Lossless Bit-plane Compression of Images with Context Tree Modeling

Minjie Chen, Pasi Fränti
School of Computing
University of Eastern Finland
Joensuu, Finland
{mchen,franti}@cs.joensuu.fi

Mantao Xu
School of Electrical Engineering
Shanghai Dianji University
Shanghai, China
Mantao.xu@gmail.com

***Abstract*—Bit-plane coding has been widely used in lossless compression of gray-scale images or color palette images. In the context-based encoding procedure, the values of previous significant planes are also considered, which is simplified as surrounding pixel's expectation values. In this paper, we further improved the encoding scheme by context-tree modeling, in which both the order and depth of the context template is optimized. In probability estimation scheme, a forgetting factor and context weighting are utilized. According to our experiments, the proposed method compete with the existing algorithms in lossless compression of gray-scale images, and outperforms all competitive methods in compression or color palette images.**

I. INTRODUCTION

The motivation of image compression is to represent an input image in smaller space for saving storage capacity. Lossless compression aims at doing this without any loss of information so that the original image can be restored identically without any loss after the decompression. Most image compression algorithms are based on predictive modeling, such as JPEG-LS [1] and CALIC [2]. The current encoding pixel is predicted by a function of already encoded neighborhood pixels and subsequently the difference between the actual and predicted value, which is called *prediction error*, is encoded. Despite of its apparent simplicity, predictive coding is quite effective and widely practiced in the state of the art compression algorithms.

A prevailing practice is to decompose or represent the image of interest using a set of palette images, which are used for representing computer generated imagery in web and other multimedia platforms. Instead of using smoothly continuous tone of colors, these images are interpreted by using only the pixel level detailed structures. They cannot be compressed by predictive methods (CALIC and JPEG-LS) because they are designed for continuous-tone images. A more intuitive realization is to use *bit-plane coding*: divide the image into binary layers according to the colors or bit values, and then compress the bit-planes separately by a highly optimized binary image compression method. This approach is general since any bit-plane compressor can be used after the bit-plane separation without any modification needed to adapt to the image type.

In bi-level compression, most conventional algorithms encode the image pixel by pixel in raster scan order using context-based probability and arithmetic coding. The probability of local context is estimated according to different strategies. For example, in JBIG [5], a 10-pixel fixed template is used. *Context tree modeling* has been extensively studied [6, 7] for the sake of attacking the context dilution problem. State-of-the-art context weighting approach [8] is considered that adaptively weights multiple fixed context models based on their relative accuracy. In [8, 9], a forgetting factor is incorporated so that a recent pixels have greater influence on the probability estimation of the current pixel than the earlier pixels. Compared with traditional methods that estimate the conditional probability globally, this algorithm is efficient especially when different patterns exist in the same image. It has been reported that the compression ratio increases about 5%-10% to that of JBIG for CCITT dataset [8, 9].

Another challenge in the bit-plane separation is that the semantic image structures are quite often broken especially at the lower bit planes. It is therefore difficult to capture the spatial dependencies efficiently even if a very large larger context template is used. One idea is to use multi-layer context template, following with a greedy context reordering and tree pruning process [3]. Expectation-based bit-plane coding has been proposed [4] by using prediction technique based on expectation value for representing the context values. Since context bits are computed by its expectation values, it is realistic to include future pixels in the context template during encoding the less significant bit-planes (LSB). In this paper, we combine the approach in [4] with a highly efficient context tree model to the best of the compression performance. The context ordering and tree pruning are executed in order to minimize the sum of adaptive code lengths for the encoding image.

978-1-4244-6878-2/10/$26.00 ©2010 IEEE

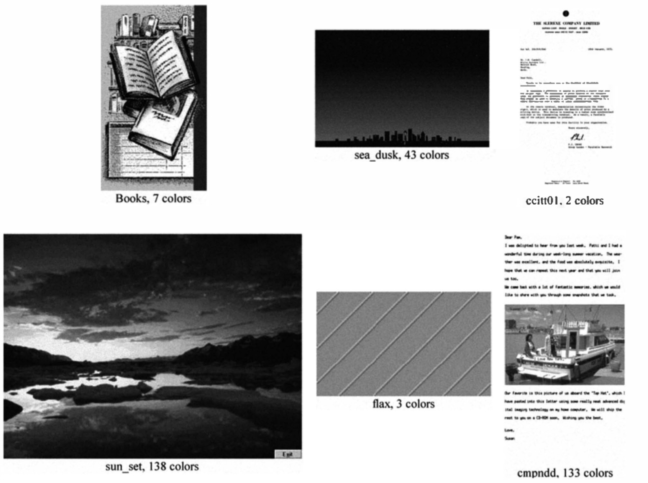

Figure 1. Examples of palette images.

