ABHIYANTRIKI
An International Journal of Engineering & Technology
(A Peer Reviewed & Indexed Journal)
Vol. 3, No. 5 (May, 2016) http://www.aijet.in/ eISSN: 2394-627X

Hierarchical Prediction and Context Adaptive Coding for Lossless Color Image Compression

M V Deepthi Ethiraj
PG Student (DSCE)
Deptt. of ECE, PBR VITS
Kavali, Nellore, India

P Sreenivasulu*
Associate Professor
Deptt. of ECE, PBR VITS
Kavali, Nellore, India

Abstract

Paper introduced a new lossless color image compression algorithm based on the hierarchical prediction and context-adaptive arithmetic coding. For that first converted RGB image into Y CbCr form using a reversible color transforms, then Y component was encoded by conventional lossless grayscale image compression method. The hierarchical scheme that enables the use of upper, left, and lower pixels for the pixel prediction was used for encoding the chrominance images. An appropriate context model for the prediction error was also defined and context adaptive coding was applied for the error signal. The proposed method provides better compression of the image at the boundaries and further reduces the bit rates compared with JPEG2000 and JPEG-XR.

Keywords*: Lossless color image compression, Adaptive arithmetic coding, Run length coding, Reversible color transform, Hierarchical prediction.*

****Author for correspondence*** vardhan.sreenivasulu@gmail.com

1. Introduction

Digital images are usually encoded by lossy compression methods due to their large memory or bandwidth requirements. The lossy compression methods achieve high compression ratio at the cost of image quality degradation. However, there are many cases where the loss of information or artifacts due to compression needs to be avoided, such as medical, prepress, scientific and artistic images. As cameras and display systems are going high quality and as the cost of memory is lowered, we may also wish to keep our precious and artistic photos free from compression artifacts. Hence efficient lossless compression will become more and more important, although the lossy compressed images are usually satisfactory in many cases. Along with the standardization or independently, many lossless image compression algorithms have been proposed. Among a variety of algorithms, the most widely used ones may be lossless JPEG [1], JPEG-LS [2], LOCO-I [3], CALIC [4], JPEG2000 [5] (lossless mode) and JPEG XR [6]. The LOCO-I and CALIC were developed in the process of JPEG standardization, where most ideas in LOCO-I are accepted for the JPEG-LS standard although the CALIC provides better compression performance at the cost of more computations. For the compression of color images, the color components are first de-correlated by a color transform, and each of the transformed components is independently compressed by the above referenced methods. For example, the RGB to Y CbCr transform [7] may be the most frequently used one for the lossy compression of color image and video. However, in the case of lossless compression, most color transforms cannot be used due to their un-invertibility with integer arithmetic. Hence an invertible version of color transform, the reversible color transform (RCT) was defined and used in JPEG2000 [5]. There has also been much research for finding better RCTs [8, 9, 10], among which we adopt a transform proposed in [9] because it approximates the conventional Y CbCr transform very well. The purpose of the study is to develop a hierarchical prediction scheme, while most of existing prediction methods in lossless compression are based on the raster scan prediction which is sometimes inefficient in the high frequency region. The "hierarchical" prediction for the compression was already proposed in [11], but only pixel interpolation is used here. In this paper, we design an edge directed predictor and context adaptive model for this hierarchical scheme. To be specific, we propose a method that can use lower row pixels as well as the upper and left pixels for the prediction of a pixel to be encoded. For the compression of color images, the RGB is first transformed to Y CuCv by an RCT mentioned above [9], and Y channel is encoded by a conventional grayscale image compression algorithm. In the case of chrominance channels (Cu and Cv), the signal variation is generally much smaller than that of RGB, but still large near the edges. For more accurate prediction of these signals, and also for accurate modeling of prediction errors, we use the hierarchical scheme: the chrominance image is decomposed into two sub-images i.e. a set of even numbered rows and a set of odd numbered rows respectively. Once the even row sub-image Xe is encoded, we can use all the pixels in Xe for the prediction of a pixel in the odd row sub-image Xo. In addition, since the statistical properties of two sub-images are not much different, the PDF of prediction errors of a sub-image can be accurately modeled from the other one, which contributes to better context modeling for arithmetic coding. Experiments on various kinds of images are performed, and it is shown that the proposed method provides higher coding gain than JPEG2000 and JPEG-XR in many cases.

