Real-Time Foveation Techniques for Low Bit Rate Video Coding

Hamid R. Sheikh, Brian L. Evans and Alan C. Bovik

Abstract

Lossy video compression methods often rely on modeling the abilities and limitations of the intended receiver, the Human Visual System (HVS), to achieve the highest possible compression with as little effect on perceived quality as possible. Foveation, which is non-uniform resolution perception of the visual stimulus by the HVS due to the non-uniform density of photoreceptor cells in the eye, has been demonstrated to be useful for reducing bit rates beyond the abilities of uniform resolution video coders. In this work, we present real-time foveation techniques for low bit rate video coding. First, we develop an approximate model for foveation. Then, we demonstrate that foveation, as described by this model, can be incorporated into standard motion compensation and Discrete Cosine Transform (DCT) based video coding techniques for low bit rate video coding, such as the H.263 or MPEG-4 video coding standards, without incurring prohibitive complexity overhead. We demonstrate that foveation in the DCT domain can actually result in computational speedups. The techniques presented can be implemented using the baseline modes in the video coding standards and do not require any modification to, or post processing at, the decoder.

Index Terms

Real-time Human Visual System Modeling, Foveated Video Coding, Variable Resolution Video Coding.

The authors are affiliated with the Laboratory for Image and Video Engineering, Department of Electrical & Computer Engineering, The University of Texas at Austin, Austin, TX 78712-1084 USA, Phone: (512) 471-2887, email: (sheikh, bevans, bovik)@ece.utexas.edu.

This research was supported in part by Texas Instruments, Inc., and by the State of Texas Advanced Technology Program. The authors wish to thank Youngjun Yoo and Raj Talluri of Texas Instruments' Worldwide Imaging Business Unit for their support.

I. Introduction

THE field of lossy signal compression is directed towards reducing the bandwidth requirements of digitized signals by allowing compression algorithms to introduce distortion in the signal in such a way that it is least perceptible to the receiver for a given bit rate [1]. For lossy image and video compression, the receiver is the Human Visual System (HVS). Knowledge of the abilities and limitations of the HVS makes it possible to design lossy compression methods such that the introduced distortion is least perceptible to the HVS, while the fidelity of those aspects of the signal for which it is more sensitive is retained [2].

Foveation is a layer of HVS modeling that describes its inability to perceive an entire visual stimulus at full resolution because of the non-uniform spacing of sensor neurons. This limitation enables the removal of extraneous resolution information to obtain an increase in compression gain without sacrificing perceived quality. Video coding that incorporates foveation modeling is called *foveated* video coding. Foveated video coding can provide a significant increase in compression gain beyond the abilities of uniform resolution coders [3], [4], [5], [6].

Apart from coding gain, the computational complexity of a compression algorithm plays a vital role in determining its feasibility. Foveation requires extra processing at the encoder. Although fast foveation techniques have been explored previously [3], [7], the need to combine foveation processing with standard-compliant video coding techniques for real-time operation, especially on embedded programmable processors, requires further research into reducing, and possibly eliminating, the complexity overhead. In this paper, we explore ways of efficiently implementing foveated video coding that can actually provide a *reduction* in the computational requirements. We also analyze the effects of foveation on Motion Compensated Prediction (MCP) used in standard video coders, since the prediction and predicted signals will generally not have the same resolution information.

Section II presents background on foveated image and video coding. Section III develops a foveation model suitable for real-time computation. Section IV presents efficient techniques for real-time foveated video coding. Section V gives the results of applying the techniques in Section IV. Section VI makes concluding remarks.

II. Background

A. Motivation

The Human Visual System consists of a complex system of optical, physiological and psychological components that interplay in such a way that the sensitivity of the HVS is different for different aspects of the visual stimulus, such as brightness, contrast, texture, edges, temporal changes, and frequency content. Understanding and modeling the HVS has been helpful in image and video engineering in determining, for example, the frame rate, optimal color spaces, resolution of chrominance components and spectral quantization noise sensitivities of the HVS [8].

In a human eye, the retina (the membrane that lines the back of the eye and on which the optical image is formed) does not have a uniform density of photoreceptor cells. The point on the retina that lies on the visual axis is called the *fovea.* The fovea is a circular region of about 1.5 mm in diameter (see Figure 1) [9] and has the highest density of sensor cells in the retina. This density decreases rapidly with distance (measured as *eccentricity*, or the angle with the visual axis) from the fovea. Whenever the eye is observing a visual stimulus (which may be a still image or a video sequence), the optical system in the eye projects the image of the region at which the observer is fixating onto the fovea. Consequently, only the fixation region is perceived by the HVS with maximum resolution, and the perceived resolution decreases progressively for regions that are projected away from the fovea. We say that the eye *foveates* the visual stimulus it receives. Figure 2 shows an example of foveation. It shows the uniform resolution image (left) and the image that the HVS perceives (right) when the observer fixates at the point marked with 'X'. Any transmission, coding and display of resolution information higher than the perceivable limit is redundant. Images (and video frames) can be foveated by removing this extraneous information prior to encoding [3], [4], [5]. Encoding a foveated image or video with a standard encoder results in a lower data rate than the corresponding uniform resolution version. In Section IV, we present another approach to foveated video coding in which the process of foveation is embedded inside the encoder. This is done in such a way that the bitstream syntax remains standard-compliant.

Foveated video coding may be considered as a case of "regions of varying interest" coding in which the number of regions, their boundaries and their relative importance conform to a model of the human visual system, rather than being application or constraint dependent, as in [10], [11]. Foveated video coding using the techniques presented in this paper can be implemented very simply by using only the baseline modes in most video coding specifications.

B. Foveation Model

The sensitivity of the HVS at different regions in the eye (foveal and peripheral) has been studied in terms of contrast sensitivity functions [12], [13]. Giesler and Perry give a fit to the sensitivity data and derive a contrast sensitivity model [3]. The model relates the sensitivity of the human eye CT to different spatial frequencies, f, presented to a human observer at different eccentricities, e, as

$$CT(f,e) \quad = \quad CT_0 e^{\;\alpha f \frac{e_2+e}{e_2}} \qquad (1)$$

where

$$e = \arctan\left(\frac{\sqrt{(x-x_f)^2+(y-y_f)^2}}{V}\right)$$

This model in (1) assumes a simplification that the eye is equally sensitive at all orientations (see Figure 3).

In (1), CT is the minimum contrast required to perceive a sinusoid of spatial frequency f (in cycles per degree) at an eccentricity e (in degrees), and CT_0, α and e_2 are model parameters (model values from [3] are $CT_0 = \frac{1}{64}$, $\alpha = 0.106$ and $e_2 = 2.3^o$). An intensity sinusoid $A\sin(\cdot)+1$ is said to have a contrast A $(0 \leq A \leq 1)$ where the maximum possible value of the sinusoid is white and the minimum is black. CT_0 is the minimum possible observable contrast by the HVS, α is the spatial frequency decay constant, and e_2 is the half-resolution eccentricity (degrees). A foveation cut-off frequency model can be derived from (1) that gives the maximum detectable spatial frequency at any eccentricity [3]. We can therefore consider foveation as a case of non-uniform 2D sampling, where at each point, the maximum detectable spatial frequency is proportional to the sampling density (the density of the photoreceptors) at that point.

C. Selection of Fixation Point(s)

Modeling foveation in the HVS requires that the fixation point and the viewing distance be known to the encoder at the time of encoding. There are a number of approaches that can provide this information, depending on the application: (a) dynamic acquisition of the fixation point using an eye-tracking (or user controlled pointing) device [14], (b) object detection and tracking (such as face or head detection) [15], [16], [17], (c) visual objects in object-based video coding techniques such as MPEG-4 [18], (d) statistical inferences from image processing algorithms or motion analysis [11], [19] or (e) assumed fixation point trajectory such as the center of the screen. For video conferencing applications, simple face detection algorithms can provide this information. *In this work, we assume that the fixation point information is available by using one of the above methods.*

D. H.263 Encoder

The techniques presented in this paper are generic enough to be applied to any motion compensated Discrete Cosine Transform (DCT) based video coder [8]. However, since foveation particularly targets low bit rate applications, the baseline H.263 video coding specifications [20] have been used for this paper.

