Motion Estimation and Compensation of H.263 Video via Foveation

Serene Banerjee

Multidimensional Digital Signal Processing Literature Survey

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 non-uniformly. Utilizing this property in video compression, more bits can be allocated around the point of fixation, i.e. the foveation point, to produce low bit rate foveated video. For the same bit rate, foveated video will have higher subjective quality compared to unfoveated video for a single observer. Utilizing the foveation property, the two most computationally complex video processing steps, namely, motion estimation and compensation, can be performed in a non-uniform way. In this project, I propose to implement foveated motion estimation and compensation for an H.263 video encoder.

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] 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 [2]. Utilizing the non-uniform resolution property of the human visual system, my objective is to perform motion estimation non-uniformly [3] to reduce computational complexity and yield better rate vs. quality tradeoffs.

Fig. 1 shows the uniformly sampled original high-resolution image. The corresponding foveated image is shown in Fig. 2. Assuming that our point of fixation is at the tower, both images will have the same subjective quality, when viewed from a certain distance by a single observer. Thus, via foveation, the bit rate of a video sequence can be reduced without affecting the subjective visual quality.

II. Background

A. Foveation

Whenever the human eye fixes on a certain point, i.e. the foveation point, a spatially-varying resolution image is sent 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 [4]. Using human visual system modeling, extraneous spatial frequency from a full resolution video stream can be 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 literature survey does not address this research area and assumes that the foveation points are known from eye trackers, computer keyboard, or mouse.

There may be single or multiple foveation points, and the foveation points may be fixed or moving. If the foveation point moves and the background is blurred at the same pace as the velocity of the foveation point, then the resulting motion vectors can produce an increase in the bit rate of the video sequence. Care should be taken to avoid this scenario.

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 [1]. These 23,000 lines of C code (720 kbytes) was 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.

The slice structured mode (Annex K) [1] available in the H.263 standard allows various macroblocks in the picture frame to be combined to form a slice. Thus, the picture can be subdivided into segments consisting of a variable number of macroblocks. The slice structure consists of a slice header followed by consecutive complete macroblocks. The order of transmission of these slices can be sequential or arbitrary. The shapes of the slices may be rectangular or non-rectangular. Two additional modes are signaled to indicate the order of transmission and the shape of the slices. At the receiver end, slices can be decoded in an arbitrary order, with the slice headers serving as resynchronization points.

C. Motion estimation of video sequences

During transmission of video over wireless or wireline applications, in order to exploit the spatial and temporal redundancies in any video sequence, motion estimation can compensation is 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, instead of transmitting the entire frame, only the difference of that frame with the previous I-frame is transmitted. Discrete cosine transform (DCT) is applied to the 16×16 blocks in both of these frames. DCT followed by quantization reduces the spatial redundancies of the video sequence because quantization removes most of the high-frequencies. Motion estimation and compensation reduce the temporal redundancy.

III. Foveation as a Preprocessing Step

In this approach foveation is applied as a preprocessing step. Geisler and Kortum [4] proposed the use of pyramid level representation to foveate images. Their work was followed by Lee, Pattichis and Bovik [5]. In this method, the original image is subsampled and blurred to create multiple copies of the image at different resolutions. The foveated image is then formed by combining subsets of each of these copies. So, around the foveation point, the full resolution image is retained. As the distance from the foveation point increases, the blurred images are considered. This approach is helpful for fovea-first transmission. Fig. 3 shows the original uniform resolution image. Fig. 4 shows the foveated image which is foveated using the conventional pyramid approach.

Another approach in this area was to use wavelets for foveating images. Using wavelets Chang, Mallat and Yap [6] divided the original image into four child images. Each of these child images is a quarter size of the original image. These child images can be used to recreate the original image fully, and convey low-frequency and high-frequency content of the image in both the horizontal and the vertical directions. Depending on the distance from the foveation center, high-frequencies can be masked. Thus, compression is achieved without loss of subjective visual quality. Fig. 5 shows the original uniform resolution Seinfeld image. Fig. 6 is the wavelet foveated image where the foveation point is on the leftmost face. Fig. 7 is a wavelet foveated image with two gaze points.

