symbols of each block, the encoder selects only the newly appearing symbols, i.e., the pixels' intensities that appear in the current block and disappear in the neighboring blocks. Hence, each block can be encoded with the smallest possible set representing its active symbol set based on its neighboring blocks. If \hat{j} corresponds to the position of one of the neighboring blocks, $\hat{j} \in \{0, 1, 2\}$, compute the set of the newly appearing symbols NAS and its length D^+ where:

$$NAS = X^k \setminus \hat{X}^k. \tag{6}$$

If there are no newly appearing symbols, $D^+ = 0$, then encode the bit "0", otherwise, encode each newly appearing symbol with $\lceil \log_2(|X^0| - |\hat{X}^k|) \rceil$ bits using its index in $X^0 \setminus \hat{X}^k$.

If not, i.e., $\hat{j}=3$, \hat{X}^k corresponds to the last symbol set in N^k , encode Min^k and Max^k values as additional information. We encode each value with $\lceil \log_2(|X^0|) \rceil$ bits.

Step 5: Compute the estimated symbol set used when encoding the k^{th} block given by:

$$\tilde{R}^k = X^k \cup \hat{X}^k. \tag{7}$$

Let us assume that \tilde{R}^k is an ascending sorted set of r different intensity values, $\tilde{R}^k = \alpha_0, \dots, \alpha_{r-1}$, where $\alpha_i \in \mathcal{A}$ and $r \leq |\mathcal{A}|$.

3.2 Offline Histogram Packing

Finally, we made use of the offline histogram packing with the aim to reduce the effect of histogram sparseness on image compression standards. The offline histogram packing is then used to map \tilde{R}^k . It maps ascending sorted intensity values into ascending sorted contiguous indexes.

$$h = (\alpha_0 \mapsto 0, \alpha_1 \mapsto 1, \dots, \alpha_{r-1} \mapsto r - 1)$$

Note that in this work, we have not presented an implementation detail for JPEG-LS as our proposed method does not imply any modification of its code.

A further advantage of JPEG-LS is the so-called “sample mapping procedure” which allows the decoder to map each decoded sample value into a reconstructed one [4]. This feature, although expected for handling palette images, can also be used to store additional information for our proposed process.

Furthermore, the restriction of the number of symbols for each block will improve the coding performance of our proposed algorithm in terms of speed processing and complexity as the number of computations is linear proportional with the size of the set of symbols. In addition, the modifications brought to the initial image made by the offline histogram packing reduce the unwanted effects of histogram sparseness on encoding algorithms.

4 Experimental Results

To measure the performance of our method two sets of test images are used. The first set of images contains Eleven gray-scale-converted version of a set used by Ausbeck in his PWC coder [9]. The second set regroups five images available from the University of Waterloo Greyset2 collection [17]. This set was used to illustrate the performance of JPEG2000 and JPEG-LS for this type of images [18]. The compression performance of the proposed method is also compared to that of the scheme (ARS), published in [15], when combined with an adaptive arithmetic coding (AAC).

Figure 2 illustrates the prediction error distribution obtained by the MED predictor for a sparse histogram image “Frog”. As depicted in Fig. 2b, a loss of efficiency occurs when this predictor is applied directly to this image. A significant improvement is observed when the proposed method is used. The generated error histogram shows a sharply distribution with one peak centered at zero and two-sided exponential decay, which is similar to the standard prediction error histogram for natural images. Moreover, the proposed method, when followed by the MED predictor, produces small prediction errors which confirms the superior performance of the proposed method over the offline histogram packing technique.

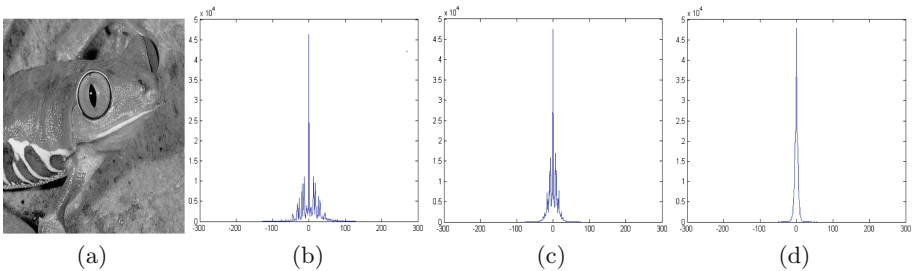


Fig. 2. Improvement of the prediction made by MED predictor with sparse histogram images: (a) Image Frog, (b) Original prediction error histogram, (c) Prediction error histogram when OFF HP is applied, (d) Prediction error histogram when the proposed method is applied.

