

# Color Image Compression with a Variable Block-Size Transform Scheme

Javier Bracamonte, Michael Ansorge, and Fausto Pellandini

Institute of Microtechnology, University of Neuchâtel  
Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland  
Phone: +41 32 7183426; Fax: +41 32 7183402  
*javier.bracamonte@imt.unine.ch*

## Abstract

*In a previous paper an adaptive transform-based coding method was introduced and used for the particular application of compressing the monochrome images of videoconference sequences [1]. In the present paper the results of applying this adaptive scheme to compress still color images are presented. The basic principle of the algorithm is to encode those larger  $2N \times 2N$ -pixel regions featuring low image activity with a single  $2N$ -point transform, and code high activity  $2N \times 2N$ -pixel regions with four  $N$ -point transform operations. Regarding the implementation issues, a strategy to reduce the computational complexity of the algorithm is also discussed.*

## 1 Introduction

In transform-based image coding the input image is divided into blocks of  $N \times N$  pixels, and on each of these blocks a linear transform is applied. Usually when the value of  $N$  has been chosen in a particular algorithm, it remains fixed. In this paper the results of using a variable size for  $N$  when coding different regions of the same input image are reported.

The next section discusses the issue of selecting the value of  $N$  in image coding. Section 3 presents the proposed variable block scheme and its coding performance is shown in section 4. An approach for reducing the computational complexity of the system is discussed in section 5. Finally the conclusions are stated in section 6.

## 2 Selection of the Block-size in Transform Coding

Several factors influence the size selection of the block of  $N \times N$  pixels in transform-based image coding. The most important are:

a) *Interpixel Correlation*: The similarity among the values of the pixels in a region is inversely proportional to the distance separating them. Since the compression efficiency of a linear transform is higher in well pixel-correlated regions, this fact suggests that choosing a small block-size is preferable. It was reported in [2] that a significant data correlation is obtained only within a region of 20 adjacent pixels, which hints a practical upper limit for the block size.

b) *Energy Packing*: This term is associated with the property of a linear transform of distributing most of the input signal energy into the smallest number of transform coefficients. Several reports [3] have indicated that as the value of  $N$  decreases so does the energy packing efficiency of the associated  $N$ -point transform, and so does consequently, the compression efficiency. In this sense, the larger the chosen block-size, the better.

c) *Computational Complexity*: The precedent two factors imply a trade off for the choice of an optimum value of  $N$ . This compromise can be settled by taking into consideration the memory and computational resources required to implement a  $N$ -point linear transform. In effect, the substantial growth of the computational complexity related with increasing values of  $N$ , in some instances overshadows the associated increase of compression efficiency.

Table 1 presents a summary of the characteristics related to four different  $N \times N$  block sizes and to their corresponding  $N$ -point linear transforms. It is observed from this table that by choosing a  $N \times N$ -pixel block-size with  $N = 8$  or 16, a good trade off is obtained between compression efficiency and computational complexity. In the here-proposed compression scheme these two block sizes are used to compress different regions within the same input image, as explained in the next section.

## 3 Compression Scheme

Figure 1 shows a block diagram of the compression scheme for color images. The input RGB image is

