Lossless Image Coding via Adaptive Linear Prediction and Classification

GIOVANNI MOTTA, STUDENT MEMBER, IEEE, JAMES A. STORER, MEMBER, IEEE, AND BRUNO CARPENTIERI, MEMBER, IEEE

In the past years, there have been several improvements in lossless image compression. All the recently proposed state-of-the-art lossless image compressors can be roughly divided into two categories: single and double-pass compressors. Linear prediction is rarely used in the first category, while TMW [7], a state-of-the-art double-pass image compressor, relies on linear prediction for its performance.

We propose a single-pass adaptive algorithm that uses context classification and multiple linear predictors, locally optimized on a pixel-by-pixel basis. Locality is also exploited in the entropy coding of the prediction error. The results we obtained on a test set of several standard images are encouraging. On the average, our ALPC obtains a compression ratio comparable to CALIC [20] while improving on some images.

Keywords*—*Adaptive coding, arithmetic codes, data compression, Golomb–Rice codes, gradient methods, image coding, linear predictive coding.

I. INTRODUCTION

After the Call for Contributions ISO/IEC JTC1/SC29/WG1 (lossless JPEG) [3], the field of gray-level lossless image compression received heightened attention. Most of the contributions are effective in compressing images, while holding down the computational complexity and the memory requirements. On the other hand, most approaches use heuristics and, even if the compression ratio achieved cannot be in practice easily improved, it is not completely clear whether or not these algorithms are able to achieve the real entropy of the image.

In [7] and [8], B. Meyer and P. Tischer proposed TMW, a dual-pass, lossless image coding algorithm that, by using linear predictors, achieves compression performance higher than CALIC [20], the best (in terms of compression ratio) single-pass algorithm known so far. TMW improves the current best results by using global optimization and blended linear predictors. A TMW compressed file consists of two parts: a header that contains the parameters of the model and the encoded data itself.

Even if TMW has a computational complexity several orders of magnitude greater than CALIC, the results obtained by this algorithm are in any case surprising because:

- Linear predictors are known not to be effective in capturing fast transitions in image luminosity (edges) [19].
- Before TMW, global optimization seemed unable to improve substantially the performance of lossless image compressors [19].
- CALIC was thought to achieve a data rate extremely close to the real entropy of the image [19].

In this paper, we propose ALPC, a single-pass adaptive algorithm that uses context classification and multiple linear predictors that are locally optimized on a pixel-by-pixel basis. The explicit use of locality is one of the key features of our algorithm, and local features of the image being encoded are also exploited in the entropy coding of the prediction errors.

We describe here a series of experiments we made with ALPC addressing the problem of gray-level lossless image compression exclusively from the viewpoint of the achievable compression ratio, without being concerned about computational complexity or memory requirements.

The results we obtained on a test set of several standard images are encouraging. On the average, ALPC obtains a compression ratio comparable to CALIC while improving on some images of the test set. Experiments suggest that, with a better encoding of the prediction error, our algorithm can be competitive with CALIC and TMW.

II. DESCRIPTION OF THE ALGORITHM

ALPC is an algorithm based on adaptive linear prediction and consists of two main steps: pixel prediction and entropy coding of the prediction error. A pseudocode description of the whole algorithm is given in Fig. 1.

An input image is encoded in a single pass, by processing its pixels in raster-scan order (i.e., proceeding from top to bottom and from left to right). The luminosity of each pixel,

Manuscript received February 15, 2000; revised June 26, 2000.

G. Motta and J. A. Storer are with the Department of Computer Science, Brandeis University, Waltham, MA 02454-9110 USA (e-mail: gim@ieee.org; storer@cs.brandeis.edu).

B. Carpentieri is with the Dipartimento di Informatica ed Applicazioni "R.M. Capocelli," Universitá degli Studi di Salerno, Baronissi 84081 (SA), Italy (e-mail: bc@udsab.dia.unisa.it).

Publisher Item Identifier S 0018-9219(00)09992-8.

```
for every pixel PIX(x,y) in the input image do begin
     Collect the pixels in W_{x,y}(R_p) and their context
     Determine n centroids C_1,...,C_n by applying the
          LBG on the contexts in W_{x,y}(R_p)
     Let K_1,...,K_n be the corresponding clusters
     Classify each pixel/context in W_{x,y}(R_p) in one of
          the clusters K_1,...,K_n
     Classify the context of the current pixel PIX(x,y);
          let k be the index of that cluster
     Let P_i={w_0,..., w_5} be the predictor that achieves
          the smallest error on C_k among a set of
          predictors P_1,...,P_n
     Apply the Gradient Descent on the pixels in C_k to
          refine the predictor P_i
     Use the refined predictor P'_i to predict PIX(x,y)
     Generate the prediction error ERR(x,y)
end
```

Fig. 1. Pseudocode description of the adaptive prediction.

