ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Algorithm for Image Compression

J.Swapna
B.Tech,
Dept of ECE,
MLR Institute of Technology, Hyderabad, T.S, India.

Md.Musharaf
B.Tech,
Dept of ECE,
MLR Institute of Technology, Hyderabad, T.S, India.

R.Bhargav
B.Tech,
Dept of ECE,
MLR Institute of Technology,
Hyderabad, T.S, India.

S.N.L. Prasanna
B.Tech,
Dept of ECE,
MLR Institute of Technology,
Hyderabad, T.S, India.

S.V.S.Prasad
Professor,
Department of ECE,
MLR Institute of Technology,
Hyderabad, T.S, India.

Abstract:

In recent years, the development and demand of multimedia product grows increasingly fast, contributing to insufficient bandwidth of network and storage of memory device. Therefore, the theory of data compression becomes more and more significant for reducing the data redundancy to save more hardware space and transmission bandwidth. In computer science and information theory, data compression or source coding is the process of encoding information using fewer bits or other information-bearing units than an unencoded representation.

Compression is useful because it helps reduce the consumption of expensive resources such as hard disk space or transmission bandwidth. In project, a spatial domain image compression algorithm is proposed. This algorithm is based on the prediction of each pixel from the previous ones. The errors of the predicted values are encoded using a set of logarithmically distributed possible values, which are called hops. The performance of our algorithm is remarkable when compared to an algorithm in [4] and shows much better quality, in terms of PSNR than an algorithm in [4].

Index Terms:

block matching algorithm, motion estimation, fast search algorithm.

1. INTRODUCTION:

There are three main concepts that set the limits for image compression techniques: image complexity [1], desired quality and computational cost. This paper presents logarithmical hopping encoding (LHE) algorithm, a computationally efficient algorithm for image compression. The proposed algorithm relies on Weber–Fechner law, which states that subjective sensation is proportional to the logarithm of the stimulus intensity [2]. LHE applies this law to prediction errors instead of the stimulus itself (in this case the original luminance and chrominance signals). More concretely, LHE estimates a luminance and chrominance prediction for each pixel (using the surrounding pixels) and then encodes the prediction error using a set of logarithmically distributed and dynamically adjusted values.

This procedure is performed in the space domain, avoiding the need of any costly transformation to the frequency domain, and therefore reducing the computational complexity. The remaining of this paper is organised as follows. Section 2 surveys the most relevant work related to the proposed algorithm. Section 3 describes the structure and the workflow of LHE. Section 4 shows the main results that have been obtained in the evaluation stage of the algorithm. Finally, Section 5 summarises the main contributions of this paper and outlines the future lines of work.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

2. RELATED WORK:

LHE can be defined as a spatial domain image compression algorithm. In the literature of image compression, spatial domain-based algorithms have been extensively studied. In [4], a spatial domain image compression algorithm is proposed. This algorithm encodes the difference between the minimum pixel value of an m × n pixel block and the current pixel. For each block, an 11 bits header is included in order to represent the minimum value of the block (8 bits) and the number of pixels required to encode the pixel difference with respect to the minimum (3 bits).

The authors of [5] present a modified approach to the previous algorithm where the final number of bits is reduced significantly by reducing the overhead bits. In [6], a variation of the previous algorithm that encodes the difference between adjacent pixels is proposed. In [7], a logarithmic function is used as a pre-processing stage for an image compression algorithm. This algorithm comprises four stages: logarithmic transform, neighboring difference, repeat reduction and Huffman encoding. In this paper, the logarithmic function is used to reduce the range of the difference between neighboring pixels.

LHE also has similarities to ADPCM (adaptive differential pulse-code modulation) [8]. ADPCM uses an adaptive predictor and an adaptive quantiser. The quantiser levels for a given pixel are generated by scaling the levels used for the previous pixel by a factor that depends on the reconstruction level used for the previous pixel. ADPCM dynamically adapts the quantiser step size to the input signal, and the set of possible unary codes is linearly distributed for each sample.

However, LHE unary codes are logarithmically distributed for each sample. This change in step distribution provides better results than ADPCM.LOCO-I [9] is the algorithm at the core of the ISO/ITU standard for lossless and near-lossless compression of continuous-tone images, JPEG-LS.LOCO-I uses the prediction of samples based on a finite subset of available past data and the context modeling of the prediction error. The purpose of this context modeling is to exploit high order structures, for example, texture patterns, by analyzing the level of activities, such as smoothness and edginess of the neighboring samples. This context modeling provides a probabilistic model for the prediction residual (or error signal), which can be efficiently used in combination with Golomb-Rice codes. LHE uses a similar approach, but focused on lossy image compression and taking into account the logarithmical nature of human perception.

