SparseCast: Hybrid Digital-Analog Wireless Image Transmission Exploiting Frequency Domain Sparsity

Tze-Yang Tung and Deniz Gündüz

Abstract—**A hybrid digital-analog wireless image transmission scheme, called SparseCast, is introduced, which provides graceful degradation with channel quality. SparseCast achieves improved end-to-end reconstruction quality while reducing the bandwidth requirement by exploiting frequency domain sparsity through compressed sensing. The proposed algorithm produces a linear relationship between the channel signal-to-noise ratio (CSNR) and peak signal-to-noise ratio (PSNR), without requiring the channel state knowledge at the transmitter. This is particularly attractive when transmitting to multiple receivers or over unknown time-varying channels, as the receiver PSNR depends on the experienced channel quality, and is not bottlenecked by the worst channel. SparseCast is benchmarked against two alternative algorithms: SoftCast and BCS-SPL. Our findings show that the proposed algorithm outperforms SoftCast by approximately 3.5 dB and BCS-SPL by 15.2 dB.**

I. INTRODUCTION

Conventional wireless image/video transmission systems consist of two components: a source encoder for compression, and a channel encoder that introduces redundancy against noise and interference. This separate design is without loss of optimality according to Shannon's separation theorem, and has dominated practical implementations. Although the optimality breaks down for multi-user systems or time-varying channels, there has been a steady move towards fully digital and separate architectures (e.g., digital TV/radio) thanks to the modularity and flexibility it provides. However, many emerging applications from tactile Internet to autonomous vehicles require wireless transmission of image/video files under extreme latency, energy and complexity constraints, which preclude the use of advanced compression and channel coding techniques.

A surprising result in [1] shows that, when transmitting independent Gaussian samples over a Gaussian channel, with one sample per channel use on average, uncoded transmission, where each sample is simply scaled and transmitted, meets the theoretical Shannon bound. With digital transmission, the same performance can only be achieved by vector-quantising an arbitrarily long sequence of source samples, followed by a capacity achieving channel code. Benefits of analog transmission has since been shown in various settings [2], [3]. We highlight that, analog transmission here does not refer to traditional analog modulation techniques, i.e., amplitude or frequency modulation. Instead, it refers to a transmission scheme, in which both the source and channel encoder/decoder employ sampling; however, the samples are allowed to take continuous values, rather than being limited to a discrete set of quantised values or constellation points.

Motivated by the theoretical properties of uncoded transmission, a practical joint source-channel coding (JSCC) scheme, called SoftCast, was proposed in [4]. SoftCast applies a discrete cosine transform (DCT) on the image, and transmits the DCT coefficients directly over the channel using a dense constellation. Compression is obtained by discarding blocks of DCT coefficients whose energy is below a threshold. Index of the discarded blocks is sent as meta-data to the receiver for reconstruction. Since the encoder is linear and the coefficients are corrupted by additive noise directly, the resultant video peak signal-to-noise ratio (PSNR) is linearly related to the channel signal-to-noise ratio (CSNR), solving the *cliff effect* problem encountered in separate source and channel coding.

This letter aims to reduce the bandwidth usage in uncoded image/video transmission by utilising a novel grouping of DCT coefficients, and by incorporating compressed sensing (CS) and sparse signal recovery. CS theory demonstrates that a system of underdetermined equations can be solved with high probability if the solution is sparse [5]. This implies that if an image or a video frame can be transformed into a sparse domain, even if we send a few linear combinations of the pixel values to the receiver, it can still recover the original frame. CS has been previously used for wireless video transmission in [6] [7], where the l_1 approximation for recovery is considered. While this allows the receiver to employ convex optimisation, it may still be computationally complex for video streaming applications with strict delay constraints. Iterative algorithms have been developed to approximate the solution faster at the cost of greater error. One such algorithm, used in [8] and [9], called block CS-smooth projected Landweber (BCS-SPL) [10], achieves reconstruction through iterative thresholding. This algorithm applies CS on the pixels directly and requires no meta-data as long as the measurement matrix is agreed a priori between the transmitter and receiver. It is important to note here that in [8], optimal power allocation is not considered and an additional image processing technique was employed following BCS-SPL to improve the output image quality.

