On Predictive-Based Lossless Compression of Images with Higher Bit Depths

Aleksej Avramović, *Member, IEEE,* and Goran Banjac, *Member, IEEE*

Abstract — **Due to the rapidly increasing requirements for data transmission and storage, applications for fast and efficient compression of data have a very important role. Lossless compression must be applied when data acquisition is expensive. For example, lossless image compression must be applied in aerial, medical and space imaging. Besides the requirements for high compression ratios as much as it is possible, lossless image coding algorithms should be as fast as possible. During the late nineties of the previous century, many predictive-based algorithms for lossless compression of 8-bit images were introduced. These algorithms were usually expanded to enable processing of images with higher bit depths. All predictive based algorithms used more or less efficient predictors to remove spatial redundancy in images. This paper gives a comparative analysis of predictor efficiency with special emphasis on images with higher bit depths. A novel predictive-based, lossless image compression algorithm with a simple context-based entropy coder is presented, as well. A comparison with standardized lossless compression algorithms JPEG-LS and JPEG2000 is made on a large set of 12-bit medical images of different modalities and 12-bit and 16-bit natural images. It is shown that the proposed solution can achieve approximately the same bitrates as standardized algorithms even though it is much simpler.**

Keywords — **Bit depth, compression, lossless, medical images, prediction.**

I. Introduction

ALTHOUGH rapid technology development enables increasing of data storage and transmission capabilities, a need for more efficient compression algorithms is ever-present, because technology development is followed by increased data generation [1]-[4]. Images, as well as text or videos, represent a piece of data that caries information. Redundancy occurs when we have more data than it is necessary to carry a certain amount of information. Compression is a type of data transform which reduces an amount of data, preserving more or less information carried by the data. Standard image compression algorithms exploit the fact that the human visual system is unable to detect some spectral components, but still, lossless image compression techniques are required in many practical cases [2]. For example, space and aerial images are expensive to acquire and may contain valuable data undetectable by human eye. Digital medical images are usually written with higher bit depths than natural images. In some countries there are legal constraints about lossy compression of medical images, because they may lose their diagnostic value.

This paper is the result of research under the project Lossless image compression, which is supported by the Ministry of Science and Technology of the Republic of Srpska, Contract No 06/0-020/961-166/09. Also, authors would like to thank KBC Paprikovac, Banja Luka, which supplied medical images for testing that are used in this paper.

Corresponding author Aleksej Avramović is with the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina; (e-mail: aleksej@etfbl.net).

Goran Banjac is with the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina; (e-mail: goran.banjac@etfbl.net).

Image compression algorithms can be divided into two main categories according to a method that is used to remove spatial redundancy [1]. Prediction-based techniques use a simple assumption that a pixel value can be partially or totally represented as a linear combination of neighbor pixels. Transformation-based techniques transform data in such a way that allows better exploitation of spectral components that are present on an image. For lossless image compression, prediction is chosen before transformation because it is simple, fast and, most important, it can guarantee a lossless data recovery. Intensive research on lossless image compression resulted in proposing several prediction-based algorithms, such as LOCO-I [5], [6], CALIC [7], [8], etc. in the late nineties of the previous century. These algorithms use:

1. a simple and efficient predictor that removes most of the existing redundancy,
2. context-modeling, that further exploits a repeating pattern to improve prediction, and
3. entropy coding to finally remove statistical redundancy.

LOCO-I was further improved and standardized as the JPEG-LS lossless compression algorithm [2], [6]. At the beginning of the 21st century, a very flexible and efficient wavelet-based image compression technique was presented. Called JPEG2000, this algorithm joined both lossy and lossless compression, at the same time answering additional demands, such as multispectral compression, region-of-interest based compression, volumetric images compression, etc [2]. Still, in the case of lossless image compression, JPEG-LS algorithm has shown to be more efficient, besides the fact that it is much simpler and computationally less expensive [2].

