IMPROVING POLA TECHNIQUE FOR IMAGE DATA COMPRESSION

Victor-Emil NEAGOE* and Dorin COMANICIU*

*Department of Electronics and Telecommunications
Polytechnical University of Bucharest, Bucharest 16, Romania 77206**

RÉSUMÉ

Dans cet article on présente une méthode nouvelle pour le codage d'images dénommée Ordonation Prédictive et Approximation Linéaire Amélioré (en anglais: *Predictive Ordering and Linear Approximation*, **I-POLA**). En considérant la ligne antérieure reconstruite de l'image, les pixels de la ligne actuelle sont placés dans un ordre décroissant d'amplitudes. La ligne ordonnée est segmentée et la ligne de régression est estimée pour chaque segment. Puis on utilise les raffinements suivants: quantification vectorielle des différences verticales, détection et suppression des suracroissements ainsi que codage de type bloc pour l'image des positions des erreurs de suracroissement. Les résultats de simulation montrent une bonne qualité d'images reconstruites avec un taux d'environ 0,5 bits/pixel. En évaluant une borne supérieure de la fidélité du codage pour la méthode proposée, on a obtenu des résultats très prometteurs.

ABSTRACT

This paper presents a new method for image coding called *Improved Predictive Ordering and Linear Approximation* (**I-POLA**). By taking as reference the previous received scan line, the pixels of the present line are placed in a decreasing order of amplitudes. The ordered line is segmented and the regression line is estimated for every segment. We further use the following refinements: vertical difference vector quantization, detection and suppression of overshoots, as well as block coding of overshoot error position image . Simulation results are given with good image quality at a low bit rate of about 0.5 bits/pixel. The evaluation of an upper bound of coding fidelity of the proposed method leads to very promising results.

1. INTRODUCTION

This paper presents an improvement of **POLA** (*Predictive Ordering and Linear Approximation*) for image coding proposed by Neagoe [1]. We called this new variant **I-POLA** (Improved **POLA**) or more explicitly **PO-S-LA-VDVQ-OS-BCOP**. It is obtained by adding to **POLA** the new processing stages of: (i) Segmentation (**S**); (ii) Vertical Difference Vector Quantization (**VDVQ**), where the considered difference is evaluated between the vectors containing vertically adjacent linear approximation segments of the ordered lines ; (iii) Overshoot Suppression (**OS**); (iv) Block Coding of Overshoot Positions (**BCOP**).

Section 2 deals with the main features of the proposed **I-POLA** image coding method as well as with the description of the corresponding algorithm.

In Section 3 the results of computer simulation and concluding remarks are given.

**Correspondence to: Prof. V. E.Neagoe,
P.O. Box 16-37, Bucharest 16, Romania 77500
Fax: 40-1-3120188, Phone: 40-1-7714769

2. IMPROVED POLA

2.1. Main characteristics

The main ideas of the proposed method are the following:

(a) By taking as reference the previous received scan line, the pixels of the present line are placed in a decreasing order of amplitudes (Predictive Ordering = **PO**).

(b) The ordered line (containing N pixels) is divided in L segments (Segmentation = **S**).

(c) The regression line is estimated for every segment of the above division (Linear Approximation = **LA**).

(d) The difference vectors of the parameters characterising the regression lines of two corresponding vertically adjacent segments of the same order are quantized in $M=2^q$ clusters (Vertical Difference Vector Quantization = **VDVQ**).

(e) Computation of the error vectors (corresponding to each segment of the considered division) representing the differences between the initial predictively ordered pixel vectors and their corresponding regression lines.

(f) The detection and suppression of overshoots, namely those errors (elements of the above error vectors), whose absolute values exceed a certain threshold (Overshoot Suppression = **OS**).

(g) The overshoot error pattern image (having black pixels where overshoots are detected, and white pixels otherwise) is coded as a binary image using the block coding method (Block Coding of Overshoot Positions = **BCOP**). The above processing stage consists of the segmentation of overshoot position binary image into "n x n" pixel blocks, then an optimum variable length Huffman code is constructed, having a code table of size 2^{n^2}.

2.2. Description of algorithm

• Denote by x(i,k) the signal level associated to the element of coordinates (i,k), where i = 0,1,..., N-1 represents the scan line index and k = 0,1,...,N-1 is the pixel abscissa (from left to right). The original picture line is denoted by :

$$\mathbf{X}_i = (x(i,0), x(i,1),..., x(i,N-1))^T$$

and the reconstructed line is :

$$\hat{\mathbf{X}}_i = (\hat{x}(i,0), \hat{x}(i,1),...,\hat{x}(i,N-1))^T.$$

• The algorithm uses the vertical correlation between adjacent pixels by considering that the predictor of the pixel x(i+1,k) is the reconstructed pixel $\hat{x}(i,k)$ belonging to the adjacent line and having the same abscissa [1].

