Cactus: A Hybrid Digital-Analog Wireless Video Communication System

Hao Cui
University of Science and Technology of China
Hefei, 230027, P.R. China
hao.cui@live.com

Zhihai Song
Peking University
Beijing, 100871, P.R. China
zhhsng@gmail.com

Zhe Yang
Northeastern University
Shenyang, 110004, P.R. China
yangzhe.research@gmail.com

Chong Luo
Microsoft Research Asia
Beijing, 100080, P.R. China
chong.luo@microsoft.com

Ruiqin Xiong
Peking University
Beijing, 100871, P.R. China
rqxiong@gmail.com

Feng Wu
Microsoft Research Asia
Beijing, 100080, P.R. China
fengwu@microsoft.com

ABSTRACT

This paper challenges the conventional wisdom that video redundancy should be removed as much as possible for efficient communications. We discover that, by keeping spatial redundancy at the sender and properly utilizing it at the receiver, we can build a more robust and even more efficient wireless video communication system than existing ones.

In the proposed framework, inter-frame (temporal) redundancy in video is removed at the encoder, but intra-frame (spatial) redundancy is retained. In doing so, pixel values after a transform-domain scaling are directly transmitted with amplitude modulation. At the receiver, spatial redundancy is utilized by image denoising. Note that denoising in our decoder is not a post-processing, but have to be immediately performed on channel output. We implement the video communication system which we call *Cactus* on SORA platform. The image denoising processing is made real-time through GPU implementation. Cactus is extensively evaluated in 802.11a/g WLAN environment. On average, Cactus outperforms SoftCast by 4.7 dB in video PSNR and is robust to packet losses.

1. INTRODUCTION

In 2011, mobile video traffic exceeded 50 percent of mobile traffic for the first time, and it is predicted to increase 25-fold in the next five years, according to Cisco Visual Networking Index (VNI) [6]. Wireless video communications are facing a dilemma in achieving efficiency and robustness. On one hand, videos in their raw format are huge in size, and they need to be efficiently compressed for transmission. On the other, compressed video sequences have too little redundancy left, and therefore is susceptible to channel errors.

Direct application of Shannon's separation theorem [18] suggests that source redundancy should be completely removed and channel coding is responsible for adding redundancy against noise. However, joint source-channel coding (JSCC) suggests to keep certain amount of source redundancy and has been shown to achieve better performance at limited complexity and delay. This inspires us to consider the following questions: how much source redundancy should be retained in wireless video communications in order to achieve both efficiency and robustness? Is it possible to skip channel coding and completely rely on source redundancy for channel protection?

Interestingly, the answer to the second question is a resounding YES, and the answer to the first question becomes clear after we carefully examine the two types of redundancy in video and their respective characteristics. Our research finds that: 1) Inter-frame (temporal) redundancy should be removed as much as possible at the encoder for high efficiency while intra-frame (spatial) redundancy should be retained to protect videos against channel noise. 2) Residual frames should be transmitted in spatial domain (e.g. scaled pixel values) instead of transform domain (i.e. coefficients) through analog transmission to combat losses and noises. 3) The key to fully utilize the source redundancy is to perform image denoising at the decoder based on both source and channel characteristics.

Based on these findings, we propose a hybrid digital-analog video communication system called *Cactus*. At the sender, temporal redundancy is removed by motion-compensated temporal filtering [5]. The motion information is entropy coded into digital bits and protected by the strongest channel codes during transmission. Pixel values in residual frames are transmitted using amplitude modulation. In order to minimize the mean squared error (MSE) under the average power constraint, a transform-domain scaling is performed. However, we emphasize that the sender should transmit scaled pixel values instead of transform-domain coefficients. This allows the receiver to fully utilize the source redundancy through applying image denoising techniques. In particular, Cactus employs median filter [17] to deal with packet losses and BM3D [7] to deal with additive noises.

We have implemented Cactus on SORA [20] platform and have evaluated it in 802.11a/g-based wireless LAN environments. Evaluation results confirms that our design achieves high received video quality and is robust to channel vari-

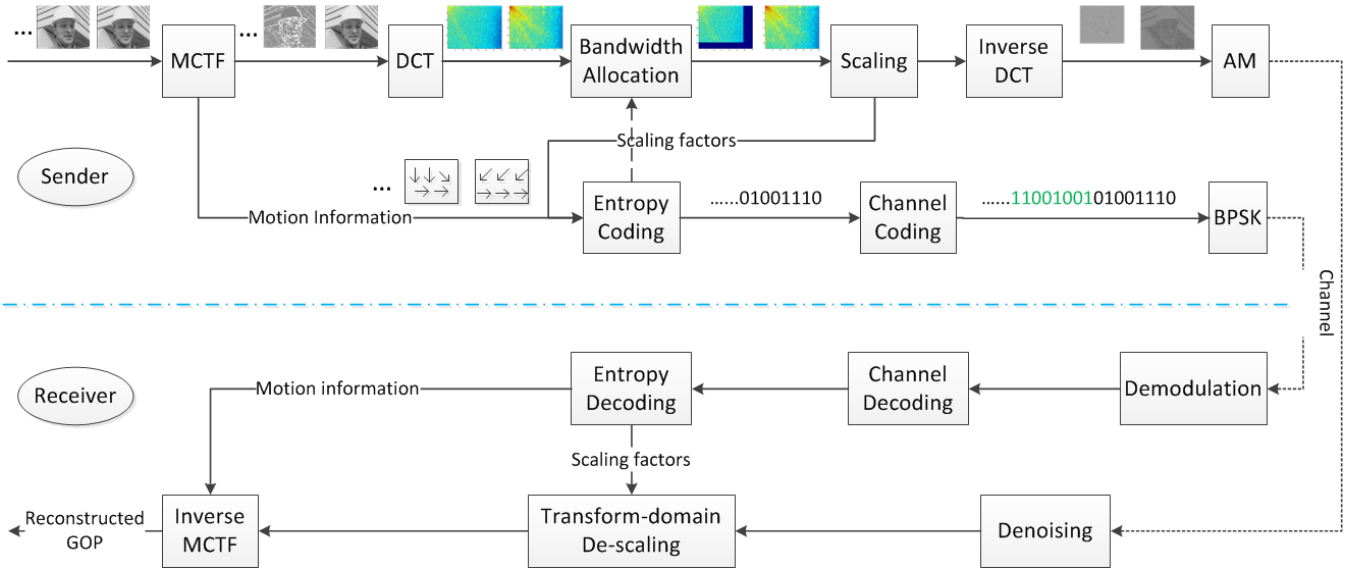


Figure 1: Cactus overview

ations. Besides, Cactus allows for graceful degradation in a wide range of receiver SNRs, and therefore can be readily used for multicasting. Trace-driven experiments show that Cactus outperforms a recent analog mobile video system SoftCast [12] by 4.7 dB in average video PSNR.