III. Foveation Model for Video Coding

A. Foveation Cut-off Frequency Model

The Contrast Sensitivity Function (CSF) of the HVS may be defined as $\frac{1}{CT(f,e)}$ [3], where $CT\left(f,e\right)$ is given in (1). The CSF gives the relative sensitivity of the HVS to a spatial frequency f presented at an eccentricity e. The CSF can be used to obtain the maximum detectable spatial frequency (or the cut-off frequency) function of the HVS by observing the maximum possible sensitivity of the HVS, as in [3]. We generalize this into a two-level quantization (visible/invisible) of the CSF and define the cut-off frequency function of the eye, $f_{c,e}\left(e\right)$, by requiring $\{f_{c,e}(e), e\}$ to be a contour of the CSF at $\frac{1}{CT_c}$, as given below ($CT_0 \leq CT_c \leq 1.0$):

$$f_{c,e}\left(e\right) = \frac{1}{\alpha}\frac{\ln\left(\frac{CTc}{CT_0}\right)}{1+\frac{1}{e_2}e} \quad \text{cycles per degree} \tag{2}$$

B. Foveation Model in the Discrete Domain

The foveation model given in (2) gives the cut-off frequency in cycles per degree (cpd). We need to adapt the model for incorporation into digital video processing algorithms by analyzing the digital-to-analog conversion at the output of the display devices. For a display device with a square sampling grid and a sampling interval of ε meters, the maximum reconstructible frequency is $\frac{1}{2\varepsilon}$ cycles per meter (cpm) in the horizontal and vertical directions [21]. Consider an intensity sinusoid $\sin(2\pi f x)$ with frequency f in cpm being viewed from a distance V. The instantaneous frequency in cpm of a sinusoid $\sin(\Phi(x))$ is defined as $\frac{1}{2\pi}\frac{d\Phi}{dx}$ where the angle $\Phi(x) = 2\pi f x$. The instantaneous frequency of the sinusoid in cpd (that is, with respect to eccentricity), with $x(e) = V\tan\left(\frac{\pi e}{180}\right)$, is:

$$\frac{1}{2\pi}\frac{d\Phi}{de} = \frac{1}{2\pi}\frac{d\Phi}{dx} \times \frac{dx}{de} = \frac{\pi f V}{180}\sec^2\left(\frac{\pi e}{180}\right) \quad \text{cpd} \tag{3}$$

Thus, the highest reconstructable spatial frequency (the display cut-off frequency) $f_{c,d}(e)$ of a display device that can display a maximum frequency of $\frac{1}{2\varepsilon}$ cpm is given by

$$f_{c,d}(e) = \frac{\pi V_N}{360}\sec^2\left(\frac{\pi e}{180}\right) \qquad \text{cpd} \tag{4}$$

where $V_N = \frac{V}{\varepsilon}$ represents the viewing distance measured in terms of pixels.

The foveation cut-off frequency model is the lowest spatial frequency that cannot be displayed or perceived: $\min\{f_{c,d}, f_{c,e}\}$. We normalize the model with the sampling frequency at the reconstruction to obtain the discrete spatial domain foveation model:

$$f_{c,n}(e) = \frac{\min\{f_{c,e}(e), f_{c,d}(e)\}}{f_{c,d}(e)} = \min\left\{1, \frac{f_{c,e}(e)}{f_{c,d}(e)}\right\} \tag{5}$$

To incorporate the fact that the human eye does not fixate steadily at one point but moves in a random spot corresponding to about one degree of eccentricity in diameter [14], we may use the normalized cut-off frequency function given as:

$$f'_{c,n}(e) = \begin{cases} f_{c,n}(0) & \text{if} \quad e \le 0.5 \\ f_{c,n}(e-0.5) & \text{if} \quad e > 0.5 \end{cases} \tag{6}$$

This modification assumes that the actual fixation point of the eye may lie at any point within the 0.5^o circle around the assumed fixation point, and for $e > 0.5^o$ (6) incorporates the worst case scenario when the fixation point is at the 0.5^o boundary.

By describing all coordinate measurements (as well as the viewing distance) in terms of the number of pixels, the foveation model can be made independent of the display device resolution (ε). In the rest of this paper, all distance measurements will be given in terms of pixels unless explicitly stated otherwise.

C. Approximate Foveation Model

Foveated image and video coding requires the knowledge of the cut-off frequency $f_c(x,y)$ at every point on the image. Since the viewing geometry is generally not completely defined at the time of programming an encoder, this computation has to be performed in real time at the time of encoding. The HVS typically changes the fixation point about three to five times per second [19] and the cut-off frequency needs to be evaluated that frequently. The computation of (6) is time consuming, especially for embedded fixed-point implementations, because of the computations for arctangents, square roots and divisions. In this section, we develop an efficient approximation to the discrete foveation cut-off frequency model.

The simplification of the model starts by assuming that the cut-off frequency is constant over a block of size $M \times M$. Thus, the *foveation regions* (regions having the same cut-off frequency) in the approximation would consist of unions of disjoint $M \times M$ blocks. Since the minimum unit of frame data that is encoded in standard video coders is a block of 16×16 pixels (a *macroblock*), we chose $M = 16$ in our implementations. For $M = 16$, the assumption that the cut-off frequency is constant over the block is reasonably valid (see Figure 4). This is also a good choice for embedded implementations, where each macroblock is treated as an independent unit and its transfer to and from slower off-chip memory can be optimized.

The next approximation is to allow n possible values of the viewing distance from V_1 through V_n and L possible values of the cut-off frequency, which is uniformly quantized to L values using the 'ceiling' ($\lceil\cdot\rceil$) operator. For ease of implementation (especially using the DCT), the preferred choices of the values of L are 2, 4, 8, etc. We observed that $L = 4$ is too small as it leads to visible boundaries between foveation regions. We therefore chose $L = 8$ for our implementation because we observed from our experiments that increasing the value of L beyond 8 gave an insignificant improvement in bit rate.

The above approximations can be written mathematically in (7) through (10) where the cut-off frequency for a macroblock is obtained by evaluating (7) at the center of the macroblock. The approximate model is related to the original model in (6) by (10).

$$
\begin{aligned}
f_c(x,y) &= \min\left\{\frac{i}{L} : d \leq B\left[i,V\right], 1 \leq i \leq L, i \in \mathbb{Z}^+, V \in \{V_1 ... V_n\}\right\} && (7)\\
d &= \left(x - x_f\right)^2 + \left(y - y_f\right)^2 && (8)\\
B\left[i,V\right] &= \max\left\{r^2 : \lceil f_c^*\left(r,V\right) \times L\rceil = i, r \in \mathbb{Z}^+\right\} && (9)\\
f_c^*\left(r,V\right) &= f'_{c,n}\left(\frac{180}{\pi}\arctan\left(\frac{r}{V}\right)\right) && (10)
\end{aligned}
$$

Figure 4 compares the approximate cutoff frequency model in (7) with the original model in (6). The value of the cutoff frequency in the approximate model is almost always greater than that of the original model. Hence, we are slightly over-designing the model by preserving more resolution information. Figure 5 shows the cut-off frequencies over the image due the fixation point at 'X'.

C.1 Lookup Table Implementation

Equations (9) and (10) depend only on the viewing distance V and can therefore be precomputed for n values of $V \in \{V_1, V_{2,} ... V_n\}$ and stored as an $L \times n$ lookup table $B[i,V]$ of positive integers. In this way, the model avoids time computations of artangents, divisions and square roots at run time.

C.2 Multiple Fixation Points

For representing multiple objects of interest in the visual signal with higher resolution, the foveation model needs to incorporate multiple fixation points. Multiple foveation points can be incorporated by using (11) for the calculation of the cut-off frequency due to m fixation points, where $f_{c,j}(x,y)$ is the cut-off frequency at point (x,y) corresponding to the j^{th} fixation point [4]:

$$f_c(x,y) = \max\{f_{c,j}(x,y), j = 1...m\} \tag{11}$$

In our approximation of the model in (7), if we make an assumption that all fixation points have the same viewing distance associated with them, then it suffices to compute (7) only once with $d = \min\{d_j\}$ where d_j is the result of computing (8) for the j^{th} fixation point. Thus, the cut-off frequency of a macroblock is determined by the fixation point that is nearest to it.

C.3 Computational Complexity

The approximate model described in Section III-C requires two multiplications, three additions and L comparisons per macroblock for a single fixation point. The lookup table $B[i,V]$ requires only nL words of storage.

D. Foveation Depth Parameter

We would like to have a design parameter that controls the amount of foveation according to the required output bit rate. This parameter may be fixed for one application or may be varied using a rate-control algorithm to increase or decrease the resolution fall-off rate depending on the available bandwidth. We modify (2) to be

$$f_{c,e}(e) = \frac{F}{1 + \frac{\zeta}{e_2} e} \tag{12}$$

where ζ is the foveation depth parameter and $F = \frac{1}{\alpha} \ln \left(\frac{CT_c}{CT_0} \right)$. For the approximate foveation model, the lookup table $B[i, V]$ may be replaced by a higher dimensional table $B[i, V, \zeta]$. The cut-off contrast threshold, CT_c, and the foveation depth parameter, ζ, give added flexibility in tuning the model for a particular application both with respect to the radius of the unfoveated region (CT_c) as well as the rate of resolution fall-off (ζ) in the foveated regions.

