Joint Source-Channel Coding for Deep Space Image Transmission using Rateless codes

Ozgun Y. Bursalioglu, Giuseppe Caire
Ming Hsieh Department of Electrical Engineering
University of Southern California
Email: {bursalio, caire}@usc.edu

Dariush Divsalar
Jet Propulsion Laboratory
California Institute of Technology
Email: Dariush.Divsalar@jpl.nasa.gov

***Abstract*—A new coding scheme for image transmission over noisy channel is proposed. Similar to standard image compression, the scheme includes a linear transform followed by embedded scalar quantization. Joint source-channel coding is implemented by optimizing the rate allocation across the source subbands, treated as the components of a parallel source model. The quantized transform coefficients are linearly mapped into channel symbols, using systematic linear encoders of appropriate rate. This fixed-to-fixed length "linear index coding" approach avoids the use of an explicit entropy coding stage (e.g., arithmetic or Huffman coding), which is typically non-robust to post-decoding residual errors. Linear codes over $GF(4)$ codes are particularly suited for this application, since they are matched to the alphabet of the quantization indices of the dead-zone embedded quantizers used in the scheme, and to the QPSK modulation used on the deep-space communication channel. Therefore, we optimize a family of systematic Raptor codes over $GF(4)$ that are particularly suited for this application since they allow for a continuum of coding rates, in order to adapt to the quantized source entropy rate (which may differ from image to image) and to channel capacity. Comparisons are provided with respect to the concatenation of state-of-the-art image coding and channel coding schemes used by Jet Propulsion Laboratories (JPL) for the Mars Exploration Rover (MER) Mission.**

I. Introduction

In conventional image transmission over noisy channels, the source compression and channel coding stages are designed and applied separately. Image coding is usually implemented by a linear transformation (transform coding), followed by the quantization of the transform coefficients, and by entropy coding of the resulting redundant discrete source formed by the sequence of quantization indices. Due to the catastrophic behavior of standard entropy encoding/decoding schemes, even if only a few bits are in error after the channel decoder, the decompressed source transform coefficients are dramatically corrupted, resulting in a substantially useless reconstructed image. In order to prevent this catastrophic error propagation, the source is partitioned into blocks, such that the errors are confined in the blocks, at the cost of some extra redundancy. In order to preserve integrity, which is a strict requirement in deep space exploration scientific missions, blocks affected by errors are requested for retransmission, at the cost of significant extra delay and power expenditure. Furthermore, due to the typically sharp waterfall BER behavior of powerful modern codes, when channel conditions change due to different atmospheric conditions, or antenna misalignment, the resulting post-decoding BER may rapidly degrade, producing a sequence of highly corrupted images that need retransmission.

In this paper we consider a scheme for Joint Source-Channel Coding (JSCC) that avoids conventional entropy coding. This JSCC scheme [1], [2] consists of a wavelet transform, a scalar embedded quantizer, and linear encoding of the quantization indices. These three components are examined in Sections III, IV and V. Before getting into the details of transform, quantizer and linear code design, Section II introduces the notation used throughout the paper and define the relevant system optimization problem for JSCC based on the concatenation of embedded quantization and channel coding in general, for practical quantization defined in terms of an operational rate distortion function, and practical coding defined in terms of a certain overhead from capacity.

II. System Setup

For the problem at hand, the deep-space transmission channel is represented by the discrete-time complex baseband equivalent model

$$y_i = \sqrt{E_s} x_i + z_i, \tag{1}$$

where $y_i \in \mathbb{C}$, x_i is a QPSK symbol and $z_i \in \mathcal{CN}(0, N_0)$ is the complex circularly symmetric AWGN sample.

Consider a "parallel" source formed by s independent components. A block of source symbols of length K is denoted by $\mathbf{S} \in \mathbb{R}^{s \times K}$, where the i-th row of $\mathbf{S}$, denoted by $\mathbf{S}^{(i)} = (S_1^{(i)}, \ldots, S_K^{(i)})$, corresponds to a source component of the parallel source.

A $(s \times K)$-to-N source-channel code for transmitting the source block $\mathbf{S}$ onto the channel (1) is formed by an encoding function that maps $\mathbf{S}$ into a codeword of N QPSK symbols, and by an encoding function that maps the sequence of channel outputs $(y_1, \ldots, y_N)$ into the reconstructed source block $\widehat{\mathbf{S}}$.

We consider a Weighted MSE (WMSE) distortion measure defined as follows. Let the MSE for the i^{th} source component be given by $d_i = \frac{1}{K}\mathbb{E}[\|\mathbf{S}^{(i)} - \widehat{\mathbf{S}}^{(i)}\|^2]$. Then, the WMSE

This research in part was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. The work by the University of Southern California, and JPL was funded through the NASA/JPL/DRDF/SURP Program.

distortion at the decoder is given by

$$D = \frac{1}{s} \sum_{i=1}^{s} v_i d_i, \qquad (2)$$

where $\{v_i \geq 0 : i = 1, \ldots, s\}$, are weights that depend on the specific application. In our case, these coefficients correspond to the weights of a bi-orthogonal wavelet transform as explained in Section III.

Let $r_i(\cdot)$ denote the R-D function of the i^{th} source component with respect to the MSE distortion. Then the R-D function of $\mathbf{S}$ with respect to the WMSE distortion is given by

$$\mathcal{R}(D) = \min \ \frac{1}{s} \sum_{i=1}^{s} r_i(d_i), \ \text{ subject to } \frac{1}{s} \sum_{i=1}^{s} v_i d_i = D, \qquad (3)$$

where the optimization is with respect to $\{d_i \geq 0 : i = 1, \ldots, s\}$, corresponding to the individual MSE distortions of the i^{th} source components. For example, for parallel Gaussian sources and equal weights ($v_i = 1$ for all i), (3) yields the well-known "reverse waterfilling" formula (see [3, Theorem 10.3.3]).

For a family of successively refinement codes with R-D functions $\{r_i(d) : i = 1, \ldots, s\}$, assumed to be convex and non-increasing [4], and identically zero for $d > \sigma_i^2 \triangleq \frac{1}{K} \mathbb{E}[\|\mathbf{S}^{(i)}\|^2]$, the operational R-D function of the parallel source $\mathbf{S}$ is also given by (3). Therefore, in the following, $\mathcal{R}(D)$ is used to denote the actual operational R-D function of for some specific, possibly suboptimal, successive refinement code.

For a source with a total number of samples equal to Ks, encoded and transmitted over the channel using N channel uses, we define $b = N/(Ks)$ as the number of channel-coded *symbol per pixel* (spp). Hence, b is a measure of the system *bandwidth efficiency*. Obviously, the minimum distortion D that can be achieved at channel capacity C and bandwidth expansion b is given by $D = \mathcal{R}^{-1}(bC)$.

