REAL-TIME FOVEATION TECHNIQUES FOR H.263 VIDEO ENCODING IN SOFTWARE

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

Laboratory for Image and Video Engineering, Department of Electrical and Computer Engineering,
The University of Texas at Austin, Austin, TX 78712-1084 USA
{sheikh, sliu2, bevans, bovik}@ece.utexas.edu

ABSTRACT

Video coding techniques employ characteristics of the Human Visual System (HVS) to achieve high coding efficiency. Lee and Bovik have exploited foveation, which is a non-uniform resolution representation of an image reflecting the sampling in the retina, for low bit-rate video coding. In this paper, we develop a fast approximation of the foveation model and demonstrate real-time foveation techniques in the spatial domain and Discrete Cosine Transform (DCT) domain. We incorporate fast DCT domain foveation into the baseline H.263 video encoding standard. We show that DCT-domain foveation requires much lower computational overhead but generates higher bit rates than spatial domain foveation. Our techniques do not require any modifications of the decoder.

1. INTRODUCTION

Data compression algorithms rely on modeling the source as well as the receiver in order to transmit information with a reduced number of bits. Lossy compression typically uses a receiver model to discard information that is unimportant to the receiver. The quantization matrices in JPEG, for example, make use of the fact that the sensitivity of the HVS is different for different spatial frequencies. In general, the more accurately the receiver is modeled, the less information needs to be sent to it.

The human eye is a very complex receiver. Lossy image and video compression standards make use of some aspects of HVS modeling. An additional layer of HVS modeling, foveation, exploits the fact that the neurons in the retina of the human eye are non-uniformly spaced with a density that decreases rapidly with the distance from the center (or fovea) of the retina. The density is highest at the fovea [2].

When an image is projected onto the retina, the HVS perceives the maximum resolution information of the image being viewed at the region whose projection falls onto the fovea. The perceived resolution at the retina quickly falls off away from the fovea. By finding the fixation point (e.g. by an eye tracker or image analysis technique [3]) and the viewing distance, we can use the foveation model to discard resolution information corresponding to image areas that are projected away from the fovea. Removing this resolution information would have little effect on the perceived image quality.

This research was supported in part by Texas Instruments, Inc., and by the Texas Advanced Technology Program.

Foveation can be modeled as non-uniform sampling of a 2-D signal. At each point on the image, the maximum detectable spatial frequency is proportional to the density of sensor neurons at the projection of that point on the retina. By locally bandlimiting the image to this maximum detectable frequency at each point at the encoder, perceptual quality should not be compromised. By eliminating higher spatial frequencies that cannot be perceived by the HVS, the amount of information that needs to be transmitted to the receiver is reduced. Foveation demonstration, source code, and filter coefficients can be found at

http://signal.ece.utexas.edu/~sheikh/foveation

2. FOVEATION MODEL

2.1. Ideal foveation model

The foveation model consists of a relation for the maximum detectable spatial frequency at a point of an image as a function of the coordinates of the fixation point (the point on the image which is under direct observation) and the viewing distance of the observer from the image. For image and video coding, any spatial frequencies greater than the maximum detectable frequency may be eliminated without compromising perceptual quality. We use the empirical model for the normalized maximum detectable frequency, f_c [4]:

$$f_c(x, y, x_f, y_f, V) = \frac{1}{1 + K \tan^{-1}\left(\frac{\sqrt{(x - x_f)^2 + (y - y_f)^2}}{V} \right)} \quad (1)$$

Here (x_f,y_f) are the coordinates of the fixation point, V is the viewing distance from the image (see Figure 1), and K = 13.75 (all distance and coordinate measurements are normalized to the physical dimensions of pixels on a viewing screen). Thus the ideal foveation of an image would consist of locally bandlimiting the image at coordinates (x,y) to $f_c(x,y)$. The computational complexity of ideal foveation is enormous. For practical implementations for video coding, faster alternatives must be considered.