The **I-POLA** algorithm has the following steps:

Step 1: Store and send to the receiver the first picture line $\mathbf{X}_0$, as a reference.

Step 2: Order the first line $\mathbf{X}_0$, obtaining the vector

$$\mathbf{Y}_0 = (x(0,0_0), x(0,0_1),...,x(0,0_{N-1}))^T,$$

where 0_h represents the position in the original picture scan line of the element which has the position h in the ordered line. Denote **IR** to be the ordering index vector, i.e.,

$$\mathbf{IR} = (0_0\ 0_1 \ldots 0_{N-1})^T. \qquad \mathbf{(2.1)}$$

Step 3: i = 1

Step 4: Predictively order the i-th line $\mathbf{X}_i$, according to the vector **IR**, obtaining the vector $\mathbf{Y}_i$,

$$\mathbf{Y}_i = (x[i, (i-1)_0], x[i, (i-1)_1],..., x[i, (i-1)_N])^T = = (y_i(0), y_i(1),..., y_i(N-1))^T. \qquad \mathbf{(2.2)}$$

Step 5: Uniformly divide the ordered line $\mathbf{Y}_i$ into L segments of m = N/L pixels each of them, namely

$$\mathbf{Y}_i = \bigcup_{k=0}^{L-1} \mathbf{W}_{ik}, \qquad \mathbf{(2.3)}$$

where

$$\mathbf{W}_{ik} = (w_{ik}(0),...,w_{ik}(m-1))^T.$$

Step 6: Linearly approximate each segment $\mathbf{W}_{ik}$.

$$z_{ik}(j) = \hat{B}_{ik} + \frac{\hat{C}_{ik}-\hat{B}_{ik}}{m-1}\, j \qquad \mathbf{(2.4)}$$

(j = 0,1,...,m-1 ; i = 0,1,...,N-1; k = 0,1,...,L-1)

where $\hat{B}_{ik}$ and $\hat{C}_{ik}$ are the rounded-off parameters of the regression line segment (see [1]).

The vector of piece-wise linear approximation of the i-th picture line is

$$\mathbf{P}_i = (z_{i0}(0)...z_{i0}(m-1)...z_{i,L-1}(0)...z_{i,L-1}(m-1))^T = (p_i(0)...p_i(N-1))^T \qquad \mathbf{(2.5)}$$

Step 7: (a) Compute the overshoot error pattern vector:

$$\Gamma_i = (\gamma_i(0), \gamma_i(1),..., \gamma_i(N-1))^T, \qquad \mathbf{(2.6)}$$

where

$$\gamma_i(j) = \begin{cases} 1, \text{ for } | y_i(j) - p_i(j)| \geq \delta \\ 0, \text{ otherwise} \end{cases} \qquad \mathbf{(2.7)}$$

δ being the overshoot threshold.

(b) For every $\gamma_i(j) = 1$, compute the overshoot sign, namely

$$q_i(j) = \begin{cases} 1, \text{ for } y_i(j) - p_i(j) \geq \delta \\ 0, \text{ for } y_i(j) - p_i(j) \leq -\delta \end{cases} \qquad \mathbf{(2.8)}$$

(c) Compute and quantize the mean overshoot value of the line, namely

$$\sigma_i = 1/n_{\varepsilon i} \sum_{j \in S_i} |\varepsilon_{ij}|, \qquad \mathbf{(2.9)}$$

where $n_{\varepsilon i}$ is the number of overshoots detected for the i-th picture line, $\varepsilon_{ij} = y_i(j) - p_i(j)$, $S_i = \{ j \mid \gamma_{ij} = 1\}$; then

$$\hat{\sigma}_i = Q (\sigma_i) \qquad \mathbf{(2.10)}$$

where Q(x) means the quantization operator.

Step 8:

(a) Compute the differences between the vectors of parameters characterising the regression lines of two corresponding vertically adjacent segments of the same order:

$$\mathbf{D}_{ik} = ((\hat{B}_{ik}-\hat{B}_{i-1,k}), (\hat{C}_{ik}-\hat{C}_{i-1,k}))^T. \qquad \mathbf{(2.11)}$$

(i =1,2,...,N-1 ; k = 0,1,...,L-1).

