Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004
June 9 - 11, 2004
Espoo, Finland

Predictive Differential Modulation for CFA compression

Arcangelo Bruna[1†], *Filippo Vella*[1], *Antonio Buemi*[1] *and Salvatore Curti*[1]

[1]STMicroelectronics
Advanced System Technology
Stradale Primosole 50, 95121 Catania
ITALY
[†]Tel. +39-095-740 4290, Fax: ++39-095-740 4004
[†]E-mail: (arcangelo.bruna, filippo.vella, antonio.buemi, salvatore.curti)@st.com

ABSTRACT

The recent, wide diffusion of Digital Still Cameras (DSCs) and mobile imaging devices disposes the need of developing efficient techniques for digital image coding, in order to reduce resources requirements for their storing and transmission. Digital cameras usually acquire the image data through a Bayer Color Filter Array (CFA), a light sensitive sensor able to acquire the red, green and blue colors. Each pixel acquires just one color component. The full color image is then restored applying a sequence of image processing algorithms interpolating the acquired data. This paper introduces a new, efficient coding method to compress the Bayer Pattern. It is based on a predictive schema followed by a Vector Quantization (VQ) technique. Simulations have demonstrated that the proposed scheme allows a visually lossless compression of Bayer pattern images with low memory cost.

1. INTRODUCTION

Most Digital Still Cameras (DSCs), phone cameras and several kinds of devices with integrated cameras acquire data in Bayer pattern format [3]. It is mainly a Color Filter Array (CFA) able to retrieve, for each pixel, the intensity information of a single color component (Red, Green and Blue) in the pattern shown in the

Fig. 1. The growth of the sensor's size and the need of transmitting or storing the Bayer data obliged to looking for compression techniques. The constraint for such techniques is, of course, to be lossless or visually lossless. So, ideally, compression of these data should be obtained through lossless techniques, reducing only the interpixel and coding redundancies.

On the other side, lossless techniques allow low compression rates and they preserve psychovisually redundant information. Lossy compression methods, instead, yield high compression ratios and ideally they should discard only visually not-relevant information. The compression scheme proposed in this paper allows performing a near-lossless compression of CFA images, by reducing both mathematical and perceptual redundancies.

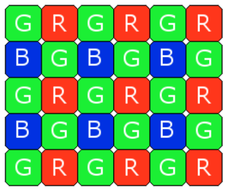

Fig. 1: Bayer Pattern CFA.

The problem of Bayer data compression is quite recent and although traditional coding techniques offer good performances on full color images, most of these do not offer the same performances when images captured by digital sensors in CFA format are processed. The most trivial and cheap solution is to split the Bayer image into its color components (Red, Green, Blue) and compress them independently using an efficient algorithm, i.e. DPCM [2], [4]. More sophisticated compression methods for images in Bayer pattern format [5] are based on Wavelet transform. Using the DWT could be useful because it makes possible to describe the edges better then the Fourier transform, making also easier to obtain a lossy compression that is perceived without loss of visual information by Human Visual System (HVS). Several subband-coding compression methods have been introduced by Toi [6] and Le Gall [7].

Others techniques are based on adapting the JPEG [7] standard algorithm to the Bayer data [9], [10], [11]. These approaches usually need to pre-process the image to convert Bayer pattern in the YCbCr format with 4:2:2 or 4:2:0 subsampling. But the Bayer is not complete, i.e. there are some missed colors, so the Y data has blank pixels. It requires another transformation in order to build the 8x8 blocks needed to apply the JPEG compression algorithm. The main drawback of all these approaches is the noticeable amount of memory and bandwidth required. More efficient compression techniques based on the Vector Quantization (VQ) coupled with consideration on the Human Visual System (HVS) have been proposed recently in [12], [13].

The basic idea is that the Bayer pattern values can be clustered into groups of two pixels accordingly to their

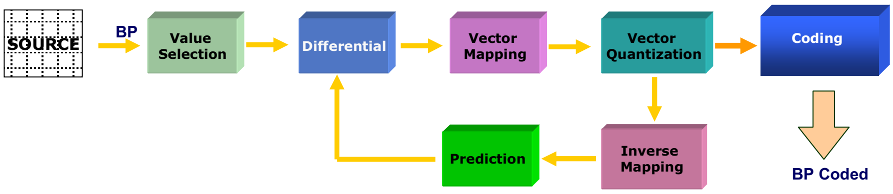

Fig. 2: Proposed algorithm basic scheme.

color channel and the generated pair can be quantized with a function considering the edge masking and the luminance masking of the HVS.

