# Motion Estimation and Compensation of H.263 Video via Foveation

Serene Banerjee

Multidimensional Digital Signal Processing Final Report

Embedded Signal Processing Laboratory

Dept. of Electrical and Computer Engineering

The University of Texas at Austin, Austin, TX 78712-1084 USA

serene@ece.utexas.edu

**Abstract**

The human visual system (HVS) samples the external world with non-uniform resolution. Visual acuity falls by half at 2.3 degrees away from the point of fixation. Utilizing this property in video compression, more bits are allocated around the point of fixation, i.e. the foveation point, to produce low bit rate foveated video. Depending on the video sequence, foveation used as a preprocessing step, gives a bit rate reduction of 30–70%. DCT domain foveation, however, gives 30–50% reduction in bit rate. In this paper, motion vector foveation is used in the video encoding loop, with both the above methods to get an additional 2% reduction in bit rate for an H.263 video encoder. Temporal foveation gives 3% bit rate reduction. All these methods produce standard compliant bit streams and require no modification of the decoder.

## I. Introduction

The two factors that limit the use of real-time video communications are network bandwidth and processing resources. The ITU-T H.263 standard [1], [2], [3], [4] for video communication over wireless and wireline networks has high computational complexity. In the H.263 encoder, the most computationally complex operation is motion estimation, even when using an efficient diamond search [5]. Utilizing the non-uniform resolution property of the HVS, my objective is to perform motion estimation non-uniformly [6], [7] to reduce computational complexity and yield better rate vs. quality tradeoffs.

To reduce spatial redundancies in a video sequence, foveation can be used as a preprocessing step [8], [9] or in the Discrete Cosine Transform (DCT) domain [10], [11], [12] to give 30–70% bit rate reduction. Motion vector foveation [6], [7], [13], used with both these methods, give an extra 2% bit rate reduction. Thus, with motion vector foveation, a bit rate closer to the target is achieved, without losing subjective visual quality proportionally.

## II. Background

### *A. Foveation*

Whenever the human eye fixes on a certain point, i.e. the foveation point, a spatially-varying resolution image goes to the brain. The photoreceptors in the eye non-uniformly sample the external world. At the foveation point, the full resolution image is retained, and as the distance from the foveation point increases, the resolution of the image decays exponentially [8]. Using human visual system modeling, extraneous spatial frequency from

a full resolution video stream is removed, if the foveation point is known a priori. Active research is on-going to choose the correct foveation point in an image or a video sequence. This paper, however, does not address this research area and assumes that the foveation points are known from sources like an eye tracker, computer keyboard, or mouse.

The three main methods for foveating a video sequence are foveation as a preprocessing step [8], DCT domain foveation [12] and foveation of motion vectors [13], [6], [7]. In the first approach the image is prefiltered with a spatially varying filter with cutoff frequency proportional to the local bandwidth. In DCT domain foveation, a non-uniform quality factor is used, so that a low quality factor will usually zero out most of the high frequency components away from the fovea. The most recent approach is motion vector foveation. Lee and Bovik [13] develop a hierarchical block matching algorithm for motion estimation, based on foveation. Bonmassar and Schwartz [7] define the exponential chirp transform to blur motion vectors away from the foveation point. Figs. 1, 2 and 3 show results of preprocessed, DCT domain and motion vector foveation, respectively. For all the three images the foveation point is at the center of the image.

Foveation is an emerging technology which is used for image and video compression. It is also used for thinwire visualization. Here it is assumed that large databases of images are stored on the server end. The client communicates with the server using a thinwire. So, progressive transmission can be obtained on the client side – where the foveation region is updated first and the background is updated accordingly.

### *B. UBC's H.263 Video Encoder*

The H.263 Version 2 (H.263+) video encoder was developed by Cote, Erol, Gallant and Kossentini [2]. These 23,000 lines of C code (720 kbytes) were written for desktop PC applications and sacrifices memory usage. This encoder incorporates the baseline H.263 encoder with optional H.263+ modes. It was developed primarily for research purposes.

### *C. Motion estimation of video sequences*

During transmission of video sequence over wireless or wireline applications, in order to exploit the spatial and temporal redundancies, motion estimation and compensation are incorporated. A transmitted video sequence, thus contains an intra (I-) coded frame

followed by a series of predicted (P-) coded frames. The I-frames are coded as they are.

For the P-frames the best matching macroblock from the previous frame is found by computing sum of absolute differences (SAD) over a search area, and selecting the macroblock that give minimum SAD. This is motion estimation (ME), and the integer shift in macroblock position is the motion vector (MV). After ME, the macroblock is predicted via motion compensation, i.e. reconstructing the macroblock by mimicing the decoder. After motion compensation, if the error between the current and the predicted macroblock is large, the block is I-coded. If the error is small, the best matching macroblock is found again by half-pixel motion search around the current integer pixel MV. The prediction error is then coded separately, and bits for the MVs are added to the bit stream.