(b) Quantize the vectors $\mathbf{D}_{ik}$ (see [3]) in M clusters. This requires a stage of learning to deduce the codebook. Then, we apply the rule of the nearest prototype to perform the vector quantization.

Step 9: Perform the overshoot correction:

$$\hat{y}_i(k) = \begin{cases} \hat{p}_i(k) + \hat{\sigma}_i, \text{ for } \gamma_{ij} = 1, q_{ij} = 1 \\ \hat{p}_i(k) - \hat{\sigma}_i, \text{ for } \gamma_{ij} = 1, q_{ij} = 0 \end{cases} \qquad \mathbf{(2.12)}$$

Step 10: Reorder the reconstructed line $\hat{\mathbf{Y}}_i$ to find the vector $\hat{\mathbf{X}}_i$, using the index vector **IR**. The element $\hat{y}_i(k)$ will have the position i_k in $\hat{\mathbf{X}}_i$.

Step 11: Order the last reconstructed line $\hat{\mathbf{X}}_i$ and deduce the new ordering index vector

$$\mathbf{IR} = (i_0\ i_1 \ldots i_{N-1})^T \qquad \mathbf{(2.13)}$$

where

$$\hat{x}(i,i_0) \geq \hat{x}(i,i_1) \geq \ldots \geq \hat{x}(i,i_{N-1}), \qquad \mathbf{(2.14)}$$

where i_h represents the position in the reconstructed picture scan line $\hat{\mathbf{X}}_i$ of that element which has the position h in the ordered line (see relation **(2.14)**).

Step 12: i = i+1

Step 13: If i < N-1, then go to Step 4.

Otherwise continue.

Step 14: Block code the overshoot error pattern image $\{ \Gamma_0, \Gamma_1,...,\Gamma_{N-1} \}$, where Γ_i is given by relations **(2.6)** and **(2.7)**.

Step 15: Stop.

3. COMPUTER SIMULATION RESULTS AND CONCLUDING REMARKS

The parameters of the experiments are:

- the image size: **256 x 256** or **512 x 512**.
- the number of gray levels: **256**, i.e., $\mathbf{x(i,j)} \in \{ \mathbf{0, 1,..., 255} \}$.

- the number of pixels per segment: **m**∈{ **16; 32; 64** }.
- the number of clusters for vector quantization: **M**∈{ **4; 8**}
- the block size for segmentation of overshoot position error image: **3 x 3**, implying a size of **512** pattern messages to be coded by Huffman technique.

The coding performance measures are:

a) the average bit rate per pixel (**R**)

b) the signal to quantization noise ratio

$$\mathbf{SQNR} = 20 \log_{10} \frac{255}{e_{rms}} \text{ (dB)} \qquad \mathbf{(3.1)}$$

where

$$\mathbf{e}_{rms} = \sqrt{\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} (x(i,k) - \hat{x}(i,k))^2} \quad . \quad \mathbf{(3.2)}$$

We have used a set of eight pictures for estimation (to obtain the cluster prototypes for vector quantization and respectively to deduce the pattern statistics for the Huffman block coding of overshoot position binary image) and respectively an another image for testing.

We have also experimented the case where the estimation and testing are performed on the same image.

The experimental coding performances are given in **Tables 1, 2** and **3** as well as in **Fig. 1 (a)** ,**(b)** and **(c)**.

We point out the following **concluding remarks**:

1. To compute the average bit rate we have to take into account that we must send to the receiver the following information:

- first line of the original image, $\mathbf{X}_0$, (as a reference).
- the codewords corresponding to the quantization of the difference vectors $\mathbf{D}_{ik}$ containing the differences of the parameters characterising the regression lines of the corresponding vertically adjacent segments.

  (i = 1, 2,..., N-1; k = 0, 1,...,L-1)
- the Huffman block coded binary image of overshoot positions { $\Gamma_0, \Gamma_1, \ldots, \Gamma_{N-1}$ }.
- the sequence of overshoot signs {$q_i(j)$ },

  (i = 1, 2, ..., N-1; j∈S_i)
- the quantization of overshoot absolute value for each line { $\hat{\sigma}_i$, i = 1, 2,..., N-1 }.

2. We obtain better performances for an image size of **512 x 512** pixels than for a size of **256 x 256** pixels, due to the increasing of correlation between neighbouring pixels when the image size increases.