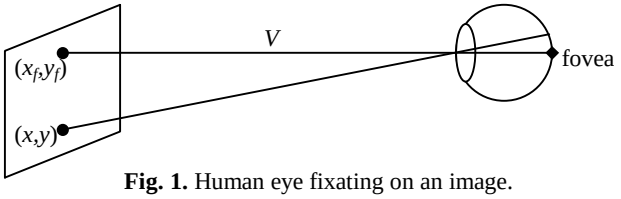

Fig. 1. Human eye fixating on an image.

2.2. Approximations to ideal foveation

The ideal foveation model described in Section 2.1 is too complex for real-time implementation. In this section, we present an approximation to the ideal foveation model to reduce the complexity of computing the maximum detectable frequency f_c for every point in a video frame. We describe the process of foveating a video sequence based on the model in Section 3.

Our approximation of the spatially varying value of f_c allows only eight possible values of the maximum detectable frequency, $f_c(x,y)$. This effectively partitions the image into a set of 'foveation' regions (maximum of eight regions) such that each region has a constant maximum detectable frequency. Our approximation further constrains each foveation region to be a union of disjoint 16 x 16 blocks (16 x 16 pixels is the size of a 'macroblock' in most video coding standards). A fast implementation would use lookup tables so as to pre-compute as much information as possible to reduce run time.

The equations for our approximation of the value of f_c' at the center coordinates (x,y) of a macroblock follow:

$$f_c'(x, y, x_f, y_f, V) = \min\left\{\frac{i}{8} : d > B[i,V], 1 \le i \le 8, i \in Z^+\right\} \quad (2)$$

$$d = (x - x_f)^2 + (y - y_f)^2 \quad (3)$$

$$B[i,V] = \min\left\{r^2 : \lceil f_c(r,V) \times 8 \rceil = i, r \in \Re^+\right\} \quad (4)$$

$$f_c(r,V) = \frac{1}{1 + K \tan^{-1}\left(\frac{r-R}{V}\right)} \quad (5)$$

Here K=13.75 as before, V is the viewing distance and belongs to the set of viewing distances for which the lookup table B has valid entries, and R is the radius of a circle centered at (x_f,y_f) that we code with full resolution. Equations (4) and (5) can be precomputed and stored in the lookup table B, thereby requiring only the computation of (2) and (3) for every macroblock at run time. Thus, the maximum detectable frequency can be computed at runtime using additions, multiplications and comparisons only. In the worst case, this model requires three additions, two multiplications and 15 fetch operations per macroblock.

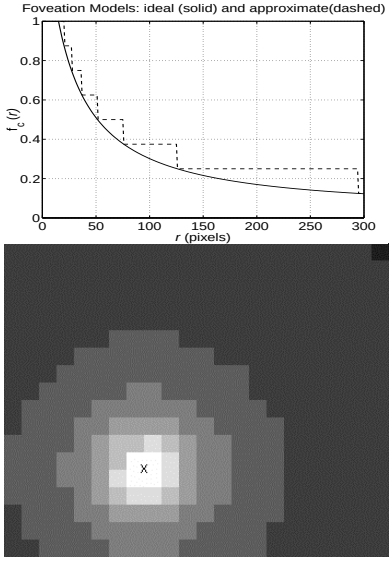

Fig. 2. Foveation models for V=500 pixels and R=15 pixels. Top: the ideal (solid) and the approximate (dashed) models as functions of distance from the fixation point. Bottom: Foveation regions where 'X' marks the fixation point.

Figure 2 shows the ideal and approximate foveation models. It also shows the foveation regions in an image corresponding to the fixation point 'X'.

2.3. Multiple fixation points

Our model can incorporate multiple observers observing multiple points of interest, or represent multiple objects of visual interest with higher resolution by using multiple fixation points. In the case of M fixation points, the maximum detectable frequency at coordinates (x,y) is:

$$f_c = \max\left\{f_{c,j}, \quad 1 \le j \le M\right\} \quad (6)$$

where $f_{c,j}$ is the maximum detectable frequency for the j^{th} fixation point.

3. REAL-TIME FOVEATED VIDEO CODING

We develop two techniques based on Section 2.2 for real-time implementation of foveation based video encoding: spatial domain preprocessing and DCT domain encoding. In our experiments, we observe that only the luminance component of the video needs to be foveated. Foveating the chrominance components give little reduction in the bit rate.