III. Wavelet Transform

The image is decomposed into a set of "parallel" source components by a Discrete Wavelet Transform (DWT). Here, we use the same DWT of JPEG2000 [5]. With W levels of DWT, the transformed image is partitioned into $3W + 1$ "subbands". A subband decomposition example is given in Fig. 1 for $W = 3$. This produces $3W + 1 = 10$ subbands, which in the figure are indicated by LL0, HL1, LH1, HH1, HL2, LH2, HH2, HL3, LH3, HH3, respectively. The subbands have different lengths, all multiples of the LL0 subband length. For simplicity, here we partition the DWT into source components of the same length, all equal to the the length of the LL0 subband. This yields $s = 2^{2W}$ source component blocks of length $K = \mathcal{K}^2/s$, where $\mathcal{K} \times \mathcal{K}$ indicates the size of the original image. Since the DWT is a bi-orthogonal transform, the MSE distortion in pixel domain is not equal to the MSE distortion in wavelet domain. In our case, for $W = 3$, the weight of a source component block in subband $w = \{1, \ldots, 10\}$ is given by the w-th coefficient of the vector $[l^6, l^5h, l^5h, l^4h^2, l^3h^3, l^3h^3, l^2h^2, lh, lh, h^2]$, where, for the particular DWT considered in this work (namely, the CDF 9/7 [6] wavelet used by JPEG2000 for lossy compression), we have $l = 1.96$ and $h = 2.08$.

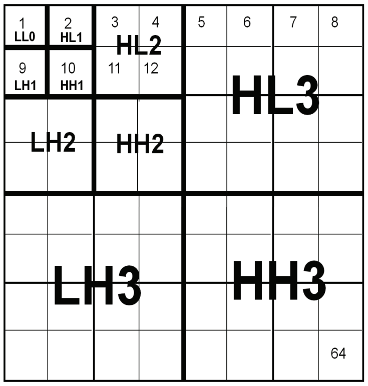

Figure 1. $W = 3$, partitioning of an image into 10 subbands and 64 source components.

The subband LL0 (that coincides with the first source component of the parallel source model) consists roughly of a decimated version of the original image. As explained in [7], in order to obtain better compression in the transform domain, Discrete Cosine Transform (DCT) is applied to subband LL0 so that its energy is "packed" into a very few, very high valued coefficients (note the nonzero density points shown by arrows in Fig. 2-b). These high values coefficients are separately transmitted as part of the header and not considered here. The remaining coefficients show sample statistics similar to the other subbands, (see Fig. 2-c).

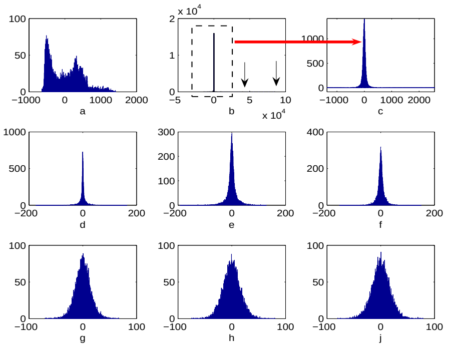

Figure 2. a: Histogram of 1^{st} source component, b: Histogram of 1^{st} source component after DCT transform, c: Zoomed in version of b, d-j: Histograms of source components $2 - 7$, respectively.

IV. Quantizer

The simplest form of quantization employed by JPEG2000 is a special uniform scalar quantizer where the center cell's width is twice the width of the other cells for any resolution level. This type of quantizers are called "dead-zone" quantizers. In Fig. 3, an embedded dead-zone quantizer is shown for 3 different resolution levels. The number of cells at any level p is given by $2^{p+1} - 1$. We indicate the cell partition at every level by symbols $\{0, 1, 2\}$ as shown in Fig.3. The scalar quantization function is denoted as $\mathcal{Q} : \mathbb{R} \to \{0, 1, 2\}^P$, where $2^{P+1} - 1$ is the number of quantization regions for the highest level of refinement.

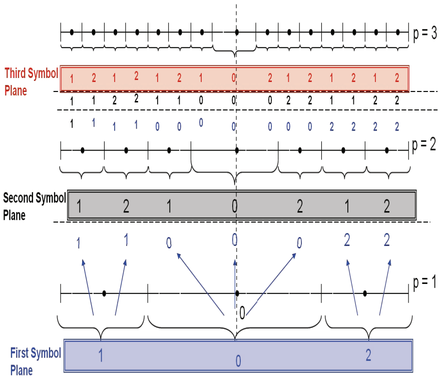

Figure 3. Quantization cell indexing for an embedded dead-zone quantizer with $p = 1, 2, 3$

Let $\mathbf{U}^{(i)} = \mathcal{Q}(\mathbf{S}^{(i)})$ denote the sequence of ternary quantization indices, formatted as a $P \times K$ array. The p-th row of $\mathbf{U}^{(i)}$, denoted by $\mathbf{U}^{(i)}_{(p,:)}$, is referred to as the p-th "symbol-plane". Without loss of generality, we let $\mathbf{U}^{(i)}_{(1,:)}, \ldots, \mathbf{U}^{(i)}_{(P,:)}$ denote the symbol-planes with decreasing order of significance. A refinement level p consists of all symbol planes from 1 to p. The quantization distortion for the i-th source component at refinement level p is denoted by $D_{\mathcal{Q},i}(p)$.

The quantizer output $\mathbf{U}^{(i)}$ can be considered as a *discrete memoryless source*, with entropy rate $H^{(i)} = \frac{1}{K} H(\mathbf{U}^{(i)})$ (in bits/source symbol). Using the chain rule of entropy [3], this can be decomposed as $H^{(i)} = \sum_{p=1}^{P} H^{(i)}_p$, with

$$H^{(i)}_p = \frac{1}{K} H \left(\mathbf{U}^{(i)}_{(p,:)} \middle| \mathbf{U}^{(i)}_{(1,:)}, \ldots, \mathbf{U}^{(i)}_{(p-1,:)} \right), \quad p = 1, \ldots, P. \tag{4}$$

Then, the set of R-D points achievable by the concatenation of embedded scalar quantizer using $0, 1, \ldots, P$ quantization levels [1] and an entropy encoder is given by

$$\left(\sum_{j=1}^{p} H^{(i)}_j, D_{\mathcal{Q},i}(p) \right), \quad p = 0, \ldots, P, \tag{5}$$

where, by definition, $D_{\mathcal{Q},i}(0) = \sigma_i^2$. Using time-sharing, any point in the convex hull of the above achievable points is also achievable. Finally, the operational R-D curve $r_i(d)$ of the scalar quantizer is given by the *lower convex envelope* of the convex hull of the points in (5). It is easy to see that $r_i(d)$ is a piecewise linear function. Therefore, the resulting function $r_i(d)$ is convex and decreasing on the domain $D_{\mathcal{Q},i}(P) \leq d \leq \sigma_i^2$. Fig. 4 shows, qualitatively, the typical shape of the functions $r_i(d)$.

