Low complexity Lossless Compression for Multicolored Images

Saurabh Shiral
Maharashtra Academy Of Engineering, Alandi
Pune, India
Email: sourabh.shiral@gmail.com

Nitin Paighowal
Maharashtra Academy Of Engineering, Alandi
Pune, India
Email: nitinpango@gmail.com

Bhumika Danej
Maharashtra Academy Of Engineering, Alandi
Pune, India
Email: bhumika.danej@gmail.com

Sumant Kumar
Maharashtra Academy Of Engineering, Alandi
Pune, India
Email: sumantk25@gmail.com

***Abstract*— This paper presents a method for compressing the images which is based on the LOCO-I (low complexity lossless compression for images) algorithm. We try to reduce complexity of the original algorithm. From experimental verification it can be found out that the proposed algorithm is better than the rice compression by almost 15 percent.**

***Keywords-** A-LOCO; LOCO- I; JPEG- LS; context; compression.*

I. INTRODUCTION *(HEADING 1)*

Image compression is defined as minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. The time required for images to be sent over the Internet or downloaded from Web pages is also reduced.

LOCO-I is a class of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. The lossy data compression, allows only an approximation of the original data to be reconstructed, but provides better compression rates. Lossless data compression is used in many applications. For example, it is used in the ZIP file format and in the UNIX tool gzip

II. LOSSLESS MODE OF OPERATION

The lossless mode employs differential pulse code modulation (DPCM) in which the current pixel value is estimated from the neighboring samples that are already coded in the image.

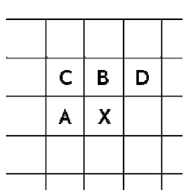

Figure 1. Three neighboring samples around the sample to be predicted

For Example, the predictor combines three to five neighboring samples. Here we consider samples at A, B, and C in order to predict the sample value at the position labeled by X.

Table 1 Predictors shown in the table below can be used to estimate the sample located at X.

Selection-value	Prediction
0	No prediction
1	A
2	B
3	C
4	A + B – C
5	A + (B – C)/2
6	B + (A – C)/2
7	(A + B)/2

The three neighboring samples must be already predicted samples. Once we predict all the samples, the differences between the samples can be obtained and entropy-coded in a lossless fashion using Huffman coding or arithmetic coding.

III. JPEG-LS

JPEG-LS is a simple algorithm which consists of two independent and distinct stages called modeling and encoding. JPEG-LS provide a low-complexity lossless and near-lossless image compression standard that offers better compression efficiency than lossless JPEG.

The Huffman coding-based JPEG lossless standard and other standards were limited in their compression performance, so it was developed. The core of JPEG-LS is based on the LOCO-I algorithm, that relies on prediction, residual modeling and context-based coding of the residuals.

Compression for JPEG-LS is generally much faster than JPEG 2000 and much better than the original lossless JPEG standard.

IV. LOCO-I ALGORITHM

Two important steps before encoding which needs to be implemented are: a) prediction and b) error modeling.

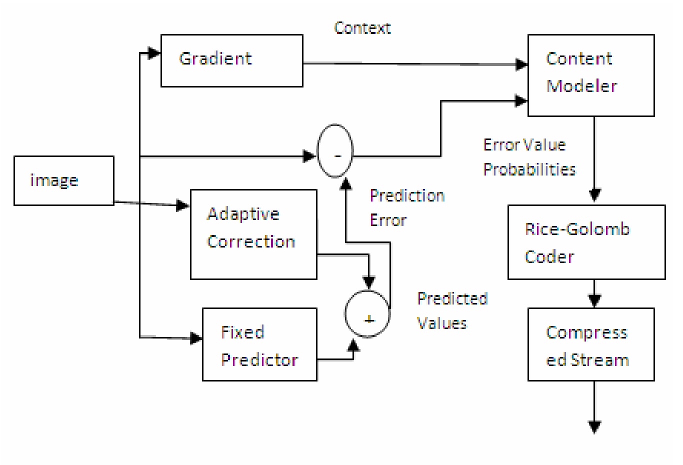

Figure. 2 Pictorial representation of the LOCO-I algorithm

A. Decorrelation/prediction

In the LOCO-I algorithm, horizontal or vertical edges are detected by examining the neighboring pixels of the current

