Image Compression Using Texture Modeling

Lahouari Ghouti, Ahmed Bouridane and Mohammad K. Ibrahim

Abstract— **We consider the problem of improving the performance of multiwavelets-based image coders through texture parametrization. Texture parametrization is designed to achieve higher compression rates while maintaining excellent visual image quality. Tradeoffs among these two conflicting goals, maximizing compression rate and minimizing distortion due to compression, are possible by taking into account the imperfections inherent to the human visual system (HVS). We present a statistical view of the texture parametrization using balanced multiwavelets and develop a hybrid image compression scheme. The statistical scheme leads to a new multiresolution-based texture parametrization relying on the accurate modeling of the marginal distribution of balanced multiwavelet coefficients using generalized Gaussian density (GGD). Furthermore, we show that the proposed texture parametrization scheme is computationally-efficient. The proposed hybrid codec can be seamlessly integrated in any embedded image coder while requiring minimal header data.**

I. Introduction

A. Motivations

Recent advances in image coding research have led to the emergence of the state-of-the-art image compression standard JPEG 2000. However, storage and transmission of digital images are still demanding for higher compression rates than ever due to stringent bandwidth requirements, and thus compressed images are often impaired by various types of artifacts such as blockiness, blur, ringing, etc. [1]. To improve the visual quality of compressed images, however, there must be an effective and simple method to make use of the imperfections inherent to the human visual system (HVS). As a result, there has been a growing interest in HVS-based image coding schemes. In this context, visually-lossless compression ratio, among several criteria, can be used to evaluate the performance of any image compression scheme. This ratio represents a quality criteria that significantly depends on the image being compressed, the compression scheme and the final viewing conditions. Compressing images at higher rates may introduce visual impairments in the compressed image. Because artifacts affecting image texture are immediately perceived by human observers and lossless encoding of texture affects the coding bitrate efficiency, it becomes clear that the textured regions should be encoded separately using specially adapted techniques [2]. Hence, instead of losslessly-encoding textured regions at the pixel level, which is usually computationally-expensive, one may use texture-based models to characterize the textured regions and then it would suffice to inform the image decoder about the model parameters only [1]. Then, the decoder uses the transmitted model parameters to generate a texture visually similar to the original one. Thus, the incorporation of the texture characterization stage would drastically reduce the coding complexity and costs.

L. Ghouti is with the School of Computer Science, Queen's University of Belfast, BT7 1NN, UK. Email: l.ghouti@qub.ac.uk

A. Bouridane is with the School of Computer Science, Queen's University of Belfast, BT7 1NN, UK. Email: A.Bouridane@qub.ac.uk

M. K. Ibrahim is with the Computer Engineering Department. King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia. Email: ibrahimm@ccse.kfupm.edu.sa

II. Existing Approaches

In this work we consider the problem of texture parametrization using a statistical approach. Our point is that, given only a low-level representation, statistical modeling provides a powerful tool to compensate for the image impairments due to compression by deceiving the HVS system. Considering texture modeling and image compression simultaneously enables us to design a hybrid image coder.

1) Stochastic Texture Modeling: Texture modeling involves analysis and synthesis steps. In the former step, texture is represented by a set of limited features. Using this set of features, the synthesis stage generates a texture that looks similar to the original one. Texture can be classified into two major classes, namely, 1) structural/deterministic texture and 2) stochastic texture [3]. While, the former can be described by a set of texture primitives and placement rules, stochastic processes can be used to model the latter [4]. The proposed work in this paper deals exclusively with stochastic texture. The orientation and frequency of texture are important clues to their discrimination [1]. Tuceryan and Jain [5] identify five major categories of features for texture identification; statistical, geometrical, structural, model-based, and signal processing features. A common denominator in most stochastic texture modeling schemes is that the textured image is submitted to a linear transform, filter, of filter bank followed by some energy measure [6]. These schemes operate directly on the gray-level information. Some schemes, such as those based on Fourier-, Wavelet- or Gabor-transforms, first apply a filter to the input image, and then work on the filtered versions of the image [6]. Textured regions in filtered images can be easily modeled as first-order processes because the filtering operation limits the information-content in each filtered image.

