Variations of JPEG-LS and Its Applications

J. Jiang
School of Computing, University of Glamorgan, UK

Abstract: JPEG-LS is the latest JPEG lossless and near lossless image compression standard distributed in 1999. From our experience of testing and evaluating this standard, we propose a number of variations to strengthen its application and efficiency. These include: (a) adaptive distortion distribution for its near lossless compression mode; (b) low-cost rate control schemes, and (c) possible improvement on prediction.

Keywords: image compression, JPEG and signal processing

1. Introduction

The latest JPEG efforts in new international standards for lossless and near lossless image compression is represented by JPEG-LS[4] in which the main compression techniques proposed can be summarised by (i) run-length coding, (ii) non-linear prediction, (iii) context based statistics modelling; and (iv) entropy coding.

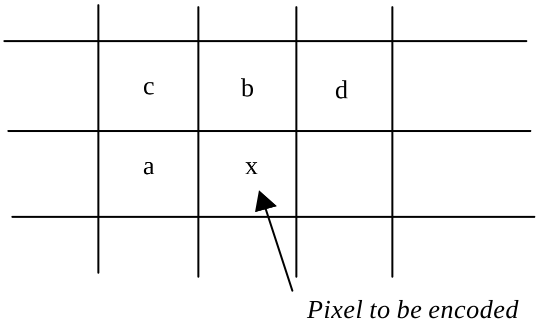


Figure 1 *JPEG-LS predictive pattern*

Run-length coding and prediction-based entropy coding are selected by a template of four neighbouring pixels as illustrated in Figure 1. To reduce the computing cost for statistical modelling and for selection of appropriate coding mode, JPEG-LS proposed the following three delta values to implement the local texture analysis.

$$\Delta_1 = d - b; \quad \Delta_2 = b - c; \quad \Delta_3 = c - a; \qquad (1)$$

For near lossless compression, information loss is introduced in JPEG-LS as a fixed, pre-determined constant value represented by a parameter *NEAR*[4]. If all the four neighbouring pixel values are the same (lossless) or their differences are less than *NEAR* (near lossless), it is a good indication that the local region surrounding the pixel to be encoded is of smooth texture. Hence, run-length coding is applied for encoding the next sequence of pixels until the run is broken. Otherwise, the texture may not be that smooth and thus a prediction based entropy coding is selected. This type of near lossless encoding takes the effect at the decoding end that the reconstructed pixels may have a maximum distortion of *NEAR* in comparison with their original values. Hence, the selection of its coding mode in JPEG-LS can be summarised below:

$$coding\_\mathrm{mod}\,e = \begin{cases} run-length & if\ \Delta_1, \Delta_2\ and\ \Delta_3 < NEAR \\ predictive\ coding & otherwise \end{cases} \qquad (2)$$

Prediction is designed by exploiting a simple local texture analysis among all the three context pixels inside the predictive template. Specifically, comparisons of intensity values of the three surrounding pixels, *a, b,* and *c*, are made to see if any horizontal or vertical edge can be detected. When an edge is detected among the three pixels, the pixel that is not on the edge will be taken as the predictive value. Otherwise, the predictive value will be a well-balanced value drawn from all three pixels. The entire prediction scheme can be described as follows:

$$\begin{array}{l} if\ (c \geq \max(a, b))\ predictive\_value = \min(a, b); \\ else\ \{ \\ if\ (c \leq \min(a, b))\ predictive\_value = \max(a, b); \\ else\ \ predictive\_value = a + b - c; \\ \} \end{array} \qquad (3)$$

To improve the compression efficiency, the predictive errors are further quantized by a fixed quantization step *(NEAR+1)* before they are entropy encoded by Golomb coding[4]. The statistical modelling is carried out by estimating a parameter, k, for each quantized context composed of the three delta values, Δ_1, Δ_2 and Δ_3. The value of k is then used to determine the length of its unary code, which controls the overall length of the code-word.

While it is beyond the category of the paper to justify the existence of such a near lossless compression mode in JPEG-LS in light of the old DCT-based JPEG lossy compression, the JPEG-LS approach does provide the following advantages from our experiences: (i) low complexity in algorithm design, (ii) low cost in implementation; and (iii) closer to lossless compression in terms of quality, compression efficiency and processing speed.