3.1. Spatial domain foveation preprocessing

Spatial domain foveation preprocessing is straightforward and it has been used in [1] as well, but with a different (and more complex) foveation model. In this approach, we preprocess each foveation region in a frame with a low-pass filter that has a cutoff frequency equal to the maximum detectable frequency f_c' for that region (computed using technique described in section 2.2). The preprocessed video may then be encoded using any video encoder. Since the preprocessing removes the high frequency information, it reduces the bit rate required to code the video.

We use seven foveation filters for the eight possible foveation regions. The region where f_c'=1 is not filtered. Each filter is a 7-tap, even-symmetric, separable 2-D FIR filter with 16-bit fixed-point coefficients. The image is symmetrically extended at the boundaries. The filters were designed using constrained least squares error minimization (Matlab command *fircls1*). The coefficients were scaled to give unity gain at DC. Foveation filtering is equivalent to filtering an image except that we switch filters each time we cross a foveation region boundary. The computational complexity of this method, in terms of multiplications and additions, is almost the same as that for separable 2-D FIR filtering of video frames.

Foveation preprocessing is independent of the video coding scheme used. This makes this method attractive where it is not possible to change or update the encoder. However, this approach is slow in execution. For embedded video processing on Digital Signal Processors (DSPs), the picture frames typically reside in off-chip memory, which is slow to access. This exacerbates the overhead for spatial domain foveation preprocessing.

3.2. DCT domain foveation

An alternative to foveation preprocessing is to incorporate the foveation filtering into the video coding loop. Standard video coding techniques (such as H.263) typically use DCT-based video coding. In such cases, the simplest way to do foveation is to weight the DCT coefficients to suppress frequencies higher than the maximum detectable frequency. Figure 3 illustrates fo-

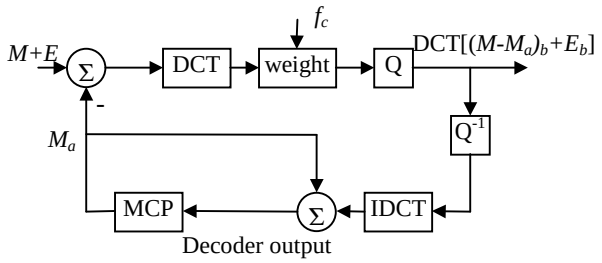

Fig. 3. DCT domain foveation in an H.263 encoding loop. Motion estimation and entropy coding are not shown.

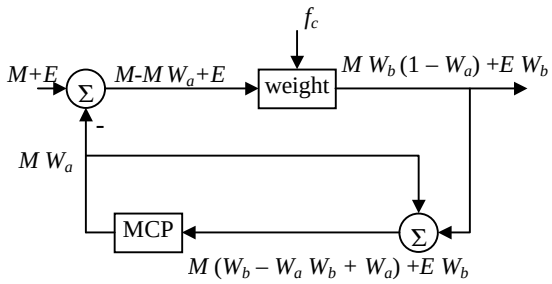

Fig. 4. DCT domain foveation in an H.263 encoding loop in the DCT domain.

veation in DCT domain embedded inside an H.263 encoding loop. A prediction macroblock M that resides in region '*a*' of the previously encoded frame is being used to predict, using motion-compensated prediction (MCP), a macroblock $M+E$ that resides in a region '*b*' in the current frame. E denotes the 'prediction error' or the 'new information' in the macroblock. 'Q' and 'Q^{-1}' denote the quantizer and the inverse-quantizer. A weighting operation corresponding to the maximum detectable frequency of region '*b*' is applied to the DCT of prediction error. We denote foveation by subscripts: M_a denotes foveation of macroblock M with the maximum detectable frequency of region '*a*' etc. We send the prediction error, bandlimited (foveated) to the maximum detectable frequency of region '*b*', to the receiver.