IV. Real-Time Foveation Techniques

Having developed a computationally efficient foveation model in Section III, we now explore fast foveation methods for video coding in this section. Two techniques for foveated video coding are presented: spatial domain foveation pre-processing and Discrete Cosine Transform domain foveation. Simulation results are presented in Section V.

A. *Spatial Domain Foveation Pre-processing*

Spatial domain foveation has been used previously for foveated video coding [3], [4], [5]. A pre-processing engine calculates the different foveation regions and their cut-off frequencies and then applies the corresponding lowpass filter to each region. In this paper, we present our implementation of the spatial domain foveation to put the DCT domain foveation (presented later in Section IV-B) into perspective, and to serve as a basis for the evaluation of the DCT domain foveation algorithm.

The arithmetic computational complexity of spatial domain foveation pre-processing is slightly higher than that of an N-tap, separable, 2-D FIR filter ($2N$ multiplications and $2(N-1)$ additions per pixel). Foveation pre-processing is slow in execution not only because of the large number of operations required per pixel, but also because the video frames reside in slow off-chip memory for embedded video processors.

A.1 Implementation Details

In our implementation, symmetric, 7×7, separable 2-D FIR filters with 16-bit fixed-point coefficients were employed. The filter coefficients were scaled to give unity gain at DC. At region boundaries, an average of the outputs from the two filters corresponding to the two adjacent regions was taken to reduce the visibility of the boundary. For filtering, the rows and columns were symmetrically extended at the boundaries. The filters were designed to approximate ideal lowpass filters using constrained least-squared error minimization (MATLAB command *fircls1*) with a passband and stopband ripple of 0.05.

By numerous experiments, it was observed that foveating the chrominance components in the spatial domain did not yield a reduction in bit rate significant enough to warrant the additional complexity overhead. In the case of integrating foveation into the encoder, as described later, the increase in the computational complexity is a very small fraction of the overall complexity of the encoder, and foveating the chrominance components becomes more feasible. We observed from experiments that leaving the chrominance components at uniform resolution while foveating the luminance component did not create perceptual distortion in the reconstructed frames.

A.2 Analysis

In the case of motion compensated prediction for foveated video coding, this section analyzes what happens when the prediction of a macroblock comes from a region that has a different cut-off frequency. The section also analyzes the effect of a change in the fixation point. The analysis will be for spatial domain foveation.

Figure 6(a) shows foveation in the spatial domain by pre-processing in cascade with a video encoder. A macroblock $M+E$, which is located in a region with cut-off frequency f_2, is being predicted from a macroblock M, which is located in a region with cut-off frequency f_1. E denotes the 'new information' or the prediction error in the macroblock (and M is the prediction) whereas $M(f_k)$ denotes M foveated (bandlimited) to frequency f_k. We trace the macroblock through the loop and (ignoring quantization) observe that the decoder output is $M(f_2)+E(f_2)$, which is the desired output (macroblock $M+E$ bandlimited to f_2). However, the signal that is coded for transmission is $M(f_2)+E(f_2)-M(f_1)$, which is bandlimited to $\max(f_1, f_2)$. Thus, changes in the cut-off frequency trigger an output of the encoder even in the case of purely translational motion ($E=0$), and the encoder sends the $[f_1, f_2]$ band to the decoder. In the extreme case where the fixation point changes from one frame to the other while the scene is static, the encoder constantly sends information describing the $[f_1, f_2]$ band even though no new information is present in the signal. Thus the changes in the fixation point generate information regardless of whether the prediction comes from a lower or a higher frequency region, which, in the latter case, serves only to *remove* the extra resolution information from the prediction.

B. Discrete Cosine Transform (DCT) Domain Foveation

An alternative to foveation by pre-processing is to incorporate it within the video encoder. The DCT is used in standard video coders for reducing spatial redundancies [8]. In the DCT domain, the higher frequency components can be suppressed as desired by weighting the appropriate DCT coefficients with weights close to zero. This is the basis of our DCT domain foveation technique.

B.1 Embedding Foveation in the Video Encoder

Figure 6(b) shows a video encoder with foveation integrated into it, where the foveation filtering is performed by DCT subband weighting. The figure depicts the foveation and the video coding process in the DCT domain. The quantizers are ignored and the foveation filtering with frequency f_k is denoted by multiplication with weights W_k. Note that the prediction error is being foveated and not the actual frame. For INTRA coded frames (in which no motion compensated prediction is used), $M = 0$ so that the decoder constructs EW_2, which is the required macroblock foveated at f_2. However, in the case of INTER coded frames (for which motion compensated prediction is used), there is an added term to the reconstruction, specifically $MW_1 - MW_1W_2$ (apart from the desired term $MW_2 + EW_2$). Section IV-B.2 analyzes this error in the reconstruction, where we attempt to control and eliminate any distortion resulting from it by enforcing certain constraints on the weights used for filtering. Since the distortion term is present inside the encoding loop, it will not lead to an error drift problem at the decoder.

B.2 Constraints on DCT Subband Weights

The reconstructed macroblock at the decoder is $MW_2 + EW_2 - MW_1W_2 + MW_1$ where only the first two terms are the desired terms. Therefore, we wish $W_1 - W_1W_2 = 0$, or for the 8×8 blocks within the macroblock, $W_1(k_1, k_2)(1 - W_2(k_1, k_2)) = 0 \quad \forall k_1, k_2 \in \{0...7\}$. One solution to this constraint is given below:

$$\begin{aligned} W_2(k_1, k_2), W_1(k_1, k_2) &\in \{0, 1\} \\ W_2(k_1, k_2) &= 0 \Rightarrow W_1(k_1, k_2) = 0 \end{aligned} \tag{13}$$

Since we are using these weights to suppress higher frequency subbands, we take $W(k_1, k_2) = 1$ for $0 \leq k_1, k_2 \leq k_c$ and $W(k_1, k_2) = 0$ for $k_c < k_1, k_2 \leq 7$, where k_c denotes the index of the highest subband that is not suppressed. For such lowpass weights, the solutions to the constraint equation (13) can be written as

$$\begin{aligned} W_1(k_1,k_2) &= \begin{cases} 1 & 0 \le k_1,k_2 \le k_{c1} \\ 0 & k_{c1} < k_1,k_2 \le 7 \end{cases} \\ W_2(k_1,k_2) &= \begin{cases} 1 & 0 \le k_1,k_2 \le k_{c2} \\ 0 & k_{c2} < k_1,k_2 \le 7 \end{cases} \\ k_{c2} &\ge k_{c1} \end{aligned} \tag{14}$$

The requirement $k_{c2} \ge k_{c1}$ above translates into $f_2 \ge f_1$, or that the macroblock's prediction comes from a region of lower or equal resolution. This arises from the fact that the signal encoded and transmitted to the decoder is bandlimited to f_2. In general, the predictions for macroblocks can come from higher resolution regions as well. If only the definitions of W_1 and W_2 in (14) hold but $k_{c2} < k_{c1}$, we observe that $W_1 + W_2 - W_1W_2 = W_1$, and the reconstructed output is $MW_1 + EW_2$. That is, with the constraint $k_{c2} \ge k_{c1}$ relaxed, the decoder reconstructs the desired signal (the prediction and the error bandlimited to f_2) if $f_2 \ge f_1$. Otherwise, the reconstruction is the sum of the prediction signal bandlimited to f_1 and the prediction error signal bandlimited to f_2. We are allowing the prediction to be reconstructed at higher resolution.

In contrast to spatial domain foveation preprocessing, DCT domain foveation does not always encode the $[f_1, f_2]$ band. In the case that the macroblock is moving from a higher resolution region to a lower resolution region ($f_2 < f_1$), the encoder only transmits the prediction error signal bandlimited to f_2. This follows since the coefficients of the prediction macroblock ($W_2 - W_1W_2$) are zero. Thus, for purely translational video ($E = 0$), this foveation method generates information during a change in the fixation point *only* for those macroblocks that are being predicted from lower resolution regions.