The rest of this paper is organized as follows. In Section 2, the Cactus system design is presented. Section 3 presents the implementation of Cactus on SORA and the GPU implementation of BM3D algorithm. Section 4 presents the evaluation of Cactus and provides the performance comparison against reference schemes. We discuss related works on joint source-channel coding (JSCC) in Section 5 before we finally summarize in Section 6.

2. SYSTEM DESIGN

2.1 System Overview

We are seeking for a joint source channel coding design for wireless video communications. The intuition behind is that source redundancy may be used for channel protection, but the tradeoff between robustness and efficiency needs to be balanced. Fig.1 provides an overview of the designed hybrid digital-analog communication system named Cactus.

At the sender, a video sequence is first divided into group of pictures (GOP). Commonly used GOP sizes vary from 4, 8, 16 to 30, 32 depending on the application requirements. We select GOP size equaling to 8 in our system. Each GOP is first de-correlated in the temporal axis via motion-compensated temporal filtering (MCTF). The motion information, including mode and motion vector, needs to be faithfully received by every receiver, so they are entropy coded and transmitted using a robust digital scheme. We adopt the combination of 1/2-rate channel coding and BPSK modulation.

The temporally filtered frames are then transformed into frequency domain by DCT. According to the remaining channel bandwidth budget, a certain portion of the coefficients need to be discarded. This resource allocation is performed on GOP basis. The remaining coefficients in each frame are then divided into 10 L-shaped chunks, and being scaled accordingly. The scaling factors are transmitted through digital methods too.

Finally, inverse DCT is performed on each frame. This is a key step in our design in order to fully utilize the spatial redundancy, because the loss of pixels is more friendly to image denoising algorithms than the loss of frequency coefficients. The scaled pixel values are interleaved and transmitted with amplitude modulation. In particular, every two pixel values are transmitted as the I and Q components of a complex symbol. It should be noted that the amplitude modulation we used is actually *pseudo-analog*, because we use a discrete modulation constellation, except that it is much denser than the commonly used 16-QAM or 64-QAM. This pseudo-analog implementation allows our design to be easily integrated into existing network stack.

At the receiver, the digitally transmitted symbols are processed with a sequence of inverse operations including demodulation, channel decoding, and entropy decoding. Correct motion information and metadata can be obtained. Meanwhile, the receiver directly reads the scaled pixel values from the I/Q components of wireless symbols, and pieces together all the frames. Denoising is immediately applied on the scaled frames. Then transform-domain descaling is performed for each individual frame. Finally, frames from the same GOP are processed with inverse MCTF to output the reconstructed video sequence.

2.2 Sender Design

2.2.1 *Reduction of Temporal Redundancy*

For natural video sequences, motion compensation (MC) is an essential step to remove temporal redundancy. However, the MC in present video coding standards [21] is based on a closed-loop prediction, i.e. the prediction is based on the reconstructed frame at the decoder not the original frames. In conventional digital transmission paradigm, the transmission is assumed to be lossless if channel coding provides enough protection. Thus, the encoder could im-

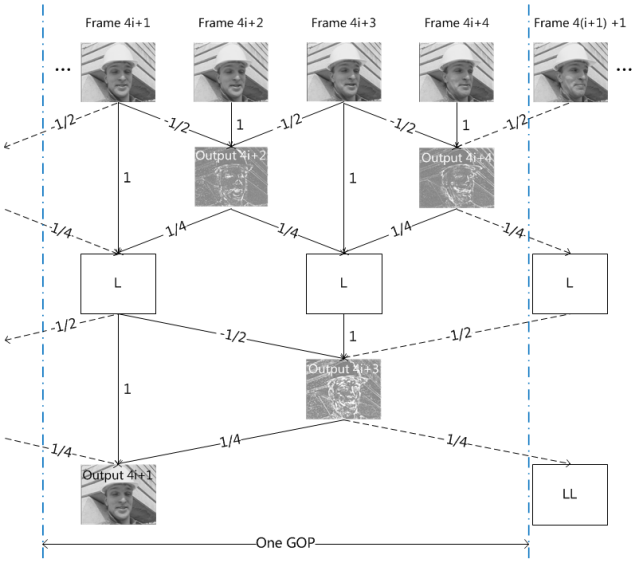


Figure 2: Lifting structure of a 2-layer 5/3 temporal filter for GOP size 4

plement a decoder, and create the reconstruction from the bit stream it generated. However, in a hybrid digital-analog transmission scheme, the encoder is not able know the exact reconstructed frame at the receiver, not to mention that in a multicast session different receivers will have different reconstructions. In this case, the closed-loop prediction will bring drifting errors.

In our system, we adopt an alternative approach called MCTF [5] to reduce temporal redundancy. MCTF is essentially motion-aligned temporal transform. It is attractive to our system because it is based on an open-loop prediction model, i.e. the prediction is based on original pixel values not the reconstructed ones. It has been shown that, the drifting errors are much smaller than its closed-loop counterpart.

Fig.2 demonstrates the lifting structure of a 2-layer 5/3 temporal filter for the i^{th} GOP when the GOP size is 4. The even frames (frame $4i+2$ and $4i+4$) are set as high-pass frames. For each block in a high-pass frame, two similar blocks are identified in the previous and following frames. The average of these two blocks creates a prediction of the current block, so that the high-pass component is computed by subtracting the prediction from the current block. After the first-layer high-pass frames are generated, the first-layer low-pass frames can be computed by adding one fourth of the high-pass components from the two adjacent frames to the current frame. It can be seen that each high-pass frame is generated from 3 original frames and each low-pass frame is generated from 5 original frames, so this process is called 5/3 filter. Similar processing steps are applied to the two low-pass frames to perform the second layer temporal filtering. We implement the barbell lifting MCTF proposed by Xiong et al. [22], and perform 3-layer filtering for each 8-frame GOP.

2.2.2 Bandwidth Allocation and Reduction

We define bandwidth ratio, denoted by ρ, as the ratio of channel bandwidth to source bandwidth. In our system, the digital transmission of motion information will occupy a certain portion of bandwidth. The exact amount can be computed from the result of entropy coding. When (BPSK, 1/2) is used, each entropy coded bit takes two complex symbols to transmit. The remaining bandwidth, denoted by the ratio ρ_c, is used to transmit pixels. When $\rho_c < 1$, not all pixel values can be transmitted, and the sender needs to decide how to reduce the bandwidth usage and how to allocate bandwidth among frames.