In this letter, after applying 2D-DCT on image blocks and thresholding, a novel grouping of the coefficients is applied, where coefficients of the same frequency component are grouped into vectors. We then multiply each vector with a pseudo-random measurement matrix whose size depends on the sparsity level of the corresponding vector. Finally, a scaling factor is applied to the results of this multiplication, which corresponds to power allocation across different frequency components. The receiver employs a combination of approxi-

T.-Y. Tung is with the University of Southern California Viterbi School of Engineering, Los Angeles, CA 90089 USA (email: tzeyangt@usc.edu).

D. Gündüz is with the Department of Electrical and Electronics Engineering, Imperial College London, London SW7 2AZ, U.K. (email: d.gunduz@imperial.ac.uk)

mate message passing (AMP) [11] and minimum mean squared error (MMSE) estimation. AMP is a low-complexity iterative thresholding algorithm for CS recovery, which does not need to know the exact positions of the nonzero elements in the sparse vector. It converges exponentially and does not assume any prior distribution on the data, incurring a minimal computation overhead.

The benefits of SparseCast can be summarised as follows: i) Instead of removing blocks as in SoftCast, it uses thresholding on individual coefficients and CS to reduce the channel bandwidth; ii) By grouping coefficients of different blocks according to their frequencies, it better exploits the sparsity of higher frequency components; iii) Unlike other CS-based image/video transmission schemes, it employs power allocation according to empirical variances, which significantly improves the reconstruction quality.

II. PROPOSED ALGORITHM

A. Encoder

The image is divided into N non-overlapping blocks of size $\sqrt{b} \times \sqrt{b}$, and 2D-DCT is applied to each block before being stacked on top of each other to form a 3D matrix of size $\sqrt{b} \times \sqrt{b} \times N$. Vectors $\mathbf{x}_j \in \mathbb{R}^N$, $j = 1, ..., b$ along the third dimension are formed corresponding to each pair of row and column indices, as illustrated in Figure 1, before being "sparsified" by setting values that are smaller than the sparsity threshold τ_K to zero. This means that the same frequency components of all the DCT blocks are stacked into the same vector. Most natural images have DCT energy focused at low frequency components, therefore high frequency components will likely be set to zero and have minimal effect on the image quality.

The *sparsity level* of vector $\mathbf{x}_j$ refers to the number of its non-zero entries, and is denoted by k_j, i.e., $k_j \triangleq ||\mathbf{x}_j||_0$. The empirical mean of each $\mathbf{x}_j$ vector is first subtracted from all its entries to obtain a zero-mean vector, such that the empirical variance of each vector is equivalent to its power, before being multiplied by a pseudo-random orthonormal measurement matrix $\mathbf{\Phi}_j \in \mathbb{R}^{\mu_j \times N}$, where $\tilde{\mathbf{y}}_j \triangleq \mathbf{\Phi}_j(\mathbf{x}_j - c_j)$ and c_j is the empirical mean of vector $\mathbf{x}_j$. We wish to maintain $k_j \leq \mu_j \ll N$, where μ_j is the number of measurements sent for vector $\mathbf{x}_j$, which provides a trade-off between the accuracy and the channel bandwidth. Note that, to distinguish the received signals for each of the vectors, the decoder needs to know μ_j's, which increases the meta-data size. To reduce the meta-data, we choose μ_j's from a set of S predefined measurement levels. Thus, for each vector, μ_j is chosen as the closest value among the set of predefined measurement levels that is higher than the desired number of measurements. The total amount of meta-data for μ_j's to be transmitted is then given by $b \cdot \log_2 S$ bits. Moreover, in the case where $\mu_j = N$, we set $\mathbf{\Phi}_j = \mathbf{I}$ and $\mathbf{x}_j$ is not sparsified as to not lose performance when there is insufficient sparsity to exploit.