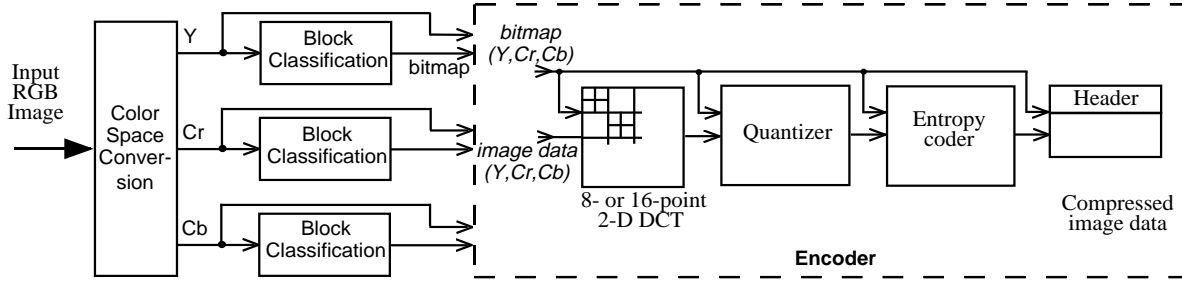


Figure 1: Compression Scheme

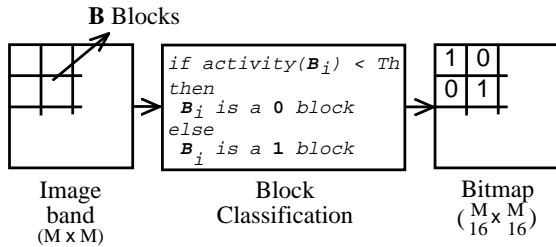


Figure 2: Classification Stage

Block Size	Pixel Correl.	Memory Req.	Comput. Compl.	Energy Packing
4x4	+++	++	++	-
8x8	++	+	+	+
16x16	+	-	-	++
32x32	-	--	--	+++

Table 1: Characteristics related to a block size and its associated  $N$ -point transform. [+/-: Degree of convenience/inconvenience]

converted into the color space YCrCb [4], as usually made in current image and video coding standards.

The classification module is shown in Figure 2. This unit receives a  $M \times M$ -pixel image band, subdivides this image into non-overlapping  $16 \times 16$ -pixel blocks ( $\mathbf{B}$  blocks) and computes an image activity metric over each of these blocks. If the activity measure is below a given threshold the current  $\mathbf{B}$  block is classified as a  $\mathbf{0}$  block, which indicates to the encoder downstream that it should be coded as a single  $16 \times 16$  block. If the activity measure is above the given threshold, then the block is classified as a  $\mathbf{1}$  block, which indicates to the encoder that this block should be further divided into 4 non-overlapping blocks, each of which being coded separately.

The scheme of the encoder in Figure 1 illustrates a baseline JPEG-based algorithm in which each of its constituent operations (DCT, quantization and entropy coding) has been modified in order to operate in two different modes: mode  $\mathbf{0}$  and mode  $\mathbf{1}$ . The same operations are carried out over the Y, Cr and

Cb image bands, except that different normalization and Huffman tables are used for the luminance and for the chrominance signals.

In mode  $\mathbf{1}$ , the input  $\mathbf{B}$  block is further divided into four  $8 \times 8$ -pixel subblocks, and the encoding operations on each of the four subblocks are carried out as defined by the JPEG standard [5, 6]. In mode  $\mathbf{0}$  all the encoding operations will be carried out with the JPEG-like scheme in which the value of  $N$  is 16 (instead of the regular  $N = 8$ ). Thus, in mode  $\mathbf{0}$ , a 2-D 16-point DCT is applied over a whole  $\mathbf{B}$  block, then the DCT coefficients are quantized using a  $16 \times 16$  normalization matrix and finally the normalized DCT coefficients are entropy coded. The coding mode for each  $\mathbf{B}$  block of an input image band is given by the corresponding binary value on the bitmap that results from the classification stage.

Obviously for color images the overhead information corresponds to the three classification bitmaps from the image bands Y, Cr, and Cb. These bitmaps are embedded in the final header of the compressed image. The overhead due to the luminance bitmap represents an increase of the coding bit rate of  $1/256$  bit per pixel (bpp). Since both Cr and Cb are sub-sampled by 2:1 in both horizontal and vertical directions before encoding, their bitmaps increase the coding bit rate by only  $1/1024$  bpp each. Thus, the total effective information overhead of this adaptive system is conveniently low: only  $3/512$  bpp.

