PREDICTION BASED ON BACKWARD ADAPTIVE RECOGNITION OF LOCAL TEXTURE ORIENTATION AND POISSON STATISTICAL MODEL FOR LOSSLESS/NEAR-LOSSLESS IMAGE COMPRESSION

Xiaohui Xue, Wen Gao

Department of Computer Science, Harbin Institute of Technology, P. R. China, 150001

xhxue@vilab.hit.edu.cn

ABSTRACT

This paper is devoted to prediction-based lossless/near-lossless image compression algorithm. Within this framework, there are three modules, including prediction model, statistical model and entropy coding. This paper focuses on the former two, and puts forward two new methods respectively, they are, prediction model based on backward adaptive recognition of local texture orientation (BAROLTO), and Poisson statistical model. As far as we know, BAROLTO is the best predictor in efficiency. Poisson model is designed to avoid the context dilution to some extent and make use of large neighborhood; therefore, we can capture more local correlation. Experiments show that our compression system (BP) based on BAROLTO prediction and Poisson model outperforms the products of IBM and HP significantly.

1. INTRODUCTION

The success of JPEG has greatly increased the research interests in lossless/near-lossless image compression in recent years. Generally, the first step to compression is decorrelation. There are many decorrelation schemes such as wavelet, hierarchical interpolation, and prediction etc. In the context of lossless/near-lossless image compression, experiments show that there is no obvious difference between the performance of these methods [1][2]; but their computational complexities are very different. Actually, seven out of the nine proposals for JPEG-LS adopted the prediction method with simplicity in mind [3].

Within the framework of prediction-based compression, there are three modules; including prediction model, statistical model and entropy coding. This paper focuses on the former two. In fact, we put forward two new methods for these two modules respectively; they are prediction model based on backward adaptive recognition of local texture orientation (BAROLTO), and Poisson statistics model. Experiments on JPEG test set show that these two new methods are very efficient. As far as we know, BAROLTO is the best predictor. The compression system (BP) based on BAROLTO prediction and Poisson model significantly outperforms the Sunset CB9 system of IBM by 4.1%, the LOCO-I/JPEG-LS system of HP by 5.3%.

Section 2 presents the BAROLTO method. Experimental results are also listed to show the efficiency of BAROLTO prediction. Section 3 describes the Poisson statistical model. Section 4 gives the experimental results of our compression system (BP) based on BAROLTO prediction and Poisson statistical model.

2. PREDICTION BASED ON BACKWARD ADAPTIVE RECOGNITION OF LOCAL TEXTURE ORIENTATION (BAROLTO)

2.1 Problem Statement and Idea of BAROLTO

Linear DPCM methods adopted by the lossless model of JPEG are far from being flexible and powerful enough to provide satisfactory prediction. The key to improve the prediction accuracy is to take advantage of the orientation of local texture. This is also the center of all the predictors in the proposals for JPEG-LS, including the MED (Median Edge Detector) of LOCO-I [3][4] and the GAP (Gradient Adjusted Predictor) of CALIC [3][5] etc. Great improvement on the traditional DPCM methods has been made. However, a closer study on the above predictors can find a potential drawback that both the recognition of edge direction and the generation of prediction value seem to be fairly arbitrary or ad hoc. The predictors can not work accurately and robustly.

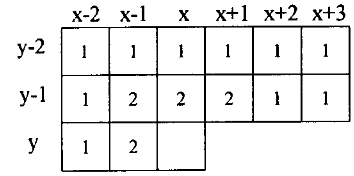
In our viewpoint, in the normal direction of the texture, the variation of the intensity values might be too complicate to provide useful deterministic prediction. Moreover, limited by the raster scan order, it becomes more impracticable to get satisfactory estimation of the profile and the value of the edges. In the tangent direction, however, the variation of the intensities might be relatively smooth, and the nearest pixels in the tangent direction might be a good enough prediction to the current pixel. Therefore, the core of BAROLTO is to recognize the orientation of the local texture accurately and robustly.

