FOVEATED MULTIPOINT VIDEOCONFERENCING AT LOW BIT RATES

Hamid R. Sheikh, Shizhong Liu, Zhou Wang 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.
Email: {sheikh, sliu2, zwang, bovik}@ece.utexas.edu

ABSTRACT

Multipoint videoconferencing (MPVC) involves three or more participants engaged in video communication over a network. A video server combines the video streams from each participant and then broadcasts the resulting stream to all participants. In this paper, we propose to use foveation, which is non-uniform resolution representation of an image reflecting the sampling in the retina, to reduce the bandwidth requirements of MPVC. We develop foveated MPVC algorithms for variable and constant bit rate MPVC. We show that foveated MPVC can provide considerable bit rate savings, and for the same bit rate, provide improvement in subjective quality.

1. INTRODUCTION

Multipoint Videoconferencing (MPVC) is an extension of the simple point-to-point videoconferencing. In this application, three or more participants wish to communicate visually with each other over a network. With recent advances in networking and communications technologies, such applications are becoming increasingly popular, and a number of techniques have been proposed in the literature to this end [1, 2, 3, 4]. In all these approaches, a number of participants wish to have a videoconferencing session. Each party has a video communication terminal (see Fig. 1) and is connected to a video server, the Multipoint Control Unit (MCU), through a communication medium (e.g. a network). Compressed video is transmitted by each party to the MCU, which combines the incoming streams from different participants into one video stream and broadcasts it to all participants. This constitutes "continuous presence" videoconferencing session, in contrast to a "switched presence" session in which the MCU broadcasts the video stream received from the speaker to all other users . For current standards, a continuous presence MPVC application with four users is particularly convenient to implement [1, 2, 3] such that each participant appears in one quadrant of the broadcast video.

Multipoint videoconferencing with four participants, however, requires about four times greater bandwidth for broadcast as compared with point-to-point videoconferencing. The problem of reducing this bandwidth requirement is therefore important to address. In this paper, we propose using *foveation* for reducing the bandwidth requirements for MPVC over low bit rate networks. Foveation, which is non-uniform resolution perception of the visual stimulus by the Human Visual System (HVS) due to the non-uniform density of photoreceptor cells in the eye, has been demonstrated to be useful for low bit rate video coding using existing standards, and real-time algorithms for foveated video coding have been explored previously [5, 6]. Foveated video coding improves the subjective quality at low bit rates, based on certain assumptions about the viewing configurations.

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

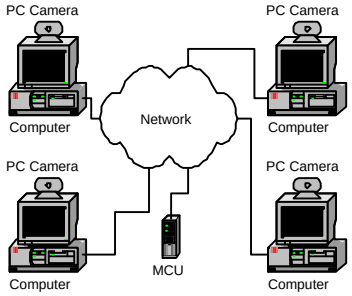

Fig. 1. Multipoint videoconferencing. The MCU combines the inputs from all participants and broadcasts it.

2. BACKGROUND

2.1. Multipoint Videoconferencing

A typical MPVC system is shown schematically in Fig. 1. The role of the MCU is crucial in all MPVC systems proposed in the literature. Besides controlling the MPVC session, the MCU combines the four incoming video streams into one stream by decoding them completely (pixel domain combining) [2], partially (coded domain combining) [1, 4], or by simple multiplexing [3], the bit streams received from each participant and re-encoding them such that each participant appears in one of the four quadrants of the output video stream. In the literature, four QCIF (176×144) streams have typically been combined into one CIF (352×288) stream using the H.261 standard. Our modification of the previous MPVC techniques can work with either approach with four QCIF streams to one CIF, or with four CIF streams to one 4CIF (704×576) stream, the latter being supported in the H.263 video coding standard for low bit rate video communication [7].

2.2. Foveated Video Coding

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. Under-

standing and modeling the limitations and abilities of HVS has been helpful in image and video engineering. Foveation is another layer of HVS modeling. 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. It 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. Thus, 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, which reduces the data rate.