2. Hierarchical Prediction

Most of the existing prediction methods in lossless compression are based on the raster scan prediction which is sometimes inefficient in the high frequency region. We design an edge directed predictor and arithmetic coding for this hierarchical scheme. To be specific, we propose a method that can use lower row pixels as well as the upper and left pixels for the prediction of a pixel to be encoded. For the compression of color images, the RGB is first transformed to Y Cu Cv by an RCT mentioned above and Y channel is encoded by a conventional gray scale image compression algorithm. In the case of chrominance channels (C*u* and C*v*), the signal variation is generally much smaller than that of RGB, but still large near the edges. For more accurate prediction of these signals, and also for accurate modelling of prediction errors, we use the hierarchical scheme: the chrominance image is decomposed into two sub images i.e. a set of

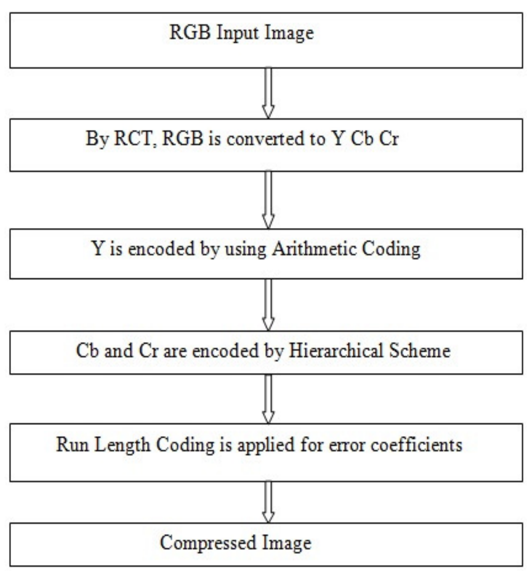

Fig.1: Flow Chart

even numbered rows and a set of odd numbered rows respectively. Once the even row sub image *Xe* is encoded, we can use all the pixels in *Xe* for the prediction of a pixel in the odd row sub image Xo. In addition, since the statistical properties of two sub images are not much different, the PDF of prediction errors of a sub image can be accurately modelled from the other one, which contributes to better context modeling for arithmetic coding.

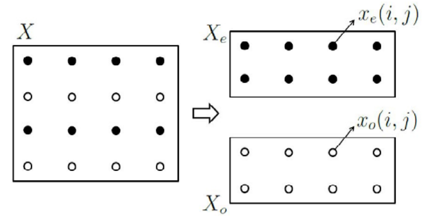

Fig. 2: Input image and its decomposition

The chrominance channels Cu and Cv resulting from the RCT usually have different statistics from Y and also different from the original color planes R, G and B. In the chrominance channels, the overall signal variation is suppressed by the color transform, but the variation is still large near the object boundaries. Hence, the prediction errors in a chrominance channel are much reduced in a smooth region, but remain relatively large near the edge or within a texture region. For the efficient lossless compression, it is important to accurately estimate the pdf of prediction error for better context modeling, along with the accurate prediction. For this, we propose a hierarchical decomposition scheme as depicted in Fig. 2, which shows that pixels in an input image *X* is separated into two sub images: an even sub image Xe and a odd sub image *Xo*.

Then Xe is encoded first and is used to predict the pixels in *Xo*. In addition, Xe is also used to estimate the statistics of prediction errors of *Xo*. For the compression of *Xo* pixels using Xe, directional prediction is employed to avoid large prediction errors near the edges. For each pixel xo(i, j) in *Xo*, the horizontal predictor x^h(i, j) and vertical predictor x^v (i, j) are defined as:

$$x\hat{}h(i,j) = xo(i,j-1)$$

$$x\hat{}v(i,j)= round((xe(i,j)+xe(i+1,j)/2) \qquad .....(1)$$

One of them is selected as a predictor for *xo(i, j)*. With these two possible predictors, the most common approach to encoding is "mode selection," where better predictor for each pixel is selected and the mode (horizontal or vertical) is also transmitted as side information. However, the vertical predictor is more often correct than the horizontal one when the predictors are defined as (1) because upper and lower pixels are used for the "vertical" whereas just a left pixel is used for the "horizontal." The horizontal predictor is more accurate only when there is a strong horizontal edge. Hence, the vertical predictor is used for most pixels, and mode selection is used only when the pixel seems to be on a strong horizontal edge. For implementing this idea, we define a variable for the direction of edge at each pixel *dir(i, j)*, which is given either *H* or *V*. Actually, it is given *H* only when the horizontal edge is strong, and given *V* for the rest.

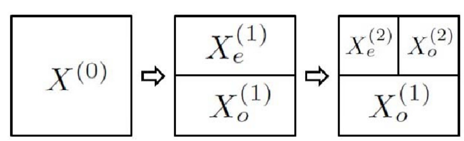

Fig. 3: Illustration of Hierarchical Decomposition

Based on the directions of pixels, the overall prediction scheme is designed. It can be seen that the mode selection is tried when more than one of *dir(i* −1, *j*) or *dir(i, j* − 1) are *H* and the vertical prediction is performed for the rest. A chrominance image X(0) ∈ {Cu, Cv} is decomposed row by row into an even sub image *X*(1)e and an odd sub image *X*(1)o as shown in Fig. 3. The sub image *X*(1)o is predicted and encoded using *X*(1)e. The sub image X(1)e can be further decomposed column by column into the even sub image X(2)e and the odd sub image X(2)o as shown in the last figure 2, where the sub image X(2)o is compressed using X(2)e.

In the predictive lossless compression, efficient encoding of the prediction error e(i, j) = xo(i, j) – xo^(i, j) plays an important role. Although the proposed prediction method usually generates small prediction errors owing to the RCT and the sophisticated prediction scheme, there are still relatively large errors near the edge or texture region, which degrades the compression performance. For the efficient compression, the statistics of symbols (prediction errors) should well be described by an appropriate model and parameters.

Run Length Coding

Run-length encoding (RLE) is a very simple form of data compression in which runs of data that is sequences in which the same data value occurs in many consecutive data elements are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could greatly increase the file size. Typical applications of this encoding are when the source information comprises long substrings of the same character or binary digit. For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWW
WWWWWWWWWWBWWWWWWWWWWWWWW

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

12W1B12W3B24W1B14W
This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc.

The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. Run-length encoding can be expressed in multiple ways to accommodate data properties as well as additional compression algorithms. For instance, one popular method encodes run lengths for runs of two or more characters only, using an “escape” symbol to identify runs, or using the character itself as the escape, so that any time a character appears twice it denotes a run.

On the previous example, this would give the following:

WW12BWW12BB3WW24BWW14

This would be interpreted as a run of twelve Ws, a B, a run of twelve Ws, a run of three Bs, etc. In data where runs are less frequent, this can significantly improve the compression rate. One other matter is the application of additional compression algorithms. Even with the runs extracted, the frequencies of different characters may be large, allowing for further compression; however, if the run lengths are written in the file in the locations where the runs occurred, the presence of these numbers interrupts the normal flow and makes it harder to compress. To overcome this, some run-length encoders separate the data and escape symbols from the run lengths, so that the two can be handled independently. For the example data, this would result in two outputs, the string "WWBWWBBWWBWW" and the numbers (12, 12, 3, 24, 14).

3. Results & Discussions

The proposed method does not always perform best for every set of images. The proposed hierarchical encoding scheme sometimes works better and sometimes worse than the conventional methods, depending on image sets and also depending on the channels (Y, C*u*, and C*v*). It is also true for every compression algorithms i.e. the coding gain of compression algorithms differ on different set of images.