In this paper, a comparative analysis of linear predictors efficiency is given. Efficiency of a predictor can be measured with entropy of output prediction-error image. A comparison of simple linear predictors, with predictors based on the least square error and least absolute deviation is given for higher bit depth images. A simple and efficient coding context is presented as well. It is based on a small number of causal pixels and it is bit-depth independent.

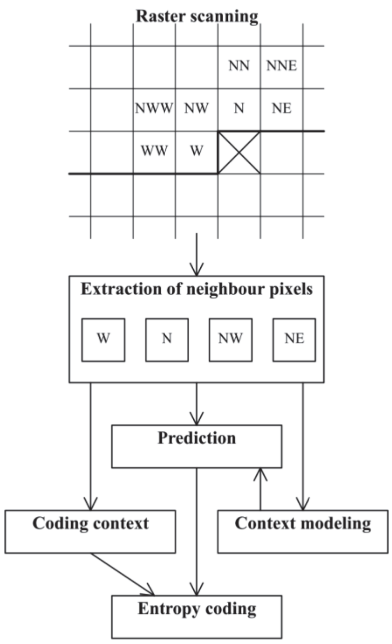

Fig. 1. General scheme for lossless image compression and common scheme for labeling causal pixels.

Therefore, the proposed coding context is suitable for compression of images with various bit depths.

This paper is organized as follows. In Section II the basic phases of predictive-based lossless compression technique are presented. Section III gives an overview of prediction with a particular overview of predictor's efficiency on higher bit depth images. Section IV is about context modeling and Section V discusses the proposed coding context. Section VI gives a discussion on experimental results of proposed algorithms and Section VII is a conclusion.

II. Predictive-Based Lossless Image Compression

Lossless compression algorithms were usually designed and optimized to process natural 8-bit images, with later extension to higher bit depth images [6]-[14]. Many types of high bit depth images are expensive to acquire and have a significant practical value. Therefore, it is important to use an efficient algorithm for lossless high-bit-depth image compression. For example, design of lossless compression of medical images is important considering the rapid development of telemedicine and demands for preservation of diagnostic value. Predictive-based lossless image compression contains three main steps of data processing, namely prediction, context modeling and entropy coding, as it is depicted in Fig. 1. Entropy coding may use neighbor pixels to determine a coding context which can improve a compression ratio. Each of these steps should remove a part of redundancy as much as it is possible. Predictive algorithms for image compression can be classified into two groups: algorithms with a single pass and algorithms with two passes. Algorithms with one pass count all required parameters for compression during one scanning of image, such as the optimal predictor parameters. Coding of prediction error can be done in combination with prediction or in a separate phase after the prediction. Algorithms with two passes have the ability to analyze image and efficiently determine the optimal parameters for the prediction and coding in the first pass, while in the second pass consist of prediction and coding. Optimization of parameters is usually based on the principle of least square error, which generally gives better results than switching predictors. Switching predictors choose a subpredictor from the set of predictors based on the context in which the pixel is located. Redundancy is an additional amount of data that carries no additional information [1], [2]. Without loss of generality, further discussion is devoted to grayscale images. A grayscale image is a matrix with each pixel given by its level of grayness. Spatial redundancy represents a fact that the gray value of one pixel may be partially calculated by values of other pixels, so a part of its information is carried by other pixels. Color images are represented by a spectral band individually and may be processed by each band separately. Additional redundancy occurs between each spectral band. A digital video is a successive array of digital images, so redundancy between frames is considered as a temporal redundancy. Psychovisual redundancy refers to information that is carried in data, but that cannot be detected by human visual system. Still, if we have acquired expensive data, that kind of information could be revealed by some techniques of digital image processing, so sometimes it must be preserved. Removing psychovisual redundancy introduces distortion in uncompressed images, so this step is skipped during lossless image compression. Statistical or coding redundancy represents how much data we have more that is actually necessary to carry original amount of information. Entropy coding removes statistical redundancy, and the most commonly used codes are Huffman code, Golomb-Rice code, or arithmetic coding [4]. Entropy coding actually performs the compression and gives the final output sequence of compressed data. Coding can be done after the prediction as an independent phase or progressively during the prediction.

