

Effective Context Lossless Image Coding Approach Based on Adaptive Prediction

Grzegorz Ulacha, and Ryszard Stasiński

Abstract—In the paper an effective context based lossless coding technique is presented. Three principal and few auxiliary contexts are defined. The predictor adaptation technique is an improved CoBALP algorithm, denoted CoBALP+. Cumulated predictor error combining 8 bias estimators is calculated. It is shown experimentally that indeed, the new technique is time-effective while it outperforms the well known methods having reasonable time complexity, and is inferior only to extremely computationally complex ones.

Keywords—Adaptive prediction, context coding, image lossless coding, prediction error bias correction.

I. INTRODUCTION

THE end of twenties century was particularly important for lossless image coding methods. In 1996 the CALIC algorithm has been described [16], even today counted among the best ones. The introduced with it context coding idea forms the base for many other widely used techniques, including LOCO-I [14], forming the base for JPEG-LS standard [15]. At that time the CALIC algorithm was too computationally complex to implement in a standard. Today the situation is different, and the place of CALIC has been taken by even better but much more sophisticated techniques like TMWLEGO (2001) [7], WAVE-WLS (2002) [17] and the newest version of MRP 0.5 named VBS & new-cost (2005) [6]. Except for MRP these algorithms are based on a highly promising predictor blending approach. Worth noting is also the application of wavelet transforms to lossless coding, e.g. in JPEG2000 standard [4], however, this approach doesn't seem to be really efficient so far.

The presented in the paper time-effective context based lossless image coding technique is founded on an improved CoBALP [10] predictor coefficient adaptation method, described in subsection C of section II. The 3 principal and 4 auxiliary contexts for data modeling are described in subsection B. Correction of cumulated predictor error involves defining of much more contexts, subsection D, 8 different corrections are combined, subsection E. Finally, data are coded by an adaptive context arithmetic coder outlined in

subsection F. Results of experiments with new coder are summarized in Table 2 and section III. The coder is faster than that for previous CoBALP_{max} implementation, while its performance is excellent, being inferior only to that of excessively complex lossless coding algorithms (TMW^{LEGO}, Multi-WLS).

II. ALGORITHM

A. Predictors

A linear predictor estimates value of coded pixel as a weighted sum of pixels surrounding it:

$$\hat{x}_n = \sum_{j=1}^r b_j \cdot P(j) \quad (1)$$

where $P(j)$ are pixels from the coded pixel x_n neighborhood, and b_j are predictor coefficients (Fig. 1). The coded pixel estimate (rounded to closest integer) is subtracted from the true pixel value, and the difference (prediction error e_n) coded:

$$e_n = x_n - \hat{x}_n \quad (2)$$

Predictor rank r is the number of pixels summed up in (1). The greater the rank, the better the predictor (but usually only to some limit), however, improvements are gradually diminishing, as the most distant pixels provide the worst contribution to the coded one estimate. Hence, neighboring pixels are ordered in accordance with their Euclidean distance from x_n . Fig. 1 illustrates pixel indexing scheme used in (1).

				46	42	38	43			
			37	32	26	24	27	33	39	
		36	29	20	16	14	17	21	30	40
45	31	19	11	8	6	9	12	22	34	
41	25	15	7	3	2	4	10	18	28	44
35	23	13	5	1	x_n					

Fig. 1 Numbering of neighborhood pixels

B. Main Contexts

In our research predictor coefficients for a context have been calculated using the adaptive method described in section C. An obvious expectation is that the greater the number of contexts, the better lossless coder is obtained, but data modeling stages using many contexts are computationally complex. Taking into account a compromise between

G. Ulacha is with the Faculty of Computer Science and Information Systems, West Pomeranian University of Technology, Szczecin, Poland (e-mail: gulacha@wi.ps.pl).

R. Stasiński is with the Faculty of Electronics and Communications, Poznań University of Technology, Poznań, Poland (corresponding author, phone: +48 61 665-3910; fax: +48 61 665-3830; e-mail: rstasins@et.put.poznan.pl).

complexity and coding efficiency, in this paper a method using 3 main and 2 or 4 auxiliary contexts are proposed. The main contexts label classes of neighborhood pixel weighted variance, 2 thresholds define 3 variance ranges. The weights are defined by neighboring pixel distances \bar{d}_j from the coded one:

$$\bar{d}_j = \left((\Delta x_j)^2 + (\Delta y_j)^2 \right)^{\frac{1}{2}} \quad (3)$$