Foveation has been modeled for video coding purposes with a foveation cut-off frequency model that gives the largest frequency detectable by the HVS at a given eccentricity [5, 6]. At any point on the display, a spatial frequency higher than the cut-off frequency is assumed to be imperceptible, and filtering it will not affect perceived quality. Here we give only the approximate model developed in [5], the cut-off frequency at a point (x,y) being given by:

$$\begin{aligned} f_c(x,y) &= \min\left\{\frac{i}{8} : d \geq B\left[i,V\right], 1 \leq i \leq 8, i \in Z^+\right\} \\ d &= (x-x_f)^2 + (y-y_f)^2 \\ B\left[i,V\right] &= \min\left\{r^2 : \lceil f_c\left(r,V\right) \times 8 \rceil = i, r \in Z^+\right\} \\ f_c\left(r,V\right) &= \frac{1}{1+K\arctan\left(\frac{r-R}{V}\right)} \end{aligned} \quad (1)$$

where (x_f,y_f) are the coordinates of the *fixation point* or the point under direct gaze, V is the viewing distance, $K = 13.75$ is a model parameter and R denotes the radius of a circular region around the fixation point that we wish to encode at full resolution, i.e. with $f_c = 1.0$. Figure 2 shows the cut-off frequency at different locations in the broadcast video as a grayscale map, when the participant in the upper left quadrant is assumed under fixation.

3. FOVEATED MULTIPOINT VIDEOCONFERENCING

Foveated MPVC is simple in concept. The video broadcast to every participant is foveated according to certain assumptions about their fixation points, using one of the efficient techniques in [5]. In one simple implementation, the MCU can use the audio stream to decide which participant is active (speaking) and then assume that all other participants are fixating on the active participant. User controlled pointing devices, or eye-tracking devices, may be used to change the default fixation point for each participant, depending on the application. Multiple fixation points can easily be incorporated into the model [5].

There are two possible implementations of foveated MPVC.

1. The MCU communicates the fixation point to the video encoder at the participant terminal. The video encoder implements a foveated video encoding algorithm [5] and transmits a foveated stream to the MCU. The MCU combines the foveated streams from each of the participants into one stream by using pixel domain or coded domain methods and broadcasts it. Alternatively, the MCU may multiplex the streams from the four participants using a transportion layer protocol.
2. The MCU assumes minimum capability at the participants' video terminals and performs the foveation itself. The participants transmit uniform-resolution video streams to the MCU, which combines them into one stream as well as performs foveation.

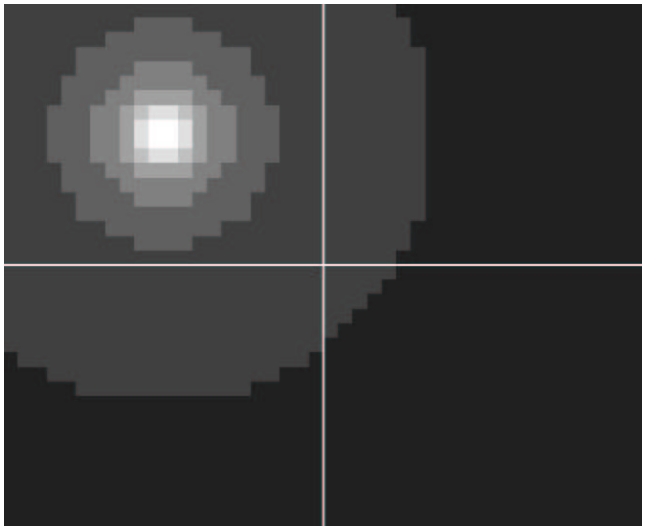

Fig. 2. Grayscale map of f_c

The above methods have their advantages and disadvantages. Method 1 is computationally cheaper than method 2 because, in the second method, the reference frames reconstructed inside the encoder are different from those at the decoder, due to foveation by the MCU. The MCU has to compensate for this reference change, either by fully decoding the video streams and then re-encoding them with foveation, or by applying some DCT domain compensations. In our simulations of the method 2, we fully decode the streams at the MCU and then re-encode them with foveation. However, method 1 is less flexible because it is assumed that the encoder at the participant's end has foveation capability.

