Optimal Parameter Choice for Wyner-Ziv Coding of Laplacian Sources with Decoder Side-Information

Debargha Mukherjee
Media Technologies Laboratory
HP Laboratories Palo Alto

HPL-2007-34
March 6, 2007*

source coding with side-information, Wyner-Ziv, Laplacian distribution, Gaussian distribution, channel codes, LDPC codes, Turbo codes, arithmetic codes

A large number of practical coding scenarios deal with sources, for instance transform coefficients that can be well modeled as Laplacians. In regular practical coding of such sources, samples are often quantized by a family of uniform quantizers possibly with a deadzone, and then entropy coded. For the Wyner-Ziv coding problem when correlated side-information is available at the decoder, the side-information can be modeled as obtained by additive Gaussian or Laplacian noise on the source. This paper deals with optimal choice of parameters for practical coding of such sources in presence of side-information, using the same quantizer structure as in the regular codec, assuming that the variances of the source and additive noise are known. We first consider memoryless coding which may be the only option in some coding scenarios, and then follow up by considering coding using powerful channel codes with soft decoding that approach the Slepian Wolfe bound. We show that in the latter case, at practical block lengths and code complexities, not pure channel coding but a hybrid combination of source coding and channel coding provides optimal rate-distortion performance. A good understanding of the optimal parameter choice mechanism is essential for building practical codecs that can be used in a variety of scenarios.

* Internal Accession Date Only
Approved for External Publication

OPTIMAL PARAMETER CHOICE FOR WYNER-ZIV CODING OF LAPLACIAN SOURCES WITH DECODER SIDE-INFORMATION

Debargha Mukherjee
HP Labs Technical Report: HPL-2007-34
Hewlett Packard Laboratories, Palo Alto, California, USA
Email: debargha@hpl.hp.com

ABSTRACT

A large number of practical coding scenarios deal with sources, for instance transform coefficients that can be well modeled as Laplacians. In regular practical coding of such sources, samples are often quantized by a family of uniform quantizers possibly with a deadzone, and then entropy coded. For the Wyner-Ziv coding problem when correlated side-information is available at the decoder, the side-information can be modeled as obtained by additive Gaussian or Laplacian noise on the source. This paper deals with optimal choice of parameters for practical coding of such sources in presence of side-information, using the same quantizer structure as in the regular codec, assuming that the variances of the source and additive noise are known. We first consider memoryless coding which may be the only option in some coding scenarios, and then follow up by considering coding using powerful channel codes with soft decoding that approach the Slepian Wolfe bound. We show that in the latter case, at practical block lengths and code complexities, not pure channel coding but a hybrid combination of source coding and channel coding provides optimal rate-distortion performance. A good understanding of the optimal parameter choice mechanism is essential for building practical codecs that can be used in a variety of scenarios.

1. INTRODUCTION

Drawing inspiration from the foundation laid by Slepian-Wolfe [1] and Wyner-Ziv [2] theorems, a great deal of attention has been devoted in recent years to practical source coding with side-information problems [3]-[13]. All such work uses some form of channel coding to convey a source assuming that correlated side-information is available at the decoder to perform appropriate channel decoding. The challenge is to design the encoder and decoder based on known statistics. A good review of the area is presented in [11]. In this work, we address a fundamental problem of optimal parameter choice for various coding options that may be considered in a practical source coding with side-information scenario, where a range of rate-distortion trade-offs are desired, assuming a model for the data and known statistics.

It has generally been accepted that Wyner-Ziv coding in the transform domain yields superior results due to effective decorrelation of the data. Transform coefficients are well modeled as Laplacians. Further, the side-information available only at the decoder in a variety of applications can be well modeled by additive noise on the source where the noise is either Gaussian or additive Laplacian. In particular, if X denotes the source Laplacian random variable with variance σ_X^2, and Y is the side-information available only at the decoder, then $Y = X + Z$, where Z is either *i.i.d.* Gaussian or *i.i.d.* Laplacian with variance σ_Z^2 in our model Generally, we will refer to the probability density function of X as $f_X(x)$, and Z as $f_Z(z)$. The paper studies and optimizes various practical Wyner-Ziv coding options for such a source and correlation model, which we believe would be very useful in many transform-domain Wyner-Ziv coding scenarios.

In any practical codec, X is quantized with a quantizer family ϕ to yield a quantization index random variable Q as follows: $Q = \phi(X, QP)$, where QP is the quantization step-size parameterizing a family of quantizers. The simplest quantizer family is the uniform quantizer, given by:

$$Q = \phi(X, QP) = round(X / QP) \tag{1}$$

A variant referred to as the uniform quantizer with deadzone, is actually more commonly used in practical codecs:

$$Q = \phi(X, QP) = sign(X) \times \lfloor |X| / QP \rfloor \tag{2}$$

While in general QP can be continuous, it is typically taken from a discrete set in a practical codec.

For these quantizers, Q is ideally assumed to take integer values in $\Omega_Q = \{-\infty, ..., -1, 0, 1, ..., \infty\}$. However, in practice the finite set $\Omega_Q = \{-q_{\max}, -q_{\max+1}, ..., -1, 0, 1, ..., q_{\max} - 1, q_{\max}\}$ may be used, where q_{max} is large enough so that the probability of the Laplacian source to take positive and negative values beyond the bins q_{max} and $-q_{max}$ respectively is negligible. While generally speaking q_{max} depends on QP and σ_X^2, for simplicity of notation we simply refer to the set of all available quantization bins as Ω_Q.

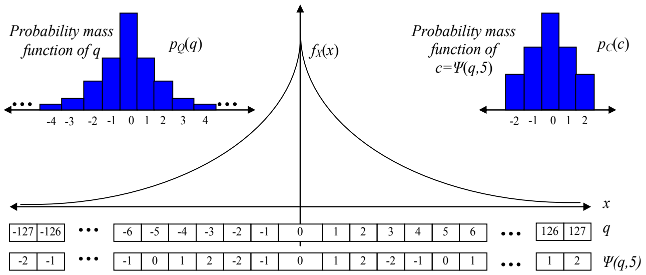


Figure 1. Probability mass function of coset indices

In the source coding with side-information scenario under consideration in this work, we assume that the same quantizer structure as in the regular codec, is used. The problem we address is then broadly stated as follows: Given a target upper-limit D_t on the overall expected distortion, and variances $\{\sigma_X^2, \sigma_z^2\}$ for Laplacian X and Gaussian Z respectively, how should X be coded based on a given quantizer structure, so that the expected rate is minimized. Sometimes, it may be convenient to specify the target distortion D_t in terms of a target quantization parameter QP_t assuming regular coding (with no side-information) based on the same quantizer family. This criterion will be referred to as *distortion target matching*.

2. MEMORYLESS COSET CODES

2.1. Coset Encoding and Decoding

In many source coding with side-information scenarios, it may be inconvenient to use longer channel codes, either because decoding has to be conducted fast, or simply because there may not be enough data samples available to code in order to reap the benefits of channel coding. In such cases, memoryless coset codes can be used.

But before defining these codes, we define the circular modulus function mod_c of two integers I, J as follows:

$$\text{mod}_c(I,J) = I - J\lfloor I/J \rfloor \tag{3}$$

taking values in the set $\{0, 1, \ldots, J–1\}$. A variant mod_{cz} of the function uses zero-centered circular modulus as follows:

$$\text{mod}_{c_z}(I,J) = \begin{cases} \text{mod}_c(I,J), & \text{mod}_c(I,J) < J/2 \\ \text{mod}_c(I,J) - J, & \text{mod}_c(I,J) \geq J/2 \end{cases} \tag{4}$$

taking values in the set $\{\lfloor -(J-1)/2 \rfloor, \ldots, -1, 0, 1, \ldots, \lfloor (J-1)/2 \rfloor\}$.

Once X has been quantized to Q with quantizer ϕ using parameter QP, cosets are computed based on Q to yield a coset index random variable C: $C = \psi(Q, M) = \psi(\phi(X, QP), M)$, M being the coset modulus, as follows:

$$C = \psi(Q,M) = \text{mod}_c(Q,M) \tag{5}$$

C takes values from the set $\Omega_C = \{0, 1, \ldots, M\text{-}1\}$. The *zero-centered* variant where coset indices are centered on zero may be preferred if an existing entropy coder for regular coding is used to code the coset indices. In this case,

$$C = \psi(Q,M) = \text{mod}_{cz}(Q,M) \tag{6}$$

C takes values from the set $\Omega_C = \{\lfloor -(M-1)/2 \rfloor, \ldots, -1, 0, 1, \ldots, \lfloor (M-1)/2 \rfloor\}$. However if a different entropy coder were designed for the coset indices, there would be no difference between them.

If quantization bin q corresponds to interval $[x_l(q), x_h(q)]$, then the probability of the bin $q \in \Omega_Q$, and the probability of a coset index $c \in \Omega_C$ are given by the probability mass functions:

$$p_Q(q) = \int_{x_l(q)}^{x_h(q)} f_X(x)dx \qquad p_C(c) = \sum_{q\in\Omega_Q:\psi(q,M)=c} p_Q(q) = \sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} f_X(x)dx \tag{7}$$

Examples of both are shown in Figure 1, for M odd in the zero-centered case and Laplacian $f_X(x)$. Note that the entropy coder that exists in the regular coder is optimized for the distribution $p_Q(q)$, and is designed to be particularly efficient for coding zeros. Because the distribution $p_C(c)$ is also symmetric for odd M, has zero as its mode and decays with increasing magnitude, the entropy coder for Q that already exists in the regular code may be reused for C, without significant loss in efficiency. However, if an entropy coder was designed specifically for coset indices there would be no difference between use of Eq. 5 and Eq. 6 for the coset modulus function.

For decoding, the minimum MSE reconstruction function $\hat{X}_{YC}(y,c)$ based on unquantized side information y and received coset index c, is given by:

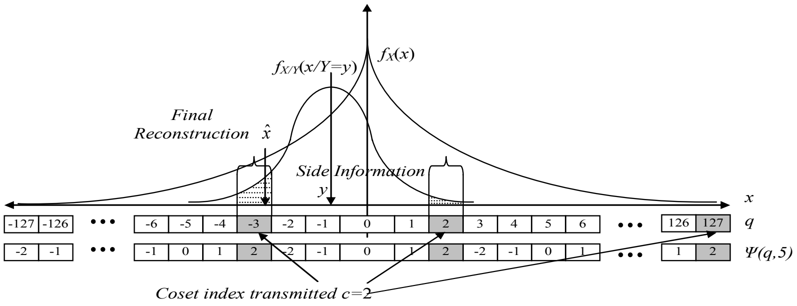


Figure 2. Decoding example

$$\hat{X}_{YC}(y,c)=E(X/Y=y,C=c)=E(X/Y=y,\psi(\phi(X,QP),M)=c)=\frac{\sum\limits_{q\in\Omega_Q:\psi(q,M)=c}\int\limits_{x_l(q)}^{x_h(q)}xf_{X/Y}(x,y)dx}{\sum\limits_{q\in\Omega_Q:\psi(q,M)=c}\int\limits_{x_l(q)}^{x_h(q)}f_{X/Y}(x,y)dx}=\frac{\sum\limits_{q\in\Omega_Q:\psi(q,M)=c}\mu(q,y)}{\sum\limits_{q\in\Omega_Q:\psi(q,M)=c}\pi(q,y)} \tag{8}$$

where we have introduced the following definitions for convenience in the rest of the paper:

$$\pi(q,y)=p_{Q/Y}(Q=q/Y=y)=\int\limits_{x_l(q)}^{x_h(q)}f_{X/Y}(x,y)dx,\quad \mu(q,y)=\int\limits_{x_l(q)}^{x_h(q)}xf_{X/Y}(x,y)dx \tag{9}$$

Note that $\pi(q,y)$ is the conditional probability of Q given Y.

Figure 2 depicts such a decoding example.

QP and M should be optimally chosen for a given target quantization parameter QP_t, and known statistics $\{\sigma_X^2, \sigma_Z^2\}$. We study this problem in detail in the rest of this section. Specifically, the rate-distortion function of memoryless coset codes are characterized in order to obtain the optimal $\{QP, M\}$ pair that yields reconstruction quality equivalent to a target quantization step size QP_t if regular (non-distributed) coding had been used. The variances of the Laplacian source (σ_X^2) and the additive white Gaussian noise (σ_Z^2), are assumed to be known.

2.2. General Rate-Distortion Characterization

In this sub-section we write down rate-distortion function expressions for various Wyner-Ziv coding scenarios, without consideration for the source and correlation model. In the next sub-section, we will specialize for the Laplacian X and Gaussian Z case, with side-information $Y=X+Z$.

2.2.1. Memoryless coset codes followed by minimum MSE reconstruction with side-information

The probability of each coset index transmitted is known from the probability mass function in Eq. 7. Assuming an ideal entropy coder for the coset indices, the expected rate would be the entropy of the source C:

$$E(R_{YC})=H(C)=-\sum_{c\in\Omega_C}p_C(c)\log_2 p_C(c)=-\sum_{c\in\Omega_C}\{\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)}f_X(x)dx\}\log_2\{\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)}f_X(x)dx\} \tag{10}$$

Defining $m_X^{(i)}(x)=\int\limits_{-\infty}^{x}x'^i f_X(x')dx'$, we can rewrite:

$$E(R_{YC})=-\sum_{c\in\Omega_C}\{\sum_{q\in\Omega_Q:\psi(q,M)=c}[m_X^{(0)}(x_h(q))-m_X^{(0)}(x_l(q))]\}\log_2\{\sum_{q\in\Omega_Q:\psi(q,M)=c}[m_X^{(0)}(x_h(q))-m_X^{(0)}(x_l(q))]\} \tag{11}$$

Assuming the minimum mean-squared-error reconstruction function in Eq. 8, the expected distortion D_{YC} given side information y and coset index c is given by:

$$E(D_{YC}/Y=y,C=c)=E([X-\hat{X}_{YC}(y,c)]^2/Y=y,C=c)=E(X^2/Y=y,C=c)-\hat{X}_{YC}(y,c)^2 \tag{12}$$

using $\hat{X}_{YC}(y,c)=E(X/Y=y,C=c)$. Marginalizing over y and c yields:

$$
\begin{aligned}
E(D_{YC}) &= E(X^2) - \int_{-\infty}^{\infty} \{ \sum_{c\in\Omega_C} \hat{X}_{YC}(y,c)^2 p_{C/Y}(C=c/Y=y)\} f_Y(y)dy \\
&== \sigma_X^2 - \int_{-\infty}^{\infty} \{ \sum_{c\in\Omega_C} \left(\frac{\sum_{q\in\Omega_Q:\psi(q,M)=c} \mu(q,y)}{\sum_{q\in\Omega_Q:\psi(q,M)=c} \pi(q,y)} \right)^2 p_{C/Y}(C=c/Y=y)\} f_Y(y)dy \\
&= \sigma_X^2 - \int_{-\infty}^{\infty} \{ \sum_{c\in\Omega_C} \left(\frac{\sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx}{\sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx} \right)^2 p_{C/Y}(C=c/Y=y)\} f_Y(y)dy
\end{aligned}
\tag{13}
$$

where $p_{C/Y}(C=c/Y=y)$ is the conditional probability mass function of *C* given *Y*. Noting that:

$$
p_{C/Y}(C=c/Y=y) = \sum_{q\in\Omega_Q:\psi(q,M)=c} p_{Q/Y}(Q=q/Y=y) = \sum_{q\in\Omega_Q:\psi(q,M)=c} \pi(q,y) = \sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx \tag{14}
$$

we can write:

$$
E(D_{YC}) = \sigma_X^2 - \int_{-\infty}^{\infty} \{ \sum_{c\in\Omega_C} \frac{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c} \mu(q,y) \right)^2}{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c} \pi(q,y) \right)} \} f_Y(y)dy = \sigma_X^2 - \int_{-\infty}^{\infty} \{ \sum_{c\in\Omega_C} \frac{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx \right)^2}{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx \right)} \} f_Y(y)dy \tag{15}
$$

Defining:

$$
m_{X/Y}^{(i)}(x,y) = \int_{-\infty}^{x} x'^i f_{X/Y}(x',y)dx' \tag{16}
$$

we have:

$$
\begin{aligned}
\pi(q,y) &= \int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx = [m_{X/Y}^{(0)}(x_h(q),y) - m_{X/Y}^{(0)}(x_l(q),y)] \\
\mu(q,y) &= \int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx =[m_{X/Y}^{(1)}(x_h(q),y) - m_{X/Y}^{(1)}(x_l(q),y)]
\end{aligned}
\tag{17}
$$

