A lossless image compression algorithm using predictive coding based on quantized colors

FUANGFAR PENSIRI, SURAPONG AUWATANAMONGKOL
Department of Computer Science, School of Applied Statistics
National Institute of Development Administration
118, Seri Thai Road, Bangkok 10240
THAILAND
f.far@live.com http://www.nida.ac.th

Abstract: - Predictive coding has proven to be effective for lossless image compression. Predictive coding estimates a pixel color value based on the pixel color values of its neighboring pixels. To enhance the accuracy of the estimation, we propose a new and simple predictive coding that estimates the pixel color value based on the quantized pixel colors of three neighboring pixels. The prediction scheme can help minimize the upper bound of the residual errors from the prediction. Experiments were conducted on a set of true color 24-bit images, whose pixel colors were quantized into 2, 4, 8 and 16 colors. The experimental results show that the proposed algorithm outperforms some well known lossless image compression algorithms such as JPEG-LS and PNG by factors of 2-3 in terms of bits per pixel. The results also show that the proposed algorithm gives the best compression rates when colors were quantized into two colors.

Key-Words: - Image compression, Lossless compression, Lossless image compression, Compression, Predictive coding, Quantized colors

1 Introduction

The major objective of image compression is to reduce or eliminate the data redundancies which may exist when storing an image so that the compressed image size can be minimal. The data redundancies comprise of three basic redundancies: coding redundancy, inter-pixel redundancy, and psycho-visual redundancy [10]. The image compression techniques have been classified into two categories: lossless and lossy compression. Lossless compression reconstructs the image while preserving its original information to the same level as before the compression is operated. For lossy compression, some information may be lost during the processing but the distortion level of the reconstructed image must be acceptable and can not be recognized by human eyes.

This paper is concerned with lossless compression using the predictive coding for RGB color images. Predictive coding is a compression method used for text and image compression. It encodes the difference between the current data estimation derived from past data and actual current data [5] to attain more efficient compression. The degree of efficiency depends very much on the accuracy of the estimation as the difference becomes smaller, the information to be encoded becomes smaller as well. There are several lossless image compression algorithms which have been developed using this predictive coding method [8,9,17]. JPEG-LS is an example of the predictive coding approach which works well on continuous-tone images [1,4]. It is one of the most popular lossless image compression algorithms. Unlike the previous JPEG compression, which is lossy compression using the Discrete Cosine Transform method (DCT) [11], JPEG-LS is not complex and works well with gray scale images. However, its performance is not as impressive when applied to indexed color or color-map images.

One of problems of predictive coding is that it cannot estimate pixel color values very well near edges, boundaries, or when there are sharp transitions of colors [6]. This is because predictive coding relies on the similarities of neighboring pixels for the estimation [7] and, therefore, the dissimilarities of pixel colors near edges or boundaries can adversely effect the accuracy of the estimation of the pixel colors.

To improve the compression efficiency of the predictive coding method, a new estimation method is proposed to give higher estimation accuracy. The new method estimates the pixel color value by the average of color values of the neighboring pixels, i.e. the adjacent pixels in the north, northwest, and west directions, with the same quantized color. If all three neighboring pixels have different quantized colors, the estimated value is simply equal to the average of the pixel color values of all three neighboring pixels. It can be shown later that by using the proposed estimation scheme the estimation errors can be reduced significantly leading to an improvement of compression efficiency.

Experiments were conducted on several 24-bit color images. The colors of each tested image were quantized into 2, 4, 8 and 16 colors using the color quantization technique proposed in [15]. The color quantization technique recursively partitions the color space of a given image into two subspaces along the color axis with the highest variance. From the experimental results, the proposed algorithm can achieve about two to three times better compression rates in bits per pixel than the well known lossless image compression algorithms such as JPEG-LS and PNG.

2 Related Methods

2.1 Predictive Method: JPEG-LS

The main component of the predictive coding method is the "Predictor" which exists in both encoder and decoder. The encoder computes the predicted color value for a pixel, denote $\hat{f}(n)$, based on the known pixel color values of its neighboring pixels. The residual error, which is the difference value between the actual color value of the current pixel $f(n)$ and the predicted one, i.e. $e(n) = f(n) - \hat{f}(n)$, is computed for all pixels. The residual errors are then encoded, usually by an encoding scheme like Huffman encoding, to generate a compressed data stream. The decoder also computes the predicted color value of the current pixel $\hat{f}(n)$ based on the previously decoded color values of neighboring pixels using the same method as the encoder. The decoder decodes the residual error $e(n)$ for the current pixel and performs the inverse operation $f(n) = e(n) + \hat{f}(n)$ to restore the color value of the current pixel [10] .