Finally, we allocate the transmit power among different frequency components according to their empirical variances. We would like to transmit the elements of these vectors over the channel by simple scaling. The optimal scaling coefficients
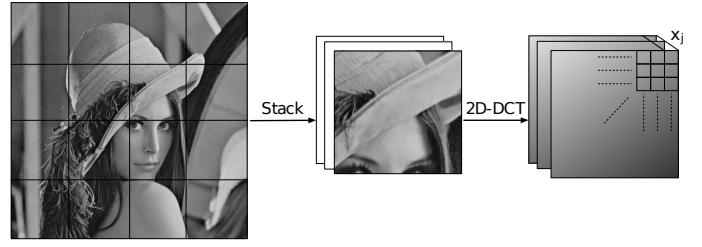

Fig. 1. Encoder divides the frame into blocks of $\sqrt{b} \times \sqrt{b}$ before computing the 2D-DCT coefficients, and forms vectors $\mathbf{x}_j$ by extracting the DCT coefficients corresponding to the same pair of indices across all blocks.
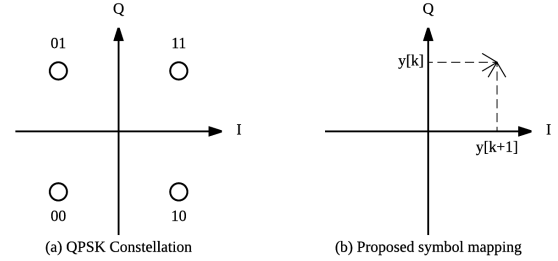

Fig. 2. Unlike traditional modulation schemes, such as QPSK, SparseCast does not have a fixed set of constellation points.

to transmit independent Gaussian samples, derived in [12], is presented in Lemma 1 for completeness.

Lemma 1: Given L data vectors $\mathbf{x}_1, ..., \mathbf{x}_L$, each consisting of $m_j = |\mathbf{x}_j|$ samples from a zero mean Gaussian distribution with variance λ_j, for $j = 1, ..., L$, the linear encoder that minimises the MMSE in the presence of additive white Gaussian noise (AWGN) scales the j-th vector by g_j, where

$$g_j = \lambda_j^{-1/4} \left(\sqrt{\frac{\sum_j m_j}{\sum_j m_j \sqrt{\lambda_j}}} \right), \quad j = 1, \ldots, L. \quad (1)$$

In our implementation, we have $m_j = \mu_j$, the number of measurements transmitted for $\mathbf{x}_j$, $j = 1, \ldots, b$.

Channel symbols are formed by pairing consecutive elements of vector $\mathbf{y}_j = g_j \tilde{\mathbf{y}}_j$ as the in-phase and quadrature (I/Q) components of a complex symbol (Figure 2). This differs from traditional systems that map a bit sequence to a predefined set of discrete constellation points, and allows the channel noise to directly corrupt the coefficients; and hence, preserves the linearity of the scheme (apart from the thresholding, which serves for compression). The generated symbols along with pilot symbols for channel estimation are sent over the channel.

Meta-data consisting of the empirical mean and variance values as well as the number of measurements μ_j for each $\mathbf{x}_j$ are sent separately to the receiver. For the proposed technique to work, the meta-data must be received without error; therefore, we employ BPSK modulation with 1/2 rate convolutional code for a strong protection against channel errors.

B. Decoder

The received vector is $\hat{\mathbf{y}}_j = \mathbf{y}_j + \mathbf{n}_j$, where $\mathbf{n}_j$ is the AWGN term with power $\frac{1}{2}\sigma_\mathbf{n}^2$, $\forall j$; the noise power is assumed to be

the same for all transmissions. The decoder employs AMP to decode the sparse $\mathbf{x}_j$ vector if $\mu_j < N$. When $\mu_j = N$, MMSE estimator is used, as the AMP algorithm performs poorly for non-sparse vectors.

The MMSE estimate of a non-sparse $\mathbf{x}_j$ vector is given by $\hat{\mathbf{x}}_j = \frac{g_j \lambda_j}{g_j^2 \lambda_j + \sigma_{\mathbf{n}}^2/2} \hat{\mathbf{y}}_j + c_j$, where c_j and λ_j (i.e. empirical mean and variance) of the vector $\mathbf{x}_j$ is obtained from the meta-data. The 2D-DCT coefficients of the image are then reconstructed from the $\hat{\mathbf{x}}_j$ vectors before inverse 2D-DCT is performed to obtain the original pixels.

