Layered Soft Video Broadcast for Heterogeneous Receivers

Xiaopeng Fan, *Member, IEEE*, Ruiqin Xiong, *Member, IEEE*, Debin Zhao, *Member, IEEE*, and Feng Wu, *Fellow, IEEE*

***Abstract*—Wireless video broadcast poses a challenge to the conventional visual communication in providing simultaneously each receiver the best video quality under its channel condition. Soft video broadcast, as a newly emerged wireless video broadcast scheme, is able to accommodate multiple receivers of different channel SNRs. However, the current soft video broadcast frameworks such as SoftCast require the bandwidth of the wireless channel to match the number of video coefficients per second. When the channel bandwidth is larger, the existing frameworks become not very efficient in bandwidth expansion. More importantly, it is possible that the users in broadcast applications have different bandwidths. However, none of the existing soft video broadcast frameworks considers bandwidth heterogeneity. In this paper, we propose a soft video broadcast framework, called LayerCast, which can simultaneously accommodate heterogeneous users with diverse SNRs and diverse bandwidths. The bandwidth expansion problem is solved by applying layered coset coding. More importantly, we derive a globally optimal power allocation between layers and, within each layer, between each DCT chunk. In simulations, the proposed framework outperforms SoftCast of up to 4 dB in video PSNR, and outperforms H.264-based framework up to 8 dB in broadcast.**

***Index Terms*—Channel bandwidth, coset coding, power allocation, soft video broadcast, softcast.**

I. Introduction

WIRELESS video broadcasting is a popular application aiming to transmit video signal simultaneously to multiple users of possibly different channel characteristics. The main challenge is the difficulty involved in fully utilizing each user's channel capacity and providing each user with the best video quality under his channel condition. The conventional digital video broadcasting (DVB) standard [1] and the 802.11 standard [2] can hardly accommodate diverse users in broadcast due to the stair effect: the server should encode the video source at one rate. However, if some users have too bad a channel condition to support this data rate, they cannot decode the video correctly; while if some users have much better channel condition, their reconstruction quality will not be improved accordingly. A typical approach to mitigating this stair effect is scalable video coding (SVC) [3], [4]. With SVC, users with bad channel condition can receive rough video signals while users with good channel condition can receive high-quality video signals. However, SVC decreases the compression efficiency of video signals. In addition, SVC cannot entirely mitigate the stair effect, but divides one big stair into two or three small stairs.

Recently, a novel soft video broadcast framework, called SoftCast, has been proposed for wireless video broadcasting [5], [6]. Different from conventional frameworks, SoftCast simply applies linear transforms on the video signal, and transmits the transform coefficients directly in a wireless channel without quantization, entropy coding, and channel coding. This makes the magnitude of the transmitted wireless signal proportional to the DCT coefficients of video frames. Since the channel noise is translated into a small perturbation in DCT coefficients, SoftCast allows graceful degradation with increasing noise. A high SNR user can automatically get high-quality video while a low SNR user can also decode low-quality video. In wireless broadcasting application, SoftCast achieves significant gain over the conventional DVB framework [6].

Several soft video frameworks have been proposed recently for more functionality and better performance. Aditya and Katti [7] proposed a unicast framework called Flexcast. It does not have entropy coding, but adopts rateless channel coding to encode and transmit DCT coefficients. Hence, it can adapt to channel variation with the capability of quality control. Liu *et al.* [8] proposed to achieve receive antenna heterogeneity in MIMO systems using compressive sensing. We proposed a distributed soft video broadcast framework called DCast [9], [10] based on distributed video coding [11]. We also proposed a soft video broadcast framework called WaveCast [12] based on 3-D wavelet transform [13]–[15]. Peng *et al.* [16] and Wu *et al.* [17] proposed another distributed coding system for satellite image transmission. Yu *et al.* [18] proposed a hybrid digital-analog (HDA) framework with a base layer coded by H.264 [19] and an enhance layer coded like SoftCast. Xiong *et al.* [20] proposed a gradient-based framework for perception-friendly wireless soft video broadcast. Wang *et al.* [21] proposed a wireless soft video broadcast

Manuscript received July 14, 2014; revised October 15, 2014 and December 11, 2014; accepted February 4, 2015. Date of publication February 11, 2015; date of current version October 28, 2015. This work was supported in part by the Major State Basic Research Development Program (973 Program) of China under Grant 2015CB351804; in part by the National Science Foundation of China under Grants 61472101, 61272386, and 61390513; in part by the Program for New Century Excellent Talents in University of China under Grant NCET-11-0797; and in part by the Fundamental Research Funds for the Central Universities under Grant HIT.BRETIII.201221. This paper was recommended by Associate Editor E. Steinbach.

X. Fan and D. Zhao are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: fxp@hit.edu.cn).

R. Xiong is with Peking University, Beijing 100871, China.

F. Wu is with University of Science and Technology of China, Hefei 230000, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2402831

framework based on compressive sensing. Cui *et al.* [22] investigate soft video broadcast over a wireless fast fading channel.

Although there are already many soft video broadcast frameworks, most of them focus on a special case that the channel bandwidth matches the number of DCT/Wavelet coefficients to transmit per second. However, the channel bandwidth may be smaller or larger than the video requirement in real applications. It was proposed in [6] to transmit each coefficient multiple times when the channel bandwidth is larger than the video requirement and to skip some DCT coefficients otherwise. Although this retransmission scheme can estimate each coefficient based on its multiple observations at the decoder, its coding efficiency is not high because the power is partially wasted in transmitting redundant information. This is the so-called bandwidth expansion problem mentioned in [6]. To solve this problem, it is desirable to remove the redundant part and keep only the residual part, such that the magnitude of the value to transmit is reduced. Under the same transmission power, this will improve the coding gain and the precision.

More importantly, it is possible that the receivers in broadcast applications have different bandwidths. For example, the server can allow VIP users to access full bandwidth, but limit the other users to access partial bandwidths by encryption. Another example is that, some mobile receivers with a small screen size may only want to access partial bandwidths and get low-quality video, for longer battery life. However, none of the existing soft video broadcast frameworks considers bandwidth heterogeneity.

In this paper, we propose a layered soft video broadcast framework called LayerCast which can accommodate simultaneously heterogeneous users with diverse SNRs and diverse bandwidths. Each DCT chunk is coded into several layers of chunks by applying layered coset coding. The base layer coset chunk is enough to reconstruct a low-quality DCT chunk for narrow band users, while each enhancement layer provides refinement information of the DCT chunk for wide-band users. In a preliminary work of this paper, we have shown that our DCast framework [23] can utilize extra bandwidth more efficiently than SoftCast, although our DCast is originally designed for motion compensation. However, our DCast is a single-layer framework without the consideration of bandwidth heterogeneity. To extend it to multilayer framework and deal with bandwidth heterogeneity, the key problem that we need to solve is the optimal power allocation across different layers and different DCT chunks.

The key technical contribution in this paper is the global power-distortion optimization (PDO) including the optimal coset quantization and the optimal coset power allocation. Although optimal power allocation has been investigated in both SoftCast [6] and our DCast [9], [10], [23], the power allocation problem in this paper is more complicated. Since the proposed LayerCast is a layered framework, the transmission power needs to be allocated among not only DCT chunks but also layers. The result in this paper shows that it is nonoptimal to use the power allocation of SoftCast or DCast in a multilayer situation. Therefore, we have derived the globally optimal power allocation across DCT chunks and layers in this paper.

The rest of this paper is organized as follows. Section II introduces related works. Section III presents the proposed LayerCast framework. Section IV provides the solution of PDO. Section V presents simulation results and Section VI concludes this paper.

II. Related Works

A. Joint Source and Channel Coding

A conventional single-layer video broadcast framework with separated video coding and channel coding suffers from a sharp threshold effect for users with diverse channel conditions. In contrast, a layered video broadcast framework with joint source and channel coding (JSCC) can provide a stepwise graceful degradation and improve the behavior in terms of coverage and robustness of the transmission scheme [24]. The method to generate layered video bits is called SVC [3], [4]. SVC generates layered video bitstreams with different levels of video quality to accommodate heterogeneous receivers.

Bit allocation and rate distortion optimization (RDO) are two typical techniques in JSCC. For single-layer frameworks, Bystrom and Modestino [25] proposed bit allocation between source coding and channel coding to balance the video quality and the error protection capability. For layered JSCC frameworks, the bit allocation is not only between source and channel coding, but also between layers [26], [27]. Based on rate distortion analysis, He *et al.* [28] proposed a JSCC framework that optimizes the mode selection and rate control of video coding according to the channel errors. Zhai *et al.* [29] proposed to optimize jointly the available error control components. For broadcast/multicast, the optimization should not consider the quality of one particular receiver only. Therefore, Ji *et al.* [30] proposed to optimize for the overall receiving quality of the heterogeneous QoS receivers by modeling the layered video broadcast as an aggregate utility achieving problem. Besides the transmission power, the energy consumption of some mobile receivers is also critical. Therefore, Singhal *et al.* [31] proposed to optimize jointly for video distortion and energy consumption.

Distributed source coding (DSC) [32] is an alternative method to JSCC. The first attempt to implement JSCC by DSC is the layered coding scheme in [33], where the enhancement layer uses Raptor code for both video refinement and data protection. In another frame-based JSCC scheme [34], the functionality of both video compression and channel coding are achieved universally by one error correction code.

Although the layered JSCC framework can provide a stepwise graceful degradation, it cannot fully mitigate the stair effect. In addition, the more layers we have, the less compression efficiency we will achieve in SVC.

B. Soft Video Broadcast

The first soft video broadcast framework is SoftCast, which is a simple but comprehensive design for wireless video multicast, covering the functionality of video compression, data protection, and transmission in one scheme [6].

