ELSEVIER

Signal Processing: *Image Communication* 17 (2002) 85–104

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.com/locate/image

An overview of the visual optimization tools in JPEG 2000

Wenjun Zeng[1]*, Scott Daly, Shawmin Lei

Sharp Laboratories of America, 5750 NW Pacific Rim Blvd., Camas, WA 98607, USA

Abstract

The human visual system plays a key role in the final perceived quality of the compressed images. It is therefore desirable to allow system designers and users to take advantage of the current knowledge of visual perception and models in a compression system. In this paper, we review the various tools in JPEG 2000 that allow its users to exploit many properties of the human visual system such as spatial frequency sensitivity, color sensitivity, and the visual masking effects. We show that the visual tool sets in JPEG 2000 are much richer than what is achievable in JPEG, where only spatially invariant frequency weighting can be exploited. As a result, the visually optimized JPEG 2000 images can usually have much better visual quality than the visually optimized JPEG images at the same bit rates. Some visual comparisons between different visual optimization tools, as well as some visual comparisons between JPEG 2000 and JPEG, will be shown.

Keywords: Visual optimization; Wavelet compression; JPEG 2000 standard; Perceptual model; Visual masking; Frequency sensitivity

1. Introduction

In the Sydney JPEG meeting (where initial JPEG 2000 proposals were made), the contribution from Sharp Labs of America [16] demonstrated the impressive visual improvement that frequency weighting can offer, particularly at display/print resolutions greater than 200 dpi (127 μm/pixel). Since then, the JPEG committee working on JPEG 2000 has been aggressively pursuing the goal of removing perceptual irrelevancy, in addition to statistical redundancy, of the image data. Fig. 1 shows a typical visual quality improvement that frequency and color weighting can achieve for JPEG 2000. It can be seen that the reconstructed image with proper frequency and color weighting (left) preserves the fine texture much better than the one generated without using visual optimization tools (right). What is more interesting is that the right-side image has 1.3 dB better peak signal-to-noise ratio (PSNR) than the other one, although its visual quality is much worse. This is but one example that shows that mean square error may not be a good measure of image visual quality.

There has been substantial work in vision science that tries to understand and model the human visual system's behavior. It has been recognized that the visual sensitivity varies as a function of several key image dimensions, such as light level [27], spatial frequency [13,30], color [17], local image contrast [27,23], eccentricity [26] and temporal frequency [22]. The most common method of visually optimizing compression

*Corresponding author.

E-mail addresses: wzeng@pv.com (W. Zeng), daly@sharplabs.com (S. Daly), shawmin@sharplabs.com (S. Lei).

[1] Now with PacketVideo Corporation, 4820 Eastgate Mall, San Diego, CA. This work was performed while he was with Sharp Labs.

0923-5965/02/$ - see front matter
PII: S0923-5965(01)00029-7

Fig. 1. Portions of the compressed "woman" images (crop size: 310×400, original image size: 2048×2560) using JPEG 2000 at 0.75 bpp. Left: baseline JPEG 2000 with frequency and color weighting using Table 2, PSNR = 34.6 dB; Right: baseline JPEG 2000 without frequency and color weighting, PSNR = 35.9 dB.

algorithms is to transform the amplitudes of the image to a perceptually uniform domain. Since the visual system's gray-scale behavior is approximately characterized by a cube-root front-end amplitude nonlinearity, the theory is to convert the image into that domain, and then quantize. Then to display, the inverse nonlinearity is used to convert to photon flux. The cascade of the display's inverse nonlinearity with the visual system's nonlinearity results in the quantization levels being perceptually uniform. This technique forms part of nearly all video [21], with the exception that the power function of 3 is replaced by values around 2.2–2.4; this domain is generally referred to as *gamma-corrected*. Most compression algorithms do this by default, as a consequence of compressing images represented in the format of video standards. The advantage of this approach is so substantial that it is essentially *de facto* in any compression algorithm. The key remaining dimensions of a still image that can be visually optimized are along spatial frequencies, color, and the visual masking by the image content.

Given the knowledge of the human visual system's behavior, the next step is to figure out how to exploit these properties effectively in a practical compression system. Some compression systems may allow a more thorough exploitation of the properties of the HVS than others. Table 1 shows the visual tools supported by JPEG 2000, as compared to JPEG. It is well known that JPEG has the Q tables that allow one to apply frequency and color weighting to each 8×8 block. JPEG 2000, however, supports many more new features such as visual progressive weighting, neighborhood masking, self-masking and point-wise extended masking. We will discuss each one in some detail in the following sections, and will explain why these features are feasible in JPEG 2000, but not in JPEG. The other three features—local light adaptation, eccentricity and temporal frequency, are also supported by the structure of JPEG 2000, although they are not currently available in the JPEG 2000 Verification Model (VM) software [14].

JPEG 2000 [11] is a wavelet-based bit-plane coder where coefficients in each wavelet subband

Table 1
Visual optimization tools supported by JPEG 2000 and JPEG

HVS properties	JPEG 2000	JPEG
Frequency weighting	+	+
Color weighting	+	+
Visual progressive weighting	+	–
Neighborhood masking	+	–
Point-wise self-masking	+	–
Point-wise extended masking	+	–
Local light adaptation	+	–
Eccentricity	+	–
Temporal frequency	+	–

+: supported; –: not supported.

are divided into blocks of the same size (called code-block) and each code-block is embedded coded independently (see Fig. 2). It introduces the concept of abstract quality layers that allows a post-compression optimization process where sub-bitstreams from each code-block are assembled in certain (e.g., a rate-distortion (R–D) optimized) order to form the final bitstream. This quality layer formation process is a key component in JPEG 2000. Basically, after an embedded bit-stream is generated for each code-block (shown as vertical bars in Fig. 2 where bits at the top are more important than those at the bottom), it is up to the encoder to determine how to assemble a sub-bitstream from each code-block to form the quality layers. In other words, it is up to the encoder to determine how to draw the quality layer lines or choose the truncation points as shown in Fig. 2. This flexibility basically enables a code-block-wise adaptive bit allocation. As we will see later, it is this flexibility that makes many visual tools in JPEG 2000 feasible.

In this paper, we review the tools in JPEG 2000 that allow its users to take advantages of the various properties of the HVS such as spatial frequency sensitivity, color sensitivity, and the visual masking effects. We will show that the visual tool sets in JPEG 2000 are much richer than what is achievable in JPEG, where only spatially invariant frequency weighting can be exploited. As a result, the visually optimized JPEG 2000 images can usually have much better visual quality than the visually optimized JPEG images at the same bit rates. This paper is organized as follows. Section 2 presents the visual tools that allow the exploitation of the spatial frequency sensitivity and the color sensitivity, including fixed frequency weighting and visual progressive weighting. Three different ways of exploiting the visual masking effects are discussed in Section 3. Finally, Section 4 shows some visual comparisons with some discussions.

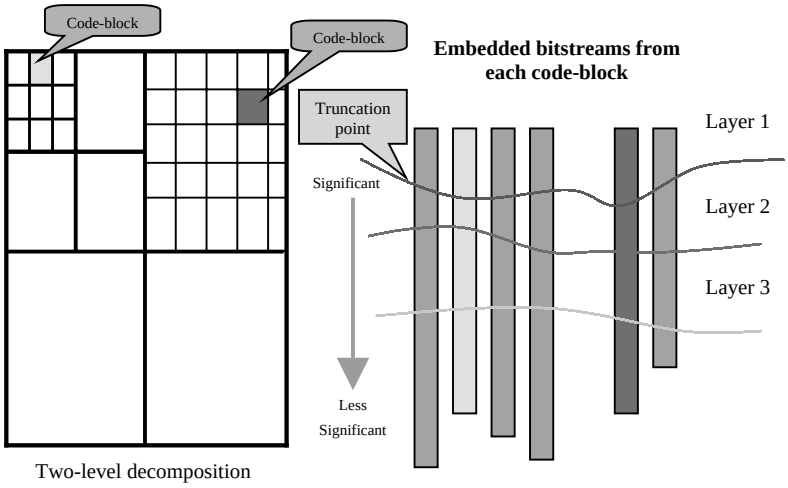

Fig. 2. JPEG 2000 codestream quality layer formation.

2. Visual frequency weighting

One common visual optimization strategy for compression is to make use of the contrast sensitivity function (CSF) that characterizes the varying sensitivity of the visual system to 2D spatial frequencies [13,30], as shown in Fig. 3. In general, human eyes are less sensitive to high-frequency errors than to low-frequency errors. The CSF can be used to determine the relative accuracies needed across differing spatial frequencies, where the term "weight" is used to describe the desired proportional accuracy. To use the CSF, which is usually described in visual frequencies of cycles/degree (cpd), it must be mapped to the compression domain of digital frequencies such as cycle/pixel. The design of the CSF weights is an encoder issue and depends on the specific viewing condition under which the decoded image is to be viewed [13]. Recent studies [33,31,5] suggest that it may also depend on the distortion/bit rate of the compressed image. In JPEG 2000, three default weighting tables have been recommended for three common viewing distances [11]. Weighting tables for color images have also been recommended [5,18]. A sample weighting table is shown in Table 2. It can be seen that the weights for the low-frequency subbands are larger than the high-frequency subbands. There is also a bias toward the luminance component against the Cb, Cr components. Note that in JPEG, the differing importance of the channels is generally handled by subsampling the chrominance components by a factor of 2 in the vertical and horizontal directions. Additionally, the downsampled chrominance components use more aggressive quantization tables than the luminance component. JPEG 2000 has the advantage that a "soft downsampling" (using the weighting table without explicitly downsampling the chrominance components), as opposed to JPEG's "hard downsampling", can be implemented. A proper frequency and color weighting can usually result in significant detail and texture preservation with no introduction of color distortions (see, e.g., Fig. 1). In general, frequency weighting is more effective for large viewing distance or high dpi printing. In fact, it can also be used to reduce the flicking artifacts of Motion JPEG 2000, as demonstrated in [9].

