High Quality, Low Delay Foveated Visual Communications over Mobile Channels[1]

Sanghoon Lee
Yonsei University, Korea

and

Alan C. Bovik[2]
The University of Texas at Austin, Austin, TX, 78712
E-mail: bovik@ece.utexas.edu

and

Young Yong Kim
Yonsei University, Korea

Version: May 28, 2004

Foveated video streams are created by selectively varying the spatio-temporal resolution of video data according to the assumed or measured fixation patterns of the intended viewers. The significant potential of foveated video lies in the considerable entropy reduction relative to the original video data while minimizing the loss of apparent visual information. By exploiting new human-machine interface techniques, such as visual eye-trackers, combined with foveated video compression and communication protocols, more efficient visual multimedia services operating over low bandwidths should become available in the near future. In this paper, we introduce a prototype foveated visual communication system suitable for implementation as a core element of an interactive multimedia wireless communication environment. We demonstrate the benefit of using foveation in noisy wireless low bandwidth applications, using measured fading statistics from the downtown area of Austin, Texas as an example. Based on a maximum source throughput criterion, the source and channel video transmission rates and the target video coding bit rate are adaptively decided according to the channel dynamics. In the simulations, we use the channel throughput, the spatial/temporal resolution, and the transmission delay as criteria to compare the performance of the foveated approach relative to normal (non-foveated) video. The results clearly underline the significant potential of foveated video communication protocols for wireless multimedia applications.

Key Words: foveated visual communications, foveated video, wireless video
Keywords: foveation, video compression, image compression, digital video, rate control.

[1]This work was supported (in part) by the Ministry of Information & Communication, Korea, under the Information Technology Research Center (ITRC) Support Program

[2]Corresponding author: Professor Alan C. Bovik, the Laboratory for Image and Video Engineering (LIVE), Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712-1084 USA; Phone: (512) 471-5370; Fax: (512) 471-1225; E-mail: bovik@ece.utexas.edu.

1. INTRODUCTION

Research into real-time visual communications over wireless or wireline channels has made significant recent progress because of significant advances in understanding human vision, video compression, network design, communication protocols, and the interplay between these areas.

In applications involving point-to-point visual communications, each subscriber is allocated a given bandwidth that is determined according to the required quality of the reconstructed video data. Because of the increased availability of very low-bandwidth digital communications channels coupled with multimedia applications that require high-quality visual service, the development of more powerful video compression and transmission techniques remains of significant topical interest. One promising means for increasing the efficacy of video transmission and compression takes into account the interface between the human user and the displayed video, as well as an important property of the human visual system.

The retina of the human eye samples visual information non-uniformly in space: the resolution is highest where the optical axis of the eye strikes the retina (known as the *fovea*) and it decreases as a function of angular distance (*eccentricity*) from this point. The repurcussion of this is that a human observer sees only a small part of any scene with high resolution at a given moment, with the rest of the scene serving as lower-resolution context or memory from fixations at prior moments. Likewise, a human observer of a video display of sufficient size (relative to the viewing distance) sees only a part of the display at high resolution. It therefore makes engineering sense that, if the fixation point of the observer(s) could be known, then the video could be transmitted with a variable resolution decreasing from fixation, with a consequent reduction in the requisite bandwidth. Indeed, this idea is not new, and several authors have proposed methods for increasing compression efficiency by exploiting the nonuniform sensing apparatus of the human eye via the creation of *foveated images* and *foveated video* [1][2][3][4].

The goal of foveated video compression algorithms is generically to improve the apparent (subjective) visual quality of the data by allocating more resources to perceptually important regions, which usually means those regions which are being fixated, but which could also encompass other measures of "interest" such as image motion, spatial images features, or even context.

Recently, foveated video has been actively studied as a promising means for managing high-quality video transmission in lowbit rate video applications involving human interaction [5][6][7][8]. For point-to-point visual communication applications, a fixation point can be chosen by an end user using an interactive approach such as a visual eye tracker, a mouse, or a touch screen. Among these, only the eyetracker may be regarded as automatic in some sense, although calibration of the viewer and monitor is generally required at the beginning of a session. Given a decided fixation point, and an assumed or measured knowledge of the viewing distance, the original video stream may be modified to match the nonuniform sampling of the eye. This may be interpreted as a process of removing undetectable high visual frequencies that occur away from the fixation point, resulting in significant entropy reduction and compressibility with a minimal loss of apparent visual information. This additional compression can be gained while maintaining compliance with standard compression formats such as MPEG, H.263 and their variants [7]. A significant advantage of foveated video is that the compression obtained by the foveation process multiplies that obtained by a standard compression format.

Hence, foveated video compression is particularly appealing for very low bitrate video coding/communication applications, such as wireless.

Fig. 1 (a) shows a single foveated image, where the foveation point is indicated by a small black "x", and where (b) is the uniformly sampled version of the image in (a). The foveated image in (a) was created by a process called *foveation filtering* [7] where a bank of lowpass filters of different bandwidths is selectively applied from spatial point-to-point according to the presumed distance form fixation in the image. In (c), the local bandwidth is displayed as a contour image (points equidistant from fixation are iso-bandwidth), while (d) depicts the fall-off in bandwidth as measured along a vertical axis passing through the foveation point.

Determining the efficacy of foveated video in terms of quality/cost/compression tradeoffs is an important goal. In [7][8], we defined a new visual quality metric termed the FSNR (*foveal signal-to-noise-ratio*) which seeks to measure video quality in a classical SNR sense, while assuming that a foveated sensor, such as the human eye, is observing the data. We have also developed methods for optimizing foveated video via new rate control protocols and foveated motion estimation and compensation algorithms [8][9]. In [10], we introduced foveation-based error resilience and unequal error protection techniques. In [9][10], a foveated image with multiple foveation points or a foveated region was obtained by using foveation filtering over the original coordinate domain. In [11], multiple foveation points or foveation regions were detected by using a face detection algorithm in I frames and an unequal error thresholding method in P frames. A live demonstration of a foveated and H.263-compressed video stream (originally, the CIF "News" sequence) at 30 Kbps (P frame bit rate) is available on the website: [12].

Based on these developments, we present in the current paper a prototype end-to-end foveated video communications system suitable for human interactive multimedia applications over wireless channels. The performance is tested using real fading channel statistics measured in the downtown area of Austin, Texas [13]. The channel statistics are decomposed into low and high frequency signals corresponding to large-scale fading (local mean or log-normal fading) and Rayleigh fading (multipath or small-scale fading), respectively. Based on a maximum source throughput criterion, we decide the target frame rate for source coding and the channel redundancy for channel coding using the decomposed fading signals. The FSNR is used as a quality assessment criterion in the rate control module. As a method of ECC (error correction coding), a BCH code and a convolutional code are used. We evaluate the performance of the system by deriving the amount of buffer underflow using the encoder and decoder underflow curves. We define the source transmission delay based on this value.

We present simulations where we construct an imaginary infinite size cellular system, where each cell has the same channel attenuation as measured in downtown Austin, Texas. For the cellular system, we measure visual quality, temporal resolution, buffer underflow and source transmission delay in terms of foveated vs. regular video, the adaptive source-channel vs. the channel separated coding, and the rate control schemes.

2. FOVEATED VISUAL COMMUNICATIONS

2.1. End-to-end system

Fig. 2 (a) depicts a block diagram of an end-to-end foveated visual communications system operating over a wireless channel. In the system, foveated video is serviced to a subscriber, where the fixation point(s) are selected by the subscriber using a mouse or an eyetracker. Since the fixation points might be multiple, or might not actually be visually fixated upon, we will generically refer to these points as *foveation points*. The selected foveation point(s) are transmitted to the service-provider or the corresponding subscriber. Object that are fixated on are temporally tracked. A process of *foveation filtering* is applied to the video sequence which nonuniformly removed higher spatial frequencies with the local bandwidth decreasing away from the foveation point(s), thus yielding a foveated video sequence. The video codec, which is generally a standard algorithm such as MPEG or H.263, can then be optimized to improve the apparent visual quality and/or compression of the video stream by exploiting the foveation-induced local spatial band-limitedness of the data. The result is a standard-compliant foveated video sequence that can be received and decoded by any subscriber on the network that is equipped with a standard decoder.

Even if there are a few questions on the exact tracking of gaze direction, it is well known that eye trackers can track the gaze direction in real-time. In the human visual system, it takes about 750 msec to recover full spatial resolution after an eye movement. Commercial systems are available that track the gazing direction with few problems, and the technologies are improving. Note that for real-time video coding, it is not necessary to use segmentation techniques (such as MPEG-4 shape coding) if foveation is the measure of region-of-interest. So, it is possible to implement foveated video in both real-time/non-real-time visual communications with low cost and high performance. For non-real time application, the foveation area is chosen prior to encoding, or a transcoding method can be employed for a given bit-stream based on the selected points. Several visual communication companies have developed security systems, web-based video broadcasting, video conferencing systems combined with eye-trackers, active cameras, and target tracking algorithms. Moreover, foveated visual communication systems have been studied in the field of virtual reality. In our view, the potential payoff suggests it to be of significance for investigation and for communicating the results of the investigations to researchers in the visual communication field.

2.2. Foveated video over wireless environments

Traditionally, wireless terminals have a relatively small size display due to the portability of equipment as well as the limited capacity of wireless links. Hence, only a limited angle of vision is needed to watch visual content on the terminals. Thus, foveated compression algorithms may not find applicability in such environments.

However, recently, highly enhanced link capacity and spectral efficiency have been demonstrated in wireless networks by utilizing wideband, multiple antenna, OFDM(orthogonal frequency division multiplexing), adaptive modulation, and so on. By utilizing the improved link capacity, it is possible to provide high bit rate video services for multiple users simultaneously under a base station.

In the meanwhile, several types of display technologies have been demonstrated to accommodate higher quality visual services over the improved link speed, such as

folding display, overhead projection and hologram. In addition, the display size of mobile terminals is also increased to adapt the increased link capacity. For instance, 3G cellular terminals are already widespread in some countries, which boast a few inches of display size in PDA form, with several Mbps of link speed. In some Asian countries, HPI(High speed Personal Internet) services more than 10Mbps user speed is about to emerge as a commercial product. These increased transmission speed will likely lead to increased display sizes. The rapid development of foveated video algorithms, eyetracking technologies, and automatic fixation-prediction algorithms suggests that the techniques discussed herein will become increasingly relevant in the near future.

2.3. Foveation point selection

Depending on the application scenario, the foveation point(s) can either be chosen by the sender (or service provider) or by the receiver (or subscriber). For stored video applications such as web-news, the foveation points can be selected prior to video compression. This could be accomplished by a skilled expert using a mouse or similar device, or even by determining the fixation points of a large sample of people when viewing the particular video, which might be useful for quality applications such as digital cinema. Of course, since the resulting stream is standard-compliant, a decision can be made at any point to cease foveation, so that any part of the video could be compressed in regular mode. Thus, foveated or regular mode can be adaptively decided according to the video content.

In a typical real-time visual application scenario, each end-to-end user may have two open windows on the screen. One window (Window 1) will display the decoded video received from the corresponding user. In the other window (Window 2), the encoded video bitstream is visually displayed as it is transmitted over the channel to the corresponding user. If the user clicks a point on Window 1, the coordinates of that point are transmitted over the channel to the corresponding encoder and used as the foveation point for a foveation filtering. If the user clicks a point on Window 2, then the foveation point is used by the transmitting encoding system at his end. Thus, foveation points can be chosen by both sender and receiver as they communicate with each other. Both users can update the foveation points or switch between foveation mode and regular mode.

In simple applications, the foveation point can be fixed at the spatial center of each frame in the video sequence. In specific applications involving known objects, such as video conferencing, the foveation point(s) can be automatically determined by performing face detection, by tracking facial motion or by using other detected human visual parameters. The general problem of automatically determining where to foveate a video sequence remains a forbidding one, however, since the problem amounts to determining where the observer will look, and what is of interest.