The advantage of the wavelet approach over the pyramidal decomposition is that no redundant information is added to build the wavelet pyramid. But wavelet method is computationally more complex and gives more block artifacts. For both of these methods the encoder need not be changed as foveation is added as a pre-processing engine. The drawback, however, is that foveation adds to the complexity of encoding.

IV. Foveation in DCT Domain

A second approach, found in the literature, is the use of variable-quality DCT compression. This method was proposed by Tsumura, Endo, Haneishi and Miyake [7]. Faster algorithms for DCT domain filtering is presented by Kresch and Merhav [8]. In the block DCT compression algorithm, a single quality factor is set for the whole image. For foveation, however, a non-uniform quality factor is used. Around the foveation point the quality factor approaches 100% and as the distance from the foveation point increases, the quality factor is reduced. The low quality factor will usually zero out most of the high frequency components of the image. Thus, spatially-varying filtering is obtained.

This method has very low additional complexity for foveation, as foveation steps involves only the division by a constant amount which means that division can be implemented by multiplication of the DCT coefficients, which have to be computed anyway. However, the potential problem with this approach finding the accurate quality factor which will model the HVS correctly. Also, this approach usually leads to blocking artifacts. In the H.263 video encoding this blocking artifact can however be removed using the optional de-blocking filter mode (Annex J) [1]. Fig. 8 shows a DCT domain foveated image. As around the foveation point the quality factor approaches 100%, the image quality there is better than the image quality produced by the pyramidal decomposition.

V. Foveating Motion Vectors

The most recent approach focuses on foveating the motion vectors in a video sequence. The first work on foveation of motion vectors was done by Reeves and Robinson [9] for MPEG II. In a given image, they define regions of interest (ROI), and segment the image, as shown in Fig. 9. A foveation point was placed at each of these ROIs. They argued that as the viewer will be concentrating at the foveation points, non-uniform sampling can be performed in the temporal domain by having different frame update rates for the ROIs and the periphery. As the distance between the foveation point and pixel in consideration increases, the 16×16 macroblock containing that pixel is updated less often. His concepts were later used for MPEG IV.

Lee and Bovik [10] develop a hierarchical block matching algorithm for motion estima-

tion. Each image is decomposed into a multiresolution pyramid, in which the downsampled images are blurred versions of the original image as discussed in [5]. For each frame, the subsampling rate is adaptively changed according to the local bandwidth of the foveated frame. The motion vectors are computed using the subsampled version.

In order to compare two foveated images, they define a new criterion called the *foveal mean absolute distortion* (FMAD). So, during motion estimation, two macroblocks are compared using the FMAD criteria. In the normal motion estimation method, for each 16×16 macroblock, the mean absolute distortion (MAD) is measured for all the macroblocks in the search area. The best match is the macroblock that gives the minimum MAD. For the FMAD criterion, the absolute difference between the pixels are weighted depending on the distance of the pixel from the fovea. So, more weightage is given to the pixels near the foveation point and weighting decays exponentially. The results are shown in Fig. 10.

To compare the quality of two foveated video sequences, they also defined 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 $a(x_n)$.

Kresch and Merhav [8] have another method of foveating motion vectors. They argue that the eye acts like an 'open shutter'. For example, in the picture of the highway as shown in Fig. 11, the motion of the vehicles closer to us will appear to be blurred. Although, the distant vehicles are moving at the same speed, their velocity on the final projection plane, being relatively low, will produce little or no blur. So, if some nearby object is away from the foveation point, the motion vectors for it can be blurred.

In order to blur motion vectors away from the foveation point, in [11] they define the exponential chirp transform. This transform weighs the motion vectors near the foveation

point more heavily than the motion vectors away from the foveation point. They developed fast algorithms for implementing the chirp transform and the complexity of the process is $N_1N_2\log(N_1N_2)$ where $N_1 \times N_2$ is the dimension of the log transformed image. At the receiver end, a variant of the Wiener filter is used to restore the quality blurred image [3].

VI. Comparison of Previous Methods

In any video system, the key design considerations are compression, complexity and quality. For all the three methods, the compression figures stated are comparable. Table I summarizes the two other parameters for the methods discussed above.