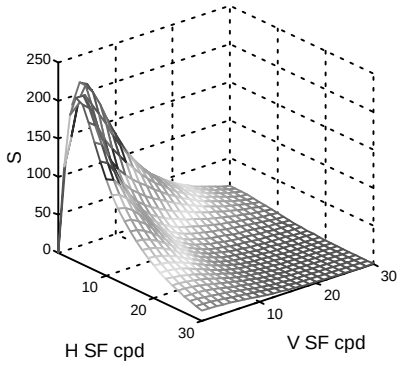

Fig. 3. A general unsampled 2D CSF.

2.1. Fixed frequency weighting

In general, the CSF curve is a continuous function of the spatial frequency. However, for a discrete wavelet transform, it is common that only

Table 2
A sample weighting table for a viewing distance of about 1700 pixels for a five-level decomposition. A larger "level" value corresponds to a lower resolution. A larger weight indicates higher quantization accuracy

Level	Y (LH HL HH)	Cb (LH HL HH)	Cr (LH HL HH)
1	0.275783 0.275783 0.090078	0.089950 0.089950 0.027441	0.166647 0.166647 0.070185
2	0.837755 0.837755 0.701837	0.267216 0.267216 0.141965	0.375176 0.375176 0.236030
3	0.999994 0.999994 0.999988	0.488887 0.488887 0.348719	0.587213 0.587213 0.457826
4	1.000000 1.000000 1.000000	0.679829 0.679829 0.567414	0.749805 0.749805 0.655884
5	1.000000 1.000000 1.000000	0.812612 0.812612 0.737656	0.856065 0.856065 0.796593

one CSF weight is chosen for each subband to facilitate the implementation. This way of applying visual frequency weighting is referred to as fixed frequency weighting. For example, based on the specific viewing condition, the weight corresponding to the sensitivity of the mid-frequency of a subband could be chosen for that particular subband [13]. The set of CSF weights can be incorporated in one of two ways in JPEG 2000, as described in the following. In both cases, the CSF weights do not need to be explicitly transmitted to the decoder.

2.1.1. Modify the quantization step size

At the encoder, the quantization step size q_i of the transform coefficients of subband i is adjusted to be inversely proportional to the CSF weight w_i. The CSF-normalized quantization indices are then treated uniformly in the R–D optimization process. The CSF weighting information is reflected in the quantization step sizes that are explicitly transmitted for each subband. This approach needs to explicitly specify the quantizers so it may not be suitable for embedded coding that generates a bitstream from lossy all the way to lossless. (If lossless is not required, this approach works fine.) Furthermore, this approach cannot be extended to perform visual progressive weighting where weights need to be changed at different bit rates during embedded coding. This implementation can be invoked in the JPEG 2000 VM software [14] by supplying the same file of visual weights to both the "-Fsteps" and "-Fweights" arguments. This approach may be easier to understand and to implement than the next approach.

2.1.2. Modify the embedded coding order

In this implementation, the quantization step sizes are not modified, but the distortion weights fed into the R–D optimization are altered instead, linearly proportional to the CSF weight for each subband. This effectively controls the relative significance of including different numbers of coding passes from the embedded bitstream of each code-block to form the bitstream quality layers (see Fig. 2). This is an encoder issue only. The decoder does not need to be aware of it. This implementation can be invoked through the "-Fweights" option in the VM software. This implementation is recommended since it produces similar results as the first implementation and is compatible with lossless compression as well as visual progressive weighting. The extension of this approach to the visual progressive weighting will be described in Section 2.2.

It is also possible to do cell-adaptive CSF weighting [17,19], which allows a better adaptation of the CSF weight to the signal spectrum in a sub-region (e.g., code-block) of a subband. Basically, a data-adaptive weighting factor can be determined for a sub-region by filtering the wavelet coefficients in that sub-region with the CSF filter of the corresponding sub-part of the CSF. The weighting can be done at the encoder only. Theoretically, by considering the actual frequency content of the sub-region, this approach would generate more accurate weighting factors than just choosing the CSF weight corresponding to the middle frequency of the subband. However, it has been shown [19] that, under the framework of JPEG 2000, the advantage of this strategy over the above-mentioned fixed frequency weighting is rather small for the compression of natural images, but it might be of bigger impact for images of non-natural scenery.

2.2. Visual progressive weighting

JPEG 2000 allows the implementation of visual progressive weighting, where different sets of CSF weights can be applied at different stages of the embedding to form different quality layers [11,15]. In particular, to implement the visual progressive weighting, the JPEG 2000 VM (using the "-Cvpw" argument) changes, on the fly, the order in which code-block sub-bitplanes should appear in the overall embedded bitstream based on several sets of frequency weights targeted for different bit-rate ranges.

The initial motivation for visual progressive weighting is that "as the embedded bitstream may be truncated later, the viewing conditions for different stages of embedding may be very different" [15]. Visual progressive weighting thus allows the use of different sets of CSF weights that correspond to different viewing distances at

different stages of the embedding. However, it remains unclear what viewing distance should be considered for a specific bit-rate range, or if that is entirely application dependent.

Recent studies [33,31] have shown that even with a fixed viewing distance, a more aggressive weighting usually results in a better visual quality than the "matched" weighting targeted for that viewing distance, when the bit rate is low. This is because, traditionally, CSF is usually derived through experiments based on the just-noticeable-difference (JND) criterion and such CSF may not be fully applicable to the low-bit-rate cases where coding errors are usually quite visible. A distortion-adaptive visual weighting strategy, based on a visual signal estimation approach (in addition to the traditional visual signal detection approach), has been proposed [33] to address visual weighting at low bit rates for both fixed frequency weighting and visual progressive weighting.

In traditional psychophysical experiments, the amplitude of each frequency basis function is increased until it reaches a just noticeable frequency threshold (JND) where people can detect the existence of the signal under a specific viewing condition [13,30]. These frequency JNDs are then used to generate the CSF curve to represent the relative visual significance of each frequency component. Typically, $w_i = k/T_i$, where w_i and T_i, respectively, are the CSF weight and the frequency detection threshold for the ith frequency basis function, and k is a constant normalization factor. Most previous works on perceptual coding usually implicitly assume that the relative weights will remain unchanged for different distortions/bit rates.

Experiments have shown [33,31] that the traditional CSF weights do not seem to work well in low bit-rate scenarios. This is not necessarily surprising because the traditional CSF curve is derived based on just noticeable detection thresholds (corresponding to a visually near-lossless condition). At lower bit rates, the distortion is quite visible and the visual effect has not been conclusively understood in the literature. A distortion-adaptive CSF weighting strategy was proposed in [33] to address the visual frequency sensitivity under the condition of large distortions.

It was argued that, for low bit rates, the effect of visual distortion is an estimation problem rather than just a detection problem [33]. In other words, it becomes important to estimate the amount of visual distortion of each frequency component perceived by the human eyes when measuring the frequency sensitivity. For wavelet-based systems, different basis functions usually have different spatial supports, and different non-flat envelops. In general, low-frequency basis functions have larger spatial supports than high-frequency basis functions. In the visually near lossless scenario, the side lobes of the basis function remain largely undetected. So the spatial support of the basis function is not of significant impact on the perception. However, at low bit rates, the distortion signal strength is increased and the side lobes of the basis function become visible. The spatial support of the basis function starts affecting the perception of the distortion.