As seen from Fig. 4, it is possible to represent $r_i(d)$ as the pointwise maximum of lines joining consecutive points in the set given in (5). Hence, we can write

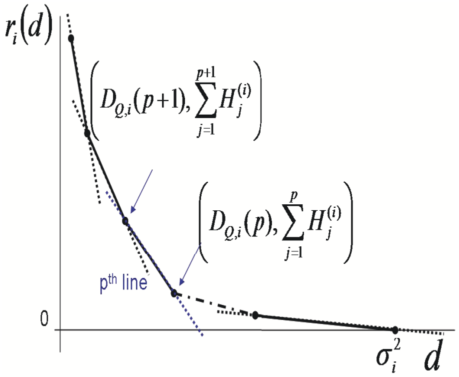

Figure 4. Piecewise linear operational R-D function for the i-th source corresponding to a set of discrete R-D points.

$$r_i(d) = \max_{p=1,\ldots,P} \{a_{i,p} d + b_{i,p}\}, \tag{6}$$

where the coefficients $a_{i,p}$ and $b_{i,p}$ are obtained from (5) (details are trivial, and omitted for the sake of brevity). Using (6) into (3), we obtain the operational R-D function of the parallel source as the solution of a *linear program*. Introducing the auxiliary variables γ_i, the minimum WMSE distortion with capacity C and bandwidth expansion b is given by

Min Weighted Total Distortion (MWTD):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{s} \sum_{i=1}^{s} v_i d_i \\ \text{subject to} \quad & \frac{1}{s} \sum_{i=1}^{s} \gamma_i \leq bC, \\ & D_{\mathcal{Q},i}(P) \leq d_i \leq \sigma_i^2, \;\; \forall\, i, \\ & \gamma_i \geq a_{i,p} d_i + b_{i,p}, \;\; \forall\, i, p. \end{aligned} \tag{7}$$

(linear program in $\{d_i\}$ and $\{\gamma_i\}$, that can be solved by standard optimization tools).

Note that MWTD problem (7) is derived assuming separate source and channel coding scheme and a capacity achieving channel code. On the other hand, for the proposed JSCC scheme, each refinement level (symbol plane) of each source component is encoded separately with some practical code. For example, consider the p^{th} plane of the i^{th} source component whose conditional entropy is given by $H^{(i)}_p$, let $n^{(i)}_p$ denote the number of encoded channel symbols for this plane. The sum $\sum_{i=1}^{s} \sum_{p=1}^{P} n^{(i)}_p$ yields the overall coding block length N (channel symbols). Consistent with the definition of the Raptor code *overhead* for channel coding applications [8], we define the overhead $\theta^{(i)}_p$ for JSCC as the factor relating $n^{(i)}_p$ to its ideal value $K H^{(i)}_p / C$ as follows:

$$n^{(i)}_p = \frac{K H^{(i)}_p (1 + \theta^{(i)}_p)}{C}. \tag{8}$$

As shown in [1], the MWTD problem for JSCC with overheads $\theta^{(i)}_p$ and entropies $H^{(i)}_p$ takes on the same form of (7), where

[1] Notice: 0 quantization levels indicates that the whole source component is reconstructed at its mean value.

the coefficients $\{a_{i,p}, b_{i,p})$ correspond to the modified R-D points

$$\left(\sum_{j=1}^{p} H_j^{(i)}(1+\theta_j^{(i)}), D_{\mathcal{Q},i}(p) \right), \quad p = 0, \ldots, P, \qquad (9)$$

instead of (5). For given code families and block lengths, the overhead factors $\theta_p^{(i)}$ are experimentally determined, and used in the system design according to the modified optimization problem (7). Fortunately, the overhead factors are only sensitive to the entropy of the plane $H_p^{(i)}$, and to the coding block length (which is an a priori decided system parameter). Hence instead of finding $\theta_p^{(i)}$ for each i and p, one can simply find the overheads for different entropy values on a sufficiently fine grid.

To get an idea about the range of entropy values resulting from deep-space image quantization, here we report the conditional entropies of the first source component of an image from the Mars Exploration Rover.[2] We have:

$$\begin{aligned} \{H_1^{(1)}, \ldots, H_8^{(1)}\} \;&=\; \Big\{0.0562,\, 0.0825,\, 0.2147,\, 0.4453, \\ &\quad 0.8639,\, 1.1872,\, 1.1917,\, 1.1118\Big\} \end{aligned} \qquad (10)$$

These values span the range of entropy values for all MER images we used in this work, and can be considered as "typical" for this application.

V. Code Design

In this section we discuss the code design for a single discrete source of entropy H, corresponding to a symbol plane, and the QPSK/AWGN channel with capacity C. For notation simplicity, source and plane indices are omitted.

As discussed in Sec. I, in the proposed JSCC scheme symbol planes are directly mapped to channel symbols. Since symbol planes are nonbinary, i.e. $\{0, 1, 2\}$, and we consider QPSK modulation, we consider linear codes over $GF(4)$. This is particularly well suited to this problem for the following reason. Assume that a block length of K symbols is encoded by a systematic encoder of rate $K/(K + n)$. Only the n parity symbols are effectively sent through the channel. At the receiver, the decoder uses the a-priori non-uniform source probability for the systematic symbols and the posterior symbol-by-symbol probability (given the channel outputs $y_1, \ldots, y_n$) for the parity symbols. It turns out that if the physical channel is "matched", such that the transition probability is symmetric (in the sense defined by [9]) with respect to the sum operation in $GF(4)$, then the source-channel decoding problem is completely equivalent to decoding the all-zero codeword of the same systematic code, transmitted over a "virtual" two-block symmetric channel (see Fig.5), where the first K symbols go through an additive noise channel over $GF(4)$ whose noise sequence realization is *exactly* equal to the source block, and the remaining n parity symbols go through the physical channel. This equivalence holds also for the Belief Propagation iterative decoder (see [2] for details). Notice that the additive noise channel over $GF(4)$ has capacity $2 - H$ bit/symbol, where H is the source entropy. It turns out that if we represent the symbols of $GF(4)$ using the additive vector-space representation of the field over $GF(2)$, i.e., as the binary pairs $(0, 0), (0, 1), (1, 0), (1, 1)$ and map the "bits" over the QPSK symbols using Gray mapping, then the symmetry of the physical channel under $GF(4)$ sum holds. Therefore, systematic linear coding over $GF(4)$ is particularly suited to "linear index coding" of the symbol planes produced by the embedded dead-zone quantizer and for the QPSK/AWGN deep space communication channel.