Although AMP increases the complexity of decoding compared to the linear decoding in SoftCast, its per iteration complexity is dominated by a matrix-vector multiplication operation with a worst-case complexity of $O(\mu_j N)$ [11], and the required number of iterations is typically on the order of tens. As we will see in the next section, this slightly increased decoding complexity can be justified with the increased performance.

III. Results

We compare SparseCast with sparse recovery using BCS-SPL as described in [8], SoftCast and standard digital encoding by comparing the achieved PSNR across a range of CSNR values. A smooth and linear relationship between PSNR and CSNR is desirable particularly when transmitting to multiple receivers or over a time-varying unknown channel, while a higher PSNR for given CSNR and channel bandwidth constraints is indicative of better bandwidth utilisation efficiency. The algorithms are first simulated in MATLAB and then implemented using USRP NI2900 and LabView Communications Design Suite 2.0 for real world testing.

Figure 3 shows a linear relationship between CSNR and PSNR for the considered uncoded transmission schemes, indicating they can achieve strong multicasting performance, and are less sensitive to inaccuracies or the lack of channel state information at the transmitter. SparseCast is approximately 3.5 dB and 15.2 dB better in PSNR than SoftCast and BCS-SPL, respectively, showing the superiority of SparseCast in terms of its ability to adapt to varying CSNR, as well as bandwidth efficiency over a wide range of channel conditions.

Advantages of uncoded transmission schemes are clear when compared with digital transmission with JPEG compression followed by conventional constellations and codes from the 802.11a standard. The points on Figure 3 correspond to the achieved PSNR values at the corresponding CSNR threshold. The CSNR threshold for each constellation and convolutional code rate pair with 10% packet loss rate are given in Table I. Different constellations and code rates have different compression requirements. For example, BPSK (1 bit/symbol) with 1/2 rate (input bits/output bits) convolutional code and 131,000 available channel symbols implies the source image size must be compressed under 1 bit/symbol $\times$ 131000 symbols $\times$ 0.5 rate $=$ 65,500 bits. In contrast, JSCC schemes do not suffer from the cliff effect and can smoothly adjust the output PSNR with respect to CSNR. SparseCast and SoftCast both follow and even sometimes surpass the envelope formed by the different digital modulation schemes, especially at low

TABLE I. CSNR THRESHOLDS FOR UNCODED/CODED CONSTELLATIONS IN 802.11A

Constellation	Code rate	CSNR threshold (dB)			
		Uncoded	Coded 1/2	Coded 2/3	Coded 3/4
BPSK	1/2 or 3/4	8	3	-	5
QPSK	1/2 or 3/4	11	6	-	8
16-QAM	1/2 or 3/4	18	11	-	15
64-QAM	2/3 or 3/4	24	-	19	21

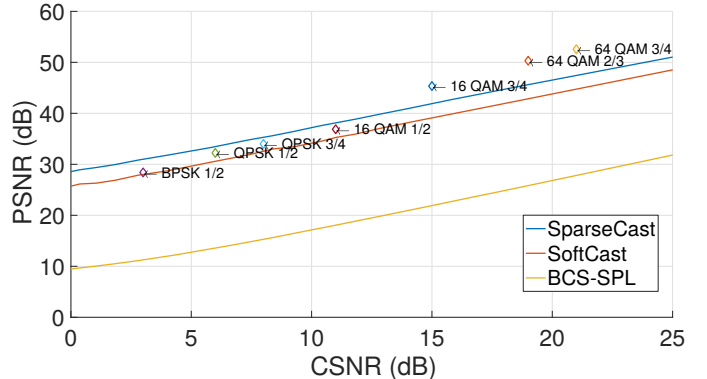CSNR values, suggesting their superiority for wireless image transmission while avoiding the cliff effect.

