A SIMPLE REVERSED-COMPLEXITY WYNER-ZIV VIDEO CODING MODE BASED ON A SPATIAL REDUCTION FRAMEWORK

Debargha Mukherjee[†], *Bruno Macchiavello*[*], *Ricardo L. de Queiroz*[*]
[†] *Hewlett Packard Laboratories, Palo Alto, California, USA, Email: debargha@hpl.hp.com*
[*] *Universidade de Brasilia, Brazil, Email: bruno@image.unb.br, queiroz@ieee.org*

ABSTRACT

A spatial-resolution reduction based framework for incorporation of a Wyner-Ziv frame coding mode in existing video codecs is presented, to enable a mode of operation with low encoding complexity. The core Wyner-Ziv frame coder works on the Laplacian residual of a lower-resolution frame encoded by a regular codec at reduced resolution. The quantized transform coefficients of the residual frame are mapped to cosets to reduce the bit-rate. A detailed rate-distortion analysis and procedure for obtaining the optimal parameters based on a realistic statistical model for the transform coefficients and the side information is also presented. The decoder iteratively conducts motion-based side-information generation and coset decoding, to gradually refine the estimate of the frame. Preliminary results are presented for application to the H.263+ video codec.

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 distributed coding of various kinds of sources, notably video [3]-[10]. A good review of the area is presented in [11]. Besides improving noise resilience, one scenario where distributed video coding is promising is in creating *reversed complexity* codecs for power-constrained (hand-held) devices that capture and encode video either for real-time transmission or storage for subsequent decoding on a PC/server. Unlike regular broadcast-oriented video codecs with high encoding complexity and low decoding complexity, reversed complexity codecs have low encoding complexity but high decoding complexity. Prior work [4]-[6] address this scenario and propose encoding methods requiring no motion estimation at the encoder. Related work [7][8] address SNR scalability, and [9] address spatio-temporal scalability using distributed coding, but they also enable complexity reduction within their respective frameworks.

However, the true usage scenario for a power-constrained device may be somewhat different. For instance, low complexity encoding of captured video may be used only optionally when battery power is low, and bit-stream scalability may not be required. On the other hand, the same handheld device would very likely need to decode and playback received content not only from other handheld devices but also from other more powerful devices. While supporting two separate codecs is one option, it would be more convenient to have a single encoder that acts in two different modes with the ability to step-down to a lower (reversed) complexity encoding mode as required. Additionally, on the decoder side, it would be convenient if a lower quality version of the received content could still be played back immediately by simple decoding, while a higher quality version may be recovered only by a more intensive decoding process. Thus, a power constrained device should be able to switch to low complexity encoding mode when required, and its decoder should be able to support both regular decoding for a received regular bit-stream as well as at least reduced quality decoding for a received reversed complexity mode bit-stream. Further, this enhancement in functionality should be incorporated by a relatively modest change to an existing regular codec to minimize the impact on footprint, and facilitate adoption by the industry. Another consideration in our design has been the issue of efficiency. Most existing work in this area has been too aggressive in reducing complexity leading to a somewhat unacceptable loss in R-D efficiency. Our approach is moderate in complexity reduction target, but the target efficiency is higher.

We propose a spatial resolution reduction based framework [13] applicable to any existing video codec ([14], [15], etc.), where the encoding complexity as well as coding rate is reduced by lower resolution encoding through the same encoder, while the residual is Wyner-Ziv encoded with the rate savings. This enables a useful functionality fully integrated within an existing codec with minimal overheads. Recent work [12] also explores spatial reduction, but our mixed resolution approach can potentially yield a better rate-distortion performance by enabling better side-information generation.

