Lossless Color Image Coding Based on Probability Model Optimization Utilizing Example Search and Adaptive Prediction

Kyohei Unno (member)†,‡, Yusuke Kameda (member)†‡, Ichiro Matsuda (member)†‡, Susumu Itoh (honorary member)†‡, Sei Naito (member)‡

Abstract We previously proposed a novel lossless coding method that utilizes example search and adaptive prediction within a framework of probability model optimization for gray-scale images. In this paper, we extend the method for RGB 4:4:4 formatted color images. In the proposed method, multiple examples are collected from the causal area in not only the same color signal to be encoded but also other color signals as far as they have already been encoded. Moreover, multiple affine predictors trained on a pel-by-pel basis are also utilized to exploit intra- and inter-color correlations. The probability distribution of the color signal at each pel is dynamically modeled by using both examples and predictors. Then a few parameters used in the probability model are numerically optimized for efficient entropy coding. The experimental results show that the proposed method achieves better coding performance than other state-of-the-art lossless coding methods.

Key words: Lossless coding, Example search, Affine prediction, Probability modeling, Quasi-Newton method

1. Introduction

Lossless coding is useful for specific applications where degradation in image quality is unacceptable, such as post-production of professional photography and digital archiving for cultural property. In general, the most important factor for evaluating efficiency of the lossless image coding is bit-rate savings, or coding rates, because image quality, which could be another factor in the lossy coding case, is perfectly kept. Efficient coding techniques in this sense typically have two processing stages: a de-correlation stage and an entropy coding stage. In this framework, de-correlated signals generated by the first stage, i.e. prediction residuals or transform coefficients, are expected to follow a symmetric single-peaked distribution centered around zero. For the lossless image coding, the least-square method [1,2], or weighted least-square method [2–4] is often used to design adaptive predictors in the first stage. In either case, the predictor design is carried out independently of the succeeding entropy coding stage. On the other hand, in our past study, an entropy coding algorithm was considered in the de-correlation stage and multiple predictors were iteratively optimized to minimize the resulting coding rates [5]. Nevertheless, the prediction residuals were always modeled by the single-peaked functions in the entropy coding stage.

To handle rather complicated probability distributions, a multi-peaked probability model whose peak positions were given by multiple linear predictors was employed in [6]. In addition, some methods are being studied to directly estimate the probability distribution by using deep neural networks [7,8]. Along with this approach, we recently developed a new lossless coding method for gray-scale images [9]. In this method, the probability distribution of the image intensity value at each pel is approximated by a Gaussian mixture model (GMM) without the de-correlation stage. The peak positions of the respective Gaussian distributions are estimated by examples that are collected from the already encoded area of the same image by template matching. Additionally, multiple affine predictors locally trained in the causal area are also used for estimating the peak positions in [10]. The shape of the probability distribution is controlled by parametric functions associated with the reliability of the corresponding Gaussian distributions. Furthermore, some model parameters used in the functions are numerically optimized by the quasi-Newton method to minimize the resulting coding rate.

In this paper, we extend our previous method to enable efficient coding of RGB 4:4:4 formatted color im-
ages. In the proposed method, the search window for the template matching is expanded to include other color signals as far as they have already been encoded. Likewise, the multiple affine predictors are modified to exploit intra- and inter-color correlations simultaneously.

2. Proposed Method

2.1 Overview

In this paper, it is assumed that three 8-bit color signals stored in the given RGB 4:4:4 formatted color image are encoded one-by-one in the order of R→G→B.

Figure 1 illustrates a block diagram of the proposed encoding process. A color signal \( f(p_k) \) at a target pel \( p_k \) is directly encoded by a multi-level arithmetic coding technique [11] under the assumption of a probability distribution \( \text{Pr}(f) \) estimated by the GMM. To be more specific, the probability distribution is modeled as a linear combination of \( M \) and \( N \) Gaussian functions, whose peak positions \( \{ f_{k,m} \} \) and their reliability metrics \( \{ d_{k,m} \} \) are calculated by processes of example search (for \( m = 1, \ldots, M \) ) and adaptive prediction (for \( m = M + 1, \ldots, M + N \) ), respectively. The shape of each Gaussian function is determined by parametric functions of the reliability metric \( d_{k,m} \) and a locally calculated feature value \( u_{k,m} \). A few model parameters used in definitions of the parametric functions are numerically optimized to minimize the bitrate for every region of \( 64 \times 64 \) pels. The respective processing stages are described below.