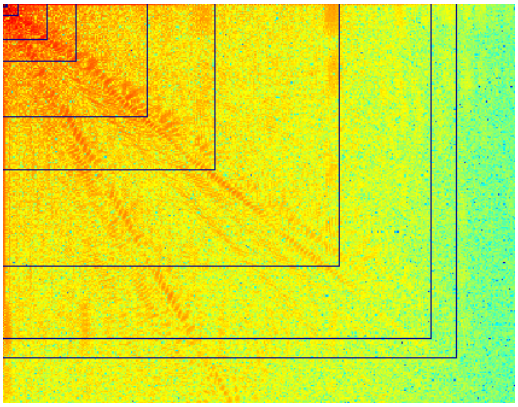


Figure 3: L-shaped chunk division for the first frame of Foreman

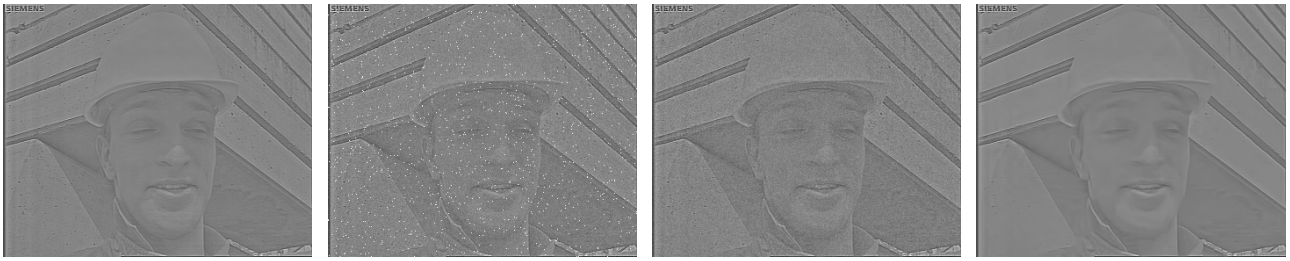
It is well understood from digital image/video coding that the truncation of data should be based on *energy*. Therefore, we perform DCT for each individual frame. As the low-pass and high-pass frames in a GOP differ drastically in energy, the bandwidth allocation should be per GOP basis. A straightforward solution, which divides the transform coefficients into equal-sized blocks and discards the least-energy blocks, cannot be applied in our design. This is because we transmit scaled pixel values instead of DCT coefficients. Even though a right portion of DCT coefficients are discarded and padded with zeros, the number of pixels after inverse DCT does not change.

We solve this problem by transmitting a down-sampled frame. It is based on an interesting property of DCT. Let I be an image with resolution $W \times H$, and C be its DCT coefficients. If we truncate C into a $W' \times H'$ matrix C' where $C'(w,h) = C(w,h)$ for all $1 \le w \le W'$ and $1 \le h \le H'$, then the inverse DCT transform of C' using a $W' \times H'$ transform matrix will create $I'_{W' \times H'}$, which is a down-sampled image of I. Therefore, transmitting I' instead of I achieves bandwidth reduction.

2.2.3 L-shaped Chunk Division and Scaling

To optimally transmit the pixels under MSE criterion in a power-constrained system, one should first de-correlate the pixel values through transform, then each transform coefficient should be scaled by a factor which is inversely proportional to the fourth root of its variance [15]. As it is not practical to scale each coefficient individually, Jakubczak and Katabi [12] propose to group nearby coefficients into chunks and model the values in each chunk as random variables (RVs) from the same distribution. Then the coefficients in the same chunk will be scaled by the same factor. The scaling factors (which is also called metadata) need to be reliably transmitted to the receiver for decoding.

We propose a new adaptive L-shaped chunk division method.



(a) Channel input (b) Channel output (c) After denoising by median filter (d) After denoising by BM3D

Figure 4: Transmitting a low-pass frame of Foreman over a 5dB AWGN channel and 1% loss rate

The motivations are two-fold. First, in the previous step of our system, bandwidth deduction will discard L-shaped coefficients from the peripheral of the frame. Second, we observe that transform coefficients decay rapidly from low-frequency to high-frequency, and those belonging to similar frequency band are more likely to have similar values.

The problem can be mathematically described as follows. Let P be the total power budget. Divide the transform coefficients into M chunks, and let λ_i and g_i denote the variance and scaling factor of the i^{th} chunk. It is known that:

$$g_i = \lambda_i^{-\frac{1}{4}} \sqrt{\frac{P}{\sum_i \sqrt{\lambda_i}}} \quad (1)$$

An optimal chunk division should minimize $\sum_i \sqrt{\lambda_i}$. For L-shaped chunk division, the adjustable parameters are r_j ($j = 1, 2, ...M-1$), which are the positions of chunk boundaries.

We adopt an iterative approach to search for the optimal set of $\{r_j\}$. The initial values of $r_j's$ are evenly spaced. Then the algorithm iteratively updates the parameters one by one. In updating r_j, the values of r_{j-1} and r_{j+1} are fixed. Fig.3 shows our chunk division for the first frame of *foreman* when $M = 10$. In this case, only 20 metadata (10 scaling factors and 10 chunk boundaries) need to be transmitted.

Actually, both bandwidth reduction and power scaling are performed on transform domain. Therefore, the sender should perform IDCT after these two steps. Transmitting scaled pixel values does not change the overall power because IDCT is an orthonormal transform. Fig.4(a) shows the channel input for the first frame of *foreman*. The original frame is 8-bit grayscale (pixel values are from 0 to 255). After transform-domain scaling, the pixel values range from -8.82 to 10.46 for this particular frame. We amplify each value by 10 times (then plus the shift 128) just for the viewing purpose.

2.3 Receiver Design

One key finding in our research is that source redundancy can provide channel protection under the premise that it is fully utilized at the receiver. We propose to use image denoising technqiues at the receiver, and emphasize that denoising should be immediately applied to channel output.

The denoising processes for low-pass and high-pass frames are identical. We use different denoising techniques to deal with packet losses and random-valued noises. In particular, we adopt the classic median filter [17] to handle losses. Under ideal interleaving, packet loss creates randomly dispersed pixel "holes" in the frame. These holes are filled with the median of surrounding eight pixel values. We have tried more advanced median filter such as directional weighted median filter [8], but the performance improvement is marginal at moderate packet loss ratios.

Then BM3D [7] is adopted to reduce the random noise for two reasons. First, BM3D is the state-of-the-art denoising algorithm. Second, there is a video version of BM3D which utilizes temporal redundancy to denoise. This provides an alternative to our MCTF design and could help us to evaluate whether and in which cases temporal redundancy should be removed at the encoder.

The complete BM3D algorithm has two estimate steps: basic estimate and final estimate. Each estimate is again composed of two steps: block-wise estimate and aggregation. In block-wise estimate, each block find similar blocks in a large neighborhood and stack them in a 3D array. Then, 3D transformation, hard thresholding (Weiner filtering in final estimate), and inverse 3D transformation are consecutively performed to generate estimates for all the involved pixels. After all the blocks are processed, overlapping estimates are aggregated through weighted sum operation.