Table 1. Comparison between bit rates obtained by lossless JPEG-LS with (OFFHP columns) and without offline histogram packing (Normal columns) and the proposed approach (ABBHP columns) for the first and the second image sets.

Name	Image			ARS+AAC	JPEG-LS		
	Size (Byte)	Intensities	Entropie		Normal	OFFHP	ABBHP
Benjerry	27,960	41	2.39	1.49	1.92	1.40	1.36
Books	56,922	7	2.19	1.44	5.61	1.97	1.60
Cmpndd	393,216	142	2.48	1.20	1.45	1.52	1.25
Cmpndn	393,216	137	2.41	1.04	1.21	1.29	1.02
Gate	60,912	72	4.29	2.43	3.60	2.71	2.48
Music	12,321	8	2.11	1.12	2.88	1.59	1.42
Netscape	61,200	30	3.48	2.08	2.82	1.76	1.82
Sea_dusk	157,300	42	4.50	0.74	0.20	0.19	0.24
Sunset	307,200	133	6.44	2.47	2.13	1.91	1.86
Winaw	294,345	10	2.09	0.58	1.30	0.59	0.55
Yahoo	27,140	138	3.01	1.66	2.60	2.43	1.70
Average	162,884	69.09	3.21	1.47	2.34	1.58	1.39
France	333,312	249	6.28	2.21	1.41	1.40	0.78
Frog	309,258	102	4.97	3.30	6.04	5.17	3.94
Library	163,328	221	5.85	4.73	5.10	5.00	4.95
Mountain	307,200	110	6.22	5.20	6.42	5.24	5.16
Washsat	262,144	35	2.87	2.53	4.12	2.00	2.08
Average	275.048	143.40	5.24	3.60	4.61	3.76	3.38

Note: OFFHP, Offline Histogram Packing; ABBHP, Adaptive Block Based Histogram Packing; ARS, alphabet reduction scheme; AAC, adaptive arithmetic coding.

Table 1 presents the compression results obtained with the lossless JPEG-LS for blocks of 16×16 . The coding rate shown in boldface represents the best result for each image set. This table confirms that the proposed method outperforms JPEG-LS with and without offline histogram packing for almost the test images. Offline histogram packing is lightly favorable for only three images which is due to the additional bits required for the decoding process. When compared to the method of [15], we can record an interesting bit rate reduction for the two test image sets.

In Fig. 3 we compare the bitrate values between original images and compressed ones using JPEG-LS without and with preprocessing techniques. Results confirm the loss of performance of JPEG-LS when handling sparse histogram images. As we can see, a significant loss of bitrate between original and compressed images occurs when using JPEG-LS directly for several test images which are; Music, Books, Frog, Mountain and Washsat. Entropy increase is due to the loss of efficiency during the prediction stage of JPEG-LS as we have shown in Sect. 2.

Moreover, the proposed method was designed to be efficient and suitable for both hardware and software implementations. It is important to note that the

overall complexity of the proposed algorithm is linear in the size of the image. The simplicity of the algorithm and the low complexity are shown with the interesting improvement of the compression performance.

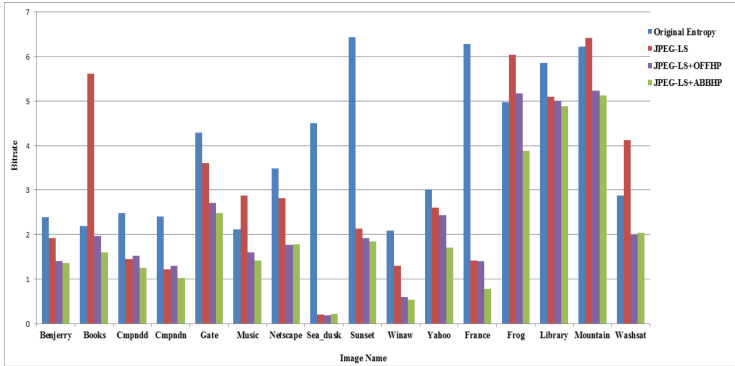


Fig. 3. Bitrate values obtained for each image comparison of bitrate between original and compressed images when applying JPEG-LS without and with: offline histogram packing OFFHP and the proposed method ABBHP.