2.2 Example Search

The proposed method performs template matching to collect \( M \) kinds of ‘examples’ that have similar local textures to the target pel to be encoded. The search window for the template matching is limited within a distance of \( S = 80 \) pels from the target pel and assigned not only to the same color signal as the target pel but also other color signals as far as they have already been encoded. Figure 2 illustrates this example search process for the thirdly encoded B signal. In this figure, \( p_k \in \mathbb{Z}^2 \) indicates the target pel that is processed \( k \)-th in the raster scan order for the color signal \( c_0 = (B) \). A template consists of 12 pels \( \{ p_k + r_i \mid i = 1, 2, \ldots, 12 \} \) located within \( \| r_i \| \leq 3 \), where \( r_i \in \mathbb{Z}^2 \) denotes the relative position from \( p_k \). The cost function used for the example search is defined by:

\[
J_k(c, q) = \left[ \sum_{i=1}^{12} w_{ik} \cdot \left( f_c(q) - f_{c_0}(p_k + r_i) + \mu_f(q) \right) \right]^2 + \lambda_d \cdot \| q - p_k \|_1, \tag{1}
\]

\[
\mu_f(q) = \sum_{i=1}^{12} w_{ik} \cdot f_c(q + r_i), \tag{2}
\]

\[
w_{ik} = \frac{\exp(-\frac{1}{2} \| r_i \|^2 / \sigma_t^2)}{\sum_{i=1}^{12} \exp(-\frac{1}{2} \| r_i \|^2 / \sigma_t^2)}. \tag{3}
\]

The first term on the right-hand side of Eq.(1) means the similarity of textures measured by weighted root mean squared differences. \( f_c(q) \) indicates the intensity of the color signal \( c \) at a pel \( q \). For the firstly encoded R signal, \( c \) is always the same color as \( c_0 = R \) and the example search is performed in the same way as the grayscale image coding [10]. However, when signals \( c_0 = G \) and \( c_0 = B \) are being encoded, the example search is performed across multiple color signals, i.e. \( c \in \{ R, G \} \) and \( c \in \{ R, G, B \} \), respectively. \( \mu_f(q) \) is a weighted mean of the color signal \( c \) in the causal neighborhood of the pel \( q \), and \( w_{ik} \) is a weighting factor defined by the Gaussian function with \( \sigma_t = 1.25 \). The second term of Eq.(1) imposes a penalty for spatially distant examples from the target pel. A weighting factor of \( \lambda_d = 0.03 \) is used in this paper based on the experiment reported in [9].

As a result of the above example search, \( M \) kinds of examples \( \{ (c_{k,1}, q_{k,1}), (c_{k,2}, q_{k,2}), \ldots, (c_{k,M}, q_{k,M}) \} \) are collected in ascending order of the matching cost. From the color signals specified by these examples, we can ob-
tain estimations of the unknown value $f(p_k)$ by compensating for their local mean value.

$$f_{k,m} = f_{c_{k,m}}(q_{k,m}) - \mu_{c_{k,m}}(q_{k,m}) + \mu_c(p_k) \quad (m = 1, 2, \ldots, M). \quad \text{(4)}$$

These estimations are used as peak positions of the Gaussian functions in the probability modeling described in 2.4. Moreover, the reliability metrics of the respective examples are defined by the matching cost.

$$d_{k,m} = J_k(c_{k,m}, q_{k,m}) \quad (m = 1, 2, \ldots, M). \quad \text{(5)}$$

### 2.3 Adaptive Prediction

The proposed method utilizes pel-wise adaptive affine prediction for capturing local information including inter-color correlations. Namely, predicted values $\hat{f}_{co}(p_k, n)$ ($n = 1, \ldots, N$) are used in addition to $f_{k,m}$ ($m = 1, \ldots, M$) for the probability modeling. These predicted values are calculated by $N$ different affine predictors. Each predictor uses multiple color signals as far as they have already been encoded. In the case of the thirdly encoded B signal, the $n$-th predicted value at the target pel $p_k$ is derived as:

$$\hat{f}_{co}(p_k, n) = b_{n,0} + \sum_{l=1}^{L_n} b_{n,l}f_{co}(p_k + v_{n,l})$$

$$+ \sum_{l=1}^{L_n'} b_{n,L_n+l}f_{co}(p_k + v'_{n,l})$$

$$+ \sum_{l=1}^{L_n'} b_{n,L_n+l'+L_n'}f_{co}(p_k + v'_{n,l'}), \quad \text{(6)}$$