3. For the maximum image size (N = 512) a trade-off between the bit rate and coding fidelity (evaluated by SQNR) leads to best performances for M = 8, δ = 20 and for either m = 32 (with R = 0.56 bits/pel and SQNR = 29.25 dB) or respectively m = 64 (with R = 0.53 bits/pel and SQNR = 28.95 dB).

4. The proposed **I-POLA** has clearly better performances than those of original **POLA** [1], namely for M = 8 , δ = 20, m=32, the bit rate is reduced with 46% (from 1.04 bits/pel to 0.56 bit/pel) while the signal -to- quantization noise increases with 7.93 % representing 2.15 dB (from 27.10dB to 29.25dB).

5. The segmentation and the application of linear approximation on each segment (instead on the whole line as in original **POLA)** implies the increasing of the image approximation accuracy. Moreover, since the segmentation is applied after predictive ordering it has no negative effect on the image quality. The pixels belonging to a certain segment will be spread at various places (some of them belonging to other segments) in the reconstructed line; therefore, this pre-processing stage of whole line ordering has a prophylactic function to avoid the undesired consequences of segmentation.

6. The Predictive Ordering (**PO**) is an exciting idea equivalent to build a "predictive histogram" of the line based on the vertical correlation between adjacent pixels, in order to improve the efficiency of a subsequent Linear Approximation (**LA**) as a particular case of waweform coding. This ordering concentrates the signal energy into low frequency regions.

7. As an upper bound of coding fidelity for *Predictive Ordering and Linear Approximation* (**POLA**), we considered the *Exact Ordering and Linear Approximation* (**EOLA**) consisting of the following stages: (i) exact ordering of picture line; (ii) segmentation of the ordered line in L regions of m pixels each of them (mL = N); (iii) linear approximation of the each segment; (iv) image reconstruction. The experimental results are given in **Table 3.** They open us a promising window: if the predictive ordering tends close to exact ordering, the signal-to-quantization noise ratio (for example, at m=32) would tend to 45.18 dB !

TABLE 1.
EXPERIMENTAL I-POLA CODING PERFORMANCES
(Test picture: **LENA of 512 x 512** pixels; Vpp = 255)

R b/p \ SQNR dB		I-POLA δ =16	I-POLA δ =20	I-POLA δ =24	POLA
m= 16	M=4	0.73 / 29.35	0.6 / 28.62	0.51 / 27.73	1.06 / 27.10
	M=8	0.77 / 30.14	0.64 / 29.25	0.56 / 28.38	
m= 32	M=4	0.67 / 29.86	0.54 / 28.89	0.46 / 27.95	
	M=8	0.69 / 30.20	0.56 / 29.25	0.48 / 28.42	
m= 64	M=4	0.65 / 29.93	0.51 / 28.84	0.43 / 27.90	
	M=8	0.67 / 29.92	0.53 / 28.95	0.44 / 27.96	

TABLE 2
EXPERIMENTAL I-POLA CODING PERFORMANCES
(Image size: **256 x 256** pixels; Vpp = 255; m = 32; M = 8; δ=20)

R b/p / SQNR	I-POLA	POLA
LENA	0.67 / 28.41 dB	1.09 / 26.09 dB
MOUNT	0.75 / 27.47 dB	1.09 / 26.34 dB

TABLE 3.
EXPERIMENTAL EOLA CODING FIDELITY MEASURE AS AN UPPER BOUND OF POLA PERFORMANCE
(Test picture: **LENA** of **512 x 512** pixels; Vpp = 255 ; No refinements of VDVQ, OS or BCOP)

m	4	8	16	32	64	128	256	512
SQNR dB	53.4	50.5	47.8	45.2	41.7	37.4	33.5	30.8

(a)

(b)

(c)

Fig.1.
ORIGINAL AND RECONSTRUCTED CODED PICTURES FOR LENA AS A TEST IMAGE
(a) Original (N = 512; Vpp = 255).
(b) I-POLA (m = 32; M = 8; δ = 20; R_B = 0.56 bits/pixel ; SQNR = 29.25 dB).
(c) EOLA (m =32; SQNR = 45.2 dB).

REFERENCES

[1] **V.E.Neagoe**, "Predictive Ordering and Linear Approximation for Image Data Compression", IEEE Trans. Commun. vol. COM-36, pp.1179-1182, Oct.1988.

[2] **V.E.Neagoe**, "Predictive Ordering Technique and Feedback Transform Coding for Data Compression of Still Pictures", IEEE Trans. Commun. vol. COM-40, pp.385-396, Febr.1992.

[3] **A.N.Netravali**, Digital Pictures (Representation and Compression), NewYork, Plenum Press, 1988.