3.2.1 Designing DCT weights

We can design DCT weights that are better than a rectangular window. For 1-D signals, it has been shown [5] that for a length $2N$ FIR filter $h(n)$ whose Discrete Fourier Transform (DFT) $H_F[k]$ is real and even, multiplying the DCT coefficients $X[k]$ of a length N signal $x[n]$ with the first N coefficients of $H_F[k]$ is equivalent to circularly filtering the symmetrically padded signal $x[n]$ with $h[n]$ in the time domain. For video coding, an 8-point DCT is used, so we use N=8. This means that we can use 16-tap filters to derive the weights $W[n]$.

In order to make the filter $h[n]$ consistent with the definitions in [5], we can start with a length $2N$-1 even symmetric filter $h_l[n]$ (e.g. using the *fircls1* command in Matlab). By symmetry, the DFT of length $2N$ filter $h_l[n$-1] is real and even. A circular shift of N will then give the required $h[n]$ that is consistent with the definitions in [5]. We can summarize the design as follows:

$$\begin{aligned} h[n] &= h_l[n+N-1] \quad 0 \le n < N \\ h[n] &= 0 \qquad\qquad\quad n = N \\ h[n] &= h_l[n-N-1] \quad N+1 \le n < 2N \end{aligned} \tag{7}$$

3.2.2 Analysis of DCT domain foveation

The first thing to note about DCT domain foveation (Figure 3) is that for predicted pictures (or P-pictures) we are foveating the prediction error and not the actual frame. The decoder adds the prediction from the previous frame to the prediction error to reconstruct the current frame. In Figure 3, the decoder receives $(M-M_a)_b+E_b$ where as it should ideally receive $M_b+E_b-M_a$ so that the reconstructed macroblock is M_b+E_b. We therefore need to evaluate the effects of prediction error filtering on the reconstruction at the decoder. To facilitate analysis, we analyze the encoder in the DCT domain assuming that quantization is fine enough for equation (8) to hold. We may then represent the encoding loop in the DCT domain without the quantizers as shown in Figure 4.

$$DCT[M] \approx Q^{-1}[Q[DCT[M]]] \tag{8}$$

In Figure 4, we explicitly write the weight factors with the macroblocks, W_a denoting the weight matrix corresponding to region '*a*' etc. We also assume that the previous frame was coded as an I-frame so that the actual picture frame is foveated. Thus the output of the predictor is $M\,W_a$.

For proper reconstruction, the encoder should transmit $M\,W_b + E\,W_b - M\,W_a$ to the decoder which could then reconstruct $M\,W_b + E\,W_b$ (the original macroblock $M+E$ foveated to the maximum detectable frequency of region '*b*') by adding the prediction $M\,W_a$. But this is not so for our case, where the decoder receives $M\,W_b - M\,W_a\,W_b + E\,W_b$ instead. Thus, we impose a condition on the DCT weights:

$$\begin{aligned} W_b - W_a W_b + W_a &= W_b \quad f_{c,a} \le f_{c,b} \\ W_b - W_a W_b + W_a &= W_a \quad f_{c,a} \ge f_{c,b} \end{aligned} \tag{9}$$

where $f_{c,x}$ denotes the maximum detectable frequency for region '*x*'.

The ideal constraints are exactly satisfied by the rectangular window only. However using a rectangular window gives rise to large ripples at strong edges in the reconstructed frame, which degrades the prediction for future frames. For weights designed by the method in Section 3.2.1, we observe that (9) was satisfied within an error of a few percent. However, when $W_a=W_b=W$, the resulting reconstruction is $M(2W–W^2)+EW$. The weights $2W-W^2$ are still lowpass but have a bandwidth that is slightly larger than f_c. The subjective quality of the foveated video using these weights is thus better than that with rectangular window.

To conclude, DCT foveation of prediction error using correctly designed weights does not lead to error drift problem in subsequently coded prediction frames.

3.2.3 Implementation issues

DCT domain foveation has a disadvantage in that it requires modification of the video encoder. However, it is significantly faster to implement than spatial domain foveation preprocessing. Specifically, DCT domain foveation requires one multiplication per pixel of additional overhead. This can be compared with $2N$ multiplications and $2N$ additions for the case of spatial domain foveation filtering (N is the length of the FIR filter).