B.3 Designing DCT Subband Weights

In the previous section, we observed that using rectangular weights for DCT subband weighting leads to the desired output. From simulations, however, we observed that using the rectangular weights in (14) promotes blocking artifacts in regions where the cut-off frequency is small. This is because in these regions, the approximation to foveation filtering by subband weighting becomes crude. We would therefore like to use weights that have smoother transitions from unity to zero. Design of smoother DCT weights from lowpass filters has been explored in [22]. However we would like to keep the benefits of the weights designed in [22] but still reduce complexity by eliminating the need to perform multiplications. Here we present multiplierless triangular-transition weights given in (15). We observed from experiments that the performance in terms of blocking artifacts, subjective quality and output rate of these weights are as good as the weights designed from the filters in [22]:

$$w(k) = \begin{cases} 1 & 0 \le k \le k_c \\ 0.5 & k = k_c + 1 \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$W(k_1, k_2) = w(k_1)w(k_2)$$

B.4 Implementation Issues

DCT domain filtering using subband weighting is extremely efficient in terms of implementation complexity. It requires only one multiplication per DCT coefficient, which can be performed efficiently by using the fast DCT algorithm in [23]. This algorithm has a scaling operation in the final stage of the computation. This final scaling stage can incorporate the foveation weights. Thus, in the implementations of the baseline H.263 specification, for example [24], that use [23] for computation of the DCT, foveation by DCT subband weighting will come at no extra computational cost.

If the DCT weights in the implementation are rectangular or triangular-transition, the implementation can be *sped up* by terminating the DCT computation at the appropriate coefficient. The multiplication by 0.5 in (15) can be implemented by scaled DCT comptuation or by shifting in the quantization routine. In this way, the DCT domain foveation can give bit rate as well as complexity reductions.

V. Results

In this section, we present simulation results obtained by applying the algorithms described in this paper. We evaluate computational efficiency, reduction in bit rate, blocking artifacts and subjective visual quality of the proposed foveated video compression algorithms.

A. Simulation Details

Table I gives the values of the various parameters used for the simulations. The H.263 encoder written by the University of British Columbia [25] was used in the baseline mode. For accurate profiling, the encoder was run under Linux (kernel version 2.2.13) on a Pentium-II 450MHz machine with 128 MB RAM running with a single user load, after being compiled with the GNU C compiler using level-three compiler optimization (-o3 flag). The DCT domain foveation algorithm was also tested on the Texas Instruments' TMS320DSC21 programmable digital signal processing solution for digital still cameras [26]. The DSC21 architecture is capable of doing real-time 320x240 H.263 video encoding as well. Foveated video coding was implemented on the DSC21 with live video capture.

We present the results of applying our algorithms on two video sequences, 'News' and 'Mobile' of sixty frames each, which represent the low-motion, low-detail case as well as the high-motion, high-detail case (Figure 8). For the 'News' sequence, we present three configurations of fixation geometry: single static fixation point at the center of the left face, two static fixation points at the center of the two faces, and a single dynamic fixation point that moves from the left face to the right face in the vicinity of the 30^{th} frame. For the 'Mobile' sequence, we present results for two configurations: single static fixation point at the center of the ball in the first frame and a single dynamic fixation point that moves from the center of the ball to the '1' in the calendar around the vicinity of the 30^{th} frame.

B. Bit Rate Reduction with Foveated Video Coding

Table II shows the file sizes of uniform resolution 'News' and 'Mobile' sequences compressed using the baseline H.263. Only the first frame is intra-coded. Tables III and IV show the file sizes from various foveation algorithms expressed *as percentages of the uniformly coded versions* of the corresponding sequences. Figure 7 shows the case where the output bit rate is plotted versus the frame number.

C. Quality Assessment of Foveation Algorithms

Quality assessment of foveated images or video with respect to the original signal or different foveated versions of the same signal is a tricky problem. There are uniform-resolution image quality assessment techniques used routinely in the literature, such as the Mean Squared Error (MSE) or the Peak Signal-to-Noise Ratio (PSNR). However, they will not provide a fair comparison of the quality of a foveated compressed image with that of a uniform-resolution compressed version. This is because PSNR and MSE are global uniform-resolution metrics that measure fidelity, in a mathematical sense, uniformly all over the image and do not incorporate foveation modeling. Foveating an image deliberately introduces large distortions in regions that are assumed to be in the peripheral vision. MSE or PSNR will report large distortions in these regions, which may be perceptually insignificant under the assumed fixation geometry.

Foveated image quality assessment metrics have also been proposed in the literature, such as the Foveated PSNR (FPSNR) [4], [27] and the Foveated Wavelet Quality Index (FWQI) [28] but they assume that the reference and the distorted images are foveated. These metrics will be unfair to uniform resolution images. Also, these metrics assume that the foveated distorted and reference images come from the same foveation model and implementation, and hence will not provide a fair evaluation of DCT domain foveation against spatial domain foveation.

If O_u were the original uniform-resolution image, D_u were the uniform-resolution distorted (or compressed) image, and D_f were the foveated (and distorted) image, then a comparison of D_u against O_u using MSE or PSNR would be fair, but that of D_f against O_u or D_u would be unfair. If we assume, however, that we have a reference foveated version of O_u (which we denote by O_f), then a comparison of D_f with O_f with foveated quality assessment metrics is fair, but that of D_u against O_f is unfair. Also, a comparison of FPSNR(O_f,D_f) and PSNR(O_u,D_u) will be meaningless due to the lack of correspondence between the FPSNR and the PSNR scales.

Theoretically, O_u and O_f are perceptually equivalent *by construction*, assuming validity of the foveation model (and implementation) under the assumed fixation geometry and model parameters. However, one acceptable method to judge the loss of quality during foveation could be subjective measurements by human observers who evaluate the quality by viewing the images with a viewing geometry that is consistent with the model parameters chosen for foveation. For subjective quality assessment, the image outputs from different algorithms are presented in Figures 8-10.

However, since the prime focus of this paper is on real-time algorithms for foveated video coding, especially the DCT domain foveation algorithm, we can use the blocking artifact measurement algorithm described in [29] to assess the relative qualities of the outputs. This is reasonable since we know that the DCT domain foveation algorithm promotes blocking artifacts in regions with small cut-off frequency, and any increase in the blocking measure could be attributed solely to the foveation algorithm. However, it should be kept in mind that the blocking effect measurements represent a global average; the blocking effect close to the fixation region is similar for foveated and the uniform-resolution cases. Tables V, VI and VII show the measured blocking artifacts for different cases.

D. Computational Complexity

Table VIII gives the computational complexity of the foveation algorithms for the PC implementation. For embedded implementation on the DSC21, the computational overhead of the DCT domain foveation algorithm using the weights in (15) is given in Table IX. However, for the DSC21 implementation, the computation of DCT was *not* optimized (as explained in Sec. IV-B.4) and hence the foveated encoder runs slightly slower than the uniform resolution encoder. The size of the lookup table, $B[i, V, \zeta]$, was 8 words (using only $\zeta = 1.6$ and $V = 1500$ pixels).

The spatial domain foveation algorithm is computationally too complex to run in real-time on the DSC21, considering that it needs to run in tandem with an H.263 encoder, and hence was not tested on the DSC21.

E. Remarks

E.1 Output Bit Rate

Significant bit rate reductions can be achieved by foveation for video sequences that have high motion and high spectral information, such as detail and strong edges (Tables III and IV). Spatial domain foveation produces lower bit rates than DCT domain foveation.

Additional fixation points increase the bit rate because a larger region in the image is now being represented with higher resolution. The savings in the bit rate are relatively constant throughout the video sequence if the fixation points do not change. Changing the fixation point increases the output bit rate for as long as the fixation point is moving (Figure 7).

E.2 Output Quality

Blocking artifacts are significantly larger for DCT domain foveation as compared with spatial domain foveation (Tables VI and VII) despite the fact that the latter technique generates fewer bits. Multiple fixation points increase the bit rate but the subjective quality of the reconstruction is much better. The subjective quality of the reconstruction, in regions closer to the fixation point is better for DCT domain foveation than for spatial domain foveation. The reason for higher visual quality close to the fixation point is that DCT domain filtering by coefficient weighting does not suppress the high frequencies as efficiently as the filters in the spatial domain method. For macroblocks farther away from the fixation point, blocking artifacts become more prominent and the subjective quality of spatial domain foveation becomes superior in the peripheral regions, as shown in Figure 9.

For sequences with low activity, moving the fixation point in the DCT domain foveation case preserves the background at the highest resolution at which the decoder had previously reconstructed it. For spatial domain foveation, the reconstruction has a lower resolution in the static background even if the decoder already had the resolution information in the previous frames, as shown in Figure 10.

E.3 Computational Complexity

Calculation of the foveation model is very efficient. Spatial domain foveation is 20 times more computationally expensive than DCT domain foveation with triangular-transition weights. DCT domain foveation using triangular-transition weights actually *speeds up* the encoder. For embedded implementations, the DCT domain foveation algorithm is very efficient in terms of computational overhead as well as memory overhead (Tables VIII and IX).