Then we can rewrite Eq. 15 as:

$$
E(D_{YC}) = \sigma_X^2 - \int_{-\infty}^{\infty} \{ \sum_{c\in\Omega_C} \frac{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c} [m_{X/Y}^{(1)}(x_h(q),y) - m_{X/Y}^{(1)}(x_l(q),y)] \right)^2}{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c} [m_{X/Y}^{(0)}(x_h(q),y) - m_{X/Y}^{(0)}(x_l(q),y)] \right)} \} f_Y(y)dy \tag{18}
$$

2.2.2. Ideal Slepian-Wolf coding followed by minimum MSE reconstruction with side-information

Next, we consider the expected rate and distortion when using ideal Slepian-Wolf coding for the quantization bins. The ideal Slepian Wolf coder would use a rate no larger than *H*(*Q*/*Y*) to convey the quantization bins error-free. Once the bins have been conveyed error-free, a minimum MSE reconstruction can be still conducted but only within the decoded bin. The expected rate is then given by:

$$
\begin{aligned}
E(R_{YQ}) &= H(Q/Y) \\
&= -\int_{-\infty}^{\infty} \{ \sum_{q\in\Omega_Q} p_{Q/Y}(Q=q/Y=y) \log_2 p_{Q/Y}(Q=q/Y=y) \} f_Y(y)dy = -\int_{-\infty}^{\infty} \{ \sum_{q\in\Omega_Q} \pi(q,y) \log_2 \pi(q,y) \} f_Y(y)dy \\
&= -\int_{-\infty}^{\infty} \{ \sum_{q\in\Omega_Q} [m_{X/Y}^{(0)}(x_h(q),y) - m_{X/Y}^{(0)}(x_l(q),y)] \log_2 [m_{X/Y}^{(0)}(x_h(q),y) - m_{X/Y}^{(0)}(x_l(q),y)]] \} f_Y(y)dy
\end{aligned}
\tag{19}
$$

The expected Distortion D_{YQ} is the distortion incurred by a minimum MSE reconstruction function within a quantization bin, given the side information *y* and bin index *q*. This reconstruction function $\hat{X}_{YQ}(y,q)$ is given by:

$$\begin{aligned}\hat{X}_{YQ}(y,q) &= E(X/Y=y,Q=q) = E(X/Y=y,\phi(X,QP)=c)\\ &= \frac{\int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx}{\int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx} = \frac{\mu(q,y)}{\pi(q,y)} = \frac{m^{(1)}_{X/Y}(x_h(q),y) - m^{(1)}_{X/Y}(x_l(q),y)}{m^{(0)}_{X/Y}(x_h(q),y) - m^{(0)}_{X/Y}(x_l(q),y)}\end{aligned} \tag{20}$$

Using this reconstruction, the expected Distortion with noise-free quantization bins (denoted D_{YQ}) is given by:

$$\begin{aligned}E(D_{YQ}) &= \sigma_X^2 - \int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}\frac{\left(\int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx\right)^2}{\left(\int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx\right)}\}f_Y(y)dy = \sigma_X^2 - \int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}\frac{\mu(q,y)^2}{\pi(q,y)}\}f_Y(y)dy\\ &= \sigma_X^2 - \int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}\frac{\left(m^{(1)}_{X/Y}(x_h(q),y) - m^{(1)}_{X/Y}(x_l(q),y)\right)^2}{\left(m^{(0)}_{X/Y}(x_h(q),y) - m^{(0)}_{X/Y}(x_l(q),y)\right)}\}f_Y(y)dy\end{aligned} \tag{21}$$

This case will be considered in more detail in the Section 3, but we present the rate-distortion characterization here in order to make a comparison with memoryless codes.

2.2.3. Regular encoding followed by minimum MSE reconstruction with and without side-information

Next, we consider the rate and distortion if no distributed coding on the quantization bins were done at the encoder. In this case, the expected rate is just the entropy of Q.

$$E(R_Q) = H(Q) = -\sum_{q\in\Omega_Q} p_Q(q)\log_2 p_Q(q) = -\sum_{q\in\Omega_Q}[m^{(0)}_X(x_h(q)) - m^{(0)}_X(x_l(q))]\log_2[m^{(0)}_X(x_h(q)) - m^{(0)}_X(x_l(q))] \tag{22}$$

The decoder can still use distributed decoding if side-information Y is available. In this case, the reconstruction function and the corresponding expected distortion are given by Eq. 20 and Eq. 21 respectively. On the other hand, if there is no side-information available, the expected distortion D_Q is the distortion incurred by a minimum MSE reconstruction function just based on the bin index q. This reconstruction function $\hat{X}_Q(q)$ is then given by:

$$\hat{X}_Q(q) = E(X/Q=q) = E(X/\phi(X,QP)=q) = \frac{\int_{x_l(q)}^{x_h(q)} x f_X(x)dx}{\int_{x_l(q)}^{x_h(q)} f_X(x)dx} = \frac{m^{(1)}_X(x_h(q)) - m^{(1)}_X(x_l(q))}{m^{(0)}_X(x_h(q)) - m^{(0)}_X(x_l(q))} \tag{23}$$

while the expected distortion is given by:

$$E(D_Q) = \sigma_X^2 - \sum_{q\in\Omega_Q}\frac{\left(\int_{x_l(q)}^{x_h(q)} x f_X(x)dx\right)^2}{\left(\int_{x_l(q)}^{x_h(q)} f_X(x)dx\right)} = \sigma_X^2 - \sum_{q\in\Omega_Q}\frac{\left(m^{(1)}_X(x_h(q)) - m^{(1)}_X(x_l(q))\right)^2}{\left(m^{(0)}_X(x_h(q)) - m^{(0)}_X(x_l(q))\right)} \tag{24}$$

The overall objective of the distortion matched parameter choice mechanism can now be expressed in terms of the above rate-distortion functions: Given a target quantization step size QP_t for regular encoding and decoding, the target expected distortion $E(D_Q)$ can be readily computed from Eq. 24. The parameters QP and M for memoryless coset codes should be chosen such that the lowest rate $E(R_{YC})$ given by Eq. 10 is obtained, with the expected distortion $E(D_{YC})$ given by Eq. 18 being equivalent to the target distortion.

2.2.4. Zero rate encoder with minimum MSE reconstruction with side-information

The final case is when no information is transmitted corresponding to X, so that the rate is 0. The decoder performs the minimum MSE reconstruction function $\hat{X}_Y(y)$:

$$\hat{X}_Y(y) = E(X/Y=y) = \int_{-\infty}^{\infty} x f_{X/Y}(x,y)dx = m^{(1)}_{X/Y}(\infty,y) \tag{25}$$

The expected zero-rate distortion D_Y is given by:

$$E(D_Y) = \sigma_X^2 - \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} x f_{X/Y}(x,y)dx\right)^2 f_Y(y)dy = \sigma_X^2 - \int_{-\infty}^{\infty} m^{(1)}_{X/Y}(\infty,y)^2 f_Y(y)dy \tag{26}$$

2.3. Laplacian Source Specifics

While the expressions in the previous sub-section are generic, we now specialize for the case of Laplacian X and for two particular cases for Z, Gaussian and Laplacian. While the results presented in the rest of the paper correspond only to these models, the methodology presented applies to any other viable model, including a mixture of Gaussians.

2.3.1. Expressions for Gaussian noise Z

We first specialize for the case of Laplacian X and Gaussian Z, *i.e.*:

$$f_X(x) = \frac{1}{\sqrt{2}\sigma_X} e^{-\left|\frac{\sqrt{2}x}{\sigma_X}\right|}, \quad f_Z(z) = \frac{1}{\sqrt{2\pi}\sigma_Z} e^{-\frac{1}{2}\left|\frac{z}{\sigma_Z}\right|^2} \tag{27}$$

In the following, we assume:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{28}$$

Then, defining

$$\beta(x) = e^{\frac{\sqrt{2}x}{\sigma_X}} \tag{29}$$

we have

$$m_X^{(0)}(x) = \begin{cases} \dfrac{\beta(x)}{2}, & x \le 0 \\ 1 - \dfrac{1}{2\beta(x)}, & x > 0 \end{cases} \quad m_X^{(1)}(x) = \begin{cases} \dfrac{\beta(x)}{2\sqrt{2}}(\sqrt{2}x - \sigma_X), & x \le 0 \\ -\dfrac{1}{2\sqrt{2}\beta(x)}(\sqrt{2}x + \sigma_X), & x > 0 \end{cases} \tag{30}$$

Further defining:

$$\gamma_1(x) = erf(\frac{\sigma_X x - \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}), \quad \gamma_2(x) = erf(\frac{\sigma_X x + \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}) \tag{31}$$

and using Y=X+Z, we have:

$$\begin{aligned} f_{XY}(x,y) &= \frac{1}{2\sqrt{\pi}\sigma_X\sigma_Z} e^{-\left|\frac{x\sqrt{2}}{\sigma_X}\right|} e^{-\frac{1}{2}(\frac{y-x}{\sigma_Z})^2} \\ f_Y(y) &= \int_{-\infty}^{\infty} f_{XY}(x,y)dx = \frac{1}{2\sqrt{2}\beta(y)\sigma_X} e^{\sigma_X^2/\sigma_Z^2}[\gamma_1(y)+1.0 - \beta(y)^2(\gamma_2(y)-1.0)] \\ f_{X/Y}(x,y) &= \frac{f_{XY}(x,y)}{f_Y(y)} = \frac{\sqrt{2}\beta(y)}{\sqrt{\pi}\sigma_Z} \frac{e^{-\left|\frac{x\sqrt{2}}{\sigma_X}\right| - \frac{1}{2}(\frac{y-x}{\sigma_Z})^2 - \frac{\sigma_X^2}{\sigma_Z^2}}}{[\gamma_1(y)+1.0 - \beta(y)^2(\gamma_2(y)-1.0)]} \end{aligned} \tag{32}$$

Given $f_{X/Y}(x, y)$, the moments can now be computed:

$$\begin{aligned} m_{X/Y}^{(0)}(x,y) &= \begin{cases} \dfrac{1}{[\gamma_1(y)+1.0-\beta(y)^2(\gamma_2(y)-1.0)]}\beta(y)^2[1-erf(\dfrac{\sigma_X(y-x)+\sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z})], & x \le 0 \\ 1 - \dfrac{1}{[\gamma_1(y)+1.0-\beta(y)^2(\gamma_2(y)-1.0)]}[1+erf(\dfrac{\sigma_X(y-x)-\sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z})], & x > 0 \end{cases} \\ m_{X/Y}^{(1)}(x,y) &= \begin{cases} \dfrac{\beta(y)^2[y+\sqrt{2}\dfrac{\sigma_Z^2}{\sigma_X}][1-erf(\dfrac{\sigma_X(y-x)+\sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z})] - \dfrac{\sqrt{2}}{\sqrt{\pi}}\sigma_Z\beta(x)^2 e^{\frac{-\frac{1}{2}\left(\sigma_X(y-x)-\sqrt{2}\sigma_Z^2\right)^2}{\sigma_X^2\sigma_Z^2}}}{[\gamma_1(y)+1-\beta(y)^2(\gamma_2(y)-1)]}, & x \le 0 \\ \dfrac{-\beta(y)^2[y+\sqrt{2}\dfrac{\sigma_Z^2}{\sigma_X}](\gamma_2(y)-1) + [y-\sqrt{2}\dfrac{\sigma_Z^2}{\sigma_X}][\gamma_1(y) - erf(\dfrac{\sigma_X(y-x)-\sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z})] - \dfrac{\sqrt{2}}{\sqrt{\pi}}\sigma_Z e^{\frac{-\frac{1}{2}\left(\sigma_X(y-x)-\sqrt{2}\sigma_Z^2\right)^2}{\sigma_X^2\sigma_Z^2}}}{[\gamma_1(y)+1-\beta(y)^2(\gamma_2(y)-1)]}, & x > 0 \end{cases} \end{aligned} \tag{33}$$

A special case used for the optimal reconstruction and distortion functions in the zero-rate case is when x→∞. In this case,

$$m^{(1)}_{X/Y}(\infty, y) = \frac{-\beta(y)^2[y+\sqrt{2}\frac{\sigma_Z^2}{\sigma_X}](\gamma_2(y)-1)+[y-\sqrt{2}\frac{\sigma_Z^2}{\sigma_X}](\gamma_1(y)+1)}{[\gamma_1(y)+1-\beta(y)^2(\gamma_2(y)-1)]} = y-\sqrt{2}\frac{\sigma_Z^2}{\sigma_X}\frac{[\gamma_1(y)+1+\beta(y)^2(\gamma_2(y)-1)]}{[\gamma_1(y)+1-\beta(y)^2(\gamma_2(y)-1)]} \tag{34}$$

The *erf*() function used in the above expressions for moments and $f_Y(y)$ can be evaluated based on a 9th order polynomial approximation provided in *Numerical Recipes* [14]. All the expected rate and distortion functions in Section 5.1 then can be evaluated based on these moments in conjunction with numerical integration with $f_Y(y)$, given the quantization function ϕ and the coset modulus function ψ .

2.3.2. Expressions for Laplacian noise Z

We next specialize for the case of Laplacian *X* and Laplacian *Z, i.e.*:

$$f_X(x) = \frac{1}{\sqrt{2}\sigma_X}e^{-\left|\frac{\sqrt{2}x}{\sigma_X}\right|}, \quad f_Z(z) = \frac{1}{\sqrt{2}\sigma_Z}e^{-\left|\frac{\sqrt{2}z}{\sigma_Z}\right|} \tag{35}$$

Defining:

$$\alpha(x) = e^{\frac{\sqrt{2}x}{\sigma Z}}, \ \beta(x) = e^{\frac{\sqrt{2}x}{\sigma X}}, \ \gamma(x) = (\sigma_X\alpha(x) - \sigma_Z\beta(x)) \tag{36}$$

Eq. 30 still applies for $m_X^{(0)}(x)$ and $m_X^{(1)}(x)$. Further, using *Y*=*X*+*Z*, we have

$$f_{XY}(x,y) = \frac{1}{2\sigma_X\sigma_Z}e^{-\sqrt{2}\left(\frac{|x|}{\sigma X}+\frac{|x-y|}{\sigma Z}\right)}$$

$$f_Y(y) = \begin{cases} \dfrac{\gamma(|y|)e^{-\sqrt{2}|y|\left(\frac{1}{\sigma X}+\frac{1}{\sigma Z}\right)}}{\sqrt{2}(\sigma_X^2-\sigma_Z^2)}, & \sigma_Z \neq \sigma_X \\ \dfrac{(\sigma_X+\sqrt{2}|y|)e^{-\sqrt{2}\frac{|y|}{\sigma X}}}{2\sqrt{2}\sigma_X^2}, & \sigma_Z = \sigma_X \end{cases} \qquad f_{X/Y}(x,y) = \begin{cases} \dfrac{e^{-\sqrt{2}\left(\frac{|x|-|y|}{\sigma X}+\frac{|x-y|-|y|}{\sigma Z}\right)}(\sigma_X^2-\sigma_Z^2)}{\sqrt{2}\sigma_X\sigma_Z\gamma(|y|)}, & \sigma_Z \neq \sigma_X \\ \dfrac{\sqrt{2}e^{-\sqrt{2}\left(\frac{|x|-|y|+|x-y|}{\sigma X}\right)}}{(\sigma_X+\sqrt{2}|y|)}, & \sigma_Z = \sigma_X \end{cases} \tag{37}$$

