IMAGE CODING BY THE MESH BLOCK TRANSFORM

E.A.B. da Silva and G.V. Mendonça

Universidade Federal do Rio de Janeiro
Departamento de Engenharia Eletrônica / EE / COPPE
CP. 68504 – 21945 – RJ – Brazil

ABSTRACT

A new method for image coding, the Mesh Block Transform (MBT) is proposed. It aims at reducing the well known blocking effects, artifacts very common in transform coding schemes. The approach used in this work is to divide an image of dimensions MN x MN into M^2 meshes of dimensions N x N. Adjacent pixels in a mesh are separated by M pixels in the image. Each mesh is transformed and the same frequency coefficients from all meshes are grouped into blocks. A second transform is then applied to these blocks and its coefficients are quantized. A mathematical analysis of the MBT is done and a coder is proposed. The performance of the method is evaluated.

I. INTRODUCTION

Transform coding is a very efficient approach used for image coding. In the classical transform coding schemes [1] an image is first divided into smaller subimages, or blocks, each one being coded independently. This approach, despite being very efficient, has one major drawback, the so called blocking effects. They consist of artificial boundary discontinuities being apparent between the subimages after the decoding process. Its main cause is the fact that the quantization errors made during the coding of one subimage are independent of the quantization errors in its neighboring blocks.

A remarkable fact about the above mentioned methods is that the redundancy that exists among the different blocks is not exploited. Several methods have been proposed in order to exploit this redundancy, as in [2]. However, most of these methods require a large computational complexity, which does not make them well suited to real time applications.

This paper describes the Mesh Block Transform (MBT) [6], which is an attempt to reduce the blocking effects by exploiting, to a certain extent, the redundancy among the subimages, with the advantage of being suitable to real time applications.

II. THE MBT

One alternative to exploit the redundancy between the subimages in a transform coding scheme is to apply a second transform to the coefficients of the transforms of the image, as it is done in the case of the Hierarchical Transforms [3]. The MBT is, to a certain extent, a Hierarchical Transform, except for the fact that the image is not divided at first into blocks, but into meshes. This means dividing an image of dimensions MN x MN into M^2 meshes of dimensions N x N. The pixel (i,j) from the mesh (k,l) is the pixel (Mi+k,Mj+l) in the image. A transform is then applied to each mesh. Afterwards, blocks are formed consisting each one of transform coefficients with same frequency, and a second transform is applied to each one of these blocks. Some facts can be stated about the MBT:

- The redundancy between any two pixels in the whole image is exploited, since the first transform reduces the redundancy inside the meshes and the second transform reduces the redundancy among the different meshes;
- As adjacent pixels in a mesh are separated by M pixels in the original image, the correlation among them is much lower than the correlation between adjacent pixels in the original image. So, it is expected that the coding gain obtained after the first transformation will not be high [4]. However, as each mesh is spread over the whole image, the meshes are reasonably alike. This means that the transform coefficients of the same frequency from adjacent meshes will have a high correlation. So, it is expected that a high coding gain will be obtained after the second transformation. Hence, the overall coding gain of the MBT will be high;
- As the image is not divided initially into blocks, but into meshes, the blocking effects will tend to be greatly reduced. Instead, "meashing effects" will appear, that can be less annoying than the blocking effects, or can be more suitable to treatment with some kind of post–processing;

In the next section, an analytical study will be done about the MBT, which will provide a deeper understanding of its properties.

III. ANALYTICAL STUDY

The calculation of the MBT of an one–dimensional vector $\mathbf{X}$ of dimension MN is equivalent to the following steps: 1) regroup the elements of the vector $\mathbf{X}$ into a vector $\overline{\mathbf{X}}$; 2)apply a block transform $\mathbf{A}$ of dimension M to every block of $\overline{\mathbf{X}}$ (the meshes of $\mathbf{X}$)

forming the vector $\overline{Y}$; 3) regroup the same frequency coefficients into blocks forming the vector **Y**; 4) apply a transform **B** of dimension N to each of its blocks forming the vector **Z**, which represents the MBT of the vector **X** [6].