Fig.4 uses an example to illustrate the denoising process in our decoder. We assume an AWGN channel with 5 dB receiver SNR and additional 1% loss rate. Fig.4(b) shows the channel output where white dots indicate the lost pixels. The entire image is contaminated with noise, but interestingly, most image features are still recognizable. This phenomenon supports our argument that spatial redundancy can provide channel protection, and image denoising is the necessary step to utilize such redundancy. Fig.4(c) and Fig.4(d) show the result after median filter and BM3D respectively. The resulting image is very similar to channel input.

After denoising, transform-domain de-scaling is performed on each frame. This is accomplished by DCT transform, scaling, and inverse DCT transform. If the frame size is smaller than the regular size, indicating a portion of the coefficients have been dropped, the decoder will pad zeros to form a frame of regular size, then perform inverse DCT. The de-scaled frames and decoded motion information will then be used to reconstruct the GOP by inverse MCTF.

3. IMPLEMENTATION

3.1 Cactus Implementation

The Cactus system is composed of application-layer CODEC (coder and decoder) and physical-layer modules. Cactus encoder only needs the available bandwidth information from

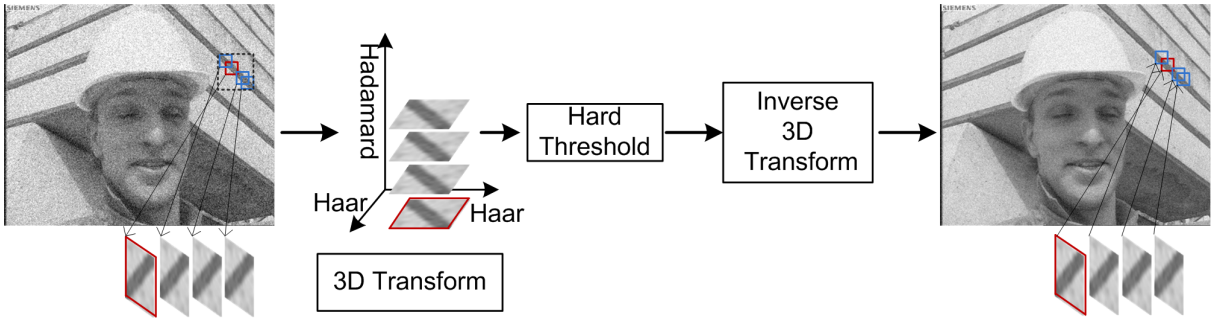


Figure 5: GPU implementation of the basic estimate step of BM3D algorithm

the channel, which can be pre-determined. In the Cactus encoder, we use a reference C code for MCTF, and implement all the other modules, including transform, bandwidth allocation and reduction, L-shaped chunk division and scaling, entropy coding, and channel coding, in Matlab. All the modules except MCTF could process CIF (352×288) videos in real-time. However, we believe that MCTF can be run in real-time, because it has very similar processing steps and complexity as hierarchical-B coding structure in H.264 [22]. The latter already has a real-time implementation x.264 [2], which can encode four or more 1080p streams in real-time on a single consumer-level computer. In particular, the two schemes have similar computational complexity in motion estimation step (both find motions in previous and following frames), which is known to be the most time-consuming module in a video encoder.

We implement Cactus on OFDM PHY defined in IEEE 802.11a/g. Specifically, the channel is divided into 64 subcarriers and 48 of them are used to transmit modulation symbols. To reduce the overhead of PLCP header, we use 100 OFDM symbols in each PLCP frame for data transmission. Therefore, the total number of modulation symbols in each transmission is 4800. Metadata is transmitted at the lowest rate 0.5 bits/s/Hz (BPSK with 1/2 coding) in 802.11a/g.

To resist packet loss, the adjacent symbols from a picture are pseudo-randomly shuffled across different PLCP frames. We limit the shuffling in a GOP of video frames to reduce the decoding delay. We generate the shuffle mapping through sorting a set of random numbers between 0 and 1. The sorted index is used for shuffle mapping. The random numbers can be produced by a pre-defined random seed and it does not introduce additional overhead. The shuffled symbols are sequentially placed on each OFDM symbol. Therefore, when a PLCP frame is lost, it creates randomly dispersed "holes" in the video frame, which can be easily processed by median filter.

In the Cactus decoder, we implement channel decoding, entropy decoding, and transform-domain de-scaling in Matlab. The inverse MCTF has a much lower computational complexity than MCTF encoder. Therefore, the decoder also can be implemented in real-time. We use the median function in Matlab to perform the median filter denoising, and use the Matlab code published by the authors [1] to perform BM3D denoising for all the evaluations. The processing time for one CIF video frame is around 1.4 seconds using Intel Core Quad CPU (Q9550) 2.83GHz.

Module	Data	Size (KB)	Cores
Block Matching	1 block	1.5	192
Haar	24 blocks	6.1	192
Hadamard	48 blocks	12.3	192
Inverse Hadamard	48 blocks	12.3	192
Inverse Haar	16 blocks	4	128
Blending	384 blocks	9.3	192

Table 1: Memory and core usage in one SM by each BM3D processing step

3.2 GPU Implementation of BM3D

We notice that the current implementation of BM3D has a high computational complexity. Fortunately, it is very suitable for parallel computing (e.g., a specially designed chip, a FPGA or a GPU). In order to validate that our system can run in real-time, we implement BM3D through GPU NVIDIA GTX680. It has 8 streaming multiprocessors (SMs). Each SM has 192 cores and 64KB shared memory in which 48KB can be used to store data. We implement BM3D in GPU following two optimization rules. The first rule is to fully utilize all 192 cores. Second, because accessing the display memory (size up to 2GB) is slow, the data processed by the 192 cores should not exceed the SM's memory size which is 48KB.

We implement the basic estimate step of BM3D as shown in Fig.5. Every 8x8 block looks for matching blocks in a given rectangle region. The original block and matched blocks are organized in a 3D array. Then it is transformed by 2D Haar and 1D Hadamard transform. The noise is removed by hard thresholding. Finally, inverse transforms are performed, and pixel values corresponding to the same position are aggregated. Table 1 shows the memory usage, core usage, and involved data size for each SM. It can be seen that we make full use of 192 cores in almost all the processing steps. All the eight SMs perform identical operations.