2.4. Quality assessment and entropy reduction

In [7], a new objective quality criterion was defined that is based on the foveated response of the human visual system. The FMSE (*foveal mean square error*) was defined as:

$$\mathrm{FMSE} = \frac{1}{\sum_{n=1}^{N_p} f_n^2} \sum_{n=1}^{N_p} \left[a(\mathbf{x}_n) - b(\mathbf{x}_n)\right]^2 f_n^2, \tag{1}$$

and the FPSNR (*foveal peak signal-to-noise ratio*) as

$$\text{FPSNR} = 10 \log_{10} \frac{\max[a(\mathbf{x}_n)]^2}{\text{FMSE}} \tag{2}$$

where N_p is the number of pixels in a picture, f_n is the local bandwidth at the n^{th} pixel, $a(\mathbf{x}_n)$ is the original image or the foveated image, and $b(\mathbf{x}_n)$ is the coded version of $a(\mathbf{x}_n)$.

For example, the foveated image as shown in Fig. 1 (a) is obtained using foveation filtering from the original image. In order to take into account the weighting factor due to the nonuniform property of the human visual system, the nonuniform foveated image is mapped into curvilinear coordinates where it becomes a uniform image. Then, the PSNR/MSE is calculated in the curvilinear coordinates. The obtained PSNR/MSE is defined as the FPSNR/FMSE and expressed by (1)/(2). Using the local bandwidth (f_n) displayed in Fig. 1 (c), the mapping ratio can be obtained. The weighted factor f_n^2 in (1) represents the mapping ratio (area ratio) of the n^{th} pixel relative to the original. Thus, it is possible to quantify the visual quality over virtual curvilinear coordinates without taking a coordinate transformation.

Similarly, the visual entropy can be obtained in accordance with the nonuniform property of the human visual system. In [8], the differential entropy over curvilinear coordinates was defined as the visual entropy $H(\Phi)$ by:

$$H(\Phi) = -\int_{\phi \in S_c} \rho(\phi) \log\rho(\phi)\, d\phi \tag{3}$$

where S_c is the region over curvilinear coordinates, Φ is a continuous random variable and $\rho(\phi)$ is the probability density function of Φ=ϕ. Let $\rho(x)$ be a probability density function of X. Let A_o and A_c be the area of an image in Cartesian and curvilinear coordinates as the displayed picture size in Fig. 1 (a) and (b). According to the definition of the visual entropy, the area differential ratio becomes the entropy gain: $G = \frac{A_o - A_c}{A_o}$.

In real video processing, the compression gain is not equal to the entropy gain due to the non-linear operation. In [8], the compression gain G_c^t was obtained by

$$G_c^t = \frac{R_o - \tilde{R}_c}{R_o} \text{ subject to } \tilde{P}_c = P_c \text{ and } q_o = q_c \tag{4}$$

where R_o is the number of generated bits when the original image is compressed using quantization parameter q_o, and $\tilde{R}_c$ be the number of generated bits when the foveated image is compressed using a non-uniform quantization, $\tilde{P}_c$ is the FPSNR of the foveated image by the non-uniform quantization, P_c is the FPSNR of the foveated image by using $q_c = q_o$.

2.5. Improved motion handling and low delay transmission

Foveated video streams afford robust motion-handling for mobile video communications applications. In natural moving scenes, the motion compensated macroblocks used by the video coding standards often contain significant high frequency DCT coefficients. In such applications involving regular video, the video traffic is usually adjusted to a low transmission rate by using large quantization parameters or by reducing the spatial/temporal resolution (frame size/frame rate). However,

by using foveated compressed video streams, much unessential high frequency DCT information is eliminated, resulting in significantly reduced coding overhead. Of course, the saved bits could also be assigned to points near foveation providing higher visual quality, if the bandwidth is available [7][8].

Foveated video also allows for low delay transmission in difficult noisy environments. If there are sudden changes to be coded, such as scene changes or fast movements, then the coder may generate more bits than the given target bit rate. In order to match the target rate, the encoder will reduce coding rate by frame skipping, resulting in degraded temporal resolution manifested as, for example, blocking artifacts, lip-synchronization difficulties, and jerkiness. Foveated video streams ameliorate these effects since the need for frame skips is greatly reduced.

In our proposed system, we utilize the ARQ (automatic reQuest) protocol to make the bitstream more robust to transmission errors. The ARQ protocol retransmits erroneous packets that arise from channel errors [14]. Channel fading in wireless systems can reduce the channel throughput to be near zero over an extended period. In such cases, channel errors can propagate both spatially and temporally in the reconstructed video.

2.6. Time division multiplexing access scheme

In the system proposed here, a time division multiplexing accesss scheme (TDMA) is employed to deliver the bitstream over a simulated wireless channel. Fig. 3 shows the TDMA scheme combined with ARQ flow control. In the example, a transmission interval consists of three slots and a packet is carried over one of the slots during a single transmission interval. The next packet is not transmitted until the transmitter receives an acknowledgment (ACK) that the previous packet was received. Since the packet propagation time in a typical wireless environment is shorter than the transmission interval, it is assumed that the ACK or the NACK (Negative ACK) signal arrives in the transmitter before the next slot becomes available. Thus, the transmitter can retransmit the current packet in the next slot whenever channel errors occur in the current packet. In [10], we limited the number of retransmission according to a maximum queuing delay bound for the foveated layer(100 msec)and for the background layer(50 msec) respectively. Here, we focus on measuring the channel adaptability of the foveated video in terms of the average queuing delay and waiting time. Thus, we do not limit the number of retransmission in the ARQ scheme.

2.7. Channel throughput at packet level transmission

The major difference between wireless and wireline channels is the bit error probability. When the bit error rate (BER) is low (e.g., 10^{-9}), the data loss can be largely ignored. In wireless mobile networks, channel errors can consecutively occur for a few seconds due to large-scale fading. Therefore, the service rate is no longer constant and depends on the channel status. When using an ARQ protocol, the service rate can be measured by counting the number of ACKs. Since the data is transmitted using a packet, the channel throughput can be calculated using the packet error probability which is a function of the channel signal-to-noise ratio (SNR), the modulation scheme, the packet size and the method of error correction coding.

Let SNR_o be the average SNR within a cell. Then, the SNR signal $SNR(t)$

corresponding to the received signal $r(t)$ can be obtained by

$$SNR(t) = r_n^2(t) SNR_o \tag{5}$$

where $r_n(t) = (r(t) - \bar{r})/\sigma_r$ is a normalized version of $r(t)$, and $\bar{r}$ and σ_r are the mean and variance of $r(t)$. Thus, the mean and variance of $r_n(t)$ become 0 and 1 respectively. Here, the ideal $\pi/4$ QPSK is assumed to be used. Then, the bit error probability $P_b(t)$ becomes [15]

$$P_b(t) = Q(\sqrt{2SNR(t)}) \tag{6}$$

where $Q(.)$ is the complementary error function defined as

$$Q(z) = \frac{1}{2\pi} \int_z^\infty e^{-x^2/2} dx. \tag{7}$$

Then, the packet error probability $P_e(t)$ without channel coding becomes

$$P_e(t) = 1 - [1 - P_b(t)]^{L_p} \tag{8}$$

where L_p is the number of bits in a packet. The above equations are based on the assumption that the fading variation is slow compared to the packet transmission time, i.e., the channel SNR during a packet transmission time is a constant. If the packet propagation time is shorter than the packet transmission interval, then the average number of transmitted bits in the packet at time t can be obtained by

$$\bar{R}(t) = [1 - P_e(t)] L_p. \tag{9}$$

2.8. The effect of fading channel on the queuing performance

Studies of the queuing response to fading channel statistics have shown that channel correlation statistics play an important role in queuing performance [16][17]. A high correlation in the channel fading generally has an adverse impact on the queuing performance. Other factors such as mobile speed, mobile driving pattern, and channel speed have been noted to effect the queuing performance.

As the mobile speed decreases, the correlation between the channel statistics increases which has adverse impact on the queuing performance. This effect becomes more pronounced the when channel speed (packets/sec) increases, since in a low speed channel, the fading variation resulting from mobile speed is less significant compared to the input traffic dynamics The queuing performance also depends on mobile driving patterns. As compared to directional driving, random driving reveals a significantly increased correlation when all other parameters are fixed. The reason for this is that a random driver tends to stay within the vicinity of a certain area longer than a directional driver. Generally, the driving pattern can change the channel dynamics and the queuing performance accordingly. Note that with different driving patterns, the first-order channel statistics may remain the same, while the second-order statistics (correlation) vary significantly, resulting in unpredictable queuing performance. Also, as the channel speed increases, the interval between any fading sample decreases, thus increasing the correlation of the fading statistics.

In channel-dependent video coding, channel parameters such as the mobile speed, the mobile driving pattern and the channel speed influence the encoder

buffer dynamics, resulting in variation in the target bit rate, and in the encoded picture quality. To analyze the performance of video transmission, we fix those parameters, then measure the visual quality and the channel adaptation for each coding algorithm: the mobile speed is set to be the speed of a car used to measure the channel characteristics in downtown Austin, Texas; the mobile driving pattern is set by the direction probabilities; and the channel speed is set to be a fixed value in the system. The relative performance of using foveated visual communications will be studied by measuring how well the proposed methods adjust to time-varying channel conditions, and by how much the video quality is improved by efficiently regulating the source rate in response to the correlated channel behavior.

3. ADAPTIVE SOURCE-CHANNEL CODING OVER MOBILE CHANNELS

In wireless channels, the SNR detected in the receiver can be fed back into the transmitter as in the IS-95 standard. Then, the transmission power or the channel coding redundancy can be adaptively changed to improve the data transmission throughput based on measurements of the time-varying channel SNR. With this approach it is easy to maintain the intrinsic function of the source and channel coders and compatibility with video standards. In [18][19], a feedback channel was used to decide source and channel coding rates. In mobile channels, the channel throughput can be severely degraded due to large-scale fading necessitating the use of ARQ. The improved channel throughput by ARQ results in ameliorating the degrading of reconstructed video quality.

In the MPEG and H.263 video standards, the expected distortion $\bar{D}_t$ of the reconstructed frames can be divided into the source coder distortion $\bar{D}_s$ and the channel distortion $\bar{D}_c$: $\bar{D}_t = \bar{D}_s + \bar{D}_c$. Spatial error propagation is caused by synchronization failures in the VLC (variable length coder). Spatial error propagation in the reference frames (I or P frames) leads to temporal error propagation over subsequent interpolated frames (P or B frames). Let $\bar{D}_{c,s}$ and $\bar{D}_{c,t}$ be the spatial and temporal distortions due to the error propagation. Then, $\bar{D}_t = \bar{D}_s + \bar{D}_{c,s} + \bar{D}_{c,t}$. For a given video coding algorithm, $\bar{D}_s$ is uniquely decided for each frame according to target bit rate. On the other hand, $\bar{D}_c$ depends on the error resilience algorithms used by the decoder and on the channel parameters (channel coding, channel SNR and data link protocol). When the ARQ protocol is used, $\bar{D}_c = 0$ and $\bar{D}_t = \bar{D}_s$. Since $\bar{D}_t$ depends only on the source distortion, $\bar{D}_t$ can be minimized by using optimal rate control protocols.

3.1. Maximum source throughput criterion

Fig. 4 shows the system overview to achieve adaptive source-channel coding at the system level. In this scheme, the channel SNR is measured by using a matched filter detector. Based on the SNR, the receiver calculates the number of target bits for source coding and the number of channel parity bits for channel coding. This information is periodically delivered into the sender to enable the selection of the source and channel rates.

Let t_f be the transmission interval in the TDMA system, m be the m^{th} packet, t_p be the target frame interval, and k be the k^{th} source frame. For brevity, the index m and k are also used to indicate the time mt_f and kt_p. Let $P_e(m)$ be the m^{th} packet error probability in (8). Then, the average channel throughput for the

traffic of the k^{th} picture becomes $\bar{R}_T^k = \bar{R}_c + \bar{R}_s$.

$$\bar{R}_T^k = \sum_m [1 - P_e(m)]L_p, \text{where } m \in [k, k+1) \tag{10}$$

The optimal rate control problem is to find the state vector $\vec{Q}$ which minimizes the overall distortion $D(\vec{Q})$ subject to the rate constraint $R(\vec{Q}) \leq \bar{R}_s$. By introducing a Lagrange multiplier $\lambda \geq 0$, the constrained problem can be defined and solved. For λ ranging from 0 to ∞, an optimal quantization state vector $\vec{Q}^*$ is obtained which minimizes the Lagrangian cost function $J(\vec{Q}, \lambda) = D(\vec{Q}) + \lambda R(\vec{Q})$ while satisfying the rate constraint. If $\lambda_1 < \lambda_2 < \lambda_3$, $R(\lambda_1) < R(\lambda_2) < R(\lambda_3)$ and $D(\lambda_1) > D(\lambda_2) > D(\lambda_3)$, then the distortion of the reconstructed image can be minimized at the maximum rate $\bar{R}_s$.