III. Removing Spatial Redundancy

Prediction is the crucial part of compression, because it removes most of the spatial redundancy, and the choice of the optimal predictor is essential for the efficiency of compression methods. The prediction may be linear or nonlinear. Linear prediction is based on one or more fixed predictors, which can be combined with appropriate weights or it is possible to choose the optimal subpredictor on the basis of the properties of a certain part of image, i.e. on the basis of context of the region. The context can be determined for each pixel separately, for example, during raster scanning, a pixel can be found in the region of the horizontal edge, vertical edge, sloping edges, texture, smooth region, etc. Linear predictors based on a finite group of subpredictors are simple and fast. Several predictors, based on the least squares optimization, have been proposed, which in comparison with fixed predictors can be more efficient, but they are much more computationally demanding. Optimization is based on a finite group of casual pixels, so called context. Since the optimization for each pixel can be computationally demanding, the adaptation of coefficients is usually done when another type of region occurs, for example an edge. For effective adaptation a larger context, comparing with fixed predictors, is required. Nonlinear prediction is based on neural networks, vector quantization, etc.

A. Lossless JPEG Predictors

The first version of lossless JPEG, based on prediction, introduced seven simple predictors for removing spatial redundancy. These predictors are *W*, *N*, *NW*, *W+N-NW*, *W+(N-NW)/2*, *N+(W-NW)/2* and *(N+W)/2*, where labels are in agreement with Fig. 1. After prediction, an error-image is coded using an entropy coder.

B. Median Edge Detection Predictor

A Median Edge Detection (MED) predictor belongs to the group of switching predictors that select one of the three optimal subpredictors depending on whether it found a vertical edge, horizontal edge, or a smooth region [5]. In fact, MED predictor selects the median value between three possibilities *W*, *N* and *W+N-NW* (pixels are commonly labeled after sides of the world, Fig. 1). MED has proven to be an optimal combination of simplicity and efficiency.

C. Gradient Adjusted Predictor

A Gradient Adjusted Predictor (GAP) is based on gradient estimation around a current pixel. Gradient estimation is done by the context of a current pixel, which in combination with predefined thresholds gives a final prediction. GAP distinguishes three types of edges, strong, simple and a soft edge, and is characterized by high flexibility to different regions. The first step is to estimate a local horizontal and vertical gradient:

$$\begin{aligned} g_v &= \left|W - WW\right| + \left|N - NW\right| + \left|N - NE\right|, \\ g_h &= \left|W - NW\right| + \left|N - NN\right| + \left|NE - NNE\right| \end{aligned} \quad (1)$$

Then, prediction *P* is done according to these estimations:

$$\begin{gathered} \mathit{if}\ g_v - g_h > 80, P = W;\ \mathit{elseif}\ g_v - g_h < -80, P = N; \\ \mathit{else}\ P = \left(W + N\right)/2 + \left(NE - NW\right)/4; \\ \mathit{if}\ g_v - g_h > 32, P = \left(P + W\right)/2; \\ \mathit{elseif}\ g_v - g_h > 8, P = \left(3P + W\right)/4; \\ \mathit{elseif}\ g_v - g_h < -32, P = \left(P + N\right)/2; \\ \mathit{elseif}\ g_v - g_h < -8, P = \left(3P + N\right)/4. \end{gathered} \quad (2)$$

D. History Based Blending Predictor

Authors in [10] have proposed adaptive prediction based on a combination of thirteen simple predictors and the appropriate penalty of predictors which result in large prediction errors. In the same paper, authors concluded that similar results can be obtained using only six simple predictors. The six predictors defined in a P6 set are *W*, *N*, *W+N-NW*, *NE*, *(NW+W)/2*, *NW*. The authors concluded that a reduced and much simpler version with six predictors gives negligibly worse results than the version with thirteen subpredictors. Therefore, in this paper only History Based Blending (HBB) algorithm based on six predictors will be discussed.