PIX(x, y), is predicted by a weighted sum of its neighbors (or context). ALPC uses the fixed-shape context that is depicted in Fig. 2.

Pixels forming the context of PIX(x, y) have been previously encoded, and both encoder and decoder know their values exactly; to keep encoder and decoder synchronized it is sufficient to round the prediction to its nearest integer value

$$\begin{aligned}\overline{PIX}(x, y) = \text{int}(&w_0{}^*PIX(x, y-2)\\ &+ w_1{}^*PIX(x-1, y-1)\\ &+ w_2{}^*PIX(x, y-1)\\ &+ w_3{}^*PIX(x+1, y-1)\\ &+ w_4{}^*PIX(x-2, y)\\ &+ w_5{}^*PIX(x-1, y)).\end{aligned}$$

Pixels are individually encoded by representing the *prediction error* ERR(x, y), i.e., the difference between the pixel being encoded, PIX(x, y) and its predicted luminosity

$$ERR(x, y) = \overline{PIX}(x, y) - PIX(x, y).$$

The prediction error ERR(x, y) is the only information that is sent to the decoder for the reconstruction of the input image. Encoding the prediction error is advantageous because its typical distribution allows an efficient entropy coding.

The predictor's weights $w_0, \cdots, w_5$ are adapted during the encoding process to take into account local features of the image. Starting from a default set of weights, $w_0, \cdots, w_5$ are changed and optimized on a per-pixel basis.

For each pixel, a new predictor is determined by minimizing the energy of the prediction error inside a small window $W_{x,y}(R_p)$ of previously encoded pixels

$$\begin{aligned}&\min_{w_0, \cdots, w_5} E(x, y)\\ &\quad = \min_{w_0, \cdots, w_5} \sum_{PIX(x', y') \in W_{x,y}(R_p)} (ERR(x', y'))^2.\end{aligned}$$

Fig. 3 shows the window $W_{x,y}(R_p)$ with radius R_p, centered on PIX(x, y). The radius R_p allows explicit control on the degree of locality that our algorithm exploits.

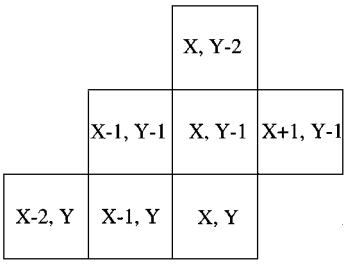

Fig. 2. Pixel PIX(x, y) and its context.

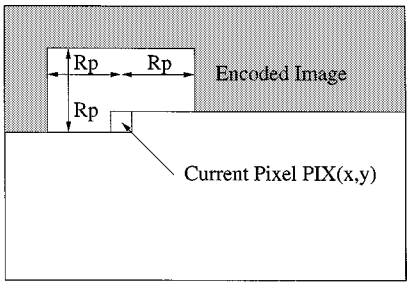

Fig. 3. Prediction window $W_{x,y}(R_p)$ of radius R_p and centered on PIX(x, y).

Not all the samples collected in the window are used for the optimization. The pixels contained in the window are partitioned into clusters by minimizing the distance between their contexts. A centroid is determined for each cluster and the context of the current pixel is classified. Only the cluster whose centroid is closest to the current pixel's context is used to select and refine a predictor.

By using a window of previously encoded pixels, ALPC implements a backward prediction scheme so the encoder has no need to send any side information to the decoder. On the other hand, backward prediction has as a well-known major drawback: poor performance in the presence of edges.

The radius R_p of the window $W_{x,y}(R_p)$ (see Fig. 3) is one of the essential features of our algorithm. Its size affects the prediction quality because if R_p is too small, only few samples are in the window and the predictor "overspecializes," making big errors when in the presence of edges. On the other hand, too many samples in the window (R_p too big) tend to generate predictors that are not specific enough to adapt to local variations in the image.

Although in our experiments we decided to keep R_p constant and equal for all images, it is possible to design algorithms that dynamically adapt the size of the radius to image characteristics so that prediction and convergence speed may be improved.

As mentioned before, to improve prediction, predictor optimization is performed only on a subset of samples collected in the window. The rationale is that we want the predictors' weights to be representative of the relation existing between the context and the pixel being encoded. By discarding samples whose context varies too much from the one of the pixels currently being encoded, we can specialize the prediction and follow fine periodic patterns in the window.

Most algorithms existing in the literature use a simple pixel predictor and compensate the poor prediction with

sophisticated heuristics to model the error probability as a function of the context in which it occurs (see, e.g., LOCO-I [14]). In ALPC, instead, because of the classification, the contextual encoding is embedded in the error prediction mechanism.