We evaluate our GPU implementation of BM3D over 16 CIF test video sequences under 5 dB AWGN channel. The denoising results are listed in Table 2. The anchor results in the second column is achieved by the official Matlab code [1]. On average, our implementation has 0.19 dB loss in video PSNR. This is due to two simplifications. First, we do not implement the final estimation step of the complete BM3D algorithm. Second, the original 2D bi-orthogonal transform is replaced by Haar wavelet transform. The last column in the table shows the processing speed in fps. On average, GPU can process CIF videos at the speed of 35 fps, which

Sequence	Anchor (dB)	GPU (dB)	Speed (fps)
1	42.92	42.33	33.60
2	32.32	32.20	35.60
3	35.12	34.89	33.97
4	39.63	39.53	35.00
5	32.32	32.23	35.07
6	31.97	31.89	39.46
7	37.48	36.97	34.83
8	32.95	32.81	34.46
9	26.79	26.81	34.76
10	38.54	38.20	35.66
11	38.80	38.53	34.50
12	36.03	35.88	35.49
13	33.53	33.32	35.75
14	33.20	33.31	34.43
15	35.62	35.52	34.27
16	38.04	37.79	33.64
Average	35.33	35.14	35.03

Table 2: Reconstructed PSNR and speed of GPU implementaion for CIF videos under 5 dB AWGN channel

is almost 50x of the CPU speed. It verifies the feasibility to use BM3D as part of a real-time video communication system.

4. EVALUATION

4.1 Settings

Wireless environment: Evaluations are carried out with SORA (equipped with WARP radio board) over 802.11a/g-based WLAN. The carrier frequency is 2.4GHz. The channel bandwidth is 12MHz and the data bandwidth is around 11.4MHz. We define bandwidth ratio ρ as the ratio of channel bandwidth to source bandwidth.

We perform 52 test runs. In each test run, we transmit data generated by Cactus encoder. The receiver records the received wireless symbols. These symbols are not only used for Cactus decoding, but are compared with the exact channel inputs to generate the traced channel noise. The traced data are labeled from 1 to 52 (according to the time they are obtained). Our comparisons with reference schemes are trace-driven to ensure fairness.

The effect of packet loss is evaluated by assuming an interferer who sends packets at constant intervals.

Video Source: We create two monochrome video sequences of different resolutions for our evaluation. The CIF sequence has a resolution of 352×288, and the frame rate is 30 fps (frame per second). Hence, the source bandwidth is 1.52 MHz (in complex symbols). This sequence is created by extracting the first 32 frames from the following 16 standard video test sequences including *akiyo, bus, coastguard, crew, ower, football, foreman, harbour, husky, ice, news, soccer, stefan, tempete, tennis, waterfall.* Hence, it has 512 frames in total. It is similar to the test sequence used in SoftCast [12], with the only difference that the resolution used in SoftCast is 352×240.

The other HD (720p) sequence has a resolution of 1280×720, and the frame rate is 30 fps too. Hence, the source bandwidth is 13.8 MHz. In order to transmit it in a 11.4 MHz channel, bandwidth compaction is needed and the ratio is around 0.826. This sequence contains the first 32 frames from 10 standard video test sequences, including *Intotree, Shields, Stockholm, City, Jets, Panslow, Parkrun, Sheriff, ShuttleStart, Spincalendar.* Therefore, the total length is 320 frames.

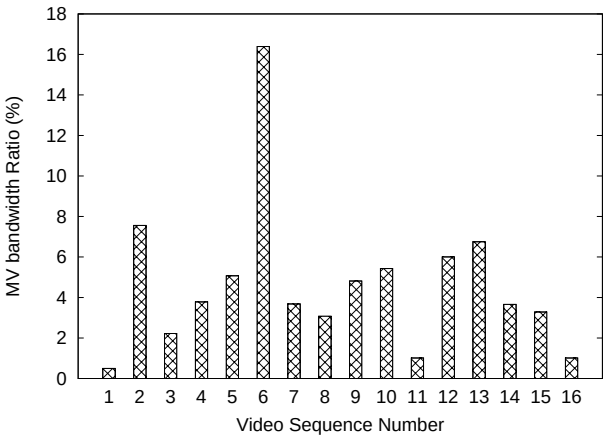


Figure 6: The bandwidth percentage used by transmitting motion information in (BPSK, 1/2)

Reference Schemes: Two reference schemes are considered. The first one is SoftCast. Our implementation has only one difference from that described in [12]. In order for a fair comparison with Cactus which uses GOP size 8, the GOP size of SoftCast is set to 8 too. We actually evaluated both schemes when GOP size is increased to 16, and found that both schemes will have 0.3-0.5 dB performance gain in PSNR. SoftCast needs to transmit metadata with digital method too. There are 64 variances per each video frame. We do not actually transmit these metadata for SoftCast.

The other reference is based on H.264 or MPEG4 AVC digital video coding standard [21]. We adopt a publicly available encoder called x264 [2] to encode test sequences at different rates and obtain a R-D (rate-distortion) curve. Similarly, GOP size is set to 8 for fairness. In the case of multicast, we simply call it MPEG. In the case of multicast, we name it Omni-MPEG because we assume the sender in this scheme can immediately obtain the SNR of the previous packet, and use this SNR to guide the rate selection of the next packet. The possible rates are those defined in 802.11a/g. We then calculate the Goodput rate for an entire test run, and find the corresponding distortion from the R-D curve, as if the encoder had known the channel conditions in advance. The performance of Omni-MPEG provides an upper bound for the conventional digital schemes in Unicast.

We do not compare with the Scalable Video Coding (SVC) extension of H.264/AVC because it has been shown in SoftCast that the performance is inferior to SoftCast in all cases.

Performance metric: We evaluate the video delivery quality with the standard peak signal-to-noise ratio (PSNR) in dB. We compute the PSNR for each video frame by $PSNR = 10log_{10}\frac{255^2}{MSE}$, where MSE is the mean squared error of all pixels. Then the PSNR is averaged across frames.

4.2 Micro-benchmarks

Micro-benchmarks verify our design choices. The results

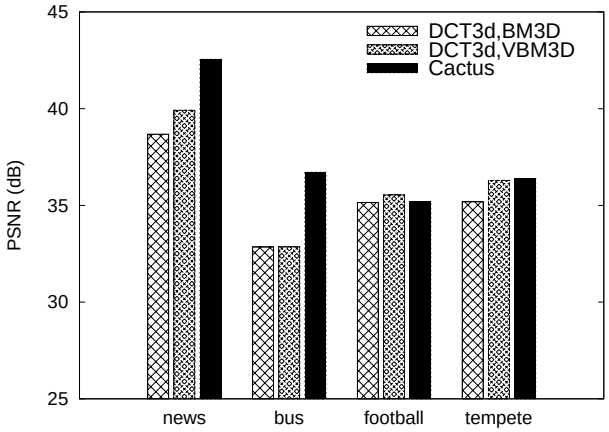


Figure 7: Four representative sequences which show different Compare three schemes which use temporal redundancy differently

