Local Prediction for Lossless Image Compression

Ahmad Daaboul

Institut Gaspard Monge
University of Marne-la-Vallée
Cité Descartes, 5, Bd Descartes, Champs-sur-Marne
77454 Marne-la-Vallée CEDEX 2
France

e-mail: `daaboul@monge.univ-mlv.fr`

Abstract. In predictive coding a group of neighboring picture elements is used to select a suitable prediction value for a current pixel. In this paper, we propose two techniques for lossless images compression based on predictive coding. In the first technique which called, *the predictors*, we replace each pixel in the image by the predicted pixel; we use various schemes to predict the value of a pixel. In the second, which is based on predictor technique, and called *optimal prediction schemes*, we divide the original image into blocks or lines and seek the best predictor for each (among a selected set of eight) that provides the best prediction. The errors image is encoded through arithmetic coding, during the final step of compression. The gains of compression that we obtained are observed in the lossless image compression.

Key words: Image compression and Predictive coding.

1 Introduction

The aim of image compression is to represent a given image with the minimal number of bits in order to accelerate transmission or reduce storage. Image compression can be divided into two categories. In the first, *lossy compression* : we accept a difference between the original image and the decompressed one. Second, *lossless compression*: after a cycle of compression/decompression, the decompressed image is identical to the original image.

Most image compression techniques are lossy. However, there are many applications which require lossless compression. For example in medical and satellite images no loss of information can be tolerated. In our work we focused lossless image compression.

Among the various methods which have been devised for lossless compression, *predictive coding* is perhaps the most simple and efficient. In predictive coding, a prediction is made for the current pixel based on the values of previously encountered neighboring pixels. For every input x_N pixel, a *predictor* generates a prediction value which is calculated from $N-1$ preceding samples. A *predictor* is a linear or non-linear combination of neighboring pixels of a current pixel. We call *error image* the difference between the original image and the predicted image. If the prediction scheme is satisfactory then the distribution of prediction error is concentrated near zero and the error image has a significantly lower entropy compared to the original image. Lossless image compression techniques [TLR85] identify two basic steps: *decorrelation* and *coding*. In the decorrelation step, redundancies among the pixels are reduced. In the coding step, the *error image* is encoded into a binary string using a variable length code, such as a *Huffman coding* [Hu52] or *arithmetic coding* [BCW90, R76].

The predictors proposed by Wallace [W91] define the JPEG lossless image compression standard. Harrison [H52] proposed two predictors, others were defined by Todd, Langdon and Rissanen [TLR85]. In this paper, we describe new predictors. Secondly, we select some JPEG predictors, Harrison predictors and two of our predictors. This set of selected predictors is used to predict a sample of pixels and we choose (among the set of selected predictors) the one that provides the best prediction for this sample. The error image obtained is encoded by a zero order arithmetic coding.

We consider an image to be an array P of integers of two dimensions $M \times N$ such that $0 \leq m < M$ and $0 \leq n < N$, where M denotes the number of lines of P and N, the number of columns. In this paper Pr represents a predictor.

2 Predictors techniques

Among all the methods of lossless image compression, the methods based on predictors are the simplest. These methods take into account the value of a pixel compared to its neighbors. Different predictors have been proposed to predict the value of a pixel at the location (m, n). Harrison [H52] has proposed some predictors. These predictors are called *slope predictor* (Pr_s):

$$Pr_s(m,n) \ = 2 * P[m, n-1] - P[m, n-2].$$

and *Plane 3 predictor* (Pr_{p3}):

$$Pr_{p3}(m,n) \ = \frac{2}{3}P[m,n-1] + \frac{2}{3}P[m-1,n] - \frac{1}{3}P[m-1,n-1].$$

Other predictors are defined in [TLR85], they are called *Plane* 2 *predictor* (Pr_{p2}):

$$Pr_{p2}(m,n) \ = P[m,n-1] + (P[m-1,n+1] - P[m-1,n-1])/2.$$

and *Right diagonal* (Pr_{p3}):

$$Pr_{rd}(m,n) \ = P[m-1,n-1].$$

Table 1 contains the predictors proposed by Wallace [W91] which are used as the JPEG lossless image compression standard.