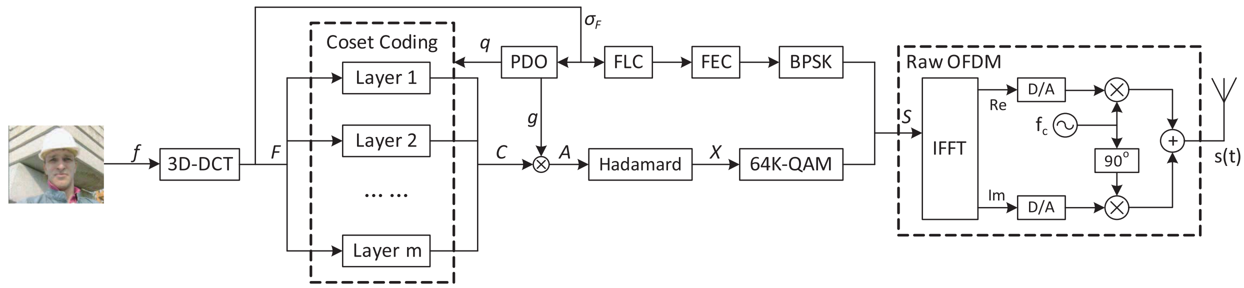


Fig. 1. LayerCast server.

The SoftCast encoder consists of the following steps: DCT transform, power allocation, Hadamard transform, and direct dense modulation. Transform removes the spatial redundancy of a video frame. Power allocation minimizes the total distortion by optimally scaling the transform coefficients. Hadamard transform is in some sense a precoding to make packets with equal power and equal importance. After that, the data is directly mapped into wireless symbols by a very dense quadratic-amplitude modulation (QAM). The decoder uses linear least square estimator to reconstruct the signal. Almost all the steps in SoftCast are linear operations and thus the channel noise is directly transformed into reconstruction noise of the video. Therefore, SoftCast is asymptotically robust in the sense that each user can get the visual quality matching his channel SNR.

One limitation of SoftCast is that it requires the bandwidth of the wireless channel to match the number of video coefficients to transmit per second. When the channel bandwidth is larger than the video requirement, SoftCast transmits each coefficient multiple times and lets the decoder estimate the coefficient based on these multiple samples. This retransmission scheme is not very efficient because the power is partially wasted in transmitting redundant information. However, to exploit the redundancy between these multiple samples is not straightforward due to the channel noise. This is the so-called bandwidth expansion problem mentioned in [6]. Till now, there is no efficient solution for this problem and this limits the potential application of soft video broadcast. The intuition of this paper is to remove the redundant part and keep only the residual part, such that the magnitude of the value to transmit is reduced. Under the same transmission power, this means higher gain and higher precision.

The proposed LayerCast solved the bandwidth expansion problem by layered coset coding. The coset quantization step sizes are optimally derived, such that each layer of coset chunk contains exclusive refinement information of the original video signal. This exploits the redundancy because each layer only contains refinement information. Recently, we also notice that Kochman and Zamir [35] have studied the utilization of coset coding in the Wyner-Ziv Dirty-Paper problem and proved its optimality and asymptotical robustness in multicast application. It can be considered in general as the theoretical foundation to support the proposed LayerCast.

C. Power-Distortion Optimization

In contrast to conventional video coding frameworks, soft video broadcast framework does not have bit rate as well as RDO. Instead, soft video broadcast frameworks can be optimized by allocating transmission power, through PDO.

In soft video broadcast, PDO has a big impact on the coding performance [36]. Jakubczak and Katabi [6] derived the optimal power allocation between different DCT chunks for SoftCast: Let F_i be the ith DCT chunks. Let P be the total power, and P_i be the power allocated to the ith chunk. The optimal power allocation can be approximately achieved by

$$P_i \approx \frac{\sigma_{F_i}}{\sum_{i=1}^{n} \sigma_{F_i}} P \tag{1}$$

where σ_{F_i} is the standard deviation of F_i and n is the total number of significant DCT chunks.

Different soft video broadcast frameworks may have different PDO. Liu *et al.* [37] solved the optimal source-channel mapping and power allocation problem in the MIMO system. Xiong *et al.* [36] analyzed the gain of transform in SoftCast with different power allocation schemes. Xiong *et al.* [38] also proposed an adaptive chunk division method and corresponding power allocation. We solved the optimal power allocation between motion data and residue data for DCast [10]. Yu *et al.* [18] derived the optimal power allocation between digital and analog parts of their HDA framework. Cui *et al.* [22] derived the optimal power allocation and channel allocation for soft video broadcast over a wireless fast fading channel.

The proposed LayerCast is a layered framework. Therefore, the transmission power needs to be optimally allocated among each layer and each DCT chunk. In this paper, we have solved this problem and achieved a global optimal power allocation. In this global solution, each layer gets equal transmission power regardless of the number of layers. However, the power allocated to each DCT chunk is related to the number of layers. Nevertheless, our solution reduces to the one in [6] when there is only one layer.

III. Proposed Framework

The proposed LayerCast framework is a wireless soft video broadcast framework based on layered coset coding. Soft transmission and layered coset coding enable LayerCast to accommodate simultaneously heterogeneous users with diverse channel SNRs and diverse bandwidths.

Fig. 1 shows the framework of a LayerCast server. First, LayerCast transforms the input video signal by 3D-DCT. The DCT coefficients are coded into multiple layers by coset module for bandwidth expansion and then scaled for PDO.

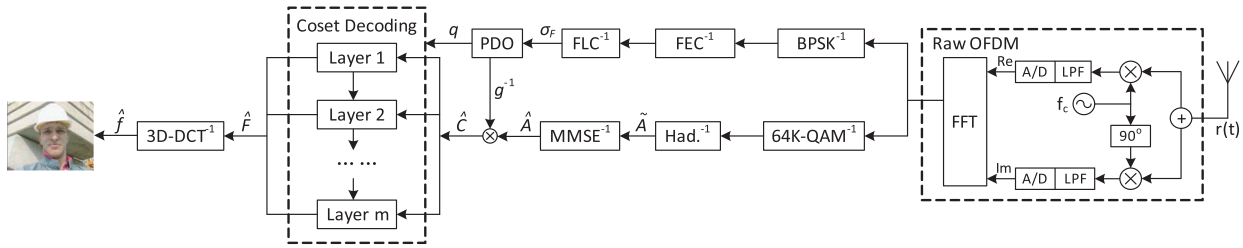


Fig. 2. LayerCast client.

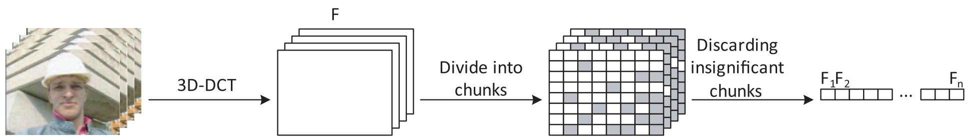


Fig. 3. Preprocessing before coset coding.

The coset coded coefficients of each layer are transformed by Hadamard and then mapped to complex symbols by a very dense constellation (64K-QAM): each coefficient is quantized into an 8-bit integer number and every two integers compose one complex number of 65 536 possible values. The meta data, i.e., the scaling factor of PDO, are coded using conventional scheme consisting of fix length coding, forward error correction (FEC), and binary phase-shift keying (BPSK) mapping. Finally, the modulated symbols are passed into the raw OFDM module undergoing IFFT and D/A conversion, and the analog signals are then modulated with carrier waves to generate transmitted signals.

The client side of LayerCast is shown in Fig. 2. The OFDM module receives the signal and reconstructs the modulated complex symbols of both the scaled coefficients and the meta data. The meta data are demodulated and decoded to get the scaling factor. The scaled coefficients are optimally reconstructed by a linear minimal mean square error (LMMSE) estimation module following the inverse 64K-QAM and inverse Hadamard transform. The inverse 64K-QAM here does nothing but decouples each complex value back into two real values. Each real value here is actually an 8-bit integer number plus channel noise. After coset decoding, the DCT coefficients are estimated by LMMSE again. At last, the coefficients are inversely transformed by 3-D-DCT, to generate the final reconstruction.

A. Coset Coding

LayerCast utilizes coset coding to generate multiple layer of video data. Coset coding is a typical technique used in DSC [32], [39]. One typical problem of DSC is how to encode the source when the side information (i.e., its predictor) is only available at the decoder. Coset coding partitions the set of possible input source values into several cosets and transmits the coset index to the decoder. With the coset index and the predictor, the decoder can recover the source value by choosing the one in the coset closest to the predictor. Coset coding achieves compression because the coset index typically has an entropy lower than the source value.

An example of coset coding is as follows. Suppose the encoder wants to send the number 103 to the decoder, and the decoder knows that this number is close to 100 and the distance is less than 5. Then the encoder can send the modulo $103\%10 = 3$ instead of 103 to the decoder. Getting the modulo 3, the decoder has many candidate numbers such as 3, 23, 93, and 103. However, among them 103 is the only one whose distance to 100 is less than 5. The number 100 in this example is called side information in DSC and the condition $|103 - 100| < 5$ guarantees the unique decoding.

Before applying coset coding, LayerCast skips those insignificant DCT chunks like SoftCast [6]. The DCT coefficients are divided into several chunks as shown in Fig. 3. The variance of each chunk is compared with a threshold. If the variance is larger than the threshold, the corresponding chunk is considered to be significant. Otherwise, the corresponding chunk is skipped to save bandwidth. Let n be the number of significant chunks and let F_i $(i = 1, 2, \ldots, n)$ be the ith significant chunk. Each F_i contains a chunk of DCT coefficients.

