Image Modeling with Parametric Texture Sources for Design and Analysis of Image Processing Algorithms

Chuo-Ling Chang and Bernd Girod

Stanford University, Stanford, California, USA *

ABSTRACT

A novel statistical image model is proposed to facilitate the design and analysis of image processing algorithms. A mean-removed image neighborhood is modeled as a scaled segment of a hypothetical texture source, characterized as a 2-D stationary zero-mean unit-variance random field, specified by its autocorrelation function. Assuming that statistically similar image neighborhoods are derived from the same texture source, a clustering algorithm is developed to optimize both the texture sources and the cluster of neighborhoods associated with each texture source. Additionally, a novel parameterization of the texture source autocorrelation function and the corresponding power spectral density is incorporated into the clustering algorithm. The parametric autocorrelation function is anisotropic, suitable for describing directional features such as edges and lines in images. Experimental results demonstrate the application of the proposed model for designing linear predictors and analyzing the performance of wavelet-based image coding methods.

Keywords: image model, texture clustering, parametric autocorrelation, parametric image spectra

1. INTRODUCTION

A model that captures the important attributes of a signal is an essential tool to analyze the performance of signal processing algorithms. A good model not only provides accurate predictions of the actual performance, but also enables calculation of theoretical bounds and gives insights to further improvements of the algorithm.

For the design and analysis of image processing algorithms, two classic statistical image models are widely adopted: the separable model[1] and the isotropic model.[2,3] In both models, image pixels are assumed to be samples of a filtered image source, with the source modeled as a 2-D stationary random field. Although the autocorrelation function in both models is composed of a simple exponential function, the contours of equal autocorrelation are rhombi in the former, with the diagonals aligned with the vertical and horizontal axes, whereas they are circles in the latter.

Despite their simplicity and popularity, we observed two weaknesses in these models. First, they assume homogeneous texture in the image such that a global model suffices for the entire image. Such a global model neglects the diversity of individual image features, hence incapable of evaluating spatially adaptive algorithms. Second, directional features such as edges and lines appear frequently in typical images. The high-contrast nature of these features affects the performance of image processing algorithms significantly for both objective and subjective assessments. However, these features are usually neither separable nor isotropic.

More recently, Ref. 4 and Ref. 5 proposed an anisotropic model of image spectra that accounts for the directionality in images. Still, being a global model, it is not suitable for the design and analysis of adaptive algorithms.

In this paper, we devise a clustering algorithm that partitions the image into clusters of image local neighborhoods with statistical similarity. The mean-removed image neighborhoods in the same cluster are modeled as scaled segments of a hypothetical texture source, characterized as a 2-D stationary zero-mean unit-variance random field. In addition, we propose a novel parameterization of the autocorrelation function and the power spectral density of the texture sources. The parametric formulation closely approximates the statistical properties observed in a variety of image features. Furthermore, the parameterization process is incorporated into the clustering algorithm to optimize both the texture source parameters and the corresponding partition.

*Please send correspondence to chuoling@stanford.edu.

In the remainder of this paper, we first describe the proposed clustering algorithm in Sec. 2. The parametric formulation is discussed in Sec. 3. Experimental results with the application of designing linear predictors and analyzing the performance of wavelet-based image coding methods are reported in Sec. 4.

2. IMAGE MODELING WITH TEXTURE SOURCE CLUSTERING

2.1 Modeling with Sampling Covariance Matrices

Denote the image pixels by $s[\mathbf{l}] = s[l_x, l_y]$, $l_x, l_y \in \mathbb{Z}$. For any square neighborhood of $S_n \times S_n$ pixels in the image, we rearrange the pixels in the neighborhood to a vector $\mathbf{n_l} \in \mathbb{R}^{S_n^2}$, where $\mathbf{l}$ indicates the location of the center pixel of the neighborhood, and $\mathbf{n}_\mathbf{l}^{(u)} = s[\mathbf{l} + i(u)]$, where $i(u)$ maps the index of an element in $\mathbf{n_l}$ to a 2-D shift from the neighborhood center.

Additionally, divide the image into L_B disjoint blocks of $S_B \times S_B$ pixels, each denoted by $\mathcal{B}_b$, $b = 0, \cdots, L_B - 1$. We define a block of neighborhoods (BoN), denoted by $\mathcal{N}_b$, as the set of neighborhoods with their center pixels contained in $\mathcal{B}_b$, i.e., $\mathcal{N}_b = \{\mathbf{n_l} | \mathbf{l} \in \mathcal{B}_b\}$. Assuming the neighborhoods in $\mathcal{N}_b$ are all realizations of the same random vector, the mean and covariance matrix of this random vector can then be estimated by the sample mean, $\boldsymbol{\mu}_b = \frac{1}{|\mathcal{B}_b|} \sum_{\mathbf{l} \in \mathcal{B}_b} \mathbf{n_l} \in \mathbb{R}^{S_n^2}$, and the sample covariance, $\mathbf{C}_b = \frac{1}{|\mathcal{B}_b|} \sum_{\mathbf{l} \in \mathcal{B}_b} \mathbf{n_l} \mathbf{n_l}^T - \boldsymbol{\mu}_b \boldsymbol{\mu}_b^T \in \mathbb{R}^{S_n^2 \times S_n^2}$, respectively.

Consider an arbitrary 2-D linear filter with a support smaller than the $S_n \times S_n$ neighborhood. The filter kernel can be expressed by $\mathbf{h} \in \mathbb{R}^{S_n^2}$ such that when applied to a neighborhood $\mathbf{n_l}$, the filter output is $\mathbf{h}^T \mathbf{n_l}$. As a result, the average energy of the filter output located in block $\mathcal{B}_b$ is $\mathbf{h}^T (\mathbf{C}_b + \boldsymbol{\mu}_b \boldsymbol{\mu}_b^T) \mathbf{h}$. Therefore, modeling a BoN by $\boldsymbol{\mu}_b$ and $\mathbf{C}_b$ yields exact prediction of linear image processing algorithms in terms of the average output energy. However, there are $(S_n^4 + S_n^2)/2$ unique elements in $\mathbf{C}_b$. As an example, for $S_B = 16$ and a small neighborhood $S_n = 5$, there are already 325 unique elements in $\mathbf{C}_b$, comparable to the number of pixels involved in the BoN, hence defeating the purpose of using a model to predict the performance.