2.2 Notation

Consider gray image $I[x,y]$. Let $P[x,y]$, $e[x,y]$, $\tilde{e}[x,y]$, $\tilde{I}[x,y]$ be the prediction value, the prediction error, the quantized prediction error and the reconstruction value respectively, the formula of which can be found later. Define the neighborhood of $[x,y]$ as a set Ω, as illustrated in Fig. 1. Define the set of prediction directions $I = \{i \mid i = 0,1,2,3,4\}$, and the mapping $M : I \mapsto \Omega$, which are illustrated in Fig 2. For example:

$$M_x(0) = x-1, M_y(0) = y,\ M_x(1) = x+1, M_y(1) = y-1,$$

For each prediction direction $i \in I$, define the image of error



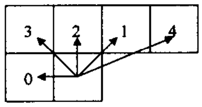


Fig.1 Neighborhood Fig. 2 Prediction directions.

$$D^i[x,y] = \left|\tilde{I}[x,y] - \tilde{I}[M_x(i), M_y(i)]\right|$$

and define the sum of error

$$S^i[x',y'] = \sum_{[x',y'] \in \Omega} \alpha[x',y'] \cdot D[x',y']$$

where $\alpha[x',y']$ are the weights assigned in the neighborhood Ω, as illustrated in Fig. 1.

2.3 Algorithm of BAROLTO

For every $[x,y]$,

Step 1. Compute the sums of error $\{S^i[x,y], i \in I\}$,

Step 2. Sort the sums of error. Suppose

$$S^{i_0}[x,y] \le S^{i_1}[x,y] \le \ldots$$

Step 3. Generate the prediction value

$$P[x,y] = \sum_{j=0}^{1} w[j] \cdot \tilde{I}[M_x(i_j), M_y(i_j)] \Big/ \sum_{j=0}^{1} w[j]$$

where $w[0] = S^{i_1}[x,y]$, $w[1] = S^{i_0}[x,y]$.

Step 4. Generate prediction error

$$e[x,y] = I[x,y] - P[x,y]$$

$$\tilde{e}[x,y] = \begin{cases} \lfloor (e[x,y]+q)/(2q+1) \rfloor & \text{if } e[x,y] \ge 0 \\ \lfloor (e[x,y]-q)/(2q+1) \rfloor & \text{if } e[x,y] < 0 \end{cases}$$

Step 5. Reconstruct value $\tilde{I}[x,y] = P[x,y] + \tilde{e}[x,y]$,

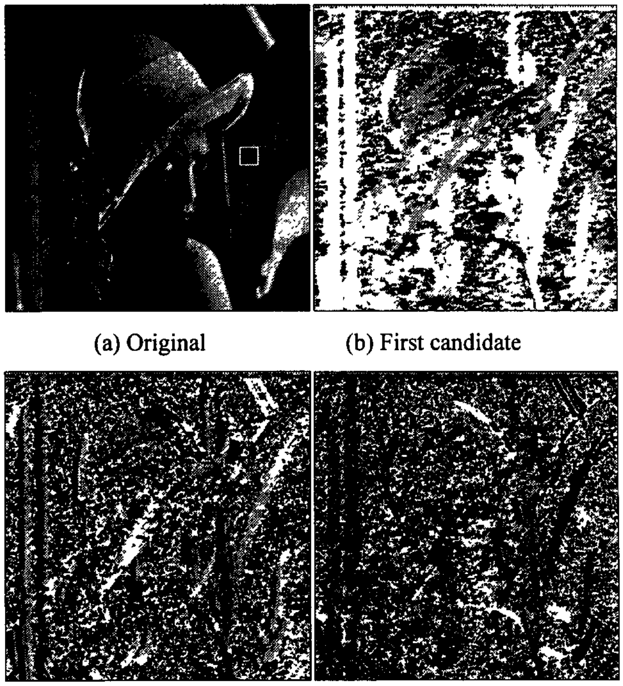
Step 6. Generate the image of error $\{D^i[x,y], i \in I\}$.

Remark: 1). $S^i[x,y]$ is the locally statistical result of the errors of prediction value provided merely by the direction *i*. $S^i[x,y]$ reflects the fitness of the prediction direction locally. The best prediction direction is recognized as the first candidate for the orientation of the local texture. The algorithm selects the best two directions and produces the actual prediction value by a linear combination of the prediction values provided by these two directions; the weights of the combination are also related to the performance of the direction. 2). The recognition of the orientation of local texture is backward adaptive and localized. $\#\Omega$ is selected for the accuracy and robustness of the recognition. 3) We only describe the encoder actually. The decoder is omitted because the algorithm is fairly symmetric. 4) The algorithm can be generalized. The set of prediction directions can be replaced by a set of predictors. As a matter of fact, similar method can be found in [6]. However, such generalization losses the important concept of direction.