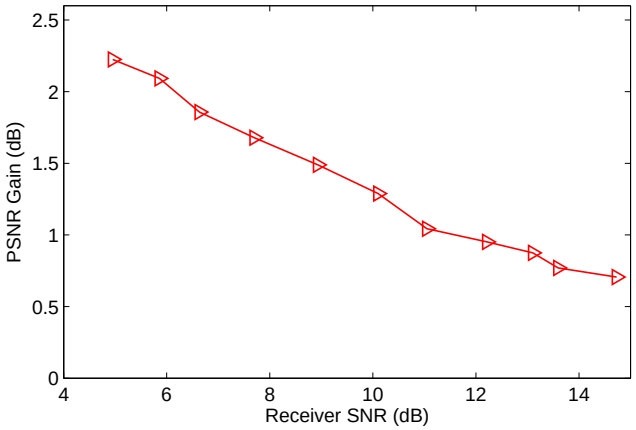


Figure 8: Denoising gain as a function of receiver SNR

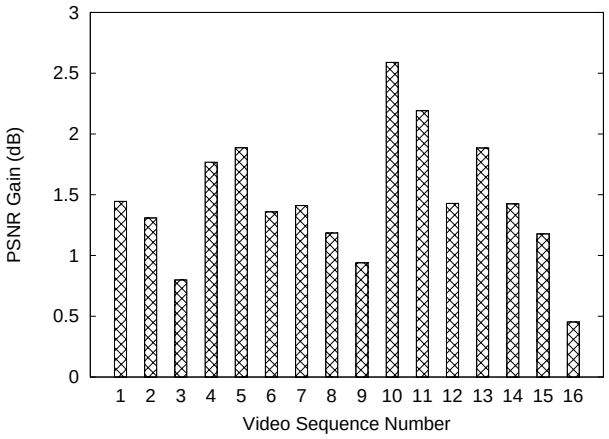


Figure 9: Denoising gain on different sequences

are obtained with the CIF video sequence, and when the bandwidth ratio ρ is 1, if not otherwise stated.

Use of temporal redundancy. Cactus removes temporal redundancy by MCTF, and encodes the motion information into digital streams. The digital stream will share bandwidth with the analog transmission of scaled pixel values. We first examine the bandwidth percentage of motion information assuming a very robust transmission scheme (BPSK, 1/2), and check whether the use of bandwidth is worthy. Fig. 6 shows the bandwidth ratio and we can see that the amount of motion information differs greatly among sequences. Sequence #6 (*football*) has very complex motions, while Sequence #1 (*akiyo*) and #11 (*news*) have simple or small motions.

Then we verify our claim that temporal redundancy should be removed at the encoder, and examine in what cases this claim does not hold. We compare the final design of Cactus (MCTF at encoder and BM3D at decoder) against two alternatives. Method (DCT3d, BM3D) uses 3D-DCT at the encoder. Hence the temporal redundancy is not fully exploited. Method (DCT3d, VBM3D) exploits temporal redundancy at the decoder by using video BM3D. We have mentioned earlier that video BM3D searches for matching blocks not only in current frame but in adjacent frames as well. Both alternatives have more bandwidth than Cactus to transmit coefficients, because the encoder does not generate motion information.

We run this test over Trace 6 (average receiver SNR = 9.97 dB) for all 16 CIF sequences. On average, Cactus has 1.67 dB gain over (DCT3d, VBM3D), and the latter has 2.39 dB gain over (DCT3d, BM3D). Fig. 7 presents four representative sequences. Most of the skipped sequences share the same trend as *news*. For this sequence, Cactus performs the best, and using video BM3D brings certain gain over using BM3D. Both *bus* and *ice* show an interesting result that video BM3D does not bring any gain over BM3D. This means that most patches in the video have enough similar patches in the same frame to smooth out the noise. This is exactly the case in *bus* where the trees, fences and bricks have similar patterns, and in *ice* too where the ice patches resemble each other.

The *football* is the only sequence that Cactus does not perform the best. This is because the motion information occupies too much bandwidth (around 16%), and too many coefficients are dropped which introduces loss. This suggests that there exists a trade-off point beyond which the temporal redundancy is better utilized at the receiver.

Image denoising. Image denoising is a key module in Cactus decoder. This experiment evaluates the performance gain brought by this module, and examine the impacting factors. Fig.8 shows the average denoising gain for all 16 CIF sequences under different receiver SNRs. The gain shows a clear decreasing trend. This suggests that for a Cactus receiver, if the measured channel condition is better than a certain SNR, it may turn off the denoising module without much loss in performance.

The denoising gain also depends on the video characteristics. Fig.9 shows the PSNR gain on each of the sequences, and the value presented is averaged over all 32 frames and all traces. The sequence #10 (*ice*) benefits the most from denoising, with an average gain over 2.5 dB. This is because the frames in this sequence are all very smooth. Sequence #16 (*waterfall*) gains the least from denoising, because the frames contain too much texture and details.

Transmission in spatial domain v.s. in transform

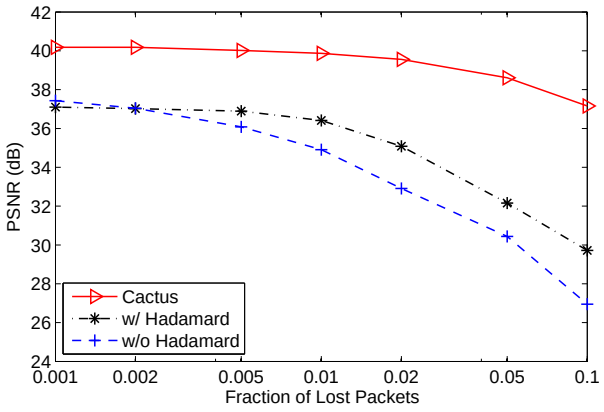


Figure 10: Comparing transmitting sequence #10 (*ice*) in spatial domain or transform domain under different loss rates

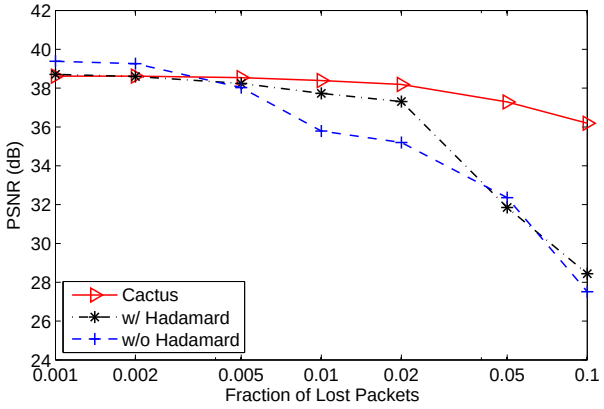


Figure 11: Comparing transmitting sequence #16 (*waterfall*) in spatial domain or transform domain under different loss rates

domain Actually, Cactus could choose between transmitting in spatial domain (scaled pixel values) or in transform domain (scaled DCT coefficients). The two choices cost the same transmission power as DCT transform is orthonormal. We have embraced the spatial-domain transmission in the final design of Cactus, and this experiment is going to verify this choice.