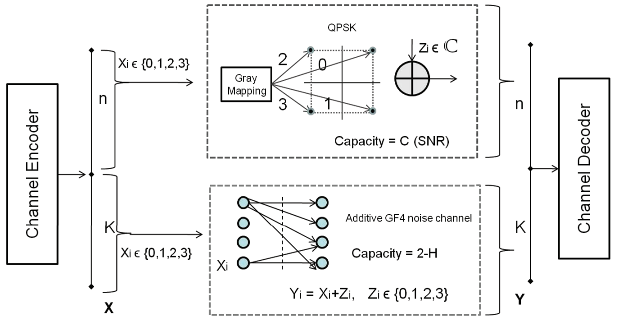

Figure 5. Block memoryless channel with two symmetric channel components

Given this equivalence, it turns out that the optimization of a linear systematic code ensemble for the source-channel coding problem is obtained through the more familiar optimization for a channel coding problem, where the channel is composite, and consists of two blocks, one discrete symmetric quaternary DMC with capacity $2 - H$, and one quaternary-input continuous output symmetric channel induced by (1) and by Gray mapping, with capacity $0 \leq C \leq 2$ (that depends on the SNR $\triangleq 10 \log_{10} E_s/N_0$).

Different symbol planes may have different entropies and therefore they may result in codes with different overhead values for various SNR conditions. Nonuniversality of Raptor codes is shown in [8] using the fact the *Stability Condition* on the fraction of degree-2 output nodes depends on the channel parameter for Binary Symmetric and Binary Input AWGN channels. Following the derivation in [8], we established a stability condition on degree-2 output nodes which is a function of both H and C. Hence the code ensembles must be optimized for each pair of (H, C) values. In this paper, we consider the optimization of Raptor codes over $GF(4)$ for various H and C pairs chosen with respect to a fine grid spanning the typical range of the source symbol plane entropies (see (10)) and the typical range of deep space channel capacities [10].

In order to optimize the Raptor code ensembles, we consider an EXIT chart and linear programming method inspired by [8] and [9], extended to handle the two-block memoryless channel as in Fig. 5.

[2] Specifically image, "1F178787358EFF5927P1219L0M1".

A. EXIT CHART ANALYSIS

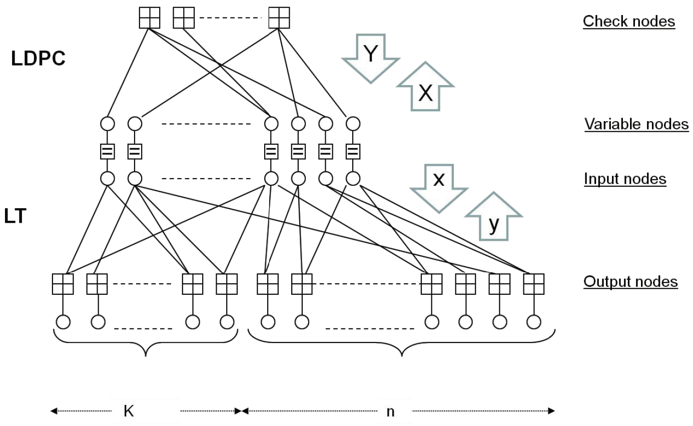

Figure 6. Raptor Code with LDPC outer code

EXIT chart analysis for the JSCC scheme using Raptor codes can be found in [2] for the binary case. The derivations of [2] can be easily modified to the case of nonbinary codes, under the symmetry "matched" condition said above, and using a Gaussian approximation as in [9], the conditional distribution of each message $\mathbf{L}$ is $\sim \mathcal{N}(\mu\mathbf{1}, \Sigma_\mu)$ where $[\Sigma_\mu]_{i,j} = 2\mu$ for $i = j$ and $[\Sigma_\mu]_{i,j} = \mu$ for $i \neq j$. Letting V the code variable corresponding to the edge message $\mathbf{L}$, we define the mutual information function

$$J(\mu) \triangleq I(V;\mathbf{L}) = 1 - \mathbb{E}\left[\log_4\left(1 + \sum_{i=1}^{3} e^{-L_i}\right)\right].$$

We use base-4 logarithm for mutual information calculations, hence in these sections H and C are in units of two bits per source symbol or per channel symbol, respectively. We first introduce the necessary notation and then write the EXIT equations for our problem.

A Raptor code is formed by the concatenation of a pre-code, here implemented by a high rate regular LDPC code, and an "LT" code, which is a low-density generator matrix code with a special generator matrix degree distribution [8]. For the Tanner graph of the LT code, we define the input nodes and the output nodes. For the Tanner graph of the LDPC code, we define the variable nodes and the check nodes (see Fig. 6).

EXIT charts can be seen as a multidimensional dynamic system. The EXIT "state variables" are $\mathsf{x}, \mathsf{y}, \mathsf{X}$ and Y defined as follows (See Fig. 6) :

x denotes the average mutual information between a randomly chosen input node symbol and a message sent on a left-bound adjacent edge (from input to output nodes).

y denotes the average mutual information between a randomly chosen input node symbol and a message sent on a right-bound adjacent edge (from output to input nodes).

X denotes the average mutual information between a randomly chosen variable node symbol and a message sent on a right-bound edge (from variable to check nodes).

Y denotes the average mutual information between a randomly chosen variable node symbol and a message sent on a left-bound edge (from check to variable nodes).