The following measure has been proposed in [33] to compensate for the "side lobe effect". Let $f_i(x)$ denote the basis function with *unit* peak-to-mean amplitude for the ith subband. Assume the distortion to each basis function is $df_i(x)$ where d is the normalized peak-to-mean amplitude (in the unit of T_i). The normalization is with respect to the frequency detection threshold T_i of each basis function. It accounts for the visual sensitivity to spatial frequency. We define the "effective" basis distortion function $g_i(x; d)$ as

$$\begin{aligned} g_i(x;d) &= |df_i(x)| \quad \text{if } [df_i(x)] > 1, \\ &= 0 \qquad\quad \text{otherwise.} \end{aligned} \tag{1}$$

The coring to zero is a rough model of the threshold aspect of the CSF. The compensation factor λ_i that accounts for the "side lobe effect" can be defined as

$$\lambda_i(d) = \left(\int_{-\infty}^{+\infty} |g_i(x;d)|^{\mathrm{p}}\, \mathrm{dx} \right)^{1/\mathrm{p}} \quad \text{if } d > 1, \tag{2}$$

where $0 \leqslant p < \infty$. If $d \leqslant 1$, $\lambda_i(d)$ will all be set to 1. Therefore, if d is less than 1 (or equivalently, the peak-to-mean amplitude of the distortion to each basis function is less than the frequency detection threshold T_i), there is no compensation for the "side lobe effect". If the actual peak-to-mean amplitude of the basis distortion function is

Table 3
Two sets of weights (for luminance only) for visual progressive weighting under 1000-pixel viewing distance condition. Left: traditional weights; right: effective weights with $p=2$ and $d=\infty$ (where the distortion is assumed to be very large)

Level	1000.tbl (LL, LH, HL, HH)	1000_∞.tbl (LL, LH, HL, HH)
1	1.0000 0.5608 0.5608 0.2841	1.0000 0.1833 0.1833 0.0884
2	1.0000 1.0000 1.0000 0.7271	1.0000 0.5251 0.5251 0.3092
3	1.0000 1.0000 1.0000 1.0000	1.0000 1.0000 1.0000 0.7876
4	1.0000 1.0000 1.0000 1.0000	1.0000 1.0000 1.0000 1.0000
5	1.0000 1.0000 1.0000 1.0000	1.0000 1.0000 1.0000 1.0000

greater than T_i, then the portion of the basis distortion function that has an amplitude exceeding the threshold T_i will contribute to the visual distortion. A special case is that $p=2$, and $d\to\infty$. In this case, the compensation factor λ_i is in fact the square root of the energy of the basis function with unit peak-to-mean amplitude, subject to a constant factor that is common to all basis functions. In general, low-frequency basis function with unit peak-to-mean amplitude will have larger energy than high-frequency basis function with unit peak-to-mean amplitude. This suggests that low-frequency basis function is more sensitive to distortion than high-frequency basis function, thus demanding more protection than what the traditional CSF curve suggests that only accounts for frequency sensitivity, but not the "side-lobe effect". The final effective CSF weight for a distortion to the ith basis function with a T_i-normalized peak-to-mean amplitude of d should be

$$w_i' = w_i\lambda_i, \tag{3}$$

subject to a constant normalization factor.

The model described above tries to characterize the different amounts of distortion perceived by the human eyes when the distortion signal for each frequency has an amplitude that is d times of its frequency detection threshold T_i. Note that previous works on perceptual coding usually assume that these visual distortions are the same. The proposed model therefore provides a fine adjustment of the frequency weights based on the instant T_i-normalized peak-to-mean amplitude of the distortion signal during the embedded coding process.

The traditional CSF curve usually has a dip at very low frequencies and reaches the peak value at some mid-frequency f_{peak}. In practice, the weights are usually set to 1 for all frequencies no larger than f_{peak}, which is done to consider the more practical usage of a range of viewing distances, where only the closest is known or designed for. For a reasonable compensation of the "side-lobe effect", we assume that the peak will be assumed at the next lower frequency level. For example, in Table 3, the original CSF weights 1000.tbl have a peak at level 3. The effective weights 1000_∞.tbl will then have a peak at level 4.

This technique, referred to as the distortion-adaptive visual progressive weighting (DAVPW) strategy, was implemented in [31] based on JPEG 2000 VM7.0 [14]. In particular, for each quality layer, the instant T_i-normalized average distortion of the whole image after encoding the previous quality layer will be used to calculate the compensation factors and update the effective weights.

We compared the performance of the distortion-adaptive visual progressive weighting to that of the fixed weighting using the 1000.tbl table for the on-screen 1000-pixel viewing distance case. Fig. 4 shows that DAVPW provides noticeable visual improvement over fixed weighting at the low bit rates. In general, the lower the bit rate, the larger the visual improvement. Fig. 5 shows that while the aggressive 1000_∞.tbl table for fixed weighting performs well at 0.25 bpp, it results in high-frequency artifacts at 0.75 bpp. On the other hand, DAVPW automatically adjusts the effective weights, thus provides good visual quality across different bit rates/distortions.

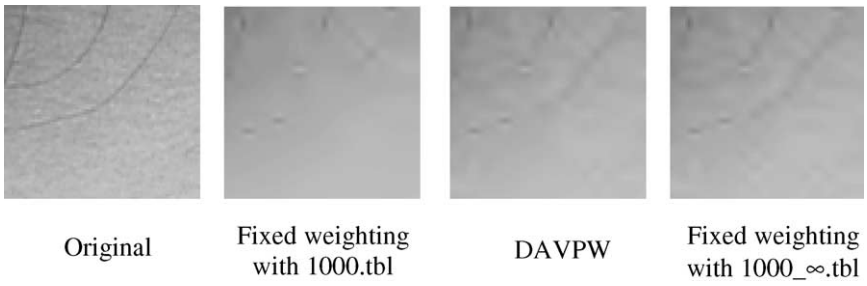

Fig. 4. Foreheads of the "woman" image coded at 0.25 bpp using different weighting strategies.

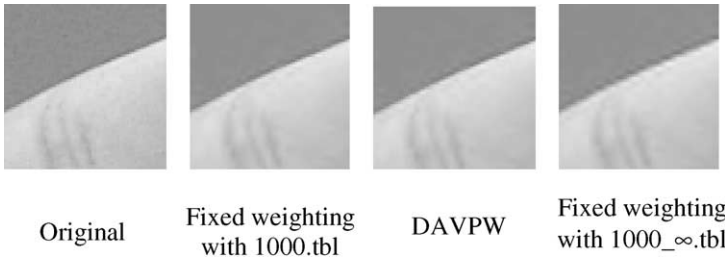

Fig. 5. Fingers of the "woman" image coded at 0.75 bpp using different weighting strategies.

3. Visual masking

Frequency weighting is usually very effective for applications with a high-resolution display or large viewing distance. In both cases, the viewing distance expressed in units of pixels will be greater than around 1500. The advantage of this technique, however, becomes less noticeable for lower resolution display and closer viewing distance, since the CSF curve mapped to the digital domain tends to be flat under those viewing conditions (that is, when the Nyquist is low when expressed in visual frequency). In this case, visual masking provides more leverage for improving the visual quality.

Visual masking is a perceptual phenomenon where signals are locally masked (i.e., hidden) by a background texture. In compression applications the image acts as a background that reduces the visibility of the false signals generated by the distortion. JPEG 2000 supports the exploitation of self-masking [8], neighborhood masking [24] and point-wise extended masking [32,12], as will be discussed in this section. The visual masking approaches in JPEG 2000 allow bitstream scalability, as opposed to many previous works [27,23,10].

3.1. Psychophysics background for masking

The design of a compression system that exploits visual masking effects is based on psychophysical data for the threshold versus masking contrast, as shown in Fig. 6. These curves describe the elevation of threshold, which in the context of compression relates to the maximum allowable distortion. Also note that the inverse of threshold is the visual sensitivity. The data show the visual system's behavior for two types of masking patterns. One type is noise, having uncorrelated phase and whose results are shown as the dashed line. The shape of this result will occur if the noise is white or narrow band. The other key type of mask is a sine wave, which is entirely correlated in phase. At low mask contrasts, the threshold is the same as if it was presented on a uniform field (zero contrast). This is true for both noise and sine masks. As the contrast increases for the noise

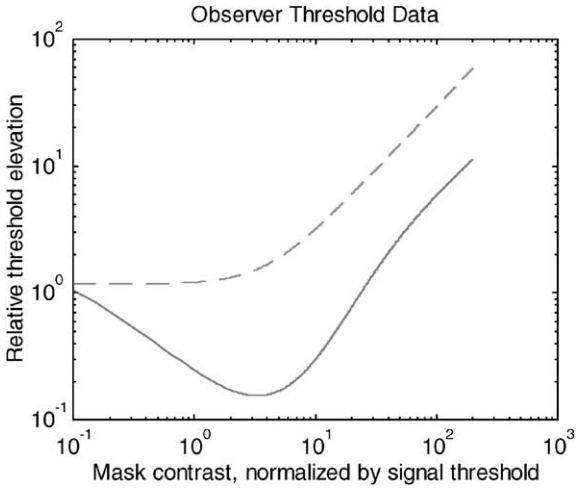

Fig. 6. Threshold versus masking contrast for noise mask (dash line) and sine mask (solid line).

mask, the threshold initially stays constant but then the slope increases until it reaches a constant slope in this log–log plot. The plot can be described by the two asymptotic regions; one with a zero slope for low contrast and one with a slope near 1.0 for high noise contrast.

The data for sine masking are shown as the solid line and there is an additional region where the threshold is actually reduced from that of the uniform field. This region indicates that masking is not occurring, but rather the opposite, where the background masking content actually makes the visual system more sensitive. This effect is referred to as facilitation, and the curve shape is referred to as the dipper effect. This type of masking usually displays a lower slope for high contrasts, and a value of 0.7 is typical. Actual images consist of regions that are various blends between these types of masks. For a detailed discussion of how the masking functions within the spatial frequency channels of the visual system, as well as how it is affected by global frequency weighting of the CSF, see [7]. More recent work in attempting to unify the understanding of masking by patterns and in natural images can be found in [29].