The compression algorithm introduced in this paper exploits the concept of "perceptive VQ" in order to efficiently compress the Bayer pattern. It is based on a predictive schema and the quantization of the prediction error, computed by a simple DPCM system. As result, a low cost, lossy (but visually lossless) compression technique oriented to the Bayer Pattern has been devised.

The paper is organized as follows: Section 2 describes the proposed method. Section 3 presents the results of the experiments. Conclusions and final remarks are discussed in the last section.

2. COLOR FILTER ARRAY COMPRESSION

2.1 The proposed algorithm

The proposed algorithm basic scheme is showed in Fig. 2. First, the well-known DPCM algorithm [2] is applied to compute the difference between the current and the predicted values of pixels gathered in vectors of two elements. The prediction is performed assuming the adjacent samples of the same color component having similar brightness values. Thus, the samples to be compressed are the consecutive pixel of the same color channel. The error between the current and the predicted vector is then processed by the "Vector Mapping" block. It deletes some symmetries providing better compression ratio. Finally a lossy compression is permorfed by the "Vector Quantization" block. The vector quantizer has been designed to discard information which is believed not perceivable, accordingly to psychovisual considerations based on HVS properties. The quantized values are then processed by the "Coding" block and its output is the compressed stream. The quantized vector is decoded in order to predict the vector to be used in the next step. The decoding allows to avoid the error propagation.

The algorithm was developed and tested with 10-bit per pixel images and 12 bit per pixel codes. Since each couple of 10-bit values will generate a 12-bit code, a compression of 12/20 (40%) is achieved. The same process could be easily extended to the case of n-bit images.

Next subsections describe deeply each step of the coding procedure.

2.2 Step 1: Predictive DPCM

The blocks in the fig. 2 named "Differential" and "Prediction" perform the Predictive DPCM step. It is based on the reduction of the entropy of the information source by coding the difference between the current value and its prediction.

The function used in our experiments performs the prediction between 2-dimensional vectors obtained applying VQ. In particular, let (V_i,V_j) be the sample that should be currently coded and (V_{i-1},V_{j-1}) the previous one. Different prediction formula can be also used. We noticed that better results are obtained using (V_{j-1},V_{j-1}) as predictor for (V_i,V_j). This strategy has been chosen because the V_{j-1} is spatially closer to (V_i,V_j) and usually closer samples are also more correlated than the others. A typical error distribution for this kind of prediction scheme is shown in **Fig. 3**

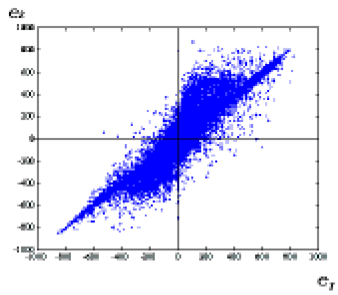

Fig. 3: Typical prediction error spreading.

The diagram shows that a very high percentile of values fall near origin, so the vector quantizer used in the next step has been created to modelling this characteristic distribution, as described below.

Another important feature is that the prediction function processes the restored values, as shown in Fig. 2, in order to avoid errors propagation. In fact, using the reconstruct value, the encoder uses the same vector than the decoder.

2.3 Step 2: Vector Mapping

In fig. 3 can be notice a odd symmetry in the prediction error distribution. It has been exploited in order to reduce the quantization table size. All samples to be compressed are processed in the "Vector Mapping" block to determine if the value falls in the I and in the II quadrant or in the III and in the IV quadrant. In the first case no changes are performed in the input vector. Otherwise, the values are changed in sign (*Vout=-Vin)*. In this manner the output vector is always mapped in the first and in the second quadrant. A bit in the output code takes into account if the swap has been done.

2.4 Step 3: Vector Quantization

The basic concept of Vector Quantization [1] can be described geometrically. Given a vector $(X_1,\ldots,X_n)$ of size N, it can be seen as a set of N coordinates locating a unique point in the N-dimensional geometrical space. The quantization is performed partitioning the space with *N*-dimensional cells (e.g. hyper cubes) with no gaps and no overlaps. As the point defined by the input vector falls in one of these cells, the quantization process returns a single vector associated with the selected cell. Finally, such vector is mapped to a unique binary representation, which is the actual output of the vector quantizer. This binary representation (code) can have fixed or variable length.

A vector quantizer is defined "uniform" if the same quantization step is applied to each vector element, so that the N-dimensional space is divided into regular cells. If the space is partitioned into regions of different size, corresponding to different quantization steps, the quantizer is defined "not-uniform". The "target" vector is called "codevector" and the set of all codevectors is the "codebook".