JPEG-LS is a well-known lossless image compression algorithm developed by the Joint Photographic Experts Group (JPEG), a joint ISO/ITU committee. There are eight different predictive schemes proposed by the JPEG-LS standard, in which a user can try to give the best compression for a given image. The first scheme makes no prediction, i.e. $\hat{I} = 0$. The next three schemes are called one-dimensional predictors. The schemes make predictions based only on the color value of the previous pixel on the current line. The other four schemes are called two dimensional prediction schemes. The schemes make predictions based on the color values of previous pixels in a left-to-right, top-to-bottom scan of an image. The eight prediction schemes of JPEG-LS are as follows [12,18]:

$$\hat{I}(i, j) = 0 \tag{1}$$

$$\hat{I}(i, j) = I(i-1, j) \tag{2}$$

$$\hat{I}(i, j) = I(i, j-1) \tag{3}$$

$$\hat{I}(i, j) = I(i-1, j-1) \tag{4}$$

$$\hat{I}(i, j) = I(i, j-1) + I(i-1, j) + I(i-1, j-1) \tag{5}$$

$$\hat{I}(i, j) = I(i, j-1) + I((i-1, j) + I(i-1, j-1))/2 \tag{6}$$

$$\hat{I}(i, j) = I(i-1, j) + I((i, j-1) + I(i-1, j-1))/2 \tag{7}$$

$$\hat{I}(i, j) = I(i-1, j) + I(i, j-1)/2 \tag{8}$$

Where $\hat{I}(i, j)$ is the predicted color value of the current pixel at the coordinate (i, j). $I(i, j-1)$, $I(i-1, j-1)$ and $I(i-1, j)$ are the color values of the adjacent pixels located on the north, northwest and west direction respectively.

2.2 Predictive Method: PNG

Portable Network Graphics (PNG) is based on a predictive scheme and entropy coding [16]. PNG format supports true colors (16 million colors) and many types of images, e.g. gray scale or indexed, and works well on web browsers. PNG compression operates on an image in two phases. The first phase is the data filtering which converts pixel color values to numbers. The data filtering is performed on a byte basis which means that the given image data is viewed as a stream of bytes. Each pixel color value may be one or two bytes long or several pixel values can be packed into a single byte. This depends on the type of image. The second phase is to encode the

output byte stream from the first phase. PNG offers five filtering methods [4], which are as follows:
Filtering type 0 : no filtering
Filtering type 1 : set byte $B_{i,j}$ to the difference $B_{i,j} - B_{i-t,j}$ (the byte of the north pixel)
Filtering type 2 : set byte $B_{i,j}$ to the difference $B_{i,j} - B_{i,j-1}$ (the byte of the west pixel)
Filtering type 3 : set byte $B_{i,j}$ to the difference $B_{i,j} - (B_{i-t,j} + B_{i,j-1}) / 2$
Filtering type 4 : set byte $B_{i,j}$ to $B_{i,j}$ – PaethPredict $(B_{i-t,j}, B_{i,j-1}, B_{i-t,j-1})$

Where t is the interval between the current byte $B_{i,j}$ and its correlated predecessor e.g. $B_{i-t,j}$ is the byte of neighboring pixel in north and $B_{i-t,j-1}$ is the byte of neighboring pixel in a northwest direction, and PaethPredict is a prediction function that chooses whichever of its three parameters is closest to $B_{i-t,j} + B_{i,j-1} - B_{i-t,j-1}$. The improvement on compression can be achieved by adaptively choosing different filtering types on a row by row basis using a heuristic method.

2.3 Color Quantization

Color quantization is a process of dividing a color space of an image into regions. Each region can then be represented by a representative color, normally the centroid of the region. The process can be used to represent a color image using a number of representative colors which take fewer bits to represent. However, this representation scheme introduces image distortion that need to be minimized. The distortion can be measured by the total quantization error which is the sum of squared distances between actual pixel colors and their color representatives. The distance between a pair of color points can be measured using the Euclidean Distance as follows

$$d(c_m, c_n) = \sqrt{(r_m - r_n)^2 + (g_m - g_n)^2 + (b_m - b_n)^2} \quad (9)$$

Where the $d(c_m, c_n)$ is the Euclidean distance between the two color points c_m and c_n, r_m, g_m and b_m are the color values of c_m on red, green and blue color axes, respectively and r_n, g_n and b_n are the color values of c_n on red, green and blue color axes, respectively.