The proposed framework uses a special coset code with real value input and real value output. Let m be the number of layers. The proposed approach quantizes each significant DCT chunk by m different quantizers and gets m different coset chunks for each chunk as

$$C_{k,i} = F_i - \mathsf{Q}_{\mathsf{k,i}}(F_i), \quad k = 1, 2, \ldots, m; \quad i = 1, 2, \ldots, n \tag{2}$$

where $\mathsf{Q}_{\mathsf{k,i}}(\cdot)$ is a quantization function. In this formula, each F_i represents one chunk of coefficients and the quantization function $\mathsf{Q}_{\mathsf{k,i}}$ can be either scalar quantization (SQ) or vector quantization. In our LayerCast, we have implemented both SQ and trellis coded quantization (TCQ) [40]. We define $q_{k,i}$ as the quantization step size. For SQ, $\mathsf{Q}_{\mathsf{k,i}}$ can be written as

$$\mathsf{Q}_{\mathsf{k,i}}(F_i) = \left\lfloor \frac{F_i}{q_{k,i}} + \frac{1}{2} \right\rfloor q_{k,i}, \quad k = 1, \ldots, m; \quad i = 1, 2, \ldots, n. \tag{3}$$

Each coset chunk $C_{k,i}$ contains partial information of the DCT chunk F_i. The encoder will transmit each coset chunk to the decoder side, instead of transmitting F_i m times. The decoder will use these coset chunks to estimate each F_i. For each chunk i, the first quantization function $\mathsf{Q}_{1,i}(\cdot)$ is designed to be coarse enough (e.g., set the quantization step size to infinity) to guarantee $\mathsf{Q}_{1,i}(\cdot) = 0$ and hence

$$C_{1,i} = F_i, \quad i = 1, 2, \ldots, n \tag{4}$$

which means that the first coset chunk is just the chunk F_i itself. Furthermore, the quantization functions $\mathsf{Q}_{k,i}(\cdot)$ are designed to be from coarse to fine when k goes from 1 to m, i.e., the quantization step size decreases successively

$$q_{m,i} \leq \cdots \leq q_{2,i} \leq q_{1,i} = \infty, \quad i = 1, 2, \ldots, n \tag{5}$$

such that each $C_{k,i}$ represents different scale of detail of F_i.

B. Coset Decoding

The decoder applies an m-layer coset decoding to reconstruct each significant DCT chunk F_i. Let $\hat{C}_{k,i}$ represent the reconstruction of the kth coset chunk of F_i. According to (4), at the first layer, each F_i is reconstructed by

$$\hat{F}_i^{(1)} = \hat{C}_{1,i}, \quad i = 1, 2, \ldots, n. \tag{6}$$

Then, at all the following layers, the reconstruction $\hat{F}_i^{(k)}$ is decoded using $\hat{F}_i^{(k-1)}$ as the side information. According to (2), F_i can be decoded by summing up the coset $C_{k,i}$ and the quantized value $\mathsf{Q}_{k,i}(F_i)$. At the decoder side, we do not have $\mathsf{Q}_{k,i}(F_i)$ but can use $\mathsf{Q}_{k,i}(\hat{F}_i^{(k-1)} - \hat{C}_{k,i})$ instead

$$\hat{F}_i^{(k)} = \hat{C}_{k,i} + \mathsf{Q}_{k,i}\big(\hat{F}_i^{(k-1)} - \hat{C}_{k,i}\big), \quad k = 2, 3, \ldots, m \tag{7}$$

and accordingly the successful condition for coset decoding is

$$\mathsf{Q}_{k,i}(F_i) = \mathsf{Q}_{k,i}\big(\hat{F}_i^{(k-1)} - \hat{C}_{k,i}\big), \quad k = 2, 3, \ldots, m. \tag{8}$$

This successful condition is not hard to satisfy. Since $\hat{F}_i^{(k-1)}$ and $\hat{C}_{k,i}$ are noisy version of F_i and $C_{k,i}$, respectively, their difference is a noisy version of $\mathsf{Q}_{k,i}(F_i)$ according to (2). Therefore, applying the same quantization on $\hat{F}_i^{(k-1)} - \hat{C}_{k,i}$ will remove the noise if the quantization step size is large enough compared with the magnitude of the noise.

When the successful condition is satisfied, the reconstruction noise comes only from the channel, and at each layer k the distortion of the DCT chunk F_i is equal to the distortion of the coset chunk

$$\hat{F}_i^{(k)} - F_i = \hat{C}_{k,i} - C_{k,i}, \quad k = 1, 2, \ldots, m. \tag{9}$$

The final reconstruction is LMMSE estimation given the reconstruction of all layers

$$\hat{F}_i = \mathbb{E}\big(F_i | \hat{F}_i^{(1)}, \hat{F}_i^{(2)}, \ldots, \hat{F}_i^{(m)}\big). \tag{10}$$

The reconstruction quality is dominated by the last layer because it has best quality among all the layers, and therefore we have

$$\hat{F}_i \approx \hat{F}_i^{(m)}. \tag{11}$$

C. Discussion on Coset Coding and Retransmission

The retransmission scheme in [6] with maximal ratio combining (MRC) at receiver can be considered as a special case of our scheme. When all the quantization step size tends to infinity, all the coset chunks $C_{k,i}$ of different layers will be equal to the corresponding DCT chunk F_i. The LMMSE estimation in our scheme is equivalent to the MRC in that situation.

However, if all the layers transmit duplicate information, there is redundancy between each layer. This redundancy can be exploited by utilizing DSC. At each enhancement layer, the reconstruction of F_i from a lower layer can be used as side information for DSC to improve the coding efficiency. This is the reason why we utilize coset coding in our framework. The coset coding removes the redundant part and keeps only the residual part. This reduces the magnitude of the value to transmit. Under the same transmission power, this means higher gain and higher precision.

It seems that our LayerCast may suffer error propagation from lower layers to higher layers, since each layer uses the reconstruction of its nearest lower layer as side information. However, the principle of DSC is to correct the error in the side information, and it has been used in the layered framework to stop error propagation [33]. In LayerCast, the coset code is an error correction code and the decoding in each layer is in fact the process to correct the reconstruction error of the nearest lower layer.

As to the encoding and decoding complexity, retransmission is almost free while coset coding is also simple enough. The coset encoding needs m (the number of layers) times of quantization, multiplication, and subtraction for each DCT coefficient. Its complexity is much lower than that of DCT. The coset decoding has a complexity similar to that of coset encoding.

D. Channel Coding of Coset Data

The channel coding of the coset data is directly based on real-valued symbols rather than binary symbols. Similar to SoftCast, the channel coding consists of power allocation, Hadamard transform, packaging, and 64K-QAM.

After coset coding, the coset coefficients are scaled for optimal power allocation. Let $g_{k,i}$ be the scaling factor of $C_{k,i}$, and let $A_{k,i}$ be the coset chunk after power allocation. Then

$$A_{k,i} = g_{k,i} C_{k,i}. \tag{12}$$

The optimal value of all the scaling factor $g_{k,i}$ will be discussed in Section IV. After optimal power allocation, all layers of coset chunks will be scaled to similar dynamic range.

The channel coding of the scaled coset chunks is layer by layer. It includes Hadamard transform and 64K-QAM constellation similar to [6]. After power allocation, the variances of each chunk in the same layer may be still different. To redistribute energy, the coefficients from different chunks in the same layer are combined together to form several new vectors and each new vector has a similar norm. It follows that the Hadamard transform is applied on the new vectors

$$X = \mathsf{Hadamard}(A) \tag{13}$$

TABLE I

COMPARISON OF THREE FRAMEWORKS (THE META DATA IS IGNORED IN THIS TABLE)

	SoftCast	LayerCast	Conventional
Video compression	3D-DCT	3D-DCT	H.264
Bandwidth expansion	Retransmission	Coset coding	SVC
Compression result	Real	Real	Binary
Optimization	PDO	PDO	RDO
Resource allocation	Power allocation between chunks	Power allocation between chunks and layers	Bit allocation
Protection	MRC, Hadamard	Coset coding, Hadamard	FEC/UEP
Constellation	64K-QAM	64K-QAM	BPSK,QPSK,$\cdots$

then the transformed coefficients X in the same layer are randomly grouped together. This creates coset packets with equal energy and equal importance. The coefficients of each packet are then directly mapped to complex symbols by a very dense constellation, 64K-QAM [6]. Each coefficient is quantized into an 8-bit integer number and every two integers compose one complex symbol (of 65 536 possible values). Let $\mathsf{Q}_{8\text{bits}}(\cdot)$ be the quantization function. The 64K-QAM can be expressed as

$$Z_t = \mathsf{Q}_{8\text{bits}}(X_{2t}) + j\mathsf{Q}_{8\text{bits}}(X_{2t+1}). \tag{14}$$

The additional distortion caused by this mapping is negligible because the 8-bits quantization is fine enough for video coefficients.

E. Transmission

The decoder requires both the coset quantization step size and the scaling factor. As shown in next section, these values are all calculated at decoder based on the signal standard deviation σ_{F_i}. The transmission of each σ_{F_i} is through traditional communication scheme consisting of entropy coding, channel coding, and modulation. Each σ_{F_i} is quantized by an 8-bit scalar quantizer and coded by fixed length coding. The compressed bitstream is then further coded using the 1/2 convolutional code (with generator polynomials {133, 171}) and BPSK constellation. This forms the meta data packet.

The transmissions of the meta data and coset data are similar to the one in SoftCast [6]. LayerCast first transmits the meta data packet and then the coset data packets layer by layer. There is no special power allocation between each packet. Since the meta data is coded by 1/2 FEC and BPSK constellation, it can be correctly decoded when the channel SNR is in 802.11's typical working range (5–25 dB). Note that the size of the meta data is very small compared with the coset data. According to our experiments, the proportion of the bandwidth required by headers is less than 3%.

The transmission of each packet is by the following raw OFDM [6]. Let vector S be a packet of data. Each element in S is a complex value. An inverse fast Fourier transform (FFT) is computed on each packet of elements, giving a set of complex time-domain samples. These samples are then quadrature-mixed to passband in the standard way. The real and imaginary components are first converted into the analog domain using D/A converters; the analog signals are then used to modulate cosine and sine waves at the carrier frequency, f_c, respectively. These signals are then summed to give the transmission signal, $s(t)$

$$s(t) = \mathsf{Re}\{\mathsf{IFFT}(S)e^{2\pi j f_c t}\}. \tag{15}$$

F. Receiver

The receiver gets signal $r(t)$ from the channel and reconstructs the video signal. The OFDM module receives the signal and reconstructs the modulated complex symbols of both the coset data and the parameter data. The received signal $r(t)$ is quadrature-mixed down to baseband using cosine and sine waves at the carrier frequency. After applying the low-pass filters, the OFDM module samples and converts the baseband signals into digital numbers, and uses a forward FFT to convert them back into the frequency domain.