In our experience, it is found that a number of modifications can be made to enhance the near lossless compression performance of JPEG-LS depending on specific applications. This ranges from adding rate control, re-distribute the information to its prediction scheme, which can be detailed in those sections to follow.

2. Redistribution of Information Loss

Considering the fact that JPEG-LS is designed to compress each pixel in one of the three different ways: (i) run-length coding; (ii) non-edge detected prediction and (iii) edge detected prediction, three different information loss levels are firstly designed to implement the principle that information loss should be introduced in regions where larger distortion is tolerable in terms of human visual perception. Consequently, the constant information loss parameter, *NEAR* in JPEG-LS, can be made to vary within the three different levels represented by *NEAR_L*, *NEAR_M* and *NEAR_H*, in the proposed algorithm, to indicate low level information loss, medium level information loss and high level information loss respectively. The corresponding coding operation can be modified below:

$$NEAR = \begin{cases} NEAR\_L & for\, c.m.\, selection\, and\, run-length coding\, \mathrm{mode} \\ NEAR\_M & for\, non-edge\, \mathrm{det}\, ected\ prediction \\ NEAR\_H & for\, edge\, \mathrm{det}\, ected\ prediction \end{cases} \quad (4)$$

Generally, it is expected that the local region with predictive coding mode would normally represent a noisier texture. But it is difficult to substantiate that the edge detected area should be noisier than the non-edge detected. Considering the nature of the predictive coding proposed by JPEG-LS, further analysis reveals that the balanced predictive value, *a+b-c*, in fact accommodates those uncertain situations including (a) similar intensity values among all *a, b, c* pixels, and (b) an edge involves the pixel to be encoded, which is not detectable from the predictive pattern only. Therefore, it may not be exactly right to allocate *NEAR_M* to non-edge detected prediction and *NEAR_H* to edge detected prediction. As proved by our experiments presented in the next section, the initial values selected for both cases are actually the same. The design in equation (5), however, remains useful to reserve its flexibility and variety in consideration of the principle adopted for information loss distribution.

It is also noted that the edge-detected prediction could represent a local area where visual perception may not tolerate high information loss. This occurs when the three pixels in the predictive pattern coincidentally represent a pure step edge inside the input image. Considering the fact that the prediction proposed in JPEG-LS is basically a median filter rather than a proper edge detector, it would be very unlikely that the edge detected by such a prediction scheme happens to be a pure step edge inside the image. In fact, the JPEG-LS prediction is only designed to reflect the intensity variety among a small group of three pixels to produce a good prediction. Hence, in most cases, the so-called edge that is detected by the prediction scheme may not represent any true edge inside the local area. On the other hand, the prediction may not detect any edge even when a pure step edge does exist. In addition, the distortion level allocated is location sensitive, down to each individual pixel. In a very unlikely case that a high distortion level is mistakenly allocated to a pure step edge, the effect is normally compensated in the next few pixels, since a smoother texture is bound to follow and the distortion level will be reduced accordingly. Indeed, our visual inspection proves this point. Having said that, further improvement may be achievable if a more reliable texture analysis method is used to pinpoint the information loss level for each pixel encoded. This would significantly increase the complexity of the overall algorithm design, which may not be justified in the case of JPEG-LS.

3. Rate Control Design

In computer science, storage of image information requires that each image should have a predictable file size and thus a dedicated space can be allocated for the convenience of storage management. When data compression is applied, however, the compression ratio is dependent on the compressibility of each input image and hence the size of each image after compression may vary. In this circumstance, rate control would be desired to control the compression ratio at a target level regardless of its compressibility.

To design such a rate control for JPEG-LS near lossless compression mode, the important issues would include: (a) the level of complexity and computing cost should be low; (b) the reconstructed image quality at the decoding end should be the best possible, or the effect of rate control upon the quality should be minimised. To this end, a number of compression assessment points are designed at the end of each row. At each assessment point, the on-line compression ratio is compared with the target compression ratio to determine whether the information loss parameter, (the value of NEAR), for the next row of pixels should be increased or decreased. Whenever the on-line compression ratio is detected to be smaller than the target, we increase the information loss parameter to force the compression ratio changing towards the target. Otherwise, we would decrease the parameters to prevent any excessive distortion from being introduced and to improve the image quality. To facilitate the rate control operation of identifying the correct parameter for the next adjustment, we name the array, which holds the remaining parameters, I-array (for increase) or D-array (for decrease) depending on whether the information loss parameter has just been increased or decreased for the current row. Once this array is named correctly, the other array must be either D-array or I-array accordingly. In this way, the action by rate control upon the information loss parameter for the next row can be described as follows:

(i) To increase the information loss parameter, the algorithm will search the I-array in a top-down direction. The first parameter accessed will be increased and moved to the D-array at its corresponding position.