Suppose that the channel redundancy is able to be changed at every packet. Then, the optimal number of parity-check digits a_m^* for the m^{th} packet is obtained by

$$a_m^* = \text{argmin}[\ a_m\ ] \text{ for } \max[\ \bar{R}_s(m)\ ] \text{ subject to } a_m + b_m = L_p \tag{11}$$

where a_m is the number of parity bits and b_m is the number of source bits at the m^{th} packet. Then, $\bar{R}(m) = [1 - P_e(m, a_m^*)]L_p = \bar{R}_s(m) + \bar{R}_c(m)$ is the average number of transfered bits at the m^{th} packet, and $\bar{R}_T^k$ in (10) is given by

$$\bar{R}_T^k = \sum_m [1 - P_e(m, a_m^*)]L_p, \text{where } m \in [k, k+1) \tag{12}$$

and the maximum expected source target rate is $\bar{R}_s^* = \bar{R}_T^k - \bar{R}_c = \bar{R}_T^k - \sum_m \bar{R}_c(m)$, $m \in [k, k+1)$.

3.2. Channel signal decomposition

In mobile channels, the received signal $r(t)$ can be expressed in terms of large-scale fading $l(t)$ and Rayleigh fading $s(t)$: $r(t) = l(t) \times s(t)$. The signal $l(t)$ can be extracted by using the local average power of $r(t)$ [20][21]. Let $2T$ be the time interval for averaging the local power of $r(t)$. In [20], the interval $2T$ is chosen in the range of 20λ to 40λ based on the variance of the estimated large-scale signal.

Given a carrier frequency 1.9 GHz, the range of distance required to extract $l(t)$ is 3.158 m to 6.316 m. Thus, the interval $2T$ becomes 227 msec to 455 msec at 50 km/hour (the vehicle speed) and 379 msec to 758 msec at 30 km/hour. If the target frame rate is 10 frame/sec and the transmission time for each source frame is 100 msec, then the interval $2T$ is bigger than the transmission time.

Define the following parameters: the carrier frequency f_c, the channel speed T_c, the target transmission rate for each video source T_r, and the number of serviced video sources N_s. Then, $T_c/T_r = N_s$ and $t_f = L_p/T_r$. In a TDMA system, the instantaneously fading signal for the m^{th} packet becomes $r(m)$, where the sampling period is equal to the transmission interval t_f. Let $\mathbf{r}(m)$ and $\mathbf{l}(m)$ be the power of $r(m)$ and $l(m)$, respectively, measued in decibels (dB). Then, the sample average is

$$\bar{\mathbf{r}}(m) = \frac{1}{N_f} \sum_{i=-N_f/2}^{N_f/2-1} \mathbf{r}(m+i) \tag{13}$$

where $N_f = [2T/t_f]$. If the samples $\mathbf{r}(m+i)$ are uncorrelated, then the expected value of $\bar{\mathbf{r}}(m)$ becomes $\mathbf{l}(m)$: $E[\bar{\mathbf{r}}(m)] = E[\mathbf{r}(m+i)] = \mathbf{l}(m)$. Let $\bar{\sigma}$ and σ be the variance of $\bar{\mathbf{r}}(m)$ and $\mathbf{r}(m+i)$. Then, $\bar{\sigma}^2 = \sigma^2/N_f$. In [20], it was shown that more than 90 % estimation accuracy and less than 1 dB from the mean can be maintained when $N_f \geq 36$. Also, the packet interval t_f for uncorrelated samples is greater than 0.8 λ (practical) and 0.5 λ (theoretical). Based on these specifications, we use a TDMA system for our simulation environment with the following specifications: f_c = 1.9 GHz, T_c = 1.022 Mbps, L_p = 511 bits, T_r = 51.1 Kbps, t_f = 10 msec, and N_s = 20 video sources.

In a real system, we only know the power of previous signals $\mathbf{r}(m)$ for $-N_f/2 \leq m \leq -1$. If the channel statistics are assumed to be the same during the N_f sampling period, then the large-scale fading signal $\mathbf{l}(m)$ is approximately obtained by :

$$\mathbf{l}(m) \approx \frac{2}{N_f} \sum_{i=-N_f/2}^{-1} \mathbf{r}(m+i). \tag{14}$$

Based on the value of $\mathbf{l}(m)$, a pair of the source and channel codes is decided. Using (16), a_m^* can be obtained by (11) based on the signal power $\mathbf{l}(m)$ in (14). Then, the number of transmitted source bits becomes $b_m = L_p - a_m^*$. The average channel delay for the i^{th} packet is

$$\bar{t}_i = \left[\sum_{k=0}^{\infty} \prod_{n=0}^{k} P_e(i+k)(1-P_e(i+k-1))^n \right] (k+1)t_f \tag{15}$$

3.3. Channel coding selection

Concatenated coding has been widely used to correct mixtures of random and bursty errors. Convolutional codes can be used as an inner code to reduce scattered random errors. Uncorrected bursty errors can be corrected by using an outer code such as Reed-Solomon (RS) codes. This combination of coding techniques yields powerful error correction capability at low channel SNR. However, due to the number of parity bits, the source throughput is limited. For example, when a $\frac{1}{2}$ convolutional code is used, the maximum normalized source throughput is less than 1/2. Thus, in terms of channel bandwidth utilization, an adaptive ECC is suitable for a wide range of channel SNRs.

BCH codes are capable of correcting all random error patterns, and have been extended to nonbinary codes over the Galois field. In particular, when the channel SNR increases, BCH demonstrates efficient error correction capability with a low number of channel parity bits. In addition, BCH codes can be used as an outer code combined with convolutional codes.

BCH codes provide a large class of easily constructed codes of arbitrary block length and code rate. For a binary t-error correcting BCH code where the code length $c = 2^r - 1$ ($r \in \mathbf{N}$) and the number of source bits b, the number of parity-check digits $a = c - b$ can be expressed : $a = c - b \leq rt$. Using primitive polynomials, a set of generated polynomials are uniquely obtained and the parameter values c, b and t are decided according to the minimum distance of the code. If the ECC is capable of correcting up to t bit errors, then

$$P_e(m) = \sum_{i=t+1}^{L_p} \binom{L_p}{i} P_b(m)^i (1-P_b(m))^{L_p-i} \tag{16}$$

On the other hand, the error correction capability of convolutional codes depends on the number of parity bits, the number of shift registers, the hard/soft decision and the channel correlation. As the number of shift registers increases, the error correction capability also increases at the cost of computation overhead for decoding. The channel correlation depends on the mobile velocity or on multipath fading. In order to reduce the channel correlations, interleaving schemes are used, but they require an additional transmission delay.

Since the main goal in this paper is not on the optimization of channel coding, we employ a simplified channel coding scheme based on the simulation results. In order to increase the source throughput, we use different channel coding methods according to the channel SNR. When the channel SNR is over 15 dB, then we do not use any channel coding. When the channel SNR is in the range 5 to 15 dB, a BCH(511,484) code is used. When the channel SNR is less than 5 dB, we use a concatenated code by combining a 1/2 convolutional code as the inner code and a BCH(255, 215) code as the outer code. The 1/2 convolutional code has 8 shift registers and uses (753, 561) generators.

3.4. Performance measurements for ECC

In order to measure the performance of the ECC methods, *Jakes*' method is used to generate a random fading signal $r(t)$ [22]. In this model, the number of incident fields is set to 32. From the fading signal $r(t)$, the corresponding BER is calculated according to the $\pi/4$ QPSK modulation scheme in (6). Using (8), we measure the PER and the normalized source throughput according to the ECC schemes.

Fig. 5 (a) shows the BER for the 1/2 convolutional codes vs. non-channel coding according to the mobile velocity. It can be observed that the BER is higher over the fading relative to the gaussian noise in both cases. The time-spreading mechanism due to multipath has a dual relation with the time-variant mechanism due to motion. The vehicle motion introduces a Doppler shift, which leads to variation in the amplitude of the radiating waves over time, and spectral broadening over frequency. Due to the Doppler frequency, the temporal correlation of the received signal exhibits a time-variant behavior. With decreasing mobile velocity, the correlation increases as does the duration of deep fading. Thus, in fast moving mobile situations, random errors increase with decreasing bursty bit errors. Using the convolutional code, the BER is reduced at lower channel SNRs. However, due to the spectral broadening in fast-moving mobile situations, incoherent waves sum and further locally degrade the SNR of the received signal. Even if the average channel SNR increases, the local bursty errors occur resulting is degraded performance. In Fig. 5 (a), the BER at mobile velocity 50 km/h is less than the BER at mobile velocity 25 km/h when the channel SNR is low. However, when the channel SNR increases, the performance is reversed due to bursty bit errors at mobile velocity 50 km/h.

Fig. 5 (b) shows the normalized redundancy of parity bits a_m/L_p at mobile speed 50 km/h. For the optimal BCH, the optimal number of parity bits a_m^* is obtained using (11).

The parity bit redundancy of the concatenated code (optimal BCH + 1/2 convolutional) is less than that of the optimal BCH over the channel SNR from -2.5 dB to 2 dB. When the channel SNR is greater than 2 dB, the redundancy of the optimal BCH is rapidly reduced compared to the concatenated code. In terms of

the coding redundancy, we expect that the coding performance of the concatenated code is better (relative to the BCH code) when the channel SNR is low, and worse when the channel SNR is high. The simplified ECC is an asymptotic fit to the curve of the optimal BCH code at $SNR > 5$ dB, and to the curve of the concatenated code at 0 dB $< SNR <$ 5 dB. When the channel SNR is less than -5 dB, the source throughput is near zero, so that the number of parity bits does not make an important influence on the coding performance.

In Fig. 5 (c), the normalized average source throughput $[\bar{R}_s(m)/L_p]$ is plotted for each ECC method assuming a mobile speed of 50 km/h. Optimal BCH coding gives the highest source throughput when the channel SNR is greater than 1 dB, and the optimal BCH combined with the convolutional coding gives the highest source throughput when the channel SNR is less than 1 dB. Due to the fixed number of parity bits, the convolutional code causes the source throughput to decrease when the channel SNR is greater than 5 dB. The simplified ECC method produces a compromised performance for both low and high channel SNR. Compared to non-channel coding, the normalized source throughput is improved when the SNR falls in the range -5 dB $< SNR <$ 7 dB by using the simplified ECC method.

In Fig. 5 (d), the packet error probability is shown. Using the simplified ECC, P_e can be effectively reduced when the SNR falls in the range -8 dB $< SNR <$ 9 dB even though P_e is relatively higher when the channel SNR falls in the range 5-7 dB compared to the optimal BCH methods. Since a_m^* in the optimal BCH coding is decided for maximizing $\bar{R}_s(m)$ instead of $\bar{R}(m)$, the amplitude of P_e fluctuates.

3.5. Target bit estimation and allocation

In the MPEG or H.263 video standards, target bit allocation is usually accomplished at the picture layer. Thus, the target bit information is assumed to be sent to the transmitter at the target frame rate. After predicting the channel throughput over the target frame rate, we then use it as the target rate for the next frame. The number of packets corresponding to the target frame rate is $N_a = N_s/(T_r L_p)$. Let $\bar{R}_T$ be the average number of transmitted bits over the next N_a packets. Here, $\bar{R}_T$ is estimated using two methods. The first is the estimate based on the average channel throughput of previous $N_f/2$ packets:

$$\tilde{R}_T = \frac{2N_a}{N_f} \sum_{m=-N_f/2}^{-1} \bar{R}[\mathbf{r}(m)] \tag{17}$$

The other is the estimate based on the average of the received power $\mathbf{r}(p)$ from $m-N_f/2$ to $m-1$. Then, the estimation error (%) is given by $(\bar{R}_T - \tilde{R}_T)\times 100/\bar{R}_T$. Fig. 6 shows the estimation error for both methods assuming 20-minute directional driving. It is shown that the estimation error based on the average channel throughput is less than 10 %, while the estimation accuracy is better than the other method.