The weighted mean is:

$$\tilde{p} = \frac{1}{\delta} \sum_{j=1}^n \bar{d}_j \cdot P(j) \quad (4)$$

where:

$$\delta = \sum_{j=1}^n \bar{d}_j \quad (5)$$

Finally, the weighted variance is calculated from the formula:

$$\tilde{\sigma}^2 = \frac{1}{\delta} \sum_{j=1}^n \bar{d}_j \cdot (P(j) - \tilde{p})^2 \quad (6)$$

Similarly as in [13], the variance thresholds has been defined as $\beta_1 \cdot \sigma^2$ and $\beta_2 \cdot \sigma^2$ (σ^2 is the arithmetic mean of all weighted variances $\tilde{\sigma}^2$).

The auxiliary contexts are based on edge detection. The idea is taken from [3], but our approach is different. The 3 contexts as above have threshold parameters (optimized experimentally for 45 test images) $\beta_1 = 0.05$, $\beta_2 = 0.7$, $n = 30$. Then, gradients d_h and d_v are computed:

$$\begin{aligned} d_h &= |P(1) - P(5)| + |P(2) - P(3)| + |P(2) - P(4)| \\ d_v &= |P(1) - P(3)| + |P(2) - P(6)| + |P(4) - P(9)| \end{aligned} \quad (7)$$

When neighborhood has high variance (context 3), the difference between horizontal d_h and vertical gradient d_v is tested:

if ($d_h > 2 \cdot d_v$) context = 4;
else if ($d_v > 1.5 \cdot d_h$) context = 5;

For images larger than 256×256 pixels similar comparison is done for context 2:

if ($d_h > 1.7 \cdot d_v$) context = 6;
else if ($d_v > 1.7 \cdot d_h$) context = 7;

C. Adaptation of Predictor Coefficients

In some data modeling techniques predictors are optimized for each coded image. The optimization may be computationally complex, which can be overcome by using a backward predictor coefficients adaptation method. Moreover, adaptive approach cause that predictors are gaining signal tracking properties. The most popular is the LMS method, nevertheless, in our adaptation algorithm better results have been obtained by using improved version of Co-BALP technique.

The original CoBALP method (Context-Based Adaptive Linear Prediction) [10] used 199 contexts; our research has proven that reduction of this number to 7 contexts (or even to 5 for small images) combined with bias cancellation gives a better algorithm. This is in part due to significant shortening of start-up phase adaptation for each of 7 predictors, initially set to zero.

TABLE I
DEFINITIONS OF DIFFERENCES d_j AND SCALING FACTORS η_j

j	d_j	η_j	j	d_j	η_j
1	$P(1) - P(3)$	315	24	$P(12) - P(22)$	70
2	$P(3) - P(2)$	110	25	$P(13) - P(23)$	60
3	$P(2) - P(4)$	250	26	$P(14) - P(24)$	80
4	$P(1) - P(5)$	240	27	$P(15) - P(25)$	23
5	$P(2) - P(6)$	180	28	$P(18) - P(28)$	45
6	$P(3) - P(8)$	130	29	$P(16) - P(26)$	50
7	$P(3) - P(7)$	100	30	$P(24) - P(27)$	40
8	$P(4) - P(9)$	140	31	$P(19) - P(29)$	50
9	$P(4) - P(10)$	90	32	$P(22) - P(30)$	55
10	$P(2) - P(8)$	100	33	$P(19) - P(31)$	45
11	$P(6) - P(14)$	100	34	$P(20) - P(32)$	55
12	$P(4) - P(12)$	100	35	$P(21) - P(33)$	70
13	$P(5) - P(13)$	100	36	$P(28) - P(34)$	50
14	$P(7) - P(15)$	55	37	$P(23) - P(35)$	60
15	$P(10) - P(18)$	80	38	$P(24) - P(38)$	80
16	$P(1) - P(2)$	260	39	$P(31) - P(36)$	40
17	$P(3) - P(11)$	80	40	$P(32) - P(37)$	55
18	$P(14) - P(17)$	45	41	$P(30) - P(39)$	15
19	$P(8) - P(16)$	90	42	$P(34) - P(40)$	90
20	$P(6) - P(9)$	130	43	$P(35) - P(41)$	23
21	$P(11) - P(19)$	55	44	$P(26) - P(42)$	25
22	$P(11) - P(20)$	40	45	$P(41) - P(45)$	20
23	$P(12) - P(21)$	70	46	$P(32) - P(46)$	33