2.4 Experimental Result

A. Recognition of Local Texture Orientation

Fig 3(a) is the original Lena. Fig 3(b)(c)(d) illustrate the top 3 candidates for the orientation of local texture among the direction set. Gray level 0, 192, 255, 128, 64 correspond to the direction value 0, 1, 2, 3, 4 respectively. e.g. the brightest points in Fig 3(b) indicate that the corresponding direction value is 2, hence, the first candidate for the local texture orientation at the position is 2. The brightest points in Fig 3(c) indicate that the direction value is 2, hence, the second candidate is 2. Fig 4 lists the values of the first candidate for the local texture orientation corresponding to the marked area in the original Lena.



(a) Original (b) First candidate

(c) Second candidate. (d) Third candidate

Fig. 3 Recognition of local texture orientation.

This experiment shows that the recognition of the local texture orientation is accurate and robust.

B. Prediction Performance of BAROLTO

We compare the prediction performance of BAROLTO with that of other famous predictors in Tab. 1. The values of zero-order entropy of the prediction error on JPEG test set are listed. The data in column 2 to 6 are from [1][2]. BJPEG is the best one of the seven DPCM predictors of JPEG. MAP, OLP, HINT and SCAN are all famous prediction methods, in which MAP is actually adopted by LOCO-I. We can see that our method consistently outperforms all the other predictors by a large margin.

The predictor of CALIC, GAP, is not included in this table because of the absence of data. According to [3], the performance of GAP might be approximately the same as that of MAP.

1	1	1	2	3	1	2	2	2	2	2	2	2	2	1	2
2	2	2	2	1	1	0	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	4	1	1	2	2	2	2	2	2	2	2	2	2	2	2
2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	0	0	0	2	2	2	2	2	2	2	2	2	2	2	1
0	0	0	3	2	2	2	2	2	2	2	2	2	2	1	1
0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	1	2	2	1	1	1	2
2	2	2	2	2	2	2	2	2	2	1	2	2	2	4	4
2	2	2	2	2	2	2	2	2	2	2	2	2	4	4	2
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0

Fig. 4 The values of the first candidate of the marked area

3. POISSON STATISTICAL MODEL

3.1 Problem Statement

The statistical model in lossless/near-lossless compression can be briefly described as follows. For current symbol s_0, the neighborhood of s_0, defined to be a vector $(s_1, s_2, ..., s_k)$, is an ordered set of the past k symbols. k is called the degree of the model. The context of s_0 is defined as a mapping $f(s_1, s_2, ..., s_k)$. The goal of the statistical model is to estimate the probability under the conditional of the context,

$$p(s_0 \mid f(s_1, s_2, ..., s_k))$$

e.g. in the case $f(s_1, s_2, ..., s_k) = (s_1, s_2, ..., s_k)$, which means the context does nothing on the neighborhood of s_0, the output of the statistical model is the *k*-order conditional probability. The cost of the model is S^k, where S is the size of the alphabet of the input symbols.

The main problem on the statistical model is "Context Dilution" [7][8]. Theoretically, the higher the degree, the higher the efficiency. But, in practice, with a high degree, the data must be spread over too many contexts. Consequently, the model can not converge to an acceptable precision. The proposals for JPEG-LS suffer a lot from the context dilution.

In this paper, Poisson statistical model is put forward to avoid the context dilution problem to some extent. So we can make use of large neighborhood and capture more local correlation.

3.2 Algorithm of Poisson Model

Actually, statistical model deals with the quantized prediction error $\tilde{e}[x,y]$. For easy writing, we omit the coordinates, and denote $\tilde{e}$ by e.

Step 1. Quantize $|e|$ to be $|\tilde{e}|$, $|\tilde{e}| \in \{0,1,...,\tilde{n}\}$, $\tilde{n}$ is a small integer number, e.g. $\tilde{n} = 3$.

Step 2. Denote the histogram of $|\tilde{e}|$ in Ω as $\{\tilde{c}[0], \tilde{c}[1], .., \tilde{c}[\tilde{n}]\}$, where $\tilde{c}[i]$ is the count of $|\tilde{e}| = i$ in Ω. $\sum_{i=0}^{\tilde{n}} \tilde{c}[i] = \#\Omega$, $\tilde{c}[i] \geq 0$.