2.2 Modeling with Stationary Texture Sources

To simplify the model, we first define the scalar sample mean of $\mathcal{N}_b$ by $\bar{\mu}_b = \frac{1}{S_n^2} \boldsymbol{\mu}_b^T \mathbf{1}$. With $\bar{\mu}_b$, we then define the function $v_\mathbf{l}[\mathbf{d}] = v_\mathbf{l}[d_x, d_y]$, $d_x, d_y \in \mathbb{Z}$ and $|d_x|, |d_y| < S_n$, to be the average product between two $\bar{\mu}_b$-removed pixels in neighborhood $\mathbf{n_l}$ separated by $\mathbf{d}$, i.e.,

$$v_\mathbf{l}[\mathbf{d}] = v_\mathbf{l}[-\mathbf{d}] = \frac{1}{|\{(u_1, u_2) | i(u_1) - i(u_2) = \mathbf{d}\}|} \sum_{\{(u_1,u_2) | i(u_1) - i(u_2) = \mathbf{d}\}} (\mathbf{n}_\mathbf{l}^{(u_1)} - \bar{\mu}_b)(\mathbf{n}_\mathbf{l}^{(u_2)} - \bar{\mu}_b). \tag{1}$$

Consequently, we define the sample autocovariance function of $\mathcal{N}_b$ by $v_b[\mathbf{d}] = \frac{1}{|\mathcal{B}_b|} \sum_{\mathbf{l} \in \mathcal{B}_b} v_\mathbf{l}[\mathbf{d}]$. When modeling $\mathcal{N}_b$ by $\bar{\mu}_b$ and $v_b[\mathbf{d}]$ as an approximation of $\boldsymbol{\mu}_b$ and $\mathbf{C}_b$ respectively, every element in $\boldsymbol{\mu}_b$ is approximated by $\bar{\mu}_b$, and all of the elements $\mathbf{C}_b^{(u_1,u_2)}$ such that $i(u_1) - i(u_2) = \mathbf{d}$ or $i(u_1) - i(u_2) = -\mathbf{d}$ are approximated by $v_b[\mathbf{d}]$.

Modeling a BoN by the scalar sample mean and the sample autocovariance function essentially assumes that all neighborhoods in the BoN are scaled and offset (in pixel intensity) spatial segments of the realizations of a 2-D discrete-space texture source, modeled as a discrete-space stationary zero-mean unit-variance random field. To model a BoN with mean $\bar{\mu}_b$ and sample autocovariance $v_b[\mathbf{d}]$, the corresponding scaling is $\sqrt{v_b[\mathbf{0}]}$, the offset is $\bar{\mu}_b$, and the autocorrelation function of the texture source is $v_b[\mathbf{d}]/v_b[\mathbf{0}]$, defined for $|d_x|, |d_y| < S_n$. Note that the number of unique elements in $v_b[\mathbf{d}]$ is $2S_n^2 - 2S_n + 1$, significantly less than that in $\mathbf{C}_b$.

2.3 Texture Source Clustering

To further simplify the model, we allow multiple BoNs to be derived from the same texture source while each BoN still having its own mean and variance. Assume the BoNs are grouped into L_C clusters, and the BoNs in the same cluster are derived from the same texture source. Each cluster is denoted by $\mathcal{C}_c = \{\mathcal{N}_b | m(b) = c\}$, $c = 0, \cdots, L_C - 1$, where $m(b)$ is a membership function that assigns $\mathcal{N}_b$ to cluster $\mathcal{C}_{m(b)}$, and each cluster is associated with the autocorrelation function of its texture source, $r_c[\mathbf{d}]$. Note that $r_c[\mathbf{0}] = 1$. Given $m(b)$ and $r_c[\mathbf{d}]$, a BoN, $\mathcal{N}_b$, is now modeled by its mean, $\bar{\mu}_b$, and the scaled texture autocorrelation, $v_b[\mathbf{0}] r_{m(b)}[\mathbf{d}]$,

rather than the sample autocovariance function, $v_b[\mathbf{d}]$. To quantify the approximation error between $v_b[\mathbf{d}]$ and $v_b[\mathbf{0}]r_{m(b)}[\mathbf{d}]$, we define the distortion between the two functions by

$$d(v_b, r_c) = \sum_{|d_x|,|d_y|<S_n} w[\mathbf{d}](v_b[\mathbf{d}] - v_b[\mathbf{0}]r_c[\mathbf{d}])^2 = k_{0,b} + \sum_{|d_x|,|d_y|<S_n} (-2k_{1,b}[\mathbf{d}]r_c[\mathbf{d}] + k_{2,b}[\mathbf{d}]r_c[\mathbf{d}]^2) \tag{2}$$

where $k_{0,b} = \sum_{|d_x|,|d_y|<S_n} w[\mathbf{d}]v_b[\mathbf{d}]^2$, $k_{1,b}[\mathbf{d}] = v_b[\mathbf{0}]w[\mathbf{d}]v_b[\mathbf{d}]$, $k_{2,b}[\mathbf{d}] = v_b[\mathbf{0}]^2 w[\mathbf{d}]$, and $w[\mathbf{d}]$ depends on the linear processing algorithm of interest. For some linear filters, the energy of the filter output calculated via the model may be more sensitive to the approximation error in some of the autocovariance values, hence requiring larger weights for these values. For modeling of the image neighborhoods in general without targeting on a specific algorithm, we simply consider $w[\mathbf{d}] = 1$.

In the following discussion, we describe a clustering algorithm that optimizes $m(b)$ and $r_c[\mathbf{d}]$ for a given set of $v_b[\mathbf{d}]$, $b = 0, \cdots, L_B - 1$, such that the overall distortion, $\sum_{b=0,\cdots,L_B-1} d(v_b, r_{m(b)})$, is minimized. We begin the algorithm by initializing $m(b)$, for instance, via evenly partitioning the BoNs into L_C clusters. Given the initialized $m(b)$, the set of $r_c[\mathbf{d}]$ that minimizes the overall distortion is determined by

$$r_c[\mathbf{d}] = r_c[d_x, d_y] = \frac{\sum_{b|m(b)=c} k_{1,b}[\mathbf{d}]}{\sum_{b|m(b)=c} k_{2,b}[\mathbf{d}]}, \ \forall\ |d_x|, |d_y| < S_n, \ \ c = 0, \cdots, L_C - 1. \tag{3}$$

Given this set of $r_c[\mathbf{d}]$, the $m(b)$ can be modified to further decrease the overall distortion. The optimal $m(b)$ is determined by

$$m(b) = \underset{c=0,\cdots,L_C-1}{\operatorname{argmin}} d(v_b, r_c), \ \forall\ b = 0, \cdots, L_B - 1. \tag{4}$$

The algorithm continues by iterating between (3) and (4) until the overall distortion converges to a local minimum.