The classification algorithm that we used in our experimentation is a variation of the generalized Lloyd algorithm (or LBG [5]). This classification method, although not optimal in our framework, is powerful enough to improve the performance of the basic adaptive predictor. We are confident that a more appropriate classification algorithm will result in improved performance.

Once the samples in the window are classified and a representative centroid is determined for each cluster, one of the clusters is selected according to the minimum distance between the context of the corresponding centroid and the context of the pixel being encoded. Similarly, in a set of predictors, the one that achieves the lowest prediction error on the selected centroid is chosen. This is the predictor that is further refined by applying a gradient descent optimization on the samples of the selected cluster. At each step t of the optimization, while the difference between the previous and the current errors is smaller than a fixed threshold, the weights w_i of the predictor are changed according to

$$w_i(t+1) = w_i(t) - \mu \frac{\partial E}{\partial w_i}$$

where E is the error energy and μ a small constant. When too few samples are collected in the window (e.g., at the beginning, when PIX(x, y) is close to the top or to the left border of the picture), a default fixed predictor is used in the prediction and the gradient descent optimization is not applied. In our implementation, we used as default predictor the "planar predictor" described in [19]

$$P_{def} = \{w_0 = 0, w_1 = -1, w_2 = 1, \\ w_3 = 0, w_4 = 0, w_5 = 1\}.$$

To make the algorithm independent of the predictors' initialization, we use at each step the set of predictors that were refined from the previous iterations. With this choice, we observed that the predictors $P_1, \cdots, P_n$ can be initialized to random values without compromising the performance. We also experimented by initializing with the predictors used in JPEG-LS; this only resulted in a slightly faster convergence of the gradient descent optimization. Reinitializing the predictors at every step, instead of using the previous refined weights, while resulting in a much slower convergence, does not seem to affect substantially the compression ratio.

III. Entropy Coding

Image prediction error is commonly modeled in literature by using a Laplacian distribution [2]. Likewise, ALPC's adaptive linear prediction generates a skewed Laplacian-like distribution, centered on zero and with quasi-symmetric very long tails (See Fig. 4).

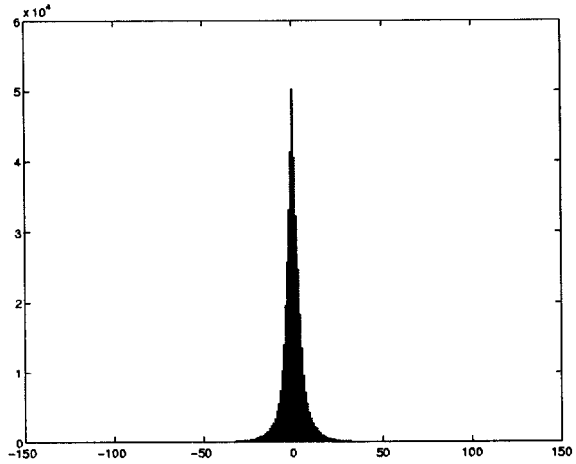

Fig. 4. Histogram of the prediction error for the test image "hotel."

Efficient entropy encoding can be performed by using an arithmetic encoder [17] or, for reduced complexity, by using a Golomb–Rice code as described in [14].

We experimented with both methods, using both a standard approach and determining the parameters of the encoder's model in a window centered on the error being encoded. Some of the experiments we made with the arithmetic encoder were previously reported in [10].

Entropy coding with an arithmetic encoder is performed by dividing the coding step in two parts: the determination of a probabilistic model for the source and the entropy coding that uses that model. In general, when a simple adaptive model is used, the data stream is encoded symbol by symbol, by updating the probabilistic model after each encoding. Every once in a while, the model is renormalized, both to avoid overflow and to provide a way of "discounting" the past statistics. Renormalization improves the exploitation of nonstationarities of the data source; however, when encoding images, this method is not very effective because:

- the data are bidimensional but the model adaptation proceeds in a linear order;
- careful control of the time window is not possible.

To evaluate the effect of explicit discounting, we experimented both with the classic adaptive method and with determining the probabilistic model for the prediction error currently being encoded (ERR(x, y)) in a window $W_{x,y}(R_e)$ of radius R_e centered on ERR(x, y). Results obtained on a set of test images are listed in Table 1. For the test images we used, highest compression was achieved when determining the model of the arithmetic coder in windows of radius R_e comprised between eight and 12 pixels.