What is even more significant is that the weighting may be incorporated into the computation of the DCT [6]. Using the scaled DCT computation given in [6], weighting can be done with zero arithmetic overhead, with DCT and foveation costing 144 multiplications and 464 additions per 8x8 block. Thus, in fast optimizations of H.263, for example in [7], that use the techniques in [6] for computing DCT, DCT domain foveation will come at no extra arithmetic computational cost.

4. RESULTS

We report results for on the 'news' and 'mobile' sequences with a single fixation point using the H.263 encoder by [8]. The 'news' sequence has low motion content and smooth background, whereas the 'mobile' sequence has considerable motion

as well as strong edges. The results were computed using a 167 MHz Sun UltraSparc-I workstation without any packed arithmetic optimizations. The results do not account for any speedup by simultaneous computation of DCT and foveation.

Table 1 shows the implementation complexity of the techniques presented as frames processed per second (fps). First, computation of maximum detectable frequency (foveation setup) is very fast. Second, DCT domain foveation is about 15 times less computationally intensive than spatial domain foveation despite the fact that we did not combine computation of DCT with foveation. However, the output bit-rate is 15-30% larger (for high motion sequences, the subjective quality of the reconstruction is better as shown in Figure 5). The reason for this is the lack of inter-block filtering. Even so, we observe (Table 1) that foveation can reduce the bit-rate significantly especially for sequences with high motion and detail without significantly degrading quality. For the 'mobile' sequence, the required bit-rate is reduced by a factor of 2.3. For smooth, low-motion 'news' sequence, the reduction in bit-rate is about 20%. Figure 5 shows reconstructed frames for the different foveation techniques for the 'mobile' sequence. The bitstreams were reconstructed using a standard H.263 decoder.

5. CONCLUSIONS

We have demonstrated two fast foveation techniques for baseline H.263 compliant foveated video coding. We have shown that DCT domain foveation by weighting has a very low computational cost and that it can be combined with the computation of the DCT. DCT domain foveation requires more bits than its spatial domain counterpart and modification of the encoder. Spatial domain foveation preprocessing does not require any modification of the encoder. Both techniques result in standard compliant bit streams, so they do not require any modification of the decoder.

	Spatial domain	DCT domain
Foveation setup	10000 fps	10000 fps
Foveation complexity	13 fps	200 fps
'news' (unfov=30.7 kB) 'mobile' (unfov=306 kB)	21.5 kB 95.6 kB	24.5 kB 133 kB
Subjective quality		'Blocking' at low bitrates

Table 1. Results for 60 frames of color CIF (352 x 288) sequences for single fixation point.

6. REFERENCES

[1] S. Lee, *Foveated video compression and visual communications over wireless and wireline networks*, *Ph.D. Dissertation*, The Univ. of Texas at Austin, May 2000.

[2] B. A. Wandell, *Foundations of Vision*, Sinauer Associates Inc., 1995.

[3] C. M. Privitera and L. W. Stark, "Algorithms for Defining Visual Regions-of-Interests: Comparisons with Eye Fixations", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 970-982, Sep. 2000.

[4] W. S. Geisler, and J. S. Perry, "A real-time foveated multiresolution system for low-bandwidth video communication", *Proc. SPIE*, vol. 3299, pp. 294-305, Jul. 1998.

[5] B. Chitprasert and K. R. Rao, "Discrete cosine transform filtering", *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, vol. 3, pp. 1281-1284, pp. 1281-1284, Apr. 1990.

[6] Y. Arai, T. Agui and M. Nakajima, "A fast DCT-SQ scheme for images", *Trans. IEICE*, vol. 71, no. 11, pp. 1095-1097, Nov. 1988.

[7] B. Erol and F. Kossentini, "Efficient 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, no. 6, pp. 843-856, Sep. 2000.

[8] Signal Processing and Multimedia Group, (1997, Sept.) *TMN H.263+ encoder/decoder, version 3.0*, Univ. of British Columbia, Vancouver, B. C., Canada. [Online] Available: http://spmg.ece.ubc.ca

Fig. 5. Tenth reconstruction frame at the output of the H.263 decoder. Top to bottom: no foveation (uniform resolution) at the encoder, spatial domain foveation and DCT domain foveation. Fixation point is the center of the ball.