VI. Conclusions

We have developed and demonstrated techniques for standard-compliant real-time implementation of foveated video coding, with our tests focusing on the baseline mode of the H.263 standard. We have presented a computationally efficient foveation model, as well as techniques for foveation in the spatial and DCT domains. We have analyzed the techniques in terms of their effect on the reconstruction, output bit rate, blocking artifacts and subjective quality. Foveated video coding has proven advantages at reducing the bit rate. Its implementation in software and in embedded systems can be efficient by embedding foveation inside the video coding loop. In case of the DCT domain foveation algorithm, we obtain reduction in complexity as well as bit rate. Foveation preserves the semantics of the bit stream and requires no modification of the decoder.

References

[1] K. Sayood, *Introduction to Data Compression.* Morgan Kaufmann Publishers Inc., 1996.

[2] N. Jayant, J. Johnston, and R. Safranek, “Signal compression based on models of human perception,” *Proc. IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.

[3] W. S. Geisler and J. S. Perry, “A real-time foveated multiresolution system for low-bandwidth video communication,” in *SPIE proceedings*, vol. 3299, 1998.

[4] S. Lee, *Foveated Video Compression and Visual Communications over Wireless and Wireline Networks.* PhD thesis, Dept. of Electrical and Computer Engineering, The

University of Texas at Austin, Austin, TX 78731, May 2000.

[5] S. Lee, M. S. Pattichis, and A. C. Bovik, "Foveated video compression with optimal rate control," *IEEE Trans. Image Processing*, vol. 10, pp. 972–992, July 2001.

[6] A. Basu and K. J. Wiebe, "Enhancing videoconferencing using spatially varying sensing," *IEEE Trans. on Sys., Man and Cybernetics – Part A: Systems and Humans*, vol. 28, pp. 137–148, Mar. 1998.

[7] W. S. Geisler and J. S. Perry, "Variable resolution displays for visual communications and simulation," *Soc. for Information Display Tech. Digest*, vol. 30, pp. 420–423, 1999.

[8] M. Ghanbari, *Video coding: An Introduction to Standard Codecs.* The Inst. of Electrical Engineers, 1999.

[9] B. A. Wandell, *Foundations of Vision.* Sunderland, MA: Sinauer Associates, Inc., 1994.

[10] T. H. Reeves and J. A. Robinson, "Adaptive foveation of MPEG video," in *ACM Multimedia*, pp. 231–241, 1996.

[11] E. Nguyen, C. Labit, and J.-M. Odobez, "A ROI approach for hybrid image sequence coding," in *Proc. Int. Conf. on Image Proc. (ICIP-94)*, pp. 245–249, 1994.

[12] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vision Research*, vol. 21, pp. 409–418, 1981.

[13] M. S. Banks, A. B. Sekuler, and S. J. Anderson, "Peripheral spatial vision: limits imposed by optics, photoreceptors, and receptor pooling," *Journal of the Optical Socity of America*, vol. 8, pp. 1775–1787, Nov. 1991.

[14] "Choosing an eye tracking system." Applied Science Laboratories. [Online] http://www.a-s-l.com.

[15] H. Wang and S. F. Chang, "A highly efficient system for automatic face region detection in mpeg video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, pp. 615–

628, Aug. 1997.

[16] H.-L. Eng and K.-K. Ma, "Segmentation and tracking of faces in color images," *Proc. IEEE Int. Conf. Image Proc.*, vol. 3, pp. 758–761, 2000.

[17] A. Bors and I. Pitas, "Prediction and tracking of moving objects in image sequences," *IEEE Trans. Image Processing*, vol. 8, pp. 1441–1445, Aug. 2000.

[18] C. Kim and J.-N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, pp. 122–129, Feb. 2002.

[19] C. M. Privitera and L. W. Stark, "Algorithms for defining visual regions-of-interest: Comparison with eye fixations," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 970–982, Sep. 2000.

[20] "Video coding for low bitrate communication." ITU-T Rec. H.263, Mar. 1996.

[21] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing.* Prentice Hall, 1984.

[22] H. R. Sheikh, S. Liu, B. L. Evans, and A. C. Bovik, "Real-time foveation techniques for H.263 video encoding in software," in *Proc. IEEE Int. Conf. on Accoustics, Speech and Signal Proc.*, vol. 3, pp. 1781–1784, 2001.

[23] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Trans. IEICE*, vol. 71, pp. 1095–1097, Nov. 1988.

[24] B. Erol and F. Kossentini, "Efficicient coding and mapping algorithms for software-only real-time video coding at low bit rates," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, pp. 843–856, Sep. 2000.

[25] "TMN H.263+ encoder/decoder, version 3.0." University of British Columbia, Vancouver, B.C. Canada. Signal Processing and Multimedia Group [Online] Available:

http://spmg.ece.ubc.ca.

[26] DSC Business Unit, “TMS320DSC21 — a high performance, programmable, single chip digital signal processing solution to digital still cameras.” Texas Instruments, Dallas, Texas. [Online] Available: http://www.ti.com/sc/digitalcamera.

[27] S. Lee, M. S. Pattichis, and A. C. Bovik, “Foveated video quality assessment,” *IEEE Trans. on Multimedia*, vol. 4, pp. 129–132, Mar. 2002.

[28] Z. Wang and A. C. Bovik, “Embedded foveation image coding,” *IEEE Trans. on Image Processing*, vol. 10, pp. 1397–1410, Oct. 2001.

[29] Z. Wang, A. C. Bovik, and B. L. Evans, “Blind measurement of blocking artifacts in images,” in *Proc. IEEE Int. Conf. Image Proc.*, (Vancouver, Canada), pp. 981–984, Oct. 2000.

Hamid Rahim Sheikh was born in Lahore, Pakistan, on December 24, 1974. He received his B.Sc. degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan, and his M.S. degree in Engineering from the University of Texas at Austin in May 2001, where he is currently pursuing a Ph.D. degree. His research interests include lossy signal compression techniques, human visual system modeling, embedded implementation of video codecs, and image and video quality assessment.

Brian L. Evans received his BSEECS degree from the Rose-Hulman Institute of Technology in May 1987, and his MSEE and Ph.D. degrees from the Georgia Institute of Technology in December 1988 and September 1993, respectively. From 1993 to 1996, he was a post-doctoral researcher at the University of California at Berkeley where he worked on electronic design automation for embedded systems as a member of the Ptolemy project. He is the primary architect of the *Signals and Systems Pack* for Mathematica, which has been on the market since October of 1995.

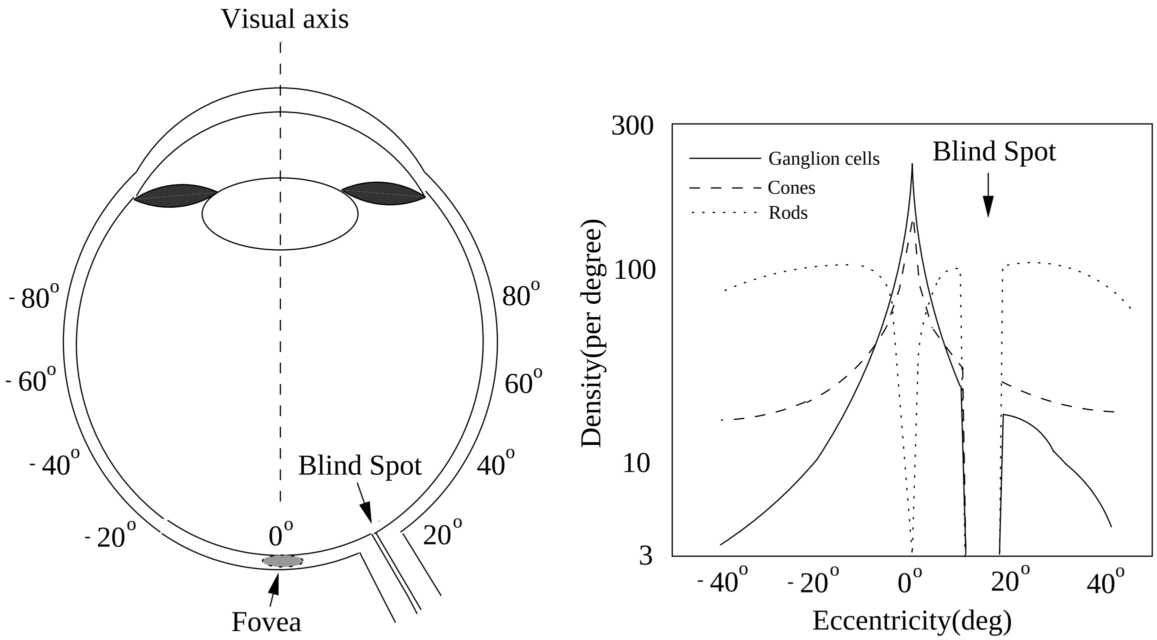
He is currently a tenured Associate Professor in the Department of Electrical and Computer Engineering at The University of Texas at Austin. He enjoys conducting research and teaching courses in embedded real-time signal and image processing systems. His current research focuses on the design and real-time implementation of ADSL/VDSL transceivers, desktop printer pipelines, video codecs, and 3-D sonar imaging systems. He has published more than 100 refereed journal and conference papers on these topics. He developed and currently teaches Multidimensional Digital Signal Processing, Embedded Software Systems, and Real-Time Digital Signal Processing Laboratory. Dr. Evans is an Associate Editor of the *IEEE Transactions on Image Processing*, a member of the Design and Implementation of Signal Processing Systems Technical Committee of the IEEE Signal Processing Society, and the recipient of a 1997 National Science Foundation CAREER Award.