In order to obtain $\overline{\mathbf{X}}$ the vector **X** has to be multiplied by the matrix $\mathbf{R}^{M,N}$ defined as:

$$\mathbf{R}^{M,N} = \begin{bmatrix} \mathbf{R}^{M,N}_{1,1} & \mathbf{R}^{M,N}_{2,1} \ldots \mathbf{R}^{M,N}_{M,1} \\ \mathbf{R}^{M,N}_{1,2} & \mathbf{R}^{M,N}_{2,2} \ldots \mathbf{R}^{M,N}_{M,2} \\ \vdots & \vdots \quad \ldots \vdots \\ \mathbf{R}^{M,N}_{1,N} & \mathbf{R}^{M,N}_{2,N} \ldots \mathbf{R}^{M,N}_{M,N} \end{bmatrix}, \quad \mathbf{R}^{M,N}_{k,l}(i,j) = \begin{cases} 1, & \begin{matrix} k=i \\ l=j \end{matrix} \\ & 1 \leq i \leq M \\ 0, & \begin{matrix} k \neq i \\ l \neq j \end{matrix} \\ & 1 \leq j \leq N \end{cases}$$

The vector $\overline{\mathbf{Y}}$, representing the transform **A** applied to each of the blocks of $\overline{\mathbf{X}}$ (meshes of **X**) is obtained as follows [6]:

$$\overline{\mathbf{Y}} = (\mathbf{I}_N \otimes \mathbf{A})\, \overline{\mathbf{X}} \Rightarrow \overline{\mathbf{Y}} = (\mathbf{I}_N \otimes \mathbf{A})\, \mathbf{R}^{M,N}\, \mathbf{X} \qquad (1)$$

where: $\otimes$ is the Kroenecker product [5]
$\mathbf{I}_N$ is the identity matrix of order N

It can be proved that the regrouping of the same frequency coefficients of the vector $\overline{\mathbf{Y}}$, forming the vector **Y**, is equivalent to [6] multiplying $\overline{\mathbf{Y}}$ by $(\mathbf{R}^{M,N})^t \Rightarrow$

$$\Rightarrow \text{ from Eq. (1)} \Rightarrow \mathbf{Y} = (\mathbf{R}^{M,N})^t (\mathbf{I}_N \otimes \mathbf{A})\, \mathbf{R}^{M,N} \mathbf{X} \qquad (2)$$

In [6] it is proved that equation (2) is equivalent to:

$$\mathbf{Y} = (\mathbf{A} \otimes \mathbf{I}_N)\mathbf{X} \qquad (3)$$

The application of the transform **B** to each block of the vector **Y** will lead to the vector **Z**, the MBT of the vector **X**. Following a reasoning similar to the one which led to equation (1), vector **Z** may be stated as:

$$\mathbf{Z} = (\mathbf{I}_M \otimes \mathbf{B})\mathbf{Y} = (\mathbf{I}_M \otimes \mathbf{B})(\mathbf{A} \otimes \mathbf{I}_N)\mathbf{X} \qquad (4)$$

It is known from [5] that $(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = (\mathbf{AC})\otimes(\mathbf{BD}) \Rightarrow$

$$\Rightarrow \mathbf{Z} = (\mathbf{A} \otimes \mathbf{B})\mathbf{X} \qquad (5)$$

From equation (5) it can be seen that the MBT of a vector **X** is equivalent to a transform $(\mathbf{A}\otimes\mathbf{B})$ applied to the whole vector, when a transform **A** is applied to its meshes and a transform **B** is applied to its blocks of same frequency cofficients.

The two–dimensional MBT of **X**, represented by the matrix of coefficients **Z**, can be defined as follows:

$$\mathbf{Z} = (\mathbf{A} \otimes \mathbf{B})\, \mathbf{X}\, (\mathbf{A} \otimes \mathbf{B})^t \qquad (6)$$

Also from equation (5) it can be proved that if **A** and **B** are unitary, the MBT is a unitary transform [6].

As is stated in equation (6), the MBT is equivalent to a full–frame transform. It means that its basis functions extend over the whole image. So, a quantization error in a coefficient is not restricted to a block, but is spread all over the image. This serves as an alternative explanation to the reduction of the blocking effects in the MBT.

In the next section, a coder based on the MBT will be described.

IV. A CODER FOR THE MBT

a) Transforms to be used

The first step to be followed is to determine which transforms are to be applied to the meshes and to the blocks (the optimum **A** and **B**). Insigth into this matter is given by an interesting fact about the MBT. If matrices $\hat{\mathbf{X}}$ and $\hat{\mathbf{Z}}$ of dimensions M x N are formed from the vectors **X** and **Z** by dividing each one into M vectors of equal length and making them the lines of the respective matrices [6], equation (5) is equivalent to:

$$\hat{\mathbf{Z}} = \mathbf{A}\, \hat{\mathbf{X}}\, \mathbf{B} \qquad (7)$$

It can be seen that $\hat{\mathbf{Z}}$ represents the coefficients of a bidimensional transform applied to the matrix $\hat{\mathbf{X}}$ in which the transform **B** is applied to the lines of $\hat{\mathbf{X}}$ and the transform **A** to its columns. This means that the transform **B** is applied to neighboring pixels in the original image (lines of $\hat{\mathbf{X}}$). This elects the Discrete Cosine Transform (DCT) as the transform **B**, which has a fast algorithm, that is closest to the optimal [1]. However, the optimum **A** has to be investigated, because it is applied to pixels that are separated by N pixels in the original image (columns of $\hat{\mathbf{X}}$). This was done by estimating the autocovariance function of the meshes, and using the scheme proposed by JAIN [7]. It has been shown that if both low and highly detailed images are considered, the DCT is the mesh transform closest to the optimum. However, if only highly detailed images are considered, the Discrete Sine Transform (DST) [1] is the best one. The coder described here was implemented using the DCT as the transform **A**.