where $b_{n,l}$ ($l = 0, 1, \ldots, L_n + 2L_n'$) are prediction coefficients including a bias term $b_{n,0}$. $L_n$ and $L_n'$ are the numbers of reference pels used for intra-color ($c_0=B$) and inter-color ($c_1=G$, $c_2=R$) signals, respectively. The positions of these reference pels are designated by two sets of vectors $\left\{v_{n,l} \mid l = 1, 2, \ldots, L_n\right\}$ and $\left\{v'_{n,l} \mid l = 1, 2, \ldots, L_n'\right\}$. When R and G signals are being encoded, the assignments of the color signals are $(c_0,c_1,c_2) = (R,-,-)$ and $(c_0,c_1,c_2) = (G,R,-)$, respectively, and the third and fourth terms on the right-hand side of Eq.(6) are ignored if the corresponding color signals are denoted by ‘-’. $N$ sets of the prediction coefficients are trained pel-by-pel in a weighted least-squares sense.

$$(b_{n,0}, \ldots, b_{n,L_n+2L_n'}) = \arg\min_{\{b_{n,l}\}} \left[\sum_{p_k \in T} \frac{1}{\sigma_{n,i}^2} \left( f_{co}(p_i) - \hat{f}_{co}(p_i, n) \right)^2 \right], \quad \text{(7)}$$

where $T$ is a training region in the causal area and its size is determined by the maximum distance $D$ from the target pel $p_k$ as shown in Fig. 3. In this paper, we set $D = 10$ based on the results of our preliminary experiments. The weight for each position in the training region $T$ is given by a reciprocal of the mean squared differences between reference pels of $p_k$ and $p_i$. 

$$\sigma_{n,i}^2 = \frac{1}{L_n+2L_n'} \sum_{l=1}^{L_n} \left( f_{co}(p_k + v_{n,l}) - f_{co}(p_i + v_{n,l}) \right)^2$$

$$+ \sum_{c \in \{c_1,c_2\}} \sum_{l=1}^{L_n'} \left( f_{co}(p_k + v'_{n,l}) - f_{co}(p_i + v'_{n,l}) \right)^2. \quad \text{(8)}$$

In practice, the minimum value of $\sigma_{n,i}^2$ is clipped by 1/64 to avoid zero division in Eq.(7).

The predicted values obtained in this way are concatenated to a series of $\{f_{k,m}\}$ as follows.

$$f_{k,M+n} = \hat{f}_{co}(p_k, n) \quad (n = 1, \ldots, N) \quad \text{(9)}$$

Moreover, the reliability metrics $\{d_{k,m}\}$ for these predicted values are given by regression errors in the training region.

$$d_{k,M+n} = \left[ \sum_{p_i \in T} \frac{\epsilon_n^2}{\sigma_{n,i}} \right]^{\frac{1}{2}}, \quad \text{(10)}$$

where $\epsilon_n^2$ means the minimized values of the objective function shown between brackets in Eq.(7).

### 2.4 Optimization of Probability Models

By performing the above processes of the example search and the adaptive prediction, we obtain $M+N$ kinds of estimations $F_k = \{f_{k,1}, f_{k,2}, \ldots, f_{k,M+N}\}$ and their reliability metrics $D_k = \{d_{k,1}, d_{k,2}, \ldots, d_{k,M+N}\}$. Such information can be considered as prior knowledge of the color signal $f_{co}(p_i)$. Therefore, conditional probabilities of possible values of the color signal are modeled by a linear combination of Gaussian functions as follows:
Thus, the number of coding bits required for entropy
probability being zero. Moreover, the parameters are defined as
parameters that control the height and width of each Gaussian function. Therefore, they are defined as
parameters that reflect the reliability of the corresponding Gaussian function. Therefore, they are defined as
parametric functions of the reliability metric $d_{k,m}$ as well as $u_{k,m}$.

$$h_{k,m} = \exp(-a_1 \cdot d_{k,m}),$$

$$w_{k,m} = a_0 \cdot \exp(-a_2 \cdot d_{k,m}) \cdot \exp(-a_3 \cdot u_{k,m}).$$

According to this model, the probability of the actual color signal $f_{c \circ}(p_k)$ can be calculated by normalizing Eq. (11) for all the possible values:

$$P(f_{c \circ}(p_k) \mid F_k, D_k, u_k) = \frac{P(f_{c \circ}(p_k) \mid F_k, D_k, u_k)}{\sum_{f=0}^{255} P(f \mid F_k, D_k, u_k)}$$