We believe that performance can be further improved by using a more appropriate implementation of the arithmetic coder. The implementation that we used [16] has no special modeling for the probabilities of symbols that have not been seen before [18], and a small default probability is always assigned to every symbol of the alphabet. When only a small number of samples are available to model the distribution (a thing that happens frequently when the model is determined

Table 1
Comparison Between Four Entropy Coding Methods: Golomb–Rice Coding (GR), Arithmetic Coding (AC), Golomb–Rice with the Model in a Window $W_{x,y}(R_e)$ (GR-W), Arithmetic Coding with the Model in a Window $W_{x,y}(R_e)$ (AC-W). Results are Shown in Bits Per Pixel. Test Images are 512×512 pixels (except MRI and X-ray that are 256×256), 8 Bits/Pixel. These Results are Obtained by Using Two Predictors and $R_p = 10$

	GR	AC	GR-W	AC-W
Airplane	3.80	3.81	3.66	3.62
Airport	6.65	6.65	6.69	6.60
Crowd	4.31	4.21	3.98	4.03
Goldhill	4.71	4.70	4.69	4.66
Hursley	4.79	4.77	4.54	4.55
Lake	5.07	5.11	4.97	4.93
Landsat	4.16	4.14	4.09	4.04
Lax	5.84	5.83	5.81	5.75
Lena	4.18	4.21	4.12	4.05
Lenna	4.05	4.09	3.97	3.91
Mandrill	5.76	5.79	5.73	5.65
Milkdrop	3.71	3.66	3.64	3.57
Mri	3.49	3.48	3.25	3.13
Mskull	2.78	2.57	2.62	2.26
Peppers	4.27	4.25	4.22	4.16
Woman1	4.65	4.64	4.52	4.49
Woman2	3.26	3.28	3.14	3.06
Xray	2.63	2.63	2.52	2.56
Avg. (bpp)	**4.45**	**4.43**	**4.35**	**4.28**

by using a window), the contribution of the default probabilities become relevant and the encoder efficiency is compromised.

Although entropy coding contributes only marginally to the complexity of our algorithm, the nature of the error distribution allows us to enhance the speed by implementing the entropy coding by mean of a low-complexity Golomb–Rice code.

Golomb–Rice codes are a particularization of the Golomb codes introduced by S. W. Golomb in [1] to encode nonnegative symbol run lengths. These codes are characterized by an encoding parameter m that is constrained to be a power of two. Nonnegative integers $n \leq 0$ are encoded in two parts: a binary representation of $n \bmod m$ and a unary representation of $\text{int}(n/m)$; such codes are described in [14] and in [15]. Because the parameter is a power of two, the operations of modulus and integer division can be simplified to masking bits and can be implemented very efficiently both in software and in hardware.

As described in [14], a further simplification can be introduced: The parameter m that characterizes the distribution can be reliably estimated by using the expectation of the magnitude of the prediction errors. To map prediction errors to nonnegative values, we "folded" the distribution by using the mapping also described in [14]

$$M(ERR(x,y)) = \begin{cases} 2 \cdot ERR(x,y), & \text{if } ERR(x,y) \geq 0 \\ 2 \cdot |ERR(x,y)| - 1, & \text{otherwise.} \end{cases}$$

Table 1 shows the results obtained on a set of test images both for an adaptive Golomb–Rice coding and when determining the parameter k in a window $W_{x,y}(R_e)$ centered on ERR(x, y). Experimentally we found that for Golomb–Rice codes, windows of radius R_e of between three and five pixels achieve highest compression.

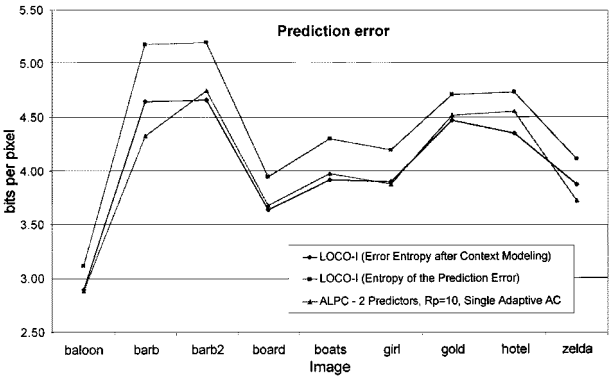

Fig. 5. Comparisons with the entropy of the prediction error in LOCO-I.

Although Golomb–Rice coding is optimal for exponentially distributed sources, the main drawback of this method is that at least one bit must be sent for each pixel. This well-known limitation penalizes "simple" images having long runs of errors having the same identical value (typically zeros). Although we did not implement this solution, a common fix to this problem is to detect long runs of symbols and switch the encoding to the so-called "run length" mode.