3. LHE: BASIC ALGORITHM

The basic algorithm of LHE is based on the prediction of colour space values (e.g. YUV) of each pixel from the previous ones. The errors of the predicted values are encoded using a set of logarithmically distributed possible values of luminance and chrominance, which are called hops. The main blocks of the basic algorithm of LHE, grouped as LHE quantiser, are depicted in Fig. 1. Detailed information about these blocks is provided in the following subsections.

Pixel Prediction

LHE uses the YUV colour space to represent the pixel information. YUV is defined in terms of one luminance value (Y) and two chrominance components (UV). The human eye has fairly little spatial sensitivity to colour, thus luminance component has far more impact on the image detail than the chrominance. For a given pixel,

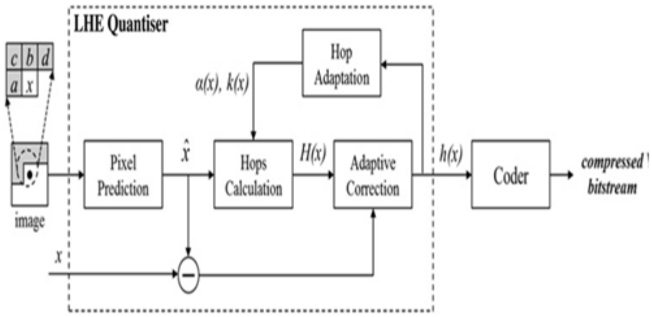

Fig. 1 LHE basic: block diagram

LHE predicts each colour component (Y, U or V) as the average of the colour component of the top (b) and left (a) pixels, as in the following equation

$$\hat{x} = \left(\frac{a+b}{2}\right)$$

The prediction of each colour component should be computed individually. Therefore, for a given pixel x, ˆx represents the predicted value of the luminance (Y) or chrominance (UV). In the remaining of this section, luminance will be used as an example of the three colour components. As the predictions of the pixels depend on the previous ones, the first pixel of the image is not processed by the LHE quantiser. Thus, its colour components are included uncompressed in order to allow the decoder to process subsequent pixels.

Logarithmical Hops

As aforementioned, LHE encodes the errors of the predicted color components from a set of possible logarithmically distributed values (called hops) for each pixel, $H(x) = \{h_{-N}, h_{-(N+1)}, ..., h_{-1}, h_0, h_1, ..., h_{N-1}, h_N\}$. The null hop h_0 means that the error associated with the predicted colour component is lower than the one achieved by a different hop value. The smallest positive and negative hops, h_1 and h_{-1}, are not logarithmically assigned. LHE algorithm adjusts automatically, within a certain range, the value of the hops h_1 and h_{-1} for each pixel depending on the previously encoded pixel through the parameter α(x), which is described in Section 3.4 'hop adaptation'. The first time LHE is executed, an initial fixed value for the parameter α is used, for example, α = 8.The image compression rate of LHE depends on the number of hops is considered (2N + 1), the smallest non-null hops (h_1 and h_{-1}, defined by the parameter α(x)) and the parameter k(x). The higher the cardinality of H, the lower the compression rate of LHE. The parameters α(x) and k(x) are responsible for the compactness of the set of hops H(x) for a given pixel. The following equation details the different values of the set of hops H(x) for a given pixel

$$h_i = \begin{cases} 0, & if i = 0 \\ a(x), & if i = 1 \\ -a(x), & if i = -1 \\ h_{i-1} * \dfrac{255 - \dfrac{\hat{x}}{k(x)}}{\dfrac{1}{k(x)}}, & if i > 1 \\ h_{i+1} * \dfrac{\left(\dfrac{\hat{x}}{k(x)}\right)}{\dfrac{1}{k(x)}}, & if i < 1 \end{cases}$$

The image compression rate of LHE depends on the number of hops is considered (2N + 1), the smallest non-null hops (h_1 and h_{-1}, defined by the parameter α(x)) and the parameter k(x). The higher the cardinality of H, the lower the compression rate of LHE. The parameters α(x) and k(x) are responsible for the compactness of the set of hops H(x) for a given pixel.Different values of k(x) allow expanding and shrinking the range covered by the set of logarithmical hops. In image areas where there are high component fluctuations, a low value of k(x) covers the maximum range and provides better results. On the other hand, in soft detailed areas, a high value of k(x) shrinks the set of hops, gaining more accuracy for small changes on colour component. The value of k(x) is determined locally, at each pixel, taking into account the set of surrounding hops

$$k(x) = f(h(a), h(b), h(c), h(d))$$

For each combination of hops corresponding to the pixels a, b, c and d (pixel positions are shown in Fig. 1), there is an optimal value for k (x), which minimizes the error when a new hop for x is chosen. Although a formula could be defined, a pre-calculated table of optimal k(x) values can be generated testing over all pixels from all images from an image database, and

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

therefore setting the best values for any type of image. This strategy avoids deducing the 'best logic' for the formula and therefore simplifies the problem.