This estimation can be seen as a local fitness of the probability model and is used for the definition of the feature values to feedback the model accuracy. In the case of the examples, the feature values are irrelevant to $m$ as:

$$u_{k,m} = \sum_{i=1}^{12} w_i \cdot l_{c \circ}(p_k + v_i) \quad (m = 1, 2, \ldots, M),$$

while those values for the adaptive prediction depend on the arrangement of the reference pels:

$$u_{k,M+n} = \frac{1}{L_n} \sum_{i=1}^{L_n} l_{c \circ}(p_k + v_n) \quad (n = 1, 2, \ldots, N).$$

In the proposed method, the shape of the probability distribution model is controllable by four parameters $a_0, a_1, a_2, a_3$. These model parameters are numerically optimized by the quasi-Newton method [12] to minimize the number of coded bits in each region $\Omega$ of $64 \times 64$ pixels and then encoded as side-information [9].

### 2.5 Histogram Packing

The above-mentioned probability modeling assumes that the probability distributions are smooth and never fall to zero at all the possible intensity values. However, this is not the case when images are computer-generated or affected by image processing (e.g., contrast enhancement). With respect to such images, the efficiency of lossless coding can be improved by exploiting histogram sparseness. For this purpose, a technique called histogram sparsification, which presumes non-occurring intensity values in a backward adaptive manner, was utilized in [2]. In the proposed method, another approach known as histogram packing [13] is provided as an optional pre-processing tool. When this option is enabled, we pick up all the image intensity values that actually appear in the given image and map them into successive values before the encoding process. In order to recover the original intensity values at the decoder side, a one-bit flag indicates whether the value appears in the given image or not. In the given image is encoded for every possible value as side-information ($256 \times 3$ bits in total).

### 3. Implementation Details

#### 3.1 The Number of Examples ($M$)

In this paper, we optimize the number of examples used for the probability modeling in each region $\Omega$ within a range of $0$ to $M_{\text{max}}$. This optimization is carried out in the following manner.

First, $M$ is initialized to a value of $M_{\text{max}}$. In this paper, we set $M_{\text{max}}$ to 31. After updating the model parameters $a_0, a_1, a_2, a_3$ using the quasi-Newton method, the probability modeling is repeatedly performed by temporarily restricting the number of examples to $M' \in \{0, \ldots, M\}$. Under this restriction, the total amount of coded bits is calculated in the region $\Omega$. 

$$u_{k,M+n} = \frac{1}{L_n} \sum_{i=1}^{L_n} l_{c \circ}(p_k + v_n) \quad (n = 1, 2, \ldots, N).$$

In the proposed method, the shape of the probability distribution model is controllable by four parameters $a_0, a_1, a_2, a_3$. These model parameters are numerically optimized by the quasi-Newton method [12] to minimize the number of coded bits in each region $\Omega$ of $64 \times 64$ pixels and then encoded as side-information [9].

In the proposed method, the shape of the probability distribution model is controllable by four parameters $a_0, a_1, a_2, a_3$. These model parameters are numerically optimized by the quasi-Newton method [12] to minimize the number of coded bits in each region $\Omega$ of $64 \times 64$ pixels and then encoded as side-information [9].

In the proposed method, the shape of the probability distribution model is controllable by four parameters $a_0, a_1, a_2, a_3$. These model parameters are numerically optimized by the quasi-Newton method [12] to minimize the number of coded bits in each region $\Omega$ of $64 \times 64$ pixels and then encoded as side-information [9].

In the proposed method, the shape of the probability distribution model is controllable by four parameters $a_0, a_1, a_2, a_3$. These model parameters are numerically optimized by the quasi-Newton method [12] to minimize the number of coded bits in each region $\Omega$ of $64 \times 64$ pixels and then encoded as side-information [9].
\[ \mathcal{L}_{c_0}(\Omega, M') = - \sum_{p_k \in \Omega} \log_2 \Pr_{M'}(f_{c_0}(p_k) \mid F_k, D_k, u_k), \]  

(19)

where \( \Pr_{M'}(f_{c_0}(p_k) \mid F_k, D_k, u_k) \) is a conditional probability of \( f_{c_0}(p_k) \) calculated by restricting the number of examples to \( M' \) in Eq.(11). Calculation of Eq.(19) for successive values of \( M' \) can be effectively performed by adding the Gaussian function \( g_{k,m}(p_k) \) one by one in Eq.(11). Thus we can obtain the optimum value of \( M \) that minimizes the total coding bits.

\[ M = \arg\min_{M'} \mathcal{L}_{c_0}(\Omega, M'). \]  

(20)

The final value of \( M \) is sent to the decoder side with a fixed-length code of 5 bits as side-information.