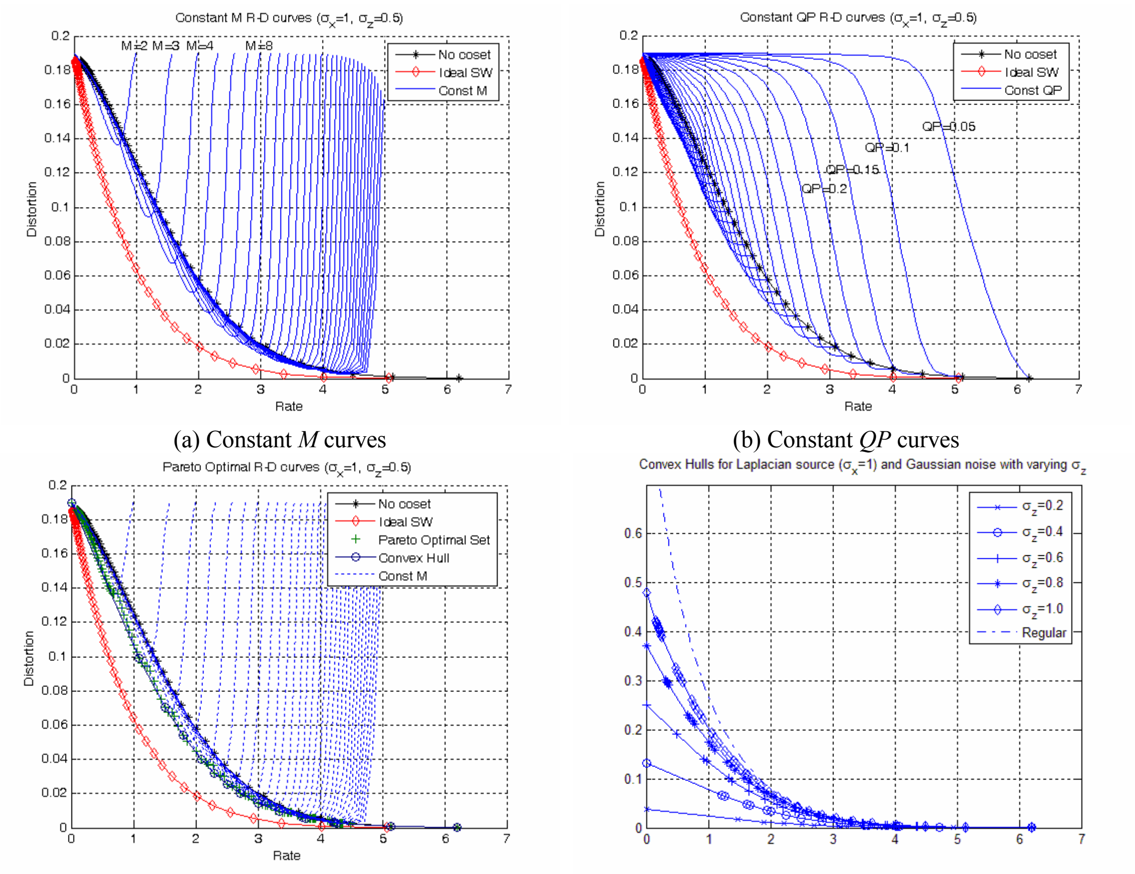
The partial moments can now be calculated as follows:

$$m^{(0)}_{X/Y}(x,y) = \begin{cases} 1-m^{(0)}_{X/Y+}(-x,y) & y \le 0 \\ m^{(0)}_{X/Y+}(x,y) & y > 0 \end{cases} \quad \text{where } m^{(0)}_{X/Y+}(x,y) = \left\{ \left.\begin{cases} \dfrac{(\sigma_X-\sigma_Z)\beta(x+|y|)\alpha(x)}{2\gamma(|y|)} & x \le 0 \\ \dfrac{\beta(|y|)((\sigma_X+\sigma_Z)\alpha(x)/\beta(x)-2\sigma_Z)}{2\gamma(|y|)} & x \le |y| \\ 1-\dfrac{(\sigma_X-\sigma_Z)\alpha^2(|y|)\beta(|y|)}{2\gamma(|y|)\alpha(x)\beta(x)} & x > |y| \end{cases}\right\}_{\substack{for \\ \sigma_X \neq \sigma_Z}} \quad \left.\begin{cases} \dfrac{\sigma_X\beta^2(x)}{2(\sigma_X+\sqrt{2}|y|)} & x \le 0 \\ \dfrac{\sigma_X+2\sqrt{2}x}{2(\sigma_X+\sqrt{2}|y|)} & x \le |y| \\ 1-\dfrac{\sigma_X\beta^2(|y|)}{2(\sigma_X+\sqrt{2}|y|)\beta^2(x)} & x > |y| \end{cases}\right\}_{\substack{for \\ \sigma_X = \sigma_Z}} \right.$$

$$m^{(1)}_{X/Y}(x,y) = \begin{cases} m^{(1)}_{X/Y+}(-x,y) - m^{(1)}_{X/Y+}(\infty,y) & y \le 0 \\ m^{(1)}_{X/Y+}(x,y) & y > 0 \end{cases} \quad \text{where:}$$

$$m^{(1)}_{X/Y+}(x,y) = \begin{cases} \left.\begin{cases} \dfrac{(\sqrt{2}x(\sigma_X+\sigma_Z)-\sigma_X\sigma_Z)(\sigma_X-\sigma_Z)\alpha(x)\beta(x)\beta(|y|)}{2\sqrt{2}\gamma(|y|)(\sigma_X+\sigma_Z)} & x \le 0 \\ \dfrac{\beta(|y|)\{(\sigma_X+\sigma_Z)^2(\sqrt{2}x(\sigma_X-\sigma_Z)-\sigma_X\sigma_Z)\alpha(x)/\beta(x)+4\sigma_X^2\sigma_Z^2\}}{2\sqrt{2}\gamma(|y|)(\sigma_X^2-\sigma_Z^2)} & x \le |y| \\ \dfrac{2\sqrt{2}y\beta(|y|)\sigma_X(\sigma_X^2-\sigma_Z^2)+4\sigma_X^2\sigma_Z^2(\beta(|y|)-\alpha(|y|))-(\sigma_X-\sigma_Z)^2(\sigma_X\sigma_Z+\sqrt{2}x(\sigma_X+\sigma_Z))\frac{\alpha^2(|y|)\beta(|y|)}{\alpha(x)\beta(x)}}{2\gamma(|y|)} & x > |y| \end{cases}\right\}_{\substack{for \\ \sigma_X \neq \sigma_Z}} \\ \left.\begin{cases} \dfrac{\sqrt{2}\sigma_X\beta^2(x)(2\sqrt{2}x-\sigma_X)}{8(\sigma_X+\sqrt{2}|y|)} & x \le 0 \\ \dfrac{\sqrt{2}(4x^2-\sigma_X^2)}{8(\sigma_X+\sqrt{2}|y|)} & x \le |y| \\ \dfrac{1}{2}|y| - \dfrac{\sigma_X\beta^2(|y|)(\sqrt{2}\sigma_X+4x)}{8(\sigma_X+\sqrt{2}|y|)\beta^2(x)} & x > |y| \end{cases}\right\}_{\substack{for \\ \sigma_X = \sigma_Z}} \end{cases} \tag{38}$$

Also note:



(a) Constant *M* curves

(b) Constant *QP* curves

(c) Pareto Optimal Set and Convex Hull

(d) Convex Hulls for varying σ_Z

Figure 3. R-D curves obtained by changing *QP* and *M* for Laplacian *X* and Gaussian *Z*.

$$m^{(1)}_{X/Y}(\infty, y) = \mathrm{sgn}(y).m^{(1)}_{X/Y+}(\infty, y), \text{ where:}$$
$$m^{(1)}_{X/Y+}(\infty, y) = \begin{cases} \dfrac{|y|\alpha(|y|)\sigma_X(\sigma_X^2 - \sigma_Z^2) + \sqrt{2}\sigma_X^2\sigma_Z^2(\beta(|y|) - \alpha(|y|))}{\gamma(|y|)(\sigma_X^2 - \sigma_Z^2)}, & \sigma_X \neq \sigma_Z \\ |y|/2, & \sigma_X = \sigma_Z \end{cases} \quad (39)$$

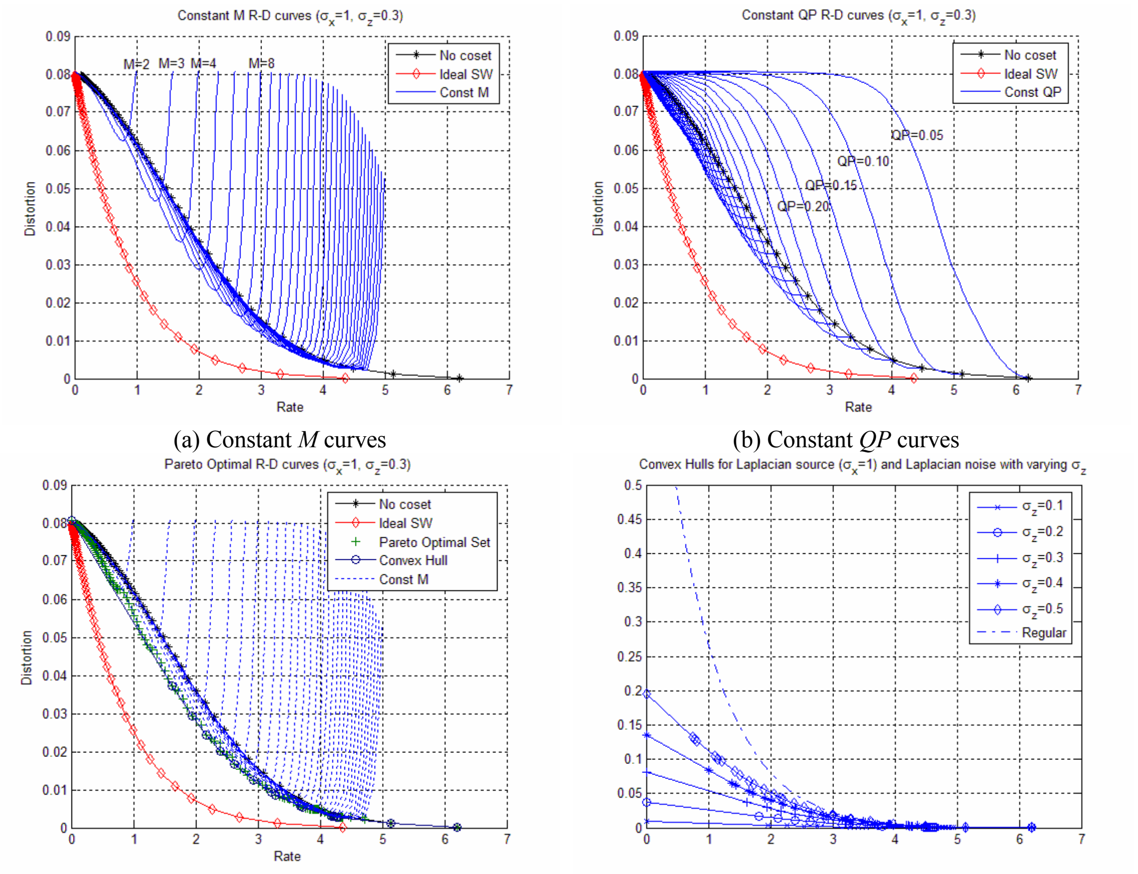
for use in $m^{(1)}_{X/Y}(x, y)$ in Eq. 38, and also for optimal reconstruction and distortion functions in the zero-rate case.

2.4. R-D curves for deadzone quantizer and optimal parameter choice

We next present the R-D curves for a deadzone quantizer given by Eq. 2 and the coset modulus function given by Eq. 5, obtained by changing the parameters *QP* and *M* for the Laplacian-Gaussian model. Note that while *M* is always discrete, *QP* is in general continuous. However we have sampled it at regular intervals in the R-D curves presented below, and the sampling interval can be arbitrarily small to approximate the continuous case. On the other hand, for most real codecs, the *QP* is indeed discrete.

2.4.1. Constant M and Constant QP R-D curves

Figure 3(a) and (b) shows two ways of presenting the curves for the specific case of Laplacian *X* (σ_X=1), and Gaussian *Z* (σ_Z=0.5), while Figure 4(a) and (b) shows the corresponding results for the same Laplacian *X* (σ_X=1) and Laplacian Z (σ_Z=0.3). In Figure 3(a) and Figure 4(a) each R-D curve is generated by fixing *M* and changing *QP* at finely sampled intervals of 0.05 between 0.05 and 3.15. The following discussion assumes *QP* to be continuous. The case *QP*→∞ for any *M* corresponds to the zero-rate case, and yields the R-D point $\{0, E(D_Y)\}$ where all the curves start, with $E(D_Y)$ given by Eq. 26. Alternatively, this point can also be viewed as the *M*=1 curve which degenerates to a point. The other extreme is the case where *QP*→0+. In this case, for any *M*, each coset index has equal probability and so the entropy converges to $\log_2 M$. However, the distortion then becomes the same as the zero-rate case $E(D_Y)$, since the coset indices do not provide any useful information. For the purpose of comparison, the line with '*'s correspond to the non-distributed coding case



(a) Constant *M* curves

(b) Constant *QP* curves

(c) Pareto Optimal Set and Convex Hull

(d) Convex Hulls for varying σ_Z

Figure 4. R-D curves obtained by changing *QP* and *M* for Laplacian *X* and Laplacian *Z*.

with minimum MSE reconstruction using side-information given by Eq. 22 and Eq. 21 respectively, while the line with diamonds correspond to ideal Slepian-Wolfe coding followed by minimum MSE reconstruction. Figure 3(b) and Figure 4(b) shows the same results but now using constant *QP* curves. Each curve in this figure are generated by fixing *QP* and increasing *M* starting from 1 upwards. All the curves start from the zero-rate point $\{0, E(D_Y)\}$ corresponding to $M = 1$. This point is also the $QP\rightarrow\infty$ curve that degenerates to a point. As $M\rightarrow\infty$ however, the coder becomes the same as a regular encoder not using cosets. Consequently, each constant *QP* curve ends on a point on the curve corresponding to non-distributed coding with minimum MSE reconstruction using side-information. The line with 'diamonds' correspond to the ideal Slepian Wolfe coding case followed by minimum MSE reconstruction.

2.4.2. Pareto Optimal Set and Convex Hull

From the curves it is obvious that not all choices for *QP* and *M* are necessarily better than regular coding followed by minimum MSE reconstruction using side-information. The sub-optimal choices for $\{QP, M\}$ combination can be pruned out by finding the *Pareto-Optimal* set *P*, wherein each point is such that no other point is *superior* to it, i.e. yields a lower or equal distortion at a lower or equal rate (assuming that the rate-distortion points are all distinct). These points are marked as '+' in Figure 3(c) and Figure 4(c). Now, given a target distortion D_t in terms of the quantization parameter QP_t for regular coding with no side-information using Eq. 24, one can search the Pareto Optimal set *P* for the point that yields the closest distortion to D_t, and choose that.

However, a strategy yielding superior rate-distortion performance is to operate on the *convex hull* of the set of R-D points generated by all $\{QP, M\}$ combinations. The convex hull is a piecewise linear function generated from the Pareto optimal set of points *P* by generating an ordered subset of points called the *convex hull set H* in descending order of distortion, and joining these points by straight line segments. The procedure is explained below, assuming zero-based indexing for ordered *P* and *H*:

1. Sort the points in *P* in descending order of distortion.

2. Include first (highest distortion) point of P corresponding to zero-rate in H: $H[0]=P[0]$, $n_P= 1$, $n_H= 1$
3. While $n_P <|P|$ (the total number of points in P)

   Compute the gradient to the last point included in H to other points in P with lower distortion. Choose the point that yields the steepest negative gradient, and include that point in the convex hull set:

   $$k^* = \underset{k=n_P+1,\ldots,|P|-1}{\arg\min}\ \{(D_{H[n_H-1]} - D_{P[k]})/(R_{H[n_H-1]} - R_{P[k]})\},\quad H[n_H]=P[k^*],\ \ n_H = n_H+1,\ \ n_P = k^* \tag{40}$$

   where $D_{H[i]}$ ($D_{P[i]}$) and $R_{H[i]}$ ($R_{P[i]}$) are the distortion and rate values corresponding to the ith point in the set H (P).

   End.
4. Join the resultant n_H ordered points in H by straight line segments.

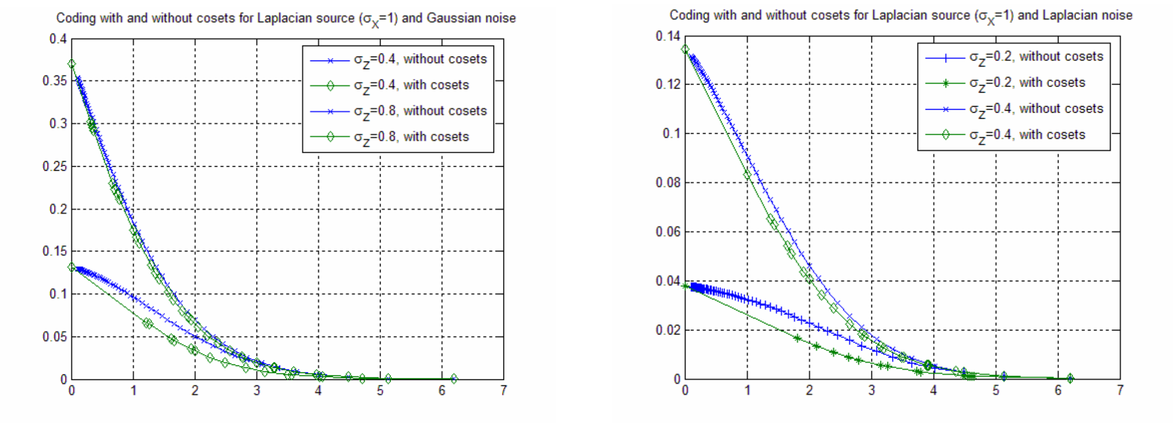
Note that when QP is indeed continuous, the convex hull actually becomes a curve which has small continuous sections lying on a constant M curve, followed by linear sections joining another constant M curve (for a different M). However, in a practical codec usually a discrete set of QP_t or QP values are allowed, and hence we stay within this set for our chosen QP.