There had been several proposed color quantization methods [2,3,13,14]. In this study, a fast and effective color quantization method proposed in [15] is to be used. The method is based on recursively partitioning color sub regions using a plane perpendicular to the color axis with the highest color variance. As color points spread widely along this color axis due to the highest variance, partitioning the color sub region into two smaller ones along this axis would reduce the total quantization error more than the other two axes. The process of the color quantization methods can be described step by step as following.

Step 1 : The color points of the image are sorted in ascending order based on the color values for each of the color axes. A sorted list of the color points for each color axis is created.

Step 2 : The axis with highest variance of color values of the current sub region is selected to be the principal axis for partitioning the sub region.

Step 3 : Identify the m^{th} color point in the sorted list of the principal axis to become the partitioning point where the cutting plane will passes through. Fig.1 shows an example that a sub region C (displayed only two dimensions) is partitioned into two smaller sub regions, C_1 and C_2, using the m^{th} color point as the partitioning point. After the partitioning, the total quantization error of the sub region C is the sum of the total quantization errors of C_1 and C_2. Fig.2 shows the relationship between the total quantization error of the sub region C and the position of the partitioning point (line 3), the relationship between the total quantization error of C_1 and the position of the partitioning point (line 1), and the relationship between the total quantization error of C_2 and the position of the partitioning point (line 2). As the position of the partitioning point moves from the first to the n^{th} (last) color point in the sorted list, the total quantization error of C_1 increases from zero to the maximum value while the total quantization error of C_2 decreases from the maximum value to zero. Hence, the total quantization error of the sub region C, which is the sum of the total quantization errors of C_1 and C_2, becomes minimal at the point where the two quantization errors of C_1 and C_2 are equal.

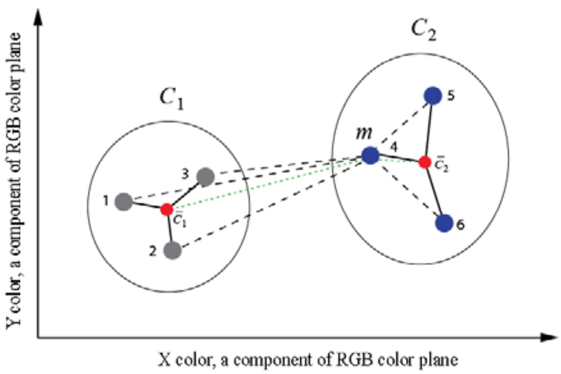


Fig.1 The sub region partitioning using the m^{th} point along the principal axis as the partitioning point

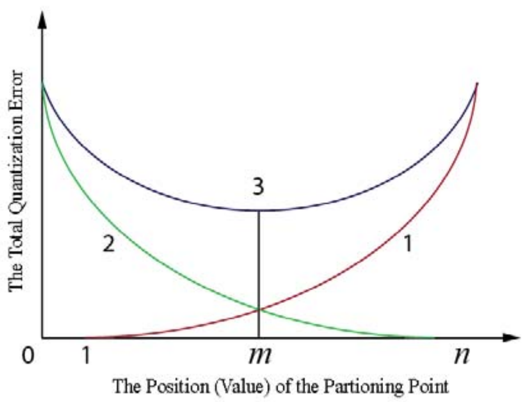


Fig.2 Plots of the total quantization errors of the current sub region C, C_1 and C_2 against the position of the partitioning point m

Let c_j and c_{j+1} be two adjacent color points in the sorted list of the principal axis, and $D_j = d(c_j, c_{j+1})^2$, where $d(c_j, c_{j+1})$ is the Euclidean distance between the two adjacent color points. Let $dsum_i$ be the sum of the squared Euclidean distances, D_1 to D_i as follows:

$$dsum_i = \sum_{j=1}^{i} D_j \qquad (10)$$

It can be shown [15] that the position of the optimal partitioning point that yields the minimal value of the quantization of the sub region can be determined by the following equation.

$$centroidDist = \sum_{i=1}^{i} f_i \bullet dsum_i \,/\, \sum_{i=1}^{n} f_i \qquad (11)$$

Where f_i is the number of pixels whose color corresponds to the i^{th} color point c_i, centrioidDist value is the distance from c_1 (the first point) to the optimal partitioning point, the m^{th} point along the principal axis, see Fig.3.