### 3.2 Training of the Affine Predictors

Regarding the number of affine predictors trained for each pel, we use a fixed setting of \( N = 25 \) in this paper. The prediction order as well as the arrangement of reference pels used in the respective predictors are determined in an experimental approach. Two ellipses with different sizes given by Eqs.(21) and (22), where \( p_k \) corresponds to their origin, are used for this purpose.

\[ O(x, y) = \beta \cdot \{(\gamma \cdot t_1)^2 + (t_2)^2\} \leq 1^2, \]  

(21)

\[ O'(x, y) = \beta \cdot \{(\gamma \cdot t_1)^2 + (t_2)^2\} \leq (1/2)^2, \]  

(22)

The shapes of both ellipses are determined by three parameters \( \beta, \gamma \) and \( \theta \). Integer positions \( (x, y) \) within the large and small ellipses are picked up and used as two sets of vectors \( \{v_{n,1}\} \) and \( \{v_{n,2}\} \), respectively. In the case of Eq.(21), positions in a non-causal area are discarded from \( \{v_{n,1}\} \) as their color signals are unavailable for intra-color prediction. As mentioned above, we set the area of reference pels for inter-color prediction smaller than that for intra-color prediction, because spatial correlations among different colors tend to be smaller than those within the same color. Currently, \( N \) combinations of the parameters \( \beta, \gamma \) and \( \theta \) are determined in a rather heuristic way. Figure 5 shows the arrangement of reference pels for \( N = 25 \) types of predictors as a result of our parameter choice. We will explore a more sophisticated method of parameter choice with a variable setting of \( N \) in the future.

### 3.3 Processing Procedures

The processing procedures at the encoder side are summarized as follows.

(a) Histogram packing is applied to R, G and B signals as described in 2.5 (optional).

(b) The process of the example search described in 2.2 is carried out, then \( M = M_{\text{max}} \) values of \( f_{k,m} \) and \( d_{k,m} (m = 1, 2, \ldots, M) \) are stored in \( F_k \) and \( D_k \), respectively.

(c) \( N = 25 \) types of affine predictors are trained as described in 3.1, then \( N \) values of \( f_{k,m} \) and \( d_{k,m} \) \( (m = M + 1, \ldots, M + N) \) are added to \( F_k \) and \( D_k \), respectively.

(d) A set of local feature values \( u_k \) is calculated. Thereupon, the probability model defined by Eq.(11) is constructed.

(e) The above processes of (b)–(d) are performed for each pel \( p_k \) in raster scan order.

(f) When the process of (e) has been completed for 64 lines, a slice of those lines is horizontally divided to form regions composed of 64 × 64 pels. For every obtained region \( \Omega \), the following optimization procedures are conducted.

(i) The model parameters \( a_0, \ldots, a_3 \) are updated according to the iterative step of quasi-Newton method [12]. This step is repeated up to 10 times.

(ii) The sets of feature values \( \{u_k\} \) are updated.

(iii) The number of examples \( M \) is updated as described in 3.1. After that, unnecessary elements of \( F_k, D_k \) and \( u_k \) are removed for each \( p_k \in \Omega \)

(iv) The processes (i)–(iii) are iterated up to 5 times.

(v) The optimized model parameters \( a_0, \ldots, a_3 \) are uniformly quantized with 8-bit accuracy, and encoded together with \( M \) as side-
(g) The actual value of \( f(p_k) \) is encoded by arithmetic coding based on the probabilities calculated using the quantized model parameters.

(h) All the above procedures are conducted for the whole color signal in the order of R→G→B.