3. PARAMETRIC TEXTURE SOURCES

In this section, we propose a parametric formulation for the autocorrelation function of the texture sources. This parameterization serves several purposes. First, it significantly reduces the amount of information required to represent the autocorrelation values. For image communication applications where the model needs to be signaled as side information, this reduction is essential to avoid the excess overhead. Second, it extrapolates from the limited set of observed sample covariance values to provide the full autocorrelation function and the corresponding power spectral density (PSD) of the texture source. As a result, it enables performance analysis of linear filtering with a support larger than the neighborhood size. Third, the proposed parametric texture autocorrelation is a positive definite function, i.e., any covariance matrix derived from it is positive definite without the need of additional regularization. These purposes will be elaborated in the following discussion.

3.1 Parametric Continuous-Space Autocorrelation Function

We assume that a discrete-space texture source, modeled as a 2-D discrete-space stationary zero-mean unit-variance random field $t[\mathbf{l}] = t[l_x, l_y]$ (Section 2.2), consists of samples of the filtered and scaled version of a continuous-space texture source, modeled as a 2-D continuous-space stationary zero-mean unit-variance random field $\tilde{t}(\mathbf{p}) = \tilde{t}(p_x, p_y)$. The filtering accounts for the aperture function of the imaging device, and the scaling normalizes the discrete-space random field so that it is also unit-variance.

In this subsection, we propose a parametric formulation for the autocorrelation function and the PSD of the continuous-space texture source, denoted by $r_{\tilde{t}\tilde{t}}(\boldsymbol{\tau}) = r_{\tilde{t}\tilde{t}}(\tau_x, \tau_y)$ and $\Phi_{\tilde{t}\tilde{t}}(\boldsymbol{\omega}) = \Phi_{\tilde{t}\tilde{t}}(\omega_x, \omega_y)$ respectively. The corresponding parametric autocorrelation function and PSD of the discrete-space texture source is derived in Section 3.2.

To devise a parameterization that describes a variety of sample autocovariance functions observed in typical image neighborhoods, we define two types of local image features: the regular features, such as an edge, a line (two edges), and stripes (periodic edges); and the irregular features, accounting for other, usually more complex, features.

For simplicity, we first consider the 1-D case. In the locally mean-removed 1-D neighborhoods with textures that are complex and lacking in dominant regular features, zero-crossing occurs in an irregular manner, thus classified as irregular features. Note that even a flat region in the image appears as irregular features after local mean-removal. This zero-crossing pattern is similar to that of a mean-removed random telegraph signal, which is a bi-level signal with the changes in level (zero-crossing) occurring according to a Poisson process. It can be shown that the autocorrelation function of a zero-mean unit-variance random telegraph signal is $\exp(-2\lambda_p\tau)$ where λ_p is the rate of the underlying Poisson process. Due to this similarity, we assume that the texture autocorrelation of the irregular features generally decays exponentially with the distance between two points, consistent with the exponential function in both the separable model[1] and the isotropic model.[2,3]

For the regular features, however, this assumption does not hold. Consider a group of 1-D mean-removed horizontal neighborhoods that intersects the same vertical line in the image. In this example, there are always two zero-crossings, with a fixed separation, in every neighborhood. This fixed zero-crossing pattern can no longer be approximated by a Poisson random process. Instead, we approximate the texture source of these neighborhoods by a random square-wave signal, defined as

$$\tilde{s}(p|T_s, r_s) = \begin{cases} A_0, & \exists k \in \mathbb{Z} \text{ s.t. } 0 \le p + kT_s + \Theta < r_s T_s \\ A_1, & \text{otherwise,} \end{cases} \tag{5}$$

where T_s is the period, $0 < r_s < 1$ determines the duration at the two levels, and Θ is a random phase. Continuing with the above example, the neighborhoods containing a line can be considered as segments of a random square-wave signal with T_s close to the width of the 1-D neighborhood, and r_s selected according to the width of the line. Similarly, the neighborhoods containing a single dominant edge are segments of a random square-wave signal with T_s close to twice the neighborhood width and $r_s = 0.5$ so that an edge appears only once in every neighborhood. Modeling the neighborhoods containing periodic stripes is straightforward. Finally, it can be shown that the autocorrelation function of a zero-mean unit-variance random square-wave signal is

$$r_{\tilde{s}\tilde{s}}(\tau|T_s, r_s) = \frac{2r_s}{1 - r_s} \sum_{k=1}^{\infty} sinc(kr_s)^2 cos(k\frac{2\pi}{T_s}\tau). \tag{6}$$

Therefore, we assume that the texture source of regular features has a parametric autocorrelation function with the same form. Note that $r_{\tilde{s}\tilde{s}}(\tau|T_s, r_s) = r_{\tilde{s}\tilde{s}}(\tau|T_s, 1 - r_s)$.

Based on the above discussion, we propose the following parameterization for the texture autocorrelation:

$$r_{\tilde{t}\tilde{t}}(\boldsymbol{\tau}|\beta, \theta, \lambda_a, \lambda_b, \omega_s, r_s) = \beta \exp(-(\lambda_a^2\tau_a^2 + \lambda_b^2\tau_b^2)^{\frac{1}{2}}) + (1 - \beta) \sum_{k=1}^{K} \gamma(k, r_s) cos(k\omega_s\tau_a) \tag{7a}$$

$$\begin{pmatrix} \tau_a \\ \tau_b \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \boldsymbol{\tau} \tag{7b}$$

$$\gamma(k, r_s) = \frac{sinc(kr_s)^2}{\sum_{k=1}^{K} sinc(kr_s)^2} \tag{7c}$$

where $0 \le \beta \le 1$, $0 \le \theta < \pi$, $0 < \lambda_b \le \lambda_a$, $\omega_s > 0$, and $0 < r_s \le \frac{1}{2}$. The first term is responsible for the irregular features, while the second describes the regular ones. Taking a weighted average of the two terms by β explicitly assumes that the texture source is composed of an irregular texture and a regular texture component, and the two components are uncorrelated.