3.1. Constant Bit Rate foveated MPVC

There are two options in foveated MPVC: variable bit rate (VBR) foveated MPVC and constant bit rate (CBR) foveated MPVC. In the VBR MPVC the video broadcast to the participants by the MCU has a bit rate that varies with the content of the video. In CBR MPVC, the MCU has to maintain the output bit rate. While rate control is built into standard video encoders, we can optimize it by allocating fewer bits to the streams corresponding to inactive participants. For CBR coding, a target bit rate has to be communicated by the MCU to the participant encoders. Here we develop an allocation scheme that divides the total available bit rate to the MCU into target bit rates for the participants encoders based on the cut-off frequency model.

	(a)	(b)
T_f	0.6384	0.97
T_h	0.1275	0.26
T_v	0.1591	0.25
T_d	0.0750	0.12

Table 1. Target bit rate share of each quadrant ($V = 500, R = 15$ pixels): (a) method 1 (b) method 2 .

3.1.1. CBR MPVC for method 1

Bit allocation for foveated video coding has been explored previously [8], where the number of bits assigned to a region in the original cartesian coordinates is proportional to the area of the region after a coordinate transform. This coordinate transform $\Phi(\mathbf{x})$ is defined such that the non-uniform sampling density in the original coordinate system becomes uniform in the new coordinate system. For a given spatial region R, the area of its corresponding image in the new coordinate system is

$$A_c = \int_R |J_\Phi|$$

where J_Φ is the Jacobian determinant of $\Phi(\mathbf{x})$. Assuming that $|J_\Phi|$ is proportional to the square of the cut-off frequency, then we can design a bit allocation scheme using f_c defined in (1). For a target bit rate of T_{MCU} bits per second for broadcast by the MCU, we define T_f to be the fraction of T_{MCU} allocated to the quadrant with the fixation point (e.g. the active participant), T_h to be the share of the quadrant horizontally across the active participant, T_v to be the share of the vertically across quadrant and T_d to be the share of the diagonally across quadrant and let R_f, R_h, R_v and R_d be the respective spatial regions. Then T_f is given as:

$$T_f = \frac{\left(\int_{R_f} f_c^2\right)}{I} \quad (2)$$

where I denotes the integral of f_c^2 over the display region, i.e. the union of the four quadrants. Other ratios, T_h, T_v and T_d are similarly defined. For evaluating the integrals, we may either use the approximate foveation model given in (1) or use the exact model in [5]. The values calculated using (1) are given in Table 1 (a) where the fixation point is assumed to be the center of the active participant quadrant. The MCU communicates the target bit rate to each of the participants by computing their respective shares of the total bandwidth using these ratios.

3.1.2. CBR MPVC for method 2

For method 2, the encoders at the participants' video terminals are assumed to be uniform resolution encoders (without foveation) but capable of doing rate control. In this case, T_{MCU} needs to be divided such that after foveation by the MCU and rate control, the output bit rate is T_{MCU}. Foveation will provide savings in each of the four quadrants depending upon the video sequence. If we assume that we know the relative savings in each quadrant, we can convert the bandwidth share computed in Table 1 (a) into bandwidth shares for method 2. In our simulations, we estimated the relative savings using trials and then updated Table 1 (a) as Table 1 (b) by multiplying each entry by the corresponding compression ratio by foveation for that quadrant. This is a very rudimentary technique and the encoder in the MCU can use some adaptive technique to calculate the compression ratios by foveation for each of the participants to compute the bandwidth ratios for each participant based on T_{MCU}.