At the decoder side, the model parameters \( a_0, \ldots, a_3 \) and the number of examples \( M \) are decoded for every region \( \Omega \) as the first step. Then the processes of the example search and the adaptive prediction are performed as in (b) and (c) to obtain the prior knowledge of \( F_k \) and \( D_k \). Moreover, the set of feature values \( u_k \) is calculated by averaging the estimated coding bits in a causal neighborhood. Thus we can calculate the conditional probabilities of \( \Pr(f | F_k, D_k, u_k) \) \((f = 0, 1, \ldots, 255)\), and these probabilities are used for decoding the value of \( f_{\text{bin}}(p_k) \) at each pel \( p_k \). If the option of the histogram packing is enabled, the original color signals are finally recovered using the one-bit flags given as side-information.

### 4. Experimental Results

#### 4.1 Test Images

To evaluate the effectiveness of the proposed method, we conducted coding simulation for the RGB 4:4:4 formatted natural color images shown in Fig. 6. The proposed method is fully implemented in C++.

#### 4.2 Comparison of Proposed Tools

Firstly, we experimented to verify the effects of some coding tools used in the proposed method. Table 1 shows the experimental results in the case some coding tools in the proposed method are turned on/off. Here, “Base” means the base method [10] where three color signals are encoded independently with adaptive \( \Omega \) \((M_{\text{max}} = 31)\) and fixed \( N = 25 \). “+CC Ex” means inter-color correlations are utilized in the example search as an extension to “Base”. In other words, examples can be collected from other color signals as far as they have already been encoded. “+CC Pred” indicates the inter-color correlations are further utilized in the adaptive prediction. Finally, “+Hist” means the option of histogram packing is enabled in the “+CC Pred” method.

In comparison with “Base”, the coding performances of “+CC Ex” and “+CC Pred” are improved for all test images owing to utilization of inter-color correlations in the example search and the adaptive prediction, respectively. Those gains are relatively high in the case of “+CC Pred”. In the example search, calculations of both the matching cost in Eq.(1) and the estimation \( f_{k,m} \) in Eq.(4) compensate for local mean values but do nothing for different amplitudes in waveforms between color signals. This could be the reason for the rather small gains in “+CC Ex” and suggests room for further improvement in the example search.

With respect to “+Hist”, significant gains can be seen for specific images named “Couple” and “Girl”. These images have sparse histograms and the tool of histogram packing is very beneficial to such images. In some cases, coding rates are slightly increased due to the overhead of the side-information \((256 \times 3 \text{ bits/image})\). In terms of the average coding rate, however, “+Hist” achieves the best performance. Therefore, we consider “+Hist” as our proposed method in the following experiments.

#### 4.3 Comparison with other methods

The coding rates of the proposed method are reported with those of the other state-of-the-art methods in Table 2. The compared methods are: MRP [5], Vanile WLS D (version 1.0) [2], BMF [14], GraLIC [15], JPEG-LS [16], and JPEG 2000 [17]. In this table, the best cod-

| Image | Size | Base | +CC Ex | +CC Pred | +Hist |
|-------|------|------|--------|----------|-------|
| Couple | 256×256 | 10.878 | 10.759 | 10.173 | 9.272 |
| Earth | 512×512 | 11.098 | 10.944 | 10.107 | 10.108 |
| Girl | 10.707 | 10.459 | 8.654 | 8.657 |
| House | 10.487 | 10.471 | 10.185 | 10.182 |
| Parrots | | | | | |
| Airplane | 11.008 | 10.944 | 10.107 | 10.108 |
| Lena | 11.247 | 12.401 | 11.780 | 11.773 |
| Mandrill | 17.416 | 17.385 | 16.292 | 16.265 |
| Milkdrop | 9.758 | 9.744 | 9.255 | 9.257 |
| Peppers | 10.707 | 10.459 | 8.654 | 8.657 |
| Sailboat | 14.580 | 14.536 | 13.961 | 13.890 |
| Tiffany | 10.487 | 10.471 | 10.185 | 10.182 |

Table 1: Comparison of coding rates (bits/pel).

![Fig. 6 Test images.](image-url)
Table 2  Comparison of coding rates with state-of-the-art methods (bits/pel).