(ii) If the next adjustment is to decrease the information loss parameter, the algorithm would search the D-array instead in a bottom-up direction and move the first parameter accessed to the I-array at its corresponding position;

(iii) Whenever the parameter being moved is the last one inside the array (either I or D), the array which is full of parameters need to be renamed as an I-array if the next adjustment is to increase and D-array if the next move is to decrease, respectively. In other words, the array full of parameters is always renamed by the nature of the next adjustment. This is designed to ensure that the correct order of movement is always maintained even when one of the array becomes empty.

The overall rate-control algorithm can also be described by a flow-chart given in Figure 2.

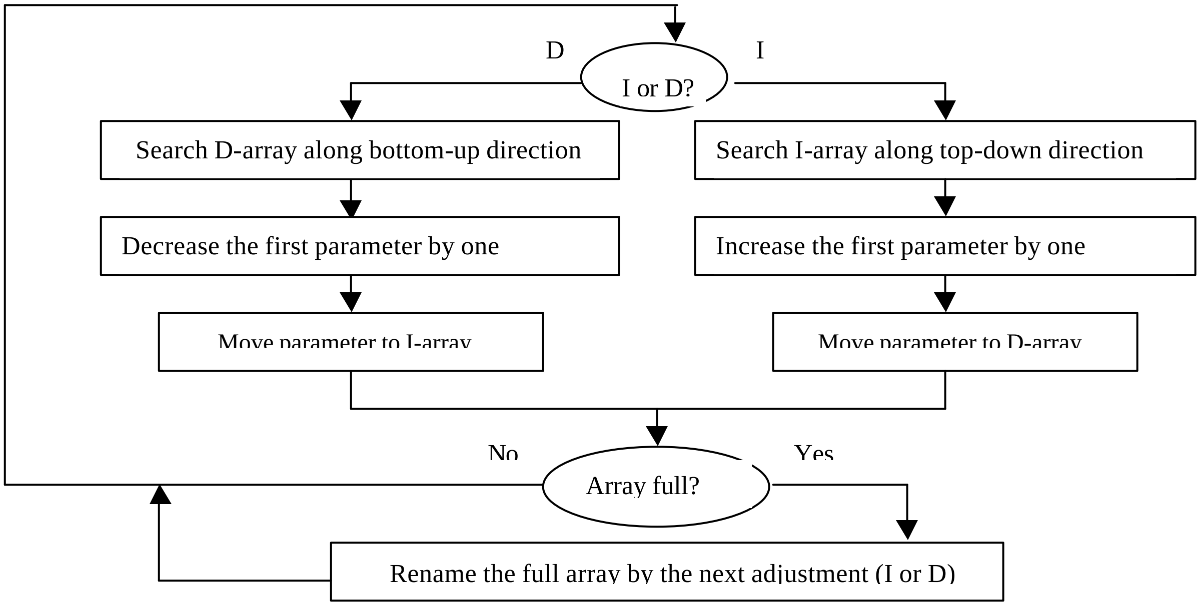


Figure 2 *Algorithm flow chart*

To assess the proposed rate control algorithm, we designed a series of experiments to test the rate control effectiveness and its efficiency. The effectiveness of rate control is to show that the proposed algorithm is able to achieve a number of target compression ratios, which include 2:1 and 3:1 for a group of six image samples selected from public domain. Since JPEG-LS is a standard for lossless/near lossless image compression, intention for any higher target compression ratios has to be referred to JPEG lossy compression standard (DCT based). The rate control efficiency refers to the assessment of the reconstructed image quality for those image samples measured by both PSNR values and visual inspection. In all the assessments, the non-rate-controlled JPEG-LS will be used as our benchmark to provide a comparable indication of the performance of the proposed rate control algorithm.