IV. Results and Discussion

Experiments were performed in order to assess the algorithm on different test sets composed, respectively, of nine images of $720 \cdot 576$ pixels, 16 images of $512 \cdot 512$ pixels, and two images of $256 \cdot 256$ pixels. These test sets are widely used for comparisons in the lossless data compression literature and can be downloaded from several ftp sites, including [9] and [23]. All images were gray-level (continuous-tone) digitized with a resolution of 8 bits/pixel (256 gray levels).

We report our results by giving for each image the average number of bits used to encode a pixel (empirical entropy).

Fig. 5 gives information on the efficiency of ALPC's predictors by comparing the entropy of the prediction error of the simple fixed predictor used in LOCO-I with the entropy of the prediction error achieved by our algorithm. The results reported in Fig. 5 were obtained by using two predictors and by optimizing the predictors in a window of radius $R_p = 10$. As a reference, we also report the overall performance of LOCO-I after the entropy coding that uses sophisticated context modeling.

It is evident how our adaptive linear predictors are (understandably) much more powerful than the fixed predictor used in LOCO-I; however, even when enriched by the classification, our adaptive prediction does not have enough power to capture edges and sharp transitions, present, for example, in the picture "hotel" (see bottom of Fig. 6).

Tables 2–4 summarize the experiments we made in order to understand the sensitivity of the algorithm to its parameters (number of predictors and window radii). In these experiments, we measured the variations on the compression achieved when parameters are changed one at a time.