Adaptive Correction

The adaptive correction module takes into account two parameters: the set of possible hops H(x), computed in the previous module, and the error associated to the predicted colour component, *e*.

$$e = x - \hat{x}$$

The output of this module is the hop h(x) from the set H(x), that is, h (x) ∈ H(x) that is closer to the above described error. In other words, the hop h(x) is the quantized error made by the LHE algorithm in the colour component prediction ˆx. This hop h(x) is the output of the LHE quantiser

h(x) = arg_{h_i} min(|h_i− e|), h_i ∈ H(x)

In the particular case that there are two different hops with the same distance to the error *e*, the hop with the smaller value is chosen. The reason behind this approach is that in statistical compression of images, smaller codes are assigned to small hops (see Section 'coder')

If ∃(hj ,hk) E H(x) | |hj− e|= |hk− e| ⇒ h(x)

= hi |i = min (|j|, |k|)

Hop Adaptation

Once the quantized error of the actual pixel is assigned, the hop adaptation module updates the parameter α(x), which has the same absolute colour component value as the smallest non-null hops h_1 and h_{-1}, for the next pixel. The parameter α(x) varies within a certain fixed range [αmin, αmax], for example, [4, 8]. According to (2) the parameter α(x) is used for computing the set of possible hops H of the next pixel. As aforementioned, an initial start value for the parameter α is fixed for the first pixel encoded by LHE, for example, α = 8. The adjustment of the value α is based on the following rules:

- If the assigned hops of two consecutive pixels {h(x–1), h(x)} are small, that is, they are either null hops h0 or the smallest non-null hops {h_{-1}, h_1}, then the updated value α(x) becomes one unit smaller than the smallest positive non-null hop h1, up to a certain minimum given by α_{min}

If {h(x − 1), h(x)} ∈ $\{h_{-1}, h_0, h_1\}$ ⇒α(x)= max(h_1, −1, $\propto_{min}$)

If the quantized error h(x), assigned in the previous module, is different to the null hop or the smallest non-null hops {h_{-1}, h_1}, then the updated value α is set to its maximum α_{max}

If h(x) ∉$\{h_{-1}, h_0, h_1\}$ ⇒α= $\propto_{min}$

Coder

The coder module translates the quantized hops of all pixels into a compressed stream of bits. One possible approach for the compression technique can be based on the existing redundancy of images across its axes, that is, any pixel is generally similar to the previous one, and thus small hops are more frequently assigned. Although different compression techniques can be applied, this paper recommends the use of Huffman coding algorithm. It has variable-length codes for defining the quantized errors (called hops) based on its frequency of appearance.

However, analysis over two image databases [10, 11] reveals that the smallest hops are assigned in more than 90% of pixels. Therefore, in order to avoid the frequency analysis and enable real-time encoding, an effective statistical compression of hops can be achieved by assigning the smaller codes to the smaller hops. Table 1 shows an example of a LHE statistical coder with five hops codes.

Table 1:Statistical compression for five hopes

Quantised error (Hop)	Code, bits
h_0	1
h_1	01
h_{-1}	001
h_2	0001
h_{-2}	00001

LHE Decoding

The LHE decoder performs similar operations as in the LHE quantiser but in the reverse order. The following lines described the phases of the LHE decoding process for a given pixel x. It should be noted that previous pixels to x has been already decoded and

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

therefore the value of the parameter α(x) for this pixel has been already computed.

1. The binary stream is translated into symbols, in this case, into a certain hop value h(x).
2. The current predicted pixel ˆx is computed as the average of the colour components of the top and left pixels, as in (1).
3. Given the value of ˆx and α, the set of hops H(x) for this pixel are computed by following (2). At this moment, the colour component value of h(x) is known.
4. The decoded colour component x′ is computed as follows

$x' = \hat{x} + h(x)$

5. Finally, the new value for the parameter α is computed, following the rules described in Section hop adaptation.

As aforementioned, the first pixel colour components are included in raw-format in the binary stream, in order to enable the decoding process of the subsequent pixels.