In order to determine an appropriate initial setting for the three parameters, (L,M,H), we present the experimental results in two phases, i.e., the test of rate control efficiency as the first phase, and the test of rate control effectiveness as the second phase.

The first phase of the experiments requires assessment of image quality after the rate control is added. The overall objective is to optimise the image quality or to minimise the negative effect of rate control, although it may be inevitable to introduce additional information loss in order to force the compression ratio close to the target.

Before the testing is started, the initial values of the three parameters, L, M and H, need to be determined. The principle involved is to avoid unnecessary transitions or adjustments for those cases, where the target compression ratio is obviously far away from some initial settings. One extreme example is that, when the target compression is 3:1, an initial setting of (0,0,0) would take quite a few unnecessary adjustments before they are anywhere near to the target. Specifically, for any range of target compression ratios, the initial setting must be designed to reflect an appropriate balance between compression ratio and quality. If the initial settings are too high, excessive information loss would have been introduced even before the rate control is started. On the other hand, lower values of the initial setting might fail the control algorithm to achieve the target compression.

To measure the rate controlled quality by PSNR values and compare the results with non-controlled JPEG-LS, we realise the difficulties in producing the same compression ratios before the comparison can be made. This is because that it would be pointless to compare the PSNR values with different compression ratios. To this end, we firstly run the JPEG-LS on the same group of image samples to produce a set of variable compression ratios and PSNR values. We then use those compression ratios achieved by JPEG-LS as the target compression ratio to run our rate controlled program and then compare the PSNR values to assess the rate controlled quality. As a matter of fact, this is not really a fair comparison, since the JPEG-LS does not have any rate control mechanism and hence no side effect from rate control can be possibly incurred. On the other hand, rate control will inevitably introduce some additional distortion no matter how optimised the algorithm may be. Nonetheless, we expect the comparison to give us an indication on how negative our rate control effect is in comparison with the ideal situation where no rate control is added.

The test results under the design are given in Table I. The quality performance can be measured by a relative difference between the rate-controlled PSNR and the non-rate-controlled PSNR values, which can be defined as follow

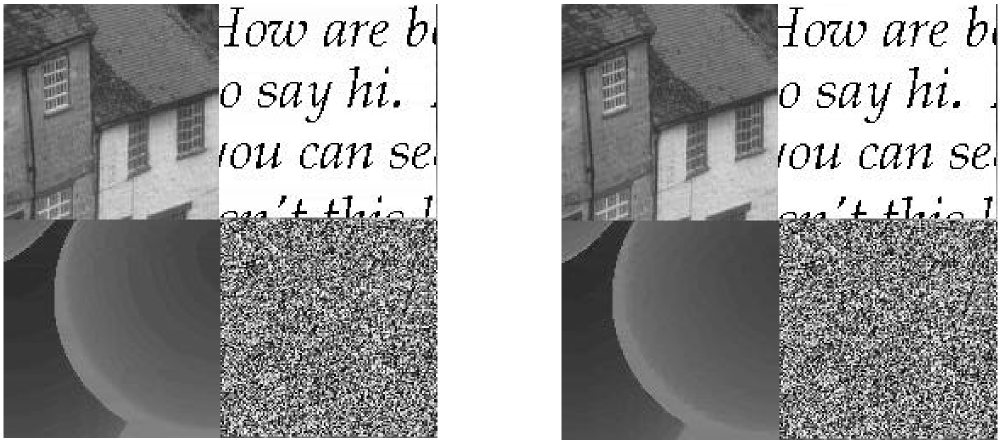
$$\mathrm{Re}\,lative\_diff = \frac{PSNR_{not\_controlled} - PSNR_{rate\_controlled}}{PSNR_{not\_controlled}} \quad (5)$$

All the values of *Relative_diff* are given in the last column of Table I. From the results given in Table I, it is clearly seen that the rate control effect varies between the range of (1.2%, 9.2%), which is a reasonable price to pay for the rate-controlled compression.