The frequency domain signal after FFT includes coset data and parameter data. The parameter data are decoded first. The soft information of the original bitstream is estimated by a soft BPSK detector. Then the receiver uses a Viterbi algorithm to correct the errors in the bitstream, and then decodes the bitstream to get each standard deviation σ_{F_i}. The quantization parameters and the scaling factors are calculated based on all the standard deviation σ_{F_i}.

The scaled coset chunks are reconstructed by inverse 64K-QAM and inverse Hadamard. The inverse 64K-QAM here just splits each complex symbol back into two real values. Each real value here is actually the 8-bits integer number plus channel noise. The decoder applies inverse Hadamard transform on the real values. Let $\tilde{A}$ be the coefficients after inverse Hadamard transform. $\tilde{A}$ can be written as

$$\tilde{A} = A + N \tag{16}$$

where N is the equivalent channel noise after inverse Hadamard transform. Under the assumption that the channel noise is white Gaussian, N is also white Gaussian. Therefore, the LMMSE estimation of each chunk $A_{k,i}$ is

$$\hat{A}_{k,i} = \frac{\sigma^2_{A_{k,i}}}{\sigma^2_{A_{k,i}} + \sigma^2_N}\tilde{A}_{k,i}. \tag{17}$$

Then corresponding to the power allocation in (12), the coset chunk $C_{k,i}$ is reconstructed by

$$\hat{C}_{k,i} = \hat{A}_{k,i}/g_{k,i}. \tag{18}$$

With coset chunks, each significant DCT chunk F_i is decoded as explained in Section III-B. At last, the video frames are reconstructed by inverse 3-D-DCT.

G. Comparison With Existing Frameworks

Table I shows a comparison of the techniques used in three different frameworks: SoftCast, LayerCast, and conventional H.264-based framework. Note that the transmission of the meta data in SoftCast and LayerCast is the same as the transmission of H.264 bitstreams, thus ignored in this table.

In video compression, both SoftCast and LayerCast use 3-D-DCT, while conventional framework uses more complicated codec such as H.264. This means that the conventional framework has higher compression efficiency. However, the compression result of the conventional framework is binary sequence, which is sensitive to channel errors. In contrast, the compression result of SoftCast and LayerCast is a real-valued sequence, which is robust to channel noise. After modulation, the result is still robust because the 64K-QAM is just a slight quantization in SoftCast/LayerCast. Therefore, the conventional framework needs extra bandwidth to apply FEC/UEP, while SoftCast and LayerCast apply Hadamard transform to handle packet loss without introducing extra bandwidth. Another difference is that SoftCast and LayerCast do not have bit rate and the distortion is related to the transmission power. Therefore, they need PDO and power allocation, instead of RDO and bit allocation.

The main difference between SoftCast and LayerCast is that LayerCast is designed for layered video broadcast. When some receivers have more bandwidth, SoftCast simply retransmits the signal but LayerCast utilizes coset coding to exploit the cross-layer redundancy. Coset coding as an error correction code can protect the signal in LayerCast, while SoftCast can use MRC to denoise. More importantly, the power allocation in SoftCast is only between DCT chunks, while the power allocation in LayerCast is between DCT chunks and layers. We have solved this problem and achieved a global optimal solution as shown in the following section.

IV. Power-Distortion Optimization

This section focuses on the PDO of the proposed LayerCast, including the optimal quantization step size of coset coding and the optimal power allocation between layers and DCT chunks.

A. Quantization Step of Coset Coding

The value of each quantization parameter $q_{k,i}$ is very important to the performance of coset coding. If the quantization step size is too large, then the gain of coset coding will be small. Otherwise, the coset decoder may suffer error.

Starting from the successful condition (8), for $k = 2, 3, \ldots, m$, we have

$$\begin{aligned}
\mathsf{Q}_{\mathsf{k,i}}(F_i) &\overset{(8)}{=} \mathsf{Q}_{\mathsf{k,i}}\big(\hat{F}_i^{(k-1)} - \hat{C}_{k,i}\big) && (19)\\
&\overset{(2)}{=} \mathsf{Q}_{\mathsf{k,i}}\big(\hat{F}_i^{(k-1)} - \hat{C}_{k,i} + C_{k,i} - F_i + \mathsf{Q}_{\mathsf{k,i}}(F_i)\big) && (20)\\
&\overset{(3)}{=} \mathsf{Q}_{\mathsf{k,i}}\big(\hat{F}_i^{(k-1)} - F_i + C_{k,i} - \hat{C}_{k,i}\big) + \mathsf{Q}_{\mathsf{k,i}}(F_i) && (21)\\
&\overset{(9)}{=} \mathsf{Q}_{\mathsf{k,i}}\big(\hat{F}_i^{(k-1)} - F_i + F_i - \hat{F}_i^{(k)}\big) + \mathsf{Q}_{\mathsf{k,i}}(F_i) && (22)\\
&= \mathsf{Q}_{\mathsf{k,i}}\big(\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}\big) + \mathsf{Q}_{\mathsf{k,i}}(F_i) && (23)
\end{aligned}$$

that is

$$\mathsf{Q}_{\mathsf{k,i}}\big(\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}\big) = 0. \quad (24)$$

According to the definition (3), the quantization result is 0 if and only if the input's magnitude is smaller than $(q_{k,i}/2)$. Let $\sigma_{\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}}$ be the standard deviation of chunk $\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}$. We let the quantization step size to satisfy

$$\frac{q_{k,i}}{2} = T\sigma_{\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}}, \quad k = 2, 3, \ldots, m \quad (25)$$

such that the elements in chunk $\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}$ will be quantized to 0 in high probability, where T is a threshold to control the probability. If the elements in chunk $\hat{F}_i^{(k-1)} - \hat{F}_i^{(k)}$ satisfy Gaussian distribution (actually this is true of AWGN channel), then $T = 3$ means that more than 99.7% elements will be quantized to 0. Furthermore, chunk F_i is closer to $\hat{F}_i^{(k)}$ than $\hat{F}_i^{(k-1)}$ because $\hat{F}_i^{(k)}$ is the reconstruction of higher layer. Therefore, we have

$$\begin{aligned}
\frac{q_{k,i}}{2} &\approx T\sigma_{\hat{F}_i^{(k-1)} - F_i} && (26)\\
&\overset{(9)}{=} T\sigma_{C_{k-1,i} - \hat{C}_{k-1,i}}, \quad k = 2, 3, \ldots, m && (27)
\end{aligned}$$

where $\sigma_{C_{k-1,i} - \hat{C}_{k-1,i}}$ is the standard deviation of elements in chunk $C_{k-1,i} - \hat{C}_{k-1,i}$.

Notice that the quantization step size needs to be calculated at the transmitter but it depends on the reconstruction distortion. In multicast, to guarantee successful coset decoding, the quantization step size is determined by the minimal channel SNR of all receivers.

B. Reconstruction Distortion

For each coset chunk $C_{k,i}$, the decoding process is equivalent to an LMMSE denoising process, and the reconstruction distortion is related to the signal power $\sigma_{A_{k,i}}^2$ and the channel noise power σ_N^2 as follows:

$$\sigma^2_{C_{k,i} - \hat{C}_{k,i}} = \frac{\sigma^2_{C_{k,i}} \sigma^2_N}{\sigma^2_{A_{k,i}} + \sigma^2_N}, \quad k = 1, 2, \ldots, m. \quad (28)$$

According to (2), the coset $C_{k,i}$ is actually the quantization residue of a scalar quantizer of step size $q_{k,i}$. Typically, a video encoder will assume that the quantization residue follows uniform distribution and thus the quantization distortion is $(q_{k,i}^2/12)$. This together with (26) implies

$$\sigma^2_{C_{k,i}} = \frac{q^2_{k,i}}{12} = \frac{T^2}{3}\sigma^2_{C_{k-1,i} - \hat{C}_{k-1,i}}, \quad k = 2, 3, \ldots, m. \quad (29)$$

By recursively applying (28) and (29), we get

$$\sigma^2_{C_{m,i} - \hat{C}_{m,i}} = \left(\frac{T^2}{3}\right)^{m-1} \sigma^2_{C_{1,i}} \prod_{k=1}^{m} \frac{\sigma^2_N}{\sigma^2_{A_{k,i}} + \sigma^2_N}. \quad (30)$$

According to (9) and (4), we have

$$\sigma^2_{C_{m,i} - \hat{C}_{m,i}} = \sigma^2_{\hat{F}_i^{(m)} - F_i} \quad \text{and} \quad \sigma^2_{C_{1,i}} = \sigma^2_{F_i}. \quad (31)$$

Thus, the distortion of the ith chunk at final layer can be expressed as

$$\sigma^2_{\hat{F}_i^{(m)} - F_i} = \left(\frac{T^2}{3}\right)^{m-1} \sigma^2_{F_i} \prod_{k=1}^{m} \frac{\sigma^2_N}{\sigma^2_{A_{k,i}} + \sigma^2_N}. \quad (32)$$

Let D be the average distortion of all chunks, then

$$D = \frac{1}{n}\sum_{i=1}^{n}\left(\left(\frac{T^2}{3}\right)^{m-1}\sigma_{F_i}^2\prod_{k=1}^{m}\frac{\sigma_N^2}{\sigma_{A_{k,i}}^2+\sigma_N^2}\right). \tag{33}$$