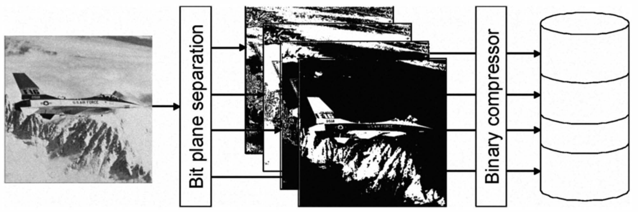

Figure 2. Idea of the bit-plane coding

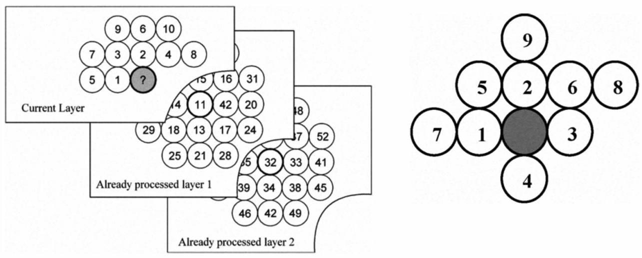

Figure 3. Example of context template. MCT(left) and expectation-based bit-plane coding(right)

II. PROPOSED METHOD

A. Bit-plane Coding

Bit-plane coding can be used for compression gray-scale images. Commonly, it consists of two independent stages, decomposition and lossless compression. An example of bit-plane coding can be seen in Fig.2.

In the first stage, the gray-scale image is decomposed into a set of binary images (layers). Different decomposition methods are proposed such as simple bit-plane separation (BPS), gray code separation (GCS), prediction error separation (PCS) and gray code prediction error separation (GCPES). A reasonable summary can be found in [3].

In the second stage, a lossless compression algorithm is then applied for the values on the bit-planes. An intuitive idea is to use some bi-level image compression standard such as JBIG for coding the bit-planes. However, this approach is only efficient for the most significant bit-planes (MSB). In practice, there is low correlation of the pixels on the less significant bit-planes (LSB) and the bit-rate is close to 1bit/pixel. In [3, 11], multi-layer context tree modeling (MCT) has been proposed, in which the context pixel is selected not only from the same bit-plane, but also from previously coded bit-planes, see fig.3. Pre-calculation if every gray-level is appeared in the processing image can also be considered. When gray-levels of the image are less than 256, in some LSBs, the coding cost for some pixels can be saved. For example, for a 8-bit image, if there are only one gray-level appeared on the image between 0 and 2^k-1, encoding process can be skipped for these pixels on bit-plane 0,1,…k-1.

For color-indexed images, they are typically compressed by GIF and PNG, but much better results have been obtained by specialized methods such as PWC [13], MCT [3] or GCT [12]. Bit-plane separation can also be made in progressive manner using color clustering [14, 15]. The method in [14] provided two improvements over the first approach in [15]: better clustering method using merge-based algorithm rather than the splitting technique, and better compression method by context tree model rather than fixed-side template. They are useful when color progression is needed, but otherwise best non-progressive methods are superior in general. The basic idea, however, is applicable in the bit-plane separation step and therefore relevant here.

The method in [16] uses variable context size by decreasing the number of template pixels towards the LSB bit-planes. This is reasonable heuristic approach since the pixels in the higher-level bit planes are more correlated and wider exploitation via larger context template can be used.

A position paper can be found in [10] that includes partial accessibility and other factors relevant to practical implementations. This shows the usability of a modular approach where the problem is simplified into smaller sub-problems by dividing the image into binary layers by color decomposition, and smaller sub-blocks for random access to partial images.