Tab. 1 Zero-order entropy of prediction errors

Images	BJPEG	MAP	OLP	HINT	SCAN	BAROLTO
Baloon.y	3.17	3.16	3.19	3.29	3.01	**2.94**
Barb1.y	5.29	5.22	5.26	5.41	5.06	**4.86**
Barb2.y	5.23	5.21	5.25	5.47	5.08	**4.95**
Board.y	4.18	3.97	4.15	4.21	3.87	**3.72**
Boats.y	4.46	4.33	4.40	4.59	4.32	**4.19**
Girl.y	4.26	4.23	4.30	4.43	4.17	**3.94**
Gold.y	4.85	4.74	4.75	4.90	4.71	**4.67**
Hotel.y	4.93	4.75	4.87	5.04	4.68	**4.55**
Zelda.y	4.15	4.14	4.12	4.13	4.00	**3.87**
Average	4.50	4.42	4.48	4.61	4.32	**4.19**

Step 3. Quantize the histogram to be $\{\tilde{\tilde{c}}[0], \tilde{\tilde{c}}[1], .., \tilde{\tilde{c}}[\tilde{n}]\}$, such that $\tilde{\tilde{c}}[i] \in \{0,1,...,\tilde{\tilde{n}}\}$. $\tilde{\tilde{n}}$ is a small integer number, e.g. $\tilde{\tilde{n}} = 3$.

Step 4. Let $\{\tilde{\tilde{c}}[0], \tilde{\tilde{c}}[1], .., \tilde{\tilde{c}}[\tilde{n}]\}$ be the context of e.

Remark: 1) As far as we know, Ω is the largest neighborhood in the statistical model in the reported papers. 2) The context is not from the values of the prediction errors or the gradients, instead, the context is from the fuzzy histogram of the quantized error. 3) Large as the neighborhood is, it has no structure. We don't care the position of large error; instead, we care the frequency of the large error. Many author put emphasis on the relative positions of large errors to obtain the two-dimensional structure information of the image [9], which brings about the context dilution unavoidably. However, we observe that it is fairly impracticable to get useful estimation about the two-dimensional structure of the image from just a small and incomplete neighborhood. The incompleteness is caused by the raster scan; such incomplete neighborhood is called 180^0 type [8].

3.3 Optimization of Poisson Model

There is two-time quantizations in Poisson model, which generate some equivalence classes in the (error, count) plane. Thus optimizing the Poisson model is equal to optimizing the quantizers. Let us formalize the optimization briefly.

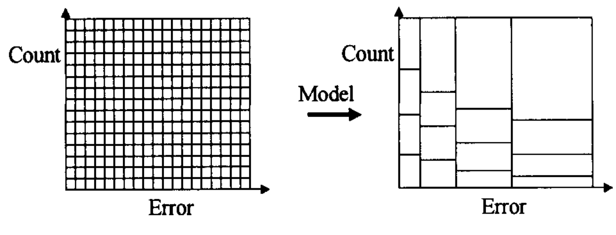


Fig. 5 Quantization of Poisson model

For easy writing, in this section we denote the error by x, denote the count value by y. The distribution of x is approximated by generalized Gaussian density (GDD),

$$p(x) = \left[\frac{vq(v,\sigma)}{2\Gamma(2/v)}\right] \exp(-[q(v,\sigma)\,|\,x\,|]^v)$$

For detail information about GDD, we refer to [10].

Hence, the probability of error x is

$$p_x = \int_{x-0.5}^{x+0.5} p(t)dt$$

The count value of x, denoted by y_x, is approximated by Poisson distribution with parameter $\lambda(x) = p_x \cdot \# \Omega$,

$$p(y_x = k) = e^{-\lambda(x)} \frac{\lambda(x)^k}{k!}$$

The quantizaion of x, denoted by q_x^n, and the quantization of y, denoted by q_y^m, can be formalized as

$$q_x^n(x) = i, \ \ if \ \ x^i \le x < x^{i+1}$$

$$q_y^m(x) = i, \ \ if \ \ y^i \le y < y^{i+1}$$

The computation of Poisson model can be writthen

1). Quantize x: $q_x^n(\Omega) = (q_x^n(x_1), q_x^n(x_2), ..., q_x^n(x_{\#\Omega}))$

2). Count x: $\# q_x^n(\Omega) = (y_0, y_1, ..., y_n)$

3). Quantize count value:

$$q_y^m(\# q_x^n(\Omega)) = (q_y^m(y_0), q_y^m(y_1), ..., q_y^m(y_n))$$

Now, we can write the entropy of Poisson model as

$$H(q_x^n, q_y^m) = -\sum_{x, x_i} p(x, x_1, ..., x_{\#\Omega}) \log_2 p(x \mid (q_y^m(y_0), ... q_y^m(y_n)))$$