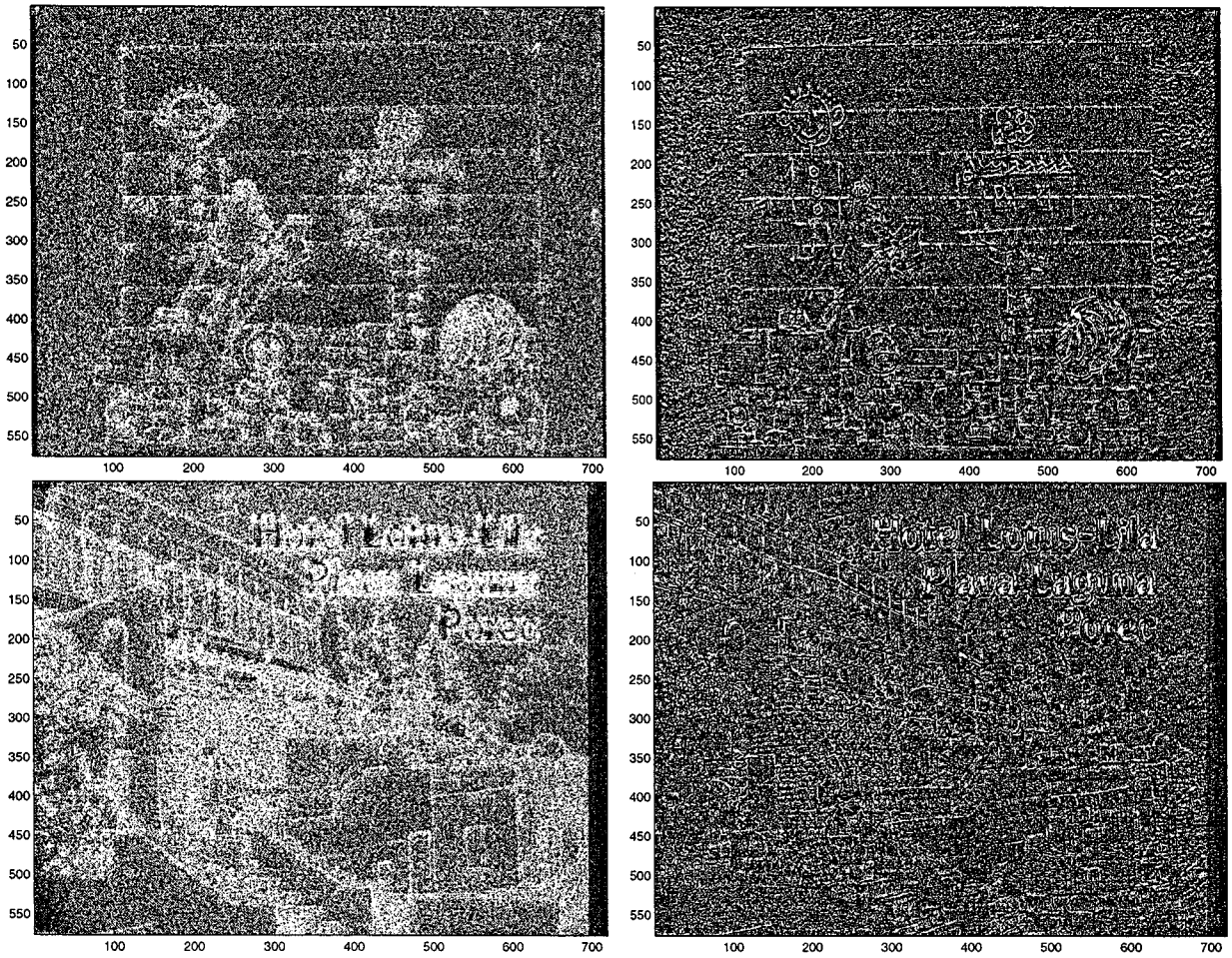

Fig. 6. Top: Magnitude (left) and sign (right) of the prediction error of the test image "Board." Bottom: Magnitude (left) and sign (right) of the prediction error of the test image "Hotel."

Table 2
Entropy in Bits Per Pixel Versus Number of Predictors. Results Shown for a Window of Radius $R_p = 6$; Error is Coded by Using a Simple Adaptive Arithmetic Encoder

Predictors	1	2	4	6	8
Baloon	2.976	2.901	2.906	2.898	2.899
Barb	4.391	4.320	4.336	4.345	4.358
Barb2	4.827	4.836	4.911	4.956	4.988
Board	3.724	3.666	3.675	3.670	3.677
Boats	4.055	4.013	4.040	4.042	4.062
Girl	3.935	3.897	3.903	3.904	3.908
Gold	4.546	4.579	4.605	4.618	4.636
Hotel	4.553	4.570	4.615	4.630	4.644
Zelda	3.763	3.739	3.753	3.765	3.771
Avg. (bpp)	**4.086**	**4.058**	**4.082**	**4.092**	**4.105**

Table 3
Entropy in Bits Per Pixel Versus Window Radius R_P. The Number of Predictors Used is Two; Prediction Error is Entropy Encoded by Using a Simple Adaptive Arithmetic Encoder

Rp	6	8	10	12	14
Baloon	2.901	2.892	2.891	2.894	2.899
Barb	4.320	4.311	4.332	4.347	4.362
Barb2	4.836	4.810	4.748	4.765	4.750
Board	3.666	3.671	3.683	3.698	3.714
Boats	4.013	3.986	3.977	3.978	3.983
Girl	3.897	3.881	3.879	3.885	3.891
Gold	4.579	4.540	4.518	4.515	4.515
Hotel	4.570	4.544	4.550	4.556	4.563
Zelda	3.739	3.724	3.720	3.721	3.725
Avg. (bpp)	**4.058**	**4.040**	**4.033**	**4.040**	**4.045**

In Table 2, the number of predictors is changed while keeping fixed the window radius $R_p = 6$; conversely, in Table 3, the number of predictors is kept fixed at two and the performance with respect the window size changes are evaluated.

Both experiments described in Tables 2 and 3 were performed by using a simple entropy coding scheme for the prediction error: a single adaptive arithmetic coder. As we also verified experimentally, the performance of a single adaptive arithmetic encoder is a close approximation of the first-order entropy of the prediction error.

Table 4 reports the conclusive experiments; the number of predictors is kept fixed to two $R_p = 10$, and performance is evaluated by encoding the prediction error with a probabilistic model determined in a window. Results are reported for changes in the value of R_e.

Comparisons with some popular lossless image codecs (see Tables 5, 6 and Fig. 7) show that the proposed algorithm achieves good performance on most test set images. The cases were we fall short of CALIC confirm that linear prediction, even when combined with classification, is not adequate to model image edginess. Also, unlike CALIC, our codec does not use any special mode to encode high contrast image zones, so our results are penalized by images like

Table 4
Entropy in Bits Per Pixel Versus Error Window Radius R_e. Two Predictors are Used and $R_p = 10$. Prediction Error is Encoded by Determining the Model in a Window Centered On the Current Pixel

Re	6	8	10	12	14	16	18	20
Baloon	2.846	2.840	2.840	2.842	2.845	2.848	2.851	2.853
Barb	4.213	4.180	4.168	4.165	4.166	4.169	4.174	4.178
Barb2	4.582	4.528	4.500	4.489	4.484	4.483	4.484	4.488
Board	3.608	3.595	3.591	3.592	3.594	3.597	3.600	3.603
Boats	3.932	3.900	3.889	3.886	3.887	3.889	3.892	3.896
Girl	3.820	3.805	3.800	3.801	3.803	3.805	3.807	3.809
Gold	4.487	4.449	4.431	4.423	4.420	4.418	4.418	4.418
Hotel	4.458	4.422	4.410	4.407	4.408	4.410	4.413	4.416
Zelda	3.671	3.651	3.642	3.638	3.636	3.635	3.636	3.636
Avg. (bpp)	3.958	3.930	3.919	3.916	3.916	3.917	3.919	3.922