5 Conclusion

In this work, we have proposed an efficient preprocessing technique to improve the lossless compression efficiency for images with sparse histograms using the ARS with the offline histogram packing. This method reduces the size of the additional information by encoding the newly appearing symbols for each block using its neighboring blocks. In addition, it maps the sparse samples into a contiguous set. The experimental results confirm that this novel method outperforms JPEG-LS by 40,6% and by 12% when using the offline histogram packing as a preprocessing. Moreover, it does not require any modification of the code of the compression technique which means it can be used with any image encoder.

References

1. Information technology-lossless and near-lossless coding of continuous tone still images. ISO/IEC International Standard 14495-1, ITU Recommend. T.87 (1999)
2. Weinberger, M.J., Seroussi, G., Sapiro, G.: LOCO-I: a low complexity, context-based, lossless image compression algorithm. In: Proceedings of IEEE Data Compression Conference, pp. 140-149. IEEE (1996)
3. Pinho, A.J.: On the impact of histogram sparseness on some lossless image compression techniques. In: Proceedings of IEEE International Conference on Image Processing (ICIP), vol. III, pp. 442-445. IEEE (2001)

4. Weinberger, M.J., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *IEEE Trans. Image Process.* **9**(8), 1309–1324 (2000)
5. Starosolski, R.: Compressing images of sparse histograms. In: *SPIE Proceedings*. vol. 5959, pp. 209–217. Citeseer (2005)
6. Ferreira, P.J.S.G., Pinho, A.J.: Histogram packing, total variation, and lossless image compression. In: *The 11th European Signal Processing Conference*, pp. 1–4. IEEE (2002)
7. Ferreira, P.J.S.G., Pinho, A.J.: Why does histogram packing improve lossless compression rates? *IEEE Signal Process. Lett.* **9**(8), 259–261 (2002)
8. Pinho, A.J.: An online preprocessing technique for improving the lossless compression of images with sparse histograms. *IEEE Signal Process. Lett.* **9**(1), 5–7 (2002)
9. Ausbeck, P.J.: The piecewise-constant image model. *Proc. IEEE* **88**(11), 1779–1789 (2000)
10. Pinho, A.J.: A comparison of methods for improving the lossless compression of images with sparse histograms. In: *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. II, pp. 673–676. IEEE (2002)
11. Pinho, A.J., Neves, A.J.R.: Block-based histogram packing of color-quantized images. In: *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, vol. I, pp. 341–344. IEEE (2003)
12. Masmoudi, A., Masmoudi, A.: A new arithmetic coding model for a block-based lossless image compression based on exploiting inter-block correlation. *Signal Image Video Process.* **9**(5), 1021–1027 (2015)
13. Masmoudi, A., Puech, W., Masmoudi, A.: An improved lossless image compression based arithmetic coding using mixture of non-parametric distributions. *Multimedia Tools Appl.* **74**(23), 10605–10619 (2015)
14. Pinho, A.J.: Preprocessing techniques for improving the lossless compression of images with quasi-sparse and locally sparse histograms. In: *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*. vol. I, pp. 633–636. IEEE (2002)
15. Masmoudi, A., Zouari, S., Ghribi, A.: Adaptive block-wise alphabet reduction scheme for lossless compression of images with sparse and locally sparse histograms. *J. Electron. Imaging* **24**, 063001 (2015)
16. Minewaki, S., Iwahashi, M., Kobayashi, H., Yoshida, T., Kiya, H.: Near lossless coding of sparse histogram images based on zero-skip quantization. *Multimedia Tools Appl.* 1–19 (2017)
17. Kominek, J.: Waterloo bragzone and fractals repository, January 2007. <http://links.uwaterloo.ca/Repository.html>
18. Carpentieri, B., Weinberger, M.J., Seroussi, G.: Lossless compression of continuous-tone images. *Proc. IEEE* **88**, 1797–1809 (2000)