Depending on the range of channel SNRs, the Rayleigh fading signal plays an important role in improving or reducing the channel throughput which must be considered when estimating the target rate. In particular, the estimation error is peak at $SNR_o = 10$ dB, since it is difficult to predict the random power attenuation of Rayleigh fading due to the nonlinearity of $Q(.)$. On the other hand, when the average SNR increases over 10 dB, the normalized channel throughput approaches 1 so that the estimation error can be reduced.

In the TMN7 and TMN8 rate controls [23][24], the encoder skips original frames until the buffer fullness falls below a fixed threshold value. However, since mobile channel statistics are time-varying, the threshold value can be adaptively changed according to the channel statistics to achieve more elaborate traffic control. Using $\tilde{R}_T$ in (17), the threshold T_h is decided by

$$T_h = \max[\tilde{R}_T,\ 0.7 \times T_r]. \tag{18}$$

If the current buffer occupancy B is greater than T_h, then the encoder skips frames until the buffer status is less than T_h. Similarly to TMN8, the target rate for the next frame is decided by

$$\begin{aligned} R_T &= \tilde{R}_T - B/F_r, \text{ for } \tilde{R}_T > 0.1T_h \\ &= \tilde{R}_T - B + 0.1T_h, \text{ otherwise.} \end{aligned} \tag{19}$$

This allocation is different using the TMN7 and TMN8 rate controls. The number of target bits depends on the channel statistics.

3.6. Rate control for foveated video

In [8][25], the visual performance of regular and foveated video streams was objectively compared by maximizing (respectively) the PSNR and the FPSNR for regular and foveated video. Using the FPSNR, we allot R_T in (19) into each macroblock in the foveated video.

Since the Lagrangian cost function of each macroblock is independent,

$$J(\vec{Q}, \lambda) = \sum_{k=1}^{N_m} \min[d_k(q) + \lambda r_k(q)] \tag{20}$$

where $N_m = N_p/256$ is the number of macroblock, $d_k(q)$ and $r_k(q)$ are the distortion and rate of the k^{th} macroblock associated with the QP q. Using the FMSE, $d_k(.)$ can be normalized. After finding an optimal QP set (using an iterative procedure) under the assumption that the QP difference is zero, then a sub-optimal QP set is decided for each macroblock associated with the corresponding local bandwidth [8].

4. TRANSMISSION DELAY FOR REAL-TIME VIDEO

4.1. Synchronous and asynchronous video

Video decoding can be divided into synchronous and asynchronous decoding according to the application. In commercial digital TV broadcasting or real-time distributed video service, the temporal resolution is usually constant. Thus, the decoding is synchronized to the encoding in such a way that each "access unit" is decoded from the decoder buffer using timing information, such as a decoding time stamp. In synchronous video, the video traffic is controlled by using a rate controller without violating the delay constraints.

In point-to-point video communications, an asynchronous decoder can be used, where the decoding and presentation time depends on the data reception time. In this case, the encoder can generate more flexible traffic depending on the complexity of the image sequence and the picture coding type, while maintaining the target

transmission rate. In low bitrate applications, skip frames can be used to regulate the traffic. Therefore, the temporal resolution also depends on the (variable) number of skip frames. When an ARQ protocol is used, the channel throughput is a function of the channel dynamics, which influences the encoder buffer occupancy. When the instantaneous source traffic is relatively larger than the corresponding channel capacity, then the number of skip frames and the transmission delay increase.

The buffer underflow curves have been used to smooth VBR video traffic over wireline networks [26][27][28]. The encoder (decoder) underflow curves indicates the minimum accumulative amount of data required for preventing encoder (decoder) buffer underflow. The amount of buffer underflow depends on the queuing delay in the encoder and the decoding waiting time in the decoder. In this paper, we utilize the underflow curves to measure the performance of the foveated video coding. For a given wireless channel attenuation, the spatial/temporal resolution and the source transmission delay are obtained by using the curves.

4.2. Encoder underflow and queuing delay

Here, we use the following notation :

L_p	the number of bits in a packet
t_f	the transmission interval in the TDMA system
m	the m^{th} packet at time mt_f
t_s	one slice or GOB (group of blocks) period which is used as a time unit for measuring the traffic
n	the slice index for indicating time nt_s
$B_e(m)/B_d(n)$	the number of bits stored in the encoder/decoder buffer at time m/n
$U_e(m)/U_d(n)$	the amount of the encoder/decoder buffer underflow at time m/n
$E(n)$	the number of bits coming from the encoder in $[n,\ n+1)$
$\hat{R}_s(m)/\hat{R}_c(m)$	the pair of dedicated source/channel bits $(b,\ a)$ in the m^{th} packet
$R_s(m)$	the number of transmitted source bits in the m^{th} packet

For brevity, we will use the notation n (m) instead of time nt_s (mt_f). Then, $L_p = \hat{R}_s(m) + \hat{R}_c(m)$ and $\hat{R}_s(m) = R_s(m) + U_e(m)$. Since the encoder buffer underflow may occur when the traffic in the encoder buffer is carried over a packet, $B_e(.)$ and $U_e(.)$ must be represented by the time index m instead of n. Therefore, $B_e(m+1)$ is given by

$$B_e(m+1) = B_e(m) + \sum_p E(n+p) - R_s(m) \tag{21}$$

where $mt_f/t_s \le n+p < (m+1)t_f/t_s$. If the time index corresponding to the first slice is n, then $E(n+p-1)$ is the generated traffic for the p^{th} slice. Generally,

$$\begin{aligned} B_e(m+l) &= B_e(m) + \sum_p E(n+p) - \sum_{q=0}^{l-1} R_s(m+q) \qquad (22) \\ &= B_e(m) + \sum_p E(n+p) - \sum_{q=0}^{l-1} \hat{R}_s(m+q) + \sum_{q=0}^{l-1} U_e(m+q) \end{aligned}$$

where $mt_f/t_s \le n+p < (m+l)t_f/t_s$. Suppose that the encoding is started from the time n for a frame and $B_e(m) = 0$. Since $B_e(m+l) \ge 0$, the encoder buffer

underflow $U_e(m+l-1)$ becomes

$$U_e(m+l-1) = max[0, \ -\sum_{p} E(n+p) + \sum_{q=0}^{l-1} \hat{R}_s(m+q) - \sum_{q=0}^{l-2} U_e(m+q)]. \quad (23)$$

Using (23), the sum of the encoder buffer underflow is obtained by

$$U_e^t = \sum_{q=0}^{Q-1} U_e(m+q) \quad (24)$$

where $Q = St_s/t_f$ is the total number of transmitted packets when S is the total number of slices in the sequence.

At time $(m+l)t_f$, $B_e(m+l)$ becomes

$$B_e(m+l) = -\sum_{p} E(n+p) + \sum_{q=0}^{l-1} \hat{R}_s(m+q) - \sum_{q=0}^{l-1} U_e(m+q) \quad (25)$$

In order to transmit $B_e(m+l)$, let $\bar{R}_s(m+l)$ be the average source throughput over the window (N_f packet interval) explained in Section 3.2 and 3.5. Then, $\tilde{R}_s(m+l)$ is written by

$$\tilde{R}_s(m+l) = \frac{1}{N_f} \sum_{i=-N_f/2}^{N_f/2-1} \bar{R}_s(m+l+i). \quad (26)$$

Then, the queuing delay at the time $(m+l)t_f$ is $t_{q_{m+l}} = d_{m+l}t_f$ where d_{m+l} is approximately obtained by

$$d_{m+l} \approx \frac{B_e(m+l)}{\tilde{R}_s(m+l)} \quad (27)$$

and the average queuing delay is $\bar{d}t_f$ where

$$\bar{d} = \frac{1}{Q} \sum_{l=0}^{Q-1} d_{m+l}. \quad (28)$$

4.3. Decoder underflow and waiting time

Decoder underflow occurs when the volume of data required for decoding exceeds the volume of transmitted data. Thus, the decoder buffer occupancy and the decoder underflow can be expressed using the slice indices $B_d(n)$ and $U_d(n)$ rather than the packet index m. The waiting time $t_{w_l} = w_l t_s$ in the decoder is defined by: the waiting time until the decoder starts decoding after a packet arrives in the decoder buffer.

Let $t_{w_1} = w_1 t_s$ be the initial decoder waiting time. Since the decoder buffer underflow curve is a time-shifted version of the encoder buffer underflow curve by $d_1 t_f + w_1 t_s$, the decoder underflow is calculated using the decoder buffer occupancy $B_d(n)$ and the encoded bits $E(n)$. Denote $n' = n + [d_1 t_f/t_s] + w_1$ and $m' = [n' t_s/t_f]$. Then,

$$B_d(n'+1) = B_d(n') - E(n') + \sum_{q} R_s(m'+q) + U_d(n') \quad (29)$$

where $n't_s/t_f \leq m'+q < (n'+1)t_s/t_f$. By definition of t_{w_1}, $B_d(n') = \sum_q R_s(m'+q)$ where $-n_{w_1}t_s/t_f \leq q < 0$. In the time span $n'+l$, $B_d(n'+l)$ becomes

$$B_d(n'+l) = B_d(n') - \sum_{p=0}^{l-1} E(n'+p) + \sum_q R_s(m'+q) + \sum_{p=0}^{l-1} U_d(n'+p) \qquad (30)$$

where $n't_s/t_f \leq m'+q < (n'+l)t_s/t_f$. Since $B_d(n'+l) \geq 0$, the decoder underflow is

$$U_d(n'+l-1) = max[0, -B_d(n') + \sum_{p=0}^{l-1} E(n'+p) - \sum_q R_s(m'+q) - \sum_{p=0}^{l-2} U_d(n'+p)] \quad (31)$$

Then, the total amount of decoder underflow during St_s is

$$U_d^t = \sum_{p=0}^{S-1} U_d(n'+p) \qquad (32)$$

In order to calculate the waiting time, the target bit rate of the frame obtained by (19) can be used. Let $\bar{E}(l)$ be the target bit rate per each slice. Then, the waiting time is $t_{w_{n'+l}} = w_{n'+l}t_s$, and $w_{n'+l}$ is approximately obtained by

$$w_{n'+l} \approx \frac{B_d(n'+l)}{\bar{E}(n'+l)} \qquad (33)$$

and the average waiting time is $\bar{w}t_s$ where

$$\bar{w} = \frac{1}{S}\sum_{l=0}^{S-1} w_l. \qquad (34)$$

.

4.4. Examples using the buffer underflow curves

For simple description, we set $t_f = t_s$. In Fig. 7 (a), two solid lines represent the encoder and decoder underflow curves which indicate the amount of accumulative encoded and decoded data respectively over the time index p: $\sum_p E(n+p)$ and $\sum_p E(n+p+d_1+w_1)$ where $n = 0$. Note that the decoder buffer underflow curve is a shifted version of the encoder buffer underflow curve by the time interval $d_1 + w_1$ where $d_1 = 3/2$ and $w_1 = 3/2$.

Between two curves, there is a gray solid curve which represents the amount of accumulative transmission data: $\sum_q R_s(m+q)$ where $m = 0$ and $p = q$ due to $t_f = t_s$. If there is no encoder buffer underflow, the curve indicates both the accumulative bandwidth and transmission data, i.e., $\sum_p R_s(p) = \sum_p \hat{R}_s(p)$. When $\sum_p \hat{R}_s(p) > \sum_p E(p)$, a dot curve $\sum_p \hat{R}_s(p)$ starts splitting from the gray solid curve $\sum_p R_s(p)$ due to the encoder buffer underflow $\sum_p U_e(p)$: $\sum_p \hat{R}_s(p) = \sum_p R_s(p) + \sum_p U_e(p)$. Eventually, the value $\sum_p U_e(p)$ becomes the total unused bandwidth. In the figure, a constant bandwidth is allocated at regular time intervals so that the slope of $\sum_p \hat{R}_s(p)$ is a constant.