The first term is an extension of the isotropic model[2,3] to allow some directionality in the irregular texture. The angle θ indicates the direction of the largest variation in the texture source. The autocorrelation function is $\exp(-\lambda_a\tau_a)$ along θ, whereas it is $\exp(-\lambda_b\tau_b)$ along the direction orthogonal to θ, decaying at a smaller or equal rate. The autocorrelation along other directions are also derived from λ_a and λ_b such that the contours of equal autocorrelation are concentric ellipses where the minor axis is aligned with θ. Additionally, the length of the minor axis and that of the major axis are proportional to λ_a^{-1} and λ_b^{-1} respectively.

In the second term, the tangent of the edges and lines in the regular texture aligns with θ, sharing the same direction of the largest variation with the irregular texture for simplicity. Along θ, the second term is essentially the autocorrelation of the random square-wave signal in (6) with $\omega_s = \frac{2\pi}{T_s}$ and the infinite sum approximated by the sum of the first K harmonics. In practice, we choose $K = 3$. Orthogonal to θ, the autocorrelation is constant. Note that the incorporation of the regular texture enables modeling of negative autocorrelation values, which cannot be described by the irregular component.

The corresponding parametric PSD can be expressed as

$$\Phi_{\tilde{t}\tilde{t}}(\boldsymbol{\omega}|\beta,\theta,\lambda_a,\lambda_b,\omega_s,r_s) = \beta\frac{2\pi}{\lambda_a\lambda_b}(1+(\frac{\omega_a}{\lambda_a})^2+(\frac{\omega_b}{\lambda_b})^2)^{-\frac{3}{2}} \tag{8a}$$

$$+(1-\beta)\sum_{k=1}^{K}\gamma(k,r_s)\pi(\delta(\omega_a-k\omega_s)+\delta(\omega_a+k\omega_s)) \tag{8b}$$

$$\begin{pmatrix}\omega_a\\ \omega_b\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}\boldsymbol{\omega} \tag{8c}$$

In the irregular texture, the contours of equal PSD are also concentric ellipses with the major axis aligned to θ. The length of the major axis and that of the minor axis are proportional to λ_a and λ_b respectively. The irregular texture appears as pairs of delta impulses along θ in the PSD, accounting for the off-center peaks frequently observed in the spectrum estimation from the discrete-space samples, e.g., the periodogram estimation.

3.2 Continuous-Space to Discrete-Space Conversion

Denote the aperture function of the imaging device by $\tilde{h}_a(\mathbf{p})$ and the corresponding transfer function by $\tilde{H}_a(\boldsymbol{\omega})$. The ideal aperture function is the ideal anti-aliasing filter with the cut-off frequency at $\omega_x = \pm\pi f_{sx}$ and $\omega_y = \pm\pi f_{sy}$, where f_{sx} and f_{sy} denote the horizontal and the vertical sampling frequency of the discrete-space texture source respectively. For practical imaging devices, we model the aperture function as a separable filter, i.e., $\tilde{H}_a(\boldsymbol{\omega}) = \tilde{H}_{a,1}(\omega_x|f_{sx})\tilde{H}_{a,1}(\omega_y|f_{sy})$, and the 1-D transfer function $\tilde{H}_{a,1}(\omega|f_s)$ is modeled as a root-raised-cosine filter with the roll-off factor $\beta_r = 0.5$, i.e.,

$$|\tilde{H}_{a,1}(\omega|f_s)|^2 = \begin{cases}\frac{1}{f_s}, & |\omega| \le (1-\beta_r)\pi f_s\\ \frac{1}{2f_s}(1+cos(\frac{1}{2\beta_r f_s}(|\omega|-(1-\beta_r)\pi f_s))), & (1-\beta_r)\pi f_s < |\omega| \le (1+\beta_r)\pi f_s\\ 0, & \text{otherwise,}\end{cases} \tag{9}$$

hence the corresponding 1-D impulse response, $\tilde{h}_{a,1}(p|f_s)$, is defined such that

$$\tilde{h}_{a,1}(p|f_s) * \tilde{h}_{a,1}(-p|f_s) = (1-4\beta_r^2 f_s^2 p^2)^{-1}\cos(\beta_r\pi f_s p)\cdot sinc(f_s p) \tag{10}$$

Given $r_{\tilde{t}\tilde{t}}(\boldsymbol{\tau})$ and $\Phi_{\tilde{t}\tilde{t}}(\boldsymbol{\omega})$ as in (7) and (8), the autocorrelation function and the PSD of the filtered source, denoted by $\hat{t}(\mathbf{p})$, can be obtained by

$$r_{\hat{t}\hat{t}}(\boldsymbol{\tau}) = r_{\tilde{t}\tilde{t}}(\boldsymbol{\tau}) * \tilde{h}_a(\boldsymbol{\tau}) * \tilde{h}_a(-\boldsymbol{\tau}) \tag{11a}$$

$$\Phi_{\hat{t}\hat{t}}(\boldsymbol{\omega}) = \Phi_{\tilde{t}\tilde{t}}(\boldsymbol{\omega})\cdot|\tilde{H}_a(\boldsymbol{\omega})|^2 \tag{11b}$$

Note that $\Phi_{\hat{t}\hat{t}}(\boldsymbol{\omega})$ is now band-limited with the aperture function defined in (9). Finally, the autocorrelation function of the discrete-space unit-variance texture source is obtained by

$$r_{tt}[\mathbf{d}] = r_{tt}[d_x,d_y] = \frac{1}{r_{\hat{t}\hat{t}}(\mathbf{0})}r_{\hat{t}\hat{t}}(\frac{d_x}{f_{sx}},\frac{d_y}{f_{sy}}) \tag{12}$$

and the corresponding PSD is

$$\Phi_{tt}(e^{j\boldsymbol{\Omega}}) = \Phi_{tt}(e^{j\Omega_x},e^{j\Omega_y}) = \frac{1}{r_{\hat{t}\hat{t}}(\mathbf{0})}\sum_{z_x=-1}^{1}\sum_{z_y=-1}^{1}\Phi_{\hat{t}\hat{t}}((\Omega_x-2z_x\pi)f_{sx},(\Omega_y-2z_y\pi)f_{sy}) \tag{13}$$