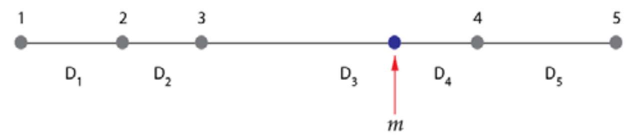


Fig.3 The optimal partitioning point at the m^{th} color point along the principal axis

Step 4 : The current sub region C is partitioned into two smaller sub regions C_1 and C_2 using the partitioning point identified from step 3. Each of the three sorted lists, one for each axis, is also split into two lists according to the partitioning point.

Step 5 : The partitioning process from step 2 to 4 is performed recursively on both C_1 and C_2 until the number of sub regions equals to a specified value.

3 Proposed Algorithm

The encoding part of the proposed lossless image compression comprises of three main tasks. The first task performs color quantization with a given number of regions on the input image using the method discussed in section 2. The outputs from the first task are the centroids of all the regions. The next task calculates the predicted color value of each pixel, one at a time, starting from the leftmost column to the rightmost column and from the top row to the bottom row, based on the quantized colors of their adjacent pixels. The residual error of each pixel is then computed. The last task encodes the residual errors as well as some parameters, such as the number of centroids and the values of the centroids, using the Huffman Coding method.

The decoding part of the proposed algorithm performs the reverse process of the encoding part. It first decodes the Huffman codes then calculates the predicted color value of each pixel based on the quantized colors of its neighboring pixels. Finally, it calculates the original color value of each pixel by adding the predicted value with the decoded error residual of the pixel. The whole process of the proposed method is shown in Fig.4

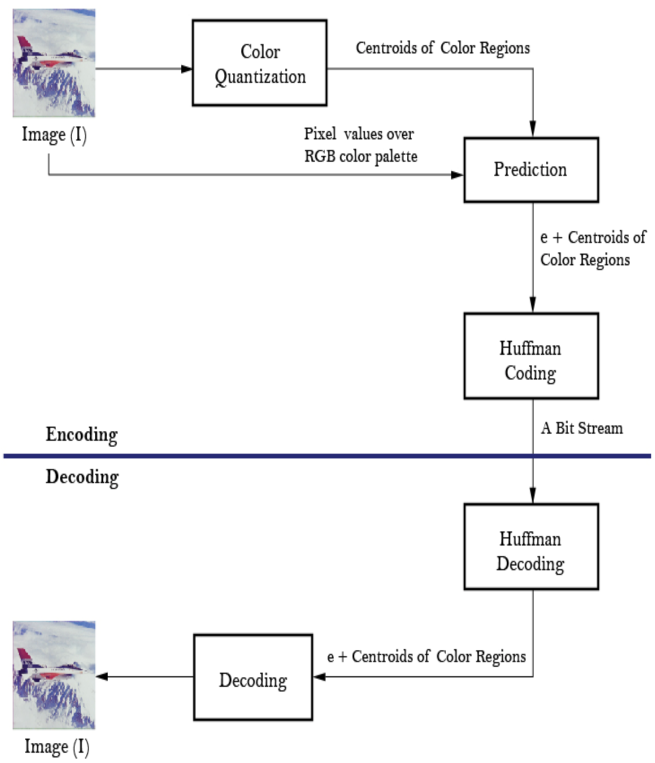


Fig.4 The process of the proposed lossless compression algorithm

The prediction scheme of the proposed algorithm depends on the location of the current pixel (i, j). Consider the three adjacent pixels, N, NW and W, of the current pixel X (see Fig.5). Let X_c, N_c, NW_c and W_c represent the pixel color values of X, N, NW and W respectively. The prediction scheme uses the following rules to estimate the color value of X, X':

1. If the current pixel is the top leftmost one; then there is no prediction since there are no adjacent pixels and no prior information for prediction. So in this case, $X'=0$, otherwise,
2. If the current pixel X is not the top leftmost one but is in the first row, then $X'=W_c$, otherwise,
3. If the current pixel X is not the top leftmost one but is in the first column, then $X'=N_c$, otherwise,
4. In this case, the current pixel X has all three adjacent pixels. If at least two of the three adjacent pixels have their colors located in the same quantized color regions, $X'=$ the average value of the color values of these pixels; otherwise, the three adjacent pixels have their colors located in different quantized color regions, and so $X'=$ the average value of the color values of the three pixels. To find which quantized color region a pixel color belongs to, we can simply search for the region that has the nearest centroid from the pixel color.