For a given $t_{q_1} = d_1 t_f$, the minimum queuing delay d_1^* needed to prevent the encoder buffer underflow is obtained by :

$$d_1^* = \text{argmin}[\ d_1\ ] \ \text{ for } U_e^t = 0. \qquad (35)$$

Similarly, the initial waiting time $w_1 t_s$ required to avoid the decoder buffer underflow is

$$w_1^* = \text{argmin } [\ w_1\ ] \text{ for } U_d^t = 0. \tag{36}$$

Fig. 7 (b) shows how to obtain d_1^* and w_1^*. Given $\sum_p E(p)$, the curve $\sum_q \hat{R}_s(d_1+q)$ is shifted to the right (d_1 is increased from zero) until it is not overlapped with the encoder buffer underflow curve. Then, the first index occurring no overlapping becomes d_1^*. Similarly, w_1^* can be obtained by increasing w_1 from zero until the decoder underflow curve $\sum_p E(p+d_1^*+w_1)$ is not overlapped with the curve $\sum_q \hat{R}_s(d_1+q)$.

4.5. Channel adaptation of foveated video

In Section 2.8, it is described that the queuing response to fading channel statistics depends on the following factors: channel correlation, mobile speed, mobile driving pattern and channel speed. The time-variant channel charateristics may lead to the abrupt fluctuation of the queuing delay at the encoder buffer. In order to ameliorate such fluctuation, video encoder controls the traffic by using skip frames or changing quantization parameters.

Foveated video has a high potential of channel adaptation capability. Suppose that a video sequence is compressed in the range $100 - 1000$ *kbps* using quantization parameters {1,...,31}. If the visual redundancy is reduced up to 40 % by foveation filtering, the range of the transmission rate can be scaled down to $60 - 600$ *kbps*. Therefore, it is possible to adjust to time-varying channel dynamics more adaptively at low bit rate transmission. Fig. 7 (c) and (d) depict a channel adaptation of regular and foveated videos. When the buffer occupancy is increased, the encoder reduces the encoded data rate by using a rate control algorithm. However, due to the bound of the compression rate, the encoder buffer fullness may reach more than a threshold T_h (18). Then, the encoder adjusts to the channel capacity by using skip frames as shown in Fig. 7 (c). On the other hand, Fig. 7 (d) shows that the foveated video dynamically adapts to the lower channel capacity by reducing the coding rate without using skip frames.

In the foveated video, the waiting time can be also reduced due to the low range of the traffic generation rate. Fig. 7 (d) shows that the initial waiting time is reduced to $w_1^* = 1$ from $w_1^* = 2$ in Fig. 7 (c).

5. SIMULATION RESULTS

5.1. Traffic and channel modeling

In this simulation, three different video traffics are generated according to the compression gain defined in (4): $G_c^t = 0$ for the regular video and $G_c^t = 0.2$ and 0.4 for the foveated video. Based on the definition, the traffics have the same visual quality with the traffic rate ratio $(1 : 0.8 : 0.6)$. Each traffic uses a target frame rate of 10 (frames/sec.) and each frame consists of 18 slices.

To demonstrate the efficacy of the foveated video, the average queuing delay $\bar{d}$ in (28) and the number of skip frames are measured over normalized channel throughput. Fig. 8 (a) shows that $\bar{d}$ is effectively reduced relative to the regular video when the channel throughput decreases and the coding gain increases. When the channel throughput is 0.2, $\bar{d}$ is decreased up to 250 *msec* by using $G_c^t = 0.4$ compared to $G_c^t = 0$. In addition, the relative difference of skip frames is reduced at the lower channel throughput as shown in Fig. 8 (b). When the throughput is

0.6, the number of skip frames can be reduced up to 40 % and 15 % at $G_c^t = 0.4$ and 0.2 relative to $G_c^t = 0$.

The average waiting time $\bar{w}$ defined in (34) can be also reduced as the compression gain is increased. At the throughput 0.2, the waiting time is reduced from 325 $msec$ ($G_c^t = 0$) to 225 $msec$ ($G_c^t = 0.4$). As the channel throughput is increased, the relative difference of $\bar{w}$ is lower than that of $\bar{d}$. The major reason is that the encoder controls the traffic on a frame-by-frame basis and the decoder extracts the data from the decoder buffer on a slice-by-slice basis. Fig. 8 (d) shows the end-to-end delay by adding the components of $\bar{d}$ and $\bar{w}$. Fig. 8 (e) and (f) show $\bar{w}$ and the end-to-end delay at $w_1 t_s = 100$ $msec$.

5.2. Real fading channel in the downtown area of Austin

One of the most important factors that distort transmitted signal over wireless channels is called the fading effect. Real fading statistics collected in the downtown area of Austin, Texas at 1.9 GHz are used in our simulations [13]. Fig. 2 (b) shows a simplified map in the downtown Austin area. The base station is at the corner of Congress and 7th street with an 18 meter high transmitter mounted on top of the van. In order to obtain continuous wave data, the car was driven on every street and the continuous wave signal was measured at 1 msec interval. The size of the area covered by the measurement is approximately 400 m in diameter.

In wireless mobile channels, a signal delivered to a receiver has two types of fading effects : large-scale (shadowing) and Rayleigh (multi-path) fading [29][30]. Since large-scale fading is due to terrain contours (such as the structure of buildings in downtown), the original signal is attenuated by low frequency signal distortion. On the other hand, the signal amplitude and phase attenuation due to Rayleigh fading results in high frequency signal distortion. In Fig. 9 (a), we observe two types of fading effects occuring simultaneously.

5.2.1. Simulation environment

For H.263 video, we use 51.1 kbps for the target transmission rate and 10 frames/sec for the target frame rate. The CIF "News", "Mobile", and "Akiyo" image sequences are used, where each sequence consists of 300 frames. In the TDMA system, the time frame rate is $t_f = 10$ msec and the packet size is $L_p = 511$ bits. We assume that each packet contains source and channel coding bits without other header information. For the ECC scheme, we use a BCH(511,484) code for channel SNRs ranging from 5-15 dB, and a concatenated code (1/2 convolutional and BCH(255, 215) codes) for channel SNRs less than 5 dB.

We regard the downtown area of Austin as a single square cell and extend the cell into imaginary infinite cells, where each cell is assumed to have the same channel attenuation measured in the downtown area. Thus, the channel characteristics between adjacent cells are mirrored with respect to the cell boundaries. On this imaginary cellular system, we consider two driving patterns. One is directional driving and the other is random driving. Let P_f, P_r and P_l be the probability to go straight, turn right and turn left at the cross section of the street. We set $P_f = 0.8$, $P_r = 0.1$, $P_l = 0.1$ for directional driving and $P_f = P_r = P_l = 1/3$ for random driving.

5.2.2. Visual quality/transmission delay : foveated vs. regular

Fig. 10 (a) shows the reconstructed I picture of the regular video where the number of generated bits is 402 kbits, PSNR = 31.03 dB and FPSNR = 29.98 dB. In (b), the reconstructed I-picture of the foveated video (the foveation point is marked by a small "x") is shown, where the number of generated bits is 299 Kbits, PSNR = 32.94 dB and FPSNR = 31.03 dB. Note that the FPSNR in (b) is equal to the PSNR in (a). Using the equivalent 402 kbits in (a), the foveated image is compressed as shown in (c) where PSNR = 35.13 dB and FPSNR = 35.34 dB. For the given foveation point, it can be seen that the visual quality of (a) and (b) are similar while the visual quality of (c) is much improved relative to the regular video at the equivalent rate, provided that the foveation point is fixated upon. In (d), we measure the transmission delay corresponding to the traffic (a), (b) and (c) over noiseless binary channels and for constant transmission rates. The transmission delay increases as the transmission rate is reduced. When the transmission rate is less than 100 Kbits/sec, the transmission delay in the foveated image (b) can be reduced by over 100 msec compared to other methods.

5.2.3. Channel throughput

Fig. 9 shows the normalized channel throughput $\bar{R}(m)/L_p$ corresponding to the channel SNR for a 20-minute drive. The channel SNR is measured at each packet transmission time. In (a), the SNR time series $SNR(m)$ ($SNR_o = 15$ dB) is shown. After extracting the large-scale fading signal using (13), we measure the channel throughput from the original fading and the large-scale fading in (b). The amplitude of the channel throughput for the original fading is fluctuates considerably due to the high frequency components. On the other hand, the channel throughput for the large-scale fading shows a high correlation and is attenuated or increased for a longer time. In Fig. 9 (c) and (d), the amplitude of $E[\bar{R}(m)/L_p]$ is shown for the channel SNR, the fading signal, the ECC method, and the driving pattern. Using the ECC, the channel redundancy can be chosen at each packet (every 10 msec) by (11). Then, the channel throughput becomes the upper bound of other methods, where the channel redundancy is updated every 100 msec (the target frame rate) using (14). The solid line shows the channel throughput for the above case. Even if the throughput is less than the upper bound, it is higher than that of other non-ECC methods over the widest range of the channel SNR. It also shows that the average channel throughput (the first-order statistics) are similar over the large-scale fading and the original fading. However, note that the Rayleigh fading yields a different channel correlation and queuing performance.

5.2.4. Spatial/temporal resolution

For the rate control, we employ two methods in [8]. One is *constant q* where an optimal constant QP (quantization parameter) is decided for each picture. The other is $modified\ Q\ mode$ where the optimal QPs are used for the macroblocks near foveated regions and a constant QP is used for the macroblocks whose average local bandwidth is less than a threshold.

For traffic control, two coding schemes are considered: the *channel-separated* and the *channel-adaptive* codings. In the *channel-separated* (*channel-adaptive*) coding, the target rates and the number of skip frames are decided based on the buffer fullness without (with) the channel SNR feedback information. In the

channel-adaptive coding, the target rates are controlled using (18). The *adaptive source-channel coding* combines the ECC with the *channel-adaptive* coding. In order to compare the performances of these, we use the following methods.
- Method 1 : The regular video, the *constant* q, and the *channel-separated* coding
- Method 2 : The foveated video, the *constant* q, and the *channel-separated* coding
- Method 3 : The regular video, the *modified Q mode*, and the *channel-adaptive* coding
- Method 4 : The foveated video, the *modified Q mode*, and the *channel-adaptive* coding
- Method 5 : The foveated video, the *modified Q mode*, and the *adaptive source-channel* coding

Since the channel throughput is improved by using the ECC, the source throughput (bits/sec) in *Method* 5 is the highest while the rest are all the same as shown in Fig. 11 (a). Fig. 11 (b) shows the average FPSNR of the reconstructed video. We can improve the visual quality by using foveated video, so the FPSNR in *Method* 2 and *Method* 4 is better than that in *Method* 1 and *Method* 3. Also, the visual quality is improved by using the *modified Q mode*, such that the FPSNR in *Method* 3 (*Method* 4) is better than that in *Method* 1 (*Method* 2). In [7], we observed a drift in the FPSNR resulting from high frequency reduction, depends on the video sequence. In the "Akiyo" and "News" image sequences, there is no drift. However, in the "Mobile" image sequence, there is a 2 dB FPSNR drift at low bit rates. Subtracting the average drift 0.7 dB, we find that the actual visual quality in Method 2 (Method 4) is improved to 1.1 (0.6) dB from Method 1 (Method 3) at $SNR_o = 15$ dB.

In Fig. 11 (c), it is shown that foveated video can increase temporal resolution using the same buffer control and rate control algorithms in such way that the average number of skip frames in *Method* 2 (*Method* 4) is reduced to 1 (0.42) frame relative to *Method* 1 (*Method* 3). In addition, the number of skip frames in *Method* 3 (*Method* 4) can be effectively reduced relative to *Method* 1 (*Method* 2) using feedback channel information. Since the bitrate range of foveated video is scaled down, the number of skip frames can be effectively reduced. Moreover, the lower bound of the traffic generated by the foveated video is able to be fully utilized by the temporal resolution control, adjusted to the time-varying channel dynamics. In *Method* 5, the temporal resolution is much improved at a low channel SNR due to several factors: the increasing channel throughput, the delicate bit allocation at low bit rates, and the lower coding bound of the foveated video. However, the FSNR is lower than that of *Method* 2 and *Method* 4 at a low channel SNR because of the improved temporal resolution. Fig. 11 (d) shows the improved visual quality obtained using foveated video over a part of frame index.