Table 5
Final Compression Results (in Bits Per Pixel). Test Images are 720×756, 8 Bits/Pixel. The Number of Predictors Used in ALPC is Two, and $R_p = 10$. Entropy Encoding is Performed with an Arithmetic Coder and the Model is Determined in a Window of Radius $R_e = 10$

	CALIC	LOCO-I	Sunset	UCM	TMW	ALPC
Baloon	2.78	2.90	2.89	2.81	2.60	2.84
Barb	4.31	4.69	4.64	4.44	3.83	4.16
Barb2	4.46	4.69	4.71	4.57	4.24	4.48
Board	3.51	3.68	3.72	3.57	3.27	3.59
Boats	3.78	3.93	3.99	3.85	3.53	3.89
Girls	3.72	3.93	3.90	3.81	3.47	3.80
Gold	4.35	4.48	4.60	4.45	4.22	4.42
Hotel	4.18	4.38	4.48	4.28	4.01	4.41
Zelda	3.69	3.89	3.79	3.80	3.50	3.64
Avg. (bpp)	3.88	4.06	4.08	3.95	3.63	3.91

Table 6
Final Compression Results for the Another Test Set. Test Images are 512×512 (Except MRI and X-ray that are 256×256), 8 Bits/Pixel. Two Predictors are Used in These Experiments and $R_p = 10$. Entropy Encoding is Performed with an Arithmetic Coder and the Model is Determined in a Window of Radius $R_e = 10$

	CALIC	LOCO-I	ALPC
Airplane	3.54	3.61	3.62
Airport	6.55	6.71	6.60
Crowd	3.76	3.91	4.03
Goldhill	4.63	4.71	4.66
Hursley	4.39	4.43	4.55
Lake	4.90	4.98	4.93
Landsat	3.99	4.08	4.04
Lax	5.63	5.78	5.75
Lena	4.11	4.25	4.05
Lenna	3.94	4.07	3.91
Mandrill	5.74	5.89	5.65
Milkdrop	3.56	3.63	3.57
Mri	3.15	3.36	3.13
Mskull	2.16	2.23	2.26
Peppers	4.20	4.29	4.16
Woman1	4.54	4.67	4.49
Woman2	3.20	3.20	3.06
Xray	2.59	2.46	2.56
Avg. (bpp)	4.26	4.36	4.28

"hotel" that have high-contrast regions. A closer look to the prediction error magnitude and sign for "board" and "hotel," two images in the test set, shows that most of the edges in the original image are still present in the prediction error (Fig. 7).

V. Conclusion

The results of our experiments on the test sets of standard images are encouraging. With a better classification and selection of the contexts in the prediction window and with a more sophisticated encoding of the prediction error, it may be possible to achieve stable and better results on all the test images. Also critical in the encoding of the prediction error is a careful determination of the probability of novel symbols.

Also, it is likely that the computational complexity can be substantially reduced without sacrificing the performance by using alternative methods for the optimization of the predictors [6]. Further reduction in the complexity was made possible by replacing the arithmetic coder with Golomb–Rice codes.

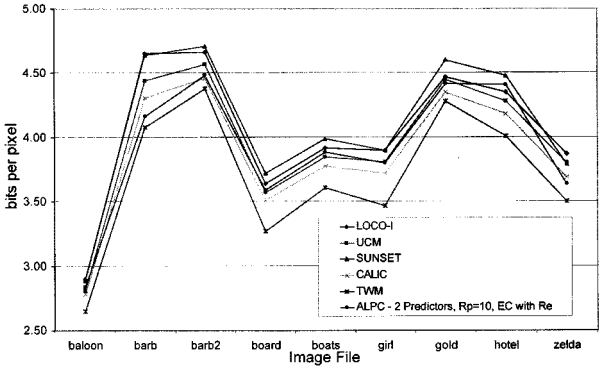

Fig. 7. Graphical representation of the data in Table 5.

Acknowledgment

The authors would like to thank M. Cohn and F. Rizzo for fruitful discussions and suggestions.

References

[1] S. W. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory*, vol. 12, pp. 399–401, July 1996.

[2] P. G. Howard, "The design and analysis of efficient lossless data compression systems," Ph.D. dissertation, Dept. Comput. Sci., Brown Univ., Providence, RI, June 1993.

[3] ISO/IEC, "Call for contributions—Lossless compression of continuous-tone images,", JTC1/SC29/WG1, Mar. 1994.

[4] G. G. Langdon Jr., "Sunset: A hardware oriented algorithm for lossless compression of gray scale images," *Proc. SPIE*, vol. 1444, pp. 272–282, Mar. 1991.