The reason why CoBALP is more effective than LMS can be attributed to modification of adaptation formula, instead of pixels x_k their differences d_k are used:

$$\hat{x}_n = P(2) + \sum_{j=1}^r b_j \cdot d_j, \quad (8)$$

where coefficients b_j of the currently used predictor are adapted as follows:

$$b_j(n+1) = b_j(n) + \mu_j \cdot \bar{e}_n \cdot d_j \quad (9)$$

with upper-bound $\varphi = 7$ on \bar{e}_n :

$$\bar{e}_n = \text{sgn}(e_n) \cdot \min\{|x_n - \hat{x}_n|, \varphi\} \quad (10)$$

The value of μ_j depends on scaling factor η_j , and variable m_j (initially set to 0):

$$\mu_j = \frac{\eta_j}{1 + m_j} \cdot 10^{-6}. \quad (11)$$

m_j is computed iteratively:

$$m_j(n+1) = \frac{7}{8} m_j(n) + \frac{1}{8} |d_j|. \quad (12)$$

In our experiments the predictor ranks has been increased from 9 [10] to $r = 46$, hence, definitions of differences d_j modified appropriately. The difference definitions and values of associated with them scaling factors η_j are presented in Table I, $P(k)$, k as in Fig. 1.

D. Contexts for Correcting Prediction Error Bias

The idea of error bias correction has been introduced in [16] (CALIC), where apart from 7 main contexts, additional 576 ones have been used to correct cumulated prediction error. Let us analyze typical approaches to context definition.

Approach 1

Context definitions in [16] use n samples values encircling the coded one and their mean. For $n = 8$ the samples are $P(1)$, $P(2)$, $P(3)$, $P(4)$, $P(5)$, $P(6)$, $\text{GradNorth} = 2 \cdot P(2) - P(6)$ and $\text{GradWest} = 2 \cdot P(1) - P(5)$. We take an 8-bit word and set to 1 its bit if the corresponding sample value is greater than the mean, and to 0 in the opposite case, and 256 contexts are defined. Additionally, we can quantize pixel standard deviation to Q values, the number of contexts grows to $Q \cdot 2^n$.

In this paper $n = 8$ and $Q = 4$ values are used, which gives 1024 contexts. Arithmetic mean is replaced by expectation value \hat{x}_n associated with predictor coefficient adaptation method, section C. The variance value (multiplied by 8) is:

$$\hat{\sigma}_n^2 = (\hat{x}_n - \text{GradNorth}_n)^2 + (\hat{x}_n - \text{GradWest}_n)^2 + \sum_{i=1}^6 (\hat{x}_n - P(i))^2 \quad (13)$$

The quantization levels have been defined by 3 experimentally found thresholds for $\hat{\sigma}_n^2$ values: 400, 2500, 8000. Similar idea has been presented in [2].

Approach 2

The second context defining approach proposed here is similar to that used in JPEG-LS [15]. Let us define 6 classes of the neighbor pixel differences d_1 , d_2 , d_3 using difference sign and 2 thresholds q_1 and q_2 , which gives $6^3 = 216$ contexts. Additionally, 3 distant differences are analyzed, d_4 , d_5 , d_8 , their absolute values are divided into 2 classes using threshold q_3 , and 1728 contexts are obtained. Definitions of differences d_i can be found in the following algorithm:

$d_1 = P(1) - P(3)$; $d_2 = P(3) - P(2)$; $d_3 = P(2) - P(4)$;
 $d_4 = \text{fabs}(P(1) - P(5))$; $d_5 = \text{fabs}(P(2) - P(6))$;
 $d_8 = \text{fabs}(P(4) - P(9))$; context = 0;

```
if ( $d_1 < 0$ ) {
    if ( $d_1 > -q_1$ ) context = 2;
    else if ( $d_1 > -q_2$ ) context = 1;
}
else {
    if ( $d_1 < q_1$ ) context = 3;
    else if ( $d_1 < q_2$ ) context = 4;
```

```
    else context = 5;
}
if ( $d_2 < 0$ ) {
    if ( $d_2 > -q_1$ ) context += 2*6;
    else if ( $d_2 > -q_2$ ) context += 1*6;
}
else {
    if ( $d_2 < q_1$ ) context += 3*6;
    else if ( $d_2 < q_2$ ) context += 4*6;
    else context += 5*6;
}
if ( $d_3 < 0$ ) {
    if ( $d_3 > -q_1$ ) context += 2*36;
    else if ( $d_3 > -q_2$ ) context += 1*36;
}
else {
    if ( $d_3 < q_1$ ) context += 3*36;
    else if ( $d_3 < q_2$ ) context += 4*36;
    else context += 5*36;
}
if ( $d_4 > q_3$ ) context += 216;
if ( $d_5 > q_3$ ) context += 432;
if ( $d_8 > q_3$ ) context += 864;
```

The experimentally found thresholds q_i are {5, 18, 20}.