C. Power Allocation Problem Definition

Let P_i be the average transmission power of the ith DCT chunk. Since ith DCT chunk has been encoded into m layers of scaled coset chunks (i.e., $A_{1,i}, A_{2,i}, \ldots, A_{m,i}$), P_i is the mean of the variance of each coset chunk

$$P_i = \frac{1}{m}\sum_{k=1}^{m}\sigma_{A_{k,i}}^2. \tag{34}$$

Let P be the average transmission power of all coset chunks. Then we have

$$P = \frac{1}{n}\sum_{i=1}^{n}P_i. \tag{35}$$

The target of power allocation is to minimize the distortion D in (33), and thus, the problem can be written as

$$\begin{aligned}\min_{\{P_i\},\left\{\sigma_{A_{k,i}}^2\right\}}\ & \frac{1}{n}\sum_{i=1}^{n}\left(\left(\frac{T^2}{3}\right)^{m-1}\sigma_{F_i}^2\prod_{k=1}^{m}\frac{\sigma_N^2}{\sigma_{A_{k,i}}^2+\sigma_N^2}\right)\\ \text{s.t.}\ & P = \tfrac{1}{n}\sum_{i=1}^{n}P_i\\ & P_i = \tfrac{1}{m}\sum_{k=1}^{m}\sigma_{A_{k,i}}^2.\end{aligned} \tag{36}$$

Since n, T, m, and σ_N^2 are all constant to this problem, (36) is equivalent to

$$\begin{aligned}\min_{\{P_i\},\left\{\sigma_{A_{k,i}}^2\right\}}\ & \sum_{i=1}^{n}\left(\sigma_{F_i}^2\prod_{k=1}^{m}\frac{1}{\sigma_{A_{k,i}}^2+\sigma_N^2}\right)\\ \text{s.t.}\ & P = \tfrac{1}{n}\sum_{i=1}^{n}P_i\\ & P_i = \tfrac{1}{m}\sum_{k=1}^{m}\sigma_{A_{k,i}}^2.\end{aligned} \tag{37}$$

D. Power Allocation Between Layers

To solve this problem, we will first derive the constraint that the optimal solution must satisfy, and then simplify the problem with the new constraint.

Let $\{P_i^*\}$ and $\{\sigma^{*2}_{A_{k,i}}\}$ ($i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, m$) be an optimal solution of (37). Then $\{\sigma^{*2}_{A_{k,i}}\}$ must be an optimal solution of the following problem [otherwise, if $\{\sigma^{+2}_{A_{k,i}}\}$ is a better solution of the following problem, then $\{P_i^*\}$ and $\{\sigma^{+2}_{A_{k,i}}\}$ will be a better solution of (37)]:

$$\begin{aligned}\min_{\left\{\sigma_{A_{k,i}}^2\right\}}\ & \sum_{i=1}^{n}\left(\sigma_{F_i}^2\prod_{k=1}^{m}\frac{1}{\sigma_{A_{k,i}}^2+\sigma_N^2}\right)\\ \text{s.t.}\ & P_i^* = \frac{1}{m}\sum_{k=1}^{m}\sigma_{A_{k,i}}^2.\end{aligned} \tag{38}$$

By observing this problem, we can see that the objective value is the summation of n items and each item is only related to one constraint. Furthermore, each constraint is independent. Therefore, this problem can be separated into n independent subproblems. For each $i (i = 1, 2, \ldots, n)$, the ith subproblem is actually the power allocation among each layer within the ith DCT chunk

$$\begin{aligned}\min_{\left\{\sigma_{A_{k,i}}^2\right\}}\ & \sigma_{F_i}^2\prod_{k=1}^{m}\frac{1}{\sigma_{A_{k,i}}^2+\sigma_N^2}\\ \text{s.t.}\ & P_i^* = \frac{1}{m}\sum_{k=1}^{m}\sigma_{A_{k,i}}^2.\end{aligned} \tag{39}$$

This subproblem is equivalent to

$$\begin{aligned}\max_{\{\sigma_{A_{k,i}}^2\}}\ & \prod_{k=1}^{m}\left(\sigma_{A_{k,i}}^2+\sigma_N^2\right)\\ \text{s.t.}\ & P_i^* = \frac{1}{m}\sum_{k=1}^{m}\sigma_{A_{k,i}}^2.\end{aligned} \tag{40}$$

This problem can be easily solved using the following inequality of arithmetic and geometric means:

$$\sqrt[m]{\prod_{k=1}^{m}\left(\sigma_{A_{k,i}}^2+\sigma_N^2\right)} \le \frac{1}{m}\sum_{k=1}^{m}\left(\sigma_{A_{k,i}}^2+\sigma_N^2\right) \tag{41}$$

and the solution is

$$\sigma^{*2}_{A_{k,i}} = P_i^*, \quad k = 1, 2, \ldots, m. \tag{42}$$

This means that all the $\sigma_{A_{k,i}}^2$ for the same i must be equal, i.e., the coset chunk of different layers within the same DCT chunk should take the same transmission power.

E. Power Allocation Between DCT Chunks

Since the optimal solution of (37) must satisfy (42), (37) can be simplified to

$$\begin{aligned}\min_{\{P_i\}}\ & \sum_{i=1}^{n}\left(\sigma_{F_i}^2\prod_{k=1}^{m}\frac{1}{P_i+\sigma_N^2}\right)\\ \text{s.t.}\ & P = \frac{1}{n}\sum_{i=1}^{n}P_i\end{aligned} \tag{43}$$

or equivalently

$$\begin{aligned}\min_{\{P_i\}}\ & \sum_{i=1}^{n}\left(\sigma_{F_i}^2\left(P_i+\sigma_N^2\right)^{-m}\right)\\ \text{s.t.}\ & nP = \sum_{i=1}^{n}P_i.\end{aligned} \tag{44}$$

Notice that P_i is the average transmission power of all layers of coset data in the ith DCT chunk. This problem is actually the power allocation between each DCT chunk. To solve this problem, we introduce a Lagrangian multiplier λ and the problem becomes

$$\begin{aligned}\min_{\{P_i\}}\ & \sum_{i=1}^{n}\left(\sigma_{F_i}^2\left(P_i+\sigma_N^2\right)^{-m}\right) - \lambda m\left(nP - \sum_{i=1}^{n}P_i\right)\\ \text{s.t.}\ & nP = \sum_{i=1}^{n}P_i.\end{aligned} \tag{45}$$

By taking derivative to each P_i, we get

$$-\sigma_{F_i}^2 \left(P_i + \sigma_N^2\right)^{-m-1} + \lambda = 0, \quad i = 1, 2, \ldots, n. \tag{46}$$

By solving the combination of (35) and (46), we get

$$\lambda = \left(\frac{\sum_{i=1}^{n} \sigma_{F_i}^{\frac{2}{m+1}}}{nP + n\sigma_N^2} \right)^{m+1} \tag{47}$$

and

$$P_i = \frac{\sigma_{F_i}^{\frac{2}{m+1}}}{\sum_{i=1}^{n} \sigma_{F_i}^{\frac{2}{m+1}}} \left(nP + n\sigma_N^2\right) - \sigma_N^2. \tag{48}$$

Therefore, the ratio of transmission power allocated to the ith DCT chunk is

$$\frac{P_i}{nP} = \frac{\sigma_{F_i}^{\frac{2}{m+1}}}{\sum_{i=1}^{n} \sigma_{F_i}^{\frac{2}{m+1}}} + \left(\frac{\sigma_{F_i}^{\frac{2}{m+1}}}{\sum_{i=1}^{n} \sigma_{F_i}^{\frac{2}{m+1}}} - \frac{1}{n} \right) \frac{\sigma_N^2}{P}. \tag{49}$$

This means the optimal power allocation should be calculated based on the channel SNR. In multicast, the channel SNR is unknown to the encoder. Therefore, we use the following approximation. When the channel SNR is large enough, i.e., $P \gg \sigma_N^2$, the optimal power allocation can be approximated as

$$\frac{P_i}{nP} \approx \frac{\sigma_{F_i}^{\frac{2}{m+1}}}{\sum_{i=1}^{n} \sigma_{F_i}^{\frac{2}{m+1}}}. \tag{50}$$

According to this result, the optimal allocation gives more transmission power to the DCT chunks with large variance. Notice that this result is the same as the one in [6] only when there is one layer. When m becomes larger, the power allocated to each chunk tends to become equal. In LayerCast, each layer of coset chunk is a relay to successively improve the video quality (i.e., decrease the distortion). If the power allocation in [6] is used, the DCT chunks with large variance will get more transmission power, and their distortion will decrease much faster than the other chunks especially when the number of layers is large. Therefore, the optimal power allocation in (50) tends to tune it back and give each chunk a similar transmission power when the number of layers increases.

With (42) and (50), optimal encoding parameters can be calculated as follows. First, the standard derivation of each DCT chunk is calculated. This is the most complicated operation because it involves all DCT coefficients. Then, each variance $\sigma_{A_{k,i}}^2$ can be determined using (42) and (50). According to (4), we have $\sigma_{C_{0,i}}^2 = \sigma_{F_i}^2$. With (28) and (29), each $\sigma_{C_{k,i}}^2$ and $\sigma_{C_{k,i}-\hat{C}_{k,i}}^2$ can be recursively calculated. It follows to calculate each quantization step size $q_{k,i}$ by (26). Finally, according to the relationship in (12), each $g_{k,i}$ can be calculated by

$$g_{k,i} = \frac{\sigma_{A_{k,i}}}{\sigma_{C_{k,i}}}. \tag{51}$$

As to the complexity, both our PDO and the one in [6] mainly require the calculation of each σ_{F_i} (i.e., the standard derivation of each DCT chunk) and are similarly simple.

V. Simulation Result

In experiments, LayerCast is compared with both SoftCast [6] and conventional framework. Both LayerCast and SoftCast use the same 3-D-DCT, and the GOP length is 4 frames. The packets generated by LayerCast and SoftCast are passed into the raw OFDM module and then transmitted over the AWGN channel simulated by MATLAB R2010b. For bandwidth expansion, we let SoftCast use retransmission at the encoder and apply MRC at the decoder. We calculate the equivalent channel SNR and use it as a parameter for the SoftCast decoder. (For the AWGN channel, MRC of two duplicate symbols increases the equivalent channel SNR by 3 dB.) We run all the following tests several times to get the average results. Since the channel is simulated and the video is long enough, every time we get a similar result. The maximal difference is less than 0.05 dB for all tests. The results shown in all figures are achieved by repeating the tests 10 times, except that the packet loss result is by repeating the tests 100 times.