$$X = \begin{cases} \min(A<B) \text{ if } C>=\max(A,B) \\ \max(A,B) \text{ if } C<=\min(A,B) \\ A+B-C \text{ otherwise} \end{cases} \qquad (1)$$

pixel X as illustrated in Fig.1. For vertical edge, pixel B is used whereas for horizontal edge pixel at A is used. This predictor is called the Median Edge Detection (MED) predictor or LOCO-I predictor. The pixel X is predicted by the predictor according to the above formula.

The three simple predictors are selected according to the conditions: (1) B is picked in cases when a vertical edge is present left of X, (2) A in cases when horizontal edge is present above X, or (3) A + B – C if no edge is detected.

B. Context modeling

The JPEG-LS uses the current sample $\bar{e}(C)$ to estimates the conditional expectations of the prediction errors $E\{\, e \mid Ctx \,\}$ within each context *Ctx*. Context modeling of the prediction error helps to understand the higher order structures like texture patterns and local activity i.e. smoothness, edginess of the image. Contexts can be determined by calculating the differences of the neighboring samples which represents the local gradient:

$$\begin{aligned} g1 &= D - B \\ g2 &= B - C \\ g3 &= C - A \end{aligned} \qquad (2)$$

The local gradients calculated above represents the varying degree of properties like smoothness and edginess of the neighboring samples. It can be observed that these differences are closely related to the statistical behavior of prediction errors. The differences found in the above equation are then quantized into roughly equiprobable and connected regions. For JPEG-LS, these differences namely, g1, g2, and g3 are quantized into 9 regions and the region are indexed from -4 to 4. Quantization helps to maximize the mutual information between the current pixel and its context so as to capture the high-order dependencies. The contexts can be calculated based on the assumption that

$$P\,(e \mid Ctx = [q_1, q_2, q_3]) = P(-e \mid Ctx = [-q_1, -q_2, -q_3]) \qquad (3)$$

Merging contexts of both positive and negative signs results in the $((2 \text{ X } 4 + 1)^3 + 1) / 2 = 365$ contexts. A bias value can be estimated by dividing the cumulative prediction errors within each context by a count of context occurrences. The LOCO-I algorithm modifies and improves the procedure so as to reduce the total number of subtractions and additions. LOCO-I prefers the division-free bias computation procedure.

C. Coding corrected prediction residuals

The regular mode of JPEG-LS standard uses the Golomb-Rice codes which are used to encode non-negative run lengths. Special case of the encoding method with the optimal encoding value 2^k enables simpler encoding procedures.

D. Run length coding in uniform areas

It is observed that Golomb-Rice codes are quite inefficient for encoding low entropy distributions due to the fact that coding rate is at least one bit per symbol, this causes a large redundancy because the smooth regions in an image can be encoded at the encoding rate less than 1 bit per symbol. To avoid the excess code length over the entropy, instead of coding individual symbols we can use alphabet extension method which codes blocks of symbols. This distributes the excess code length over many symbols. The method described is the “run” mode of JPEG-LS and it can be executed if flat or smooth context regions which are characterized by zero gradients are detected. After encoding the total run of certain length encoder returns to the “regular” mode.

V. ADVANCEMENTS IN LOW COMPLEXITY LOSSLESS COMPRESSION (A-LOCO) ALGORITHM USING LOCO- I

The A-LOCO algorithm described in the following paragraphs is based on the LOCO-I algorithm. The input to the algorithm is a rectangular image with 8-bit pixel values (the pixel values are within the range 0 to 255). The output is the compressed image which is a sequence of bits from which original image can be reconstructed.

We use 'wd' to represent the image width and 'ht' to represent the image height. (x, y) represents any pixel in general with x in the range [O, wd-l] and y in the range [O, ht-l]. We assume that: (0, 0) corresponds to the upper left comer of the image.