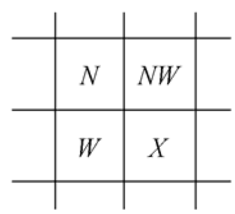


Fig.5 The positions of three adjacent pixels, N, NW and W, of the current pixel X

Since rule 4 is applicable to most of the pixels for an image, the residual errors caused by the prediction under the rule will be discussed. Let d_n be the longest distance between any pair of color points located in any n color regions produced by the color quantization method, d_0 be the longest distance between any pair of color points of the given image, i.e. without color quantization, and α_n be the proportion between d_n and d_0. The α_n must have a value of between 0 and 1. Let consider the case under rule 4 of the prediction scheme that at least two of the three adjacent pixels of X have their pixel colors located in the same color region. Let d be the distance from X_c to the farthest color point of an adjacent pixel whose color is located in the same region. Let f be the frequency or probability that the case happens and X_c is also located in the same region (see Fig. 6). Clearly, X', the centroid of the adjacent pixel colors, must lie within the sphere which center is at X_c and radius is equal to d. Hence, in this case the residual error $|X_c - X'|$ must be less than or equal to d, and so must be less than or equal to d_n. On the other hand, with probability $1-f$, either X_c is not located in the same region as the colors of the two or three adjacent pixels (see Fig.7), or the colors of all three adjacent pixels are located in different color regions. For the former, X' must lie within the sphere which center is at X_c and radius is equal to d. For the latter, X' must lie within the sphere which center is at X_c and radius is equal to the distance from X_c to the farthest color among those of the three adjacent pixels. For either of the two cases, the residual error $|X_c - X'|$ must be less than or equal to d_0. Therefore, the upper bound of the expected value of the residual error E can be computed as follows:

$$|X_c - X'| = E \le fd_n + (1-f)d_0 \tag{12}$$

$$\le f\alpha_n d_0 + d_0 - fd_0 \tag{13}$$

$$\le d_0 + (1-\alpha_n) fd_0 \tag{14}$$

$$\le (1-(1-\alpha_n) f) d_0 \tag{15}$$

We need to minimize the expected value of the residual error E in order to achieve the best compression rate. To minimize E, we will try to minimize its upper bound instead. From the above inequalities, the upper bound can be minimal if the value $(1-\alpha_n) f$ is maximal. The relationship between f and α_n can be described by Fig.8. As n becomes larger, α_n becomes smaller since each region becomes smaller. As each region becomes smaller and α_n decreases from one, the probability f also decreases from one and approaches to zero. From the Figure, we can see that the value $(1-\alpha_n) f$ corresponds to the size of the shaded rectangle area. As α_n decreases from one, the size of the rectangle increases until α_n reaches the optimal value where the rectangle becomes its largest. If α_n continues to decrease, the size of the rectangle then decreases. Therefore, there is the optimal number of quantization colors that yields the best compression rate for a particular given image.

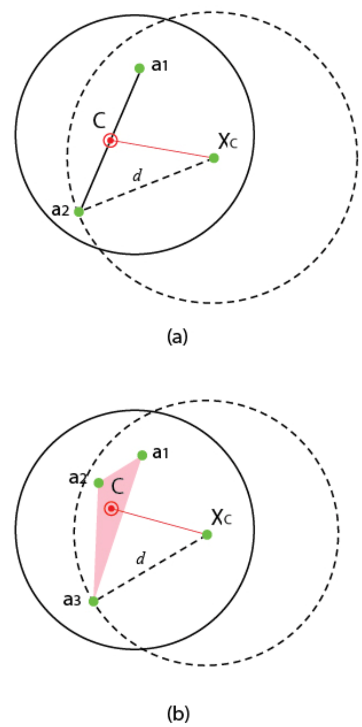


Fig.6 a) A scenario of two of the three adjacent pixels of X as well as X have their colors located in the same color region (a circle with a solid line) b) A scenario of the three adjacent pixels of X as well as X have their colors located in the same color region (a circle with a solid line)