Figure 3(c) and Figure 4(c) shows the points included in the convex hull set H as ‘o’. The convex hull is obtained by joining them with straight line segments. Note that this piecewise linear convex hull is not guaranteed to have points that are obtained with a specific $\{QP, M\}$ combination, except at the points in the convex hull set. However, the following method can be used to probabilistically operate at any intermediate point. Given a target QP_t and corresponding distortion D_t, search the decreasing distortion ordered set H to find where D_t lies. If D_t is higher than the zero-rate point distortion, i.e. $D_t > D_{H[0]}$, use zero-rate encoding. Otherwise, if D_t lies between the i^{th} and $(i+1)^{\text{th}}$ points, i.e. $D_{H[i]} \geq D_t > D_{H[i+1]}$, calculate $\alpha = (D_{H[i]}–D_t)/(D_{H[i]}– D_{H[i+1]})$; then use a uniform pseudo random number generator in the encoder to choose parameters $\{QP_{H[i]}, M_{H[i]}\}$ with probability $1–\alpha$ and $\{QP_{H[i+1]}, M_{H[i+1]}\}$ with probability α, for each sample encoded. The decoder is assumed to use a synchronized pseudorandom number generator with the same seed to obtain the right parameters for decoding each sample. Thus, all points on the convex hull are in fact achievable, and the convex hull should be chosen as the optimal operational R-D curve.

To summarize, given the statistics $\{\sigma_X, \sigma_Z\}$, each target QP_t (and consequently D_t) would map to a 5-tuple $\{QP_1, M_1, QP_2, M_2, \alpha\}$ where parameters $\{QP_1, M_1\}$ and $\{QP_2, M_2\}$ are chosen with probabilities $(1– \alpha)$ and α respectively. This mapping would typically be obtained offline for each class based on known class statistics $\{\sigma_X, \sigma_Z\}$ using training data, and stored in the form of a table in the encoder and decoder to perform the encoding and decoding accordingly. An example of such a table generated for Laplacian X (σ_X=1), and Gaussian Z (σ_Z=0.5) is shown in Table 1, where the QP are taken from a discrete set of values sampled at intervals of 0.05. Here all entries with $QP = \infty$, M=1 correspond to zero rate. Any entry with $M = \infty$ correspond to coding without cosets but using side-information based minimum MSE reconstruction. Note that as the target QP_t increases it becomes optimal to just use zero-rate encoding.

Table 1. Look-up table from target QP_t to 5-tuple parameters for Laplacian σ_X=1, Gaussian σ_Z=0.5

QP_t	QP_1	M_1	QP_2	M_2	α
0.05	0.05	∞	0.05	∞	0.00000
0.10	0.15	27	0.10	∞	0.99557
0.15	0.15	27	0.10	∞	0.02017
0.20	0.20	19	0.15	25	0.01849
0.25	0.25	14	0.20	18	0.05705
0.30	0.30	11	0.25	14	0.06097
0.35	0.35	9	0.30	11	0.02897
0.40	0.45	7	0.40	8	0.73945
0.45	0.50	6	0.45	7	0.55446
0.50	0.60	5	0.50	6	0.63447
0.55	0.70	4	0.60	5	0.89844
0.60	0.70	4	0.60	5	0.21118
0.65	0.90	3	0.70	4	0.68756
0.70	0.90	3	0.70	4	0.21994
0.75	∞	1	0.90	3	0.91249
0.80	∞	1	0.90	3	0.74701
0.85	∞	1	0.90	3	0.57388



(a) Gaussian Noise

(b) Laplacian Noise

Figure 5. Comparing memoryless coding with and without cosets

0.90	∞	1	0.90	3	0.39387
0.95	∞	1	0.90	3	0.20779
1.00	∞	1	0.90	3	0.01642
1.05	∞	1	∞	1	0.00000
1.10	∞	1	∞	1	0.00000
1.15	∞	1	∞	1	0.00000
1.20	∞	1	∞	1	0.00000

Figure 3(d) and Figure 4(d) shows the convex hulls obtained using the above procedure for differing values of σ_Z while fixing $\sigma_X = 1$. As expected, the curve shifts up with increasing σ_Z. The figure also includes the R-D curve for regular non-distributed coding using minimum MSE reconstruction *without* side-information, generated by varying QP_t with $\sigma_X = 1$ (Eq. 22 and Eq. 24). The corresponding distortion D_t on this curve for each QP_t is to be matched to the convex hulls for the given statistics. Note for smaller values of σ_Z, a significant amount of the distortion range is covered simply by using zero-rate encoding with side-information based decoding.

For both Gaussian and Laplacian Z, it can be observed that as σ_Z increases, the convex hull comes closer to the type of coding where no cosets are used but only minimum MSE reconstruction is conducted within the unambiguously conveyed quantization bin. However, for the Laplacian Z case, the difference becomes insignificant at smaller values of σ_Z than in the Gaussian Z case. This result is not surprising intuitively speaking, given the fatter tail of the Laplacian distribution that makes coset based coding less efficient at smaller values of σ_Z. Figure 5 illustrates the situation for two different values of σ_Z for both Gaussian and Laplacian noise cases.

Figure 6 shows how the R-D characteristics change with the shape of the *Z* distribution (Laplacian vs. Gaussian), for

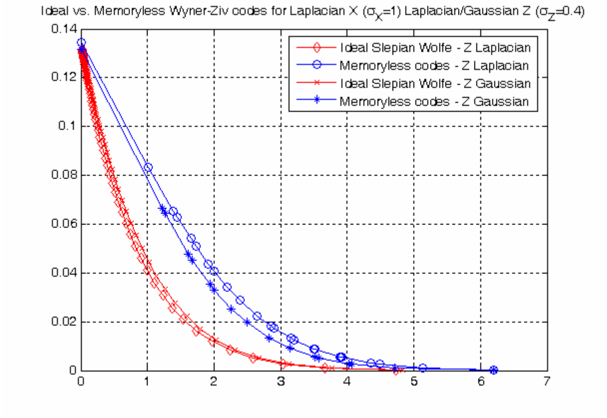


Figure 6. Comparing R-D curves for memoryless codes for Laplacian and Gaussian Z

the same Laplacian X with the same variance $\sigma_X = 1$ and the same variance of Z $\sigma_Z = 0.4$. We observe that the ideal Slepian-Wolfe bound gets worse from Laplacian to Gaussian, but the efficiency of memoryless codes get better. In other words, as the tail of the distribution gets fatter, memoryless codes get increasingly less efficient, which is not surprising.

One final point is that in some coding scenarios such as video coding for reversed complexity, the statistics (σ_X, σ_Z) may itself depend on the target QP_t for the non-Wyner-Ziv frames. In that case, each line in parameter table as in Table 1 stored in the encoder and decoder could be obtained by a different statistics (σ_X, σ_Z). The decoder additionally needs to store the actual (σ_X, σ_Z) pair for each QP_t for appropriate decoding.

2.4.3. Optimal parameter choice for a set of variables with different variances and correlation statistics

We next address the problem of optimal parameter choice for a set of N random variables: $X_0, X_1, ..., X_{N-1}$, where X_i is assumed to have variance $\sigma^2_{X_i}$ and the corresponding side information Y_i is obtained by: $Y_i = X_i + Z_i$, where Z_i is i.i.d. additive Gaussian with variance $\sigma^2_{Z_i}$. This is exactly the situation that would arise in a typical (orthogonal) transform coding scenario, where each frequency can be modeled to have different statistics. The expected distortion is then the average (sum/N) of the distortions for each X_i and the expected rate is the sum of the rates for each X_i. In order to make the optimal parameter choice, first the individual convex hull R-D curves must be generated for each i. Using typical Lagrangian optimization techniques, the optimal solution for a given total rate or distortion target should be such that points from the individual convex hull R-D curves are chosen to have the same local slope λ. The exact value of λ should be searched by bisection search or a similar method to yield the exact distortion target or the rate target. Note that since the convex hulls are piecewise linear, the slopes are decreasing piecewise constants in most parts. Therefore, interpolation of the slopes is necessary under the assumption that the virtual slope function holds its value as the true slope of a straight segment only at its mid-point.

3. APPROACHING THE SLEPIAN-WOLF BOUND

In Section 2.2.2 we have seen that the ideal rate required to transmit the quantization indices Q is no larger than H(Q/Y), as given by Eq. 19. In this section we will consider practical approaches to getting close to this bound, under the assumption of our Laplacian source and additive Gaussian noise correlation model with known parameters $\{\sigma_X^2, \sigma_Z^2\}$. Further, we will show how proper rate-allocation and soft decoding should be performed within this context.

Since the goal in this family of approaches is to achieve essentially noise-free transmission of the quantization indices with minimum possible rate, the expected distortion can be computed based on minimum MSE reconstruction within the quantization bin given by Eq. 21. Thus, for a given target distortion D_t corresponding to quantization parameter QP_t for regular coding using Eq. 24, the quantization parameter QP chosen for coding with side-information should be such that the expected distortion by Eq. 21 is equal to D_t. If the target distortion is more than the zero rate distortion given by Eq. 26, then just zero-rate coding should be used.

Table 2 shows successive values of QP_t and corresponding values for D_t and QP, and actual distortion D, which is the better of Slepian Wolfe coding or zero-rate coding, Laplacian X (σ_X=1) and Gaussian Z (σ_Z=0.5). Two scenarios – continuous and discrete QP – are considered. Continuous values correspond to the case where an arbitrary precision for QP is allowed up to a maximum of 3.15. The discrete values correspond to the case where QP is restricted to the discrete set of values from 0.05 to 3.15 at intervals of 0.05. Note that both scenarios also include zero-rate coding corresponding to QP=∞, which is chosen when the target distortion D_t is less than the zero-rate distortion. Also, note that in the continuous case at QP_t=1.0, D_t is less than the zero-rate distortion but more than the distortion for the largest possible QP (in this case 3.15). Consequently this is value is chosen as QP. In the discrete case, in order to satisfy the practical constraint that all QP values must be taken from a discrete set, the continuous QP values should be moved down to the nearest allowable discrete point such that the distortion D<D_t. In this case, however, the achieved distortion will be less than D_t.

Table 2. Mapping from QP_t to QP for ideal Slepian Wolfe Coding, along with target D_t, and actual distortion D for the case Laplacian σ_X=1, Gaussian σ_Z=0.5. Both continuous QP and discrete QP cases are considered.

***QP*$_t$**	***D*$_t$**	***QP* (continuous)**	***D* (continuous)**	***QP* (discrete)**	***D* (discrete)**
0.05	0.00025	0.05002	0.00025	0.05	0.00025
0.10	0.00115	0.10023	0.00115	0.10	0.00114
0.15	0.00287	0.15095	0.00287	0.15	0.00282
0.20	0.00556	0.20257	0.00556	0.20	0.00538
0.25	0.00930	0.25553	0.00930	0.25	0.00882
0.30	0.01415	0.31034	0.01415	0.30	0.01311
0.35	0.02016	0.36758	0.02016	0.35	0.01816

0.40	0.02731	0.42790	0.02731	0.40	0.02385
0.45	0.03562	0.49208	0.03562	0.45	0.03006
0.50	0.04503	0.56072	0.04503	0.55	0.04353
0.55	0.05553	0.63525	0.05553	0.60	0.05054
0.60	0.06704	0.71737	0.06704	0.70	0.06464
0.65	0.07953	0.80919	0.07953	0.80	0.07832
0.70	0.09292	0.91409	0.09292	0.90	0.09121
0.75	0.10715	1.03678	0.10715	1.00	0.10308
0.80	0.12214	1.18524	0.12214	1.15	0.11883
0.85	0.13783	1.37415	0.13783	1.35	0.13605
0.90	0.15413	1.63595	0.15413	1.60	0.15224
0.95	0.17099	2.07343	0.17099	2.05	0.17035
1.00	0.18833	3.15000	0.18586	3.15	0.18586
1.05	0.20608	∞	0.18982	∞	0.18982
1.10	0.22418	∞	0.18982	∞	0.18982
1.15	0.24255	∞	0.18982	∞	0.18982
1.20	0.26115	∞	0.18982	∞	0.18982

The rest of this Section deals only with the range of QP_t values for which the target distortion D_t is less than the zero-rate distortion, and therefore a non-zero rate needs to be transmitted to achieve at least equivalent distortion. The quantization parameter used must be QP, which is larger than QP_t.

3.1. Symbol-plane by symbol-plane coding

The approach that has been typically considered in prior art is the use of bit-plane by bit-plane channel coding using powerful systematic channel codes that span long sample sequences, for instance, Turbo, Low-delay Parity Check (LDPC) codes and Repeat-Accumulate (RA) codes. Specifically, the quantization index Q using quantization parameter QP for each sample is binarized up to a certain number of bit-planes. The binarized Q values of a typically long sequence of samples are stacked up, and for each bit-plane a systematic channel code of a certain rate is used to yield a set of parity bits that are transmitted in the bit-stream. The systematic bits are not sent, and left to be recovered from the side-information at the decoder, in conjunction with the parity bits. The rate allocation and corresponding decoding for each bit-plane should not only consider the source and correlation model, but also the order in which the bit-planes are to be decoded at the decoder – a consideration that is often ignored in many current systems.

3.1.1. Symbol Decomposition

Before discussing this in further detail, we consider a somewhat more generic version of this bit-plane coding approach by allowing decomposition of Q into an arbitrary number of symbols each with an arbitrary alphabet size. We consider decomposing Q into S symbols $\{Q_0, Q_1, \ldots, Q_{S-1}\}$ each associated with a finite alphabet, and one symbol X_S associated with an infinite alphabet. Here Q_i, i=0,1,…,S–1 is the (i+1)th least significant symbol (i.e. Q_0 is the least significant symbol, Q_1 is the second least significant symbol, and so on) associated with a finite l_i-ary alphabet, while X_S is the most significant symbol associated with an infinte alphabet which is the set of all integers. The following recursion may be used to obtain the symbols Q_i and X_S from Q, given the S-ary alphabet-size vector $L=\{l_0, l_1, \ldots, l_{S-1}\}$:

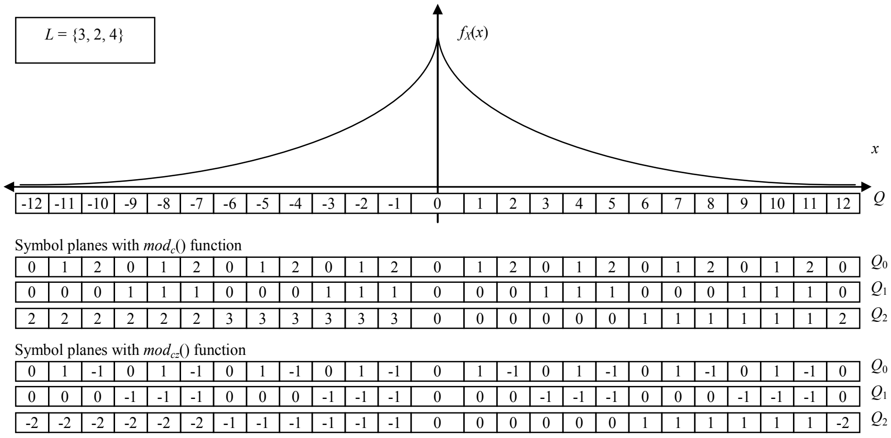
$$\begin{aligned} &Initialize: X_0 = Q \\ &Compute: Q_i = \mathrm{mod}_c(X_i, l_i),\ X_{i+1} = \lfloor X_i / l_i \rfloor, \text{for } i = 0,1,...,S-1 \end{aligned} \tag{41}$$

In this case, $Q_i \in \Omega_{Q_i} = \{0,1,..,l_i - 1\}$ for $i = 0,1,...,S-1$, and $X_S \in \{-\infty,...-1,0,1,...,\infty\}$. A variant that is identical in terms of entropy but may be preferred when an existing entropy coder for regular coding is reused for coding the symbols, uses the zero-centered circular modulus function:

$$\begin{aligned} &Initialize: X_0 = Q \\ &Compute: Q_i = \mathrm{mod}_{cz}(X_i, l_i),\ X_{i+1} = \lfloor X_i / l_i \rfloor, \text{for } i = 0,1,...,S-1 \end{aligned} \tag{42}$$

where $Q_i \in \Omega_{Q_i} = \{\lfloor -(l_i - 1)/2 \rfloor, ..., -1, 0, 1, ..., \lfloor (l_i - 1)/2 \rfloor\}$ for $i = 0,1,...,S-1$ and $X_S \in \{-\infty,...-1,0,1,...,\infty\}$ as before. Note that the (S+1)-tuple $\{Q_0, Q_1, \ldots, Q_{S-1}, X_S\}$ carries exactly the same information as Q. On the other hand, there is always some information loss, when the finite S-tuple $\{Q_0, Q_1, \ldots, Q_{S-1}\}$ is used to represent Q rather than $\{Q_0, Q_1, \ldots, Q_{S-1}, X_S\}$, due to the fact that multiple quantization bins map to the same S-tuple. In practice however, if $q_{\max}$ is the maximum magnitude quantization index Q beyond which the probabilities of the bins are trivial, and the following is satisfied:

$$\prod l_i \geq 2q_{\max} + 1, \tag{43}$$



-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	Q

Symbol planes with $mod_c()$ function

0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	Q_0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	Q_1
2	2	2	2	2	2	3	3	3	3	3	3	0	0	0	0	0	0	1	1	1	1	1	1	2	Q_2

Symbol planes with $mod_{cz}()$ function

0	1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1	0	Q_0
0	0	0	-1	-1	-1	0	0	0	-1	-1	-1	0	0	0	-1	-1	-1	0	0	0	-1	-1	-1	0	Q_1
-2	-2	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	1	1	1	1	1	1	-2	Q_2

Figure 7. Illustrating symbol-plane decomposition. The example assumes L={3,2,4}.

then the conditional entropy $H(X_S / Q_0, Q_1, ..., Q_{S-1}) \approx 0$, implying that there is no information in X_S given the S-tuple {Q_0, Q_1, …, Q_{S-1}}. In other words, $H(Q) \approx H(Q_0, Q_1, ..., Q_{S-1})$.