3.3 Parameter Estimation

Parameter estimation can be incorporated into the texture source clustering algorithm described in Section 2.3 by replacing (3) with

$$r_c[\mathbf{d}] = \underset{r_{tt}}{\operatorname{argmin}} \sum_{\{b|m(b)=c\}} d(v_b, r_{tt}|\beta, \theta, \lambda_a, \lambda_b, \omega_s, r_s),\ \forall\ c = 1, \cdots, L_C. \tag{14}$$

To reduce the parameter space, we consider $f_{sx} = f_{sy} = f_s$ and discretize β into 256 levels linearly from 0 to 1, θ into 128 levels linearly from 0 to $\frac{127}{128}\pi$, λ_a and λ_b into 64 levels from $10^{-3}\pi f_s$ to $2\pi f_s$ linearly in $\exp(-\lambda_a)$ and $\exp(-\lambda_b)$, ω_s into 64 levels linearly from $\frac{1}{64}\pi f_s$ to πf_s, and r_s into 64 levels linearly from $\frac{1}{128}$ to $\frac{1}{2}$. Not that the set of parameters can be represented by 39 bits with simple fixed-length coding.

In the first iteration of the clustering algorithm, instead of performing a full search to solve (14), we conduct the following initialization procedure composed of simple image processing tasks and single-variable discrete optimization. First, we compute the non-parametric optimal solution with (3), and denote it by $r_c^*[\mathbf{d}]$. The parameter θ is initialized as the direction with the largest average squared gradient in $r_c^*[\mathbf{d}]$. Given θ, the autocorrelation values at a sequence of points along θ, denoted by $\{d_a\}$, and orthogonal to θ, denoted by $\{d_b\}$, are interpolated from $r_c^*[\mathbf{d}]$, and denoted by $\{r_c^*[d_a]\}$ and $\{r_c^*[d_b]\}$ respectively. Second, ω_s is initialized by computing the 1-D zero-padded FFT of $\{r_c^*[d_a]\}$ and finding the location of largest FFT coefficients in magnitude. Third, to initialize r_s, we denote the zero-crossing point closest to $\tau = 0$ in (6) by τ_z. Note that (6) represents a mean-removed periodic sequence of triangles, and it can be shown that $r_s^2 - r_s + \frac{\tau_z}{T_s} = 0$. Therefore, given ω_s and the zero-crossing point τ_z estimated from $\{r_c^*[d_a]\}$, we initialize r_s by minimizing $r_s^2 - r_s + \frac{\tau_z \omega_s}{2\pi}$. Next, with ω_s and r_s available, we initialize β and λ_a jointly by minimizing the sum of squared difference between $\{r_c^*[d_a]\}$ and the corresponding values derived from $r_{tt}[\mathbf{d}]$, with the aperture function neglected, computed by

$$r_{tt}[d_a|\beta, \lambda_a] = \beta \exp(-\lambda_a \frac{d_a}{f_s}) + (1-\beta) \sum_{k=1}^{K} \gamma(k, r_s) cos(k\omega_s \frac{d_a}{f_s}). \tag{15}$$

Note that in this least square formulation, the optimal β is a function of the selected λ_a, and hence can be computed analytically. Therefore, this step performs a single-parameter search that initializes two parameters in the model. Finally, given β, λ_b is initialized similarly by minimizing the sum of squared difference between $\{r_c^*[d_b]\}$ and the sequence computed by

$$r_{tt}[d_b|\lambda_b] = \beta \exp(-\lambda_b \frac{d_b}{f_s}) + (1-\beta). \tag{16}$$

After the initialization procedure, a joint search on $\{\theta, \lambda_a, \lambda_b, \omega_s, r_s\}$ is performed in the vicinity of the initial values to further refine the estimation. Again, the optimal β can be analytically computed as a function of the selection of the other parameters. In the subsequent iterations of the clustering algorithm, the initial values can be the set of parameters estimated in the previous iteration or that obtained with the initialization procedure, whichever leads to a smaller distortion. In each iteration, for the given $m(b)$ we might only find a suboptimal solution of (14) due to the simplified search process. Nonetheless, as long as the overall distortion decreases, the iterative clustering algorithm converges to a local minimum.

3.4 Filter Performance Analysis with Parametric Texture Source

In Section 2.2 and 2.3, we described the modeling of a block of neighborhoods $\mathcal{N}_b$ with its mean $\bar{\mu}_b$, variance $v_b[\mathbf{0}]$, and a discrete-space texture source $t[\mathbf{l}]$. For any linear filter, defined by the impulse response $h_f[\mathbf{l}]$ or the transfer function $H_f(e^{j\mathbf{\Omega}})$, applied to the mean-removed $\mathcal{N}_b$, the variance of the zero-mean filter output in block $\mathcal{B}_b$, denoted by $E_{f,b}$, can be estimated through

$$E_{f,b} = v_b[\mathbf{0}] \cdot (r_{tt} * h_f * \overline{h_f})[\mathbf{0}] = v_b[\mathbf{0}] \int_{\mathbf{\Omega}} \Phi_{tt}(e^{j\mathbf{\Omega}}) |H(e^{j\mathbf{\Omega}})|^2 d\mathbf{\Omega}, \tag{17}$$

Barbara *Boats* *Pentagon* *Spoke*

Figure 1. Test images.

where $\overline{h_f}[\mathbf{d}] = h_f[-\mathbf{d}]$. With the parametric formulation, the support size of the filter to be analyzed is no longer limited by the $S_n \times S_n$ neighborhood size. The autocorrelation values with $|d_x|, |d_y| \geq S_n$ can be extrapolated from the parametric model, despite that the corresponding sample covariance values are not available.

In addition, from (8), (11b), and (13), $\Phi_{tt}(e^{j\mathbf{\Omega}})$ is clearly always positive. Hence, the estimated filter output variance computed in (17) is always nonnegative. This property does not exist with a non-parametric autocorrelation function, e.g., the one computed in (3).

4. EXPERIMENTAL RESULTS

In this section, we report the experimental results for the 512×512 test images *Barbara*, *Boats*, *Pentagon*, and *Spoke*, shown in Fig. 1. We consider two applications in the experiments: using the proposed parametric model to design linear predictors, and using the model to analyze the performance of wavelet-based image coding methods.