## 4 Results

Figures 3 show the results of the adaptive scheme applied to the color images Airplane ( $512 \times 512$  pixels) and Parrots ( $512 \times 768$  pixels). The results are compared to those obtained by using JPEG, in terms of the PSNR for the luminance of the images vs. the bpp required to encode a full color pixel. It is observed that the adaptive system outperforms quantitatively JPEG along all the range of coding bit rate values. A visual evaluation of the quality of the reconstructed images, also shows a better performance of the adaptive algorithm over JPEG. As an example, Figure 4

shows a portion of the luminance band of the image Lena, which was compressed using both JPEG and the adaptive method at 0.3 bpp.

The combination of blocks of different sizes, in the proposed algorithm, produces reconstructed images with a *less uniform* block structure. This, in comparison with the fixed block-size system, in which the *periodic* appearance of the edges of the blocks, reinforce their visual effect, particularly in smooth areas.

The results reported in this section, were obtained with the standard deviation ( $\sigma$ ) as the activity measure for the block classification, using a threshold value of  $\sigma=7$  for the three image bands.

## 5 Computational Complexity

In accordance with Table 1, it is expected that the improvement of the compression efficiency of the adaptive system is obtained ineluctably in exchange for a high increase of the computational complexity. In effect, the most complex unit of the non-adaptive JPEG encoder is the 8-point DCT module, which in the adaptive scheme, must provide enough computational and memory resources to calculate an additional (even more complex) 16-point DCT. Furthermore, the block classification algorithm equally requires a non-negligible number of operations. In the following sections a strategy to reduce considerably the computational burden of the adaptive system is reported.

### 5.1 Reducing the complexity of the 2-D 16-point DCT over a 0 block

When a particular **B** block is classified as a **0** block, it can be inferred that this block of  $16 \times 16$  pixels corresponds to a smooth region of minimum image activity. In other words, it is a block whose array of  $16 \times 16$  DCT coefficients has its elements concentrated in the low frequency zone. In effect, the 16-point DCT coefficients of a large number of test images, including luminance and chrominance bands, were analyzed after the quantization stage. The result was that, for the threshold values of interest used for the block classification (i.e.,  $7 \leq \sigma \leq 10$ ), all the quantized DCT coefficients outside the frequency zone  $[(0, 0) \leq (U, V) \leq (7, 7)]$ , (in the transform domain  $U, V = 0, 1, \dots, 15$ ) were found to be zero, for all the test images.

Table 2 illustrates the results for the luminance of the image Airplane, at different coding bit rates and block classified with a value of  $\sigma=7$ . Within the zone  $[(U > U_{max}) \text{ OR } (V > V_{max})]$  all the quantized DCT coefficients are zero, for all the **0** blocks of the image.

Given that even at high coding bit rates, most of the coefficients remain within the zone

Range (bpp)	$U_{max}$	$V_{max}$
(0.140, 0.145)	1	1
(0.145, 0.156)	1	2
(0.156, 0.197)	2	2
(0.197, 0.210)	3	3
(0.210, 0.342)	3	4
(0.342, 0.486)	3	5
(0.486, 0.718)	4	6
(0.718, 1.167)	4	7
(1.167, 2.500)	7	7

Table 2: Zone of non zero coefficients for the **0** blocks of Airplane (Y):  $[(0, 0) \leq (U, V) \leq (U_{max}, V_{max})]$