One way to use this effect appears in [25,28]. The quantization as a function of coefficient amplitude is shown in Fig. 7 (from [28]), and is given by the

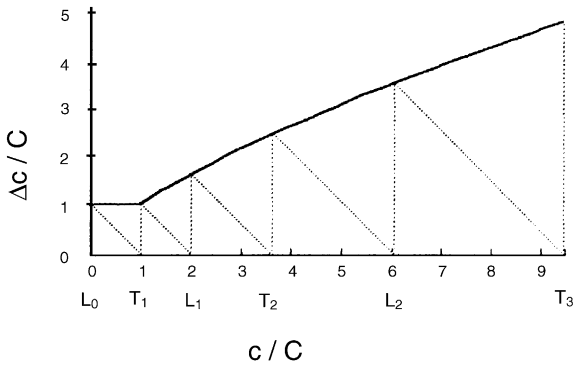

Fig. 7. Encoder quantization (from [28]).

equation [25],

$$Q = \begin{cases} g^{-1}C_{\mathrm{TO}} & \text{if } C_{\mathrm{b}} < C_{\mathrm{TO}}, \\ g^{-1}C_{\mathrm{TO}}\left(\dfrac{C_{\mathrm{b}}}{C_{\mathrm{TO}}}\right)^{s} & \text{otherwise}, \end{cases} \tag{4}$$

where Q is the quantization interval, g is the gain of the coefficient to display contrast, C_{TO} is the visual contrast threshold for the band, C_{b} is the contrast of the wavelet band coefficient to be quantized, and S is the masking slope, which is usually between 0.5 and 0.7. In the figure, the x-axis c/C is analogous to mask contrast, the quantization levels are L_0, L_1, L_2, etc., and the quantization intervals derive from the thresholds, T_1, T_2, T_3, etc.

Further work in applying masking to compression suggested its application to a Cartesian-separable wavelet transform [30], which is computationally more efficient than the Cortex transform but less accurate with respect to the visual system. The quantization strategy prior to entropy coding was suggested to be

$$Q_{\mathrm{b}} = D_{\lambda\theta}\sqrt{1+\sigma^2_{\lambda\theta}}, \tag{5}$$

where Q_{b} is the quantization scale factor of a band b having wavelength λ and orientation θ. The value $D_{\lambda\theta}$ is the visual threshold for that band b. The variance, $\sigma^2_{\lambda\theta}$, is that of the band and possibly neighboring orientation bands taken over a local area. The effect of using this masking was never demonstrated in the paper, however. This would have caused the resulting quantization to be

applied to a coefficient as a function of its band's σ, which for the AC bands is proportional to contrast. The resulting quantization makes the high-contrast asymptotic power-function slope equal to 1.0, rather than 0.7. In [10], an algorithm that locally adapts the quantizer step size at each pixel according to an estimate of the masking measure is presented. To eliminate the overhead, it exploits masking based on an estimate of the current coefficient value from neighboring already coded coefficients. The estimate, however, may not be accurate given that the coefficients are pretty much de-correlated. It is not amenable to scalable coding. Another approach is to implement the nonuniform quantization by applying a nonlinearity prior to a uniform quantizer. This will require a different nonlinearity from that shown in Fig. 6. Such a nonlinearity would be derived from the integral of the threshold data. That is the approach employed in JPEG 2000, and is to be discussed in this section.

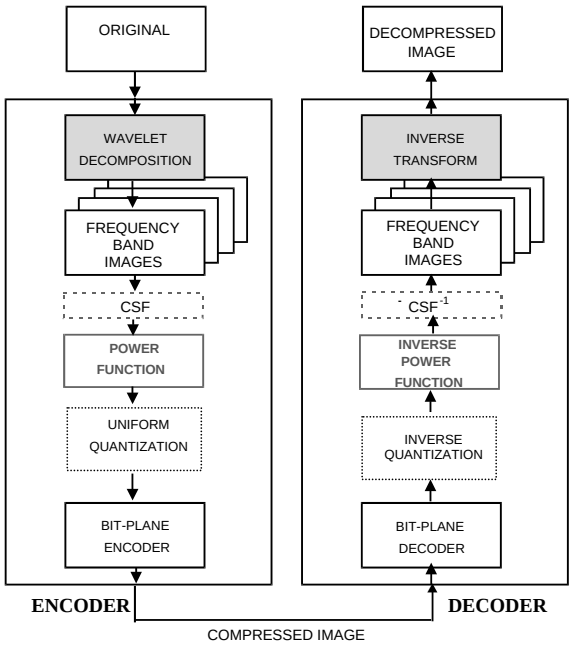

Fig. 8. Bock diagram for the self-masking approach in JPEG 2000.

3.2. Self-contrast masking

It is understood nowadays that the masking property of human vision primarily occurs locally within spatial frequency channels that are each limited in radial frequency as well as orientation. It is then possible to exploit the masking effects by nonuniform quantization which quantizes more coarsely as a function of the activity in spatial frequency and spatial location [28], as opposed to overtly adaptive techniques such as [23,6,20]. Since these masking effects are approximately the same in each channel, once normalized, the same masking procedure could be used in each channel without incurring any overhead [8].

The basic idea of this technique is to use a de-accelerating nonlinearity, referred to as a transducer function, prior to a uniform quantizer within a compression system. Ideally, a scaled derivative of these transducer functions equals the threshold function of Fig. 6. The block diagram for the system in JPEG 2000 is shown in Fig. 8. Basically, at the encoder, a power function can be used to capture the essence of the transducer function, i.e.,

$$y = x^{\alpha}, \quad 0 < \alpha \leqslant 1, \tag{6}$$

is applied to each coefficient, prior to uniform quantization and bit-plane coding. The output of the transducer function is regarded as being the perceived visual response, which is perceptually uniform. At the decoder, the inverse process is applied. Since a coefficient's quantization increase (i.e., coefficient masking) is entirely determined from that coefficient's value, we refer to this technique as self-contrast masking.

Fig. 8 shows that the band coefficient images are generally scaled in a calibration step so the coefficients are linearly scaled prior to their transform by the non-linearity. This scaling can be band-dependent, and is done for optimizing to the frequency characteristics of the visual system, shown dashed in the figure. Consequently, the x-axis of Fig. 9 should be regarded as relative amplitudes.

The decoder nonlinearity, or inverse transducer function, is shown in Fig. 10, for both the noise-based and sine-based masking curves. Since higher slopes at the decoder magnify the quantization error more, we see that less quantization error is allocated to the lower amplitude coefficients than

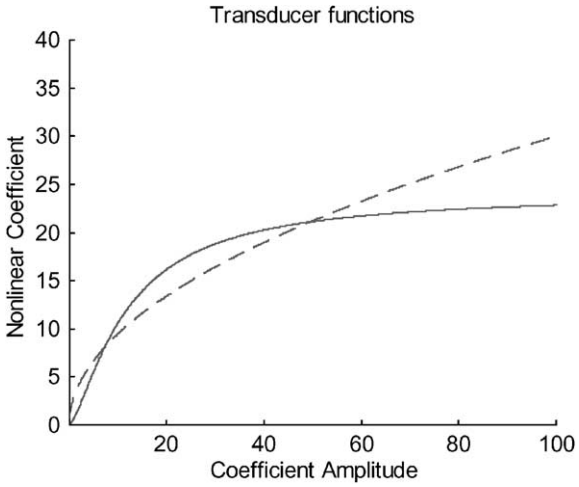

Fig. 9. Nonlinearity at the encoder for self-masking. Dashed line for noise masking, solid line for sine wave masking.

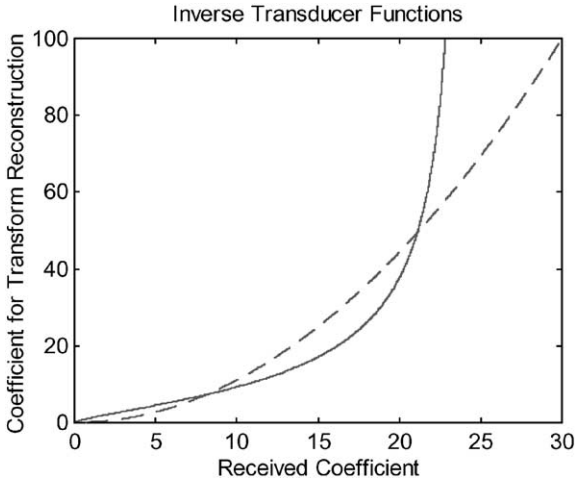

Fig. 10. Nonlinearity at the decoder for self-masking. Dashed line for noise masking, solid line for sine wave masking.

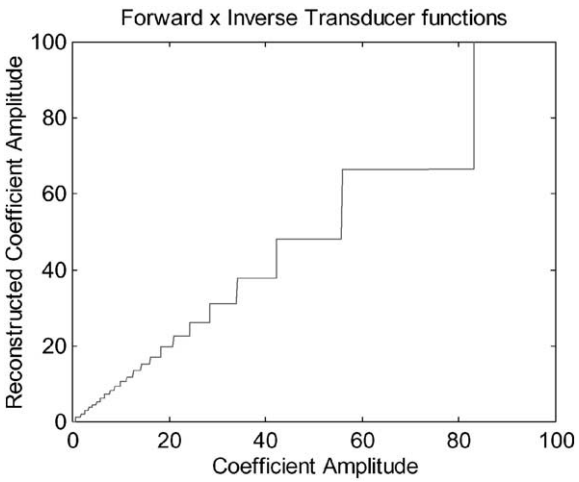

Fig. 11. Cascaded encoder/decoder nonlinearities with quantization.