Considering the fact that rate control will inevitably introduce extra information loss in order to achieve a compression ratio close to the target, it has been illustrated that the image quality produced by the proposed rate control algorithm is close to that by the non-controlled JPEG-LS, when the quality is measured by PSNR values. In fact, the proposed algorithm is aiming at optimising the image quality measured by human visual inspection rather than by PSNR values. This is because, essentially, the compression techniques are the same. The major technique proposed is to re-distribute the extra information loss incurred by rate control to those areas where local texture is detected to be noisy. Hence the extra distortion may be concealed from visual inspection since human perception of quality is less sensitive to these areas. To prove this point, we further present the *test8g*, the JPEG standard test image, in Figure 3 for a visual comparison, in which Fig.3 -(a) corresponds to the reconstructed *test8g* by the non-controlled JPEG-LS and Fig.3-(b) to the reconstructed image by the proposed algorithm. Although it may be difficult to find out which one is closer to the original *test8g*, the reconstructed image by the proposed algorithm seems to provide a smoother texture at the bottom left part of the image.

This part of experiments also helps us to determine an appropriate initial setting for the three parameters L, M, and H. From a range of possible candidates, the initial setting is determined as: (1, 1, 5).

With the same initial setting of (1,1,5), the second phase of experiments is to run the rate control algorithm on the same group of six image samples and test the effectiveness of the proposed algorithm, in which the target compression ratios are set to be 2:1 and 3:1. The experimental results are given in Table II-(a) and (b) respectively.



(a): *Reconstructed test8 by JPEG-LS* **(b):** *Reconstructed test8 by the proposed algorithm*

Figure 3: *Visual comparison of test8*

Table I: Phase-1 Rate Control Experimental Results (Initial L, M, H = 1,1,5)

Image samples	**Proposed Algorithm**		**Non Rate Controlled JPEG-LS**			**Relative-diff**
	CR	**PSNR(db)**	**CR**	**PSNR(db)**	**NEAR**	
Baboon.pgm	3.717	33.979	3.721	34.403	8	1.2%
Barb.pgm	4.645	35.921	4.644	38.449	5	6.6%
Bridge.pgm	3.791	34.526	3.813	35.532	7	2.8%
Camera.pgm	4.718	36.725	4.907	40.447	4	9.2%
Salad.pgm	4.787	34.604	4.783	35.586	7	2.7%
Test8.pgm	3.027	39.042	3.127	42.677	3	8.5%

Table II-(a) Phase-2 Rate Control Experimental Results
(Initial L, M, H = 1,1,5 and Target-Compression-Ratio = 2:1)

Image Samples	**The Proposed Algorithm**		*The Non-rate-controlled JPEG-LS*		
	C. Ratio	**PSNR(dB)**	*Closest C. Ratio*	*PSNR(dB)*	*NEAR Values*
Baboon	2.000	41.034	1.800	49.685	1
Barb	2.000	51.214	1.690	INF	0
Bridge	1.990	43.089	1.899	49.899	1
Camera	2.000	43.687	1.849	INF	0
Salad	1.990	47.602	1.578	INF	0
Test8	1.990	50.221	1.915	INF	0

Table II-(b) Rate Control Experimental Results

(Initial L, M, H = 1,1,5 and Target-Compression-Ratio = 3:1)

Image Samples	**The Proposed Algorithm**		*The Non-rate-controlled JPEG-LS*		
	C. Ratio	**PSNR(dB)**	*Closest C. Ratio*	*PSNR(dB)*	*NEAR Values*
Baboon	3.010	37.489	2.998	38.153	5
Barb	3.010	39.718	2.524	49.893	1
Bridge	2.980	36.108	2.956	39.950	4
Camera	2.990	40.285	2.802	49.927	1
Salad	2.970	38.873	2.810	45.143	2
Test8	2.940	39.134	3.127	42.677	3

In order to see the effect that, without rate control, the compression ratios achieved would be variable and dependent on the compressibility of each input image, we also presented the compression ratios achieved by the non-controlled JPEG-LS in the last column of table II. Note that the non-controlled JPEG results are obtained by varying the value of *NEAR* manually and picking up the nearest one to each target compression ratio. In fact, this is not practically possible by manually adjusting the parameter *NEAR* for each input image to achieve a compression ratio close to the target. This is because that we do not know the compressibility of each input image before the compression can be done, and it is not allowed in practice to compress each image with various NEAR values and then choose the closest one.

From all the results presented in Table II, it can be summarised that: (i) all compression ratios achieved by our algorithm are indeed controlled at the target compression ratio; (ii) without rate control, the nearest compression ratio achieved by JPEG-LS is still not close enough to the target, even when the parameter *NEAR* is adjusted manually.