E. SFALIC Predictor

In [15] authors introduced a simple and fast lossless compression algorithm. The main goal of this algorithm was not a high compression ratio, but a high processing speed. One of the proposed subpredictors is an average combination of three lossless JPEG predictors:

$$P = \left(3A + 3B - 2C\right)/4. \quad (3)$$

F. DARC Predictor

DARC is a simple and adaptive predictor which adjusts the prediction based on a simple estimation of horizontal and vertical gradient: $g_v = |W\text{-}NW|$, $g_h = |N\text{-}NW|$ and the corresponding weighted coefficient $\alpha = g_v/(g_v+g_h)$. After local gradient estimation, a predictor is calculated as:

$$P = \alpha W + (1-\alpha)N\,. \quad (4)$$

G. Gradient Edge Detection Predictor

Gradient Edge Detection (GED) predictor is designed to be a trade-off between the simplicity of MED and efficiency of GAP [16]. GED uses local gradient estimation, similarly as GAP, to choose between three subpredictors, defined as in MED predictor. The number of causal pixels is a compromise between MED and GAP, as well. Estimation of local gradient is made with:

$$\begin{aligned} g_v &= \left|NW - W\right| + \left|NN - N\right| \\ g_h &= \left|WW - W\right| + \left|NW - N\right| \end{aligned} \quad (5)$$

after which prediction is done according to equation:

$$\begin{gathered} \mathit{if}\ g_v - g_h > T, P = W \\ \mathit{elseif}\ g_v - g_h < -T, P = N \\ \mathit{else}\ P = N + W - NW \end{gathered} \quad (6)$$

where *T* is a threshold. In this research, T is 2^{bit}-*[(bit-2)/2]*.

H. Minimum Mean Square Error Predictor

Predictors based on Minimum Mean Square Error (MMSE) perform the adaptation of prediction coefficients on the basis of a training set of causal pixels [9], [11], [14]. This approach can achieve better results reducing redundancy compared to algorithms that are using a fixed number of subpredictors. However, during adaptation, it is necessary to calculate the pseudo inversion of training matrix, whose order increases with an increasing training set. If we use the last *m* pixels for the adaptation of predictor coefficients of *k*-th order predictor, the vector of adapted coefficients **a** is calculated according to the expression: $\mathbf{a} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{y}$ where the order of the training matrix **C** is *m*x*k*. The vector of the last *m* scanned pixels is **y**. A MMSE predictor is an efficient method for exploiting patterns and textures in causal neighborhood. The main drawback of this method is the impact of prediction error square during coefficients adaptation. In other words, if there is one outsider in a training set it can lead to nonoptimal coefficients. MMSE is not robust to outsiders.

I. Least Absolute Deviation Predictor

A Least Absolute Deviation (LAD) predictor adapts coefficients by minimizing the sum of absolute errors taken from a causal set. The advantage of this method is stronger robustness to outsiders. The main drawback is the fact that LAD predictor cannot be adapted in a similar manner as MMSE, because an absolute value is not a differentiable function. One possible approach is the

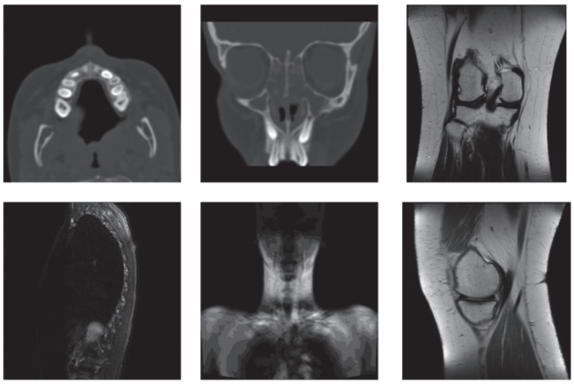

Fig. 2. Examples of medical images.