b) Variances of the coefficients

In order to have the bit allocations, the variances of the coefficients must be estimated. As the MBT is a full frame transform, this estimation cannot be done on a per image basis, as it is done in most adaptive schemes that uses block transforms [1]. So, there must be a training set consisting of images that will be used in the estimation of the variances of the coefficients. It is important to point out that the coder developed is a non–adaptive one. This means that the variances which are estimated in the training set will be supposed to be the variances of the whole set of images.

The training set has to represent as accurately as possible the whole set of images that would be coded. This set has been chosen as consisting of a low detailed image (SOLIDS), a medium detailed image (BOAT) and a highly detailed image (BABOON). These are black and white, 8 bits/pixel images of dimensions 256 x 256. The meshes were chosen to be of dimensions 16 x 16.

In order to avoid dynamic range problems, each mesh transform coefficient is normalized by its standard deviation. The variance of the coefficient (i,j) of the mesh transform can be obtained by direct calculation on the meshes of the images belonging to the training set. The variances of the final coefficients of the MBT are also estimated by direct calculation on the training set [6].

c) Bit allocations

The bit allocations are done by using the log–variance formula [1]

$$b_i = \frac{M}{n} + 0.6\left[\log_2\sigma_i^2 - \frac{1}{n}\sum_{i=1}^{n}\log_2\sigma_i^2\right]$$

where M bits are allocated among n coefficients with the i–th having the total variance σ_i^2 and being allocated b_i bits.

It is important to point out that neither zonal nor threshold sampling is done. Instead, the log–variance formula is applied iteratively [6].

d) Quantizers

The DC coefficient is quantized with an 8 bit uniform quantizer. The AC coefficients are supposed to have Laplacian density functions [1]. If the number of bits assigned to an AC coefficient is less than or equal to 8, the quantizer used is the MAX quantizer [4]. If the number of bits is greater than 8, the quantizer used is the optimal uniform quantizer.

V. EXPERIMENTAL RESULTS

The coder performance was evaluated by coding six images. The images used as a training set are BABOON, BOAT and SOLIDS, and the images used as a verifying set are GIRL, LADY and LENA. Figure 1 shows the original BABOON and LENA images. These images were coded at a rate of 1 bit/pixel. The objective fidelity measure used was the signal to noise ratio (SNR), defined as:

$$SNR = 10\log_{10}\left[\frac{(255)^2}{e_{ms}}\right], \quad e_{ms} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(x_{ij}-x_{ij}^c)^2}{MN}$$

where x represents the original image and x^c the coded image.

After the decoding process, the SNRs of the images are described in table 1, and the images BABOON and LENA coded at 1 bit/pixel are presented in figure 2.

TABLE 1: SNRs of the images coded at 1 bit/pixel

image	SNR (dB)
BABOON	26.97
BOAT	22.79
SOLIDS	32.19
GIRL	30.27
LADY	28.95
LENA	29.75

In terms of SNR, the coder developed has not shown a very good performance if compared to adaptive transform coders [1]. However, examining the coded images, it can be seen that the blocking effects are hardly noticed. Unfortunately, these images present another artifact that is a king of granularity, which appears to be an additive noise. It is the "meshing effect" which has been mentioned in section II. Looking at the BABOON image it can noticed that, due to its highly detailed nature, the "meshing effect" is not much annoying. In addition, examining more closely the LENA image, it can be noted that the part of the image which has more details do not appear much degraded after the decoding process. This fact suggests that the MBT can be used successfully in order to code highly detailed images, which are hard to do with classical transform coding schemes.

VI. CONCLUSIONS

In order make a better coder using the MBT, the estimation of the variances of the final coefficients has to be improved. One possibility would be to use a certain degree of adaptivity. The fact that the MBT is a full–frame transform does not make adaptivity suitable for the method. However, its special structure can provide alternative ways to implement it. One way to do it would be to transmit, along with an image, the 256 variances of the coefficients of the transform of the meshes. This will improve the variance estimation, adapting the coder to some particular features of the images, and producing only a small overhead.

It is also interesting to investigate how post–processing via a linear filter can improve the quality of the images.

In spite of the low SNRs attained by this coder, it must be pointed out that, after some improvements, the MBT may be a good alternative to the classical transform coding schemes, specially when highly detailed images are concerned.