To construct the model, we choose the following global parameters: neighborhood size $S_n = 13$, block size $S_B = 16$, number of texture clusters $L_C = 32$. In practice, we approximate the term $\delta(\omega_a \pm k\omega_s)$ in (8) with $\frac{\pi}{\lambda_0^2}(1 + (\frac{\omega_a \pm k\omega_s}{\lambda_0})^2 + (\frac{\omega_b}{\lambda_0})^2)^{-\frac{3}{2}}$, $\lambda_0 = 10^{-3}\pi f_s$, to avoid handling of delta impulses in the PSD. Correspondingly, the second term (regular texture) in (7) is multiplied by $\exp(-\lambda_0(\tau_a^2 + \tau_b^2)^{\frac{1}{2}})$.

For each test image, 16 resulting discrete-space texture sources, randomly selected from the total of 32, are shown in Fig. 2. The non-parametric autocorrelation functions are obtained by evaluating (3) rather than (14) in the last iteration of the clustering algorithm. In the autocorrelation functions in Fig. 2, the black level denotes -1 and white denotes 1. The parametric PSD is plotted in logarithmic scale. In most cases, the proposed six-parameter model well approximates the 313 unique values in the non-parametric function. Using 32 clusters is more than sufficient for most test images, except for *Barbara*. In *Barbara*, a small variation in the orientation or the period of the stripes requires a different texture source to represent. Additionally, the checkerboard patterns that appear frequently in the image cannot be modeled by the proposed parameterization, due to the two orthogonal edges in the same image neighborhood. Hence, the stripes that are not represented by a dedicated texture source and other intricate features are mixed into a cluster with a complex autocorrelation structure, as shown in, for example, the bottom-left cluster in Fig. 2(a_1), leading to a large distortion.

4.1 Linear Predictor Design

We assume a scenario where the model parameters are estimated at the encoder using the original test image, and then the image is decimated by a factor of two in both dimensions and transmitted to the decoder. The decoder attempts to recover the missing samples through linear prediction. The information necessary for designing the predictors is also transmitted to the decoder as overhead.

For each texture source, we design three predictors of a support size up to the size of the 13×13 neighborhoods, responsible for predicting the odd-even, even-odd, and odd-odd missing samples from the available even-even

Table 1. Linear Predictor Design with Texture Source Clusters

	fixed	para.	non-para.	non-para.
overhead per texture	0	39 bits	480 bits	48 bits
overhead per pixel	0	0.005 bits	0.06 bits	0.006 bits
Barbara	22.29 dB	+3.63 dB	+5.87 dB	+0.18 dB
Boats	28.80 dB	+0.60 dB	+1.11 dB	+0.37 dB
Pentagon	28.01 dB	+1.12 dB	+1.69 dB	+0.52 dB
Spoke	20.61 dB	+6.70 dB	+6.65 dB	+0.12 dB

samples respectively. For comparison, a fixed 1-D predictor with impulse response $[\frac{1}{32}\ 0\ \frac{-5}{32}\ 0\ \frac{5}{8}\ 0\ \frac{5}{8}\ 0\ \frac{-5}{32}\ 0\ \frac{1}{32}]$ is applied in a separable manner, and the PSNR of the reconstructed missing pixels serves as a reference.

Given the proposed parametric autocorrelation, the three optimum linear predictors can all be determined from the 6 parameters, encoded by 39 bits per texture (Section 3.3), and about 0.005 bpp with 32 texture sources. If the predictor is designed using the non-parametric autocorrelation, the 313 unique values also need to be transmitted. An alternative is to design the predictors at the encoder and transmit the predictor as side information. In this case, there are 60 unique coefficients in the 3 predictors, each quantized to 8 bits, resulting in 480 bits per texture. Note that the performance loss due to this quantization is typically smaller than 0.05 dB. We also consider a simpler case where only a 1-D vertical predictor and a 1-D horizontal predictor are designed at the encoder using the non-parametric autocorrelation function, and the odd-odd samples are predicted in a separable manner. Each of the 1-D filters has 3 unique coefficients, same as the fixed predictor, therefore the overhead per texture is 48 bits, comparable to the parametric case.

The texture sources, hence the predictors, are designed for the mean-removed image neighborhoods. Hence, the mean of each block of neighborhoods should also be available at the decoder. We assume that this mean can be reliably estimated at the decoder from the available samples, and the overhead is neglected. In addition, the membership function $m(b)$ should also be signaled, requiring up to 5 bits per block (about 0.02 bpp).

From Table 1, the gain by designing locally-adaptive predictors using the parametric model ranges from 0.6 and 6.7 dB, highly depending on the image content. The loss in performance due to the proposed parameterization can be significant, for instance more than 2 dB in *Barbara* due to the intricate image features as explained earlier. Nevertheless, when compared with the case where the predictor designed from the non-parametric function is constrained in the support size to achieve a comparable amount of overhead (39 vs. 48 bits), the parametric function performs considerably better, demonstrating the potential in image communication applications.

4.2 Wavelet Image Coding Performance Analysis

As discussed in Section 3.4, for any linear transform applied on the mean-removed image neighborhoods, the proposed model can be used to predict the variance of the transform coefficients. In the following experiments, we analyze the rate-distortion performance of image coding with the conventional discrete wavelet transform (DWT) and the direction-adaptive discrete wavelet transform (DA-DWT).[6] The DA-DWT locally adapts the wavelet filtering directions to image features so that the energy is concentrated in the low-pass coefficients to improve the coding performance.

For each texture source, we first compute the variance of each subband. The subband variance in each block is then the corresponding texture subband variance multiplied by the block variance. Based on the observation in Ref. 7, we assume that locally the subband coefficients are Gaussian distributed, and model the entropy coding performance by the rate-distortion function of Gaussian memoryless coding, i.e., $R(\theta_n) = \max\{0, \frac{1}{2}\log_2 \frac{v_{b,s}}{\theta_n}\}$ and $D(\theta_n) = \min\{\theta_n, v_{b,s}\}$, where $v_{b,s}$ denotes the variance of subband s in block $\mathcal{B}_b$, and θ_n controls the rate-distortion trade-off. The same θ_n is used across all subbands and all blocks to achieve the optimal rate allocation. Finally, the image-wise rate and distortion are computed by taking the average across subbands, weighted by the number of coefficients in the subbands, and across all blocks.