4. EXPERIMENTAL RESULTS:

The following experimental results have been obtained by applying LHE algorithm with nine hops and by using pixel averaging. LHE provides a good objective quality (PSNR) Fig. 2 shows a quality comparison between an algorithm in SLIC and LHE. It can be seen that the LHE edges are cleaner than SLIC. Furthermore, LHE noise has less impact on the subjective quality compared SLIC because of the lack of visible artifacts at block boundaries.

Fig 2: Before compression

Fig 3: After compression-SLIC at 2.6 bpp 21.9 dB PSNR

Fig 4: After compression-LHE at 2.3 bpp 26.5dB PSNR

5. CONCLUSIONS:

LHE is a loss compression algorithm suitable for static images based on adaptive logarithmical quantization. One main contribution makes possible this performance that is, LHE proposes a logarithmical quantization of the error between pixel color component predictions and the actual value of such components. This quantization is based on Weber–Fechner law and it has been proven as a linear and quality effective compression procedure.

REFERENCES:

1. Lloyd, S.: ‘Measures of complexity: a nonexhaustive list’, IEEE Control Syst. Mag., 2001, 21, (4), pp. 7–8
2. Jayant, N., Johnston, J., Safranek, R.: ‘Signal compression based on models of human perception’, Proc. IEEE, 1993, 81, (10), pp. 1385–1422

3. Cruz, D.S., Ebrahimi, T., Larsson, M., Askelof, J., Cristopoulos, C.: 'Region of Interest coding in JPEG2000 for interactive client/server applications'. IEEE Third Workshop on Multimedia Signal Processing, 1999, pp. 389–394
4. Hassan, S.A., Hussain, M.: 'Spatial domain lossless image data compression method'. Int. Conf. on Information and Communication Technologies (ICICT), July 2011, pp. 1–4
5. Hasan, M., Nur, K., Noor, T.B., Shakur, H.B.: 'Spatial domain lossless image compression technique by reducing overhead bits and run length coding', Int. J. Comput. Sci. Inf. Technol. (IJCSIT), 2012, 3, (2), pp. 3650–3654
6. Hasan, M., Nur, K.: 'A novel spatial domain lossless image compression scheme', Int. J. Comput. Appl., 2012, 39, (15), pp. 25–28
7. Huang, S.C., Chen, L.G., Chang, H.C.: 'A novel image compression algorithm by using Log-Exp transform'. Proc. 1999 IEEE Int. Symp. on Circuits and Systems (ISCAS'99), July 1999, vol. 4, pp. 17–20
8. Rabbani, M., Jones, P.W.: 'Adaptive DPCM', in 'Digital image compressiontechniques', (SPIE – The International Society for Optical Engineering Press,1991)
9. Weinberger, M.J., Seroussi, G., Sapiro, G.: 'The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS', IEEE Trans. Image Process., 2000, 9, (8), pp. 1309–1324
10. 'Kodak lossless true color image suite', http://www.r0k.us/graphics/kodak/, accessed January 2014
11. 'USC-SIPI image database', http://sipi.usc.edu/database/database.php?volume=misc, accessed January 2014
12. Christopoulos, C.A., Askelof, J., Larsson, M.: 'Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard', IEEE Signal Process. Lett., 2000, 7, (9), pp. 247–249
13. Moreno, J.M.: 'Perceptual criteria on image compression'. Ph.D. dissertation, DCC, UAB, Barcelona, Spain, 2011
14. 14 Zhang, C.N., Wu, X.: 'A hybrid approach of wavelet packet and directional decomposition for image compression'. IEEE Canadian Conf. on Electrical and Computer Engineering, Edmonton, Alta, Canada, December 1999, vol. 2, pp. 755–760
15. Zhou, Z., Venetsanopoulos, A.N.: 'Morphological methods in image coding'. IEEE Int. Conf. on Acoustics, Speech, and Signal Process (ICASSP-92),San Francisco, CA, USA, March 1992, vol. 3, pp. 481–484
16. Popovici, I., Withers, W.D.: 'Locating edges and removing ringing artifacts in JPEG images by frequency-domain analysis', IEEE Trans. Image Process., 2007, 16, (5), pp. 1470–1474
17. Mittal, A., Moorthy, A.K., Bovik, A.C.: 'No-reference image quality assessment in the spatial domain', IEEE Trans. Image Process., 2012, 21, (12), pp. 4695–4708
18. Zhang, L., Zhang, D., Mou, X., Zhang, L.: 'FSIM: a feature similarity index for image quality assessment', IEEE Trans. Image Process., 2011, 20, (8), pp. 2378–2386
19. Kok, C.W.: 'Fast algorithm for computing discrete cosine transform', IEEE Trans. Signal Process., 1997, 45, (3), pp. 757–760