2. SPATIAL REDUCTION FRAMEWORK

In the proposed framework, Wyner-Ziv coding for complexity reduction is applied to only the non-reference frames of a regular video coder, in order to eliminate drift due to incorrect decoding. These frames are called *Non-reference Wyner-Ziv* (NRWZ) *frames*. The reference frames are coded exactly as in a regular codec as I-, P- or reference B-

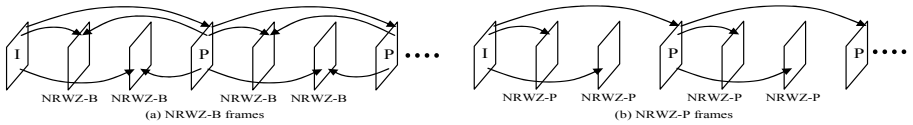


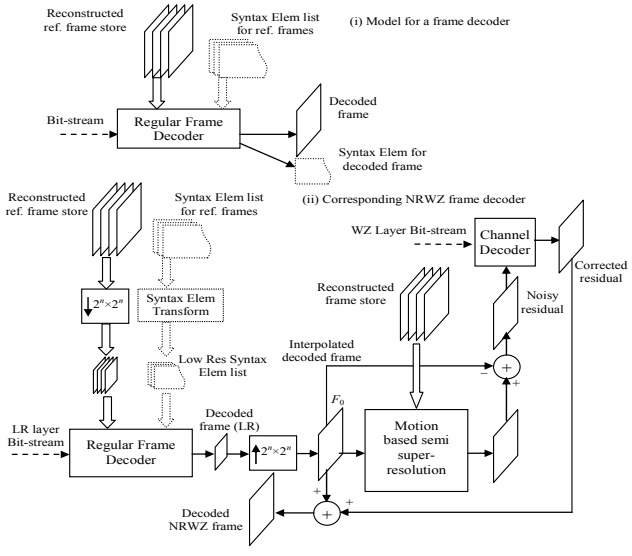
Figure 1. Use of NRWZ frames

frames. Figure 1 shows two scenarios how NRWZ frames can be used. In Figure 1(a), the B-frames of a regular coder have been converted to B-like NRWZ frames called the *NRWZ-B* frame, while Figure 1(b), shows a *low delay* case where P-like *NRWZ-P* frames are used instead. Ideally, the number of NRWZ frames in between P frames in both the cases shown can be varied dynamically based on the complexity reduction target.

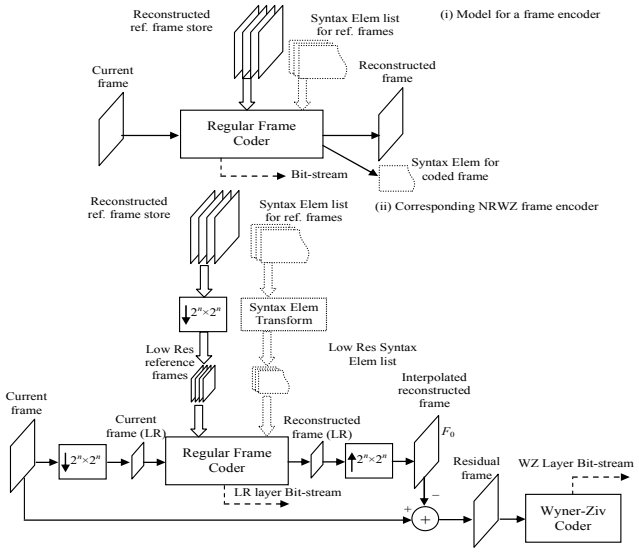
A general model for an inter frame encoder is shown in Figure 2(a)(i). Examples of usage of the syntax element object for reference frames include motion/mode information used for Direct-B prediction for B-frames, and generation of motion vector predictors for fast motion estimation. The corresponding NRWZ version of the encoder is created as shown in Figure 2 (a)(ii): First, the frames in the reconstructed frame-stores, as well as the current frame, are decimated by a factor $2^n \times 2^n$, where n can be chosen based on a complexity reduction target. The syntax element object list for reference frames are also transformed into a form that is appropriate for reduced resolution encoding. Next, the low-resolution (LR) current frame is encoded by running through the same frame encoder operating at reduced resolution, yielding the first part of the frame's bit-stream called the LR layer bit-stream. The quantization parameter used is the same as that corresponding to the target quality for the frame. The difference between the full resolution current frame and an interpolated reconstruction from the LR encoder denoted F_0, is computed to yield a residual frame. Finally, a Wyner-Ziv coder is used to code this residual, generating a Wyner-Ziv bit-stream layer. The encoder and the decoder use the same filters for decimation and interpolation.