In Fig. 3, we compare the rate-distortion curves derived from the model and that obtained from actual coders for 4 levels of the DWT and the DA-DWT with the 5/3 wavelet filters, and the DWT with the ideal rectangular low-pass and high-pass filters. Note that the subband variance of the hypothetical ideal DWT with IIR filters is

computed in the PSD domain via numerical integration. For most test images, the performance gain by locally adapting the filtering directions, i.e., the gap between the DWT and the DA-DWT, is successfully captured by the model. For *Barbara*, the gap predicted is smaller than the actual one, again, due to the fact that parts of the directional features (stripes) are not represented by a dedicated texture source. Fig. 3 also suggests that by adapting the filtering directions, the DA-DWT with simple filters can approach or exceed the performance of the conventional DWT with complex filters.

5. CONCLUSION

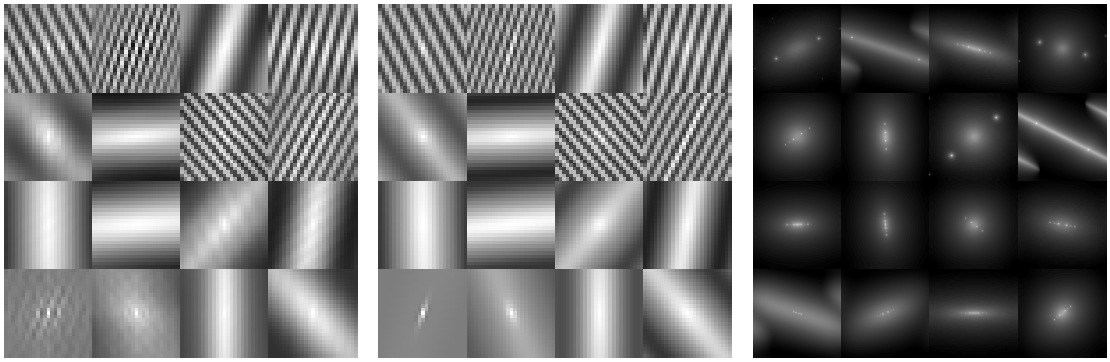
The proposed image model overcomes the weaknesses of the prior statistical models by incorporating a localized representation of the image statistics with an anisotropic parameterization. The localized representation captures local image properties, thus allowing handling of spatially-adaptive algorithms. The novel anisotropic parameterization further enables accurate description of directional image features. Potential applications include the design of adaptive transforms for image coding and the design of adaptive filters for image restoration.

ACKNOWLEDGMENTS

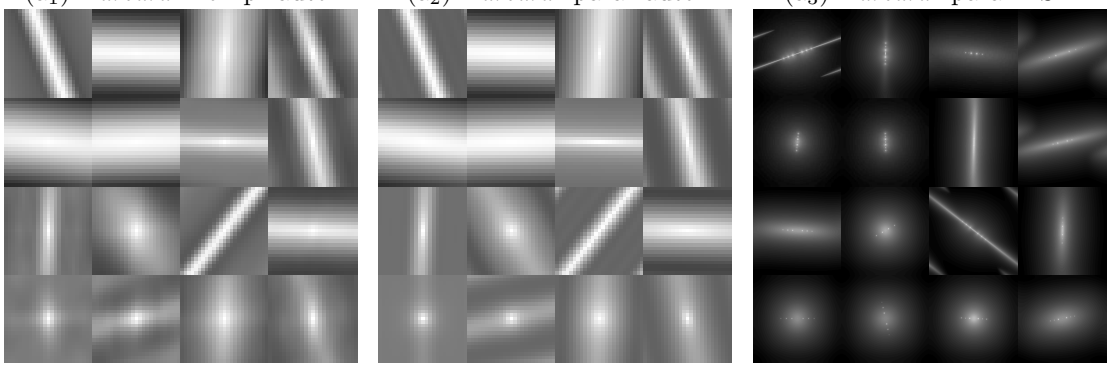
This work was supported by the Max Planck Center for Visual Computing and Communication.

REFERENCES

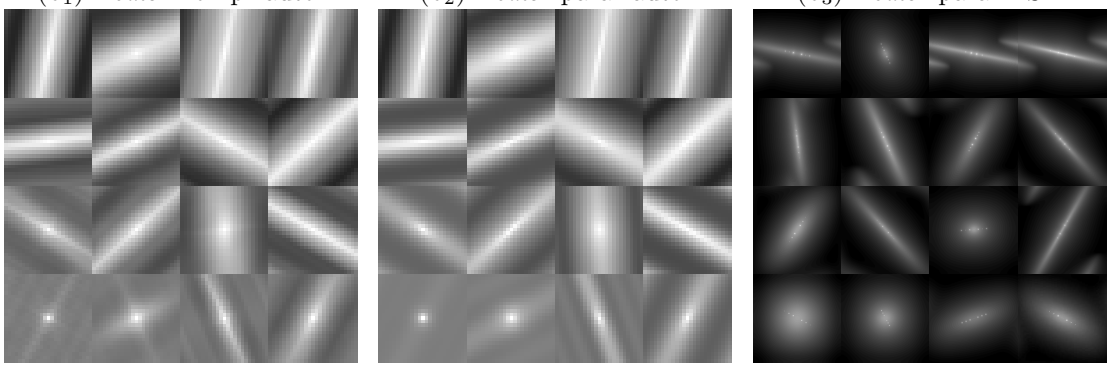
1. A. Habibi and P. A. Wintz, "Image coding by linear transformation and block quantization," *IEEE Transactions on Communication Technology* **19**, pp. 50–62, Feb. 1971.
2. D. J. Sakrison and V. R. Algazi, "Comparison of line-by-line and two-dimensional encodings of random images," *IEEE Transactions on Information Theory* **17**(4), pp. 386–398, 1971.
3. J. B. O'Neal, Jr. and T. R. Natarajan, "Coding isotropic images," *IEEE Transactions on Information Theory* **23**, pp. 697–707, Nov. 1977.
4. R. S. Prendergast and T. Nguyen, "A novel parametric power spectral density model for images," in *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers 2005*, pp. 1671–1675, (Pacific Grove, CA, USA), Nov. 2005.
5. R. S. Prendergast and T. Nguyen, "A non-isotropic parametric model for image spectra," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing 2006*, **2**, pp. 761–764, (Toulouse, France), May 2006.
6. C. L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform for image compression," *IEEE Transactions on Image Processing* **16**, pp. 1289–1302, May 2007.
7. J. M. Lervik and T. A. Ramstad, "Optimality of multiple entropy coder systems for nonstationarysources modelled by a mixture distribution," **4**, pp. 1874–1877, May 1996.