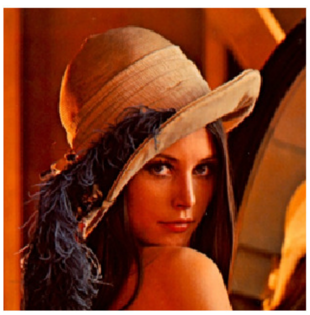

Fig. 4: Original image

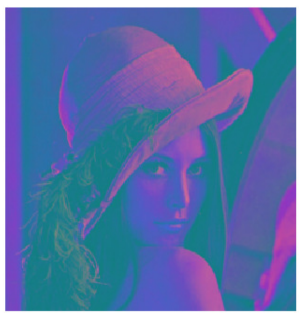

Fig. 5: YCbCr color space

Fig. 6: Chrominance image Cb

Fig. 7: Chrominance image Cr

Fig. 8: Reconstruction of Cb

Fig. 9: Reconstruction of Cr

Fig. 10: Luminance image Y

Fig. 11: Reconstruction of Y

Fig. 12: Reconstruction of YCbCr

Fig. 13: Reconstruction of Original Image

Table 1: Tabulation of Results for Classic Images

IMAGE	SIZE	MSE	PSNR	COMPRESSION RATIO
Lena	512*512	31.37	33.16	15.7927
Pears	512*512	15.88	36.12	15.7317
Barbara	640*512	28.72	33.54	15.5985
Mandrill	512*512	25.69	34.03	15.0312
Peppers	512*512	37.54	32.38	16.5998

Table 2: Performance Comparison of Compression Ratio for Classic Images

IMAGE	SIZE	CALIC	JPEG 2000	JPEG XR	Proposed
Peppers	512*512	13.1787	14.8000	15.3245	16.5998
Lena	512*512	13.8661	13.5848	14.0942	15.7927
Mandrill	512*512	18.1511	18.0939	18.2553	15.0312

It is very evident from the results that, our proposed method provides better compression ratio compared with all existing methods.

4. Conclusion

Here we have proposed a lossless color image compression method based on a hierarchical prediction scheme and context adaptive coding is used for error calculation. For the compression of an RGB image, it is first transformed into Y Cb Cr color space using an RCT. The luminance channel Y is compressed by a conventional lossless image coder. Pixels in chrominance channels are predicted by the hierarchical decomposition and directional prediction. Finally, an appropriate context modelling of prediction residuals is introduced and run length coding is applied. By applying different context adaptive coding techniques like Huffman coding and arithmetic coding for the error signal we can obtain better performance and signal to noise ratio can be further improved.

References

[1] WB Pennebaker, & JL Mitchell. *JPEG Still Image Data Compression Standard*. NewYork: Van Nostrand Reinhold, 1993.

[2] Information Technology – Lossless and Near-Lossless Compression of Continuous-Tone Still Images (JPEG-LS), ISO/IEC Standard 14495-1, 1999.

[3] M Weinberger, G Seroussi, & G Sapiro. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *IEEE Trans. Image Process*. 9(8), pp. 1309-1324, 2000.

[4] X Wu, & N Memon. Context-based, adaptive, lossless image coding. *IEEE Trans. Commun*. 45(4), pp. 437–444, 1997.

[5] Information Technology – JPEG 2000 Image Coding System – Part 1: Core Coding System, INCITS/ISO/IEC Standard 15444-1, 2000.

[6] ITU-T and ISO/IEC, JPEG XR Image Coding System – Part 2: Image Coding Specification, ISO/IEC Standard 29199-2, 2011.

[7] G Sullivan. Approximate theoretical analysis of RGB to YCbCr to RGB conversion error. ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 document JVT-I017, 2003.

[8] HS Malvar, GJ Sullivan, & S Srinivasan. Lifting-based reversible color transformations for image compression. Proc. SPIE. Vol. 707307, pp. 1-10, 2008.

[9] S Pei, & J Ding. Improved reversible integer-to-integer color transforms. Proc. 16th IEEE ICIP, Nov. 2009, pp. 473–476.

[10] T Strutz. Adaptive selection of colour transformations for reversible image compression. Proc. 20th Eur. IEEE Signal Process. Conf., pp. 1204 – 1208, 2012.

[11] P Roos, MA Viergever, MCA van Dijke, & JH Peters. Reversible intra frame compression of medical images. *IEEE Trans. Med. Image*. 7(4), pp. 328 – 336, 1988.