## III. Motion Foveation

The percentage of bits allocated for the MVs is only 5% of the total number of bits in the H.263 bit stream. So, motion foveation cannot be used independently to get comparable results with preprocessed or DCT domain foveation. But, it can be used with both these methods to get bit rates closer to the target bit rate for an acceptable amount of distortion, during rate-distortion tradeoffs.

### *A. Spatial Foveation*

Bonmassar and Schwartz [7], in order to blur motion away from the fovea use the exponential chirp transform to weigh the MVs near the foveation point more heavily than the MVs away from the foveation point. In the H.263 standard if the MVs are manipulated in this way, the prediction error increases and the block is coded as an I-block, thereby increasing the bit rate. Changing the threshold of this intra/inter block selection would prevent blocks being intra coded. But, this would also affect the option for detecting scene changes in a video sequence.

Thus, instead of changing this threshold, the bits allocated for the MVs are truncated depending on the distance from the foveation point. Also, in H.263 coding, a row of macroblocks is treated as a slice the and the row is variable length encoded. So, maximum compression is achieved, when the MVs are the same for the entire row. Thus, depending on the row number and the local bandwidth, bits allocated for MVs are truncated to

produce motion blur away from fovea.

### *B. Temporal Foveation*

Reeves and Robinson [14] introduce temporal foveation for MPEG II. The image is segmented defining regions of interest (ROI), with the foveation point placed on each of these ROIs. The update rate of these ROIs is higher than the background update rate, creating a blurred background behind a full resolution image as shown in Fig. 5. His concepts were later used for MPEG IV [15] and is efficient for progressive transmission.

In the H.263 video coding, randomly chosen macroblocks are intra coded to correct error propagation. This was modified to have update rate inversely proportional to the distance from the fovea. For the same subjective quality, this produced a 3% lower bit rate.

## IV. Results

### *A. Comparison*

Foveation of MVs is implemented along with preprocessing and DCT domain foveation. Table I summarizes bit rates for 60 frames of CIF resolution ($352 \times 288$) Mobile and News sequence. Here, methods #1, #2 and #3 refer to preprocessed, DCT domain and motion foveation, respectively. The figures in brackets indicate the bit rate of the corresponding unfoveated video sequence. The Mobile sequence has lot of motion in the background. Thus, foveation here gives 70% bit rate reduction compared to the unfoveated video stream. On contrary, the News sequence having less motion in the background gives 30% more compression with foveation. Table II compares the three methods discussed above for 60 frames of the CIF resolution Mobile sequence.

| Sequence/Method | #1 | #1,3 | #2 | #2,3 | #1,2 | #1,2,3 |
|---|---|---|---|---|---|---|
| Mobile (293) | 92 | 90 | 133 | 131 | 85 | 83 |
| News (29) | 20 | 20 | 24 | 24 | 20 | 20 |

TABLE I

Comparison of the methods for compressing 60 frames of Mobile and News sequence

[File sizes are reported in kbytes]

| Method | Method #1 | Method #2 | Method #3 |
|---|---|---|---|
| Computation | $0(N^2)$ | O(1) | O(1) |
| Encoder modification | Not required | Required | Required |
| Compression | 70% | 55% | additional 2% |
| Quality | Good | Block artifacts | Motion artifacts |

TABLE II

COMPARISON OF THE THREE METHODS FOR COMPRESSING 60 FRAMES OF MOBILE SEQUENCE

### *B. Quality Measures*

The conventionally used measures for image quality assessment are the mean squared error (MSE) and the peak signal to noise ratio (PSNR). But these metrics hold good only when the noise is additive and image independent [16]. However, foveal distortions have both additive noise and frequency distortion components. Thus, these metrics fail to quantify the visual quality for foveal systems.

Lee and Bovik [13] define the foveal mean squared error (FMSE) and the foveal peak signal noise ratio (FPSNR). In both the cases, the usual definitions of MSE and the PSNR are weighted by the local bandwidth, which in turn is dependent on the distance from the foveation point. Mathematically,

$$FMSE = \frac{1}{\Sigma_{n=1}^{N} f_n^2} \Sigma_{n=1}^{N} [a(x_n) - b(x_n)]^2 f_n^2 \tag{1}$$

and

$$FPSNR = 10 \times \log_{10} \frac{\max[a(x_n)]^2}{FMSE} \tag{2}$$

where $f_n$ is the local bandwidth at the $n^{th}$ point, and $b(x_n)$ is a compressed version of an original frame $a(x_n)$ or a foveated frame $a(x_n)$.

The difference in PSNR of foveated and unfoveated video is 18% whereas the FPSNR difference is 3.3%. Thus, it is more logical to use the FPSNR measure for foveated system as it takes into account the HVS properties. However, FPSNR measure does not take the frequency distortions into account.