A grayscale 10 bit image is described by 2-dimensional vectors of brightness values falling into the range [0, 1023]. In the proposed model, each codevector represents a region that gather a group of vectors in the 2-dimensional space. Moreover, the proposed algorithm uses a not uniform quantization driven by some properties of the HVS. In particular, two considerations has been taken into account: quantization errors are less visible along the edges and the HVS discriminates better the details at low luminance levels. Thus, in the areas near the origin (where the prediction error is low) the quantizer performs a fine quantization. On the contrary, in the regions far from the origin of the space, a coarser quantization is applied due to the presence of boundaries. Furthermore, since DPCM drifts values towards zero, a very high percentile of input samples will fall in the area around zero. In the testing database used in experiments and reported later in this paper, the 65% of values has a magnitude less than 20 and the 80% of values are less than 38. This information has been exploited to partition the two upper quadrants of the 2-dimensional space into regions shaped and distributed to minimize the quantization error. Each region has different size and position in the quantization board and it is divided into 64 "sub-regions". Such regions are obtained dividing the horizontal and vertical dimension by a constant number. In this way, bigger regions cover bigger areas and the quantization is stronger (more loss of information), while smaller areas a "light" quantized and most information is preserved (Fig. 4).

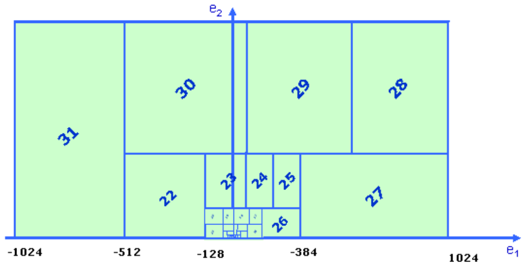

Fig. 4: Quantization space partitioning.

The quantization step, for both the horizontal and the vertical axes (*X_step* and *Y_step*) are:

$$X\_Step = \frac{RegionWidth}{HorizontalQuantizationStep}$$

$$Y\_Step = \frac{RegionHeigth}{VerticalQuantizationStep}$$

Where *RegionWidth* and *RegionHeight* are, respectively, the horizontal and the vertical size of the region; *Horizontal Quantization Step* and *Vertical Quantization Step* are the quantizer values. The *Horizontal Quantization Step* and the *Vertical Quantization Step* are set accordingly to the length of the code and the allowed distortion in quantization. In our implementation we fixed 3 bit, thus the quantizater values are 8 for both the horizontal and the vertical axes, obtaining 64 sub-regions.

2.5 Step 4: Code Generation

Each vector is represented by a fixed length code. It summarizes information about the vector mapping, the region where the point falls and the quantization steps applied in each region (Fig. 5). The first bit in the code is indicates if the swap between upper and bottom quadrants happened or not. Few bits following the first bit of the code indicates the index of the region in the quantization table. The length of this part of the code depends on the number of region of the quantization table. The remaining bits give information on the number of steps, in both vertical and horizontal direction inside the region.

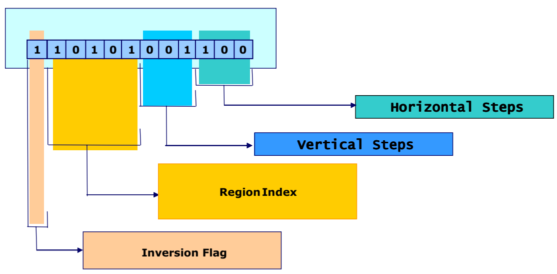

Fig. 5: Compression code generation.

In this discussion we assumed that a 12-bits code should be generated, in order to represent samples falling in a space partitioned into 32 regions and 64 "sub-regions". Thus, the code reserves five bits to index 32 regions and 6 bits to index one of the 64 sub-regions. Different code structure could be defined if the space partitioning or the target bit-rate changes.

The decoding procedure consists on three main steps. The first is the code evaluation, which allows the extraction of the compressed values. The second one is the "Inverse Vector Mapping" that, depending on the inversion flag, assigns the right sign to values and a step of reconstruction. The retrieving of the original values is done adding the decoded error to the previously restored values.

3. RESULTS

The algorithm has been tested on two sets of images acquired in different light conditions. A first set, named "Wide Range Image set", contains images with a histogram distributed in almost all the range of the intensity of each channel and acquired by a CMOS-VGA sensor. Further tests have been done on a second set of images having a very narrow histogram. The proposed method had very good performance with a PSRN of about 56 dB in the first image set (Fig. 6). Lower, but still high (about 50 dB) PSNR has been achieved when the images with a wider range have been processed (Fig. 7). In both cases the compression didn't introduce perceptible artifacts in the output images.