The conventional framework is based on H.264 and 802.11. We use the JM14.2 software as H.264 codec and use the baseline profile. We implement the FEC and QAM modulations of standard 802.11 PHY layer without pilot, synchronization, and equalization. The H.264 coded video data is packed into RTP packets of length 1200 bytes. We insert into each RTP packet a 32-bit CRC, and then encode each packet separately by the FEC code. For FEC, we generate the $1/2$ convolutional code with polynomials $\{133, 171\}$ and puncture it to get $2/3$ and $3/4$ convolutional codes. The FEC coded bits are mapped to the complex symbols by QAM constellations, including BPSK, QPSK, 16QAM, and 64QAM. The complex symbols are passed into the raw OFDM module and then transmitted over the AWGN channel simulated by MATLAB R2010b. The channel decoding is by a soft Viterbi algorithm. After that, the decoder performs CRC check for each RTP packet and forwards those error-free packets to the H.264 decoder. The H.264 decoder is able to tolerate a small percentage of RTP packet loss, by utilizing error concealment. In our test, we have configured the H.264 decoder to use the most powerful error concealment method in JM14.2, the motion copy one, to get the best reconstruction quality. The GOP length is 16. (If H.264 uses GOP length 4 like SoftCast, its performance will drop.) The search range of ME is 32×32, and the MV precision is $1/4$ pixel. The rate control algorithm is the default one of JM14.2.

The test video sequences are standard CIF sequences (352×288, 30 Hz). To evaluate the average performance of each framework, we also create a monochrome 512-frame test video sequence, called all_seq, by combining the first 32 frames of the following 16 test sequences: *Akiyo*, *Bus*, *Coastguard*, *Crew*, *Flower*, *Football*, *Foreman*, *Harbor*, *Husky*, *Ice*, *News*, *Soccer*, *Stefan*, *Tempete*, *Tennis*, and *Waterfall*.

The number of DCT coefficients of the CIF video signal is about 3M per second. Since we transmit complex symbols, this should require a channel bandwidth of about 1.5 MHz. However, some DCT chunks are skipped because their variance is smaller than a threshold. In our experiments, the threshold is 3 and about $1/3$ DCT chunks are skipped for the sequence all_seq. If we use a larger threshold, more chunks will be skipped but the performance of LayerCast and SoftCast at a high SNR will drop. Therefore, the remaining coefficients, plus the header for parameters, require a bandwidth of about 1.1 MHz. When the channel bandwidth is larger than this, we will have extra bandwidth to transmit enhancement layers.

A. Unicast Performance

Although LayerCast is mainly designed for broadcast application, its unicast performance is still important. If the unicast performance is too low, then LayerCast can hardly compete with a conventional H.264 framework when receivers have similar channel SNRs and the same bandwidth. Therefore, in this experiment we compare the unicast performance of each framework at different channel SNRs and different channel bandwidths. In this test, the encoder is allowed to know both the target channel SNR and the channel bandwidth.

The first test compares each framework at different channel bandwidths. In this test, the channel SNR is 7 dB, and the channel bandwidth is 1.1–4.4 MHz. The value of channel SNR and bandwidth range is for video PSNR to be in the range of 30–40 dB. This range is a typical range considered in video coding. For each bandwidth, there are 10 users and the results are obtained by averaging them.

When the channel bandwidth is larger than 1.1 MHz, each framework is able to transmit not only all the significant DCT chunks but also some extra data using the extra bandwidth. In this situation, SoftCast transmits each coefficient multiple times to utilize the extra bandwidth, while our LayerCast uses coset coding. The H.264-based framework uses $1/2$ FEC as channel coding and QPSK as constellation. This is the highest rate choice that can work at channel SNR of 7 dB among all the eight recommendations of the standard 802.11 PHY layer. We calculate the corresponding bit-rates respectively according to the bandwidth, and set the bit-rate constraint to the H.264 encoder for rate control. The rate control algorithm is the default one of JM14.2.

We have implemented four versions of our LayerCast for comparison between SQ and TCQ, and comparison between different PDO strategies (Proposed PDO versus the PDO in [6]). The result is given in Fig. 4. According to this figure, SoftCast and LayerCast perform similarly when channel bandwidth is low. However, when channel bandwidth becomes higher, our LayerCast performs much better than SoftCast and can gain up to 4 dB. This is because SoftCast applies retransmission and introduces redundancy, while our LayerCast utilizes coset coding and exploits the redundancy. In this figure, we can also find that the gain of applying TCQ and optimal PDO increases when the channel bandwidth increases, and can be up to 1.5 and 1 dB, respectively, when the channel

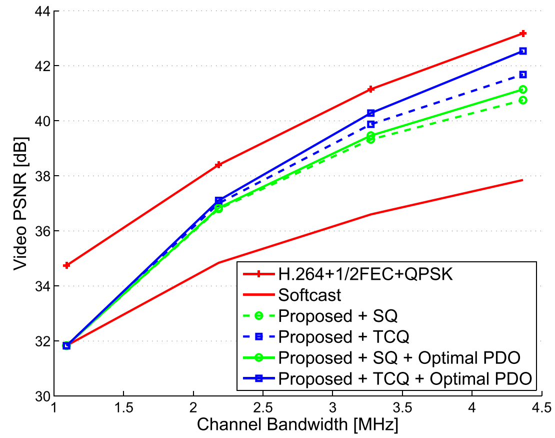


Fig. 4. Unicast performance at different bandwidths. Channel SNR is 7 dB.

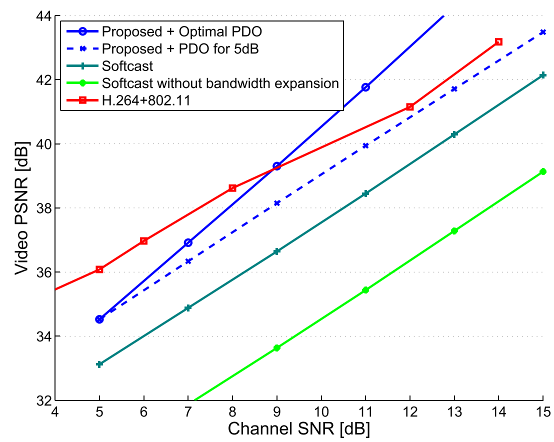


Fig. 5. Unicast performance at different channel SNRs. Channel bandwidth is 2.2 MHz.

bandwidth is large. In this test, the H.264 framework performs the best while our LayerCast gets close to it when the channel bandwidth becomes high.

The second test compares each framework at different channel SNRs. The channel bandwidth is 2.2 MHz. The channel SNR is between 5∼15 dB. The value of channel bandwidth and SNR range is also for video PSNR to be in the range of 30–40 dB. For each channel SNR, there are 10 users and the results are obtained by averaging them. We assume that the conventional H.264 framework can automatically pick up the best choice among all the eight recommended combinations of channel coding and modulation from 802.11a, according to the channel SNR. We calculate the corresponding bit-rates respectively according to the bandwidth, and set the bit-rate constraint to the H.264 encoder for rate control. The result is given in Fig. 5. According to this figure, LayerCast gains 1.3–4 dB over SoftCast at different channel SNRs. Notice that the gain becomes larger when the channel SNR increases. This is because of the inefficiency of SoftCast to exploit the redundancy between multiple transmissions at a high SNR. Our LayerCast can exploit this redundancy because each coset chunk contains exclusive refinement information of the original DCT chunk. In this figure, we also show the performance of LayerCast when PDO is optimized for

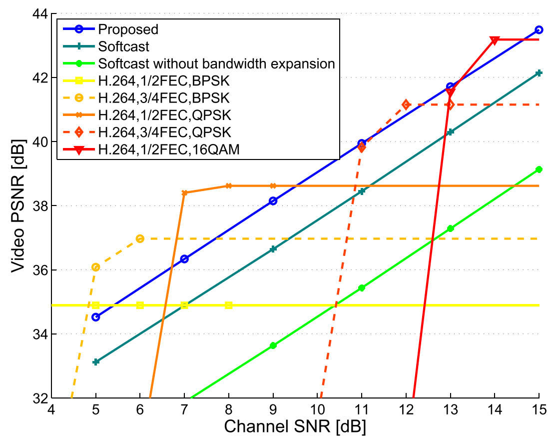


Fig. 6. Robustness comparison between LayerCast, SoftCast, and H.264. The channel bandwidth is 2.2 MHz. The LayerCast encoder is optimized for a channel SNR of 5 dB. The channel SNR is unknown to all the encoders.

a channel SNR of 5 dB for reference. It is clear that the gain of applying optimal PDO can be up to 3 dB when the channel SNR is high. In addition, we also show the performance of SoftCast without bandwidth expansion for reference (i.e., the base layer of SoftCast and LayerCast). According to this result, the gain of SoftCast's retransmission and that of LayerCast's coset coding are 3 and 7 dB respectively. In this test, the H.264 framework performs the best at a low channel SNR, while our LayerCast performs the best at a high channel SNR owing to the proposed PDO.

B. Broadcast at Fixed Bandwidth

In this test, all the frameworks broadcast video signal to users at different channel SNRs, by utilizing an equal bandwidth of 2.2 MHz. The value of channel bandwidth and SNR range is also for video PSNR to be in the range of 30–40 dB. This bandwidth is enough for SoftCast to transmit every significant DCT chunk two times. Our LayerCast applies two-layer coset coding to utilize the bandwidth efficiently.