Table III summarizes the target, achieved bit rate with and without motion foveation. At comparable distortion, motion foveation gives bit rates closer to the target bit rate, specially for low bit rates.

| Target bit rate | 50 | 100 | 150 | 200 |
|---|---|---|---|---|
| Without motion foveation | 67 | 126 | 171 | 204 |
| With motion foveation | 65 | 121 | 165 | 204 |

TABLE III

TARGET AND ACHIEVED BIT RATES (IN KBPS) WITH AND WITHOUT MOTION FOVEATION FOR COMPRESSING 60 FRAMES OF $352 \times 288$ CIF RESOLUTION MOBILE SEQUENCE

## V. Conclusion

Perceptually lossless video compression systems can be designed using foveation. Foveating the MVs reduces the bit rate by an extra 1–2%. Thus, if the distortion is acceptable, bit rates closer to the target bit rate can be achieved with motion vector foveation. If blurring of the image is not acceptable, then the image quality at the receiver end can be increased using fovea-first transmission. Although foveation will introduce some additional computational complexity, the lower bit rate for the same subjective quality achieved through foveation is worth the complexity for digital video.

To build an entire foveal system an accurate model for image quality assessment needs to be developed, which would take into account both frequency and noise artifacts. While our peripheral vision acts as a down sampler, our foveal vision acts as an upsampler. So, during object recognition, the eye focuses on points of interest and upsamples the information obtained to get recognition clues. Thus, this property can also be used for target tracking and object recognition systems.

## VI. Demonstration

The foveated H.263 video bitstreams are given in the attached floppy. For the Mobile and New sequence the foveation point is on the red ball and on the left hand side face, respectively. These bitstreams can be decoded with the standard decoder *tmndec*.

## References


[1] B. Erol, F. Kossentini and H. Alnuweiri, "Efficient Coding and Mapping Algorithms for Software-Only Real-Time Video Coding at Low Bit Rates," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, pp. 843–856, Sept. 2000.

[2] G. Cote, B. Erol, M. Gallant and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, pp. 849–866, Nov. 1998.

[3] B. Erol, F. Kossentini and H. Alnuweiri, "Implementation of a Fast H.263+ Encoder/Decoder," in *Proc. IEEE Asilomar Conf. on Signals, Systems and Comp.*, vol. 1, pp. 462–466, Nov. 1998.

[4] ITU Telecom Standardization Sector, "Video Coding for Low Bit Rate Communication," *Draft ITU-T Recommendation H.263 Version 2*, Sept. 1997.

[5] S. Banerjee, H.R. Sheikh, L.K. John, B.L. Evans and A.C. Bovik, "VLIW DSP vs. Superscalar Implementation of H.263 Video Encoder," in *Proc. IEEE Asilomar Conf. on Signals, Systems and Computers, in press*, Oct. 2000.

[6] G. Bonmassar and E.L. Schwartz, "Real-Time Restoration of Images Degraded by Uniform Motion Blur in Foveal Active Vision System," *IEEE Trans. on Image Processing*, vol. 8, pp. 1838–1842, Dec. 1999.

[7] G. Bonmassar and E.L. Schwartz, "Space-Variant Fourier Analysis: The Exponential Chirp Transform," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1080–1089, Oct. 1997.

[8] P.T. Kortum and W.S. Geisler, "Implementation of a Foveated Image-Coding System for Bandwidth Reduction of Video Images," in *Proc. SPIE Conf. on Human Vision and Electronic Imaging*, vol. 2657, pp. 350–360, Apr. 1996.

[9] S. Lee, M. S. Pattichis and A.C. Bovik, "Rate Control of Foveated MPEG/H.263 Video," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, pp. 365–369, Oct. 1998.

[10] 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 Acoustics, Speech, and Signal Proc., submitted*, vol. 2657, May 2001.

[11] R. Kresch and N. Merhav, "Fast DCT Domain Filtering Using the DCT and the DST," *IEEE Trans. on Signal Processing*, vol. 8, pp. 821–833, June 1999.

[12] N. Tsumura, C. Endo, H. Haneishi and Y. Miyake, "Image Compression and Decompression Based on Gazing Area," in *Proc. SPIE Conf. on Human Vision and Electronic Imaging*, vol. 2657, pp. 361–367, Apr. 1996.

[13] S. Lee and A.C. Bovik, "Motion Estimation and Compensation for Foveated Video," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, pp. 615–619, Oct. 1999.

[14] T.N. Reeves and J.H. Robinson, "Adaptive Foveation of MPEG Video," in *Proc. ACM Conf. on Multimedia*, vol. 1, pp. 231–237, Nov. 1996.

[15] S. Battista, F. Casalino and C. Lande, "MPEG-4: A Multimedia Standard for the Third Millennium," *IEEE Multimedia*, vol. 7, pp. 76–84, Mar. 2000.

[16] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans and A. C. Bovik, "Image Quality Assessment Based on a Degradation Model," *IEEE Trans. on Image Proc.*, vol. 9, pp. 636–650, Apr. 2000.

Fig. 1. Foveation as preprocessing

Fig. 2. DCT domain foveation

Fig. 3. Motion vector foveation

Fig. 4. Temporal domain foveation