We compare our spatial-domain transmission with two transform-domain alternatives, one with Hadamard transform (as used in SoftCast) and the other without any transform. In this evaluation, we let $\rho_c = 1$, i.e. there are just enough channel bandwidth to transmit all the pixels or coefficients. We do this simplification because the dimension of Hadamard matrix has to be a power of 2. We run the experiments for video sequence #10 and #16 on Trace 8 (receiver SNR = 7.24 dB). These two sequences are chosen because they benefit the most and the least from image denoising as shown by the previous experiment.

Fig.10 and Fig.11 show the comparison results for sequence 10 and 16 respectively. It is not surprising that transmitting *ice* in spatial domain significantly outperforms the other two choices, since *ice* is very friendly to image denoising. The gain over the transform-domain transmission with

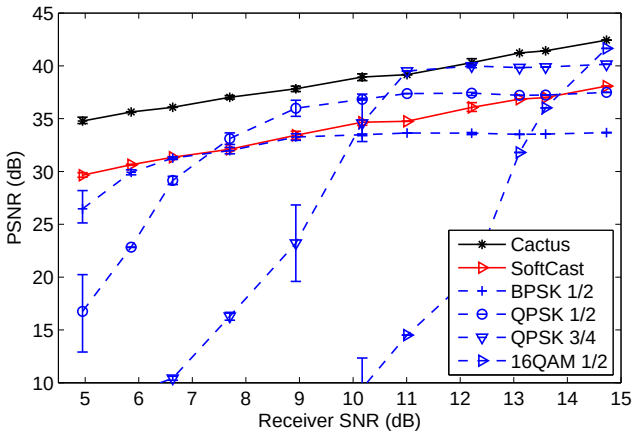


Figure 12: Compare Cactus against two reference schemes for CIF sequence in a multicast session when bandwidth ratio $\rho = 1$.

Hadamard is as high as 7.44 dB when the packet loss ratio is 0.1. The experiment on *waterfall* obtains slightly different results. When the packet loss ratio is small, transmitting the video in spatial domain does not bring any gain. However, this is the sequence which benefits the least from image denoising. Even for this sequence, Cactus outperforms the other choices in most cases. This experiment validates our choice to transmit video in spatial domain.

4.3 Comparison against Reference Systems

Fig.12 compares the performance of Cactus against two reference schemes, namely SoftCast and Omni-MPEG under multicast scenario. We run 52 traces over the CIF sequence to emulate a 52-receiver multicast session. For Cactus and SoftCast, in each test run, we compute the average receiver SNR across PHY packets and average video PSNR across sequences. To plot the video PSNR as a function of receiver SNR, we divide the receiver SNR range into 1 dB bins, and average all the (receiver SNR, PSNR) pairs whose receiver SNR fall into the same bin. Results show that although both Cactus and SoftCast achieve graceful rate adaptation, Cactus consistently outperforms SoftCast in video PSNR, and the average gain is 4.7 dB.

For MPEG, the sender has to fix a transmission rate in each test run. We run four tests, using the lower half of the PHY rates defined in 802.11a/g. Once the transmission rate is fixed, the video PSNR can be found in the R-D curve. We run each trace for each PHY rate, if the instantaneous receiver SNR is higher than expected, transmission is successful. Otherwise, the receiver can get nothing. We average the PSNR along each trace, and also plot (receiver SNR, PSNR) in bins. Fig.12 clearly shows that Cactus outperform MPEG because the latter suffers from the threshold effect.

Note that, although Cactus transmits motion information through digital method, they are always protected with the lowest rate (1/2) channel coding and transmitted using BPSK modulation. Therefore, it is insensitive to channel variations.

4.4 Robustness to Packet Loss

This experiment evaluates whether Cactus is robust to

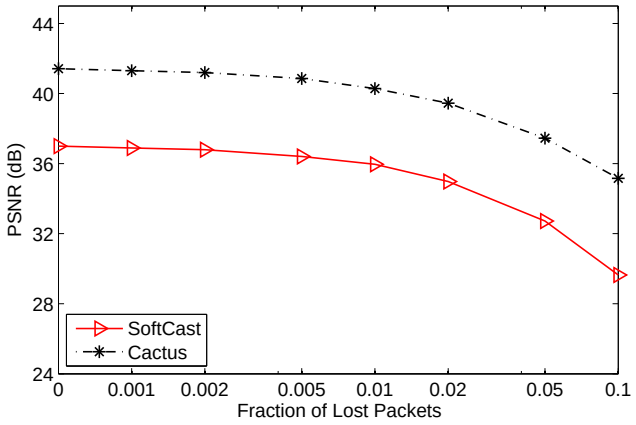


Figure 13: Comparing the error resilience capability of Cactus and SoftCast under trace #16 (receiver SNR = 13.59 dB)

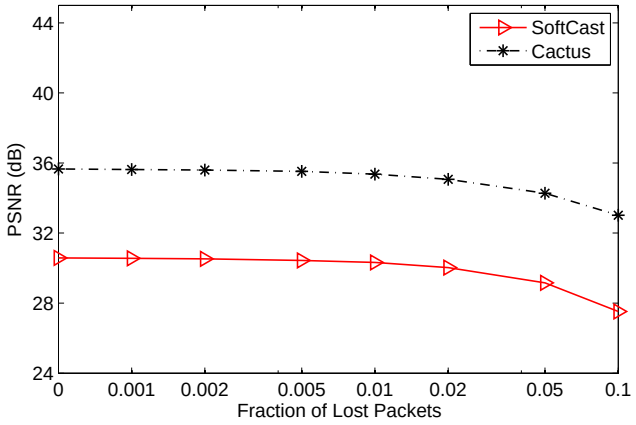


Figure 14: Comparing the error resilience capability of Cactus and SoftCast under trace #22 (receiver SNR = 5.82 dB)

packet loss. The traces used in this experiment is #16 which has a receiver SNR of 13.59 dB, and #22 which has a receiver SNR of 5.82 dB. The packet loss is simulated by assuming an interferer who transmits packets at constant intervals. We evaluate both Cactus and SoftCast when the packet loss ratios are 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, and 0.1.

Fig.13 and Fig.14 show the average video PSNR achieved by Cactus and SoftCast as a function of loss rate under channel trace #16 and #22 respectively. In both figures, the x-axis is in logarithm. When the channel condition is poor (trace #22), both Cactus and SoftCast are not sensitive to packet losses. The PSNR loss of Cactus is less than 1 dB when loss ratio is 0.02. However, when channel condition is good (trace #16), the transmissions are sensitive to losses.