4. CONCLUSIONS

The paper describes a new, low-cost algorithm for the compression of the image captured by digital sensors. The proposed technique is based on the quantization of the prediction error of the source. The compression achieved is 40% of the input data thanks to a fixed-length code assigned to each couple of input pixel. Moreover, experimental results showed that compression doesn't involve a perceptible loss of quality in the output. Compared to other algorithms (e.g. the pure DPCM, VQ, …), this one has a lower bit rate and the distortion is low too, having a PSNR ~50 dB.

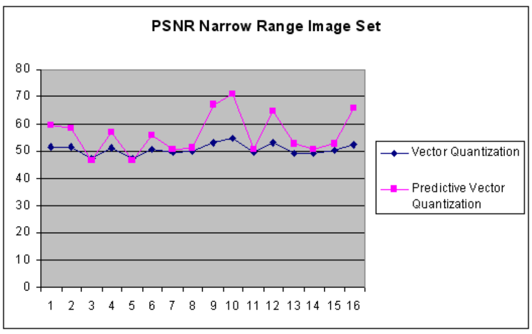

Fig. 6: PSNR on "Narrow Range Image Set".

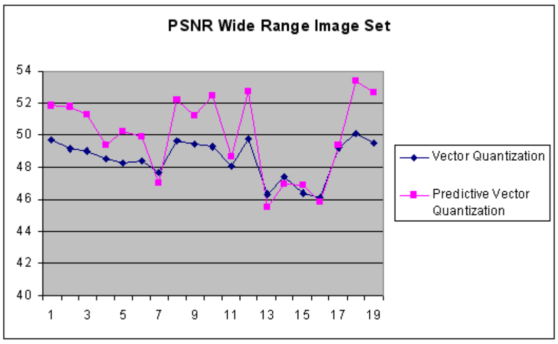

Fig. 7: PSNR on "Wide Range Image Set".

REFERENCES

[1] R.M. Gray, D.L. Neuhoff, Quantization, *IEEE Trans. on Information Theory*, vol. 44, n. 6, 1998.

[2] R.C. Gonzales, R.E. Woods, Digital Image Processing, Addison Wesley, 1993, pp.358-374.

[3] B.E. Bayer, Color Imaging Array, U.S. Patent 3,971,065, 1976.

[4] Hynix CMOS Image Sensor Application Note 2002, *Endpoints SE401 Demokit Information*, http://www.hynix.com/datasheet/pdf/system_ic/sp/SE401_Demokit.pdf

[5] T. Acharya, L.J. Karam, F. Marino Compression of Color Images Based on a 2-Dimensional Discrete Wavelet Transform Yielding a Perceptually Lossless Image, U.S.Patent 6,154,493, 2000.

[6] T.Toi, M.Ohita, A Subband Coding Technique for Image Compression in Single CCD Cameras with Bayer Color Filter Arrays, *IEEE Transaction on Consumer Electronics*, Vol. 45, N.1, 1999, pp.176-180.

[7] Le Gall, A. Tabatabai, Subband Coding of Digital ImagesUsing Symmetric Shor Kernel Filters and Arithmetic Coding Techniques, in *Proceedings of the ICASSP 88 Conference, (New York)*, pp. 761-764, 1988.

[8] G. K. Wallace, The JPEG still picture compression standard, Communications of the ACM 34(4), 1991, pp. 30-44.

[9] Y.T. Tsai, K.A. Parulsky, M. Rabbani, Compression Method an Apparatus for Single-Sensor Color Imaging Systems, U.S. Patent 5,065,229, November 12, 1991.

[10] Sang-Yong Lee, A. Ortega, A Novel Approach of Image Compression in Digital Cameras With a Bayer Color Filter Array, *In Proceedings of ICIP 2001* – International Conference on Image Processing – Vol. III 482-485 - Thessaloniki, Greece, 2001.

[11] Chin Chye Koh, J. Mukherjee, S.K. Mitra, New Efficient Method of Image Compression in Digital Cameras with Color Filter Array, *IEEE Transaction on Consumer Electronics*, Vol. 49, N. 4, November 2003, pp. 1448-1456.

[12] S. Battiato, A. Buemi, L. Della Torre, A. Vitali, "Fast Vector Quantization Engine for CFA Data Compression", *In Proceedings of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing,* NSIP Grado, Italy, 2003.

[13] S. Battiato, A. Bruna, A. Buemi, F. Naccari, "Coding Techniques For CFA Data Images", *International Conference on Image Analysis and Processing*, ICIAP 2003, Mantova, Italy, September 17-19, 2003.