The various degree distributions for the Tanner graph in Fig. 6 are defined as follows:

- For the LDPC code, we let $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_j \rho_j x^{j-1}$ denote the generating functions of the edge-centric left and right degree distributions, and we let
$$\Lambda(x) = \sum_i \Lambda_i x^i = \frac{\int_0^x \lambda(u)du}{\int_0^1 \lambda(u)du},$$
denote the node-centric left degree distribution.
- For the LT code, we let $\iota(x) = \sum_i \iota_i x^{i-1}$ denote the edge-centric degree distribution of the input nodes, and we let $\omega(x) = \sum_j \omega_j x^{j-1}$ denote the edge-centric degree distribution of the "output nodes". The node-centric degree distribution of the output nodes is given by
$$\Omega(x) = \sum_i \Omega_j x^j = \frac{\int_0^x \omega(u)du}{\int_0^1 \omega(u)du}.$$
- For the concatenation of the LT code with the LDPC code we also have the node-centric degree distribution of the LT input nodes. This is given by
$$\gimel(x) = \sum_i \gimel_i x^i = \frac{\int_0^x \iota(u)du}{\int_0^1 \iota(u)du}.$$
Note that for large number of nodes we have the following approximation for $\gimel(x) \sim e^{\alpha(x-1)} = \sum_n \frac{\alpha^n e^{-\alpha}}{n!} x^n$ where $\alpha = \sum_i \gimel_i i$ is the average node degree for the input nodes [8]. Hence $\iota(x)$ is approximated by the following coefficients
$$\iota_i = \frac{\alpha^{i-1} e^{-\alpha}}{(i-1)!}. \tag{11}$$

Then, the LT EXIT equations are given by:

$$\mathsf{x} = \sum_k \sum_i \Lambda_k \iota_i J((i-1)J^{-1}(\mathsf{y}) + kJ^{-1}(\mathsf{Y})), \tag{12}$$

$$\begin{aligned}\mathsf{y} = {} & 1 - \sum_j \omega_j \Big\{\gamma J((j-1)J^{-1}(1-\mathsf{x}) + J^{-1}(H)) \\ & +(1-\gamma)J((j-1)J^{-1}(1-\mathsf{x}) + J^{-1}(1-C))\Big\},\end{aligned} \tag{13}$$

where $\gamma = K/(K+n)$. Eq. (13) follows from the fact that a random edge (o, v) is connected with probability γ to a source node (i.e., to the channel with capacity $1-H$ symbols per channel use), while with probability $1-\gamma$ to a parity node (i.e., to the channel with capacity C). It is also easy to see that $\gamma = r_{lt} r_{ldpc}$, where r_{lt} and r_{ldpc} denote rate of the LT code and rate of the LDPC part respectively.

The LDPC EXIT equations are given by:

$$\mathsf{X} = \sum_k \sum_i \lambda_k \gimel_i J((k-1)J^{-1}(\mathsf{Y}) + iJ^{-1}(\mathsf{y})), \tag{14}$$

$$\mathsf{Y} = 1 - \sum_\ell \rho_\ell J((\ell-1)J^{-1}(1-\mathsf{X})). \tag{15}$$

Eqs. (12), (13), (14), and (15) form the state equations of the global EXIT chart of the concatenated LT – LDPC graph with parameters H, C and γ, and the degree sequences ω, ι, ρ and λ.

Let μ_j denote the mean of the LLR of a source variable node connected to a checknode of degree j, given by

$$\mu_j = J^{-1}\left(1 - J\left(jJ^{-1}(1-\mathsf{x})\right)\right) + J^{-1}(1-H).$$

Then, the average symbol error rate (SER) of the source symbols can be approximated by

$$P_e = \sum_j \Omega_j \left[1 - Q^3\left(-\frac{\sqrt{\mu_j}}{2}\right)\right]. \tag{16}$$

B. LT Degree Optimization

For simplicity, we fix the LDPC code to be a regular $(2, 100)$ code ($r_{ldpc} = 0.98$). For this LDPC code, we find the mutual information threshold $\mathcal{Y}_0$ (using (14) and (15)) such that when $\mathsf{y} = \mathcal{Y}_0$ is the input, then the LDPC EXIT converges to $\mathsf{Y} = 1$. The value of $\mathcal{Y}_0$ depends on the LT input degree distribution $\iota(x)$, which in turns depends on α via (11). Fig. 7 shows $\mathcal{Y}_0$ as a function of α. On the other hand, as seen from (14) and (15), $\mathcal{Y}_0$ does not depend on H or C. Hence, for a fixed α, we can use the same $\mathcal{Y}_0(\alpha)$ threshold for all H, C pairs.

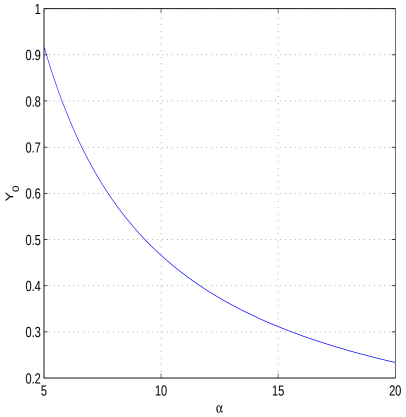

Figure 7. LDPC mutual information threshold vs α

Next, we use (12) and (13) to eliminate x and write y recursively. Note that the recursion for y depends on the input Y coming from the LDPC graph. For simplicity, we decouple the system of equations (14-15) and (12-13) by fixing the LDPC and the target mutual information $\mathcal{Y}_0(\alpha)$, and by disregarding the feedback from LDPC to LT in the BP decoder. Therefore, we let $\mathsf{Y} = 0$ in (12). The recursion function $f_j^{H,C,\gamma,\alpha}(\mathsf{y})$ for a degree-j output node is given in (17) on the top of the next page. At this point, the LT EXIT recursion converges to the target $\mathcal{Y}_0$ if

$$\mathsf{y} < 1 - \sum_j \omega_j f_j^{H,C,\gamma,\alpha}(\mathsf{y}), \;\; \forall \mathsf{y} \in [0, \mathcal{Y}_0(\alpha)] \tag{18}$$

We sample the interval $[0, \mathcal{Y}_0(\alpha)]$ on a fine grid of points, and obtain a set of linear constraints $\{\omega_i\}$ for fixed γ. The objective of this optimization consists of maximizing γ (the code rate) for a given H, C pair, where the optimization is with respect to $\{\omega_i\}$ and α. Since the LDPC code is fixed, γ is just a function of $r_{lt} = \frac{1}{\alpha \sum_j \omega_j / j}$. In order to linearize the constraints in $\{\omega_i\}$ we approximate γ in (18) with its ideal value, i.e., $C/(C+H)$, arguing that for a good code design, γ should be close to $C/(C+H)$.

The resulting optimization problem is given by:

$$\begin{aligned} \min_\alpha \quad \min_{\{\omega_j\}} \quad & \alpha \sum_j \frac{\omega_j}{j} \\ \text{s. t.} \quad & \sum_j \omega_j = 1, \; \omega_j \geq 0, \\ & \mathsf{y}_i < 1 - \sum_j \omega_j f_j^{H,C,\frac{C}{C+H},\alpha}(\mathsf{y}_i), \\ & \forall \mathsf{y}_i \in [0, \mathcal{Y}_0(\alpha)]. \end{aligned} \tag{19}$$

For fixed α, the inner optimization problem in (19) is a linear program in terms of $\{\omega_i\}$. The outer maximization of the coding rate r_{lt} can be obtained through an educated search over the value α, for each pair (H, C). With some more effort, it is also possible to write the stability condition for the degree-2 output nodes, that reads $\Omega_2 \geq g(H, C, \gamma)$, for some complicated function $g(\cdot)$ omitted here for brevity. Differently than in the classical memoryless stationary channel case, in our case γ appears in the condition in a non-linear manner. Therefore, instead of including the stability condition as a constraint in the optimization, we simply check that the result of the optimization satisfies the stability condition (otherwise, it is discarded).