It is straight-forward to see that the complexity of encoding NRWZ frames is roughly scaled down by a factor (2^{-2n} + α) irrespective of the encoder implementation, where the overheads due to decimation, interpolation, syntax element transformation, and Wyner-Ziv coding operations, are assumed to together contribute a factor α of the regular complexity of the full resolution encoder. Typically, α is low. A low complexity decoder can still playback a received sequence with decent quality by decoding only the key frames, and/or by spatial interpolation of the decoded LR layer. More complex decoding can be performed offline to recover a better quality NRWZ frames.

The decoder architecture for NRWZ frames is shown in Figure 2(b). Figure 2(b)(i) shows the model for a regular decoder, while Figure 2(b)(ii) shows the high-level decoder model for the corresponding NRWZ version. First, the low-resolution image is decoded and then interpolated with the same interpolator used in the encoder to yield the interpolated low resolution reconstruction F_0. Second, F_0 as well as the previously decoded frames in a frame-store denoted **FS**, are used in a motion-based processing module to obtain a higher resolution estimate of the frame to be decoded denoted



(a) Coding Architecture for NRWZ frames



(b) Decoding Architecture for NRWZ frames

Figure 2. Architecture for NRWZ frames

$F_0^{(HR)}$. We call this the multi-frame *semi super-resolution* problem, because except for the current frame, the other frames used are already at higher resolution, albeit corrupted with quantization noise. Third, compute the side-information residual frame $R_0 = F_0^{(HR)} - F_0$ to be used for channel decoding. Fourth, the channel decoder decodes the WZ bit-stream layer based on R_0 to obtain the corrected residual $R_0^{(\mathrm{cor})}$. The final decoded frame F_1 is obtained by computing $F_1 = R_0^{(\mathrm{cor})} + F_0$.

In practice, it is more efficient to iterate the semi-super-resolution computation followed by channel decoding in multiple passes. If $SS(F, \mathbf{FS})$ denotes the semi-super-resolution operation yielding a high resolution version of F based on the frames stored in **FS**, and $CD(R, b_{\mathrm{WZ}})$ denotes the channel decoding operation yielding a corrected residual frame based on noisy version R and the WZ layer bit-stream b_{WZ}, then iterative decoding comprises for $i = 0, 1, \ldots, N-1$:

$$F_i^{(HR)} = SS(F_i, \mathrm{FS}), R_i = F_i^{(HR)} - F_0, R_i^{(cor)} = CD(R_i, b_{WZ}), F_{i+1} = R_i^{(cor)} + F_0 : \quad F_N \text{ is the final decoded frame after N iterations} \quad (1)$$

3. SEMI SUPER-RESOLUTION SIDE-INFORMATION GENERATION

A block-based scheme for semi super-resolution was used where **FS** consists of only the past and future reference frames coded at full-resolution. First, the reference frames are low-pass filtered. Next, for every 8×8 block in frame F_i, the best sub-pixel motion vectors in the past and future filtered frames in a certain neighborhood is computed. If the corresponding best predictor blocks in the past and future filtered frames are denoted B_p and B_f respectively, several candidate predictors of the type $\alpha B_p + (1-\alpha)B_f$ with $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, are tested and the best predictor that minimizes the SAD of the current block in F_i is found. If the SAD for the best predictor is more than a certain threshold T_i, then nothing is done to the block. Otherwise, the block in F_i is replaced by the best predictor but with the compensation now conducted from unfiltered past and future frames. When all blocks in F_i have been processed, the updated frame is referred to as $F_i^{(HR)}$. In practice, the low pass filtering operation for the reference frames is eliminated after one or two iterations as the frame becomes more and more accurate. Further, the grid for block matching is offset from iteration to iteration to smooth out the blockiness and add spatial coherence. For example, the shifts used in four passes can be (0, 0), (4, 0), (0, 4) and (4, 4). Finally, the threshold T_i is also be gradually reduced from iteration to iteration, so that fewer blocks are changed in later iterations.