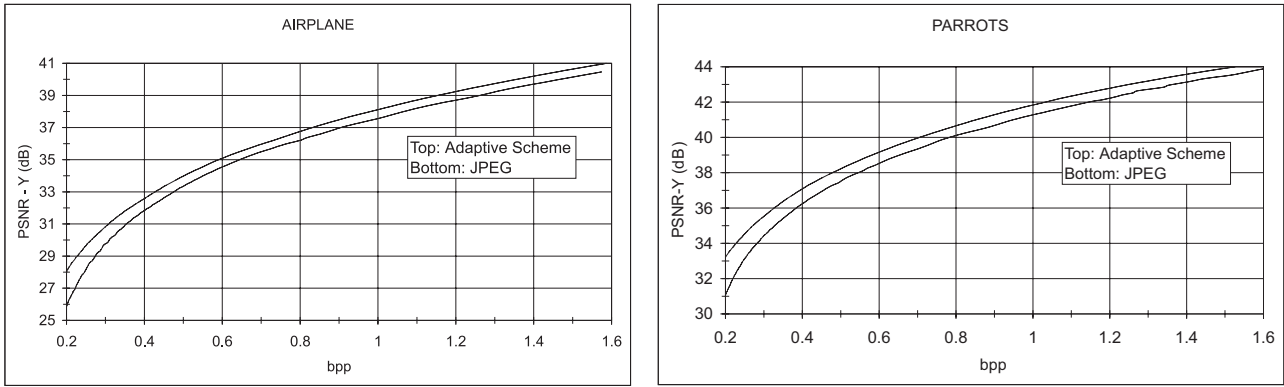
$[(U < 8, V < 8)]$  (for the chosen thresholds), it is not necessary to compute, nor to quantize, the 192 DCT coefficients lying outside this zone. This fact has a radical effect on the implementation of the system. The heavy computational overhead of a 16-point 2-D DCT is substantially reduced since only 64 coefficients are to be calculated, which allows a saving of 75% of the DCT computations. This saving of operations extends as well to the quantizer, where again, only 25% of the normalizations must be executed. The curves of the adaptive system shown in Figure 3 and the image in Figure 4(b) where generated by using this operation-saving strategy.

### 5.2 Reducing the complexity of the classification stage

The variance  $\sigma^2 = \frac{1}{N \cdot N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [x(i, j) - \bar{x}]^2$  (or its equivalent standard deviation) is a frequently used measure for estimating the degree of activity of the different regions of an image. All the same, its computational complexity is far from being negligible. Experimental results showed that without decreasing the efficiency of the algorithm, the complexity of the system can be further reduced by using the Mean Absolute Deviation, which is defined as  $MAD_{ev} = \frac{1}{N \cdot N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - \bar{x}|$ , for the block classification. A value of  $MAD_{ev} = 5.6$  was found by simulation as producing near identical results to those obtained by using a  $\sigma$  value of 7.0. Indeed, using these two values over a large number of images, showed that the two block classification bitmaps produced by both metrics match, in the worst case, in more than 95% of the bits.

## 6 Conclusions

In this paper a variable block-size transform-based coding scheme was reported along with its application to compress color images. The results presented show the improvement of the adaptive method over the



(a) Luminance Airplane

(b) Luminance Parrots

Figure 3: PSNR vs. bpp



(a) JPEG: 0.3 bpp



(b) Adaptive Scheme: 0.3 bpp

Figure 4: Portion of the image Lena (Y)

non-adaptive JPEG algorithm both in quantitative PSNR terms and very particularly, in visual quality of the reconstructed images. Furthermore, a strategy to reduce in a large amount the computational complexity of the algorithm was equally reported.

## 7 Acknowledgements

This work was supported by the Swiss Federal Office for Education and Science under Grant OFES C97.0050 (COST 254 Research).

## References

[1] J. Bracamonte, M. Ansorge, and F. Pellandini, "Adaptive Block-Size Transform Coding for Image

Compression", *Proc. IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, Vol. 4, Munich, Germany, Apr. 1997, pp. 2721–2724.

- [2] P.A. Wintz, "Transform Picture Coding", *Proc. of the IEEE*, Vol. 60, No. 7, July 1972.
- [3] P. Yip and K.R. Rao, "Energy packing efficiency for the generalized discrete transform", *IEEE Trans. on Communications*, Vol. 26, Aug. 1978, pp. 1257–1262.
- [4] ITU-T Recommendation 601. "Encoding Parameters of digital television for studios", 1982.
- [5] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, USA, 1993.
- [6] ITU-T Recommendation T.81 "Digital Compression and Coding of Continuous-tone Still Images", September, 1992.