4. Revisit of Prediction

Analysis of the prediction adopted in JPEG-LS can reveal that the existing scheme can only detect either a vertical edge or a horizontal edge among the three context pixels, and the prediction is made based on three conditions: (i) $c \ge \max(a,b)$; (ii) $c \le \min(a,b)$ and (iii) $\min(a,b) \le c \le \max(a,b)$. To further improve the prediction scheme, a diagonal edge detection can be considered in constructing the predictive value. Specifically, the three context pixels and the pixel to be encoded can be viewed as a local rectangular region. Inside the rectangular region, a diagonal edge can only exist in two possibilities, which are 45° for predictive conditions (i) and (ii), and 135° for condition (iii). Hence, all the possibilities for the existence of a diagonal edge can be described as *left-high/right-low* and *left-low/right-high*, where *–high* and *–low* correspond to high intensity and low intensity pixel values, *left-* and *right-* to the two parts divided by the diagonal edge inside the rectangular region, which is constructed by *a, b, c, and x* .

When the condition $\min(a,b) \le c \le \max(a,b)$ is satisfied, it can be seen that only a 135° diagonal edge exists, which naturally leads to two possibilities, either *left-high/right-low* or *left-low/right-high*, depending on the relationship between *a* and *b*. Consequently, specific edge type can only be determined by the unknown pixel *x*. For the case of *left-low/right-high*, as an example, the fact that the unknown pixel *x* is close to *a, b,* or *c* would give us a horizontal edge, a vertical edge or a diagonal edge. For other cases, similar analysis can be made. Therefore, it is extremely difficult to detect the edge with any reasonable accuracy, since the correct detection has to be dependent on the value of the unknown pixel *x*. Hence, it is unlikely to have any further improvement with this predictive context.

With the condition $c \ge max(a, b)$, the existing JPEG-LS scheme would give us either a horizontal edge or a vertical edge among the three pixels, depending on which pixel out of *a* and *b* is larger. Assuming that a diagonal edge does exist, it would be helpful to examine what conditions should be satisfied in order to have such a diagonal edge. Firstly, it should be established that either a vertical edge or a horizontal edge can not coexist with the diagonal edge, since it would be difficult to justify any alternative prediction. To this end, we can conclude that between the intensity values of *c* and *max(a, b)*, there should exist a certain gradient, in order to allow a diagonal edge be established and the possibility of a vertical/horizontal edge be minimised. This gives us the following condition:

$$c - max(a, b) > threshold_1 \tag{6}$$

Secondly, from the nature of the diagonal edge, other conditions can be easily identified below:

$$x < min(a, b) \quad \text{and} \quad d < b \tag{7}$$

Only the second half can be used to detect the diagonal edge, since the first half in equation (7) involves the unknown pixel *x*.

On the other hand, when such a diagonal edge is detected, the predictive error can be seen large produced by the existing JPEG-LS scheme. This is because the predictive value in JPEG-LS is always *min(a, b)*, yet the existence of a diagonal edge would produce a large difference between *x* and *min(a, b)*. Therefore, in this circumstance, further improvement over JPEG-LS is possible.

When $a < b$ is satisfied, however, it is difficult to construct a predictive value even when a diagonal edge is detected accurately by equation (6) and (7). Although $x < a$ is expected, there is no correlated pixel on this row to compensate for the difference: *x-a*. Hence, we abandon the attempt to propose any modification for this situation. This would leave us to the case: $a > b$. Since the potential for improvement over JPEG-LS lies in the fact that there could exist a large difference between *x* and *b* ($b = \min(a, b)$), it would be essential for the proposed scheme to detect such a difference. By applying the two conditions, $a > b$ and $c \geq \max(a, b)$ to the rectangular region, the difference between *b* and *x* can be indirectly detected by the difference between *a* and *x*, assuming equation (6) and (7) are established. Hence, if we introduce a gradient detection between *a* and *b*, a large difference between *a* and *x* can be guaranteed since we have: $a > b$ and $b > x$ As a result, such a gradient detection can be constructed as follows:

$$a - b \geq threshold_2 \tag{8}$$

After the detection is completed, the major problem with JPEG-LS can be revealed that the predictive value *b* is still far too large compared with *x* and thus the predictive error is large. Consequently, a compensation can be introduced by considering the value of pixel *d.* As a result, the new predictive value under this circumstance can be produced below:

$$P = \frac{d + b}{2} \tag{9}$$

Corresponding to the condition: $c \leq \min(a, b)$, we would have *left-low/right-high* in the local rectangular region. Similar analysis can be made to reveal that if a diagonal edge exists among the three context pixels, the predictive value, $\max(a, b)$, produced by JPEG-LS, may not be large enough since a significant difference between *x* and *max(a, b)* could be expected in this case. Hence, when such a condition is detected correctly, the following predictive value should be constructed to minimise the predictive error:

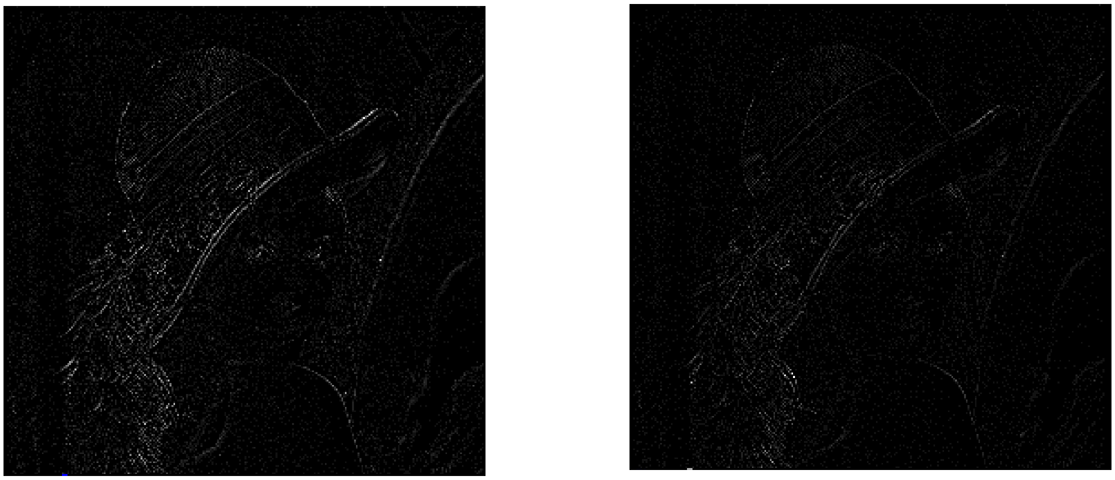
$$P = \frac{d + \max(a, b)}{2} \tag{10}$$

In summary, the proposed prediction scheme can be finalised by adding two new predictive values given in equation (9) and (10) into the diagonal edge detection when the two conditions, $c \geq \max(a, b)$ and $c \leq \min(a, b)$, are satisfied.

To test the proposed algorithm, a software implementation is run on a group of four known images: *Lena, Boat, Camera,* and *Clown*. In order to ensure a fair comparison with the existing JPEG-LS scheme, we assessed the performance of the proposed scheme in two measurements: (i) the mean-square-error between each pixel to be encoded and its predictive value; (ii) the compression ratio. With both schemes, only the prediction is different and all other parts are maintained exactly the same. The experimental results are summarised in Table III, in which we let $threshold_1 = 10$ and $threshold_2 = 5$.

Table III Experimental Results

Samples	Size	JPEG-LS MSE Values	The Proposed MSE Values	JPEG-LS Comp. Ratio	The Proposed Comp. Ratio
Lena	512×512	37.435	35.104	1.991	1.991
Boat	512×512	58.174	56.964	1.881	1.881
Camera	256×256	215.414	214.457	1.849	1.849
Clown	256×256	93.715	92.008	1.702	1.702



(a) *Predictive errors of Lena produced by JPEG-LS*

(b) *Predictive errors of Lena produced by the proposed scheme*

Figure 4 *Visual comparison of predictive error images*

From the results, it is seen that the proposed scheme outperforms JPEG-LS in terms of MSE values. Hence, the predictive errors are further minimised by the proposed scheme. However, the results in Table I also shows that the compression ratios stay the same. This is expected due to the fact that the entropy coding length is determined by statistical modelling, yet this part of operation is not revised in the light of the proposed diagonal edge detection (in JPEG-LS, the contexts for statistical modelling is based on *d-b, b-c* and *c-a*). In this sense, smaller predictive errors may not always produce higher compression efficiency.