Approach 3

The third approach is based on simplified vector quantization method, see [3]. Initial data analysis resulted in defining 16 constant vectors (centroids), each containing 7 elements, 4 errors and 3 pixels from the coded pixel neighborhood: $V = \{e(1), e(2), e(3), e(4), P(1), P(2), P(4)\}$. Centroids are initialized as follows:

```
for ( $y = 0$ ;  $y < 16$ ;  $y++$ ) {
    count[y] = 1;
    for ( $x = 0$ ;  $x < 4$ ;  $x++$ )
        centroid[y][x] = (( $y > x$ ) & 1) * 2 - 1;
    for ( $x = 4$ ;  $x < 7$ ;  $x++$ )
        centroid[y][x] = ( $y < 4$ );
}
```

A simplified adaptive method for updating centroids provides Euclidean distances of current vector V from all 16 centroids. The 4-bit label j of the closest centroid is concatenated with other bits (defined below) into a 10-bit context number (i.e. they are 1024 contexts). Adaptation of a centroid is done as follows:

$$\text{centroid}[j][i] = \frac{\text{count}[j] \cdot \text{centroid}[j][i] + V[i]}{\text{count}[j] + 1}, \quad (14)$$

for each $i = \{0, \dots, 6\}$. Then the counter $\text{count}[j]$ is increased by 1.

Next 4 bits of context number are decisions if the neighboring pixel $P(1)$, $P(2)$, $P(3)$, $P(4)$ values are close enough to expected coded pixel value \hat{x}_n : $|\hat{x}_n - P(i)| \geq 7$, $i =$

1, 2, 3, 4. Last bits are obtained from comparison of \hat{x}_n with values of $P(1)$ and $P(2)$, if $P(i) < \hat{x}_n$, then the bit is zero, $i = 1, 2$.

Approach 4

Similarly as in approach 3, only 4 neighboring pixels are considered, $P(1)$, $P(2)$, $P(3)$, $P(4)$, and 10-bit context number is constructed. The arithmetic mean of pixels \bar{x} is computed, then pixels are divided into two groups – having values smaller and greater than the mean, and arithmetic means for both groups calculated: \bar{x}_l and \bar{x}_h , respectively. The values \bar{x} , \bar{x}_l and \bar{x}_h are used as classification thresholds. As they are 4 pixels, and each is described by a 2-bit class number, 8 bits of context number are defined. The remaining 2 bits are obtained from a 4-level quantizer applied to the difference $\bar{x}_h - \bar{x}_l$, its thresholds are $\{4, 12, 30\}$. The difference value exhibits correlation with variance of the analyzed pixels $P(1)$, $P(2)$, $P(3)$, and $P(4)$.

D. Correcting the Cumulated Prediction Error

Some particular properties of coded pixel neighborhoods may induce (transient, but long lasting) DC components in prediction errors associated with contexts. Adaptive methods of removing the DC component, or bias, are proposed, called also context-based error correction techniques.

Context-based error correction methods consist in using occurrence number and cumulated error for each context for correcting current prediction error [16]. The algorithms are presented below, notation used: i is the context number, $B[i]$ is the current value of cumulated prediction error for the context i , $N[i]$ is the i -th context occurrence value, and $C[i]$ is the current correction value. This value should be added to the predictor output for the i -th context the next time it occurs:

$$\hat{x}_n = \hat{x}_n + C[i] \quad (15)$$

The $C[i]$ adaptation algorithm for CALIC is as follows:

Initial conditions: $B[i] = C[i] = 0$, $N[i] = 4$ for each i ;

$B[i] = B[i] + e_n$;
 $N[i] = N[i] + 1$;
 $C[i] = B[i]/N[i]$;

Algorithm for JPEG-LS method:

Initial conditions: $B[i] = C[i] = 0$, $N[i] = 4$ for each i ;