Predictor	Prediction
$Pr_{J1}(m,n)$	$P[m-1,n]$
$Pr_{J2}(m,n)$	$P[m,n-1]$
$Pr_{J3}(m,n)$	$P[m-1,n-1]$
$Pr_{J4}(m,n)$	$P[m-1,n]+P[m,n-1]-P[m-1,n-1]$
$Pr_{J5}(m,n)$	$P[m-1,n]+((P[m,n-1]-P[m-1,n-1])/2)$
$Pr_{J6}(m,n)$	$P[m,n-1]+((P[m-1,n]-P[m-1,n-1])/2)$
$Pr_{J7}(m,n)$	$(P[m-1,n]+P[m,n-1])/2$

Table 1: JPEG predictors.

In this work, we propose new predictors which correspond to different schemes of linear prediction. Table 2 contains our predictors.

Predictor	Prediction
$Pr_1(m,n)$	$(2*P[m,n-1]+P[m-1,n-1])/3$
$Pr_2(m,n)$	$(2*P[m,n-1]+P[m-1,n])/3$
$Pr_3(m,n)$	$max(P[m,n-1],P[m-1,n])$
$Pr_4(m,n)$	$(P[m,n-1]+P[m-1,n]+P[m-1,n-1])/3$
$Pr_5(m,n)$	$(3*P[m,n-1]+P[m-1,n]+P[m-1,n-1])/5$
$Pr_6(m,n)$	$max(P[m,n-1],P[m-1,n],P[m-1,n-1])$
$Pr_7(m,n)$	$(P[m,n-1]+P[m-1,n]+P[m-1,n-1]$ $+P[m-1,n+1])/4$
$Pr_8(m,n)$	$(P[m,n-1]+P[m-1,n])+P[m-1,n-1]$ $+P[m-1,n+1]+P[m-1,n+2])/5$

Table 2: Our predictors.

3 Optimal prediction schemes

The best method to predict a pixel is to compare it with its neighbors in the same sample, and select the neighbor that provides the best prediction, i.e. the nearest value among his neighbor. Beginning with this idea, we search for the best predictor for a sample of pixels, not of only one. For this purpose, we divide the original image into *line* and *block* as described in sections 3.1 and 3.2. The basic idea of our technique is to use a sample of predictors to predict an image or a part of an image. This sample is composed by eight predictors : Pr_{J1}, Pr_{J2}, Pr_{J5}, Pr_{J6}, Pr_{J7}, Pr_{p3}, Pr_2 and Pr_3. The reason for choosing these eight predictors is that the values of these predictors are used to detect the magnitude and orientation of edges in the input image (or sample of pixels) and make necessary adjustments in the prediction.

We predict the sample pixels using all predictors, and we compute the *zero-order entropy* of error image with each of them. The best predictor is the one that provides the lower zero-order entropy, which is given by the formula defined below.

If we have n independent symbols whose probabilities of choice are P_i, then we define the *zero-order entropy* as follows:

$$E = -\sum_{0 \prec i \preceq n} P_i \log(P_i)$$

3.1 Lines partitioning schemes

Here the image will be divided into lines, each line is of size 256. Afterwards, every line will be divided into a *Vector-Line* type. Denote, V-L_i[x] a Vector-Line of size x, the V-L_i types are:

$$\text{V-L}_i[2^i], \text{ such that } 1 \leq i \leq 8.$$

Figure 1 shows the main steps using lines partitioning schemes:

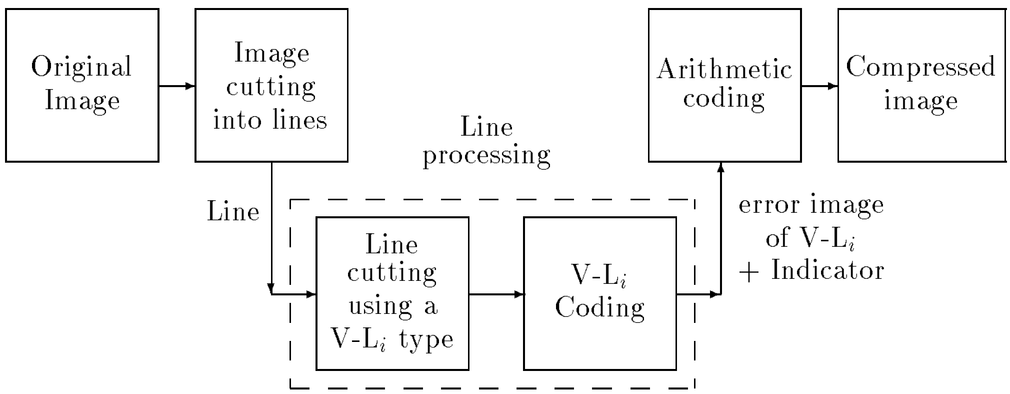

Figure 1: Lines partitioning schemes

3.1.1 V-L Encoding

Each line of the original image, is divided using a type of V-L_i. For example if i=7, V-L_7, we divide a line of the original image into two vectors of size 128 each. Let V-L_7[128] a vector of V-L_7 type, find below the main steps to encode this vector:

1. processing of V-L_7[128] using the eight predictors ;
2. calculation of V-L_7[128] error image and of the zero-order entropy of error image for each predictor ;
3. selection of the best among the eight predictors, the one which provides the lower zero-order entropy of V-L_7[128] error image ;
4. the V-L_7[128] error image , calculated using the selected predictor, and the indicator[1] which is used to indicate the selected predictor are processed with zero-order arithmetic coding [BCW90, R76].

[1] The indicator is an integer from 0 to 7, of the eight predictors.

To provide the best way of searching for the best predictor, we need to process the image in steps. First, we process the image to detect the magnitude and the orientation of edges in the input image. Secondly, according to the output of the first step, we choose the best predictor and the best block size.

3.2 Blocks partitioning schemes

Here, we divide the image into a *Vector-Block[M][N]* type where M is the number of lines, and N is the number of columns. Next, we process every Vector-Block by the eight predictors and select the predictor that provides the lower zero-order entropy. We use *V-B* to denote a Vector-Block. We describe the V-B types and their length as follows:

$$\begin{cases} \text{V-B}_i[2^i][2^i] & \text{such that } 2 \le i \le 7 \\ \text{V-B}_i[2^{i-1}][2^i] & \text{such that } i = 8. \end{cases}$$

To encode a V-B, we follow the same steps as the V-L encoding.

4 Results

We tested the presented algorithms on eight images. All the images are of size 256×256 and have 256 intensity levels. The images have been extracted from the university of Southern California and Nebraska-Lincoln Database. These images are part of a standard test set used by the image compression research community. The gain of compression is computed in the following way:

$$\%\ \text{gain} = \frac{R_o - R_e}{R_o} \times 100$$

where R_o is the size of original image and R_e is the size of compressed image. Table 3 contains the gains of compression using the following predictors: Pr_{rd} and Pr_s, Pr_{p3}, Pr_{p2} and trivial predictor (the original image).

Image	*Trivial*	Pr_{rd}	Pr_s	Pr_{p2}	Pr_{p3}
USG-Girl	21.0%	31.6%	29.6%	36.0%	38.2%
Girl	21.7%	41.0%	32.6%	39.1%	39.2%
Lady	36.3%	45.6%	42.2%	49.1%	49.9%
House	21.8%	33.8%	35.5%	40.7%	42.7%
USC-Couple	26.7%	36.1%	34.3%	41.6%	45.1
Tree	10.6%	22.8%	22.5%	27.9%	30.4%
Satellite	09.1%	15.1%	17.0%	22.0%	25.3%
X-Ray	**34.7%**	31.2%	15.9%	17.9%	19.0%

Table 3: Gain obtained using trivial and Harrison predictors.

Image	$J1$	$J2$	$J3$	$J4$	$J5$	$J6$	$J7$
USG-Girl	35.4%	35.9%	31.7%	34.5%	37.2%	36.8%	**38.7**%
Girl	44.8%	**46.1**%	41.3%	39.5%	40.1%	39.3%	42.5%
Lady	47.6%	**51.3**%	45.7%	46.8%	50.4%	48.2%	**51.2**%
House	37.2%	41.9%	33.6%	41.4%	**43.4**%	41.5%	42.1%
USC-Couple	43.1%	41.3%	35.7%	43.5%	44.2%	**45.3**%	44.2%
Tree	25.4%	29.8%	24.0%	26.8%	30.4%	28.5%	**30.9**%
Satellite	19.4%	22.3%	17.5%	21.8%	24.7%	23.5%	**25.4**%
X-Ray	**34.5**%	34.3%	31.2%	24.6%	20.0%	20.1%	23.3%

Table 4: Gain obtained using JPEG predictors.

Image	Pr_1	Pr_2	Pr_3	Pr_4	Pr_5	Pr_6	Pr_7	Pr_8
USG-G	35.4%	**38.3**%	37.5%	36.7%	37.1%	36.3%	37.8%	37.0%
Girl	39.6%	41.3%	**47.8**%	39.7%	40.1%	47.0%	39.9%	39.4%
Lady	50.4%	**51.6**%	50.7%	50.4%	51.2%	50.4%	50.7%	50.1%
House	40.7%	**42.9**%	41.2%	39.6%	41.6%	40.0%	39.7%	38.6%
USC-C	40.3%	**43.7**%	43.5%	41.4%	42.0%	42.0%	42.4%	40.9%
Tree	29.7%	**31.6**%	28.6%	29.1%	30.7%	27.4%	28.6%	27.4%
Satellite	22.8%	**25.6**%	22.9%	23.2%	24.4%	21.6%	22.9%	21.1%
X-Ray	20.0%	21.3%	**36.7**%	20.6%	20.3%	36.0%	20.1%	20.0%

Table 5: Gains obtained using our predictors.