The first experiment assumes that only the decoder knows the channel SNR. The result is given in Fig. 6. The channel SNRs tested for LayerCast are 5, 7, 9, . . . , and 15 dB. For each channel SNR, there are 10 users and the results are obtained by averaging them. According to the result, all the five combinations of conventional framework suffer very serious stair effect. For example, the combination H.264, 1/2FEC, QPSK performs well when the channel SNR is between 7 and 8 dB, but not so when the channel SNR is out of this range. When the channel SNR becomes more than 8 dB, the reconstruction quality does not increase. When the channel SNR becomes 6 dB, the reconstruction quality drops very quickly. When the channel SNR becomes even lower, the video decoder cannot work since almost all received RTP packets have bit errors. In contrast, the three soft video broadcast frameworks do not suffer the stair effect. When the channel SNR increases, the reconstruction PSNR increases accordingly, and vice versa. When the PSNR is between 32 dB and 44 dB, LayerCast is constantly 1.6 dB better than SoftCast.

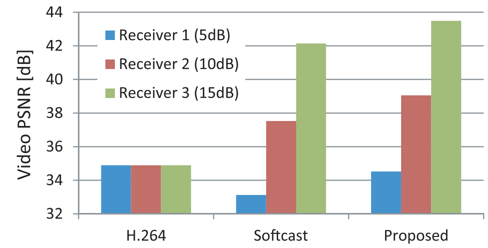


Fig. 7. Multicast to three users at different channel SNRs. The channel bandwidth is 2.2 MHz.

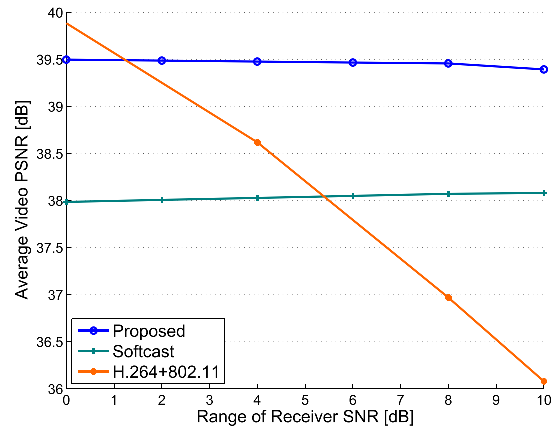


Fig. 8. Serving a group of receivers with diverse channel SNRs. The channel bandwidth is 2.2 MHz and the average SNR of each group is 10 dB.

We then let all the frameworks serve a group of three receivers with diverse channel SNRs. The channel SNR for each receiver is 5, 10, and 15 dB, respectively. For each channel SNR, there are 10 users and the results are obtained by averaging them. The test result is given in Fig. 7. In a conventional H.264 framework, the server transmits the video stream using 1/2 FEC and BPSK. It cannot use a higher transmission rate because otherwise the 5 dB user will not be able to decode the video. Due to this, although the other two receivers have better channel condition, they will also only receive low-speed 802.11 signal and reconstruct low-quality video. In constrast, SoftCast, and LayerCast can accommodate all the receivers simultaneously. Using LayerCast, the 5 dB user can get a slightly lower reconstruction quality than using H.264-based conventional framework. However, the 10 and 15 dB users get 4 and 8 dB better reconstruction quality respectively using LayerCast than conventional frameworks.

Fig. 8 compares the multicast performance of three frameworks, with respect to the range of receiver SNR. The range of receiver SNR is defined as the difference between the maximal and minimal channel SNRs of the users in the group. The group includes 10 users and the average channel SNR of them is 10 dB. When the channel SNR range is 0 dB, i.e., the channel SNRs of all the users are 10 dB, LayerCast and H.264 framework perform similarly. However, when the users' channel SNR becomes diverse, the performance of H.264 framework drops quickly.

Fig. 9 compares the performance of three frameworks at different packet loss rates. In this test, the channel bandwidth is

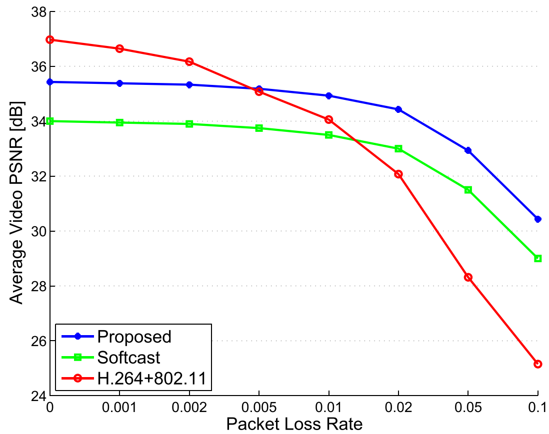


Fig. 9. Packet loss test. The channel bandwidth is 2.2 MHz and the channel SNR is 6 dB.

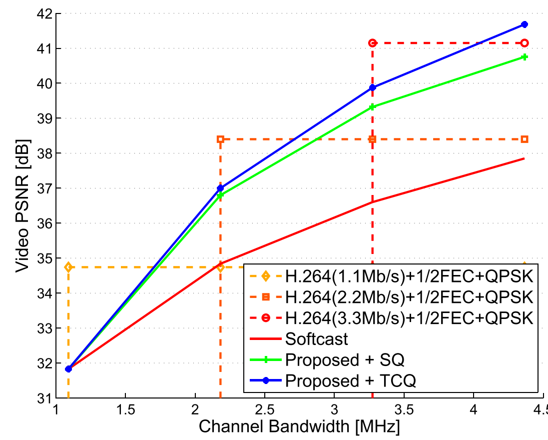


Fig. 10. Broadcast performance for users of different bandwidths. Channel SNR is 7 dB.

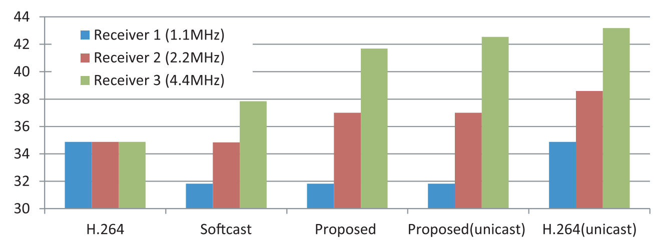


Fig. 11. Multicast to three users at different bandwidths. The channel SNR is 7 dB.

2.2 MHz and the channel SNR is 6 dB. The values of channel bandwidth and SNR are such that modest video quality is obtained (i.e., video PSNR is about 35 dB). The H.264 coded video data is packed into RTP packet of length 1200 bytes, and then further coded by 3/4 FEC and BPSK. Thus, each packet contains 12 800 channel symbols. The combination of 3/4 FEC and BPSK is the highest rate choice that can work at a channel SNR of 6 dB among all the eight recommendation of standard 802.11 PHY layer. For fair comparison, the packet size of LayerCast is also set to 12 800 symbols. In this test, we observe that different packet loss patterns cause different results, especially for H.264. Therefore, we randomly generate 100 loss patterns for each packet loss rate and obtain the average results. In our test, we have configured the H.264 decoder to use the most powerful error concealment method in JM14.2, the motion copy one, to get the best reconstruction quality. When there is no packet loss, H.264 is better than LayerCast and SoftCast. However, when the packet loss rate increases, the performance of H.264 framework drops very quickly. In contrast, LayerCast and SoftCast are quite robust. LayerCast outperforms H.264 framework even if the packet loss rate is only 0.5%. When the packet loss rate is 10%, LayerCast is 5 dB better than the H.264 framework.

C. Broadcast at Variable Bandwidth

In this test, all the frameworks broadcast video signal to users at different channel bandwidths. There are four users in this test, and their channel bandwidths are 1.1, 2.2, 3.3, and 4.4 MHz, respectively. The channel SNR is set to 7 dB such that the video PSNR is in the range of 30–40 dB. This range is a typical range considered in video coding. The H.264-based framework uses 1/2 FEC as channel coding and QPSK as constellation. This is the highest rate choice that can work at a channel SNR of 7 dB among all the eight recommendation of standard 802.11 PHY layer. We calculate the corresponding bit-rates respectively according to the bandwidth, and set the bit-rate constraint to the H.264 encoder for rate control. We run the test for 10 times and find that the difference between each test is less than 0.05 dB. The average result is given in Figs. 10 and 11.

Similar to the previous broadcast tests, the H.264-based framework can only optimize for users of one channel bandwidth. Note that the H.264 framework may get better performance by utilizing scalable coding techniques, and different scalable methods may give different performances. Thus, in Fig. 11 we also show the H.264 unicast performance, which can be considered as the upper bound of all scalable frameworks. Typically, a scalable framework performs much lower than a single-layer framework.

Although SoftCast can serve users of different bandwidths by retransmission, LayerCast can accommodate simultaneously multiple users of different bandwidths more efficiently. According to Figs. 10 and 11, LayerCast can gain more than 4 dB over SoftCast for a broadband wireless channel. When the channel bandwidth increases, the video PSNR of our LayerCast will increase accordingly. In contrast, the video PSNR of SoftCast increases more slowly than LayerCast.

D. Complexity

Table II shows the average encoding time and decoding time per frame in milliseconds. The test machine has an Intel Core i7-3770 CPU @ 3.40 GHz, 4 GB internal memory and Microsoft Windows 8 Professional Operating System. The input video is all_seq of CIF size at 30 frames per second. In this test, we compare SoftCast, LayerCast, and H.264 framework (with BPSK constellation and 1/2 FEC rate). Both SoftCast and LayerCast codecs are written in the C++ language with Microsoft Visual Studio 2008. The encoding time of SoftCast includes the time for DCT,

TABLE II
COMPARISON OF COMPLEXITY

	Encode time	Decode time	Video bit rate	Channel symbol rate
SoftCast	6ms	3ms	-	1.1M/s
LayerCast (1 layer)	6ms	3ms	-	1.1M/s
LayerCast (2 layers)	7ms	4ms	-	2.2M/s
LayerCast (3 layers)	8ms	5ms	-	3.3M/s
LayerCast (4 layers)	9ms	6ms	-	4.4M/s
H.264	22ms	4ms	550Kb/s	1.1M/s
	26ms	5ms	1100Kb/s	2.2M/s
	29ms	6ms	1650Kb/s	3.3M/s
	32ms	7ms	2200Kb/s	4.4M/s

power allocation, Hadamard transform, and 64K-QAM. The encoding time of LayerCast includes the time for all the modules in Fig. 1 except the raw OFDM module. For H.264, the encoder is x.264 (r2491-24e4fed) [41] in this test and the decoder is JM14.2. Note that a default x.264 encoder will enable the multithreading technique and assembly-level optimization techniques including MMX2, SSE2Fast, SSSE3, FastShuffle, SSE4.2, and AVX. For fair comparison, x.264 encoder is configured to disable multithreading and assembly-level optimization in this test.