Fig. 3. PSNR vs. CSNR. The channel symbol length is $131,000$ symbols. The single points correspond to the PSNR values at the CSNR threshold of each modulation scheme from the 802.11a standard.

The meta-data size also differs across algorithms, with SparseCast requiring the greatest amount. In our simulations, SparseCast requires meta-data size of around $17,000$ bits with block size 16×16. SoftCast requires $10,000 - 16,000$ bits depending on the block threshold, while BCS-SPL requires no meta-data, both with block size 32×32. Different block sizes are used to ensure the meta-data size is similar. The increase compared to SoftCast is due to the need to transmit μ_j's and the way vectors $\mathbf{x}_j$ are generated. This is subject to change for different block sizes and parameters and should only have a small impact on the overhead of the algorithm. We emphasise that the meta-data size is parameter dependent and the results show that, under similar meta-data sizes, SparseCast is able to outperform SoftCast in terms of PSNR. It should be noted here that real numbers (i.e. mean and variance values) use the single precision floating point format (32 bits).

Finally, the results obtained from the practical implementation confirm the observations made from simulations. As can be seen in Figure 4, the same linear relationship between PSNR and CSNR can be observed. However, an important difference between the USRP and simulation results is the slope of the plots. Whereas the simulation results exhibit a slope of 1, due to the AWGN assumption, the USRP results show slopes at about $1/2$. This is likely due to additional channel distortions caused by disturbances in the environment and transmitter/receiver oscillator misalignment. The latter is due to the free running oscillators in the USRP hardware, which can result in phase offset between the carrier frequencies.

The simulated results also tend to plateau at very low CSNR values as seen in Figure 4. This effect is caused by the output
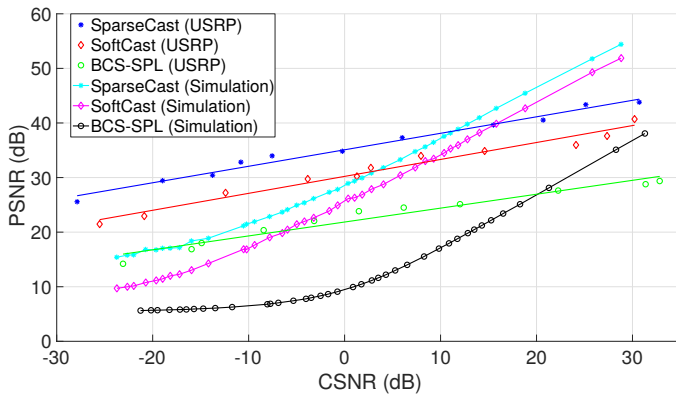
(a) High bandwidth usage ($131,000$ channel symbols).
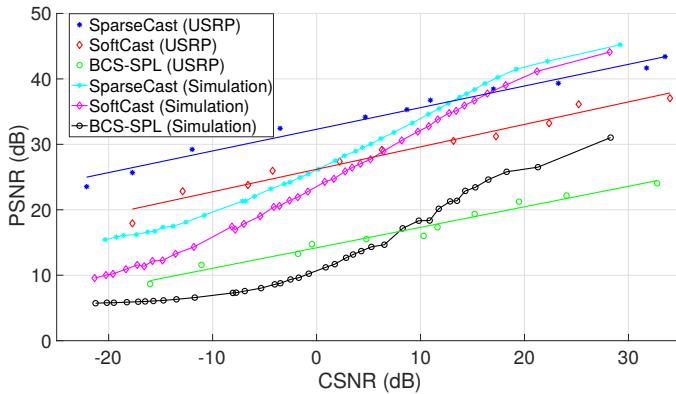
(b) Low bandwidth usage ($75,000$ channel symbols).