The strategy to be followed for transmitting Q, is to assume an S-ary finite decomposition of Q into {Q_0, Q_1, …, Q_{S-1}} given an alphabet size vector L={l_0, l_1, …, l_{S-1}}, and ignore symbol X_S. Correspondingly, Q_{S-1} is regarded as the most significant symbol rather than X_S. Thereafter, the symbols {Q_0, Q_1, …, Q_{S-1}} are transmitted but no rate is ever transmitted for X_S. The possibility of $H(X_S / Q_0, Q_1, ..., Q_{S-1}) > 0$, i.e. multiple quantization bins with non-trivial probability mapping to the same S-tuple, is however taken into account during the minimum MSE reconstruction procedure. Figure 7 illustrates the symbol-plane decomposition mechanism. Henceforth, for simplicity we will use the following notation:

$$Q_i = \xi_i^L(Q) \tag{44}$$

to denote the mapping function from Q to the ith symbol Q_i, given the alphabet-size vector L={l_0, l_1, …, l_{S-1}}.

3.1.2. Ideal rate allocation for ordered symbol coding

In the ideal case, we assume a symbol decomposition mechanism where Eq. 43 is satisfied. Since the information in Q is identical to that in {Q_0, Q_1, …, Q_{S-1}} under this assumption, i.e. $H(X_S / Q_0, Q_1, ..., Q_{S-1}) \approx 0$, the ideal Slepian-Wolfe rate can be approximated as: $H(Q/Y) \approx H(Q_0, Q_1, ..., Q_{S-1} / Y)$. If the coding for the individual symbols are conducted from least to most significant symbols, then we could decompose as follows:

$$\begin{aligned} H(Q/Y) &\approx H(Q_0, Q_1, ..., Q_{S-1} / Y) \\ &= H(Q_0 / Y) + H(Q_1 / Q_0, Y) + H(Q_2 / Q_0, Q_1, Y) + ... + H(Q_{S-1} / Q_0, Q_1, ..., Q_{S-2}, Y) \end{aligned} \tag{45}$$

Here each term corresponds to the ideal rate to be allocated for noiseless transmission of each symbol. However, to be able to achieve the rate needed for each symbol, the decoding of the symbols should be conducted also in the same order – from the least to the most significant symbol; and furthermore, decoding each symbol should be based not only on the side information Y, but also on prior decoded symbols. Likewise, if the coding order of the symbols is from the most to the least significant symbol, we would decompose as follows:

$$\begin{aligned} H(Q/Y) &\approx H(Q_0, Q_1, ..., Q_{S-1} / Y) \\ &= H(Q_{S-1} / Y) + H(Q_{S-2} / Q_{S-1}, Y) + H(Q_{S-3} / Q_{S-1}, Q_{S-2}, Y) + ... + H(Q_0 / Q_{S-1}, Q_{S-2}, ..., Q_1, Y) \end{aligned} \tag{46}$$

In general, coding of the symbols can be conducted in any order, but for each order the rate allocation per symbol would differ, and so also the decoding.

In order to exactly compute the rate allocation for a symbol i, given a subset of symbols already transmitted, we would need to compute in general the conditional entropy $H(Q_i / \{Q_k : k \in G_i\}, Y)$, where G_i is the set of indices corresponding to symbols that are to be transmitted prior to symbol Q_i. For instance, if the coding order is from the least significant symbol to the most significant symbol, we will have: $G_0 = \{\}, G_1 = \{0\}, G_2 = \{0,1\}, ..., G_{S-1} = \{0,1,...,S-2\}$.

This conditional entropy can be written as:

$$\begin{aligned}
H(Q_i/\{Q_k:k\in G_i\},Y) &= \int_{-\infty}^{\infty}\left(\sum_{\substack{q_k\in\Omega_{Qk}\\ \forall k\in G_i}}\left[\sum_{q_i\in\Omega_{Q_i}} p(Q_i=q_i/\{Q_k=q_k:k\in G_i\},Y=y)\log_2\frac{1}{p(Q_i=q_i/\{Q_k=q_k:k\in G_i\},Y=y)}\right] \times p(\{Q_k=q_k:k\in G_i\}/Y=y)\right)f_Y(y)dy \\
&= \int_{-\infty}^{\infty}\left(\sum_{\substack{q_k\in\Omega_{Qk}\\ \forall k\in G_i}}\left[\sum_{q_i\in\Omega_{Q_i}}\frac{p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)}{p(\{Q_k=q_k:k\in G_i\}/Y=y)}\log_2\frac{p(\{Q_k=q_k:k\in G_i\}/Y=y)}{p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)}\right] \times p(\{Q_k=q_k:k\in G_i\}/Y=y)\right)f_Y(y)dy \\
&= \int_{-\infty}^{\infty}\left(\sum_{\substack{q_k\in\Omega_{Qk}\\ \forall k\in G_i}}\left[\sum_{q_i\in\Omega_{Q_i}} p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)\log_2\frac{p(\{Q_k=q_k:k\in G_i\}/Y=y)}{p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)}\right]\right)f_Y(y)dy \\
&= \int_{-\infty}^{\infty}\left(\sum_{\substack{q_k\in\Omega_{Qk}\\ \forall k\in G_i\cup\{i\}}}\left[p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)\log_2\frac{p(\{Q_k=q_k:k\in G_i\}/Y=y)}{p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)}\right]\right)f_Y(y)dy
\end{aligned} \tag{47}$$

Noting that the conditional probability can be expressed as:

$$p(\{Q_k=q_k:k\in G_i\}/Y=y)=\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\pi(q,y)=\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx=\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\left[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)\right] \tag{48}$$

the conditional entropy is given by:

$$\begin{aligned}
&H(Q_i/\{Q_k:k\in G_i\},Y) \\
&= \int_{-\infty}^{\infty}\left(\sum_{\substack{q_k\in\Omega_{Qk}\\ \forall k\in G_i\cup\{i\}}}\left[\left(\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\pi(q,y)\right)\log_2\frac{\left(\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\pi(q,y)\right)}{\left(\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\pi(q,y)\right)}\right]\right)f_Y(y)dy \\
&= \int_{-\infty}^{\infty}\left(\sum_{\substack{q_k\in\Omega_{Qk}\\ \forall k\in G_i\cup\{i\}}}\left[\left(\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\left[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)\right]\right)\log_2\frac{\left(\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\left[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)\right]\right)}{\left(\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\left[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)\right]\right)}\right]\right)f_Y(y)dy
\end{aligned} \tag{49}$$

This entropy can be readily calculated based on the expressions for the partial moments presented earlier, in conjunction with numerical integration over y. Note that even though the expressions look formidable, they are fairly straight-forward to compute.

We present examples of the ideal bit-plane by bit-plane rate allocation for the case σ_X=1 (Laplacian), σ_Z=0.5 (Gaussian), for various coding orders orders below. We consider 11 bit-planes, so that the condition of Eq. 43 is easily satisfied. Table 3 shows the allocation for LSB to MSB coding. Table 4 shows the allocation for MSB to LSB coding, while Table 5 shows the allocation for an arbitrary coding order (first MSB followed by LSB, second LSB, and so on). In each table, the columns are ordered according to the order of coding of the bit-planes. The tables also provide the total conditional entropy or the ideal Slepian-Wolfe rate in the ***Sum*** column, which is the sum of the rates for the individual bit-planes. Note that this value across different tables for the same *QP* is the same, irrespective of the coding order.

Table 3. Ideal rate in bits for LSB to MSB coding for various *QP* values, for σ_X=1 (Laplacian), σ_Z=0.5 (Gaussian).

QP	***BP 0***	***BP 1***	***BP 2***	***BP 3***	***BP 4***	***BP 5***	***BP 6***	***BP 7***	***BP 8***	***BP 9***	***BP 10***	***Sum***
0.1	0.99478	0.98908	0.97379	0.83909	0.22108	0.00135	0.00000	0.00000	0.00000	0.00000	0.00000	4.01917
0.2	0.97904	0.95173	0.79178	0.19708	0.00112	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	2.92076
0.3	0.95249	0.85900	0.42585	0.01856	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	2.26589
0.4	0.91460	0.72130	0.17029	0.00097	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.80718
0.5	0.86475	0.54184	0.05772	0.00003	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.46434
0.6	0.80324	0.37860	0.01758	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.19942

0.7	0.73266	0.25288	0.00495	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.99048
0.8	0.65743	0.16459	0.00130	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.82332
0.9	0.58216	0.10563	0.00033	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.68814
1.0	0.51042	0.06730	0.00008	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.57779

Table 4. Ideal rate in bits for MSB to LSB coding for various *QP* values, for σ_X=1 (Laplacian), σ_Z=0.5 (Gaussian).

QP	***BP 10***	***BP 9***	***BP 8***	***BP 7***	***BP 6***	***BP 5***	***BP 4***	***BP 3***	***BP 2***	***BP 1***	***BP 0***	***Sum***
0.1	0.46158	0.00000	0.00000	0.00000	0.00022	0.02023	0.19565	0.56927	0.84519	0.95069	0.97625	4.01917
0.2	0.35694	0.00000	0.00000	0.00000	0.00000	0.00022	0.02033	0.19590	0.57018	0.83897	0.93823	2.92076
0.3	0.26471	0.00000	0.00000	0.00000	0.00000	0.00000	0.00212	0.06306	0.34260	0.70554	0.88787	2.26589
0.4	0.18882	0.00000	0.00000	0.00000	0.00000	0.00000	0.00022	0.02033	0.19611	0.57150	0.83018	1.80718
0.5	0.12976	0.00000	0.00000	0.00000	0.00000	0.00000	0.00002	0.00656	0.11119	0.44789	0.76892	1.46434
0.6	0.08596	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00212	0.06306	0.34335	0.70493	1.19942
0.7	0.05492	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00068	0.03579	0.26018	0.63890	0.99048
0.8	0.03384	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00022	0.02033	0.19619	0.57274	0.82332
0.9	0.02011	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00007	0.01154	0.14772	0.50868	0.68814
1.0	0.01152	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00002	0.00656	0.11120	0.44850	0.57779

Table 5. Ideal rate in bits for arbitrary coding order for various *QP v*alues, for σ_X=1 (Laplacian), σ_Z=0.5 (Gaussian).

QP	***BP 10***	***BP 0***	***BP 1***	***BP 2***	***BP 3***	***BP 4***	***BP 5***	***BP 6***	***BP 7***	***BP 8***	***BP 9***	***Sum***
0.1	0.46158	0.98295	0.95823	0.88148	0.61574	0.11867	0.00052	0.00000	0.00000	0.00000	0.00000	4.01917
0.2	0.35694	0.94740	0.87388	0.62082	0.12117	0.00054	0.00000	0.00000	0.00000	0.00000	0.00000	2.92076
0.3	0.26471	0.90629	0.76701	0.31648	0.01141	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	2.26589
0.4	0.18882	0.86409	0.62661	0.12699	0.00065	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.80718
0.5	0.12976	0.81904	0.47112	0.04439	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.46434
0.6	0.08596	0.76696	0.33234	0.01416	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	1.19942
0.7	0.05492	0.70641	0.22495	0.00420	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.99048
0.8	0.03384	0.63972	0.14860	0.00116	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.82332
0.9	0.02011	0.57087	0.09684	0.00030	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.68814
1.0	0.01152	0.50365	0.06264	0.00007	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.57779

We next present similar results for symbol-based coding, assuming only 4 symbols, with the alphabet-size vector being given by {3, 2, 4, 100}. Note that l_3=100 is chosen to be large enough so that the assumption of Eq. 43 holds. The source and correlation model is given by σ_X=1 (Laplacian), σ_Z=0.5 (Gaussian). Table 6 shows the ideal rate for LSS (least significant symbol) to MSS (most significant symbol) coding. Table 7 shows the ideal rate for MSS to LSS coding. Table 8 shows the ideal rates for an arbitrary order. In all cases, the ***Sum*** column shows the ideal Slepian-Wolfe rate, which is the sum of the rates for the corresponding row.

Table 6. Ideal rate in bits for LSS to MSS coding with 4 symbols and *L*={3, 2, 4, 100}, for various *QP* values, for σ_X=1(Laplacian), σ_Z=0.5 (Gaussian).

QP	***SP 0***	***SP 1***	***SP 2***	***SP 3***	***Sum***
0.1	1.57438	0.98245	1.43832	0.02402	4.01917
0.2	1.54134	0.90082	0.47860	0.00000	2.92076
0.3	1.48154	0.67519	0.10917	0.00000	2.26589
0.4	1.38654	0.40307	0.01755	0.00000	1.80716
0.5	1.25431	0.20790	0.00213	0.00000	1.46434
0.6	1.10054	0.09868	0.00021	0.00000	1.19942
0.7	0.94590	0.04457	0.00002	0.00000	0.99048
0.8	0.80385	0.01947	0.00000	0.00000	0.82332
0.9	0.67982	0.00829	0.00000	0.00000	0.68814
1.0	0.57433	0.00346	0.00000	0.00000	0.57779

Table 7. Ideal rate in bits for MSS to LSS coding with 4 symbols and *L*={3, 2, 4, 100}, for various *QP* values, for σ_X=1(Laplacian), σ_Z=0.5 (Gaussian).

QP	***SP 3***	***SP 2***	***SP 1***	***SP 0***	***Sum***
0.1	0.52676	1.04918	0.90455	1.53869	4.01917
0.2	0.35906	0.40504	0.70661	1.45005	2.92076
0.3	0.26478	0.15920	0.50676	1.33516	2.26589
0.4	0.18882	0.06518	0.34298	1.21018	1.80716
0.5	0.12976	0.02736	0.22595	1.08127	1.46434
0.6	0.08596	0.01161	0.14771	0.95414	1.19942

0.7	0.05492	0.00495	0.09649	0.83412	0.99048
0.8	0.03384	0.00212	0.06306	0.72430	0.82332
0.9	0.02011	0.00091	0.04124	0.62587	0.68814
1.0	0.01152	0.00039	0.02697	0.53891	0.57779

Table 8. Ideal rate in bits for MSS to LSS coding with 4 symbols and *L*={3, 2, 4, 100}, for various *QP* values, for σ_X=1(Laplacian), σ_Z=0.5 (Gaussian).

QP	*SP 3*	*SP 0*	*SP 1*	*SP 2*	*Sum*
0.1	0.52676	1.54750	0.92144	1.02347	4.01917
0.2	0.35906	1.47325	0.76677	0.32168	2.92076
0.3	0.26478	1.38561	0.54323	0.07228	2.26589
0.4	0.18882	1.28536	0.32088	0.01210	1.80716
0.5	0.12976	1.16532	0.16767	0.00159	1.46434
0.6	0.08596	1.03172	0.08157	0.00017	1.19942
0.7	0.05492	0.89755	0.03799	0.00001	0.99048
0.8	0.03384	0.77232	0.01715	0.00000	0.82332
0.9	0.02011	0.66047	0.00754	0.00000	0.68814
1.0	0.01152	0.56304	0.00323	0.00000	0.57779

3.1.3. Practical rate allocation and coding

While the conditional entropy results have been presented for arbitrary symbol decomposition, in a practical scenario, it would be convenient to choose alphabet-sizes for each symbol to be 2, or at most small powers of 2. The case where each l_i = 2 corresponds to the popular bit-plane by bit-plane coding case, where extensive prior knowledge on behavior and performance of binary error-correction codes can be brought to bear.