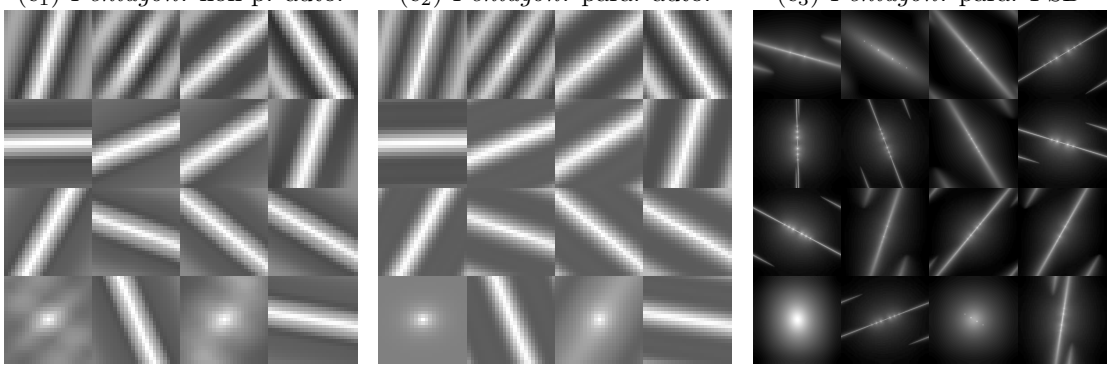
(a_1) *Barbara*: non-p. auto. (a_2) *Barbara*: para. auto. (a_3) *Barbara*: para. PSD



(b_1) *Boats*: non-p. auto. (b_2) *Boats*: para. auto. (b_3) *Boats*: para. PSD



(c_1) *Pentagon*: non-p. auto. (c_2) *Pentagon*: para. auto. (c_3) *Pentagon*: para. PSD



(d_1) *Spoke*: non-p. auto. (d_2) *Spoke*: para. auto. (d_3) *Spoke*: para. PSD

Figure 2. Non-parametric autocorrelation, parametric autocorrelation, and parametric PSD of the discrete-space texture sources.

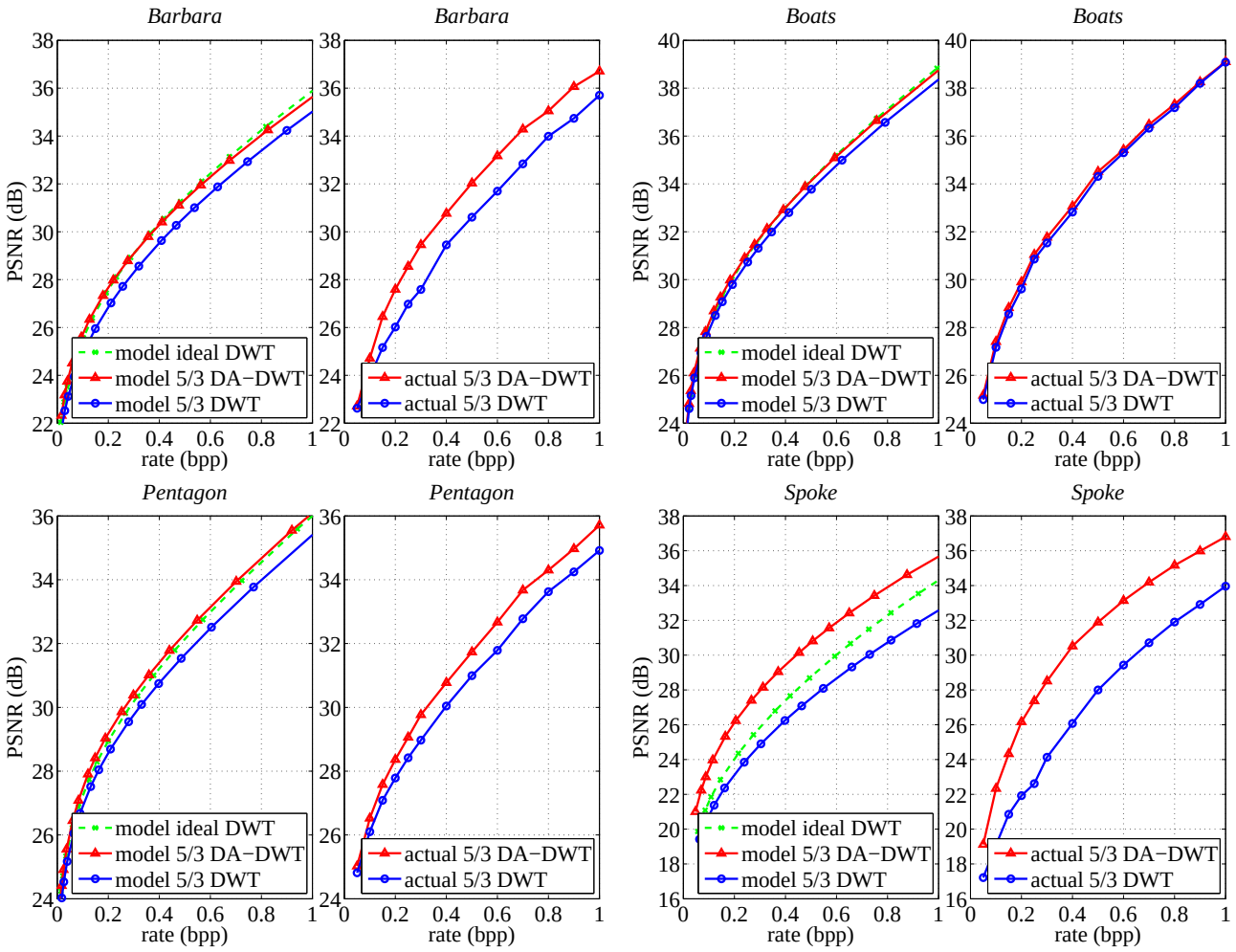


Figure 3. Rate-distortion performance of the DWT and the DA-DWT derived from the proposed model and from actual coders.