| Image   | Size  | Proposed | MRP  | Vanilc | BMF  | GraLIC | JPEG-LS | JPEG 2000 |
|---------|-------|----------|------|--------|------|--------|---------|-----------|
| Couple  | 256×256 | 9.272    | 10.476 | 9.653  | 10.385 | 9.725  | 11.991  | 12.113    |
| Earth   |       | 9.367    | 9.452 | 9.407  | 9.263 | 9.717  | 12.648  | 12.129    |
| Girl    |       | 10.054   | 12.183 | 10.255 | 12.106 | 10.444 | 13.488  | 12.702    |
| House   |       | 10.802   | 11.034 | 10.990 | 11.045 | 11.383 | 12.558  | 12.702    |
| Parrots |       | 8.259    | 8.415 | 8.656  | 8.093 | 8.333  | 11.646  | 9.391     |
| Airplane| 512×512 | 10.108   | 10.124 | 10.161 | 10.074 | 10.354 | 11.889  | 11.669    |
| Lena    |       | 11.773   | 11.874 | 11.851 | 11.892 | 12.249 | 13.605  | 13.597    |
| Mandril|       | 16.265   | 16.040 | 16.299 | 16.240 | 16.715 | 18.516  | 18.092    |
| Milkdrop| 512×512 | 9.257    | 9.268 | 9.335  | 9.272 | 9.443  | 10.704  | 11.917    |
| Peppers |       | 8.657    | 8.417 | 8.922  | 8.490 | 9.293  | 11.751  | 10.246    |
| Sailboat|       | 13.890   | 13.675 | 14.090 | 13.839 | 14.462 | 15.714  | 16.060    |
| Tiffany |       | 10.182   | 10.334 | 10.533 | 10.450 | 10.588 | 11.601  | 13.860    |
| Average |       | 9.551    | 10.312 | 9.792  | 10.179 | 9.921  | 12.466  | 11.993    |
| All     |       | 11.448   | 11.390 | 11.599 | 11.465 | 11.872 | 13.397  | 13.634    |

The coding rate is shown in **bold** and the second-best coding rate is **underlined** for each image. The proposed method achieves the lowest or the second-lowest coding rates for the majority of the test images. In the case of test images composed of 256×256 pels, the proposed method achieves the lowest and the second-lowest coding rates. On the other hand, “MRP” achieves the lowest coding rates for test images of 512×512 size on average. The proposed method and “Vanilc” can be categorized as backward types, while “MRP” is a forward type from the viewpoint of adaptation strategy. In forward type adaptation, most parameters needed for the adaptation are determined only at the encoder side, and then sent to the decoder side. Although such a strategy generally reduces the complexity of the decoder, the coding efficiency tends to depend on the trade-off between the amount of side-information and the degree of freedom in the resulting adaptation. In this scenario, the ratio of side-information needed for the adaptive prediction used in “MRP” is relatively lower in the total amount of coded bits and therefore attains better coding performance for larger images.

5. Conclusion

In this paper, our lossless coding method, which was previously reported for gray-scale images, is extended to RGB 4:4:4 formatted color images. Better coding performance is achieved by three major contributions: First, an example search for making use of non-local information on image textures is performed across multiple color signals. Second, the adaptive affine prediction is modified to exploit intra- and inter-color correlations simultaneously. Moreover, number of examples is adaptively changed within a framework of the probability model optimization. Experimental results show that the proposed method attains the best or the second-best coding efficiency among the compared state-of-the-art lossless coding methods for most of the tested images.

These results imply that the proposed method has a potential to save more storage space than the other state-of-the-art methods for specific applications allowing no quality loss. On the other hand, a main weakness of the proposed method is its complexity. For example, the encoding process for an image with the size of 256×256 pels typically takes 4 hours on a computer with the Intel Core i7-4770K@3.50GHz CPU and 16 GB main memory. About 40% and 56% of the encoding time are spent on the adaptive prediction (in 2.3) and the optimization (in 2.4), respectively. Complexity reduction is our important concern and should be tackled for aiming practical use. Moreover, we think that the proposed method still has room for improving the coding efficiency. Specifically, tuning of the number of predictors (N) as well as the arrangements of their reference pels are promising topics for future studies.

References

1) L.-J. Kau and Y.-P. Lin: “Adaptive Lossless Image Coding Using Least Squares Optimization with Edge-Look-Ahead,” IEEE Trans. on Circuits and Systems II: Express Briefs, Vol.52, No.11, pp.751–755, Nov. 2005.
2) A. Weinlich, P. Amon, A. Hutter and A. Kaup: “Probability Distribution Estimation for Autoregressive Pixel-Predictive Image Coding,” IEEE Trans. on Image Processing, Vol.25, No.3, pp.1382–1395. Mar. 2016.
3) H. Ye, G. Deng and J. C. Devlin: “A Weighted Least Squares Method for Adaptive Prediction in Lossless Image Compression,” Proc. 2003 Picture Coding Symposium (PCS 2003), pp. 489-493, Apr. 2003.