Fig. 12 (a) and (b) show the number of generated bits and the number of skip frames according to frame number. Using *Method* 4, we can reduce the number skip frames and obtain less bursty traffic compared to *Method* 3.

5.2.5. Buffer underflow and end-to-end transmission delay

Fig. 13 (a) and (b) shows the encoder/decoder buffer underflow over the encoding queuing delay and the decoder waiting time at $SNR_o = 15$ dB. Over mobile channels, the amount of encoder buffer underflow is much less than that of decoder buffer underflow due to the dynamic channel characteristics and the abrupt traffic

change in the encoder. In (b), it is shown that foveated video can reduce the decoder buffer underflow due to the flexible traffic control in *Method* 2 (*Method* 4) compared to *Method* 1 (*Method* 3). In addition, the ECC in *Method* 5 is able to reduce the buffer underflow by increasing the channel throughput.

In Fig. 13 (c), the end-to-end transmission delay is measured. When the average channel SNR is less than 15 dB, the source transmission delay rapidly increases, due to the abrupt source traffic compared to the narrow channel capacity. Utilizing the channel adaptation of the foveated video, we can reduce the delay relative to the regular video. At an $SNR_o = 15$ dB, the delay becomes 806 msec in *Method* 1, 521 msec in *Method* 2, and 207 msec in *Method* 5. Thus, using foveated video, we can reduce the delay up to 285 msec relative to the regular video. Also, using the ECC, the delay is relatively reduced up to 599 msec.

6. CONCLUSIONS

Foveated video compression is an optimized technology matched the human visual system and which is particularly appealing for low bit rate applications. Using foveation, it is possible to reduce the datarate while minimizing the loss of visual information. In this paper, we present an end-to-end foveated visual communication system. We measure performance in terms of visual quality and transmission delay using real fading channel statistics measured in downtown Austin, Texas. In the system, we used the following techniques: human foveation, target tracking, H.263 video, a TDMA scheme, ARQ flow control, convolutional coding, BCH coding, adaptive source and channel coding, buffer control and optimal rate control.

In order to reduce the expected distortion of reconstructed image, a maximum source throughput criterion is introduced, which is also used to decide the source and channel rates in packet networks. For the adaptive source/channel coding, the channel SNR and the average throughput are utilized to decide the channel redundancy and the source target rate. The FPSNR (foveal peak signal-to-noise ratio) is used to measure the distortion in agreement with the human visual system. The buffer underflow is calculated using the buffer underflow curves according to the slice and packet intervals. Then, the queuing delay and the waiting delay are defined. Based on these parameters, we derived the end-to-end transmission delay.

For the simulations, we consider an imaginary cellular system, where the channel characteristics of each cell are the same as downtown Austin, Texas. The channel dynamics according to the driving patterns (directional and random driving) are measured. According to the simulation results, the combined technique of foveated video, suboptimal rate control, channel adaptive coding, and adaptive ECC demonstrates improved performance in terms of spatial and temporal visual quality and transmission delay relative to regular video. At an average channel SNR of 15 dB, we improve the actual visual quality by up to 0.6 - 1.1 dB. Also, we reduce the end-to-end transmission delay by up to 285 msec and the number of skip frames in the range of 0.42 to 1 frame relative to regular video.

REFERENCES

[1] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE*, vol. 73, pp. 549–574, April 1985.

[2] T. Ebrahimi and M. Kunt, "Visual data compression for multimedia applications," *Proc. IEEE*, vol. 86, pp. 1109–1125, June 1998.

[3] P. L. Silsbee, A. C. Bovik, and D. Chen, "Visual pattern image sequence coding," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 3, pp. 291–301, Aug. 1993.

[4] J. Hartung, A. Jacquin, J. Pawlyk, J. Rosenberg, H. Okada, and P. E. Crouch, "Object-oriented H.263 compatible video coding platform for conferencing applications," *IEEE J. Selected Areas Commun.*, vol. 16, Jan. 1998.

[5] W. S. Geisler and J. S. Perry, "A real-time foveated multiresolution system for low-bandwidth video communincation," in *SPIE Proceedings*, vol. 3299, 1998.

[6] T. H. Reeves and J. A. Robinson, "Adaptive foveation of MPEG video," in *Proc. 4th ACM International Multimedia Conference*, (Boston, MA), pp. 231–241, 1996.

[7] S. Lee, M. S. Pattichis, and A. C. Bovik, "Foveated video quality assessment," *IEEE Trans. Multimedia*, vol. 4, pp. 129–132, March 2002.

[8] S. Lee, M. S. Pattichis, and A. C. Bovik, "Foveated video compression with optimal rate control," *IEEE Trans. Image Processing*, vol. 10, pp. 977–992, July 2001.

[9] S. Lee and A. C. Bovik, "Fast algorithms for foveated video processing," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 13, pp. 149–162, Feb. 2003.

[10] S. Lee, C. Podilchuk, V. Krishnan, and A. C. Bovik, "Foveation-based error resilience and unequal error protection over mobile networks," *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, vol. 34, pp. 149–166, March 2003.

[11] Z. Wang, L. Lu, and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," *IEEE Trans. Image Processing*, vol. 12, pp. 243–254, Feb. 2003.

[12] S. Lee and A. C. Bovik, "Foveated video demonstration," in *http://pineapple.ece.utexas.edu /class/Video/demo.html*, 1999.

[13] H. Ling, "Wireless channel modeling," in *http://ling0.ece.utexas.edu/comm/comms.html*, 1997.

[14] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Trans. Circuits Syst. Video Tech.*, pp. 1–11, Feb. 1996.

[15] T. S. Rappaport, *Wireless Communications*. New Jersey: Prentice Hall PTR, 1996.

[16] Y. Y. Kim and S. Li, "Modeling multipath fading channel dynamics for packet data performance analysis," *ACM wireless network Journal*, vol. 6, pp. 481 – 492, 2000.

[17] Y. Y. Kim and S. Li, "Capturing important statistics of a fast/shadowing channel for network performance analysis," *IEEE J. Selected Areas Commun.*, vol. 17, pp. 888 – 901, May 1999.

[18] E. V. H. Iun and A. K. Khandani, "Combined souce-channel coding for the transmission of still images over a code division multiple access(CDMA) channel," in *IEEE International Conf. on Commun.*, pp. 164–168, 1997.

[19] H. Jafarkhani, P. Ligdas, and N. Farvardin, "Adaptive rate allocation in a joint source/channel coding framework for wireless channels," in *Proc. IEEE VTC'96*, pp. 492–496, April 1996.

[20] W. C. Y. Lee, "Estimate of local average power of a mobile radio signal," *IEEE Trans. Vehicular Technology*, vol. VT-34, pp. 22 – 27, Feb. 1985.

[21] W. C. Y. Lee, "Elements of cellular mobile radio systems," *IEEE Trans. Vehicular Technology*, vol. VT-35, pp. 48 – 56, May 1986.

[22] W. C. Jakes, *Macrowave Mobile Communications*. IEEE Press, 1974.

[23] "Video codec test model, TMN7," tech. rep., ITU-T/SG15, Feb. 1997.

[24] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 9, pp. 172–185, Feb. 1999.

[25] S. Lee, M. S. Pattichis, and A. C. Bovik, "Rate control for foveated MPEG/H.263 video," in *Proc. IEEE ICIP'98*, vol. 2, (Chicago), pp. 365–369, Oct. 1998.

[26] J. D. Salehi, Z. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Trans. on Networking*, vol. 6, pp. 397 – 410, Aug. 1998.

[27] J. Rexford, J. D. S. Sen, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online smoothing of live, variable-bit-rate video in an internetwork," in *Proc. International smoothing techniques for the transmission of prerecorded compressed video*, May 1997.

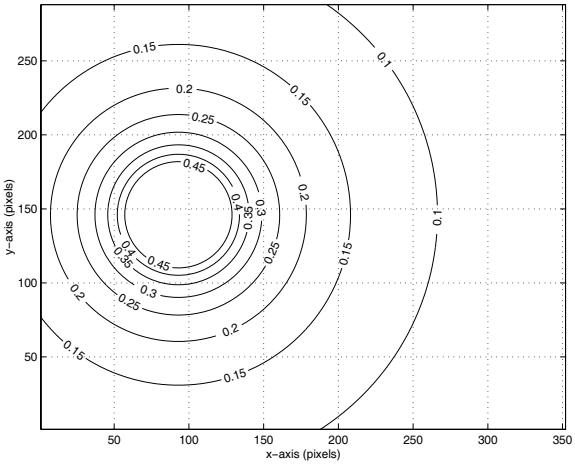
[28] S. Lee and A. C. Bovik, "Maximally flat bandwidth allocation for variable bit rate video," in *Proc. IEEE Int'l. Conf. Image Proc.*, vol. 2, (Chicago), pp. 346–350, Oct. 1998.

[29] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part I:characterization," *IEEE Commun. Mag.*, pp. 90 – 100, July 1997.

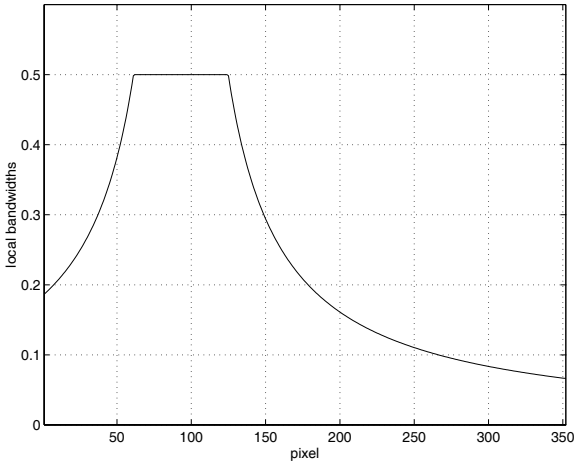
[30] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part II:mitigation," *IEEE Commun. Mag.*, pp. 102 – 109, July 1997.

(a) Foveated "News" image in cartesian coordinates

(b) Foveated "News" image in curvilinear coordinates

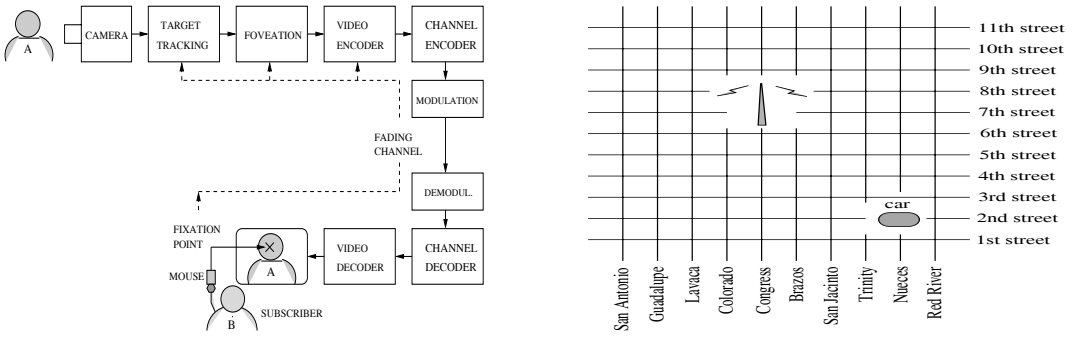


(c) Local bandwidth over the plain figure



(d) Local bandwidth from the side view at the foveation point

FIG. 1 Foveated image over cartesian and curvilinear coordinates, and local bandwidth



(a) Foveated visual communications (b) Signal attenuation measurement

FIG. 2 End-to-end system block diagram and over real fading channels in the downtown area of Austin, Texas