[5] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[6] P. A. Maragos, R. W. Schafer, and R. M. Mersereau, "Two-dimensional linear prediction and its application to adaptive predictive coding of images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, Dec. 1984.

[7] B. Meyer and P. Tischer, "TMW—A new method for lossless image compression," in *Proc. Int. Picture Coding Symp. (PCS97)*, Sept. 1997.

[8] B. Meyer and P. Tischer, "Extending TMW for near lossless compression of greyscale images," in *Proc. IEEE Data Compression Conf. (DCC98)*, Snowbird, UT, Mar. 1998.

[9] B. Meyer. TMW program and most recent results. [Online] Available: http://www.cs.monash.edu.au/ bmeyer/tmw

[10] G. Motta, J. A. Storer, and B. Carpentieri, "Adaptive linear prediction lossless image coding," in *Proc. IEEE Data Compression Conf. (DCC99)*, Snowbird, UT, Mar. 1999.

[11] G. Seroussi and M. J. Weinberger, "On adaptive strategies for an extended family of Golomb-type codes," in *Proc. IEEE Data Compression Conf. (DCC97)*, Snowbird, UT, Mar. 1997.

[12] D. Speck, "Fast robust adaptation of predictor weights from min/max neighboring pixels for minimal conditional entropy," in *Proc. 29th Asilomar Conf. Signal, Systems and Computers*, Pacific Grove, CA, Oct., pp. 234–242.
[13] D. Speck, "Activity level classification model (ALCM),", JTC 1.29.12, 1995.
[14] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based, lossless image compression algorithm," in *Proc. IEEE Data Compression Conf. (DCC96)*, Snowbird, UT, Mar. 1996.
[15] M. J. Weinberger, G. Seroussi, and G. Sapiro, "From LOCO-I to JPEG-LS standard," in *Proc. 1999 Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999.
[16] F. Wheeler. Adaptive arithmetic coding. [Online] Source code available at http://ipl.rpi.edu/wheeler/ac/
[17] I. H. Witten, R. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, Jun. 1987.
[18] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inform. Theory*, vol. 37, July 1991.
[19] X. Wu, "An algorithmic study on lossless image compression," in *Proc. IEEE Data Compression Conf. (DCC96)*, Snowbird, UT, Mar. 1996.
[20] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec," *IEEE Trans. Commun.*, vol. 45, Apr. 1997.
[21] X. Wu, "Efficient lossless compression of continuous-tone images via context selection and quantization," *IEEE Trans. Image Process.*, vol. 6, pp. 656–664, May 1997.
[22] X. Wu, W. Choi, and N. Memon, "Lossless interframe image compression via context modeling," in *Proc. IEEE Data Compression Conf. (DCC98)*, Snowbird, UT, Mar. 1998.
[23] X. Wu. Test images. [Online] ftp.csd.uwo.ca/pub/from_wu/images/

Giovanni Motta (Student Member, IEEE) received the M.S. degree in information technology from CEFRIEL, Milano, Italy, in 1993, and the "Laurea" degree in Scienze Dell'Informazione (summa cum laude) from Universitá degli Studi di Salerno, Salerno, Italy, in 1996.

He is currently a Ph.D. candidate at the Department of Computer Science, Brandeis University, Waltham, MA. His research interests are trellis vector quantization, lossless image compression, low bit-rate video coding, low bit-rate speech coding, and information theory.

James A. Storer (Member, IEEE) received the B.A. degree in mathematics and computer science from Cornell University, Ithaca, NY, in 1975, and the M.A. and Ph.D. degrees in computer science from Princeton University, Princeton, NJ, in 1977 and 1979, respectively.

From 1979 to 1981, he was a Researcher (MTS) at Bell Laboratories, Murray Hill, NJ. In 1981, he came to Brandeis University, Waltham, MA, where he is currently Chair of the Computer Science Department and Member of the Brandeis Center for Complex Systems.

Bruno Carpentieri (Member, IEEE) received the "Laurea" degree in computer science from the University of Salerno, Salerno, Italy, and the M.A. and Ph.D. degrees in computer science from Brandeis University, Waltham, MA.

Since 1991, he has been Assistant Professor of Computer Science (Ricercatore) at the University of Salerno. His research interests include lossless image compression, video compression and motion estimation, information hiding, parallel computing, and theory of computation. He was co-chair, in 1997, of the International Conference on Compression and Complexity of Sequences, and, since 1997, he has been a Program Committee Member of the IEEE Data Compression Conference. He has been responsible for two European Commission contracts regarding the report on the state-of-the-art of lossless image compression.