Fig. 4. USRP and simulation results. Block size for SparseCast is 16×16 while it is 32×32 for SoftCast and BCS-SPL. The lines drawn for USRP results are the linear regression lines for each respective algorithm. In (a), SparseCast was simulated with $\mu_j = 3k_j$ and $\tau_K = 0.1$, BCS-SPL with $\lambda = 0.001$ and sampling ratio of 0.65, SoftCast with block threshold of 7400. In (b), SparseCast was simulated with $\mu_j = 3k_j$ and $\tau_K = 3.5$, BCS-SPL with $\lambda = 0.001$ and sampling ratio of 0.4, SoftCast with a block threshold of $28,000$.

pixel luminosity being limited to the range of $[0, 255]$, limiting the maximum possible error. This cannot be seen in the USRP results as the PSNR never reached the level suggested by the simulation for the effects to be observed. To replicate this phenomenon with the USRP hardware would require a larger CSNR range which was not possible with the hardware model used due to antenna saturation and gain limitations.

IV. CONCLUSIONS

We proposed SparseCast, a novel hybrid digital-analog image transmission technique based on uncoded transmission of DCT coefficients. Sparsity in the frequency domain is exploited to improve the bandwidth usage, and a fixed set of measurement levels are used to reduce the amount of transmitted meta-data. We have used AMP to recover sparse vectors at the receiver to reduce decoder complexity. Shown by both simulation and experimental results, the combined use of MMSE and AMP for CS recovery overcomes the shortcomings of iterative CS recovery algorithms under non-sparse scenarios, while exploiting sparsity effectively. We have

(a) Digital, PSNR = $25.00dB$

(b) SparseCast, PSNR = $31.05dB$

(c) SoftCast, PSNR = $27.98dB$

(d) BCS-SPL, PSNR = $14.13dB$

Fig. 5. Simulation results for visual quality comparison of test image "Lenna" under CSNR = $5dB$ and channel symbol length $75,000$. For digital transmission JPEG compression is used with BPSK modulation and 1/2 rate convolutional coding.

focused on the transmission of a single image here, but the proposed scheme can easily be used for video transmission, similarly to [4], [6] and [7]. Non-linear mappings can also be used for bandwidth compression or expansion to better exploit the channel bandwidth as in [13], [14] with an increased encoder and decoder complexity.

REFERENCES

[1] T. Goblick, "Theoretical limitations on the transmission of data from analog sources," *IEEE Trans. on Inf. Theory*, vol. 11, no. 4, Oct. 1965.

[2] I. E. Aguerri and D. Gunduz, "Joint source-channel coding with time-varying channel and side-information," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 736–753, Feb. 2016.

[3] A. Lapidoth and S. Tinguely, "Sending a bivariate Gaussian over a Gaussian mac," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, Jun. 2010.

[4] S. Jakubczak and D. Katabi, "Softcast: One-size-fits-all wireless video," in *Proc. ACM SIGCOMM*, New York, NY, Aug. 2010, pp. 449–450.

[5] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[6] C. Li, H. Jiang, P. Wilford, and Y. Zhang, "Video coding using compressive sensing for wireless communications," in *Proc. IEEE Wireless Comm. and Networking Conf.*, Mar. 2011, pp. 2077–2082.

[7] M. B. Schenkel, F. Wu, C. Luo, and P. Frossard, "Compressed sensing based video multicast," *Proc. VCIP*, Jul. 2010.

[8] W. Yin, X. Fan, Y. Shi, R. Xiong, and D. Zhao, "Compressive sensing based soft video broadcast using spatial and temporal sparsity," *Mobile Networks and Applications*, vol. 21, no. 6, pp. 1002–1012, Dec. 2016.

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

[10] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *IEEE Int'l Conf. on Image Proc.*, Nov. 2009.

[11] D. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, Nov. 2013.

[12] K.-H. Lee and D. Petersen, "Optimal linear coding for vector channels," *IEEE Trans. on Comm.*, vol. 24, no. 12, pp. 1283–1290, Dec. 1976.

[13] A. N. Kim and F. Hekland, "Dimension reduction and expansion: Distributed source coding in a noisy environment," in *Data Compression Conference*, March 2008, pp. 332–341.

[14] A. A. Saleh, W. Chan, and F. Alajaji, "Compressed sensing with non-linear analog mapping in a noisy environment," *IEEE Signal Processing Letters*, vol. 19, no. 1, pp. 39–42, Jan 2012.