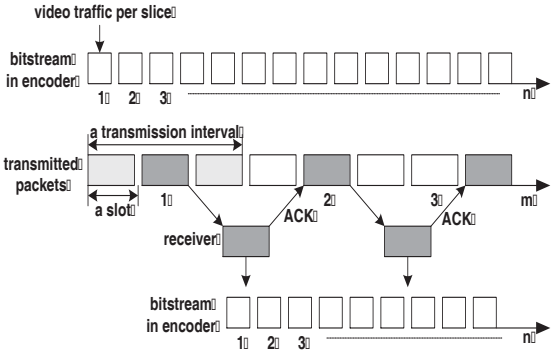


FIG. 3 TDMA system associated with stop-and-wait ARQ flow control

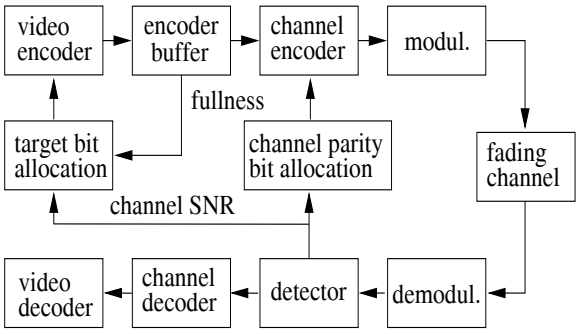
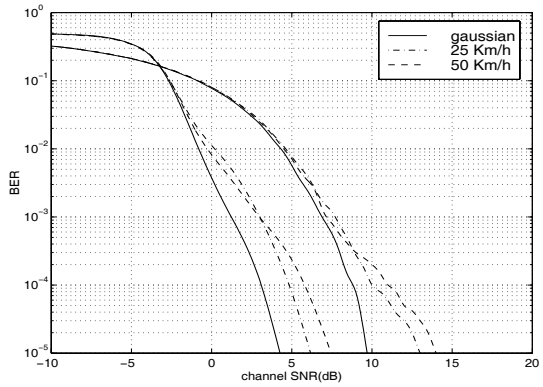
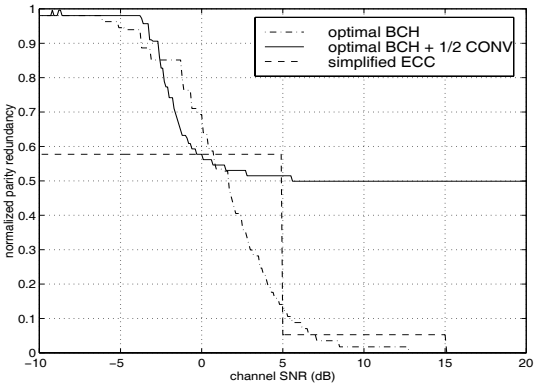


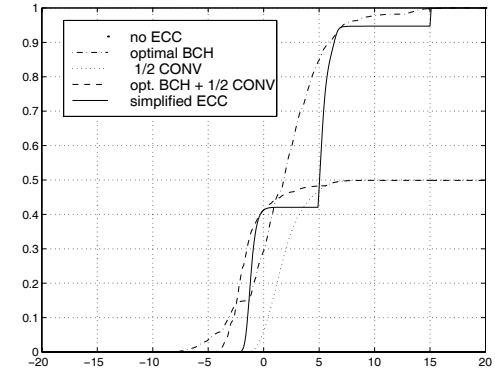
FIG. 4 Adaptive source and channel coding in system level



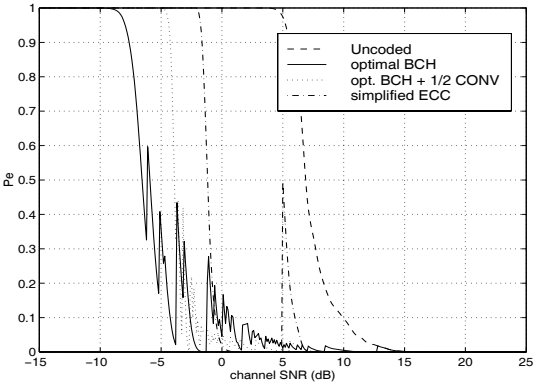
(a) The pair of BER : channel coding (1/2 conv., low BER) v.s. non-channel coding(high BER)



(b) Normalized redundancy of bits

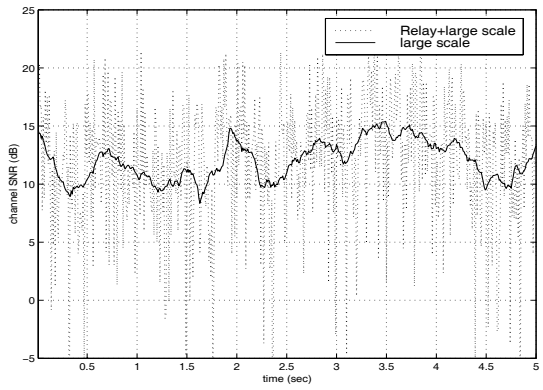


(c) Normalized source throughput

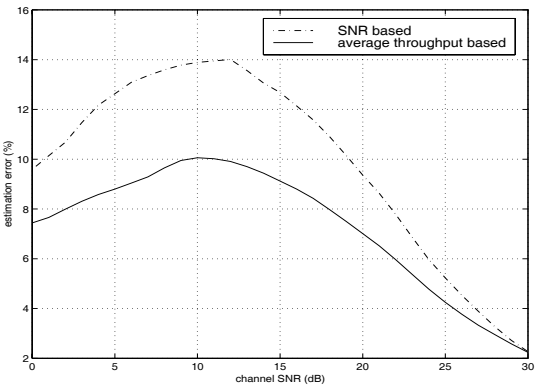


(d) Packet error probability

FIG. 5 Transmission performance according to the ECC methods and channel SNR where (b), (c) and (d) are measured at mobile speed 50 Km/hr

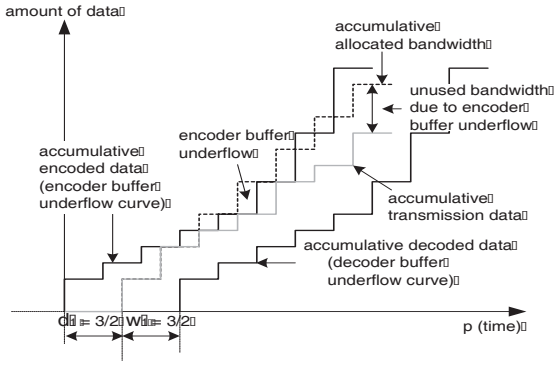


(a) Signal decomposition

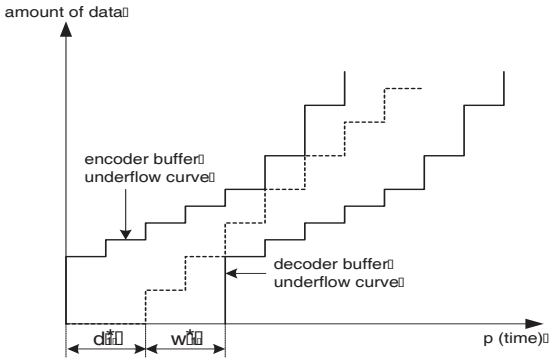


(b) Estimation error

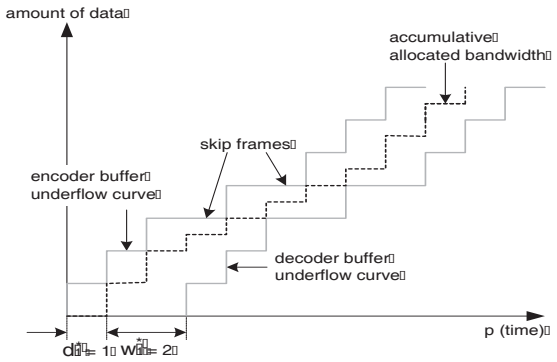
FIG. 6 Signal decomposition and channel bandwidth estimation



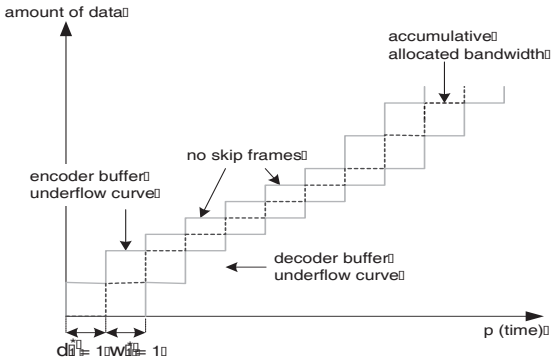
(a) An example of buffer underflow curves



(b) Optimal initial queuing delay and waiting time

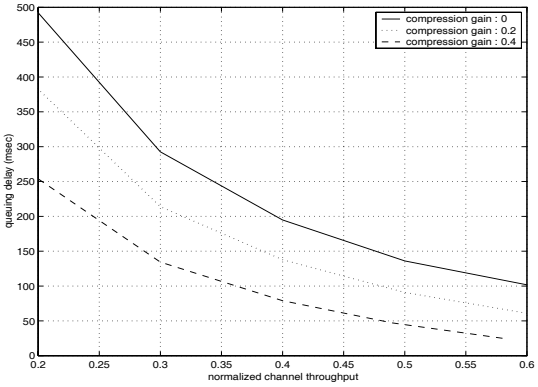


(c) Channel adaptation of regular video

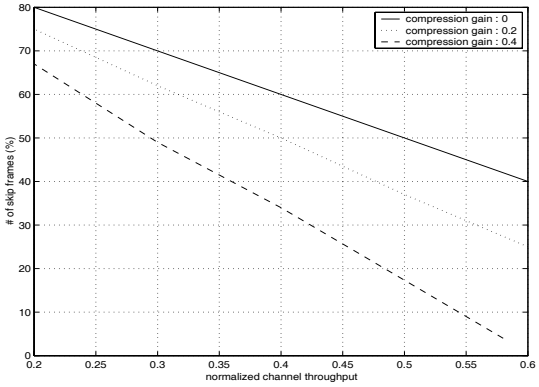


(d) Channel adaptation of foveated video

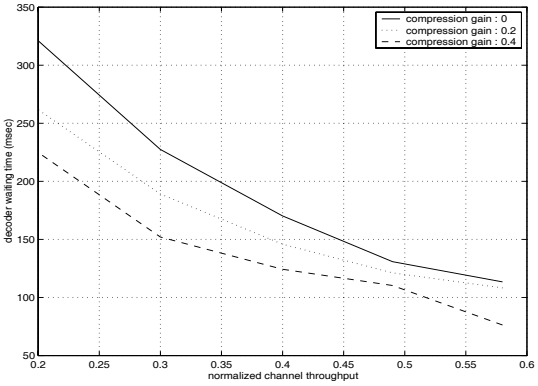
FIG. 7 Performance measurement using the buffer underflow curve



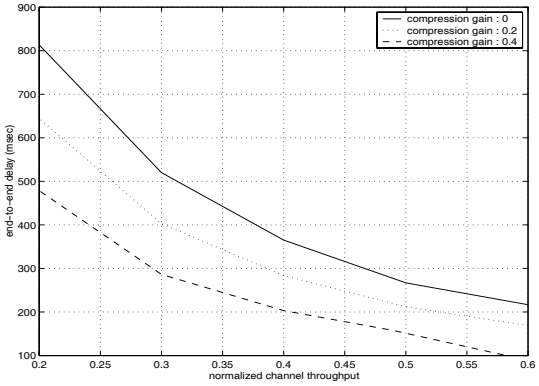
(a) The average queuing delay ($msec$) over normalized channel throughput



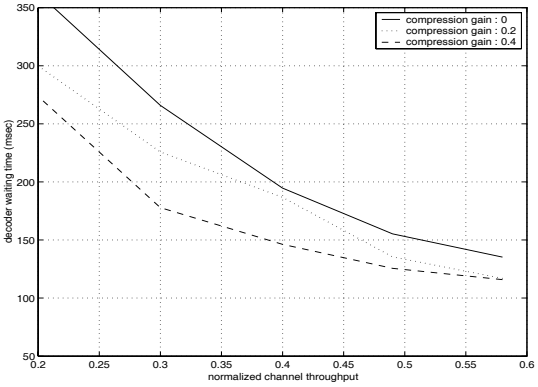
(b) The number of skip frames (%) over normalized channel throughput