Though not very obvious in the figures, when the packet loss ratio is 0.1, the PSNR loss of Cactus with respect to no loss case is 6.3 dB for trace #16 and 2.6 dB for trace #22. Under the same packet loss ratio, the PSNR loss of SoftCast with respect to its no loss case is 7.4 dB and 3.1 dB respectively. This shows that Cactus has even higher robustness to packet loss than SoftCast.

5. RELATED WORK

The design of Cactus essentially belongs to joint source-channel coding (JSCC). JSCC is an extensively studied topic both from information theoretic perspective and for the specific application on video communications.

In the category of JSCC for digital video communications, Cheung et al. [4] proposed to distribute the available source and channel coding bits among the sub-bands to minimize the expected distortion. This is analogous to the transform-domain scaling (power allocation) in Cactus design. He et al. [11] proposed a JSCC scheme which determines the optimal β in source coding, which is the percentage of blocks coded without prediction, based on the channel parameters such as bandwidth and BER. A higher β implies retaining more redundancy in the source. Flexcast [3] replaces the entropy coding module in H.264 with a rateless code, thus to achieve graceful quality degradation in video Unicast.

Research on analog JSCC mostly considers general source and channel. For instance, it is well-known that a Gaussian source achieves the capacity of an AWGN channel [10]. Gastpar et al. [9] observed that channel coding is not necessary in some cases for optimal communication, but the source and the channel have to be matched in a probabilistic sense. Compression techniques, like vector quantization (VQ) [19] and Wyner-Ziv coding [13, 16], have been adopted in hybrid digital-analog transmissions to match source and channel. Recently, Kochman and Zamir [14] showed that, by combining prediction and modulo-lattice arithmetic, one can match any stationary Gaussian source to any colored-noise Gaussian channel, hence achieve Shannon's capacity limit. They also pointed out that analog transmission scheme is more robust than its digital counterpart and is not sensitive to exact channel knowledge at the sender.

Recently, an analog mobile video system named SoftCast has been proposed [12]. In SoftCast, 3D-DCT is performed on a group of pictures, and the transform coefficients are transmitted through amplitude modulation after power scaling. SoftCast has shown that such pseudo-analog video transmission is robust to channel variations and is capable of achieving graceful degradation in a wide range of channel conditions. However, an important fact that has been neglected in SoftCast is that the retained source redundancy should be actively utilized at the receiver. Our work differs from Softcast in two main aspects. First, we adopt image denoising technique at the receiver to fully utilize the source redundancy. In doing so, we propose to transmit residual frames in pixel domain instead of transform domain, and point out that denoising should be immediately performed on channel output. Second, SoftCast uses 3D-DCT to decorrelate video in both temporal and spatial domain. All information (except scaling factors) are transmitted in analog. We discover that temporal redundancy should be removed at the sender through motion compensation. In addition, a hybrid digital-analog framework is more efficient and robust for video transmission.

6. SUMMARY AND DISCUSSION

We have described in this paper an efficient and robust wireless video communication system - Cactus. Our design validates that it is possible in video communications to skip channel coding and solely rely on source redundancy for channel protection. The encouraging results suggest the

great potential of such hybrid digital-analog scheme.

Cactus can be implemented in wireless LAN or cellular network for both Unicast and multicast video communications. Considering the fact that most video contents are stored and transmitted in compressed form, using Cactus would require the access point (AP) or the base station (BS) to partially decode the conventionally compressed video stream. Luckily, the hardware implementation of H.264/MPEG AVC codec is prevail and very cheap, so this requirement will not become an obstacle to the application of Cactus.

7. REFERENCES

[1] Bm3d algorithm and its extensions. Technical report, http://www.cs.tut.fi/ foi/GCF-BM3D/.

[2] http://www.videolan.org/developers/x264.html.

[3] S. Aditya and S. Katti. Flexcast: graceful wireless video streaming. In *Proceedings of International Conference on Mobile Computing and Networking (MobiCom'11)*, pages 277–288, 2011.

[4] G. Cheung and A. Zakhor. Bit allocation for joint source/channel coding of scalable video. *Image Processing, IEEE Transactions on*, 9(3):340 –356, mar 2000.

[5] S.-J. Choi and J. Woods. Motion-compensated 3-d subband coding of video. *Image Processing, IEEE Transactions on*, 8(2):155 –167, feb 1999.

[6] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016. Technical report, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf, Feb 2012.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080 –2095, aug. 2007.

[8] Y. Dong and S. Xu. A new directional weighted median filter for removal of random-valued impulse noise. *Signal Processing Letters, IEEE*, 14(3):193 –196, march 2007.

[9] M. Gastpar, B. Rimoldi, and M. Vetterli. To code, or not to code: lossy source-channel communication revisited. *Information Theory, IEEE Transactions on*, 49(5):1147 – 1158, may 2003.

[10] J. Goblick, T. Theoretical limitations on the transmission of data from analog sources. *Information Theory, IEEE Transactions on*, 11(4):558 – 567, oct 1965.

[11] 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. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(6):511–523, jun 2002.

[12] S. Jakubczak and D. Katabi. A cross-layer design for scalable mobile video. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 289–300, New York, NY, USA, 2011. ACM.

[13] Y. Kochman and R. Zamir. Joint wyner-ziv/dirty-paper coding by modulo-lattice modulation. *Information Theory, IEEE Transactions on*, 55:4878–4899, Nov 2009.

[14] Y. Kochman and R. Zamir. Analog matching of colored sources to colored channels. *Information Theory, IEEE Transactions on*, 57(6):3180 –3195, june 2011.

[15] K.-H. Lee and D. Petersen. Optimal linear coding for vector channels. *Communications, IEEE Transactions on*, 24(12):1283 – 1290, dec 1976.

[16] K. N. M. P. Wilson and G. Caire. Joint source channel coding with side information using hybrid digital analog codes. *Information Theory, IEEE Transactions on*, 56:4922–4940, Jul 2010.

[17] W. K. Pratt. Median filtering. Technical report, Image Processing Institute, University of Southern California, Sep. 1975.

[18] C. E. Shannon. A mathematical theory of communication. *Bell System Technology Journal*, 27:379–423, 623–656, 1948.

[19] M. Skoglund, N. Phamdo, and F. Alajaji. Design and performance of vq-based hybrid digital-analog joint source-channel codes. *Information Theory, IEEE Transactions on*, 48(3):708 –720, mar 2002.

[20] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang, and G. M. Voelker. Sora: high-performance software radio using general-purpose multi-core processors. *Commun. ACM*, 54(1):99–107, Jan. 2011.

[21] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560 –576, july 2003.

[22] R. Xiong, J. Xu, F. Wu, and S. Li. Barbell-lifting based 3-d wavelet coding scheme. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1256 –1269, sept. 2007.