2) Texture-Based Image Coding: In hybrid image coding schemes, the encoder first identifies textured regions, which are then analyzed, to produce the model parameters or features. These latter are then transmitted to the decoder that produces a synthetic texture based on these features through the synthesis stage. Mallat and Froment [7] propose an image coder where edges are extracted and coded using dyadic wavelet transform and textured regions are coded separately by employing a subband-based wavelet transform. Neves and Mendonça [8] develop an improved version of the work in [7]. In [8], the

[1] Usually, the representation of the model parameters requires few bits only.

0-7803-8834-8/05/$20.00 ©2005 IEEE.

edge selection is based on the contour length and only a reference edge map is kept assuming that the edge locations would not move across scales. It is worth noting that no statistical or structural/deterministic model has been used to characterize the textured regions in the coding schemes presented in [7], [8]. Ryan et al. [9] propose an hybrid image coding scheme where the input image is segmented into textured and non-textured regions. The filtering-based proposed scheme operates in the wavelet domain. Textured regions are modeled by an auto-regressive (AR) model. A standard wavelet image coder is used to encode the non-textured regions. The proposed scheme encodes the texture segmentation map. The texture segmentation stages contributes to the increase of the overall coder complexity. The improved visual quality achieved justifies the increased complexity. The hybrid image coder proposed by Debure and Kubota [10] first estimates a residual image for a specific compression bitrate. Then, textured regions in the residual image are modeled in the spatial domain. Each pixel in the residual image is classified into one of N (typically 4) different texture classes. An AR model describes the texture in each of the N classes. Then, the encoder encodes the AR parameters and the map of N texture classes that are transmitted to the decoder.

III. Overview of the Proposed Scheme

A. Motivations

Most efficient wavelet-based image codecs perform coding of wavelet coefficients via bit-plane coding. In this class of coding, the most significant bits of the largest wavelet coefficients are first encoded. Then, progressively the large coefficients are refined and smaller coefficients are encoded as significant through two different passes, namely significance and refinement passes [11]. Embedded zero-tree (EZW) coding of wavelet coefficients was introduced by Shapiro [12]. EZW coding achieves its embedding via binary bit-plane coding of deadzone scalar quantizer indices [11]. Building on the success of EZW coding, SPIHT algorithm has several features of its predecessor [11]. In lossy compression using bit-plane coding, the largest wavelet coefficients are decoded and fully restored. However, coefficients with small magnitude will be zero-length encoded. Usually, textured regions are represented by such type of small-magnitude coefficients. Therefore, textured regions are affected by visual artifacts after decoding such as blurring. In the proposed coding scheme, small-magnitude wavelet coefficients will be considered as stochastic texture and therefore will be modeled using the parametric model discussed below.

B. Proposed Hybrid Image Codec

In the proposed texture-based image coder, the luminance channel of the input image is first decomposed using balanced multiwavelet decomposition. For statistically consistent model parameters, texture modeling is only applied to the first two levels of decomposition. Since texture segmentation is computationally-expensive, we propose a simple method to characterize textured regions in each subband using block-by-block processing. In this case, we group the subband coefficients into several blocks of specific size, say $n \times n$. Then, model parameters are generated for each block. In this way, the computational load is kept at minimal cost and less overhead data will be transmitted to the decoder. Within each subband block, wavelet coefficients with reduced bit-plane representation will be considered as stochastic texture and fitted to the parametric texture model. At the decoder side, synthetic texture, generated using the model parameters, will be inserted at locations where subband coefficients are zero-encoded, i.e., no information is available about the original coefficient magnitude.