Coding of each symbol plane in the pre-determined order is conducted by use of a systematic channel code, where only the parity information is transmitted. The amount of parity information sent should be at least as much as the conditional entropy given by Eq. (47) and Eq. (49), in order to ensure noise-free decoding. However, since noise-free transmission is achievable only for very large block lengths, it is necessary to add a margin to the computed ideal rate. The margin may depend on the expected length of a block specific to a given application, the complexity of the code, as well as the impact of an error in decoding a symbol to the overall distortion. The margin can be a multiplicative factor, denoted γ_i for the symbol Q_i, of the ideal rate. The rate allocated for channel coding r_i^{CC} (CC stands for channel coding) is then given by:

$$r_i^{CC} = (\gamma_i + 1)H(Q_i / \{Q_k : k \in G_i\}, Y) \tag{50}$$

where $\gamma_i > 0$.

We next consider the rate needed to transmit a symbol plane noise-free with only source coding (i.e. no channel coding) conditioned on previously transmitted symbol planes. This rate denoted r_i^{SC} (SC stands for source coding) is given by the conditional entropy $H(Q_i / \{Q_k : k \in G_i\})$ as follows:

$$\begin{aligned} r_i^{SC} = H(Q_i / \{Q_k : k \in G_i\}) &= \sum_{\substack{q_k \in \Omega_{Q_k} \\ \forall k \in G_i \cup \{i\}}} \left[\left(\sum_{\substack{q \in \Omega_Q : \\ \xi_k^L(q) = q_k \forall k \in G_i \cup \{i\}}} \pi(q,y) \right) \log_2 \frac{\left(\sum_{\substack{q \in \Omega_Q : \\ \xi_k^L(q) = q_k \forall k \in G_i}} \pi(q,y) \right)}{\left(\sum_{\substack{q \in \Omega_Q : \\ \xi_k^L(q) = q_k \forall k \in G_i \cup \{i\}}} \pi(q,y) \right)} \right] \\ &= \sum_{\substack{q_k \in \Omega_{Q_k} \\ \forall k \in G_i \cup \{i\}}} \left[\left(\sum_{\substack{q \in \Omega_Q : \\ \xi_k^L(q) = q_k \forall k \in G_i \cup \{i\}}} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right) \log_2 \frac{\left(\sum_{\substack{q \in \Omega_Q : \\ \xi_k^L(q) = q_k \forall k \in G_i}} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right)}{\left(\sum_{\substack{q \in \Omega_Q : \\ \xi_k^L(q) = q_k \forall k \in G_i \cup \{i\}}} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right)} \right] \end{aligned} \tag{51}$$

This rate can be practically achieved by context-adaptive entropy (for instance, arithmetic) coding.

Even though $H(Q_i / \{Q_k : k \in G_i\}, Y) \le H(Q_i / \{Q_k : k \in G_i\})$, the margin requirement for the practical channel coding case, may make it possible that $r_i^{SC} \le r_i^{CC}$. In this case, just source coding should be used instead of channel coding.

The overall symbol plane by symbol plane coding strategy for a given order can now be outlined as follows:

1. Fix encoding order for symbol planes.
2. While not all symbol planes are done,
    a. Get next symbol plane Q_i in pre-determined order.
    b. Obtain $r_i^{CC} = (\gamma_i + 1)H(Q_i / \{Q_k : k \in G_i\}, Y)$ and $r_i^{SC} = H(Q_i / \{Q_k : k \in G_i\})$ for symbol plane Q_i, given previously transmitted symbol planes, by computing or reading/interpolating from pre-computed tables of conditional entropies, and adding pre-determined margins.
    c. If $r_i^{SC} \le r_i^{CC}$, use source coding with conditional entropy coding, using the previously transmitted symbol planes as context,
    Else use channel coding with rate r_i^{CC}.
3. Done Encoding

There is one caveat in the use of conditional source coding for symbol planes other than the first. In order to enable correct decoding of a source coded symbol plane, it must be assumed that the channel coded symbol planes transmitted prior to this plane have been decoded noise-free. While this can be ensured by having big enough margins, a more robust alternative would be to use as context for source coding only the previously transmitted source coded planes but not the channel coded planes. In this case, the source coding rate is given by Eq. 51, where G_i represents the set of indices of previously transmitted *source* coded symbol planes, rather than the set of indices of all previously transmitted symbol planes. Naturally, this leads to loss of compression efficiency. Only for the first symbol plane transmitted, there is no difference in the two approaches. The source coding rate in this case is given by the unconditional entropy of the symbol:

$$r_i^{SC} = H(Q_i) = -\sum_{q_i \in \Omega_{Q_i}} \left(\sum_{\substack{q \in \Omega_Q: \\ \xi_i^L(q) = q_i}} \pi(q,y) \right) \log_2 \left(\sum_{\substack{q \in \Omega_Q: \\ \xi_i^L(q) = q_i}} \pi(q,y) \right) = -\sum_{q_i \in \Omega_{Q_i}} \left(\sum_{\substack{q \in \Omega_Q: \\ \xi_i^L(q) = q_i}} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right) \log_2 \left(\sum_{\substack{q \in \Omega_Q: \\ \xi_i^L(q) = q_i}} \left[m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q)) \right] \right) \qquad (52)$$

Finally, we note that since the rates required for channel coded planes are arbitrary, it is inconvenient to design different codes for every possible rate. Furthermore, in many applications, the number of samples to be transmitted is variable and not known a priori. In such cases, puncturing should be used. Only certain systematic codes at fixed rates should be designed, and the intermediate rate codes are derived from the next higher rate code by removing an appropriate number of parity bits. The total number of parity bits to be transmitted for symbol plane Q_i, is given by $N_{samples} \times r_i^{CC}$. If the number of parity bits with the next higher rate code is N_{parity}, then $N_{parity} - N_{samples} \times r_i^{CC}$ parity bits must be removed. Usually removing parity bits at regular intervals so that there are $N_{samples} \times r_i^{CC}$ bits are eventually transmitted works reasonably well.

Finally, we note that even though we assume an *i.i.d.* model in this work which makes sense for block transform coefficients, for correlated sources, the actual source coding rate can be much less than that given by Eq. 51 or Eq. 52. Sophisticated modeling is often used in source coding to reduce bit-rate even when the residual correlation is limited. On the other hand, for channel coding, the correlation between neighboring samples is much harder to exploit. While there exists a framework to exploit these correlations using decoding on graphs, such decoders can be quite complicated to implement in practice with robust enough convergence characteristics. Therefore, in the general case, instead of using Eq. 51 or Eq. 52 to estimate the source coding rate, an actual source coder may be used, and the actual rate used by it may be considered to decide whether to use source coding or channel coding. In other words, if the rate required for channel coding to reliably decode a plane is less than the rate required with an actual source coder, only then channel coding should be used.

3.1.4. Decoding strategies

For decoding, a soft input decoder must be used. Such a decoder takes in as input a priori soft probabilities of systematic and parity symbols for a block in order to perform the decoding, and outputs either a *hard-decision* about the symbols (for instance using the Viterbi algorithm) or a *soft-decision* yielding the posteriori probability mass function of each symbol (for instance using the BCJR algorithm). Both cases are discussed below.

We first focus on the *soft-input hard-output* case. In this case, the prior probabilities for the systematic symbols in any plane are obtained based on the side information y, and knowledge of previously hard-decoded symbol planes. Thus, for decoding the symbol plane Q_i, given previously decoded symbols $\{Q_k = q_k : k \in G_i\}$ and side-information Y=y, the prior probability of $Q_i = q_i \in \Omega_{Q_i}$, denoted $\{p^{(prior)}(Q_i = q_i) : q_i \in \Omega_{Q_i}\}$ would be given by:

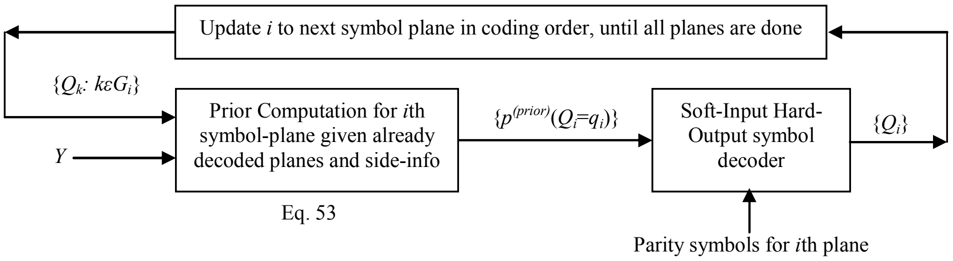

Update i to next symbol plane in coding order, until all planes are done
$\{Q_k: k\varepsilon G_i\}$
Prior Computation for ith symbol-plane given already decoded planes and side-info
Y
$\{p^{(prior)}(Q_i=q_i)\}$
Soft-Input Hard-Output symbol decoder
$\{Q_i\}$
Eq. 53
Parity symbols for ith plane

Figure 8. Decoding mechanism using Soft-Input Hard-Output symbol plane decoders

$$\begin{aligned}p^{(prior)}(Q_i=q_i)&=p(Q_i=q_i/\{Q_k=q_k:k\in G_i\},Y=y)=\frac{p(\{Q_k=q_k:k\in G_i\cup\{i\}\}/Y=y)}{p(\{Q_k=q_k:k\in G_i\}/Y=y)}\\&=\frac{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\pi(q,y)}{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\pi(q,y)}=\frac{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\left[m^{(0)}_{X/Y}(x_h(q),y)-m^{(0)}_{X/Y}(x_l(q),y)\right]}{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\left[m^{(0)}_{X/Y}(x_h(q),y)-m^{(0)}_{X/Y}(x_l(q),y)\right]}\end{aligned}\tag{53}$$

Since the parity symbols are assumed to be transmitted noise-free, their prior probabilities are taken as unity for the received symbol and zero otherwise. Figure 8 illustrates the mechanism. A drawback of this approach is that if an error has been made in decoding a symbol in one plane, the error would propagate to the rest of the symbol planes to be decoded. However, if the margin has been conservative enough, the probability of such errors should be very small.

The soft-input decoder may also make a *soft-decision* about the symbol transmitted. In this case, the decoder for each plane returns the soft posteriori probability mass functions for the decoded symbols, denoted $p^{(post)}(Q_i=q_i), q_i\in\Omega_{Q_i}$. Ability to use this soft information effectively for decoding the rest of the symbol planes, can potentially lead to better decoding performance. Assuming that soft *joint* posteriori probability mass functions of previously decoded symbol planes denoted $p^{(post)}(\{Q_k=q_k:k\in G_i\})$ are available, the prior probabilities comprising the soft input for decoding next plane Q_i, may be obtained as:

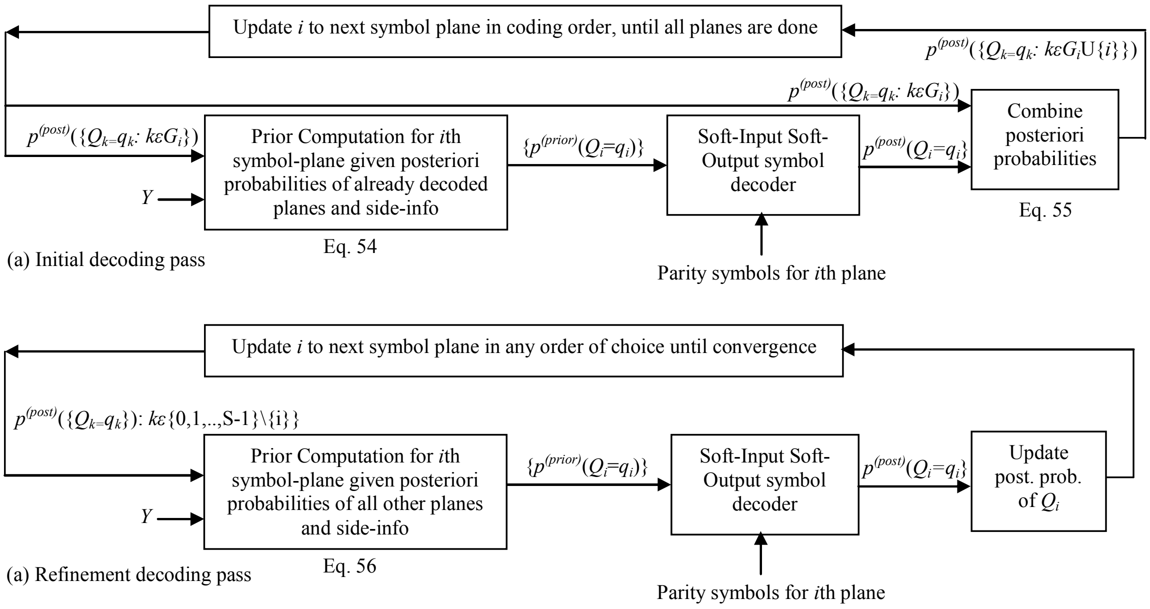
$$\begin{aligned}p^{(prior)}(Q_i=q_i)&=\sum_{\{q_k\in\Omega_{Q_k}:k\in G_i\}}p^{(post)}(\{Q_k=q_k:k\in G_i\}).p(Q_i=q_i/\{Q_k=q_k:k\in G_i\},Y=y)\\&=\sum_{\{q_k\in\Omega_{Q_k}:k\in G_i\}}p^{(post)}(\{Q_k=q_k:k\in G_i\}).\frac{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\pi(q,y)}{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\pi(q,y)}\\&=\sum_{\{q_k\in\Omega_{Q_k}:k\in G_i\}}p^{(post)}(\{Q_k=q_k:k\in G_i\}).\frac{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i\cup\{i\}}}\left[m^{(0)}_{X/Y}(x_h(q),y)-m^{(0)}_{X/Y}(x_l(q),y)\right]}{\displaystyle\sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_i}}\left[m^{(0)}_{X/Y}(x_h(q),y)-m^{(0)}_{X/Y}(x_l(q),y)\right]}\end{aligned}\tag{54}$$

Once the decoder produces the soft outputs $p^{(post)}(Q_i=q_i)$, it must be combined with the existing joint probabilities $p^{(post)}(\{Q_k=q_k:k\in G_i\})$ to obtain the updated joint probability distribution $p^{(post)}(\{Q_k=q_k:k\in G_i\cup\{i\}\})$ that includes the newly decoded symbol plane. Under the assumption of independence of the symbol planes, the joint posteriori probability distribution is simply the product of the distributions of the constituent symbol planes. The new joint distribution is then simply:

$$p^{(post)}(\{Q_k=q_k:k\in G_i\cup\{i\}\})=p^{(post)}(\{Q_k=q_k:k\in G_i\})\times p^{(post)}(Q_i=q_i)=\prod_{k\in G_i\cup\{i\}}p^{(post)}(Q_k=q_k),\quad\forall q_k\in\Omega_{Q_k},k\in G_i\cup\{i\}\tag{55}$$

This is next used to obtain the priors for decoding the next symbol plane. Once all symbol planes have been decoded, the soft posteriori probabilities for each quantization bin can be obtained, and a hard decision can be made. The approach is illustrated in Figure 9(a).

While this approach mitigates the propagation of errors from symbol plane to symbol plane, it still does not enable correcting errors that have been made in a symbol plane. In order to enable that, the following iterative decoding strategy may be used. When all the symbol planes have been decoded once in order with the above strategy, we would have obtained the posteriori probabilities of the individual symbol planes. Now, we can refine the decoding of each symbol


Update i to next symbol plane in coding order, until all planes are done
p(post)({Qk=qk: kεGiU{i}})
p(post)({Qk=qk: kεGi})
p(post)({Qk=qk: kεGi})
Prior Computation for ith symbol-plane given posteriori probabilities of already decoded planes and side-info
Y
{p(prior)(Qi=qi)}
Soft-Input Soft-Output symbol decoder
p(post)(Qi=qi}
Combine posteriori probabilities
Eq. 55
Eq. 54
Parity symbols for ith plane
Update i to next symbol plane in any order of choice until convergence
p(post)({Qk=qk}): kε{0,1,..,S-1}\{i}}
Prior Computation for ith symbol-plane given posteriori probabilities of all other planes and side-info
Y
{p(prior)(Qi=qi)}
Soft-Input Soft-Output symbol decoder
p(post)(Qi=qi}
Update post. prob. of Qi
Eq. 56
Parity symbols for ith plane

(a) Initial decoding pass

(a) Refinement decoding pass

Figure 9. Decoding mechanism using Soft-Input Soft-Output symbol plane decoders

plane in any order where the prior is assumed to be computed based on the joint distribution of all symbol planes other than the one being decoded. The joint distribution is simply the product of the individual symbols under the independence assumption.

$$\begin{aligned} p^{(prior)}(Q_i = q_i) &= \sum_{\{q_k \in \Omega_{Q_k} : k \in \{0,1,..,S-1\} \setminus \{i\}\}} p^{(post)}(\{Q_k = q_k : k \in \{0,1,..,S-1\} \setminus \{i\}) \times p(Q_i = q_i / \{Q_k = q_k : k \in \{0,1,..,S-1\} \setminus \{i\}\}, Y = y) \\ &= \sum_{\{q_k \in \Omega_{Q_k} : k \in \{0,1,..,S-1\} \setminus \{i\}\}} \left(\prod_{k \in \{0,1,..,S-1\} \setminus \{i\}} p^{(post)}(Q_k = q_k) \right) \times p(Q_i = q_i / \{Q_k = q_k : k \in \{0,1,..,S-1\} \setminus \{i\}\}, Y = y) \end{aligned} \tag{56}$$

The new decoded posteriori probabilities update the posteriori distribution of the symbol plane concerned. The process is repeated over all symbol planes, multiple times until the posteriori distributions converge. The mechanism is illustrated in Figure 9(b).Obviously, this procedure is very demanding computationally, but decoding should in general be better.