Alan Conrad Bovik was born in Kirkwood, MO on June 25, 1958. He received the B.S. degree in Computer Engineering in 1980, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering in 1982 and 1984, respectively, all from the University of Illinois, Urbana-Champaign.

He is currently the Robert Parker Centennial Endowed Professor in the Department of Electrical and Computer Engineering, the Department of Computer Sciences, and the Biomedical Engineering Program at the University of Texas at Austin, where he is the Director of the Laboratory for Image and Video Engineering (LIVE) in the Center for Vision and Image Sciences. During the Spring of 1992, he held a visiting position in the Division of Applied Sciences, Harvard University, Cambridge, Massachusetts. His current research interests include digital video, image processing, computer vision, wavelets, three-dimensional microscopy, and computational aspects of biological visual perception. He has published over 300 technical articles in these areas and holds two U.S. patents. He is also the editor/author of the *Handbook of Image and Video Processing*, published by Academic Press in April of 2000.

Dr. Bovik was named Distinguished Lecturer of the IEEE Signal Processing Society in 2000, received the IEEE Signal Processing Society Meritorious Service Award in 1998, the IEEE Third Millennium Medal in 2000, the University of Texas Engineering Foundation Halliburton Award and is a two-time Honorable Mention winner of the international Pattern Recognition Society Award for Outstanding Contribution (1988 and 1993). He is a Fellow of the IEEE and has been involved in numerous professional society activities, including: Board of Governors, IEEE Signal Processing Society, 1996-1998; Editor-in-Chief, *IEEE Transactions on Image Processing*, 1996-present; Editorial Board, *The Proceedings of the IEEE*, 1998-present; Editorial Board, *Real-Time Imaging*, 2000-present; Editorial Board, *Pattern Analysis and Applications*, 1997-1998; Area Editor, *Graphical Models and Image Processing*, 1995-1998; Associate Editor, *IEEE Signal Processing Letters*, 1993-1995; Associate Editor, *IEEE Transactions on Signal Processing*, 1989-1993; Editorial Board, *Pattern Recognition*, 1988-present; Editorial Board, *Journal of Visual Communication and Image Representation*, 1992-1995; Steering Committee, *IEEE Transactions on Image Processing*, 1991-1995; and Founding General Chairman, *First IEEE International Conference on Image Processing*, held in Austin, Texas, in November, 1994.

Dr. Bovik is a registered Professional Engineer in the State of Texas and is a frequent consultant to industry and academic institutions.



(a) Human Eye

(b) Cell Density

Fig. 1. (a) Cross-section of the human eye showing the fovea. Eccentricity is the angle made with the visual axis. (b) Relative density of sensor cells in the human retina as a function of eccentricity. The Blind Spot is ignored in the modelling since the field-of-view of one eye covers the blind spot of the other.

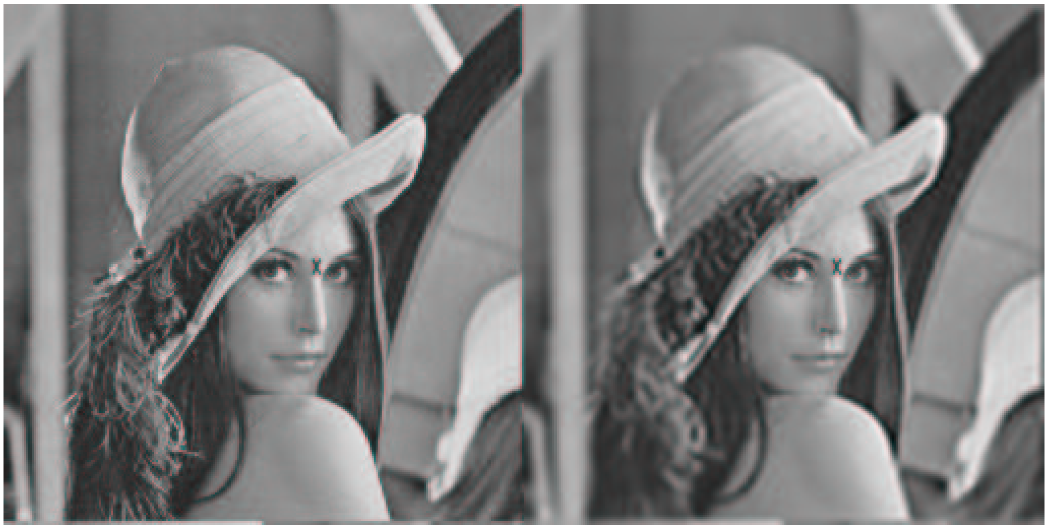


Fig. 2. Uniform resolution (left) and foveated image (right). 'X' marks the fixation point

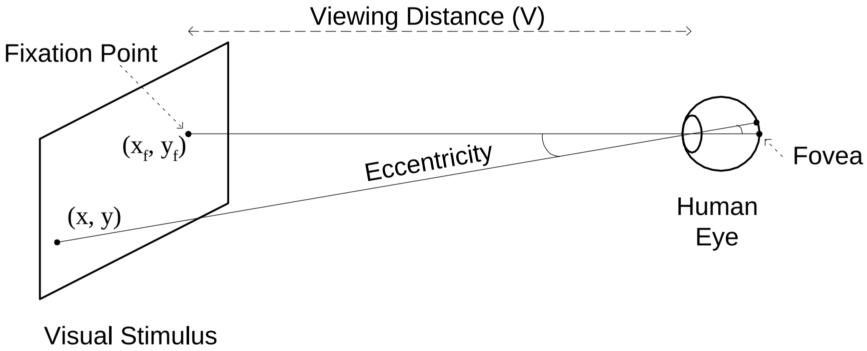


Fig. 3. Viewing Geometry: a human observer observing an image fixates at (x_f, y_f). Eccentricity is the angle indicating where the image of point (x, y) falls on the retina.

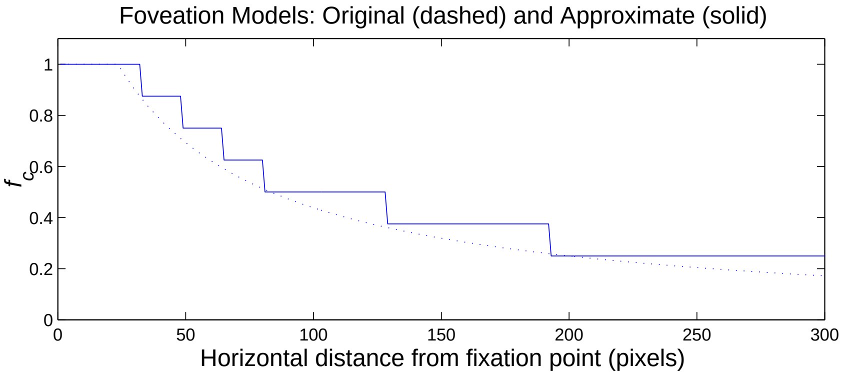


Fig. 4. Comparison of the original model with the approximate model. Parameter values are: $M = 16$, $L = 8$, $V_N = 1200$ pixels, $\frac{CT_c}{CT_o} = 4$, $\alpha = 0.106$, $e_2 = 2.3^o$.