VI. Results

In this section, we present in some details the deep space image transmission scheme currently employed by JPL - MER (Mars Exploration Rover) Mission. This scheme, that represents the benchmark for comparison to the JSCC scheme presented in this paper, is based on a separated approach including a state-of-the-art image compressor, called ICER [11], and state-of-the-art channel codes for deep space transmission. ICER is a progressive, wavelet-based image data compressor based on the same principles of JPEG2000, including a DWT, quantization, segmentation and entropy coding of the blocks of quantization indices with arithmetic coding and an adaptive probability model estimator based on context models. These blocks have differences with respect to their counterparts in JPEG2000 to handle properties of deep-space communication (details can be found in [11]).

ICER partitions an image into segments to increase robustness against channel errors. A segment "loosely" corresponds to a rectangular region of the image. Each image segment is compressed independently by ICER so that decoding error due to data loss affecting one segment has no impact on decoding of the other segments. The encoded bits corresponding to all the segments are concatenated and divided into fixed-length frames, that are individually channel encoded with a fixed channel code rate $R_c \in \mathcal{R}_c$, where $\mathcal{R}_c$ denotes a finite set

$$f_j^{H,C,\gamma,\alpha}(\mathrm{y}) \triangleq \left[\gamma J\left((j-1)J^{-1}(1-\sum_i \iota_i J((i-1)J^{-1}(\mathrm{y}))) + J^{-1}(H)\right) + (1-\gamma)J\left((j-1)J^{-1}(1-\sum_i \iota_i J((i-1)J^{-1}(\mathrm{y}))) + J^{-1}(1-C)\right)\right] \tag{17}$$

of possible coding rates. The channel coding rate is chosen according to the channel SNR.

R_c can be chosen from a set of different code rates $\mathcal{R}_c$ according to channel conditions.

A segment is generally divided into several frames. Data losses occur at the frame level. A whole segment is discarded even if a single frame corresponding to that segment is lost. When a frame loss occurs, it typically affects single segment. But it could affect two segments if the lost frame straddled the boundary between two segments. If a fixed PSNR[3] value is targeted, the discarded segments must be re-transmitted. Note that the delay and the cost of retransmission and feedback is significant when deep space image transmission is considered. In contrast, for the proposed JSCC we do not consider any retransmission.

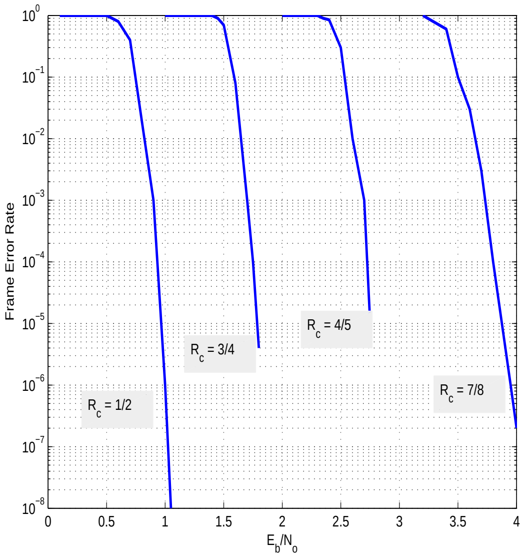

Figure 8. FER vs E_b/N_o curves for block length 16 K

In order to compare the performance of the proposed JSCC scheme with that of the baseline scheme, we first consider an experiment where a target PSNR is fixed. For both schemes, the bandwidth expansion factor b for fixed target PSNR depends on the particular image and on the channel SNR, E_s/N_0. For a given set of test images [4], we compare the two schemes in terms of b versus E_s/N_0, for the fixed target PSNR.

As explained above, when a frame is lost, the whole segment is retransmitted. Frame Error Rate (FER) is assumed to be fixed during the entire transmission process. Then the number of transmissions necessary for a segment is a geometric random variable with success probability depending only on the FER and the number of frames corresponding to the segment's data, denoted by F. Therefore, the expected number of transmissions for a segment is given by,

$$Z = (1 - FER)^{-F}.$$

Although this analysis is not exact, since the number of frames spanning each segment is not constant, in general, and some frame may straddle across two segments, nevertheless we can find tight upper and lower bounds to the average number of channel uses necessary to achieve the target PSNR, for given channel SNR and chosen coding rate R_c (details are given in [7]).

For a given SNR value, the baseline scheme chooses a code from standard JPL codes. For a well matched SNR and rate pair, the FER is very low that expected number of re-transmission is insignificant, then b is very close to the "one-shot" transmission value, i.e. $B/(2R_c)$ where B is the total number of ICER-encoded bits for the image at the given target PSNR.

For a given code rate, as the SNR increases beyond the matched point, b will stay fixed, since re-transmissions are getting more and more insignificant. On the other hand, when SNR is lower than the matched point, the FER increases significantly and re-transmissions become significant. In this case, b rapidly increases and becomes much larger than its minimum value $B/(2R_c)$. If SNR is very low with respect to a given channel code, then it might be more advantageous to switch to a lower rate code.

For the example considered in this paper, the target PSNR is 49 dB and the image used is 1024×1024 Mars image provided by JPL[5].

In Fig.9, we compare the b vs SNR performances of JSCC and of the baseline scheme. For JSCC, nonbinary code design is considered, but one can also use binary codes after dividing each nonbinary symbol plane into two bitplanes. This approach is taken in [7], where binary Raptor codes and protograph based binary LDPC codes are used (See [7] for details on these codes). Here we compare all of these cases for the sake of completeness.

- The $(*)$-curve corresponds to considering ideal capacity achieving codes for each plane in the JSCC scheme. For the range of PSNR values relevant to the MER mission, the pure compression rate (source coding only) for the scheme considered here and ICER are essentially identical (see [7] for further details on pure compression). Then, if ideal codes

[3] PSNR $= 10\log_{10}\frac{2^i-1}{D}$ where $i = 12$ since for MER mission each pixel is a 12-bit value in the original image.

[4] Provided by JPL-MER Mission Group.

[5] The name of the Mars image is 1F178787358EFF5927P1219L0M1.pgm

are assumed for both JSCC and the baseline scheme, the bandwidth efficiency of both schemes is the same. Hence the $(*)$-curve represents the best possible performance for both schemes, assuming ideal capacity achieving channel codes.