1) Texture Modeling: In this work, subband coefficients, represented by the last three bit-planes only, are considered as stochastic texture. Therefore, coefficients below a specific threshold will be treated as texture. (1) defines stochastic texture:

$$|w(m,n)| < T_s \tag{1}$$

where $w(m,n)$ represent the subband coefficient at spatial location m,n and T_s represents the threshold below which subband coefficients may be considered as stochastic texture. It is worth noting that (1) considers only subbands at decomposition levels 1 and 2 only. The marginal densities of subband coefficients are good candidate to model stochastic texture. This choice is justified by the findings of recent psychological research on human texture perception where it is suggested that two homogeneous textures are often difficult to discriminate if they produce similar marginal distributions of responses from a bank of filters [13]. Because texture modeling basically characterizes the local histogram of the subband coefficient magnitudes, the size of the local blocks is a major factor that affects the overall performance of the proposed modeling approach. In this work, three different sizes were tested, full-subband, 16×16 and 32×32 blocks, respectively. Do and Vetterli [13] suggest that experiments show that a good probability density function (PDF) approximation for the marginal density of subband coefficients at a particular decomposition level and orientation produced by various type of wavelet transforms may be achieved by adaptively varying two parameters of the GGD model, which is defined as [13]:

$$p(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} e^{-\left(\frac{|x|}{\alpha}\right)^{\beta}} \tag{2}$$

where $\Gamma(\cdot)$ is the Gamma function, i.e., $\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt, \quad z > 0$. In (2), α is the standard deviation and β is the exponent that controls the shape of the GGD distribution. The GGD model contains the Gaussian and Laplacian PDFs as special cases, using $\beta = 2$ and $\beta = 1$, respectively. In this paper, we use a maximum likelihood (ML) estimator to estimate the model parameters. The ML solution of the parameter α is given by the following relations [13]:

$$\hat{\alpha} = \left(\frac{\beta}{L}\sum_{i=1}^{L}|w_i|^{\beta}\right)^{\frac{1}{\beta}} \tag{3}$$

where L is the number of subband coefficients in the subband block under analysis. The shape parameter β is the solution of the following transcendental equation [13]:

$$1 + \frac{\Psi(1/\hat{\beta})}{\hat{\beta}} - \frac{\sum |w_i|^{\hat{\beta}} \log |w_i|}{\sum |w_i|^{\hat{\beta}}} + \frac{\log\left(\frac{\hat{\beta}}{L}\sum |w_i|^{\hat{\beta}}\right)}{\hat{\beta}} = 0 \quad (4)$$

where $\Psi(\cdot)$ is the *digamma* function, i.e., $\Psi(z) = \frac{\Gamma'(z)}{\Gamma(z)}$. (4) can be solved numerically. Do and Vetterli [13] propose an effective determination of the parameter $\hat{\beta}$ using the Newton-Raphson iterative procedure.

2) Multiwavelts and Balanced Multiwavelets: Orthogonality is a desirable property for software/hardware implementation and symmetry provides comfort to image perception [1]. In the context of image processing applications, the following three properties are important: 1) orthogonality to ensure the decorrelation of subband coefficients, 2) symmetry (i.e., linear phase) to process finite length signals without redundancy and artifacts, and 3) finite-length filters for computational efficiency. However, most real scalar wavelet transforms fail to possess these properties simultaneously. To circumvent these limitations, multiwavelets have been proposed where orthogonality and symmetry are allowed to co-exist by relaxing the time-invariance constraint [14].

Multiwavelets: Multiwavelets may be considered as generalization of scalar wavelets. However, some important differences exist between these two types of multiresolution transforms. In particular, whereas wavelets have a single scaling $\phi(t)$ and wavelet function $\psi(t)$, multiwavelets may have two or more scaling and wavelet functions. In general, r scaling functions can be written using the vector notation $\Phi(t) = [\phi_1(t)\phi_2(t)\cdots\phi_r(t)]^T$, where $\Phi(t)$ is called the *multiscaling* function. In the same way, we can define the *multiwavelet* function using r wavelet functions as $\Psi(t) = [\psi_1(t)\psi_2(t)\cdots\psi_r(t)]^T$. The scalar case is represented by $r = 1$. Most of developed multiwavelet transforms use two scaling and wavelet functions, while r can take theoretically any value. Similar to scalar wavelets and for the case where $r = 2$, the multiscaling function satisfies the following two-scale equation:

$$\Phi(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} H_k \Phi(2t-k), \quad (5)$$

$$\Psi(t) = \sqrt{2}\sum_{k=-\infty}^{\infty} G_k \Phi(2t-k), \quad (6)$$

However, it should be noted that $\{H_k\}$ and $\{G_k\}$ are 2×2 matrix filters defined as:

$$H_k = \begin{bmatrix} h_0(2k) & h_0(2k+1) \\ h_1(2k) & h_1(2k+1) \end{bmatrix}, \quad (7)$$

$$G_k = \begin{bmatrix} g_0(2k) & g_0(2k+1) \\ g_1(2k) & g_1(2k+1) \end{bmatrix} \quad (8)$$

where $\{h_k(n)\}$ and $\{g_k(n)\}$ are the scaling and wavelet filter sequences such that $\sum_n h_k^2(n) = 1$ and $\sum_n g_k^2(n) = 1$ for $k = 1, 2$. The matrix elements in the filters, given by (7) and (8), provide more degrees of freedom than scalar wavelets. However, the multi-channel nature of multiwavelets yields a subband structure that is different from that using scalar wavelets.

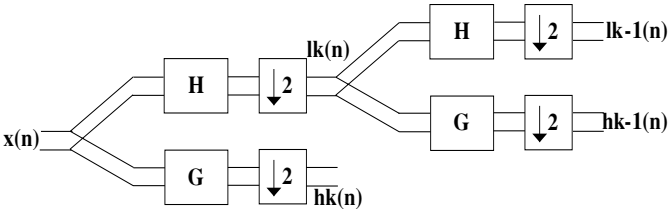

Fig. 1. Multiwavelet filter bank using one iteration.

Fig. 1 clearly shows that multiwavelets are defined for vector-valued signals (1D and 2D). Such *vectorizing* does not only introduce a fundamental asymmetry but it yields an approximation subband that does not represent a coarse approximation of the input image. The structure of the latter is different from that obtained using scalar wavelets. In the case of multiwavelets, the four sub-blocks of the approximation subband have very dissimilar spectral characteristics [14].

Balanced Multiwavelets: Lebrun and Vetterli [14] indicate that the balancing order of the multiwavelet is indicative of its energy compaction efficiency. However, a high balancing order alone does not ensure good image compression performance. For a scalar wavelet, the number of vanishing moments of its wavelet function $\int t^m \psi(t)dt = 0$ determines its vanishing order. For a scalar wavelet with vanishing order K, the highpass branch cancels a monomial of order less than K and the lowpass branch preserves it. For a multiwavelet transform, we have the similar notion of approximation order; a multiwavelet is said to have an approximation order of K if the vanishing moments of its wavelets, $\int t^m \psi_i(t)dt = 0$ for $0 \le i \le r-1$ and $0 \le m \le K-1$. An approximation order of K implies that the highpass branch cancels monomials of order less than K. But, in general, for multiwavelets, the preservation property does not automatically follow from the vanishing moments property. If the multifilter bank preserves the monomials at the lowpass branch output, the multiwavelet is said to be *balanced* [14]. The *balancing* order is p if the lowpass and highpass branches in the filter bank preserve and cancel, respectively all monomials of order less than p ($p \le K$). Multiwavelets that do not satisfy the preservation/cancellation property are said to be *unbalanced*. For unbalanced multiwavelets, the input needs suitable *prefiltering* to compensate for the absence of the preservation/cancellation property, balancing obviates the need for input prefiltering; thus, they are computationally more efficient than the unbalanced multiwavelets. A time-varying representation of balanced multiwavelets is shown in Fig. 2.

The time-varying filter bank of Fig. 2 is defined by the following relations:

$$\begin{bmatrix} H_0(z) \\ H_1(z) \end{bmatrix} = H(z^2)\begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \quad (9)$$

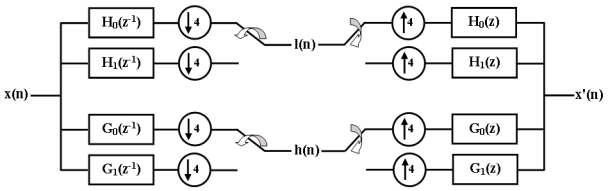

Fig. 2. Time-varying multiwavelet filter bank. Analysis stage (left). Synthesis stage (right).