Fig. 3. Examples of natural images.

weighted iterative adaptation of coefficients [17]. This approach uses several least square adaptations and it is proven to converge linearly towards an optimal LAD solution.

J. Entropies of Prediction-Error Images

Entropy is a theoretical value that describes a minimum bit rate subject to the assumption that images are zero order Markov processes. If we denote an image as a random variable X, with an alphabet $A=(a_0, a_1, a_2, .. a_{N-1})$, which means we have an N-bit image, entropy can be calculated as follows:

$$H(X) = -\sum_{x\in A} p(x)\log_2 p(x) \quad (7)$$

where $p(x)$ is the associated probability of a symbol x. According to Shannon's source coding theorem, the optimal code length for an input symbol x is $-log_2p(x)$, so the equation gives an optimal coding bitrate. Entropy calculated after removing spatial redundancy can be used for the estimation of a final compression ratio.

Usually, a prediction-error is further processed in order to increase the performance of entropy coder or to reduce alphabet size. In this research, two additional methods are used, as it was described in [15]. Firstly, prediction errors are calculated with modular arithmetic, thus avoiding the expansion of alphabet. Secondly, prediction errors are rearranged to give more Laplacian distribution:

$$\begin{aligned} &if\ E = 2^{bit-1}, X = 2E \\ &else\ X = 2\left(2^{bit} - E\right) - 1. \end{aligned} \quad (8)$$

where X is a remapped value, E is an input prediction-error and *bit* is a bit-depth. Since we are particularly interested in the performance of described predictors on images with higher bit-depths, experiments are made on a set of medical and natural 12-bit and 16-bit images. In order to have a fair comparison, all entropies are calculated after modular prediction and mapping (8). Table 1 gives averaged entropy values for described predictors. A medical evaluation set contains 1338 12-bit CT images with size *512x512* and 528 12-bit MRI images with various sizes (Fig. 2). A natural evaluation set contains 147 12-bit and 150 16-bit images with sizes *680x1024* (Fig. 3). All used images are previously uncompressed. MMSE(m,n) means that an m-th order predictor and an n-th order training set are used. LAD(m,n,k) means that an m-th order predictor and an n-th order training set with k re-weight iterations are used.

TABLE 1: EVALUATION OF PERFORMANCE OF DIFFERENT PREDICTORS ON SEVERAL SETS OF IMAGES WITH HIGHER BIT-DEPTH. NUMBERS IN THE TABLE ARE AVERAGED ENTROPIES.

Predictor	*CT*	*MRI*	*N12*	*N16*
LosslessJPEG1	5.17	6.17	9.23	11.01
LosslessJPEG2	4.96	5.97	9.40	11.35
LosslessJPEG3	5.48	6.48	9.81	12.16
LosslessJPEG4	3.95	5.75	9.10	10.80
LosslessJPEG5	4.51	5.79	9.07	12.15
LosslessJPEG6	4.37	5.69	9.14	12.26
LosslessJPEG7	4.75	5.78	9.16	12.18
MED	4.19	5.58	8.85	10.58
GAP	4.39	5.57	9.06	10.96
HBB	3.94	5.66	9.02	12.59
SFALIC	4.27	5.62	8.96	12.46
DARC	4.58	5.76	8.98	10.71
GED	4.30	5.75	9.05	10.82
MMSE(2,12)	3.96	5.54	9.09	12.66
MMSE(4,20)	3.58	5.42	8.99	12.56
MMSE(6,36)	3.34	5.32	8.92	12.48
LAD(4,12,1)	3.53	5.53	9.10	12.66
LAD(4,20,2)	3.54	5.42	8.99	12.55

Examples of images can be found on www.etfbl.net/~aleksej/compress.htm. Analyzing the results from Table 1, it can be concluded that simple predictors have equal or better performances on natural images, compared to least square based predictors with heavy computing demands. A reason for this result can be found in the fact that natural images have more complex patterns that are hard to find. The second reason might be the fact that causal contexts often have more outsiders and it is well-known that least square methods are not robust to them. Considering medical images, least square methods show an improvement when expanding a test set, but at the expense of significant computing demands. Based on these conclusions, a simple predictor *LosslessJPEG4* is chosen for the proposed solution.