$H(q_x^n, q_y^m)$ is a functional with respect to quantization methods q_x^n and q_y^m, and it can be simplified to be a multi-variables function with respect to the thresholds $\{x^i, i = 1,2,...,n\}$ and $\{y^i, i = 1,2,...,m\}$. Thus, the optimization model of Poisson statistical can be written

$$\min_{\substack{\{x^i, i=1,2,...,n\} \\ \{y^i, i=1,2,...,m\}}} H(q_x^n, q_y^m)$$

4. EXPERIMENTAL RESULT

Based on BAROLTO prediction, Poisson statistical model and arithmetic coding [11], we realized a lossless/near-lossless compression system (BP). Tab. 2 is the experimental results with maximum allowable absolute error being 1. The second column is the results of the Sunset CB9 system of IBM [12][13]. The third is of the executables of LOCO-I/JPEG-LS (V.0.90N) [14]. Our BP system consistently provides the best compression on the entire test set and outperforms the product of IBM and HP by 4.1% and 5.3%, which is a significant improvement.

Remark: 1). The famous system CALIC is not included in this table since we failed to find the data. 2). The BP system is far from being optimized. Actually, the LOCO-I/JPEG-LS system runs almost 5 times faster than BP currently. 3). We don't employ error feedback mechanism [3] in the BP system since we believe that Poisson model has captured as much high-order dependencies as possible.

Tab. 2 Experiments on original JPEG image test set (bits/pixel)

Images	IBM	HP	BP
Baloon.y	1.45	1.65	**1.48**
Barb1.y	3.10	3.15	**2.88**
Barb2.y	3.17	3.17	**3.06**
Board.y	2.22	2.20	**2.09**
Boats.y	2.48	2.48	**2.39**
Girl.y	2.38	2.45	**2.28**
Gold.y	3.06	3.00	**2.92**
Hotel.y	2.94	2.87	**2.75**
Zelda.y	2.26	2.38	**2.25**
Ave./Comp.	2.56/4.1%	2.59/5.3%	**2.46**

5. REFERENCES

[1] N. Memon, K. Sayood. "Lossless Image Compression: A Comparative Study", *Proc. SPIE*, Vol. 2814, pp8-20, 1995.

[2] Debin Zhao, "Research on Lossless Still Image Compression", PhD Dissertation, Harbin Institute of Technology, P. R. China. 1997

[3] N. Memon, V. Sippy, X. Wu. "A Comparison of Prediction Schemes for a New Lossless Compression Standard", *1996 International Symposium on Circuit & Systems*, Vol. II, pp309-312, 1996.

[4] W. Weinberger, G. Seroussi, G. Shapiro. "LOCO-I: a low complexity, lossless image compression algorithm", *Proc. of IEEE Data Compression Conf.*, pp140-149, 1996.

[5] X. Wu, et al. "L_∞-Constrained High-Fidelity Image Compression via Adaptive Context Modeling", *Proc. of IEEE Data Compression Conf.*, pp91-100, 1997.

[6] T. Seemann, P. Tisher. "Generalised Locally Adaptive DPCM", *Proc. of IEEE Data Compression Conf.*, pp473, 1997.

[7] M. Weinberger, J. Rissanen, and R. Arps. "Application of Universal Context Modeling to Lossless Compression of Gray-Scale Images", *IEEE Trans. Image Processing*. Vol. 5, No. 4, pp575-586, 1996.

[8] X. Wu. "Lossless Compression of Continueous-Tone Images via Context Selection, Quantization, and Modeling", *IEEE Trans. Image Processing*, Vol. 6, No. 5, pp656-664, 1997.

[9] Debin Zhao, Xiaohui Xue, Yan Liu. "Lossless/Near-Lossless Image Compression Using Context Selection and Arithmetic Coding". *The Fifth International Symposium: The Actual Problem of the Scientific and Technological Progress of the Far Eastern Region*, pp62-67, 1997.

[10] K. Birney, T. Fischer. "On the Modeling of DCT and Subband Image Data for Compression", *IEEE Trans. Image Processing*, Vol. 4, No. 2, 1995: pp186-193.

[11] Xiaohui Xue, Wen Gao. "High Performance Arithmetic Coding for Small Alphabets". *Proc. IEEE Data Compression Conf.*, pp477, 1997.

[12] G. Langdon. "Sunset: A Hardware-Oriented Algorithm for Lossless Compression of Gray Scale Images", *Proc. SPIE*, Vol. 1444, pp272-282, 1991.

[13] G. Langdon and C. Haidinyak. "Experiments with Lossless and Virtually Lossless Image Compression Algorithms", *Proc. SPIE*, Vol. 2418, pp21-27, 1995.

[14] http://www.hpl.hp.com/loco/.