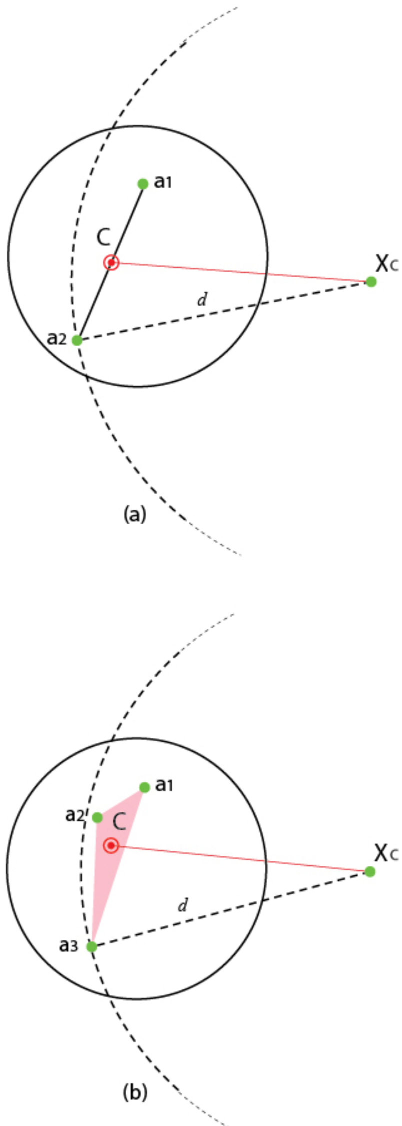


Fig.7 a) A scenario of two of the three adjacent pixels of X but not X have their colors located in the same color region (a circle with a solid line) b) A scenario of the three adjacent pixels of X but not X have their colors located in the same color region (a circle with a solid line).

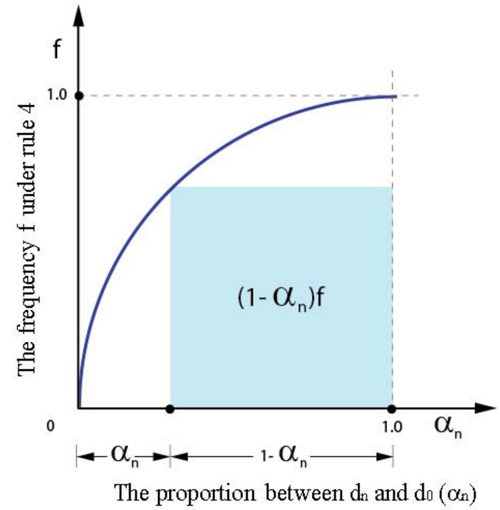


Fig.8 Relationship between α_n and f

4 Experiment Results

We performed experiments on the proposed method on fifteen images shown in Fig.9. Colors of each image were quantized into 2, 4, 8 and 16 regions. Table 1 shows the general characteristics of the fifteen images. Table 2 displays the compression rates of the proposed algorithm achieved from the experiments in bit rate per one color plane. They are compared with those of JPEG-LS and PNG algorithms. It can be seen from the results that the proposed algorithm outperforms the other two algorithms for all test images by factors between 2 to 3 in terms of the bit rate. In particular, the proposed algorithm performs much better than the other two on large images containing large areas with monotone colors, such as boat, girl, goldhill and monarch.

Table 3 shows the numbers of pixels which two or three adjacent pixels have their colors located in the same color region for the fifteen test images. From the table, it can be seen that most pixels in the test images have at least two adjacent pixels with their colors located in the same color region. It can also be seen that for all test images when the number of quantized colors increases, the number of pixels, which at least two of their adjacent pixels possess colors located in the same color region, decreases.

As more color spaces are partitioned, the sizes of color regions become smaller and so the chances that at least two adjacent pixels have their colors located in the same color region. The experimental results also show that the number of quantized colors that yields the best compression rate for all test images is two. Hence, the color quantization of two regions may be enough for the proposed algorithm and so the running time of the algorithm can be reduced significantly. This conclusion may not be universally true depending on the degree of similarities among the colors of the adjacent pixels in a given image. The number of color quantization colors can be tried starting from 2, then 4 and so on until the best compression rate is reached.


fruits


girl


goldhill


lena


airplane

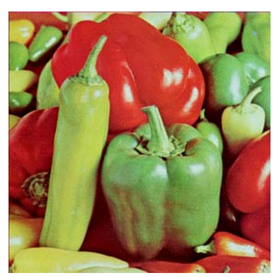

baboon

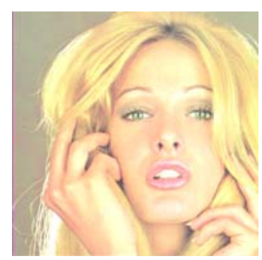

monarch


pens


boat


cablecar


peppers


tiffany


cornfield


flowers


yacht

Fig.9 The fifteen test images

Table 1 The general characteristics of fifteen test images

Image Name	Size (Pixels)	Number of colors
airplane	512 x 512	77,041
baboon	500 x 480	220
boat	787 x 576	140,971
cablecar	512 x 480	130,416
cornfield	512 x 480	134,514
flowers	500 x 362	89,648
fruits	512 x 480	160,476
girl	720 x 576	96,395
goldhill	720 x 576	90,966
lena	512 x 512	59,823
monarch	768 x 512	78,617
pens	512 x 480	121,057
peppers	512 x 512	183,525
tiffany	512 x 512	79,432
yacht	512 x 480	150,053