B. Expectation-based bit-plane coding

Expectation-based bit-plane coding (EBC) was recently proposed in [4]. For each encoding pixels and its neighborhood in the context, expectation values are calculated based on the values of higher bit-planes which is already encoded. Suppose $x = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$, x_i is the value for pixel x at bit-plane i. When the n^{th} bit-plane is being encoded, if pixel x is already encoded at bit-plane i, its expectation value is formulated as:

$$E(x) = \sum_{i=n}^{7} 2^i + (2^{n-1} - 1) \quad (1)$$

If x has not been encoded at current bit-plane, expectation value is defined as:

$$E(x) = \sum_{i=n+1}^{7} 2^i + (2^n - 1) \quad (2)$$

During encoding process, the context is determined by

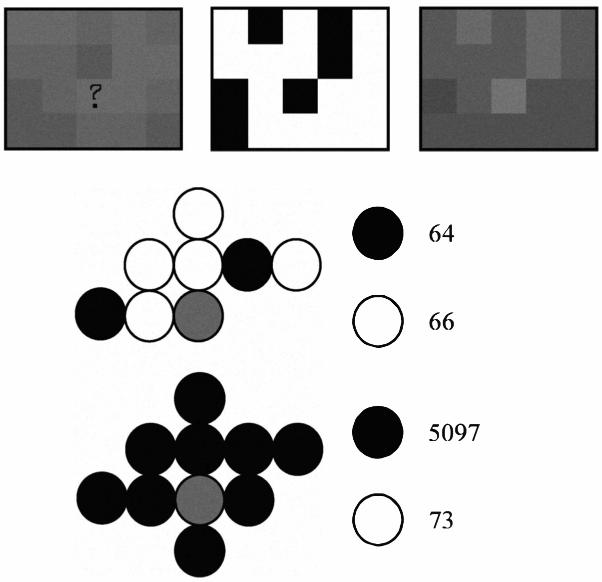

Figure 4. Example of the context value calculation during encoding bit-plane 4, where “?” is the current pixel. Original image (top-left), binary image of the current bit-plane (top-middle), expectation value (top-right). The context selection and the sample frequencies for the current pixel are shown using the bit-plane (middle), and using the expectation value (bottom).

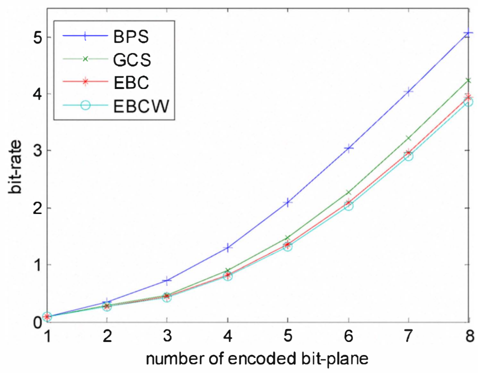

Figure 5. Coding cost of different bit-plane coding algorithm for image “airplane”.

these expectation values. If the expected value of the context pixel is lower than the expected value of the current pixel, then 0 is used, otherwise 1 is used. A 9-pixel fixed template was used (see Fig.3) and the number of template pixels is decreased gradually to 8, 7, 6, and 5 for encoding the four LSBs, respectively.

The reasoning behind this idea is that the bit-value in the current bit-plane itself is meaningful only if it is the first 1-bit in the bit-plane. Otherwise, it is merely a refining bit that fine-tunes the value of that particular pixel further and its exact value is meaningless unless the exact magnitude is known. Moreover, future pixels can also be selected in the context template and it is especially efficient for LSBs. In [4], for the future context pixels, when the expected value of a context equals to that of the current pixel, context value is set to 1. Instead, we calculate two statistical distributions by setting the value to 0 and 1 correspondingly, and apply context weighting process [8], which will be demonstrated in Section 2.2. An example of the context selection in the expectation-based bit-plane coding is shown in Fig.4.

In Fig.5, a comparison of different bit-plane coding approaches is demonstrated using simple bit-plane separation (BPS), gray code separation (GCS), expectation-based bit-plane coding (EBC) and the improved (weighted) variant of the expectation-based bit-plane coding (WEBC). The last one is selected as the recommended bit-plane coding method in the rest of the paper due to its good performance in comparison to the BPS and GCS variants.