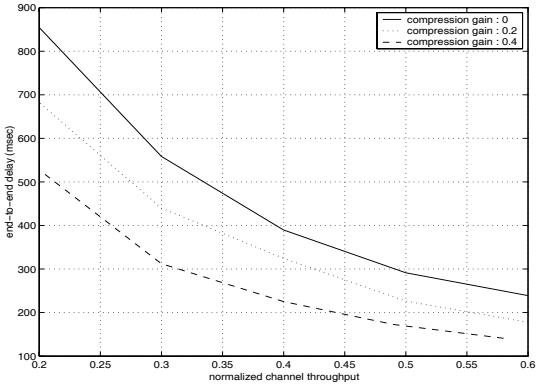
(c) The average decoder waiting time ($msec$) when $w_1 t_s = 0$



(d) The end-to-end delay when $w_1 t_s = 0$ ($msec$)

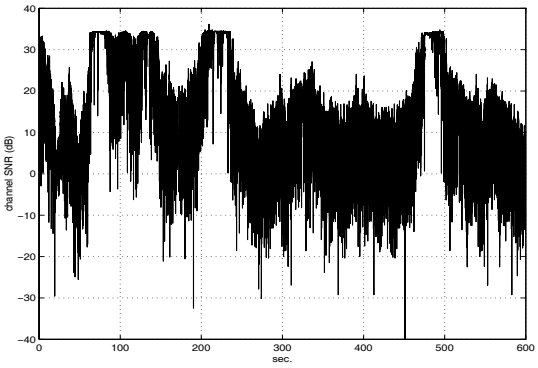


(e) The average decoder waiting time ($msec$) when $w_1 t_s = 100$

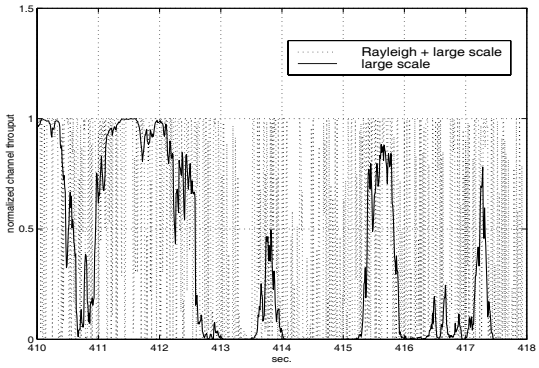


(f) The end-to-end delay when $w_1 t_s = 100$ ($msec$)

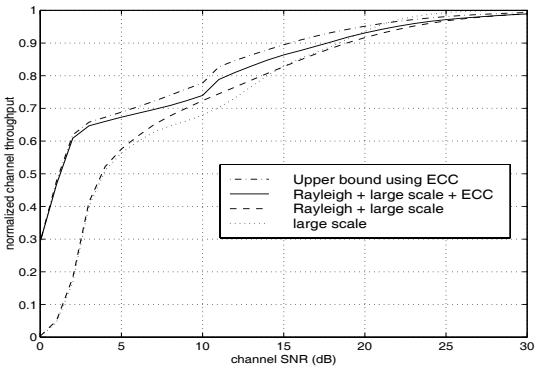
FIG. 8 Performance measurement according to the compression gain



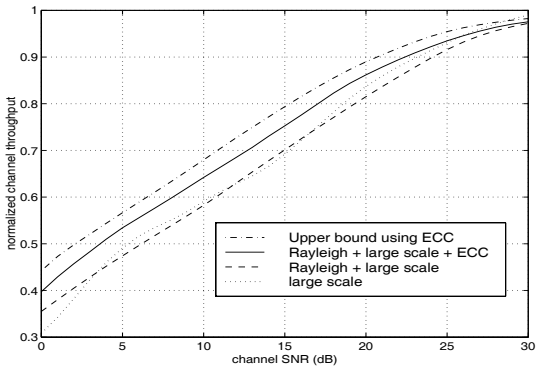
(a) Signal attenuation : $SNR_o = 15$ dB



(b) Throughput comparison



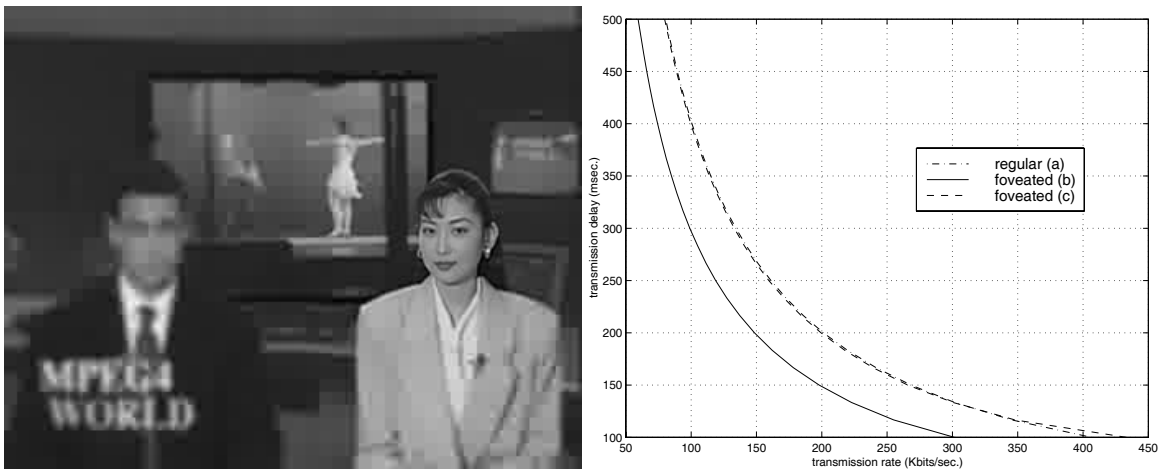
(c) Throughput for directional driving



(d) Throughput for random driving

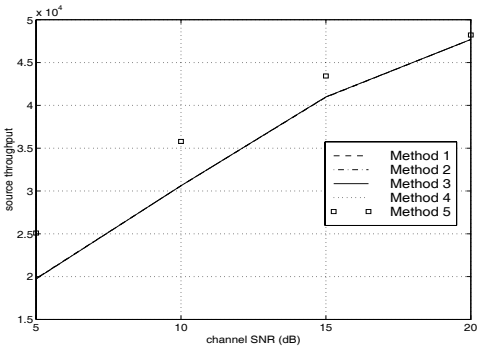
FIG. 9 Normalized throughput for 20 minute driving

(a) PSNR=31.03, FPSNR=29.98, 402 Kbits (b) PSNR=32.94, FPSNR=31.03, 299 Kbits

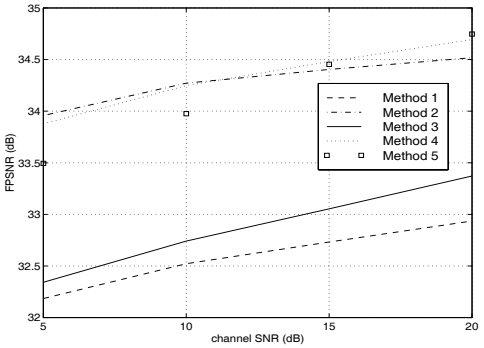


(c) PSNR=35.13, FPSNR=35.34, 402 Kbits (d) Transmission delay v.s. transmission rate

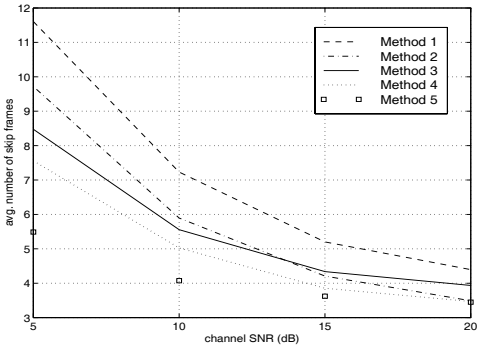
FIG. 10 Foveated v.s. regular coding : (a) regular I picture, (b) and (c) foveated I pictures.



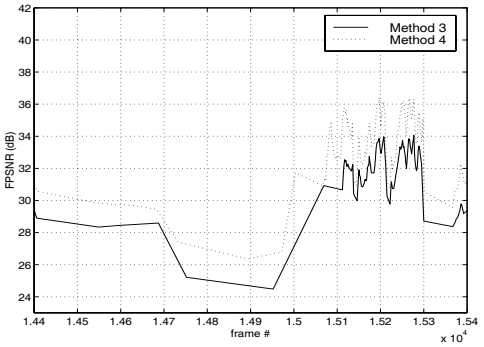
(a) Source throughput



(b) The average FPSNR

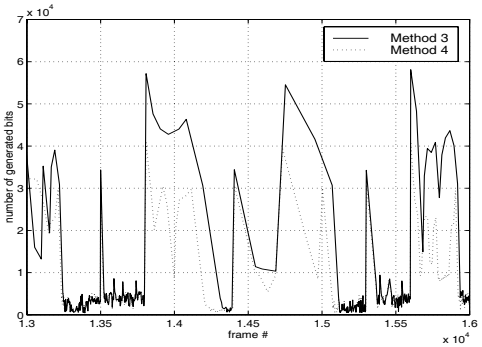


(c) The average number of skip frames

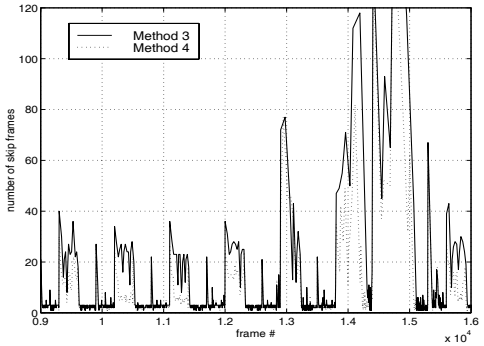


(d) The FPSNR according to frames

FIG. 11 Performance comparison according to the four methods over 10 minute random driving

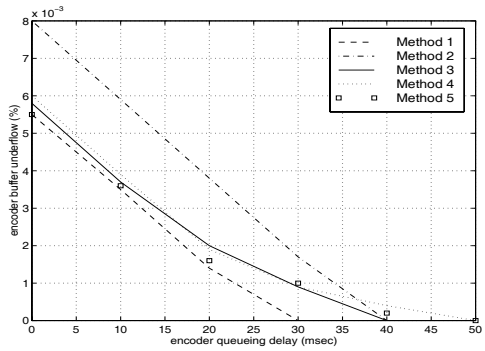


(a) The number of generated bits

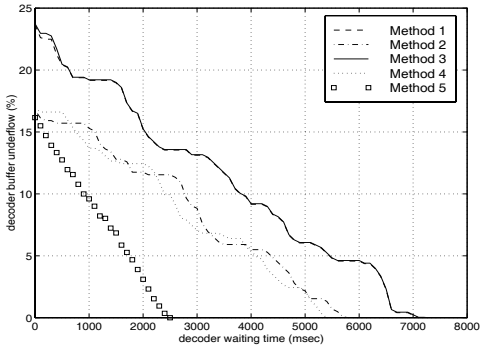


(b) The number of skip frames

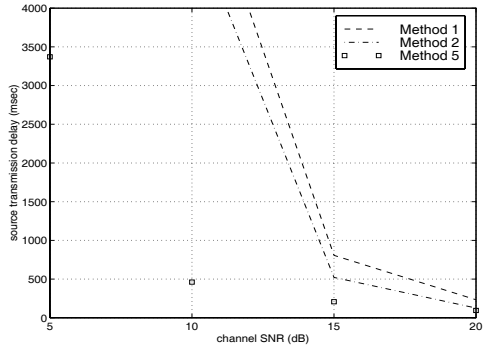
FIG. 12 Foveated v.s. regular video : the number of generated bits and skip frames



(a) Encoder buffer underflow (%)



(b) Decoder buffer underflow (%)



(c) End-to-end delay (msec)

FIG. 13 Encoder and decoder buffer underflow at $SNR_o = 15$ dB, and source transmission delay