Table 2 Results of the experiments in average bit rate per color plane

Image Name	JPEG-LS	PNG	Proposed Algorithm			
			Number of quantized colors			
			2	4	8	16
airplane	3.86	4.24	1.18	1.25	1.26	1.27
baboon	6.21	5.76	2.75	2.80	2.83	2.81
boat	3.79	4.19	1.53	1.62	1.69	1.73
cablecar	3.98	4.46	2.12	2.18	2.20	2.24
cornfield	4.47	4.88	2.11	2.25	2.30	2.27
flowers	4.57	5.04	2.26	2.30	2.34	2.37
fruits	4.03	4.47	2.11	2.15	2.20	2.25
girl	4.07	4.36	1.50	1.53	1.58	1.61
goldhill	4.58	4.98	2.15	2.21	2.30	2.33
lena	4.48	5.53	2.08	2.14	2.20	2.25
monarch	3.68	4.09	1.75	1.78	1.81	1.86
pens	4.16	4.47	2.03	2.08	2.17	2.25
peppers	4.81	5.06	1.86	1.90	1.95	2.02
tiffany	3.78	4.04	1.02	1.03	1.03	1.03
yacht	3.96	4.41	2.04	2.10	2.20	2.21

Table 3 The numbers of pixels according to conditions under the rule 4 of the prediction scheme

Image Name	Number of quantized colors	Three adjacent pixels are in the same region	Two adjacent pixels are in the same region	None of adjacent pixels are in the same region
airplane	2	250,322	10,799	-
	4	231,362	28,279	1,480
	8	213,683	42,210	5,228
	16	173,080	71,785	16,256
baboon	2	188,764	50,257	-
	4	161,952	73,072	3,997
	8	113,581	106,742	18,698
	16	73,805	119,486	45,730
boat	2	433,050	18,900	-
	4	382,806	67,575	1,569
	8	319,650	122,682	9,618
	16	242,782	175,509	33,659
cablecar	2	227,708	17,061	-
	4	202,545	40,764	1,460
	8	178,952	58,821	6,996
	16	153,172	75,880	15,717
cornfield	2	208,687	36,082	-
	4	170,288	69,024	5,457
	8	135,671	87,823	21,275
	16	104,966	93,576	46,227
flowers	2	170,126	10,013	-
	4	148,452	30,184	1,503
	8	122,047	50,288	7,804
	16	90,113	70,101	19,925
fruits	2	237,743	7,026	-
	4	219,424	25,184	161
	8	189,702	53,454	1,613
	16	155,241	81,283	8,245
girl	2	402,977	10,448	-
	4	364,798	48,063	564
	8	297,786	112,002	3,637
	16	221,184	171,757	20,484

Image Name	Number of quantized colors	Three adjacent pixels are in the same region	Two adjacent pixels are in the same region	None of adjacent pixels are in the same region
goldhill	2	369,488	43,937	-
	4	323,287	87,660	2,478
	8	265,257	132,265	15,903
	16	204,342	174,127	34,956
lena	2	245,085	16,036	-
	4	213,463	45,707	1,951
	8	173,966	79,643	7,512
	16	123,673	114,712	22,736
monarch	2	359,472	32,465	-
	4	342,662	45,301	3,974
	8	304,845	76,233	10,859
	16	255,361	114,751	21,825
pens	2	225,768	19,001	-
	4	216,091	27,430	1,248
	8	184,729	56,923	3,117
	16	140,220	93,350	11,199
peppers	2	248,857	12,264	-
	4	224,399	35,985	737
	8	189,682	67,894	3,545
	16	130,261	117,195	13,665
tiffany	2	255,955	5,166	-
	4	243,152	17,274	695
	8	222,026	34,554	4,541
	16	186,265	60,162	14,694
yacht	2	220,833	23,936	-
	4	200,450	42,816	1,503
	8	166,094	71,652	7,023
	16	130,653	96,740	17,376