The performance of LayerCast depends on the number of layers. When there is only one layer, LayerCast is the same as SoftCast. When the number of layers increases, the encoding time and decoding time increase slightly due to coset coding and decoding respectively. LayerCast has a much lower encoding complexity than the H.264 encoder mainly because LayerCast does not have motion estimation. As to the decoding complexity, LayerCast is comparable to the H.264 codec (JM14.2). Table II also shows the video bit-rate and channel symbol rate. When the modulation constellation is BPSK and the FEC rate is 1/2, the video bit rate is the half of the channel symbol rate (i.e., the channel bandwidth).

VI. CONCLUSION

In this paper, we propose a soft video broadcast framework called LayerCast. LayerCast solved the bandwidth matching problem in existing soft video broadcast frameworks using coset coding. As a result, LayerCast can accommodate diverse users of not only different channel SNRs but also different channel bandwidths. In addition, we derive for LayerCast a new PDO formula. In simulations, LayerCast outperforms SoftCast up to 4 dB when the channel SNR is high or the channel bandwidth is large, and outperforms the H.264-based framework up to 8 dB in multicast.

LayerCast in this paper is mainly designed and optimized for Gaussian channel. One possible future work is to extend the proposed LayerCast to multipath fading channel, which may require more complicated channel estimation and PDO.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments, which greatly improved this paper.

REFERENCES

[1] (2009). *Digital Video Broadcasting (DVB)*. [Online]. Available: http://www.etsi.org/deliver/etsi_en/300700_300799/300744/01.06.01_60/en_300744v010601p.pdf

[2] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2007, 2007.

[3] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 332–344, Mar. 2001.

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[5] S. Jakubczak and D. Katabi, "SoftCast: One-size-fits-all wireless video," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, Oct. 2010, pp. 449–450.

[6] S. Jakubczak and D. Katabi, "A cross-layer design for scalable mobile video," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 289–300.

[7] S. Aditya and S. Katti, "FlexCast: Graceful wireless video streaming," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 277–288.

[8] X. L. Liu, C. Luo, W. Hu, and F. Wu, "Compressive broadcast in MIMO systems with receive antenna heterogeneity," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 3011–3015.

[9] X. Fan, F. Wu, D. Zhao, O. C. Au, and W. Gao, "Distributed soft video broadcast (DCAST) with explicit motion," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2012, pp. 199–208.

[10] X. Fan, F. Wu, D. Zhao, and O. C. Au, "Distributed wireless visual communication with power distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 6, pp. 1040–1053, Jun. 2013.

[11] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[12] X. Fan, R. Xiong, F. Wu, and D. Zhao, "WaveCast: Wavelet based wireless video broadcast using lossy transmission," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Nov. 2012, pp. 1–6.

[13] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 572–588, Sep. 1994.

[14] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1530–1542, Dec. 2003.

[15] R. Xiong, J. Xu, F. Wu, and S. Li, "Barbell-lifting based 3-D wavelet coding scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1256–1269, Sep. 2007.

[16] X. Peng, J. Xu, and F. Wu, "Line-cast: Line-based semi-analog broadcasting of satellite images," in *Proc. Int. Conf. Image Process. (ICIP)*, 2012, pp. 2929–2932.

[17] F. Wu, X. Peng, and J. Xu, "LineCast: Line-based distributed coding and transmission for broadcasting satellite images," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1015–1027, Mar. 2014.

[18] L. Yu, H. Li, and W. Li, "Wireless scalable video coding using a hybrid digital-analog scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 331–345, Feb. 2013.

[19] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[20] R. Xiong, H. Liu, S. Ma, X. Fan, F. Wu, and W. Gao, "G-CAST: Gradient based image softcast for perception-friendly wireless visual communication," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2014, pp. 133–142.

[21] A. Wang, B. Zeng, and H. Chen, "Wireless multicasting of video signals based on distributed compressed sensing," *Signal Process., Image Commun.*, vol. 29, no. 5, pp. 599–606, May 2014.

[22] H. Cui, C. Luo, C. W. Chen, and F. Wu, "Robust uncoded video transmission over wireless fast fading channel," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 73–81.

[23] X. Fan, F. Wu, and D. Zhao, "D-cast: DSC based soft mobile video broadcast," in *Proc. ACM Int. Conf. Mobile Ubiquitous Multimedia (MUM)*, Beijing, China, Dec. 2011, pp. 226–235.

[24] K. Ramchandran, A. Ortega, K. M. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 1, pp. 6–23, Jan. 1993.

[25] M. Bystrom and J. W. Modestino, "Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 880–890, Jun. 2000.

[26] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 340–356, Mar. 2000.

[27] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated DCT-based SNR scalable video," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1043–1052, Sep. 2002.

[28] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, Jun. 2002.

[29] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 40–53, Jan. 2006.

[30] W. Ji, Z. Li, and Y. Chen, "Joint source-channel coding and optimization for layered video broadcasting to heterogeneous devices," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 443–455, Apr. 2012.

[31] C. Singhal, S. De, R. Trestian, and G.-M. Muntean, "Joint optimization of user-experience and energy-efficiency in wireless multimedia broadcast," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1522–1535, Jul. 2014.

[32] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inf. Theory*, vol. IT-49, no. 3, pp. 626–643, Mar. 2003.

[33] Q. Xu, V. Stankovic, and Z. Xiong, "Distributed joint source-channel coding of video using raptor codes," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 851–861, May 2007.

[34] Y. Zhang, C. Zhu, and K.-H. Yap, "A joint source-channel video coding scheme based on distributed source coding," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1648–1656, Dec. 2008.

[35] Y. Kochman and R. Zamir, "Joint Wyner–Ziv/dirty-paper coding by modulo-lattice modulation," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4878–4889, Nov. 2009.

[36] R. Xiong, F. Wu, J. Xu, and W. Gao, "Performance analysis of transform in uncoded wireless visual communication," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 1159–1162.

[37] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang, "ParCast: Soft video delivery in MIMO-OFDM WLANs," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 233–244.

[38] R. Xiong, F. Wu, X. Fan, C. Luo, S. Ma, and W. Gao, "Power-distortion optimization for wireless image/video SoftCast by transform coefficients energy modeling with adaptive chunk division," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Nov. 2013, pp. 1–6.

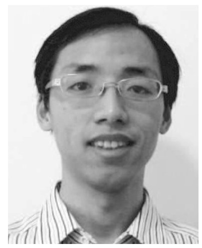
[39] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.

[40] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss–Markov sources," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 82–93, Jan. 1990.

[41] *x264: A Free H.264/AVC Encoder*. [Online]. Available: http://www.videolan.org/developers/x264.html, Jun 2014.

Xiaopeng Fan (M'09) received the B.S. and M.S. degrees from Harbin Institute of Technology (HIT), Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree from Hong Kong University of Science and Technology, Hong Kong, in 2009.

He was with Intel China Software Laboratory, Shanghai, China, as a Software Engineer, from 2003 to 2005. He joined the School of Computer Science and Technology, HIT, in 2009, where he is currently a Professor. He has authored or co-authored over 70 technical journal and conference papers. His research interests include image/videocoding and processing, video streaming, and wireless communication.

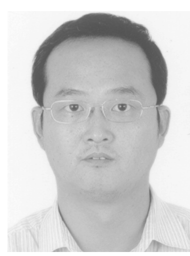


Ruiqin Xiong (M'08) received the B.S. degree from University of Science and Technology of China, Hefei, China, in 2001 and the Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2007.

He was with Microsoft Research Asia, Beijing, as a Research Intern from 2002 to 2007 and with University of New South Wales, Sydney, NSW, Australia, as a Senior Research Associate, from 2007 to 2009. He joined Peking University, Beijing, in 2010. His research interests include image and video compression, image and video restoration, joint source channel coding, and multimedia communication.

Debin Zhao (M'11) received the B.S., M.S., and Ph.D. degrees from Harbin Institute of Technology (HIT), Harbin, China, in 1985, 1988, and 1998, respectively, all in computer science.

He is a Professor with the Department of Computer Science, HIT. He has authored over 300 technical articles in refereed journals and conference proceedings. His research interests include image and video coding, video processing, videostreaming and transmission, and pattern recognition.



Feng Wu (M'99–SM'06–F'13) received the B.S. degree in electrical engineering from Xidian University, Xi'an, China, in 1992, and the M.S. and Ph.D. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1996 and 1999, respectively.

He was a Principle Researcher and a Research Manager with Microsoft Research Asia, Beijing, China. He has authored or co-authored over 200 papers, including several dozen IEEE TRANSACTIONS papers and top conference papers in the International Conference on Mobile Computing and Networking, the Special Interest Group on Information Retrieval, Computer Vision and Pattern Recognition, and ACM Multimedia. He is currently a Professor with University of Science and Technology of China, Hefei, China. He holds 77 U.S. patents. His 15 techniques have been adopted into international video coding standards. His research interests include image and video compression, media communication, and media analysis and synthesis.

Dr. Wu received the best paper award in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2009, the Pacific-Rim Conference on Multimedia in 2008, and the Society of Photo-Optical Instrumentation Engineers Visual Communications and Image Processing in 2007. He also received the Best Associate Editor Award from the IEEE Circuits and Systems Society in 2012. He was the Technical Program Committee Chair of Multimedia Signal Processing in 2011, Visual Communications and Image Processing in 2010, and Pacific-Rim Conference on Multimedia in 2009, and the Special Session Chair of the International Conference on Multimedia and Expo in 2010, and the IEEE International Symposium on Circuits and Systems in 2013. He was an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, and several other international journals.