IV. CONTEXT MODELING

Context modeling uses the repeating patterns of neighbor pixels of a current pixel to further improve prediction. It is often a case that images contain textures which are characterized by repeating values of neighboring pixels. Each time some scheme is detected, a context model is updated and learned about the probability distribution of pixel values for that scheme. For example, if we observe n previous pixels in a binary image, we can notice that there are 2^n possible schemes. To overcome this bottleneck, similar schemes are grouped into one context and the determination of context will be crucial for prediction correction. A context is detected by applying a detection rule on several neighboring pixels. For each context, an accumulator and counter are constantly updated. A context counter N is incremented each time a context is detected. Accumulator A is a sum of prediction errors for each case a context is detected. After prediction and context detection, a prediction value P is corrected:

$$P_c = P + \left[A / N \right] \tag{9}$$

A large number of contexts may lead to so called context dilution, a phenomenon when the number of contexts is high so they cannot be learned during the image scanning. A smaller number of contexts is also not optimum, because the conditional probability of following pixel values may not determine an optimum correction for prediction. Another problem are areas on image with different textures. Transition is a second texture area that may lead to non optimum correction calculated according to non learned contexts. These problems are usually solved by setting a limit of a counter for every context. After a limit is reached, a context counter and appropriate context accumulator are halved. Context modeling used in this research uses a similar context modeling algorithm as in [16]. A context is determined with three neighbor pixels, two previous prediction errors, local gradient estimation and three adjustable thresholds. Context and associated accumulator are labeled with a unique index. The index is a 7-bit number, thus a total number of contexts is 256. The first three bits are set to one if W, N and NW are larger or equal than prediction, respectively. Otherwise, they are set to zero. The second two bits are set to one if the previous two prediction errors are even, respectively. The last two bits are set according to local gradient estimation (5) and predefined thresholds:

$$\begin{gathered} g = g_v + g_h \\ if\ g \le T_1, then\ k\left(6\right) = 0\ and\ k\left(7\right) = 0, \\ elseif\ g \le T_2, then\ k\left(6\right) = 1\ and\ k\left(7\right) = 0, \\ elseif\ g \le T_3, then\ k\left(6\right) = 0\ and\ k\left(7\right) = 1, \\ else, k\left(6\right) = 1\ and\ k\left(7\right) = 1, \end{gathered} \tag{10}$$

In this research thresholds are set to default values $T_1=8$, $T_2=16$ and $T_3=32$.

V. Entropy Coding

Entropy coding is the step of lossless image compression algorithm which actually does the compression. Previous steps are removing redundancy, but the coder is forming the final bitstream of a compressed image. Statistical redundancy refers to possibility to store an image on less memory space when using variable-length code words, rather than fixed-length code words. Removing statistical redundancy is the process of exploiting the fact that some input values occur with higher possibilities than others. Those input values that occur more often are coded with minimum length code words, thus preserving storage space. When coding grayscale images, input values are the values of grayness. Entropy coders are referred to as a group of algorithms which can achieve bitrates arbitrarily close to (7). The most popular entropy coders are the Huffman coder, Golomb-Rice coder and arithmetic coder [1]-[4].