	Method 1	Method 2
Top left	1.52	–
Top right	2.05	–
Bottom Left	1.60	–
Bottom Right	1.68	–
VBR foveated MPVC	1.66	1.62

Table 2. VBR foveated MPVC compression ratios for different methods

4. RESULTS

In this section we give results of applying the algorithms in this paper. We use the spatial domain algorithm [5] for foveation and do MPVC from four CIF resolution streams to one 4CIF resolution stream using the H.263 standard. The test sequences used are 'salesman' (top left), 'akiyo' (top right), 'claire' (bottom left) and 'silent' (bottom right). The fixation point is at the center of the upper left quadrant. In our simulations, we assume lossless multiplexing by the MCU in method 1 and pixel domain combining for method 2.

Table 2 shows the compression ratios obtained by foveation alone for VBR foveated MPVC with H.263/MPEG-4 quantization parameter $QP = 10$. Note that for method 2, the first four rows are empty because the MCU receives uniform resolution video streams. We now give results of applying CBR foveated MPVC algorithm for a *target* bit rate of 256 kbps.

Figure 3 (a) shows the reconstructed 40^{th} frame from applying method 1 without foveation, where the MCU simply combines the sequences from the participants. In Fig. 3 (a) each participant is required to code at 64 kbps. Correspondingly, Fig. 3 (b) shows the result of applying method 1 with foveation. Notice that the quality of 'salesman' is superior whereas the rest of the sequences appear blurry. The averate bit rates (over first 60 frames) are 283 kbps and 218 kbps respectively.

Figure 3 (c) shows the output of method 2 without foveation, where we assume that the MCU has the ability to do rate control. Each participant sends uniform resolution video at 256 kbps. Correspondingly, Fig. 3 (d) shows the result of using method 2 with foveation. Notice again that the quality of 'salesman' is superior compared with the other participants. The averate bit rates (over first 60 frames) are 256 kbps and 258 kbps respectively.

5. CONCLUSIONS

In this paper, we have developed techniques for reducing the bandwidth requirements of MPVC by using foveation. We have developed and demonstrated the feasibility of our foveated MPVC algorithms for VBR and CBR MPVC. We have demonstrated that foveated multipoint videoconferencing can provide significant bit rate improvements, and for constant bit rate MPVC, can provide subjective quality improvements as well.

Fig. 3. Reconstructions from simulations: (a) Uniform resolution method 1 (b) Foveated method 1 (c) Uniform resolution method 2 (d) Foveated method 2

6. REFERENCES

[1] Q.-F. Zhu, L. Kerofsky, and M. B. Garrison, "Low-delay, low-complexity rate reduction and continuous presence for multipoint videoconferencing," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 9, pp. 666–676, June 1999.

[2] M.-T. Sun, T.-D. Wu, and J.-N. Hwang, "Dynamic bit allocation in video combining for multipoint conferencing," *IEEE Trans. Circuits and Syst.–II: Analog and Dig. Signal Proc.*, vol. 45, pp. 644–648, May 1998.

[3] S.-M. Lei, T.-C. Chen, and M.-T. Sun, "Video bridging based on H.261 standard," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 4, pp. 425–437, Aug. 1994.

[4] M.-T. Sun, A. C. Loui, and T.-C. Chen, "A coded-domain video combiner for multipoint continuous presence video conferencing," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 7, pp. 955–863, Dec. 1997.

[5] H. R. Sheikh, S. Liu, B. L. Evans, and A. C. Bovik, "Real-time foveation techniques for h.263 video enoding in software," in *Proc. Int. Conf. on Accoustics, Speech and Signal Proc. (ICASSP-01)*, May 2001.

[6] H. R. Sheikh, "Real-time foveation techniques for low bit rate video coding," Master's thesis, Dept. of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78731, May 2001.

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

[8] 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.