- Very tight upper and lower performance limits (actually overlapping as seen in Fig. 9) for the baseline scheme are shown by a combination of 4 knee-shaped curves, each of which corresponds to one of the codes whose FER performance is shown in Fig. (8) (see [10], [12] for details). The separated scheme requires several retransmissions of some blocks in the regimes of SNR for which the FER of selected code is significant. If for some reason (e.g., atmospheric propagation phenomena) the channel SNR worsens, eventually the separated scheme must decrease the coding rate and jump to the next available lower rate code. Hence the baseline scheme's performance is given as the lower envelope of these 4 curves. R_c value corresponding to each curve is also shown in Fig. 9.

- $(\times)$-curve is the result of EXIT calculations for protograph LDPC codes while $(+)$-curve is the finite length results of the same codes.

- $(-)$-curve is the result of EXIT calculations for binary Raptor codes when the LDPC code described in Sec.V-B is used with the LT degree distribution in (20). Shokrollahi reported this degree distribution in [13] to be used for erasure channels. $(--)$-curve corresponds to finite length simulations of the same degree distribution.

$$\Omega(x) = 0.008x + 0.494x^2 + 0.166x^3 + 0073x^4 + 0.083x^5 + 0.056x^8 + 0.037x^9 + 0.056x^{19} + 0.025x^{65} + 0.003x^{66}. \quad (20)$$

EXIT chart analysis result for nonbinary Raptor Codes when the same code (i.e. with no degree optimization) is used is given by $(-.)$-curve.

- As discussed in Sec.V, we described a method for degree optimization when the α parameter and LDPC code is fixed. Choosing the α parameter optimally for each H, C pair is a non-trivial task. Hence we only run the degree optimization linear program for each H, C pair using the α value given by the EXIT chart analysis of classic LT sequence (i.e. $(-.)$-curve). For the $(-.)$-curve, we fixed the $\omega(x)$ distribution by (20). Then, γ depends only on the α parameter. As the required rate, i.e. γ necessarily varies with respect to H, C, α parameter of the code changes as well. $(\circ)$-curve is obtained after optimizing the nonbinary codes for each H, C pair using the α value given by the EXIT chart analysis of nonbinary Raptor codes when (20) is used. Note the improvement with respect to non-optimized $(-.)$-curve, although an outer optimization on α is not run.

From Fig. 9, we observe that the performance of the JSCC scheme assuming ideal codes, $(*)$-curve, is better than the baseline scheme in two different aspects. First, the required b value is lower for any channel condition. Second, the JSCC linear encoding together with the MWTD optimization can provide a "smooth" trade-off unlike the "knee" shaped curves of the separated scheme. In Fig. 9, we observe that infinite length nonbinary Raptor codes are competitive with the lower envelope of the baseline scheme. We believe that the infinite length results can be further improved using the outer optimization on α values.

For finite length results we first focus on $E_s/N_0 = 3$ dB point where finite length results are obtained with classic LT sequence and optimized LT sequences. The finite length results of optimized and non-optimized sequences are indicated in Fig. 9 with a $\diamond$ and $\square$, respectively. Both the $\diamond$ and $\square$ are significantly above the baseline scheme's curve. Note that at $E_s/N_0 = 3$ dB, the channel code ($R_c = 3/4$) of the baseline scheme is at its matched point with the given SNR. Note that the baseline scheme switches to $R_c = 1/2$ around 2.8 dB. b value of the baseline scheme significantly increases even passes the $\diamond$ point as SNR gets lower due retransmissions keeping a fixed PSNR value. JSCC is designed to be robust to mismatched channel conditions hence JSCC can work with the same value of b given by the vertical height of $\diamond$ point. Note that keeping b fixed also avoids extra transmission delay and cost. Obviously due to mismatched channel conditions, there will be residual error in plane reconstructions. But as seen in Fig. (10-14) PSNR is gracefully degraded and the image quality given in these reconstructions are perceptually acceptable depending on the application since there are no artificial "block effects" due to loss segments.

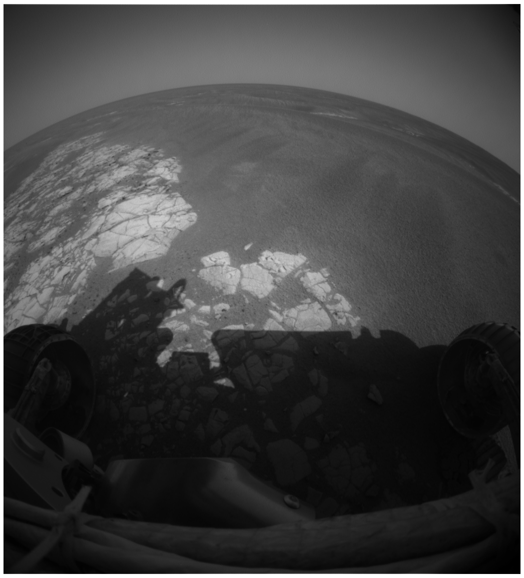

Figure 10. Original image 1F178787358EFF5927P1219L0M1.pgm

References

[1] O. Y. Bursalioglu, M. Fresia, G. Caire, and H. V. Poor, "Lossy multicasting over binary symmetric broadcast channels," January 2011, submitted to *IEEE Trans. Signal Processing* available at http://www-scf.usc.edu/ bursalio/.

[2] ——, "Lossy joint source-channel coding using raptor codes," *Int. Journal of Digital Multimedia Broadcasting*, vol. 2008, Article ID 124685, 18 pages.