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = G(z^2) \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \tag{10}$$

where $H_0(z)$ and $H_1(z)$ are the Z-transforms of the two lowpass branch filters h_0 and h_1. Similarly, $G_0(z)$ and $G_1(z)$ are the Z-transforms of the two highpass branch filters g_0 and g_1. In the time-varying filter bank implementation, the coefficients of the two lowpass (highpass) filters are simply interleaved at the output (see Fig. 2). Therefore, a separable 2D transform can now be defined in the usual way as the tensor product of two 1D transforms [14]. It is worth noting that unlike their unbalanced counterparts, the sub-blocks of the approximation subband of balanced multiwavelets have similar spectral characteristics as shown in Fig. 3 for the case of Lena image.

Fig. 3. Balanced multiwavelet approximation subband of Lena image (left). Spectral densities of subband blocks L_0L_0, L_0L_1, L_1L_0 and L_1L_1 (right).

IV. Simulation Results

Image compression experiments using balanced multiwavelets were conducted both with and without texture modeling. Several balanced multiwavelets with diverse characteristics are described in the literature [14]. In this paper, we adopt the SPIHT zerotree quantizer [11]. In this work, many high resolution images were used to test the performance of the proposed hybrid image codec. Each of these images is compressed at 0.1 and 0.2 bit per pixel (bpp) with the plain SPIHT algorithm and the hybrid SPIHT-based scheme using 5 decomposition levels using BAT-1 balanced multiwavelet [14]. We report results for Woman-43 image only. On the other hand, results using the proposed hybrid codec, as shown for the same image in Fig. 5, clearly demonstrate the improvement achieved in the visual quality of the compressed images.

Acknowledgment: L. Ghouti and M. K. Ibrahim acknowledge the support of KFUPM University.

Fig. 4. Compressed Woman-43 image at 0.2 bpp using the plain modified SPIHT coder.

Fig. 5. Compressed Woman-43 image at 0.2 bpp using the hybrid modified SPIHT coder.

References

[1] S. Mallat, *A Wavelet Tour of Signal Processing*, Second Edition, Academic Press, 1999.

[2] M. Kocher and M. Kunt, "Image data compression by contour texture modeling," in *SPIE Proceedings*, vol. 397, pp. 132-139, 1983.

[3] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786-804, May 1979.

[4] J. S. D. Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," In *Computer Graphics*, pp. 361-368, ACM SIGGRAPH, 1997.

[5] M. Tuceryan and A. K. Jain, "Texture Analysis," *Handbook Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, eds., ch. 2, pp. 235-276. Singapore: World Scientific, 1993.

[6] T. Randen and J. H. Husøy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Recognit. Machine Intell.*, vol. 21, no. 4, pp. 291-310, April 1999.

[7] S. Mallat and J. Froment, "Second Generation Compact Image Coding with Wavelets," *in Wavelets - A Tutorial in Theory and Applications (C. Chui Ed.)*, pp. 655-678, Academic Press, 1992.

[8] S. R. Neves and G. V. Mendonça, "Image Coding Based on Edges and Textures Via Wavelet Transform," in *Proc. IEEE Int. Conf. ICASSP' 98*, vol. 5, pp. 2689-2692, 1998.

[9] T. W. Ryan, D. Sanders, H. D. Fisher, and A. E. Iverson, "Image compression by texture modeling in the wavelet domain," *IEEE Trans. Image Processing*, vol. 5, no. 1, pp. 26-36, Jan. 1996.

[10] K. Debure and T. Kubato, "Autoregressive texture segmentation and synthesis for wavelet image compression," in *Proc. of Image and Multidimenional Digital Signal Processing '98*, pp. 131-134, July 1998.

[11] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuit Systems for Video Tech.*, vol. , no. , pp. 243-250, June 1996.

[12] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[13] M. N. Do and M. Vetterli, "Wavelet-Based Texture Retrievial Using Generalized Gaussian Density and Kullback-Leibler Distance," *IEEE Trans. Image Processing*, vol. 11, no. 2, pp. 146-158, Feb. 2002.

[14] J. Lebrun and M. Vetterli, "Balanced multiwavelets: Theory and design," *IEEE Trans. Signal Processing*, vol. 46, no. 4, pp. 1119-1125, Apr. 1998.