4) B. Meyer and P. Tischer: “Glicbawls — Grey Level Image Compression by Adaptive Weighted Least Squares,” Proc. 2001 Data Compression Conf. (DCC 2001), pp. 503, Mar. 2001. IEEE Trans. on Image Processing, Vol. 25, No. 3, pp. 1382-1395, Mar. 2016.

5) I. Matsuda, T. Kaneko, A. Minezawa and S. Itoh: “Lossless Coding of Color Images Using Block-Adaptive Inter-color Prediction,” Proc. 2007 IEEE International Conference on Image Processing (ICIP 2007), pp. II-329-II-332, Sep. 2007.

6) B. Meyer and P. Tischer: “TmW — A New Method for Lossless Image Compression,” Proc. 1997 Picture Coding Symposium (PCS ’97), pp. 533–538, Sep. 1997.

7) T. Salimans, A. Karpathy, X. Chen and D. P. Kingma: “PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications,” Proc. 5th International Conference on Learning Representations (ICLR 2017), Apr. 2017.

8) F. Mentzer, E. Agustsson, M. Tscannen, R. Timofte and L. V. Gool: “Practical Full Resolution Learned Lossless Image Compression,” arXiv preprint arXiv:1811.12817, Nov. 2018.

9) I. Matsuda, T. Ishikawa, Y. Kameda and S. Itoh: “A Lossless Image Coding Method Based on Probability Model Optimization,” Proc. 26th European Signal Processing Conf. (EUSIPCO 2018), pp. 156–160, Sep. 2018.

10) K. Unno, Y. Kameda, I. Matsuda, S. Itoh and S. Naito: “Lossless Image Coding Exploiting Local and Non-local Information via Probability Model Optimization,” Proc. 27th European Signal Processing Conf. (EUSIPCO 2019), Sep. 2019.

11) M. Schindler: “A Fast Renormalization for Arithmetic Coding,” Proc. Data Compression Conf. (DCC ’98), p.572, 1998.

12) W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery: “Numerical Recipes: The Art of Scientific Computing,” Third Edition,” Cambridge University Press, 2007.

13) A. J. Pinho: “A Comparison of Methods for Improving the Lossless Compression of Images with Sparse Histograms,” Proc. of International Conference on Image Processing (ICIP 2002), Vol. 2, pp. 673–676, Sep. 2002.

14) http://compression.ru/ds/.

15) http://www.imagecompression.info/gralic/Gralic111d.zip.

16) ISO/IEC, 14495-1:1999: “Information Technology — Lossless and Near-Lossless Compression of Continuous-Tone Still Images: Baseline,” Dec. 1999.

17) ITU-T Rec. T.800 — ISO/IEC 15444-1: “Information Technology — JPEG 2000 Image Coding System – Part 1: Core Coding System,” 2001.

Kyohei Unno received the B.E. and M.E. degrees in electrical engineering from Tokyo University of Science, in 2009 and 2011, respectively. He is currently studying for a Ph.D at Tokyo University of Science. He joined KDDI Research Inc. in 2018. He is currently a research engineer of Ultra-Realistic Communication Laboratory in KDDI Research Inc., Japan.

Susumu Itoh received the B.E., M.E. and Ph.D. degrees in electronic engineering from the University of Tokyo, in 1976, 1978 and 1981, respectively. Since 1981, he has been with the Department of Electrical Engineering, Tokyo University of Science, and currently he is a professor. His research interests include video coding and image processing. He is a past president of the Institute of Image Information and Television Engineers.

Sei Naito received B.E., M.E. and Ph.D degrees from Waseda University in 1994, 1996 and 2006, respectively. He joined Kokusai Densin Denwa Corporation (currently KDDI Corporation) in 1996. He is currently a senior manager of Ultra-Realistic Communication Laboratory in KDDI Research Inc., Japan.

Yusuke Kameda received the B.E., M.E. and Ph.D degrees in Engineering from Chiba University, in 2006, 2008 and 2012, respectively. He is currently a Junior Associate Professor in the Department of Electrical Engineering at Tokyo University of Science. His research interests cover Computer Vision and Video Coding.

Ichiro Matsuda received the B.E., M.E. and Ph.D. degrees in electrical engineering from Tokyo University of Science, in 1991, 1993 and 1996, respectively. Since 2014, he is a professor in the Department of Electrical Engineering, Tokyo University of Science. His main research interests cover image and video coding and processing.