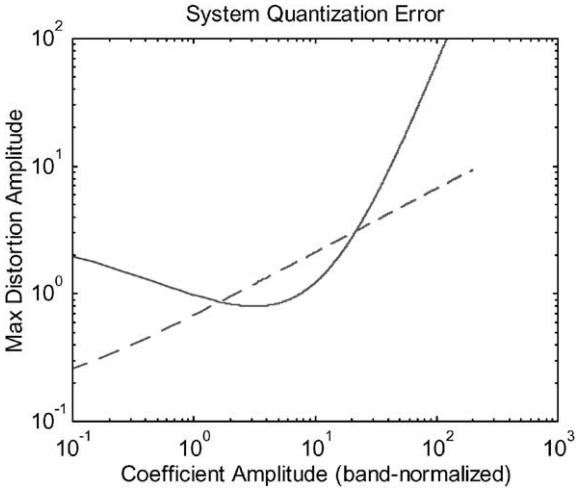

Fig. 12. Error size as a function of signal amplitude.

the higher. With quantization, the cascade effect is essentially non-uniform quantization, as shown in Fig. 11 for the sine-based masking curves of Figs. 9 and 10. It has a small region of slightly increased quantization near zero due to the dipper effect, as well as the vastly increased quantization for high coefficient values. As stated previously, these x-axis amplitudes are, of course, relative.

In general, the resulting step size as a function of coefficient amplitude is given by the derivative of the transducer function at the encoder. This is shown in Fig. 12 for the power function as a dashed line and the sigmoidal function as a solid line. This curve should be compared to the psychophysical data in Fig. 6.

In addition to the actions of the nonlinearity described before, there are a few specific details. The first of these is that the nonlinearity is not employed on specific bands. There are bands eliminated based on their location in the wavelet decomposition pyramid. For example, this

nonlinearity should never be employed on the baseband of the pyramid, that is, the band that includes the DC value of the image. It can also be advantageous to avoid applying masking to low-frequency bands, since there is little to gain in bits from these bands. It should be noted that, in Fig. 8, the frequency weighting should be applied prior to the masking non-linearity. It acts as a calibration step between coefficient and visual threshold as a function of frequency.

While consideration was given toward allowing the users to design their own transducer function (to use a sine-based or noise-based image model, for example), it proved difficult to preserve this flexibility in the JPEG 2000 VM. Consequently, only a power function nonlinearity is allowed in the standard, since its shape can be conveyed with a single parameter. The exploitation of self-contrast masking could be invoked in the VM software using the "-Xmask" option with the parameter β set to 0. A good value for α is 0.7. Note that due to the derivative relation between transducer and quantization result, the value of α corresponding to a slope of 0.7 in Fig. 6 should be 0.3. However, experiments have shown that a good value for α is the less aggressive 0.7. This is partly because there exists a mismatch between the wavelet band structure and the HVS's band structure.

3.3. Problems with self-masking in wavelet compression

Since the self-masking was designed to be closely based on current spatial models of the visual system, it should be expected to properly reallocate bits to where (in terms of spatial and frequencies) the viewer is most sensitive, thus preventing visible distortions. However, the correspondence between the visual system and wavelet structure of the compression algorithm is only approximate. Not surprisingly, issues delineating the compression algorithm from the visual model lead to non-optimal visual performance, and distortion artifacts of the compression process can indeed be more visible than expected as the bit-rate is reduced (see, e.g., Fig. 15). These issues will be discussed below.

3.3.1. Diagonal band

The most well-known difference between this implementation and a good visual model lies in the 2D spatial arrangement of the filter bank, which stems from the Cartesian-separable filter construction used in the wavelet decomposition. One of the key problems is the fact that the diagonal band contains the mixed orientations [13,1] of 45° and -45°. The reason this causes problems with masking is that in the visual system, energy near 45° does not significantly mask energy near -45°, whereas in the Cartesian-separable wavelet implementation such cross diagonal masking will occur. So at a diagonal edge, the visual system's masking would hide oriented distortions parallel to the edge, but those orthogonal would be visible. However, in the self-masking wavelet implementation, the quantization of coefficients in a diagonal edge leads to distortions both parallel as well as perpendicular to the edge, and the perpendicular distortions are easily visible, since they are not visually masked.

3.3.2. Horizontal and vertical bands encroaching on diagonal frequencies

Less well known is a problem due to the shape of the filters. Notice that along the diagonal frequencies, the horizontal and vertical bands encroach into the diagonal region at multiples of $1/2^n$ cycles/pixel (where n is the resolution level). The energy of a diagonal (D) edge can end up in the H and V bands. If the edge contrast is high enough, the amplitudes of the coefficients related to the edge in the H and V bands may be high enough so that the resulting masking effect can increase their quantization intervals. This can cause H and V linear distortions along a diagonal edge, which will easily be seen by the visual system. Further, the energy displaced away from the diagonal band into the horizontal and vertical bands will be energy that is not taken into account in the masking of diagonal structures. Any energy that causes masking in the visual system but is lost by the masking structure of the algorithm represents a higher bit rate. In rate-controlled implementations, this lost masking energy will lead to higher distortions.

3.3.3. Zero-crossings and phase sensitivity

Another serious deviation of the algorithm's masking from the visual masking is the issue of phase. The visual channels have limited phase sensitivity (i.e., phase uncertainty), which is greater than 90° for an isolated visual channel, but can be as low as 45° phase for signals with two adjacent channels [2,3]. Due to this limited phase sensitivity and since the masking effect is based on local activity in a visual channel, it is not solely limited to the peaks and valleys of a waveform. It can also extend across the zero-crossings. The effect is shown in Fig. 13 for range of phase uncertainties. In many visual models, this phase uncertainty is caused by the visual channel being a quadrature phase summation of sine and cosine receptive fields, modeled as

$$R(x) = (\sin^2(2\pi fx) + \cos^2(2\pi fx))^{1/2}, \tag{7}$$

where *R is the response of* a receptive field with dominate frequency f. This can be generalized to any phase uncertainty, $\Delta\theta$, behavior by

$$R(x) = (\sin^p(2\pi fx) + \sin^p(2\pi fx + \Delta\theta))^{1/p}. \tag{8}$$

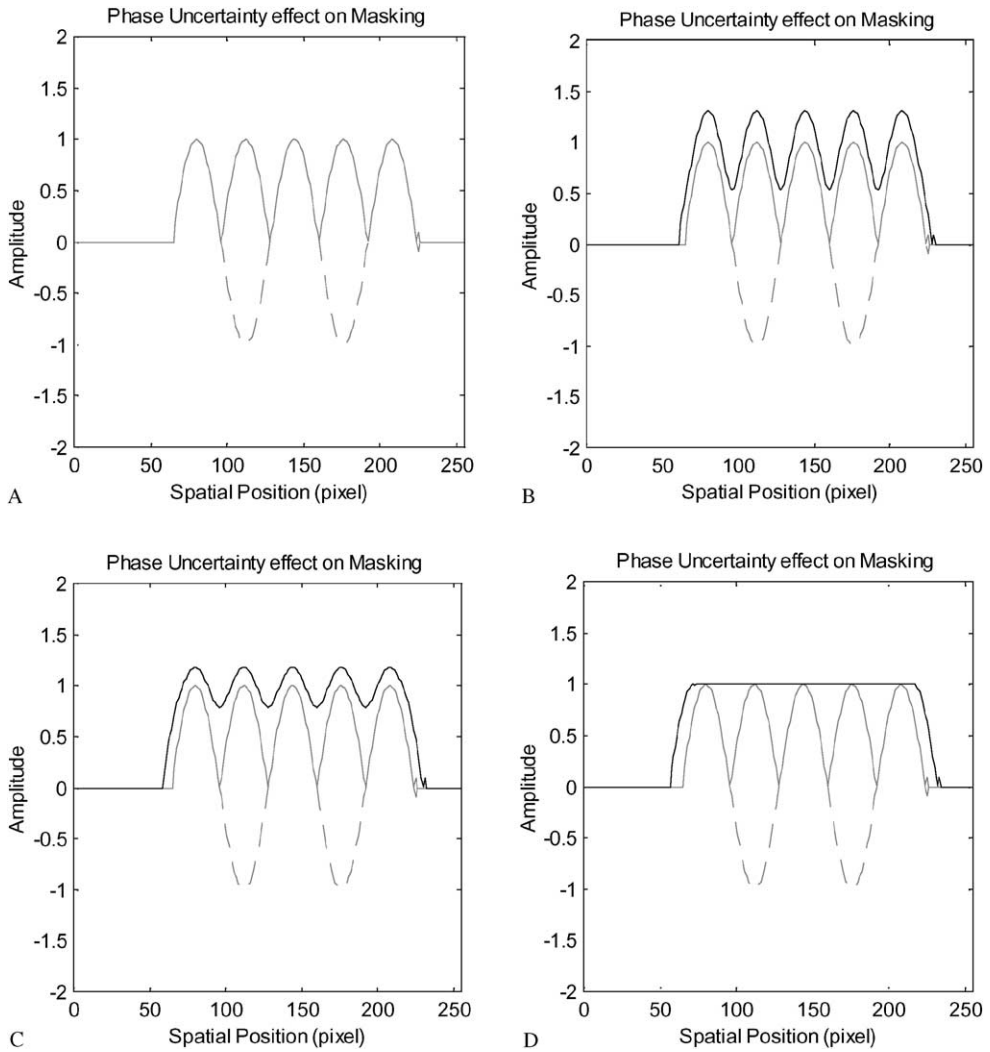

Fig. 13. Effect of phase uncertainty on spatial extent of masking (B) $\theta = \pi/4$; (C) $\theta = 3\pi/8$; (D) $\theta = \pi/2$.