The A-LOCO algorithm derives its core logic from predictive compression. During processing, the pixels of the image are scanned in raster scan order. For the raster scan, y is incremented through the range [O, ht-l], and for each y value in this range, x is incremented through the range [O, wd-l]. We put the first two pixels, with the coordinates (0, 0) and (1, 0), in output bit stream without any encoding. For all other pixels of the image, the processing that A-LOCO algorithm does can be conceptually split into four steps:

1. The pixel into consideration is classified into one of several contexts based on the values of (usually 5) previously encoded pixels.

2. Calculate the approximate pixel value from (usually 3) previously encoded pixels, and add a correction to this approximate value (called the bias), which depends on the context.

3. Calculate the difference between the approximate and the actual pixel value and map it to a non-negative integer, and then encode this integer using Golomb's variable length codes

4. Take the feedback from the new pixel value update the statistics for the context.

VI. CONCLUSION

As seen the LOCO-I algorithm uses the edge detector and the predictor (that predicts the current pixel) as separate modules resulting in the increased complexity. It is also seen that the predicted values of the pixels by LOCO-I algorithm are actually quite far from the actual values. In our proposed approach of A-LOCO algorithm we try to reduce the implementation complexity of the algorithm by combining the edge detector and the predictor into a single module. Also we update the contexts regularly by taking a feedback from previously encoded pixels so as to accurately predict the future pixels. We expect that A-LOCO algorithm will achieve a compression more by 15 percent as compared to Rice Compression. Overall, the A-LOCO algorithm will represent significant progress towards producing lossless image-compression with improved compression performance. We will be able to compress images up to 56% of the original size. Hence, through A-LOCO we aim to provide an improved way of lossless image compression than the currently available solution.

REFERENCES

[1] Liu Zheng-lin, Qian Ying, Yang Li-ying, Bo Yu, Li Hui, "An Improved Lossless Image Compression Algorithm LOCO-R", 201O International Conference on Computer Design and Applications (ICCDA 2010)

[2] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based, lossless image compression algorithm," in *Proc. 1996 Data Compression Conference*, Snowbird, UT, Mar. 1996,pp. 140–149.

[3] M. J. The paperinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Lossless ImageCompression Algorithm: Principles and Standardization into JPEG-LS", IEEETransactions on Image Processing, vol. 9, no. 8, pp. \309-\324, August 2000.

[4] Data compression – The complete reference by David Salomon.

[5] J. Rissanen, \Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629{636, July 1984.

[6] N. D. Memon and K. Sayood, \Lossless image compression: A comparative study," in *Proc. SPIE (Still-Image Compression)*, vol. 2418, pp. 8{20, Feb. 1995.

[7] J. Rissanen and G. G. Langdon, Jr., \Universal modeling and coding," *IEEE Trans.Inform. Theory*, vol. IT-27, pp. 12{23, Jan. 1981.

[8] S. W. Golomb, "Run-Length Encodings", IEEE Transactions on Information Theory, vol. IT-12, no. 3, pp. 399-401, July 1996

[9] N. Merhav, G. Seroussi, and M. J.Weinberger, \Modeling and low-complexity adaptive coding for image prediction residuals." To be presented at the 1996 Int'l Conference on Image Processing, Lausanne, Sept. 1996.

[10] X.Wu and N. Memon, "Context-based adaptive lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.

[11]W. Pennebaker and J. Mitchell, *JPEG: Still Image Data Compression Standard*. London, U.K.: Chapman & Hall, 1992.

[12] M. Kwon *et al.*, "Lossless image coder with context-based minimizing MSE prediction and entropy coding," in *Proc. Int. Symp. Circuits Systems*, vol. 4, 1999, pp. 4

[13] N. Memon, V. Sippy, and X. Wu, \A comparison of prediction schemes proposed for a new lossless image compression standard", in IEEE International Sym- posium on Circuits and Systems, vol. 2, pp. 309{312, May1995.

[14] A. Said and W. A. Pearlman, \An image multiresolution representation for lossless and lossy image compression", IEEE Trans. Image Process., vol. 5, pp. 1303{1310, September 1996.