C. *Probability estimation*

In adaptive coding framework, probability estimation is needed for encoding the current pixel. A general formulation of this is done by *Krichevski-Trofimov* (K-T) estimator, which collects *global statistics* for different context **c** during encoding procedure. In BACIC algorithm [9], a forgetting factor μ is incorporated which gives higher influence for recently encoded pixels. The probability of the current pixel is estimated as:

$$p^{\wedge}(x_n = 1 \mid \mathbf{c}) = \frac{r_c(n-1)+\Delta}{s_c(n-1)+2\Delta} \tag{3}$$

where Δ is a bias factor (Δ = 0.006), after the current pixel has been encoded, r_c and s_c are updated as:

$$r_c(n) = x_n + \mu r_c(n-1) \tag{4}$$

$$s_c(n) = 1 + \mu s_c(n-1) \tag{5}$$

with $r_c(0) = 1$ and $s_c(0)=2$.

In fact, *global statistics* can also be considered as a special case when μ=1. Arithmetic coding is performed based on this probability estimation. The total coding cost by arithmetic coding can be calculated as:

$$l(x_1^n \mid \mathbf{c}) = -\sum_{i=1}^{n} \log_2 p^{\wedge}(x_i \mid \mathbf{c}) \tag{6}$$

The reason for using the forgetting factor is that there may exists different patterns in the same image. Fig, 6 shows an example where global statistics are not a suitable solution. For the given context, at left side of the image, white color is dominant whereas black color is the dominant color at the right side. If *Krichevski-Trofimov* (K-T) estimator is used, the probability estimation of a given context will converge to a constant value at the end. In this case, BACIC can achieve better compression ratio. An example is shown in Fig.7. We should mention that for the bit-plane coding, in order to prevent possible over-estimation in the probability estimation, smaller μ can be selected for MSBs, and larger μ for LSBs.

Context weighting can also be considered to improve the probability estimation [8]. Suppose $\mathbf{c}_1$ and $\mathbf{c}_2$ are the two context models, in which the value of those contexts with equal expected value are set as 0 and 1 correspondingly. The weighted probability can be estimated by:

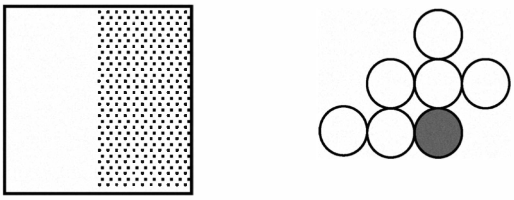

Figure 6. An example of image with multiple patterns(left), context example(right) which has a dominated probability as white color at left side of the image but with dominated black color at right side.

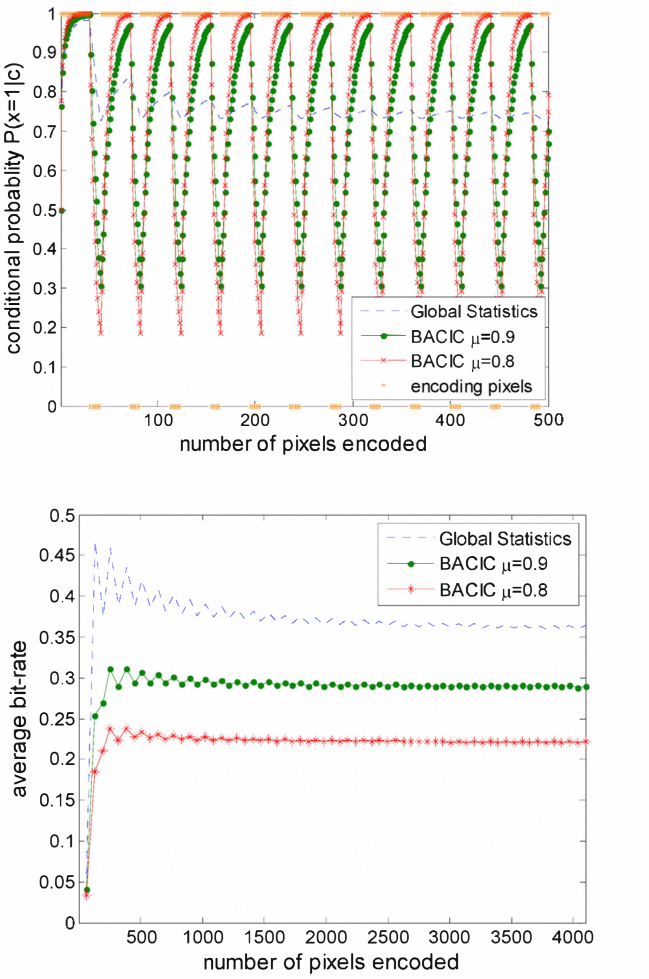

Figure 7. Example of probability estimation and bit-rate for the compression of the image in Fig. 6 under the given context using global statistic and BACIC.