In part A of the figure, the dashed line shows a localized sine wave, representing one band's view of a textured area. The solid line in that figure shows the resulting masking in the band, where we only model the rectification aspect of masking from Eq. (8) and having zero phase uncertainty. The upper line in part B shows the masking caused by a phase uncertainty ($\Delta\theta$) of $\pi/4$ via Eq. (8), while C shows a phase uncertainty of $3\pi/8$ and D shows a $\Delta\theta$ of $\pi/2$ (all with $p=2$). If such a phase uncertainty of $\pi/2$ is approximately equal to that of the visual system, it results in a uniform masking over the textured area.

In the self-masking implementation of JPEG 2000, the masking is confined to the coefficient having a phase width near zero degrees (for the highest frequency in a band), so the masking due to energy near the zero-crossings is not taken into account. While visual masking may occur over an entire texture region in actual images, the self-masking approach of the compression process can only take into account the masking at the peaks and valleys of the textured region, and not the entire region. This means that entropy savings in these regions is lost, thus driving up the bit rate or driving down the quality in other frequencies and regions.

3.4. Neighborhood masking

Another way of exploiting visual masking is through the control of individual code-block contribution in the quality layer formation process [24]. In this approach, the embedded coding of each individual code-block is performed without considering visual masking effect. In particular, there is no non-linearity interspersed between the wavelet transform stage and the quantization stage. However, in the post-compression rate-distortion optimization process, the distortion metric is modified to take into account the visual masking effect. More specifically, the distortion of each coefficient is weighted by a visual masking factor that measures the local texture activity and is in general a function of the neighboring coefficients. That is, it treats each coefficient value, V_i, as though it were equal to V_i' (from the perspective of distortion estimates), where

$$V_i' = V_i / M_i \tag{9}$$

and the masking strength function is

$$M_i = A \sum_{\{\text{k near i}\}} |V_k|^{\Upsilon}, \tag{10}$$

with A being a normalization factor, and Υ assumes a value between 0 and 1. Note that although each coefficient may have a different value of the visual masking factor M_i, this approach can only adjust the truncation points of each code-block to form different quality layers. We will refer to this type of masking as block-based neighborhood masking. This approach adjusts only the distortion metric at the encoder, which is an advantage from an implementation point of view. The decoder does not need to be aware of that. The masking effect exploited can also be spatially extensive which is not exploited in the self-masking approach discussed in Section 3.2. Another advantage is that it allows a lossless embedded bitstream to be generated, since integer implementation is feasible. This approach works very well for large images with diverse contents such as the "woman" image. Its weakness is that it can only adjust the truncation points of each code-block (i.e., the bit-allocation is code-block-based), which is a spatially coarser adjustment than the sample-by-sample compensation offered by the self-masking approach [8]. Within a code-block, no visual masking effect is exploited. As a result, it may not work very well for smaller images. This block-based neighborhood masking is accessible via the "-Cvis" option in the JPEG 2000 VM software.

3.5. Point-wise extended masking

A more comprehensive visual masking approach has been developed [32,12] that extends the point-wise "non-linearity" of self-masking [8] to an "extended nonlinearity". It also takes care of the masking effect and spatial summation contributed from spatially neighboring coefficients. This is to overcome the over-masking problem of the self-masking approach [8] that occurs at diagonal

edges, as discussed in Section 3.3. The main advantage of this strategy is its ability to distinguish between large amplitude coefficients that lie in a region of simple edge structure and those in a complex region, such as texture. This feature will assure the good visual quality of simple edges in a smooth background, which is often critical to the overall perceived quality.

This point-wise extended masking approach treats visual masking as a combination of two separate processes. The first step is to apply a point-wise power function to the original coefficient x_i, i.e.,

$$x_i \to y_i = \mathrm{sign}(x_i)|x_i|^{\alpha}. \tag{11}$$

This is basically to account for the self-masking effect. This step assumes each signal with which a coefficient is associated is lying on a common flat background. Under this assumption, $\{y_i\}$ are perceptually uniform. In a real image, however, this is usually not the case. Each signal is superimposed on other spatially neighboring signals. There is some masking effect contributed from spatially neighboring signals due to the phase uncertainty, receptive field sizes, as well as possible longer range effects that increase detection (“pooling”) [8]. To further exploit this neighborhood masking effect, the second step normalizes y_i by a neighborhood masking factor m_i that is a function of the amplitudes of the neighboring signals. A good model that has been adopted by the JPEG 2000 standard [12] is to use the nonlinear transform

$$z_i = y_i \Big/ \left(1 + a \sum_{\{\mathrm{k-near-i}\}} |\hat{x}_k|^{\beta}/|\phi_i|\right), \tag{12}$$

where $|\phi_i|$ denotes the size of a causal neighborhood, a is a normalization factor with a constant value of $(10000/2^{\mathrm{bitdepth-1}})^{\beta}$ and *bitdepth* denotes the bit depth of the original image, $\hat{x}_k$ denotes the quantized (bit-truncated) neighboring coefficients (that only retain the first few most significant bits of the quantization index to allow for embedded coding), and the neighborhood contains coefficients in the same band that lie within an $N \times N$ window centered at the current coefficient. These neighborhood coefficients also appear earlier than the current coefficient in the raster scan order (see Fig. 14 for an example). The neighborhood does not include the current coefficient itself so that an explicit solution for the inverse process is available. The causal neighborhood should also respect code-block boundaries when a “respect_block_boundaries” switch is selected at the encoder. This switch should cause the neighborhood masking weighting factor m_i not to include coefficients outside of the current code-block. Also, this switch must be transmitted to the decoder to tell it exactly how the neighborhood is formed. When the switch is on, it allows parallel implementation and restricts error propagation, but it may sacrifice some performance. The parameter α assumes a value between 0 and 1, and is used to control the degree of self-masking. A typical value of α is 0.7. The parameter β assumes a positive value, and, together with N, are used to control the degree of neighborhood masking. The parameters β and N play important roles in differentiating coefficients around simple edge from those in the complex area. The degree of averaging is controlled by N; β controls the influence of the amplitude of each coefficient. It is important that β assumes a value much smaller than 1, and a good value of β is 0.2. This is quite different from some previously proposed variance-based neighborhood activity measure [23,10]. It helps to protect coefficients around simple sharp edges, since the coefficients around sharp edges usually have high values. A variance-based measure may not be able to distinguish a local sharp edge area (with few large coefficients and all the rest close to zero) from a local complex area (with many mid-amplitude coefficients). This is because

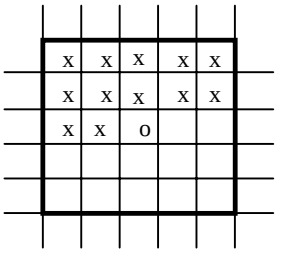

Fig. 14. An example of the causal neighborhood ($N = 5$, $|\phi_i| = 12$). o: current coefficient; x: coefficients in the causal neighborhood.

the large coefficients, although only a few, in a local sharp edge area could contribute significantly to the overall variance, due to the square operation. Note that masking is lower than expected near sharp edges (as opposed to textures) due to "pooling". A small value of β suppresses the contribution of a few large coefficients around sharp edges to the neighborhood masking factor, thus implicitly distinguishing coefficients around sharp edges from coefficients in a complex region. For example, two neighborhood sets of {5, −5, 5, −5, 5, −5, 5, −5} and {0, 0, 0, 10, −10, 0, 0, 0} have the same variance of 5. But their "L0.2-norms" are 1.38 and 0.40, respectively.

The resultant z_i values are then subject to uniform quantization. The inverse is performed at the decoder. Note that quantized neighboring coefficients will be used at the encoder to ensure that both the encoder and the decoder perform exactly the same operation to calculate m_i. For embedded coding, unfortunately, the encoder cannot do the nonlinear transformation based on the exact final compressed/quantized version of the coefficient x_k because the "extended nonlinearity" is applied prior to scalable compression, and the decoder can have any bitstream that has a lower rate than the final rate. Nevertheless, the discrepancy of m_i calculated at the encoder and the decoder can be completely eliminated or reduced by a conservative strategy where only the *same* very coarsely quantized (i.e., bit-truncated) coefficients are used to calculate the masking weighting factor m_i at both the encoder and the decoder. In fact, after z_i is quantized, only the *Bits_retained* most significant bits of the quantization index will be retained (the rest bits are replaced with 0). This modified quantization index is then converted back to the x domain, and is used for calculating m_i, As long as *Bits_retained* is small enough (with respect to the available bit rate at the decoder), the decoder will be able to obtain exactly the same bit-truncated version of the neighboring coefficients. The compromise here is a coarser granularity of m_i which may slightly affect the accuracy of the masking model. However, experiments have suggested that the performance usually is not very sensitive to which quantized version of the neighboring coefficients is used. As a result, bitstream scalability is feasible. This is essentially a coefficient-wise adaptive quantization without any overhead. The system diagram of the point-wise extended masking approach is the same as Fig. 8 except that the power function is replaced by the extended non-linearity presented in Eq. (12). Note that visual masking may be applied to all frequency levels that have an index value not less than a particular level *Minlevel* which can be specified in the bitstream. It should not be applied to the lowest frequency band (the DC band).