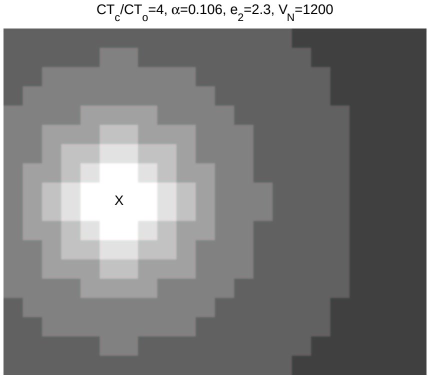
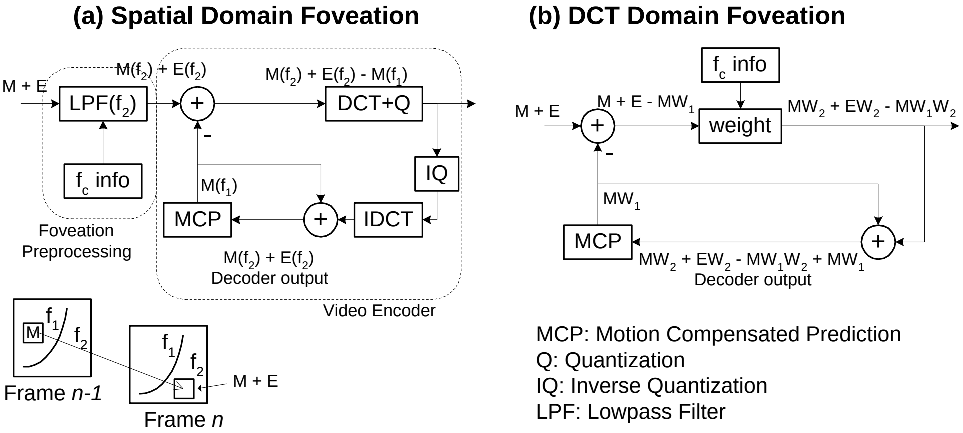


Fig. 5. Grayscale map of f_c for fixation point 'X' for a CIF resolution (352×288) image . The gray level represents the value of cut-off frequency, where 'white' denotes $f_c = 1$ and black denotes $f_c = \frac{1}{8}$ (minimum value of f_c in this figure is $\frac{2}{8}$).



MCP: Motion Compensated Prediction
Q: Quantization
IQ: Inverse Quantization
LPF: Lowpass Filter

Fig. 6. Foveated Video Coding. (a) Spatial domain foveation preprocessing and (b) DCT domain foveation by weighting in the DCT domain (the Q/IQ blocks have been ignored).

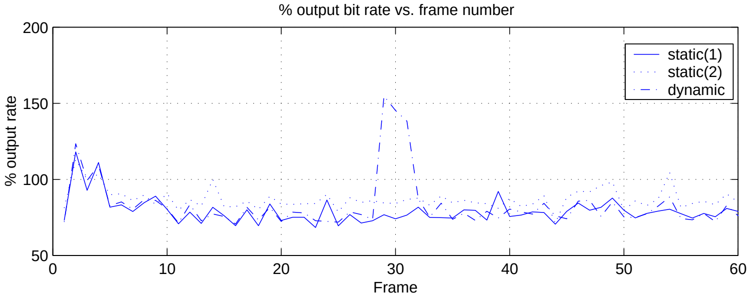


Fig. 7. Percentage output bit rate for 'News' sequence vs. frame rate for $\zeta = 1.6$ with DCT domain foveation using triangular-transition weights. The cases presented are: single, static fixation point (static (1)), two static fixation points (static (2)) and a moving fixation point (dynamic).

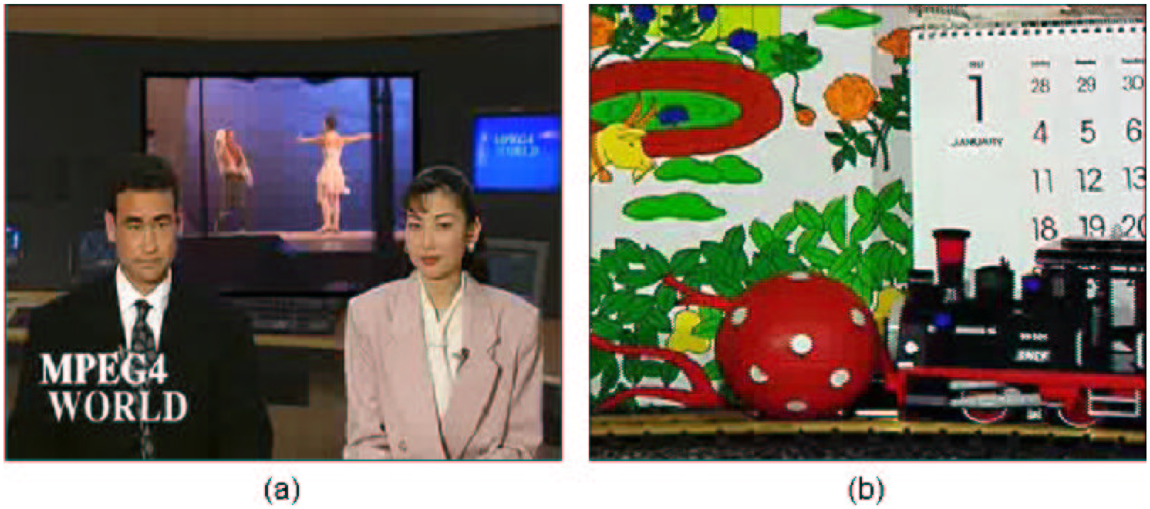


Fig. 8. Reconstructed first frames from uniform resolution H.263 coded bit streams. (a) ‘News’ and (b) ‘Mobile’.

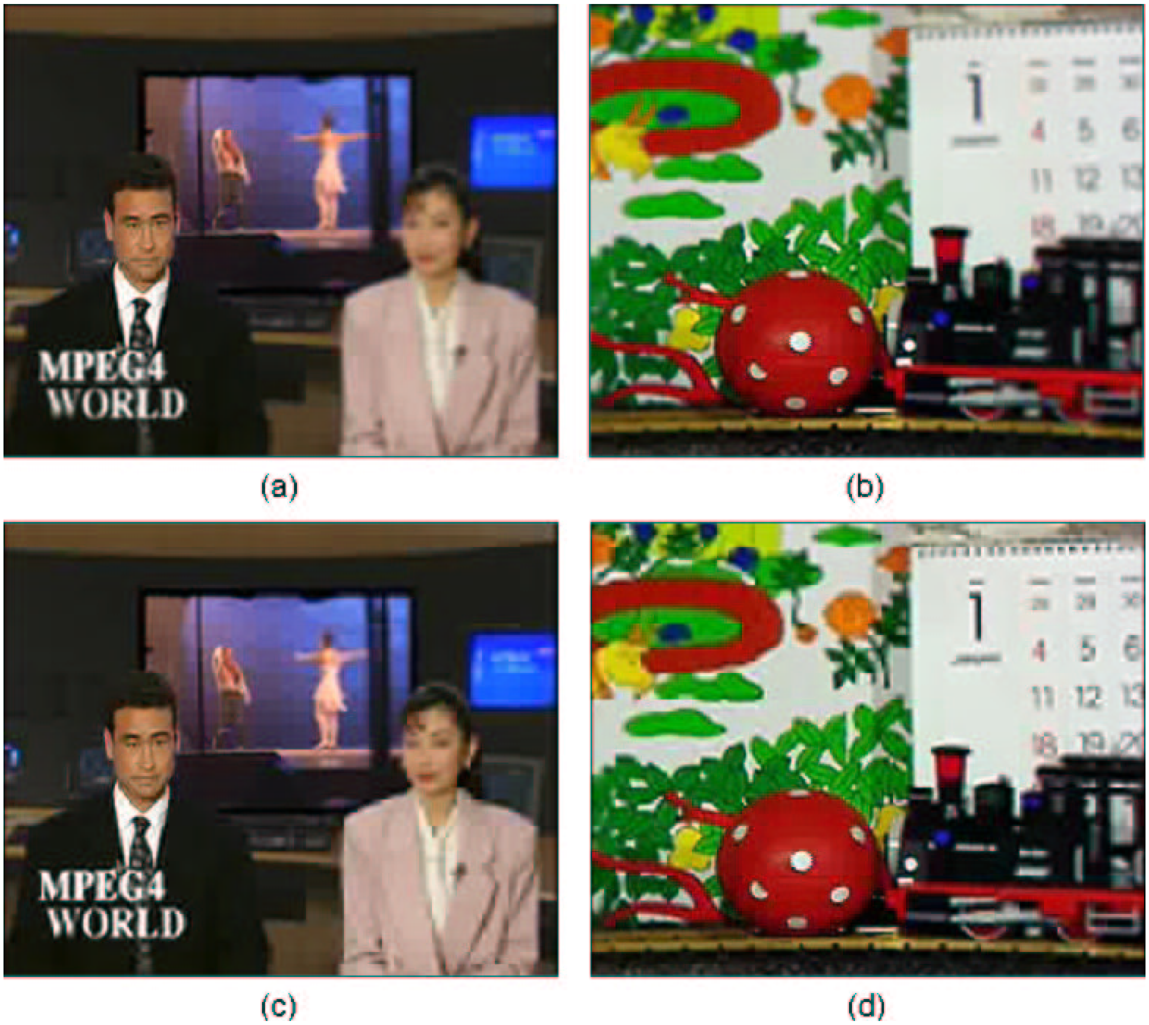

(a) (b) (c) (d)

Fig. 9. Reconstructed first frames from foveated H.263 bit streams. Top row: spatial domain foveation. Bottom row: DCT domain foveation. (Foveation depth $\zeta = 1.6$). Fixation points: center of the face for the left newscaster ((a) and (c)), and center of the ball ((b) and (d)).