$$\hat{p}_w(x_n = 1 \mid \mathbf{c}_w(n)) = \hat{p}_1(x_n = 1 \mid \mathbf{c}_1(n))\alpha_{n-1} + \hat{p}_2(x_n = 1 \mid \mathbf{c}_2(n))\beta_{n-1} \quad (7)$$

Where α and β are the weight of two context model, which is updated by:

$$\alpha_n = \frac{\hat{p}_1(x_n \mid \mathbf{c}_1(n))\alpha_{n-1}^{\mu_2}}{\hat{p}_1(x_n \mid \mathbf{c}_1(n))\alpha_{n-1}^{\mu_2} + \hat{p}_2(x_n \mid \mathbf{c}_2(n))\beta_{n-1}^{\mu_2}} \quad (8)$$

$$\beta_n = \frac{\hat{p}_2(x_n \mid \mathbf{c}_2(n))\beta_{n-1}^{\mu_2}}{\hat{p}_1(x_n \mid \mathbf{c}_1(n))\alpha_{n-1}^{\mu_2} + \hat{p}_2(x_n \mid \mathbf{c}_2(n))\beta_{n-1}^{\mu_2}} \quad (9)$$

μ_2 is the forgetting factor set to μ_2=0.975.

D. Context Tree Modeling

Compared with static context template, the use of context tree model as in MCT is more general than this ad hoc heuristic. Context tree modeling [3, 6, 11, 12] provides a more flexible approach for modeling the contexts so that a larger number of neighbor pixels can be taken into account without the context dilution problem.

Classical context tree modeling technique has been widely used by data compression community with time complexity O(N), where N is the length of a data sequence. A context tree is built by estimating the count statistics via a sequential traversal of the image pixel-by-pixel. Each node of the context tree represents a single context by storing the count statistics of black and white color for the current pixel relative to the node of context. The tree can be constructed beforehand (static approach) or optimized directly to the image to be compressed (semi-adaptive approach). In context tree modeling, both the order of the context template and the depth of the tree can be optimized in order to minimize the coding cost. It is implemented by a two-step process as follows.

Firstly, a greedy context reordering process is done [6, 7]. Given a predefined search area (see Fig. 8), at every step, we recursively search the context minimizing the sum of adaptive code lengths of (6) after splitting on the context tree. Suppose a predefined search area with N pixels $\{y_0,\ldots,y_{N-1}\}$, the sequence of ordered context with depth $d(d<N)$ is $\{j_0,\ldots,j_{d-1}\}$. j_d is determined by:

$$j_d = \underset{j_d \in \{0,1,\ldots,N-1\} \setminus \{j_0, j_1, \ldots, j_{d-1}\}}{\arg\min} l(x \mid \{j_0, j_1, \ldots, j_{d-1}\} \cup j_d) \quad (10)$$

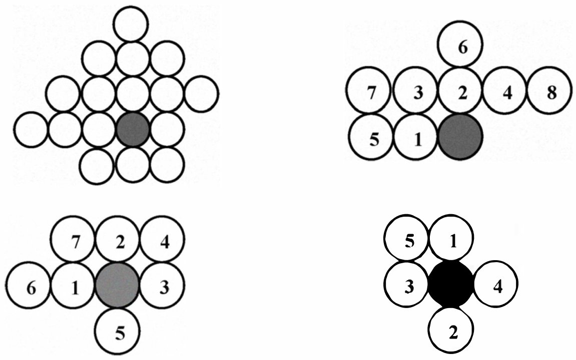

Figure 8. predefined search area (top-left), and optimized context order on bit-planes 7, 3, 0 respectively.