Method	Method #1	Method #2	Method #3
Complexity	$0(N^2)$	O(1)	O(Nlog$N$)
Quality	good	block artifacts	good

TABLE I

Comparison of previous methods for compressing $N \times N$ frames

VII. Proposed Method

In my project, I propose to implement foveation of motion vectors for the H.263 video using the optional slice structured mode. When the foveation point is fixed, depending on the distance from the foveation center various slices can be formed. The bits allocated to each of these slices can be made a function of the distance from the foveation center. The will reduce the total number of bits required compared to unfoveated video. Apart from spatial foveation, I will also implement temporal foveation, so that slices away from the foveation center can be updated less often. Since this operation is based on the principle of foveation, the reconstructed video should appear to be perceptually lossless to the observer. Thus, network bandwidth is reduced without sacrificing visual quality.

When the foveation point moves, the background will only be updated if the foveation point stays at the new position for a finite amount of time. The background will however not be foveated if the foveation point moves back and forth. This will guarantee that the bit rate will not increase for calculating the new motion vectors. After implementation,

the quality of the foveated video sequence produced will be compared to the previous methods using the FPSNR criterion [10].

VIII. Conclusion

Using foveation perceptually lossless video compression systems can be designed. At the same bitrate, foveated video will have higher subjective quality. For the same subjective quality, foveated video will have lower bit rate. So, it will find immense applications where digital images have to be transmitted over a slow channel.

If blurring of the image is not acceptable, then the image quality at the receiver end can be increased using fovea-first transmission. This will also help in image recognition systems, if foveation points are placed on the region of interests. 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.

References

[1] 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.

[2] 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*, Oct. 2000.

[3] 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.

[4] 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.

[5] 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.

[6] E. Chang, S. Mallat and C. K. Yap, "A Wavelet Approach to Foveating Images," in *Proc. ACM Symposium on Computational Geometry*, vol. 13, pp. 397–399, Mar. 1997.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

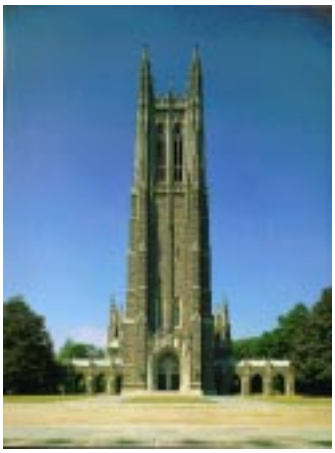

Fig. 1. Unfoveated high-resolution image.

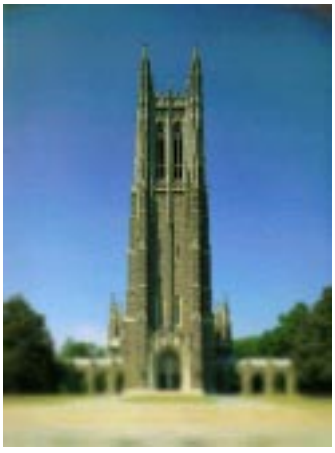

Fig. 2. Corresponding foveated image.

Fig. 3. Original uniform resolution image.

Fig. 4. Foveated image using conventional pyramids.

Fig. 5. Unfoveated high-resolution image.

Fig. 6. Corresponding wavelet foveated image, with foveation point on the left face

Fig. 7. Corresponding wavelet foveated image, with two foveation points

Fig. 8. DCT domain foveated image

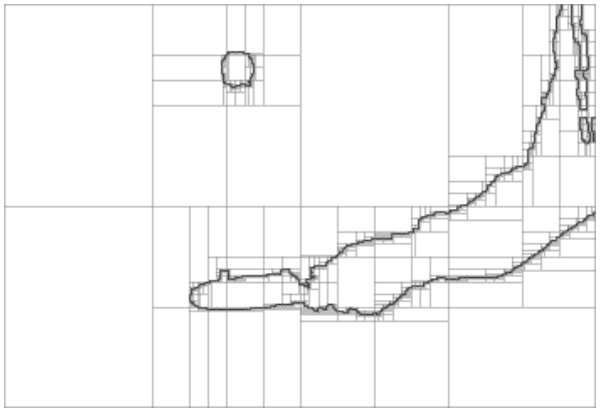

Fig. 9. Region of interest coding of motion vectors for MPEG II

Fig. 10. Hierarchical block matching algorithm for motion estimation

Fig. 11. Motion vectors away from the foveation point are blurred