To further illustrate the improvement over JPEG-LS by the proposed scheme, we present all the predictive errors as an image and thus enable a visual comparison between the two predictive schemes. Figure 4 illustrates such a visual comparison, in which the two error images, produced by JPEG-LS and the proposed scheme, are given in part (a) and (b) respectively. From these illustrations, it can be concluded that the difference between the two error images is indeed noticeable.

5. Conclusions

In this paper, we described a few different ways to modify JPEG-LS to suit some applications when the standard is to be applied to practical image compression. The modifications presented include redistribution of information loss, adding rate control and improving prediction accuracy. The first two target the near lossless image compression mode, in which appropriate balance is needed in order to achieve the best rate-distortion performance for the algorithm. The last one can only improve the prediction accuracy. To achieve an overall improved performance for image compression, more research work is required to consider the diagonal edge detection inside the statistical modelling and hence the prediction accuracy can be exploited to reduce the redundancy designed in entropy coding and quantization.

The author wishes to acknowledge the support from HP Lab Bristol for supplying the JPEG-LS source code and the draft document.

References

1. Chen C.T. and Wong A. ‘A self-governing rate buffer control strategy for pseudoconstant bit rate video coding’, IEEE Trans. Image Processing, Vol 2, Jan 1993, pp 50-59.
2. Choi J. and Park D. ‘A stable feedback control of the buffer state using controlled Lagrange multiplier method’, IEEE Trans. Image Processing, Vol 3, Sept 1994, pp 546-558.
3. Hsu C. Y., Ortega A. and Reibman A. ‘Joint selection of source and channel rate for VBR video transmission under ATM policing constraints’ IEEE J. Select. Areas Commun., Vol 15, Aug. 1997, pp 1016-1028.
4. ISO/IEC JTC 1/SC 29/WG1 FCD 14495-public draft, July 16th, 1997. http://www.jpeg.org/public/jpeglinks.htm
5. JTC 1.29.14, 15444: Call for contributions for JPEG-2000, http://www.jpeg.org
6. Keesman G., Shah I. And Klein-Gunnewiek R. ‘Bit-rate control for MPEG encoders’ Signal Processing, Image communication, Vol 6, pp 545-560, Feb 1995.
7. Lin L.J., Ortega A. and Kuo C.C.J. ‘Rate control using spline-interpolated R-D characteristics’ Proc. SPIE Visual Commun. Image Processing’96, Orlando, FL., Mar. 1996, pp 111-122.
8. Lin L.J. and Ortega A. ‘Bit-rate control using piecewise approximated rate-distortion characteristics’ IEEE Trans. Circuits & Systems for Video Technology, Vol 8, No 4, August 1998, pp 446-459.
9. MPEG-2 encoder v.1.1a, MPEG Software simulation Group, (online) http://www.mpeg.org/tristan/MPEG/MSSG
10. MPEG-2, Test model 5(TM5) Doc. ISO/IEC JTC1/SC29/WG11/93-225b, Test Model Editing Committee, Apr 1993.
11. Ortega A., Ramchandran K. and Vetterli M. ‘Optimal trellis-based buffered compression and fast approximations’, IEEE Trans. Image Processing, Vol 3, Jan 1994, pp 26-40.
12. Reibman A.R. and Haskell B.G. ‘Constraints on variable bit-rate video for ATM networks’, IEEE Trans. Circuits Syst. Video Technology, Vol 2, pp 361-372, Dec. 1992.
13. Shoham Y. and Gersho A. ‘Efficient bit allocation for an arbitrary set of quantizers’, IEEE Trans. Acoust., Speech, Signal Processing, Vol 36, Sept 1988, pp 1445-1453.
14. Wiegand T., Lightstone M, et al. ‘Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard’, IEEE Trans. Circuits Syst. Video Tech. Vol 6, Apr 1996, pp 182-190.
15. Wu S.W. and Gersho A. ‘Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV’ IEEE Trans. Circuits Syst. Video Technology, Vol 1, Mar 1991, pp100-112.
16. Zdepsky J., Raychaudhuri D. and Joseph K. ‘Statistically based buffer control policies for constant rate transmission of compressed digital video’, IEEE Trans. Communication, Vol 39, No 6, June 1991, pp 947-957.