Various combinations of the above two decoding strategies can be considered. For example, the early symbol planes in encoding order may be channel coded with a big margin or source coded, to ensure virtually noise-free transmission, while the trailing ones may be channel coded with a smaller margin. In this case, the early channel coded symbol planes can be hard-decoded, while the trailing symbol planes may use soft-output based decoding.

3.1.5. Skipped symbol planes

In some cases it will be convenient to just not transmit anything for certain symbol-planes. Typically, these symbol planes are chosen as ones for which the ideal rate (conditional entropy) is very small. Furthermore, if that is indeed the case, it is better that these symbol planes be the trailing ones in coding order, so that the decoding for the other symbol planes may be conducted without any impreciseness. The modified coding strategy can now be described as:

1. Fix encoding order for symbol planes.
2. While not all symbol planes are done,
    a. Get next symbol plane Q_i in pre-determined order.
    b. Obtain $r_i^{CC} = (\gamma_i + 1)H(Q_i / \{Q_k : k \in G_i\}, Y)$ and $r_i^{SC} = H(Q_i / \{Q_k : k \in G_i\})$ for symbol plane Q_i, given previously transmitted symbol planes, by computing or reading/interpolating from pre-computed tables of conditional entropies, and adding pre-determined margins.
    c. If $H(Q_j / \{Q_k : k \in G_i\}, Y) \le \varepsilon$ (where ε is a small threshold) for all j where Q_j has not already been transmitted, go to step 3 (i.e. terminate encoding). In other words, do not transmit anything for the symbol plane Q_i, and all other succeeding symbol planes if the ideal rates for all of them are smaller than the threshold ε .

Else if $r_i^{SC} \le r_i^{CC}$, use source coding with conditional entropy coding, using the previously transmitted symbol planes as context,

Else use channel coding with rate r_i^{CC}.

3. Done Encoding

3.1.6. Optimal reconstruction for soft-output decoding

A decoder that eventually returns soft posteriori probabilities of quantization bins must be appropriately represented in obtaining the final reconstruction. Assume that the decoder obtains the soft posteriori probabilities of a set of symbol planes in index set *G*: $p^{(post)}(\{Q_k = q_k : k \in G\})\ \forall q_k \in \Omega_{Q_k}$. Note that the planes in set *G* may not include all the symbol planes, if there are skipped symbol planes. Also, if there are planes in *G* that are source coded or channel coded with a big margin and subsequently hard decoded, the corresponding marginal probability is taken as 1 for the decoded value, and 0 for the rest.

Generally speaking, we will assume a form of the a posteriori conditional distribution $f_{X/Y}^{(post)}(x,y)$ which has the same shape as the a priori distribution $f_{X/Y}(x,y)$ within each bin, but scaled appropriately to satisfy the posteriori probabilities $p^{(post)}(\{Q_k = q_k : k \in G\})\ \forall q_k \in \Omega_{Q_k}$. The minimum MSE reconstruction function is then given by:

$$\begin{aligned}\hat{X} &= E(X/Y=y, p^{(post)}(\{Q_k = q_k : k \in G\})) = \sum_{q\in\Omega_Q} \int_{x_l(q)}^{x_h(q)} x f_{X/Y}^{(post)}(x,y)dx \\ &= \sum_{q\in\Omega_Q}\left[\frac{p^{(post)}(\{Q_k = \xi_k^L(q) : k \in G\})}{\sum\limits_{\substack{q'\in\Omega_Q:\\ \xi_k^L(q')=\xi_k^L(q)\forall k\in G}} \int_{x_l(q')}^{x_h(q')} f_{X/Y}(x,y)dx}\left(\int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx\right)\right] = \sum_{q\in\Omega_Q}\left[\frac{p^{(post)}(\{Q_k = \xi_k^L(q) : k \in G\})\times\mu(q,y)}{\sum\limits_{\substack{q'\in\Omega_Q:\\ \xi_k^L(q')=\xi_k^L(q)\forall k\in G}}\pi(q',y)}\right] \\ &= \sum_{q\in\Omega_Q}\left[\frac{p^{(post)}(\{Q_k = \xi_k^L(q) : k \in G\})\times[m_{X/Y}^{(1)}(x_h(q),y) - m_{X/Y}^{(1)}(x_l(q),y)]}{\sum\limits_{\substack{q'\in\Omega_Q:\\ \xi_k^L(q')=\xi_k^L(q)\forall k\in G}}[m_{X/Y}^{(0)}(x_h(q'),y) - m_{X/Y}^{(0)}(x_l(q'),y)]}\right]\end{aligned} \tag{57}$$

Specifically, for the case where there are some hard-decoded planes (source coded or channel coded with a big margin), and some soft decoded planes, we can denote: $G = G_{soft} \cup G_{hard}$, where G_{soft} and G_{hard} are disjoint subsets of *G* with the hard and soft-decoded symbol indices respectively. Further, if the hard decoded values are $Q_k = q_k \forall k \in G_{hard}$, the optimal reconstruction can be rewritten as:

$$\hat{X} = \sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_{hard}}}\left[\frac{p^{(post)}(\{Q_k = \xi_k^L(q) : k \in G_{soft}\})\times\mu(q,y)}{\sum\limits_{\substack{q'\in\Omega_Q:\\ \xi_k^L(q')=q_k\forall k\in G_{hard}\\ \xi_k^L(q')=\xi_k^L(q)\forall k\in G_{soft}}}\pi(q',y)}\right] = \sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G_{hard}}}\left[\frac{p^{(post)}(\{Q_k = \xi_k^L(q) : k \in G_{soft}\})\times[m_{X/Y}^{(1)}(x_h(q),y) - m_{X/Y}^{(1)}(x_l(q),y)]}{\sum\limits_{\substack{q'\in\Omega_Q:\\ \xi_k^L(q')=q_k\forall k\in G_{hard}\\ \xi_k^L(q')=\xi_k^L(q)\forall k\in G_{soft}}}[m_{X/Y}^{(0)}(x_h(q'),y) - m_{X/Y}^{(0)}(x_l(q'),y)]}\right] \tag{58}$$

When all coded planes are hard-decoded, we have:

$$\hat{X} = \sum_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G}}\left[\frac{\mu(q,y)}{\sum\limits_{\substack{q'\in\Omega_Q:\\ \xi_k^L(q')=q_k\forall k\in G}}\pi(q',y)}\right] = \frac{\sum\limits_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G}}\mu(q,y)}{\sum\limits_{\substack{q\in\Omega_Q:\\ \xi_k^L(q)=q_k\forall k\in G}}\pi(q,y)} \tag{59}$$

which is similar to Eq. 8. The expression reduces to Eq. 20 when there are no skipped planes.

When there are skipped symbol-planes, or when the channel coded planes have not been coded with a sufficiently large margin, usually a certain probability of erroneous decoding is tolerated. In such cases, (partial) soft-decoding followed by the above form for the reconstruction function yields somewhat better reconstruction in practice.

3.2. A Practical Code family

3.2.1. Source-channel combo codes

Based on the background laid above, we now present a practical but generic Wyner-Ziv codec. We consider a symbol-plane by symbol-plane coder with *S*=*K*+1 symbols, where the alphabet-size vector is given by {*M*, 2, 2, …, (*K* 2s)}, where {*M*, *K*} are parameters for the code. The coding order is LSS to MSS. The *M*-ary LSS, which is the first symbol in coding order is source coded, while the remaining binary planes are each channel coded with powerful binary channel codes. Note that as observed from the pattern in Table 3 and Table 6, for LSS to MSS coding the conditional entropy decays very fast at the higher symbol planes. The source coding rate is given by the unconditional entropy in Eq. 52.

Since this is the first symbol plane, there are no complications in implementing context adaptive source codes and no possibility of error propagation to subsequent channel coded planes,. The channel coded binary planes in low to high significance order are coded with punctured binary error correction codes with rates given by adding a margin to the ideal rate. The case of K=1 is particularly convenient since there is only one channel coded plane preceded by a noise free source coded plane, and consequently there are no complications due to the possibility of error propagation. Optimal reconstruction can be then conducted based on Eq. 20 for hard decoding or Eq. 57–58 for full or partial soft-output decoding. The case M=1 for this code is a degenerate case, where the source coded symbol plane is non-existent, so that the code essentially becomes a bit-plane by bit-plane LSB to MSB channel coder with K bit-planes. Further, the case K=0, corresponds to the memoryless codes considered before in Section 2. In this case, there is only one source coded plane, and no channel coded planes.

The goal of parameter choice for this code is to obtain the appropriate values of $\{M, K\}$ and also the ideal rates to be used for the binary channel coded planes, given the source and correlation statistics $\{\sigma_X^2, \sigma_z^2\}$. The following algorithm may be used to find the optimal value of $\{M, K\}$, based on the fact that the conditional entropy of the un-transmitted true MSS X_S must be below a small threshold ε.

1. For each k in a set of allowable values: $\{1, 2, .., K_{\max}\}$
    a. Initialize m=1.
    b. Obtain conditional entropy $H(Q_{k+1}/Q_0,Q_1,...,Q_k,Y)$ with k+2-ary $L=\{m,\ 2,\ 2,\ldots(k\ 2\text{'s}),\ p\}$, with $p=\lceil(2q_{\max}+1)/(2^k m)\rceil$ to satisfy Eq.43. Note this is equivalent to computing $H(X_{k+1}/Q_0,Q_1,...,Q_k,Y)$ for a (k+1)-ary decomposition. (If m=1, there is no information in Q_0).
    c. If $H(Q_{k+1}/Q_0,Q_1,...,Q_k,Y)>\varepsilon$ do m=m+1 and go to Step 1b, else assign $M(k)$=m and go to step 1d.
    d. Obtain source coding rate $r_0^{SC}(k)=H(Q_0)$ for code parameters $\{M(k), k\}$. (If $M(k)$=1, $H(Q_0)$=0).
    e. Obtain ideal rate for binary planes: $H(Q_1/Q_0,Y), H(Q_2/Q_0,Q_{,1},Y),..., H(Q_k/Q_0,Q_{,1},...,Q_{k-1},Y)$.
    f. If k>1, check if: $H(Q_{k+1}/Q_0,Q_1,...,Q_k,Y)+H(Q_k/Q_0,Q_1,...,Q_{k-1},Y)<\varepsilon$. If so, assign $r_{practical}(k)$ =VERY_LARGE_VALUE and go to Step 1 and continue for next k. (In this case, a lower value of k should be used rather than the one tested).
    g. Compute practical channel coding rates: $r_1^{CC}(k)=(1+\gamma_1)H(Q_1/Q_0,Y)$, $r_2^{CC}(k)=(1+\gamma_2)H(Q_2/Q_0,Q_1,Y)$,…, $r_k^{CC}(k)=(1+\gamma_k)H(Q_k/Q_0,Q_1,...,Q_{k-1},Y)$ for code parameters $\{M(k), k\}$.
    h. Obtain total practical rate: $r_{practical}(k)=r_0^{SC}(k)+r_1^{CC}(k)+r_2^{CC}(k)+...+r_k^{CC}(k)$.
2. Find $K=\arg\min_k r_{practical}(k)$. The optimal code parameters are then $\{M(K), K\}$, with the channel coding rates as computed in Step 1e for this combination.

Table 9 tabulates the parameters chosen for the above algorithm for the case of Laplacian X (σ_X=1), and Gaussian Z (σ_Z=0.5), for varying values of QP_t, with ε=0.001. Further $K_{\max}$=1, i.e. only K=1 is the allowed configuration for practical convenience, corresponding to a 3-symbol code with $L=\{M, 2, \infty\}$. The ideal rates for coding, as well as the practical rate with the first symbol source coded and second symbol channel coded with a margin are provided. The margin factor, $\gamma_i=\gamma=0.5$ is assumed to be appropriate for the expected number of samples to be coded as a block, and the code complexity, and is assumed to be the same γ for each symbol plane. Note that this factor may be decided on the fly depending on the block size, if the number of samples in a block is not known beforehand.

Table 9. Parameters for K_{max}=1 code with $L=\{M, 2, \infty\}$ for Laplacian σ_X=1, Gaussian σ_Z=0.5, ε=0.001, γ= 0.5.

QP	***M***	***Ideal channel coded SP 0***	***Ideal channel coded BP 1***	***Ideal Sum (Slepian-Wolfe)***	***Source coded SP 0***	***Channel coded BP 1 (γ= 0.5)***	***Practical rate for code***	***All channel coded (γ= 0.5)***	***All source coded***
0.1	17	3.84449	0.17408	4.01917	4.00022	0.26112	**4.26134**	6.02876	5.13395
0.2	9	2.80101	0.11949	2.92076	3.02188	0.17924	**3.20111**	4.38114	4.02306
0.3	6	2.15673	0.10898	2.26589	2.39719	0.16347	**2.56066**	3.39884	3.34468
0.4	4	1.63590	0.17029	1.80716	1.81791	0.25544	**2.07334**	2.71074	2.85168
0.5	4	1.40658	0.05777	1.46434	1.72294	0.08666	**1.80959**	2.19651	2.46545
0.6	3	1.10005	0.09868	1.19942	1.33697	0.14802	**1.48499**	1.79913	2.15022
0.7	3	0.94590	0.04457	0.99048	1.26012	0.06686	**1.32698**	1.48572	1.88637

0.8	3	0.80385	0.01947	0.82332	1.17920	0.02921	**1.20841**	1.23498	1.66178
0.9	2	0.58216	0.10563	0.68814	0.75793	0.15845	**0.91637**	1.03221	1.46835
1.0	2	0.51042	0.06730	0.57779	0.71298	0.10095	**0.81393**	0.86669	1.30033
1.1	2	0.44437	0.04271	0.48710	0.66739	0.06407	**0.73146**	0.73065	1.15345
1.2	2	0.38502	0.02705	0.41208	0.62185	0.04058	**0.66243**	0.61812	1.02442
1.3	2	0.33258	0.01711	0.34968	0.57694	0.02567	**0.60261**	0.52453	0.91066
1.4	2	0.28674	0.01080	0.29754	0.53317	0.01620	**0.54937**	0.44631	0.81007
1.5	2	0.24698	0.00680	0.25379	0.49093	0.01020	**0.50113**	0.38068	0.72092
1.6	2	0.21266	0.00427	0.21693	0.45052	0.00641	**0.45693**	0.32540	0.64177
1.7	2	0.18310	0.00267	0.18577	0.41216	0.00401	**0.41617**	0.27866	0.57141
1.8	2	0.15770	0.00166	0.15936	0.37601	0.00250	**0.37850**	0.23905	0.50879
1.9	2	0.13588	0.00103	0.13691	0.34213	0.00155	**0.34368**	0.20537	0.45303
2.0	1	0.00000	0.11715	0.11778	0.00000	0.17572	**0.17572**	0.17667	0.40334

As we can see from the table, the practical rate with this code diverges substantially from the ideal Slepian Wolfe rate. However note, if only channel coding were used for this code with the same margin requirement, the rate would be (1+ γ) times as much as the Ideal Slepian-Wolfe rate shown in the second rightmost column, which is actually larger than the rate with the 3-symbol source-channel code at higher rates. At lower rates (*QP*>1) however, the channel-only code rate is lower. Also shown for comparison in the rightmost column is the rate if pure source coding were used.

If we allowed up to 2 channel coded bit-planes ($K_{\max}$=2), the inefficiency at the lower rates can be largely removed since the coding option with two channel coded planes but no source coding can now be chosen. Table 10 shows the parameters chosen when we allow both *K*=1 (3-symbol) and *K*=2 (4-symbol) codes for Laplacian *X* (σ_X=1) and Gaussian Z (σ_Z=0.5). Any entry with '–' indicates that the symbol plane is not coded. As we see that for certain mid-*QP* values, namely *QP*=0.5, 0.6, 0.7, 0.8, it becomes optimal to use *K*=2 channel coded bit-planes. At the lower rates *QP*>1, it again becomes optimal to use *K*=2 channel coded bit-planes, but the source coded symbol plane becomes degenerate at these rates (*M*=1). In other words, only two channel coded bit-planes are used, and use of source coded symbol plane is no longer optimal. At very low rates, QP≥2.0, it becomes sufficient to use a single channel coded bit-plane. The practical rates in Table 10 are by far the best ones that can be obtained by a practical code under the assumption of $\gamma = 0.5$.