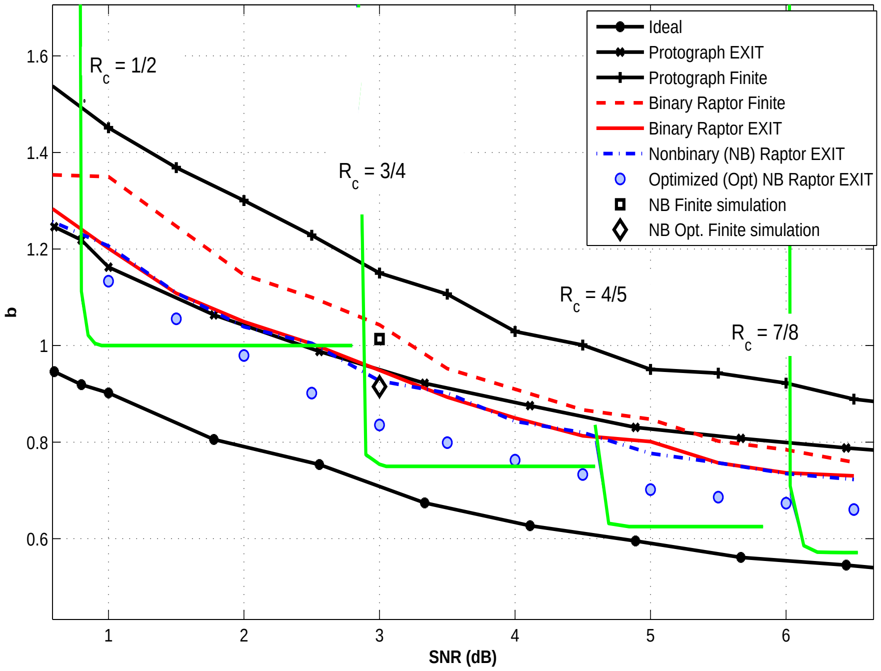

Figure 9. b vs SNR trade-off for various cases

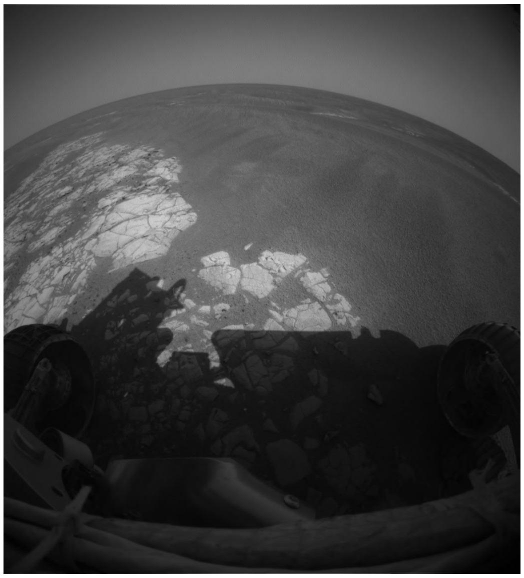

Figure 11. Image reconstruction at SNR $=$ 3 dB, PSNR $=$ 49 dB

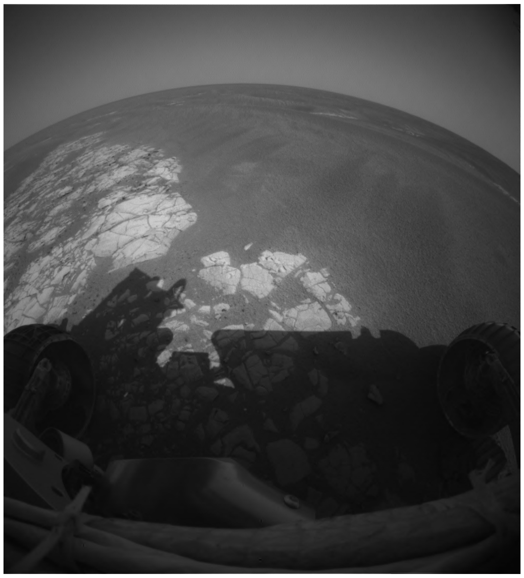

Figure 12. Image reconstruction at SNR $=$ 2.8 dB, PSNR $=$ 48.19 dB

[3] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[4] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov 1998.

[5] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practices*. Norwell, MA: Kluwer Academics Publishers, 2002.

[6] A. Cohen, I. Daubechies, and J. Feaeveau, "Biorthogonal bases of compactly supported wavelets," *Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.

[7] O. Y. Bursalioglu, G. Caire, and D. Divsalar, "Joint source-channel coding for deep space image transmission," *submitted to JPL, TMO Progr. Rep*, 2010.

[8] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp.

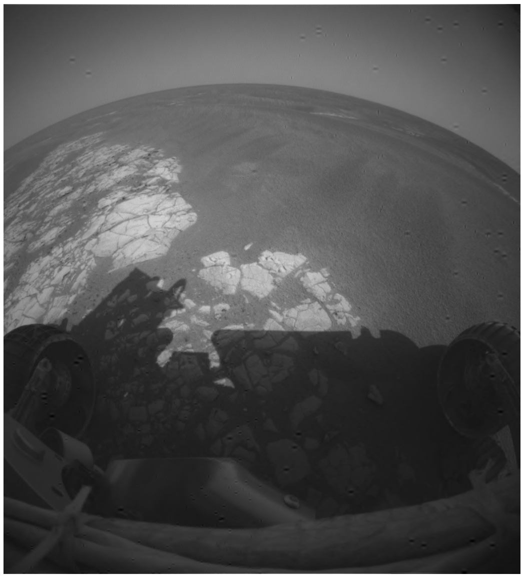

Figure 13. Image reconstruction at $\mathsf{SNR} = 2.5$ dB, $\mathsf{PSNR} = 45.63$ dB

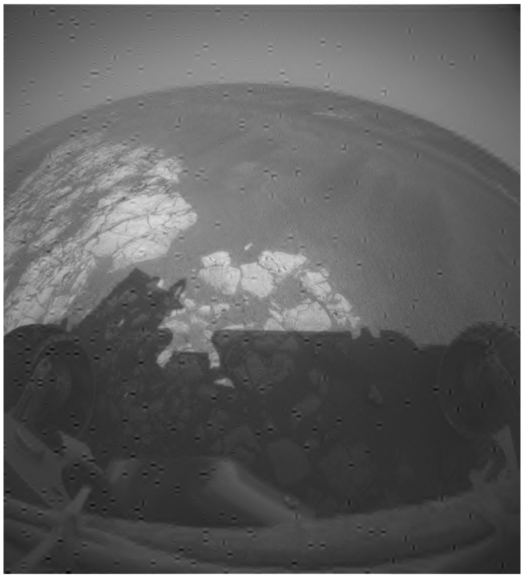

Figure 14. Image reconstruction at $\mathsf{SNR} = 2$ dB, $\mathsf{PSNR} = 38.60$ dB

2033–2051, May 2006.

[9] A. Bennatan and D. Burshtein, “Design and analysis of nonbinary ldpc codes for arbitrary discrete-memoryless channels,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 549 – 583, 2006.

[10] K. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. Jones, and F. Pollara, “The development of turbo and ldpc codes for deep-space applications,” *Proceedings of the IEEE*, vol. 95, no. 11, 2007.

[11] A. Kiely and M. Klimesh, “The icer progressive wavelet image compressor,” *IPN Progress Report*, vol. 42-155, pp. 1–46, 2003.

[12] *Low Density Parity Check Codes for Use in Near-Earth and Deep Space Applications*. (131.1-O-2 Orange Book): Consultative Committee for Space Data Systems (CCSDS), 2007.

[13] A. Shokrollahi, “Raptor codes,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 2551 – 2567, June 2006.