Fig. 15 shows that the point-wise extended masking approach significantly improves the visual quality for the "woman" image. What happens here is that the neighborhood masking factor m_i assumes a smaller value for those coefficients in the simple sharp edges (e.g., fingers) than those in the more complex areas (e.g., sweater, see Fig. 16). As a result, more bits are allocated to improve the simple sharp edge areas while the more complex areas are allocated less bits to take advantages of the texture masking effects.

4. Discussions

The various visual optimization tools in JPEG 2000 have their own merit and weakness. The visual frequency weighting is usually very effective for large viewing distances or high-resolution displays, but it is tied to a specific viewing condition. Under different viewing conditions, the perceived quality can vary a lot. In other words, the weights used at the encoder have to match the viewing condition under which the image is to be viewed. When using a viewing distance for an application or image study, it is important to use a frequency weighting set for the closest distance expected. Three sets of CSF weights have been recommended in JPEG 2000 for some common viewing/printing scenarios. These are csf1000, csf2000 and csf4000, where 1000, 2000 and 4000 refer to the viewing distance in pixels. Unlike the JPEG default, these are based solely on the CSF and hence, do not include any display MTF effects, such as the CRT MTF implicitly occurring in the JPEG default tables.

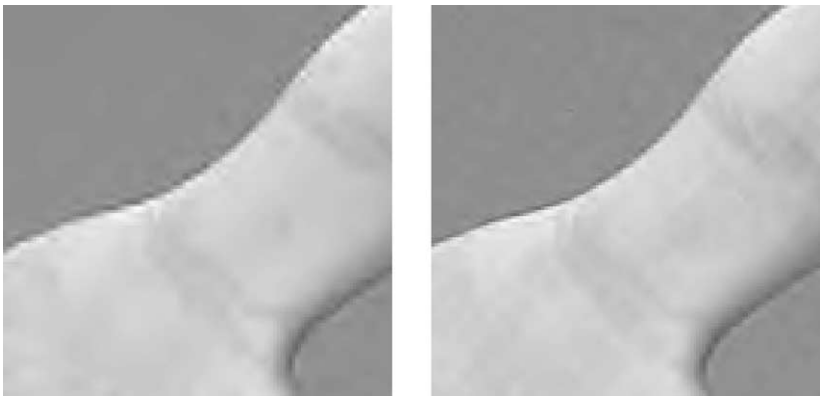

Fig. 15. Self-masking result (left) versus point-wise extended masking result (right) at 0.25 bpp.

Fig. 16. A neighborhood masking factor (m_i) map for the point-wise extended masking approach. The amplitudes have been amplified for display purpose.

We decided to omit the display MTF in the default, since with today's technology it is equally as likely that the display will be a direct view LCD, DLP projector, or hardcopy as it will be a CRT.

Fig. 17 shows some visual comparison between JPEG 2000 and JPEG, both using frequency and color weighting (the default Q tables were used for JPEG using the IJG implementation of JPEG with optimized Huffman table, and Table 2 that corresponds to a viewing distance of 1700 pixels was used for JPEG 2000). The results are based on 6 observers and 6 color images, using 300 dpi printing. The graph shows the bit rates of JPEG 2000 against JPEG to achieve similar visual quality. For example, it costs JPEG 0.53 bpp to achieve similar visual quality as using JPEG 2000 at 0.25 bpp. The dashed line represents the reference points where JPEG and JPEG 2000 assume the same bit rates. Overall, JPEG 2000 provides a bit rate saving of 10–50%, to achieve similar visual quality.

The visual masking approaches usually are less sensitive to the viewing condition. The self-masking approach usually protects the fine texture well, which is especially suitable for high-quality photographic images that contain human faces. It, however, may have some problems with sharp edges, especially at low bit rates. The block-based neighborhood masking approach usually tends to smooth out the fine texture a little bit, but protects high contrast edges well. It also has some limitations for relatively small images, mainly due to its block-based nature. It, however, has successful performance for large images with diverse content. The point-wise extended masking approach combines the strength of both self-masking and neighborhood masking, thus resulting in mutual synergism. All three masking approaches we discussed achieve the adaptive behavior without explicit segmentation, edge detectors, and overhead bits. They all allow bitstream scalability, which is very important in many applications. In general, they improve the image quality for cases where the CSF weighting does not offer much advantage. The major improvement areas are the low amplitude texture

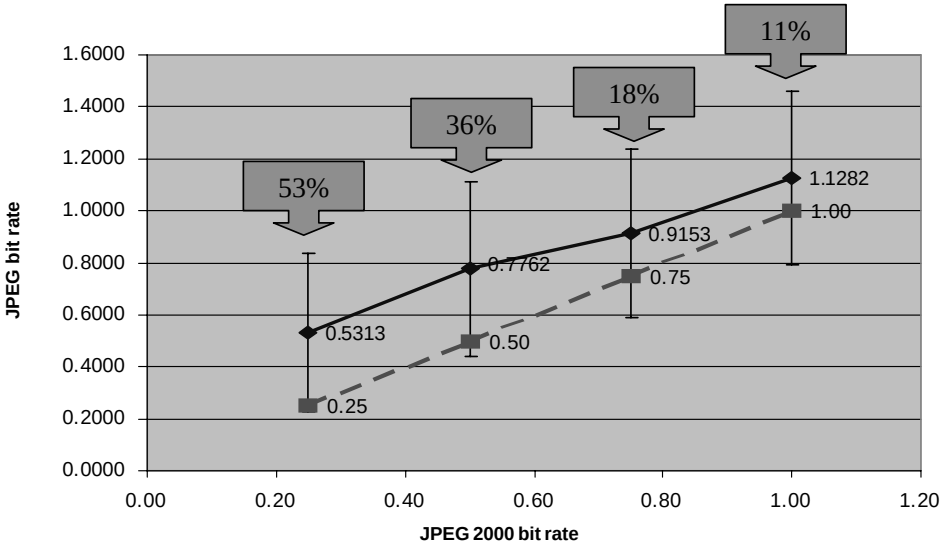

Fig. 17. Visual comparison between JPEG 2000 and JPEG. Courtesy of Troy Chinen and Alan Chien [4].

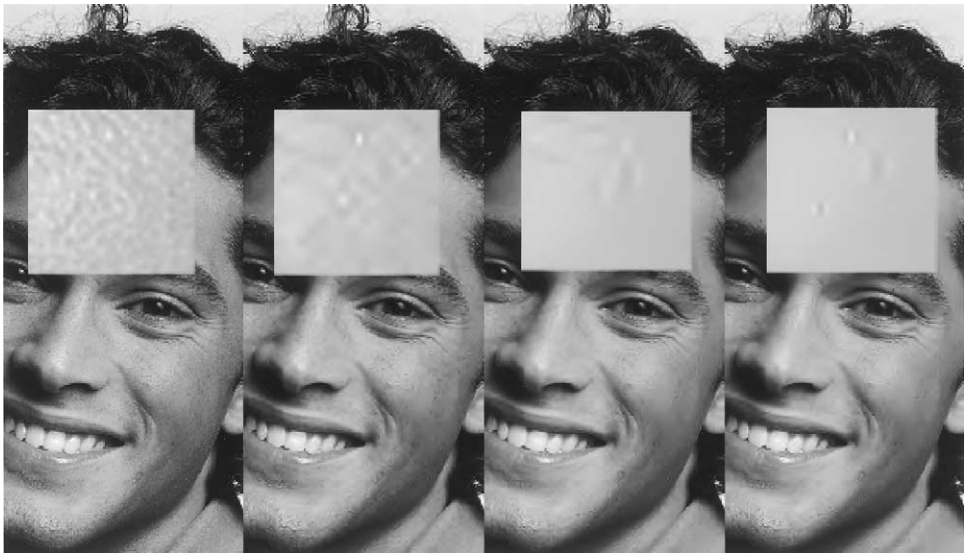

Fig. 18. Advantageous results of self-masking approach coded at 0.5 bpp, with close-up view of the foreheads.

and high-contrast sharp edges. Fig. 18 shows some results of the self-masking approach, as compared to frequency weighting case and the case where no visual tool is applied. Self-masking preserves the low amplitude texture better than the other two. This is more evident in the close-up views of the foreheads. Some comparisons between different masking approaches are shown in Fig. 19. It can be seen that the point-wise extended masking approach preserves the fine details best,