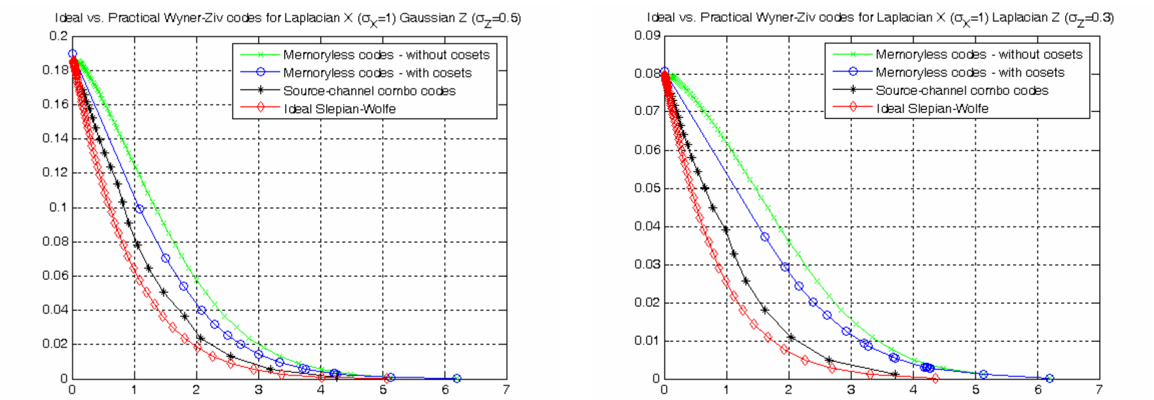
Table 10. Parameters for K_{max}=2 code for Laplacian σ_X=1, Gaussian σ_Z=0.5, ε=0.001, γ = 0.5

QP	***K***	***M***	***Ideal SP 0***	***Ideal channel coded BP1***	***Ideal channel coded BP2***	***Ideal Sum (Slepian-Wolfe)***	***Source coded SP 0***	***Channel coded BPs (γ= 0.5)***	***Practical rate for code***	***All channel coded (γ= 0.5)***	***All source coded***
0.1	1	17	3.84449	0.17408	–	4.01917	4.00022	0.26112	**4.26134**	6.02876	5.13395
0.2	1	9	2.80101	0.11949	–	2.92076	3.02188	0.17924	**3.20111**	4.38114	4.02306
0.3	1	6	2.15673	0.10898	–	2.26589	2.39719	0.16347	**2.56066**	3.39884	3.34468
0.4	1	4	1.63590	0.17029	–	1.80716	1.81791	0.25544	**2.07334**	2.71074	2.85168
0.5	2	2	0.86475	0.54184	0.05772	1.46434	0.91517	0.89934	**1.81451**	2.19651	2.46545
0.6	2	2	0.80324	0.37860	0.01758	1.19942	0.88098	0.59427	**1.47525**	1.79913	2.15022
0.7	2	2	0.73266	0.25288	0.00495	0.99048	0.84278	0.38675	**1.22953**	1.48572	1.88637
0.8	2	2	0.65743	0.16459	0.00130	0.82332	0.80148	0.24883	**1.05031**	1.23498	1.66178
0.9	1	2	0.58216	0.10563	–	0.68814	0.75793	0.15845	**0.91637**	1.03221	1.46835
1.0	1	2	0.51042	0.06730	–	0.57779	0.71298	0.10095	**0.81393**	0.86669	1.30033
1.1	2	1	–	0.44437	0.04271	0.48710	–	0.73062	**0.73062**	0.73065	1.15345
1.2	2	1	–	0.38502	0.02705	0.41208	–	0.61811	**0.61811**	0.61812	1.02442
1.3	2	1	–	0.33258	0.01711	0.34968	–	0.52453	**0.52453**	0.52453	0.91066
1.4	2	1	–	0.28674	0.01080	0.29754	–	0.44631	**0.44631**	0.44631	0.81007
1.5	2	1	–	0.24698	0.00680	0.25379	–	0.38068	**0.38068**	0.38068	0.72092
1.6	2	1	–	0.21266	0.00427	0.21693	–	0.32540	**0.32540**	0.32540	0.64177
1.7	2	1	–	0.18310	0.00267	0.18577	–	0.27866	**0.27866**	0.27866	0.57141
1.8	2	1	–	0.15770	0.00166	0.15936	–	0.23905	**0.23905**	0.23905	0.50879
1.9	2	1	–	0.13588	0.00103	0.13691	–	0.20537	**0.20537**	0.20537	0.45303
2.0	1	1	–	0.11715	–	0.11778	–	0.17572	**0.17572**	0.17667	0.40334

Table 11 shows the $K_{\max}$=2 code table for the case of Laplacian *X* (σ_X=1) and Laplacian Z (σ_Z=0.3).

Table 11. Parameters for K_{max}=2 code for Laplacian σ_X=1, Laplacian σ_Z=0.3, ε=0.001, γ = 0.5

QP	***K***	***M***	***Ideal SP 0***	***Ideal channel***	***Ideal channel***	***Ideal Sum***	***Source coded***	***Channel coded***	***Practical rate for***	***All channel***	***All source***



(a) Laplacian source (σ_X=1) Gaussian noise (σ_Z=0.5)   (b) Laplacian source (σ_X=1) Laplacian noise (σ_Z=0.3)

Figure 10. R-D curves of ideal Slepian-Wolfe coding vs. memoryless coding and practical finite memory coding with source and channel combo codes, for (a) Laplacian source *X* with σ_X=1, and Gaussian noise *Z* with σ_Z=0.5; (b) Laplacian source *X* with σ_X=1, and Laplacian noise *Z* with σ_Z=0.3.

				coded	***coded***	***(Slepian-***	***SP 0***	***BPs***	***code***	***coded***	***coded***
				BP1	***BP2***	***Wolfe)***		***(γ= 0.5)***		***(γ= 0.5)***	
0.1	2	8	2.81797	0.44170	0.05996	3.32030	2.97132	0.75248	**3.72380**	4.98044	5.13395
0.2	2	4	1.79082	0.41552	0.05396	2.26088	1.95408	0.70422	**2.65830**	3.39132	4.02306
0.3	2	3	1.31460	0.31964	0.02886	1.66327	1.51863	0.52276	**2.04139**	2.49490	3.34468
0.4	2	2	0.81938	0.40118	0.04909	1.27013	0.94452	0.67540	**1.61993**	1.90519	2.85168
0.5	2	2	0.72611	0.25195	0.01564	0.99374	0.91517	0.40138	**1.31655**	1.49061	2.46545
0.6	2	2	0.63191	0.15469	0.00496	0.79156	0.88097	0.23947	**1.12045**	1.18734	2.15022
0.7	2	2	0.54364	0.09422	0.00158	0.63945	0.84279	0.14370	**0.98649**	0.95917	1.88637
0.8	2	1	–	0.46460	0.05736	0.52246	–	0.78293	**0.78293**	0.78369	1.66178
0.9	2	1	–	0.39570	0.03501	0.43088	–	0.64607	**0.64607**	0.64632	1.46835
1.0	2	1	–	0.33660	0.02146	0.35811	–	0.53709	**0.53709**	0.53717	1.30033
1.1	2	1	–	0.28633	0.01322	0.29956	–	0.44932	**0.44932**	0.44934	1.15345
1.2	2	1	–	0.24377	0.00818	0.25195	–	0.37792	**0.37792**	0.37793	1.02442
1.3	2	1	–	0.20780	0.00508	0.21288	–	0.31932	**0.31932**	0.31932	0.91066
1.4	2	1	–	0.17741	0.00316	0.18057	–	0.27086	**0.27086**	0.27086	0.81007
1.5	2	1	–	0.15169	0.00198	0.15367	–	0.23050	**0.23050**	0.23050	0.72092
1.6	2	1	–	0.12990	0.00124	0.13114	–	0.19671	**0.19671**	0.19671	0.64177
1.7	1	1	–	0.11140	–	0.11219	–	0.16711	**0.16711**	0.16828	0.57141
1.8	1	1	–	0.09567	–	0.09617	–	0.14351	**0.14351**	0.14425	0.50879
1.9	1	1	–	0.08227	–	0.08258	–	0.12340	**0.12340**	0.12387	0.45303
2.0	1	1	–	0.07082	–	0.07101	–	0.10623	**0.10623**	0.10652	0.40334

Figure 10 compares the R-D curves for ideal Slepian-Wolfe coding followed by optimal reconstruction, with that of various practical codes including memoryless codes without cosets, the convex hull for memoryless codes with cosets and the above source-channel combo code, for both the Laplacian-Gaussian and Laplacian-Laplacian models. As expected, the last curve with memory enables getting closer to the ideal bound. Figure 11 compares R-D curves for source-channel combo codes for Laplacian and Gaussian Z with the same variance. As we see, the inefficiency with memoryless codes due to the heavier tail of the Laplacian Z has largely been eliminated.

One final point is that in certain coding scenarios the statistics (σ_X, σ_Z) may itself depend on the target QP_t for the non-Wyner-Ziv frames. In that case, each line in parameter tables above stored in the encoder and decoder could be obtained by a different statistics (σ_X, σ_Z). The decoder additionally needs to store the (σ_X, σ_Z) pair corresponding to each QP_t for appropriate decoding.

3.2.2. Practical coding architecture

The overall practical coding scheme can now be shown in Figure 12. Given a target QP_t, a pre-computed table yields the parameters *QP*, *K*, *M* as well as the ideal rates for the channel coded bit-planes. The least significant *M*-ary symbol is source coded with an entropy coder, for instance an arithmetic encoder. Ideally, each possible M should correspond to a different entropy coder, but alternatively an adaptive arithmetic encoder with an M-ary alphabet that automatically learns the distribution may be used.

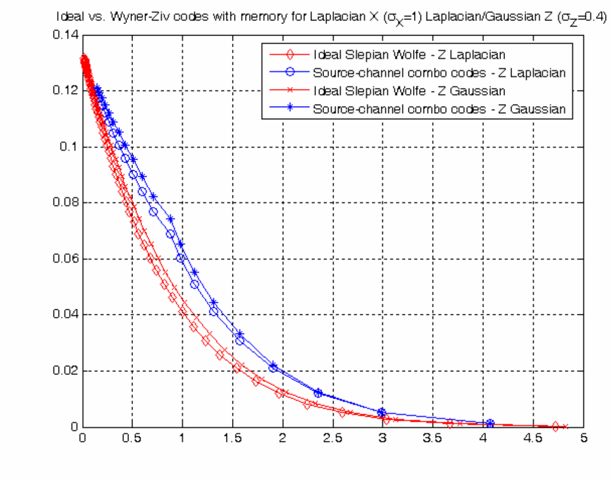


Figure 11. Comparing R-D curves for source channel combo codes for Laplacian and Gaussian Z

For actual channel coding of the bit-planes, powerful systematic codes such as LDPC codes or punctured Turbo codes may be used. However, if the number of samples is variable for each block, and not known beforehand, punctured Turbo codes will be found to be particularly advantageous for fast encoding. With LDPC codes, for every block of samples of unknown length to be coded, a new parity check matrix for a pseudo-random code with the specified rate would have to be instantiated. This set-up time during encoding can be too complex, even though once the set up is done encoding would be very simple. For punctured Turbo codes however, encoding with two constituent convolutional codes, followed by puncturing to obtain the required rate can all be done very fast in a straight-forward manner.

Decoding is conducted based on knowledge of the source decoded LSS (Q_0=q_0) if used, and the side information Y in order from the lower to higher significance. Any of the decoding strategies outlined above may be employed in the general case. However, if K=1, then there is a single channel coded bit-plane preceded by a source coded symbol-plane, and a soft-input soft-output decoder may be used very conveniently. In this case, the soft input prior probabilities are assumed to be obtained by computing: $p^{(prior)}(Q_1 = q_1) = p(Q_1 = q_1 / Q_0 = q_0, Y = y)$ using Eq. 53, while the soft-output posteriori probabilities $p^{(post)}(Q_1 = q_1)$ may be used in conjunction with Eq. 58 for the eventual reconstruction. Alternatively, Eq. 59 may be used after hard-thresholding the posteriori probabilities.

When an ensemble of samples with different statistics are coded, as is typical in a block transform coding scenario, one strategy is to code all the source coded Least Significant Symbols jointly within a block, even though they may be

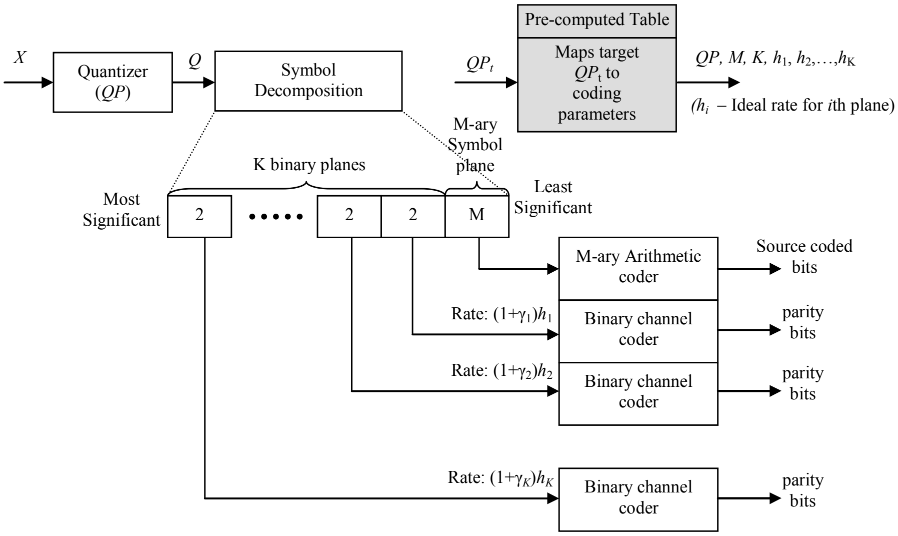


Figure 12. Illustrating Wyner-Ziv coding with source channel combo codes

associated with different size alphabets. This enables exploiting the higher order statistics of the data better to achieve better compression than our memoryless assumption yields. In addition, the channel coded bit-planes can be coded jointly for all coefficients across different source coded blocks to enable use of larger channel coding block lengths. Note that in this case, each sample may contribute a different channel coding rate for a given bit-plane to the overall the overall channel coding rate based on its statistics. The overall channel coding rate should then be taken as the average of the rate contribution for the concerned bit-plane over all the samples. The decoder, which is expected to know the statistics for each sample, computes the priors accordingly and decodes the channel code.

4. REFERENCES

[1] J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, pp. 471–480, July 1973.

[2] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.

[3] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," in *Proc. IEEE Data Compression Conf.*, 1999, pp. 158–167.

[4] A. Aaron and B. Girod, "Wyner-Ziv video coding with low-encoder complexity," *Proc. Picture Coding Symposium*, *PCS 2004*, San Francisco, CA, December 2004.

[5] A. Aaron, R. Zhang, B. Girod, "Transform-domain Wyner-Ziv coding for video," *Proc. Visual Communications and Image Processing*, San Jose, California, SPIE vol. 5308, pp. 520-528, Jan. 2004.

[6] R. Puri and K. Ramchandran, "PRISM: A 'reversed' multimedia coding paradigm*," Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, 2003.

[7] Q. Xu, Z. Xiong, "Layered Wyner-Ziv video coding," *Proc. Visual Communications and Image Processing*, San Jose, California, SPIE vol. 5308, pp. 83-91, 2004.

[8] H. Wang, N.-M. Cheung, A. Ortega, "A framework for Adaptive Scalable video coding using Wyner-Ziv techniques," *EURASIP Journal of Applied Signal Processing*, vol. 2006, pp. 1-18, Jan. 2006.

[9] M. Tagliasacchi, A. Majumdar, K. Ramachandran, "A distributed-source-coding based robust spatio-temporal scalable video codec," *Proc. Picture Coding Symposium*, San Francisco, 2004.

[10] X. Wang and M. Orchard, "Design of trellis codes for source coding with side information at the decoder," in *Proc. IEEE Data Compression Conf.*, 2001, pp. 361–370.

[11] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero**,** "Distributed video coding," *Proceedings of the IEEE*, Special Issue on Video Coding and Delivery*,* vol. 93, no. 1, pp. 71-83, January 2005.

[12] M. Wu, G. Hua, C. W. Chen, "Syndrome-based lightweight video coding for mobile wireless application," *Proc. Int. Conf. on Multimedia and Expo*, 2006, pp. 2013-2016.

[13] D. Mukherjee, "A robust reversed complexity Wyner-Ziv video codec introducing sign-modulated codes," *HP Labs Technical Report*, HPL-2006-80.

[14] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, *Numerical Recipes in C*, *Second Edition*, Cambridge University Press, 1992.