This recursive search stops when the node splitting does not give any further improvement for reducing the adaptive code length. In this way, optimal context size can be obtained. For example, for MSBs with higher correlations, a larger context template is used, while a smaller context template is used for LSBs. Meanwhile, the context reordering process makes context near the root node disperse the statistics most and simultaneously context in deeper tree levels with less importance. In this way, the subsequent pruning process can operate more efficiently. An example of the ordered context templates are shown in Fig.9.

After context reordering process, the context tree is constructed. The tree pruning process starts from the leaves of the full grown and reordered tree, which was obtained as a result of the first stage. Each nodes of that tree is visited using a bottom-up strategy by evaluating a recursively defined cost value J, which is the sum of the average adaptive code length and the model description cost. For example, for the d-depth context tree modeling, the given node s at depth d_0 has two child node s_{ch0} and s_{ch1}. If the sum of the coding costs of the s_{ch0} and s_{ch1} is greater than or equal to the coding costs at node s, pruning is done for the branches below the parent node s. Otherwise, the sum of the coding cost for the nodes s_{ch0} and s_{ch1} plus a model cost term mc is assigned to the parent node s:

$$J(s) = \begin{cases} l(x_1^n \mid s) + mc(d_0), \\ \qquad \text{if } l(x_1^n \mid s) \le l(x_1^n \mid s_{ch0}) + l(x_1^n \mid s_{ch1}) \\ l(x_1^n \mid s_{ch0}) + l(x_1^n \mid s_{ch1}) + mc(d_0), \text{otherwise} \end{cases} \tag{11}$$

where $mc(d_0) = \log_2(d - d_0 + 1)$.

III. EXPERIMENTS

In our experiments, two test image sets are evaluated. The first set consists of five classical 8 bits per pixel test images of size 512 × 512: airplane, bridge, couple, crowd and lena. This test set represents a class of natural images with smooth color gradation. The second test set represents a class of palette images (see Fig. 2) consisting of eight images. Such images can be web graphics, schemes, maps, slides, and engineering drawings, for example. The performance is compared with lossless compression algorithm, such as JBIG, JPEG-LS and MCT, which is listed in table I and II. From experiments, we found that the proposed bit-plane coding algorithm is efficient for both datasets. For image "bridge", as it only has 64 gray-levels, less bit-rate is needed by the proposed bit-plane coding algorithm.

We also evaluate the significance of every component in our algorithm. The difference of compression performance is evaluated when forgetting factor, context weighting and context tree modeling are applied on the expectation-based bit-plane coding algorithm, which is visualized on Fig 9.

IV. CONCLUSION

We have proposed an efficient bit-plane coding algorithm for lossless compression of gray-scale image or color palette images. In the proposed algorithm, the context value is determined by the expectation values of the surrounding pixels implemented by context-tree structure, in which both the order and depth of the context template is optimized in each bit-plane. A forgetting factor and context weighting are incorporated to achieve a higher influence of the recent pixels.

According to our experiments, bit-plane coding algorithm is compete with the existing algorithms in lossless compression of gray-scale images, and outperforms all competitive methods in compression or color palette images. This algorithm can also be used for progressive transmission of the images.

TABLE I. BIT RATES FOR LOSSLESS COMPRESSION OF NATURAL IMAGES (BITS PER PIXEL)

Image	JBIG	JPEG-LS	MCT	Proposed
airplane	4.24	3.81	4.12	**3.80**
bridge	5.04	5.50	4.93	**3.51**
couple	5.07	**4.26**	4.64	4.37
crowd	4.45	**3.91**	4.17	3.95
lena	4.91	**4.23**	4.49	4.27

TABLE II. BIT RATES FOR LOSSLESS COMPRESSION OF PALETTE IMAGES (BITS PER PIXEL)

Image	JBIG	JPEG-LS	MCT	Proposed
benjerry	2.06	1.91	**0.85**	1.05
books	3.27	5.60	**1.12**	1.20
ccit01	0.21	0.07	0.02	**0.02**
cmpndd	1.83	3.04	2.44	**1.14**
sea_dusk	0.12	0.21	0.05	**0.04**
sunset	2.42	2.18	1.96	**1.68**