Image	V-L_1	V-L_2	V-L_3	V-L_4	V-L_5	V-L_6	V-L_7	V-L_8
USG-G	27.8%	29.5%	33.5%	36.1%	37.3%	37.8%	37.9%	**38.1**%
Girl	39.3%	40.5%	43.3%	45.7%	47.6%	**48.3**%	**48.3**%	48.0%
Lady	41.1%	42.6%	46.3%	49.2%	51.3%	52.2%	52.4%	**52.6**%
House	31.7%	33.7%	38.3%	41.2%	43.0%	43.9%	43.9%	**44.1**%
USC-C	33.4%	37.2%	41.9%	44.5%	45.8%	**46.0**%	45.8%	45.7%
Tree	22.4%	22.5%	25.7%	28.6%	30.4%	**31.2**%	**31.2**%	31.1%
Satellite	15.7%	15.9%	19.3%	22.1%	23.9%	25.0%	25.5%	**25.7**%
X-Ray	26.9%	25.6%	28.9%	32.8%	35.7%	36.3%	36.4%	**36.5**%

Table 6: Gains obtained using our lines partitioning schemes.

Image	V-B_1	V-B_2	V-B_3	V-B_4	V-B_5	V-B_6	V-B_7	V-B_8
USG-G	18.4%	30.2%	36.1%	37.8%	38.0%	38.5%	38.6%	**38.7%**
Girl	27.0%	39.9%	46.0%	47.8%	**47.9%**	47.8%	47.8%	47.8%
Lady	30.9%	43.8%	51.1%	53.0%	**53.2%**	52.4%	50.9%	51.6%
House	20.1%	34.2%	42.1%	44.7%	**45.2%**	44.1%	42.8%	43.4%
USC-C	27.9%	39.1%	44.9%	**46.4%**	46.2%	46.3%	45.9%	46.0%
Tree	11.0%	22.5%	29.2%	31.1%	31.3%	31.0%	31.2%	**31.6%**
Satellite	03.9%	15.4%	22.1%	24.7%	25.3%	25.5%	**25.8%**	**25.8%**
X-Ray	21.9%	27.9%	34.6%	36.0%	36.3%	36.6%	36.6%	**36.7%**

Table 7: Gains obtained using our blocks partitioning schemes.

Tables 3, 4 and 5 contain the gains of compression using JPEG, Harrison and our predictors. If we compare these tables we notice that the gains in table 5 are often higher than those in tables 3 and 4. This means that the predictions that we propose are often better than those proposed by Harrison and Wallace. Table 6 and 7 contain the gains of compression using lines partitioning schemes and blocks partitioning schemes that we propose. The results in the two tables are clearly better than those in table 3, 4 and 5.

Conclusions

In this paper, we presented two techniques of lossless image compression based on predictors. If we compare our performances with the existing algorithms based on the refinement of pixels and specialized in lossless image compression, we obtain higher results. The algorithms that perform the best, are Pr_2 and Pr_3, which provide the best prediction. The optimal prediction scheme techniques obtains the best results.

References

[BCW90] Bell, T.C., Cleary, J.C. and Witten, I.H.: Text Compression. Advanced Reference Series. Prentice-Hall, Englewood Cilffs, New Jersey, 1990.

[H52] Harrison, C.W.: Experiments with linear prediction in television. Bell System Tech. J., 31, pages 764–783, July, 1952.

[Hu52] Huffman, D.A.: A method for the construction on minimum redundancy codes. Proc. IRE, 40, pages 1098–1101, 1952.

[R76] Rissanen, J.J.: Kraft inequality and Arithmetic coding. IBM Journal of Research and Development, 20, pages 198–203, May, 1976.

[TLR85] Todd, S., Langdon, G.G. and Rissanen, J.: Parameter reduction and context selection for compression of gray scale images. IBM J. Res. Develop., 29, 2, pages 188–193, March 1985.

[W91] Wallace, G.K.: The JPEG still picture compression standard. Communications of the ACM, 34, 4, pages 31–44, 1991.