A coding context exploits the fact that the conditional probability of prediction error image $p(E|c)$, where c is a context condition, is lower than the original probability of prediction error image $p(E)$. Thus, optimal determining of coding context can further remove statistical redundancy achieving lower output bitrates. Although conditional probability enables efficient selection of a coding context, this method requires additional computations. But still, a coding context provides more efficient coding and a shorter output data stream. In order to reconcile the two contradictory requirements, a very simple coding context is proposed, which is based on the fact that each of the possible values usually have several optimal coding contexts c. These coding contexts can be estimated from the neighboring pixels, as well. Since only three neighbor pixels are used for prediction and context modeling, the proposed solution uses three neighbor pixels as well. These three pixels are used to estimate an optimal characteristic number k for the Golomb-Rice coder. Thus, a simple and efficient method is used to remove statistical redundancy and to create an output bitstream. For a 12-bit image, a coding context is based on the auxiliary parameter e, which is a simple estimation of a local level of activity:

$$e = \left\lfloor \left(N + W + NW \right) / 3 \right\rfloor. \tag{11}$$

This parameter actually estimates the magnitude of the next value to be coded. Considering used mapping and modular arithmetic, the magnitude can be in the range $[0,1,..\ 2^{12}-1]$, thus k is estimated to produce the shortest possible codeword for the estimated magnitude:

$$\begin{gathered} if\ e < 2, then\ k = 0; elseif\ e < 4, then\ k = 1; \\ elseif\ e < 8, then\ k = 2; elseif\ e < 16, then\ k = 3; \\ elseif\ e < 32, then\ k = 4; elseif\ e < 64, then\ k = 5; \\ elseif\ e < 128, then\ k = 6; elseif\ e < 256, then\ k = 7; \\ elseif\ e < 512, then\ k = 8; elseif\ e < 1024, then\ k = 9; \\ elseif\ e < 2048, then\ k = 10; else\ k = 11; \end{gathered} \tag{12}$$

This simple principle can be generalized to efficiently code images with an arbitrary bit-depth. Considering the fact that the context of a current pixel can be misleading, a non optimal parameter k can be selected causing a significant extension. In this case an array of *bit* ones is used as a mask after which the current pixel is coded binary.

To further improve coding efficiency, a dictionary expansion method based on the Run-Length mode is used. This mode is efficient in cases where long series of repeating values are found in an image, which is often the case in medical images. These areas in an image have entropy lower than 1 bpp, so entropy coders are inefficient. Instead of coding each pixel, it is only necessary to code a number of repetitions.

VI. Experimental Results

Previous sections contain detailed descriptions of major steps for lossless image compression algorithm. Proposed solutions of each step are given as well. In this section, the proposed solution is tested and its performance is compared with the standardized lossless image compression algorithm JPEG-LS. This standard is proven to have a better performance than JPEG2000 considering lossless coding [2], but JPEG2000 is considered as well.

One of the easiest ways to compare compression methods is to obtain compression ratios. But averaged compression ratios do not carry any significant information, therefore the proposed solution will be compared with the standardized JPEG-LS algorithm by the memory space required to store a test dataset after compression. Table 2 gives the results of the described test, where numbers in the table are the amount of megabytes (MB), required for storage.

Table 2: Comparison of the efficiency of proposed algorithm and JPEG-LS on various datasets.

Method	***CT***	***MRI***	***N12***	***N16***
Proposed	140.9	41.6	109.8	162.9
JPEG-LS	136.7	42.4	106.6	158.1
JPEG2000	130.6	41.1	110.4	162.4

Analyzing the results from Table 2, we can see that the proposed algorithm gives better results for medical images compared to JPEG-LS, but JPEG2000 is more suited for this type of data. Considering the performance on natural images, JPEG-LS gives best results, thus proving that it is able to model data effectively on higher resolutions. It is important to notice that the proposed solution gives similar results as much more complicated and computationally demanding JPEG2000 does.

A MATLAB and C executables of the proposed lossless image compression algorithm can be found at: www.etfbl.net/~aleksej/compress.htm.

VII. Conclusion

In this paper a novel algorithm for lossless image compression is presented. This algorithm is simple and especially tailored to be invariant to bit-depth. The proposed algorithm relies on experimental results on different sets of 12-bit medical and 12-bit and 16-bit natural, previously uncompressed images. Although it is very simple, it is proven to have a similar performance on higher resolution images as standardized algorithms for lossless image compression.