$B[i] = B[i] + e_n$;
 $N[i] = N[i] + 1$;
 if $(B[i] \leq -N[i])$ {
 $C = C[i] - 1$;
 $B[i] = B[i] + N[i]$;
 if $(B[i] \leq -N[i])$ $B[i] = -N + 1$;
 }
 }

else if $(B[i] > 0)$ {
 $C = C[i] + 1$;
 $B[i] = B[i] - N[i]$;
 if $(B[i] > 0)$ $B[i] = 0$;
 }

In both algorithms “forgetting mechanism” is introduced, it improves their signal tracking properties:

if $(N[i] > 127)$ {
 $N[i] = 64$;
 $B[i] = B[i]/2$;
 }

In our method we are computing weighted sum of outputs from both bias canceling methods applied to all four approaches to context definition, described in subsection II D, i.e. they are in total 8 components of the sum. In experiments the component weights have been constant, but in general it is possible to adapt their values for each coded image. Of course, this increases the method computational complexity significantly.

High method efficiency may be attributed to its robustness – in some cases an estimate of context-based error fails to decrease the actual prediction error, but almost for sure those defined in a different way work correctly. Moreover, the estimate mixing ameliorates asymmetry of error distribution. Namely, mean value of $B[i]$ may be located outside the highest $B[i]$ histogram slot, the problem has been signaled in [11]. Experiments show that in 70% of cases results for the prediction correction approach are not worse than when no bias cancellation is implemented.

E. Note on Arithmetic Coder

The most effective entropy coding method known today is adaptive context arithmetic coding. Such coder has been used for testing presented here data modeling techniques, it has been based on the description from [8]. Additionally, coder extensions from [1] have been considered, and their improved versions implemented. They are 16 coder contexts, determined by 4 nearest neighboring pixels, and 10 predictor error values. A nonlinear quantizer determining 18 groups of absolute values of predictor errors has been defined, it has controlled the work of adaptive n -ary arithmetic coder. Sign bit is coded by a 16-context adaptive binary arithmetic coder. Coder detailed description can be found in [12].

F. Summary – Algorithms Flow-Chart

For each image pixel do:

- 1) Determine the main context number (section B).
- 2) Predict the pixel value (1).
- 3) Adapt the current main context predictor coefficients (section C).
- 4) Determine 4 bias-correction context numbers using all approaches from section D.
- 5) Compute the bias correction value in accordance with section E.

TABLE II
AVERAGE BIT PER PIXEL RATES FOR 9 TEST IMAGES

Images	CoBALP _{max} [10]	CoBALP ₊	HBB [9]	CALIC [16]	LAT-RLMS [5]	TMW ^{LEGO} [7]	Multi-WLS [17]
Balloon	2.853	2.677	2.80	2.78	2.75	2.60	2.60
Barb	4.176	4.043	4.28	4.31	4.15	3.84	3.75
Barb2	4.440	4.316	4.48	4.46	4.45	4.24	4.18
Board	3.492	3.367	3.54	3.51	3.48	3.27	3.27
Boats	3.780	3.627	3.80	3.78	3.74	3.53	3.53
Girl	3.696	3.555	3.74	3.72	3.68	3.47	3.45
Gold	4.382	4.266	4.37	4.35	4.34	4.22	4.20
Hotel	4.219	4.101	4.27	4.18	4.21	4.01	4.01
Zelda	3.749	3.564	3.72	3.69	3.61	3.50	3.51
Mean	3.865	3.724	3.889	3.864	3.823	3.631	3.611

6) Correct the predicted pixel value (15), and use it for calculation of the prediction error e_n (2).

7) Code e_n using the context arithmetic coder outlined in section F.

III. EXPERIMENTS

The new, improved version of CoBALP algorithm, denoted CoBALP₊, has been compared with other well known methods, including the best version of original CoBALP, Table II. As can be seen, the new algorithm is outperformed only by techniques having prohibitively high computational complexity (TMW^{LEGO} and Multi-WLS, several tens of minutes of coding time per image on Pentium 4). In contrast, coding time of our algorithm for Lennagrey image (512×512 pixels) on 2.8 GHz Pentium 4 has been only 2.66 seconds (not optimized C code), or equivalently – 96.24 Kpixels per second. The test of the original CoBALP algorithm is done using the most efficient but slowest option of program WaveConvert, Version 1.2, from the year 2002 by T. Strutz (hence, notation CoBALP_{max}). In our experiments coding time for this option was approximately two times longer than for our code. The complexity of three consecutive methods (HBB [9], CALIC [16], LAT-RLMS [5]) is of the same rank as that of our technique. Of course, extremely complex last two algorithms are added for reference, only.