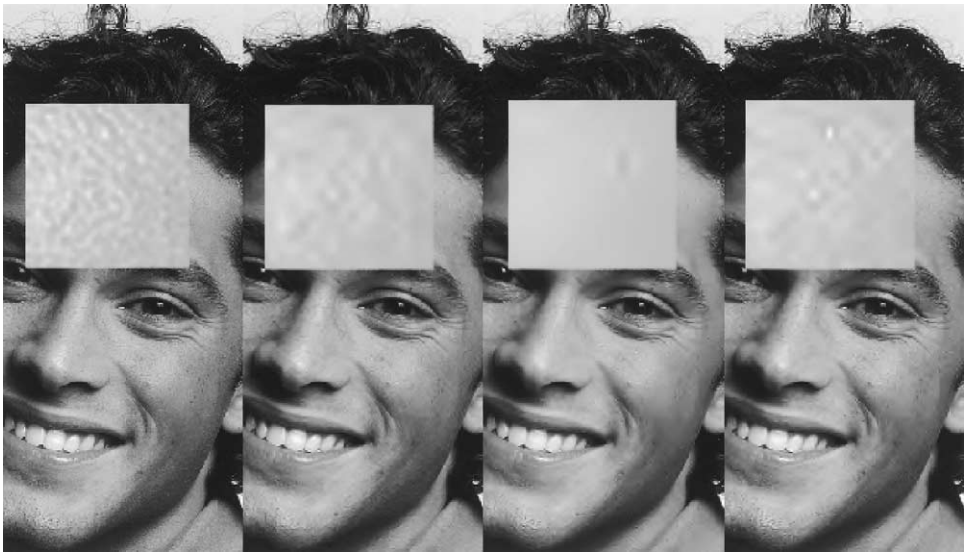

Fig. 19. Comparison of different masking approaches, coded at 0.5 bpp, with close-up view of the foreheads.

as compared to block-based neighborhood masking, and self-masking alone. The block-based neighborhood masking approach does not seem to work well on this relatively small size image (512×768).

The various visual optimization tools can in fact be combined together to maximize the visual performance. It has been observed that, for some complex images with diverse content, the visual improvement can be equivalent to a saving of up to 50% in bit rate. Finally, as mentioned before, JPEG 2000 also supports the exploitation of other HVS properties such as local light adaptation, eccentricity and temporal frequency sensitivity. These could be some of the future research topics.

Acknowledgements

We would like to thank Jin Li of Microsoft Research China, David Taubman of University of New South Wales, Marcus Nadenau and Julien Reichel of EPFL, Troy Chinen of FUJIFILM Software, Tom Flohr of SAIC, Alan Chien of Eastman Kodak, Margaret Lepley of MITRE, and many others in the JPEG committee for their contributions and support to the visual optimization work in JPEG 2000.

References

[1] E. Adelson, E. Simoncelli, Hingorani, Orthogonal pyramid transforms for image coding, SPIE Proc. 845 (1987) 150–58.

[2] T. Caelli, P. Bevan, Visual sensitivity to two-dimensional phase, Vis. Res. 72 (1982) 1375–1381.

[3] T. Caelli, M. Hubner, I. Rentschler, Detection of phase-shifts in two-dimensional images, Percept. Psychophys. 37 (1985) 536–542.

[4] T. Chinen, A. Chien, Visual evaluation of JPEG 2000 color image compression performance, ISO/IEC JTC1/SC29/WG1 N1583, Tokyo, March 2000.

[5] T. Chinen, M. Nadenau, J. Reichel, W. Zeng, Report on CE C03 (Optimizing color image compression), ISO/IEC JTC1/SC29/WG1 N1587, Tokyo, March 2000.

[6] M.R. Civanlar, S.A. Rajala, W.M. Lee, Second generation hybrid image coding techniques, in: SPIE V. 707 VCIP, 1986, pp. 132–137.

[7] S. Daly, The visible differences predictor: an algorithm for the assessment of image fidelity, in: A.B. Watson (Ed.), Digital Images and Human Vision, MIT Press, Cambridge, MA, 1993, pp. 179–206, Chapter 14.

[8] S. Daly, W. Zeng, J. Li, S. Lei, Visual masking in wavelet compression for JPEG 2000, in: Proceedings of IS&T/SPIE Conference on Image and Video Communications and Processing, San Jose, CA, Vol. 3974, January 2000.

[9] T. Fukuhara, K. Katoh, S. Kimura, K. Hosaka, A. Leung, Motion-JPEG 2000 standardization and target market), in: Proceedings of the IEEE International Conference on Image Processing, September 2000, Vancouver, Canada, 2000.
[10] I. Hontsch, L. Karam, APIC: adaptive perceptual image coding based on subband decomposition with locally adaptive perceptual weighting, in: Proceedings of the IEEE International Conference on Image Processing, Santa Barbara, CA, 1997, pp. 37–40.
[11] Information Technology—JPEG 2000 Image Coding System, ISO/IEC FDIS15444-1: 2000, August 2000.
[12] Information Technology—JPEG 2000 Image Coding System: Extensions, ISO/IEC FCD15444-2: 2000, December 2000.
[13] P. Jones, S. Daly, R. Gaborski, M. Rabbani, Comparative study of wavelet and DCT decompositions with equivalent quantization and encoding strategies for medical images, in: SPIE Proceedings of Conference on Medical Imaging, Vol. 2431, 1995, pp. 571–582.
[14] JPEG 2000 verification model 7.0 software, ISO/IEC JTC1/SC29/WG1 N1685, April 2000.
[15] J. Li, Visual progressive coding, in: Proceedings of IS&T/SPIE Conference on Visual Communications and Image Processing, San Jose, CA, Vol. 3653, January 1999.
[16] J. Li, W. Zeng, S. Lei, Sharp rate-distortion optimized embedded wavelet coding—an algorithm proposal for JPEG 2000, ISO/IEC JTC1/SC29/WG1 N621, Sydney, Australia, October 1997.
[17] M. Nadenau, J. Reichel, Compression of color images with wavelets under consideration of the HVS, in: Proceedings of SPIE Human Vision and Electronic Imaging, Vol. 3644, San Jose, CA, January 1999.
[18] M.J. Nadenau, J. Reichel, Opponent color, human vision and wavelets for image compression, in: Proceedings of the Seventh Color Imaging Conference, IS&T, Scottsdale, AZ, 16–19 November 1999, pp. 237–242.
[19] M. Nadenau, J. Reichel, Report on CE V2 (Performance analysis of csf-filtering compared to fixed weighting using recommended parameter set), ISO/IEC JTC1/SC29/WG 1 N1470, Maui, Hawaii, December 1999.
[20] T.N. Pappas, T.A. Michel, R.O. Hinds, Suprathreshold perceptual image coding, in: IEEE, ICIP, Lausanne, Switzerland, 1996, pp. 237–240.
[21] C. Poynton, A Technical Introduction to Digital Video, Wiley, New York, 1996, p. 92.
[22] J. Roufs, Dynamic prporties of vision—I, Vis. Res. 12 (1972) 261–278.
[23] R.J. Safranek, J.D. Johnston, A perceptually-tuned subband image coder with image dependent quantization and post-quantization data compression, in: IEEE International Conference on Acoustic, Speech, and Signal Process, 1989, pp. 1945–1948.
[24] D. Taubman, High performance scalable image compression with EBCOT, IEEE Transactions on Image Processing 9 (7) (July 2000) 1158–1170.
[25] C. van den Branden Lambrecht, A working spatio-temporal model of the human visual system for image restoration and quality assessment applications, in: Proceedings of ICASSP, 1996, pp. 2291–2294.
[26] V. Virsu, J. Rovano, Visual, resolution, contrast sensitivity, and the cortical magnification factor, Exp. Brain Res. 37 (1979) 475–494.
[27] A.B. Watson, DCT quantization matrices visually optimized for individual images, in: Proceedings of the SPIE Conference on Human Vision, Visual Processing, and Digital Display IV, Vol. 1913, 1993, pp. 202–216.
[28] A.B. Watson, Efficiency of an image code based on human vision, J. Opt. Soc. Am. A 4 (1987) 2401–2417.
[29] A.B. Watson, M. Taylor, R. Borthwick, Image quality and entropy masking, in: Proc. Human Vision, Visual Processing, and Digital Display VIII, San Jose, CA, SPIE, Bellingham, WA, 1997, Vol. 3016, pp. 2–12.
[30] A.B. Watson, G. Yang, J. Solomon, J. Vilasenor, Visibility of wavelet quantization noise, IEEE Trans. Image Proc. 6 (8) (1997) 1164–1175.
[31] W. Zeng, Report on CE V1 (Evaluation of the distortion-adaptive progressive CSF weighting technique), ISO/IEC JTC1/SC29/WG1 N1716, Arles, France, July 2000.
[32] W. Zeng, S. Daly, S. Lei, Point-wise extended visual masking for JPEG 2000 image compression, in: Proceedings of the IEEE International Conference on Image Processing, September Vancouver, Canada, 2000.
[33] W. Zeng, S. Lei, Report on CE V1 (CSF weighting strategy for visual progressive coding), ISO/IEC JTC1/SC29/WG1 N1584, Tokyo, March 2000.