Another contribution of this paper is a detailed analysis of predictor efficiency on 12-bit and 16-bit medical and natural images. The results of this analysis lead to a conclusion that simpler predictors have a satisfactory performance on images with a high dynamic range, compared to computationally expensive predictors based on a minimum mean square error. A novel simple entropy coder is presented and described, as well. The fact that every coded value has an optimal coding context is used to construct a simple and efficient coding algorithm.

Acknowledgment

The authors would like to thank KBC Paprikovac, Banja Luka for providing a set of medical images used in this research.

References

[1] D. Salomon, *Data Compression: The Complete Reference*, Springer, 2007.

[2] D. Taubman and M. Marcellin, *JPEG2000: Image compression Fundamentals, Standards and Practice*, Kluwer Academic Publisher Group, 2004.

[3] I. N. Bankman, editor-in-chief, *Handbook of Medical Image Processing and Analysis*, second edition, Elsevier, 2009.

[4] V. Castelli and L.D. Bergam, *Image Databases, search and Retrieval of Digital Imagery*, John Wiley & Sons, 2002.

[5] M. Wienberger, G. Seroussi, and G. Sapiro, "LOCO-I: A Low Complexity, Context Based, Lossless Image Compression Algorithm," in *Proceedings of the Conference on Data Compression*, pp. 140-149, March/April 1996.

[6] S.D. Rane and G. Sapiro, "Evaluation of JPEG-LS, the New Lossless and Near-Lossless Still Image Compression Standard for Compression of High-Resolution Elevation Data," *IEEE Transactions of Geosciences and Remote Sensing*, vol. 39. No. 10, pp. 2298-2306, October 2001.

[7] X. Wu and N. Memon, "CALIC: A Context-Based Adaptive Lossless Image Codec," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP-96*, Vol. 4, pp. 1890-1893, 1996.

[8] N. Memon and X. Wu, "Recent Developments in Context-Based Predictive Techniques for Lossless Image Compression," *The Computer Journal*, Vol. 40, No. 2-3, 1997.

[9] H. Ye, G. Deng and J.C. Devlin, "Least Squares Approach for Lossless Image Coding," in *Proceedings of the IEEE International Conference on Image Processing*, pp. 901-904, 1998.

[10] G. Deng and H. Ye, "Lossless Image Compression Using Adaptive Predictor Combination, symbol Mapping and Context Filtering," in *Proceedings of the IEEE International Conference on Image Processing*, 1999.

[11] L. Kau and Y. Lin, "Adaptive Lossless Image Coding Using Least Squares Optimization with Edge-Look-Ahead," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, Vol. 52, No. 11, pp. 751-755, November 2005.

[12] F.Y. Hsieh and K.-C. Fan, "A High Performance Lossless Image Coder," *IPPR Conf. on Computer Vision & Graphic Image Processing 2005*.

[13] B. Meyer and P. Tisher, "TMW – A New Method for Lossless Image Compression," *PCS97 1997 Picture Coding Symposium, VDE-Verlag GMBH,* Berlin Germany, pp. 533-538, 1997.

[14] B. Meyer and P. Tisher, "Glicbawls - Grey Level Image Compression By Adaptive Weighted Least Squares," in *Proceedings of Data Compression Conference,* 2001.

[15] R. Starosolski, "Simple Fast and Adaptive Lossless Image Compression Algorithm", *Software – Practice and Experience*, vol. 37, no. 1, pp. 65-91.

[16] A. Avramović, "Lossless Compression of Medical Images Based on Gradient Edge Detection," in *Proceedings of 19th Telecommunications Forum TELFOR 2011,* Belgrade 2011.

[17] E. J. Schlossmacher, "An Iterative technique For Absolute Deviations Curve Fitting," *Journal of the American Statistical Association*, vol. 68, pp. 857-859.