5 Conclusions

We have proposed a new and simple predictive coding algorithm for lossless image compression. The proposed algorithm uses information on the quantized colors of neighboring pixels to estimate the current pixel color value. The heuristic for prediction relies on the fact that if at least two of the neighboring pixels have their colors located in the same quantized color region, they can be similar and therefore the color of current pixel is likely to be similar to those of the neighboring pixels too. The neighboring pixels and the current pixel could be in a part of the image, which pixels exhibit similarities. If there are such similarities among the neighboring pixels, the color value of the current pixel is estimated to be the average of the color values of the similar neighboring pixels. Otherwise, the estimated color value will be the average color value of the three adjacent pixels, which can yield a high value of residual error. It can also be shown that the upper bound on the expected residual error value depends on the number of quantization colors or the sizes of quantized color regions. There exists the optimal number of quantization colors that yields the best compression rate for a particular image. From the results of experiments performed on fifteen test images, the proposed algorithm yields better compression rates than the two popular lossless image compression algorithms of JPEG-LS and PNG. The experimental results also show that quantization for two colors is enough to give the best compression rates for all test images and therefore reduces the processing time of the proposed algorithm.

References:

[1] A. J. R. Neves and A. J. Pinho, A Bit-Plane Approach for Lossless Compression of Color-Quantized Images , *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. 429-432.

[2] C. Chang, P. Xu, R. Xiao, and T. Srikanthan, New adaptive color quantization method based on self-organizing maps, *IEEE Transactions on Neural Networks*, Vol. 16. No. 1, 2005, pp. 237-249.

[3] D. Ozdemir and L. Akarun, A fuzzy algorithm for color quantization of images, *Pattern Recognition*, Vol. 35, No. 8, 2002, pp. 1785-1791.

[4] D. Solomon, *Data Compression : The Complete Reference*, Springer-Verlag London Limited, 2007.

[5] J. A. Robinson, Adaptive Prediction Trees for Image Compression, *IEEE Transactions on image processing*, Vol.15 No.8, 2006, pp. 2131-2145.

[6] J. A. Storer, Lossless Image Coding via Adaptive Linear Prediction and Classification, *Proceedings of the IEEE*, Vol. 88, NO. 11, 2000, pp. 1790-1796.

[7] M. Ciavarella and A. Moffat, Lossless image compression using pixel reordering, *The twenty-seventh Australian Computer Science conference*, 2005, pp. 1790-1796.

[8] M. J. Weinburger, G. Seroussi, and G. Sapiro, The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS, *IEEE Transactions on Image Processing*, Vol. 9, No. 8, pp. 1309-1324.

[9] N. V. Boulgouris, D. Tzovaras, and M. G. Strintzis, Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding, *IEEE Transactions on Image Processing*, Vol. 10, No. 1, 2001, pp. 1-14.

[10] R. C. Gonzales,. and R.E. Woods, *Digital Image Processing*, Prentice-Hall, 2002.

[11] R. N. Neelamani, R. Queiroz, Z. Fan, and R. Baraniuk, JPEG Compression History Estimation for Color Images, *IEEE Transactions on image processing*, Vol. 15 No. 6, 2006, pp. 1365-1378 .

[12] S. Khalid, *Introduction to Data Compression*, Academic Press, 2000.

[13] X. Li, T. Yuan, N. Yu and Y. Yuan, *Pattern Recognition Letters*, Vol. 24, No. 16, 2003, pp. 3165-3176.

[14] Y. C. Hu and B. H. Su, Accelerated K-Means clustering algorithm for color image quantization, *Imaging science journal, The*, Vol. 56, No. 1, 2008, pp. 29-40.

[15] Y. Sirisathikul, S. Auwatanamongkol, and B. Uyyanonvara, Color Image Quantization Using Distance between Adjacent Colors along the Color Axis with the Highest Color Variance, *Pattern Recognition Letter*, Vol. 25, No. 9, 2004, pp. 1025-1043.

[16] Z. Bojkovic and D. Milovanovic, Advanced Image Compression Techniques : Comparative Functionalilties and Performances, *Telsiks*, 2001, pp. 439-442.

[17] Z. Liu, Y. Qian, L. Yang, Y. Bo, and H. Li, An A improved lossless image compression algorithm LOCO-R, *International Conference on Computer Design and Applications (ICCDA)*, Vol. 1, 2010, pp. 328-331.

[18] Z. Pan et al, A Lossless Compression Algorithm for SAR Amplitude Imagery Base on Modified Quadtree Coding of Bit Plane, *IEEE geoscience and remote sensing letters*, Vol.7, No.4, 2010, pp. 723-726.