IV. CONCLUSION

A new time-effective lossless context coding technique has been presented in the paper. The new algorithm components are: improved CoBALP version used for adaptation of predictor coefficients, two-level contexts, 3 principal and few auxiliary ones, sophisticated prediction error bias removing formula, and adaptive context-driven arithmetic coder. It has been shown experimentally that average bit per pixel rates for the method are lower than those for algorithms having similar or smaller time complexities, only extremely complex ones are somewhat better in this respect. This means that the new algorithm is an excellent tool of advanced lossless image coding.

ACKNOWLEDGMENT

The work has been partially sponsored by Polish Ministry of Science and Higher Education grant “Algorytmy bezstratnej i prawie-bezstratnej kompresji danych multimedialnych dedykowane architekturze Networks on Chip”.

REFERENCES

- [1] G. Deng, Symbol mapping and context filtering techniques for lossless image compression, *Proceedings of IEEE 1998 International Conference on Image Processing*, v.1, pp. 526-529, Chicago, USA, Oct. 1998.
- [2] F. Golchin, K. K. Paliwal, A lossless image coder with context classification, adaptive prediction and adaptive entropy coding, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, Washington, USA, May 1998, pp. 2545-2548.
- [3] Lih-Jen Kau, Yuan-Pei Lin, Lossless image coding using a switching predictor with run-length encodings, *Proceedings of IEEE International Conference on Multimedia and Expo*, June 2004, vol. 2, pp. 1155-1158.
- [4] M. Marcellin, M. Gormish, A. Bilgin, M. Boliek, An Overview of JPEG-2000, *Proceedings of Data Compression Conference*, Snowbird, Utah, March 2000, pp. 523-541.
- [5] S. Marusic, G. Deng, Adaptive prediction for lossless image compression, *Signal Processing: Image Communications*, May 2002, vol. 17, pp. 363-372.
- [6] I. Matsuda, N. Ozaki, Y. Umez, S. Itoh, Lossless coding using Variable Blok-Size adaptive prediction optimized for each image, *Proceedings of EUSIPCO-05*, (CDROM), September 2005.
- [7] B. Meyer, P. Tischer, TMW^{LEGO} - An Object Oriented Image Modelling Framework, *Proceedings of Data Compression Conference 2001*, pp. 504.
- [8] K. Sayood, *Introduction to Data Compression*, 2nd edition, Morgan Kaufmann Publ., 2002.
- [9] T. Seemann, P. Tisher, B. Meyer, History-Based Blending of Image Sub-Predictors, *Proceedings of Picture Coding Symp.*, Berlin, Germany, 1997, pp. 147-151.
- [10] T. Strutz, Context-Based Adaptive Linear Prediction for Lossless Image Coding, *4th International ITG Conference on Source and Channel Coding*, Berlin, Germany, 28-30 January, 2002, pp. 105-109.
- [11] C. Topal, Ö. N. Gerek, Pdf sharpening for multichannel predictive coders, *Proceedings of EUSIPCO-06 CD*, Sept. 2006.
- [12] G. Ulacha, R. Stasiński, A new context lossless image coding algorithm based on adaptive context arithmetic coder, *Proceedings of 15th International Workshop on Systems, Signals & Image Processing - IWSSIP 2008*, Bratislava, Slovak Republic 2008, pp. 45-48.
- [13] G. Ulacha, R. Stasiński, New simple context-based predictive technique for lossless image compression, *Proceedings of EUSIPCO-07* (CDROM), Poznań, Poland, September 2007, pp. 990-993.

- [14] M. J. Weinberger, G. Seroussi, G. Sapiro, LOCO-I: A low complexity, context-based, lossless image compression algorithm. *Proceedings of Data Compression Conference*, March-April 1996, Snowbird, Utah, pp. 140-149.
- [15] M. J. Weinberger, G. Seroussi, G. Sapiro „LOCO-I: Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS”, *IEEE Trans. on Image Processing*, vol. 9, No. 8, August 2000, pp. 1309-1324.
- [16] X. Wu, N. D. Memon „CALIC – A Context Based Adaptive Lossless Image Coding Scheme”, *IEEE Trans. on Communications*, May 1996, vol. 45, pp. 437-444.
- [17] H. Ye, A study on lossless compression of greyscale images, PhD thesis, Department of Electronic Engineering, La Trobe University, October 2002.