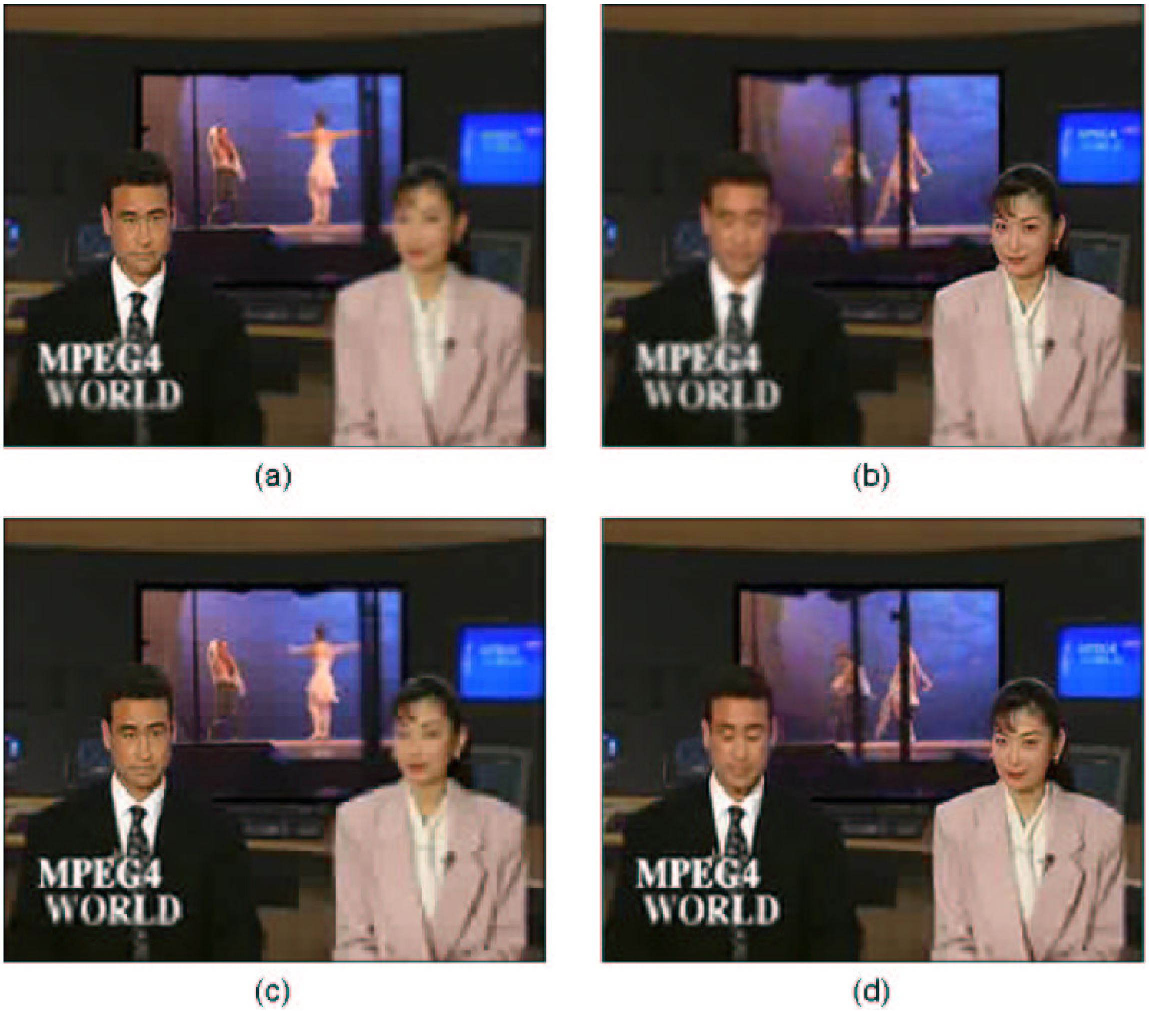


Fig. 10. The fixation point moves from the left face to the right face in the vicinity of the 30^{th} frame. Top row: spatial domain foveation. Bottom row: DCT domain foveation. Left column: reconstructed first frame. Right column: reconstructed 60^{th} frame. Note that the static background for DCT domain foveation continues to be reconstructed at higher resolution after the fixation point has moved, which is not the case for spatial domain foveation.

Parameter	*Notation*	*Value*
Cut-off contrast threshold	CT_c	16
Foveation depth parameter	ζ	$\{1, 1.6\}$
Viewing distance	V	1500 pixels
H.263 quantization parameter	Q_P	13

TABLE I

Parameter values used for simulations.

Sequence	Frames	Filesize
News	60	30.7 kBytes
Mobile	60	306 kBytes

TABLE II

Baseline H.263 encoded file sizes for uniform resolution coding at a frame size of 352×288.

	Spatial	DCT Smooth [22]	DCT Triang. Trans.
News - static	79 %	86 %	84 %
News - two points	92 %	96 %	92 %
News - moving	87 %	90 %	87 %
Mobile - static	38 %	49 %	48 %
Mobile - moving	40 %	50 %	48 %

TABLE III

Output rate for foveated video coding with baseline H.263 as a percentage of uniform resolution sequences (foveation depth $\zeta = 1.0$.)

	Spatial	DCT Smooth [22]	DCT Rectangular	DCT Triang. Trans.
News - static	70 %	79 %	79 %	77 %
News - two points	81 %	87 %	85 %	85 %
News - moving	76 %	83 %	82 %	80 %
Mobile - static	31 %	43 %	40 %	41 %
Mobile - moving	33 %	43 %	40 %	41 %

TABLE IV

Output rate for foveated video coding with baseline H.263 as a percentage of uniform resolution sequences (foveation depth $\zeta = 1.6$.)

Sequence	Blocking
News	0.4
Mobile	0.1

TABLE V

Blocking artifact measure for baseline H.263 encoded sequences for uniform-resolution coding.

	Spatial	DCT Smooth [22]	DCT Triang.
News - static	0.5	1.6	1.7
News - two points	0.6	1.2	1.2
News - dynamic	0.5	1.7	1.7
Mobile - static	0.4	1.4	1.3
Mobile - dynamic	0.3	1.4	1.3

TABLE VI

Blocking artifact measure for baseline H.263 coded sequences with foveation depth $\zeta = 1.0$.

	Spatial	DCT Smooth [22]	DCT Rectangular	DCT Triang.
News - static	0.6	2.1	2.6	2.2
News - two points	0.7	1.8	2.4	1.7
News - dynamic	0.5	2.3	2.7	2.3
Mobile - static	0.3	1.8	2.2	1.7
Mobile - dynamic	0.2	1.8	2.1	1.6

TABLE VII

Blocking artifact measure for baseline H.263 coded sequences with foveation depth $\zeta = 1.6$.

	Time	Overhead
f_c calcuation	2.8 μs	0.09 %
Spatial domain foveation	46 ms	26.90 %
DCT domain foveation	2.7 ms	1.48 %
DCT+foveation (Triangular weights)	19 ms	-9.47 %
DCT ONLY (for comparison)	37.7 ms	–

TABLE VIII

Complexity of foveation algorithms, in processing time per frame, measured in seconds and the relative overhead over the uniform-resolution baseline H.263 encoder (which takes 180 ms per frame).

	Overhead
f_c calculation overhead	0.5 %
DCT foveation routine overhead	3.0 %
Memory overhead (program & data)	165 words

TABLE IX

Foveation complexity on TMS320DSC21 expressed as percentage overhead over the uniform-resolution coder complexity for encoding a 320x240 video using a baseline H.263 video codec.