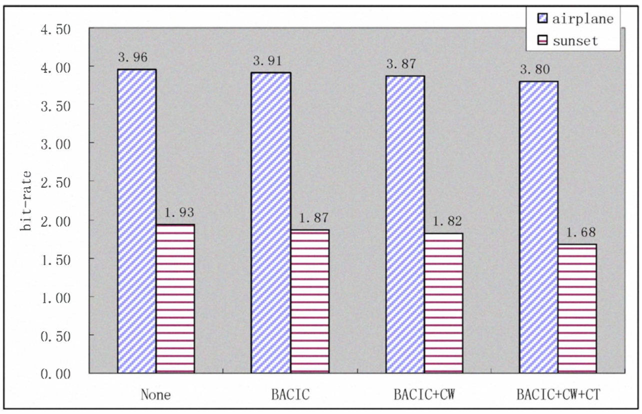

Figure 9. Significance of every component in the algorithm. Forgetting factor (BACIC), context weighting (CW) and context tree modelling (CT) are tested for compression image "airplane" and "sunset".

REFERENCES

[1] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Trans. Image Processing,* 9(8), 1309–1324, 2000.

[2] X. Wu, N. Memon, "Context-based, adaptive, lossless image coding", *IEEE Trans. on communications*, 45(4), 437-444, 1997.

[3] A. Podlasov, P. Fränti, "Lossless image compression via bit-plane separation and multi-layer context tree modeling", *Journal of Electronic Imaging*, 14 (5), 043009, October-December 2006.

[4] H. Kikuchi, K. Funahashi, and S. Muramatsu, "Simple bit-plane coding for lossless image compression and extended functionalities", *Picture Coding Symposium (PCS'09)*, Chicago, USA, May 2009.

[5] Coded representation of picture and audio information-progressive bi-level image compression ISO/IEC JTC1 CD 11544, 1993.

[6] B. Martins and S. Forchhammer, "Bi-level image compression with tree coding," *IEEE Trans. Image Process.*, vol. 7, no. 4, 517-528, Apr. 1998.

[7] M. Mrak, D. Marpe and T. Wiegand, "A context modeling algorithm and its application in video compression", *Proc. of IEEE Int. Conf. on Image Processing,* vol.3, 845-848, 2003.

[8] H. Xiao and C. G. Boncelet, "On the Use of Context-Weighting in Lossless Bilevel Image Compression", *IEEE Trans. Image Processing,* 15(11), 3253 – 3260, 2006.

[9] M. D. Reavy and C. G. Boncelet, "BACIC: a new method for lossless bi-level and grayscale image compression", *IEEE Int. Conf. on Image Processing,* vol.2, 282-285, 1997.

[10] P. Fränti, E. Ageenko, P. Kopylov, S. Gröhn and F. Berger, "Compression of map images for real-time applications", *Image and Vision Computing*, 22 (13), 1105-1115, November 2004.

[11] P. Kopylov and P. Fränti, "Compression of map images by multilayer context tree modeling", *IEEE Trans. on Image Processing*, 14 (1), 1-11, January 2005.

[12] A. Akimov, A. Kolesnikov and P. Fränti, "Lossless compression of color map images by context tree modeling", *IEEE Trans. on Image Processing*, 16 (1), 114-120, January 2007.

[13] Ausbeck, P.J., Jr. "The piecewise-constant image model", *Proceedings of the IEEE*, 88 (11), 1779-1789, Nov 2000.

[14] A. Podlasov and P. Fränti, "Merge-based color quantization and context tree modeling for compression of color quantized images", *IEEE Int. Conf. on Image Processing* (ICIP'06), Atlanta, Georgia, USA, 2277-2280, October 2006.

[15] A.J. Pinho and A.J.R. Neves. A context adaptation model for the compression of images with a reduced number of colors. *IEEE Int. Conf. on Image Processing*, (ICIP'05), Genova, Italy, September 2005.

[16] X. Chen, S. Kwong, and J.-F. Feng, "A new compression scheme for color-quantized images", *IEEE Trans. on Circuits and Systems for Video Technology*, 12 (10), 904-908, Oct. 2002.

[17] P. Fränti, E. Ageenko, P. Kopylov, S. Gröhn and F. Berger, "Compression of map images for real-time applications", *Image and Vision Computing*, 22 (13), 1105-1115, November 2004.

[18] P. Kopylov and P. Fränti, "Compression of map images by multilayer context tree modeling", *IEEE Trans. on Image Processing*, 14 (1), 1-11, January 2005.