4. CORE WYNER-ZIV CODER

Our Wyner-Ziv coder operates on the residual error frame in the block-transform domain. The same transform as used in a regular codec (ex. DCT for H.263+) can be used. In a codec where multiple transforms are used, the largest one is preferred.

4.1. Encoding

After computing the transform, the transform coefficients denoted by random variable X, are quantized, possibly with dead zone, to yield a quantization index random variable Q: $Q = \phi(X, QP)$, QP being the quantization step-size. Q takes values from the set $\Omega_Q = \{-q_{\max}, -q_{\max+1}, \ldots, -1, 0, 1, \ldots, q_{\max}-1, q_{\max}\}$. Next, 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, using:

$$\psi(Q,M) = \begin{cases} Q - M\lfloor Q/M \rfloor & , \; Q - M\lfloor Q/M \rfloor < M/2 \\ Q - M\lfloor Q/M \rfloor - M & , \; Q - M\lfloor Q/M \rfloor \geq M/2 \end{cases} \quad (2)$$

C takes values from the set $\Omega_C = \{\lfloor -(M-1)/2 \rfloor, \ldots, -1, 0, 1, \ldots, \lfloor (M-1)/2 \rfloor\}$. The above form ensures that coset indices are centered on 0. QP and M are different for each frequency (i,j) of coefficient x_{ij}.

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:

$$p_Q(q) = \int_{x_l(q)}^{x_h(q)} f_X(x)dx \qquad p_C(c) = \sum_{q \in \Omega_Q : \psi(q,M)=c} p_Q(q) = \sum_{q \in \Omega_Q : \psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} f_X(x)dx \quad (3)$$

Where $f_X(x)$ is the pdf of X. Examples of both are shown in Figure 3, for M odd 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 can be reused for c, and turns out to be quite efficient. While a different entropy coder designed specifically for coset indices can have some efficiency advantage, reuse of the same entropy coder minimizes additions needed to the regular codec.

In practice macroblocks are classified into one of several types $s \in \{0,1,\ldots,S-1\}$ based on an estimate of the noise level between the side-information block and the original. Various cues from the low resolution layer can be used for this purpose. In this work, a combination of the number of bits spent to code the corresponding residual in the low

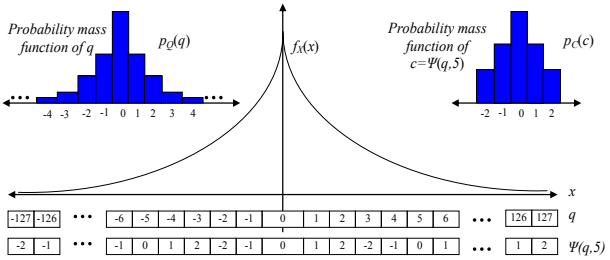


Figure 3. Probability mass function of coset indices

resolution layer and an edge activity measure is used. The coding parameters, Q and M are varied based on s, and are denoted $Q_{ij}(s)$ and $M_{ij}(s)$ respectively in the most general terms. Also, only a few low to mid frequency coefficients are sent for each block while the rest are forced to zero. The maximum number of coefficients transmitted in zigzag scan order before zero-forcing is determined based on class s, and denoted $n_{max}(s)$. Figure 4 summarizes the encoding steps.

4.2. Noise model

Ideally, the parameters $Q_{ij}(s)$ and $M_{ij}(s)$ should be matched to the correlation statistics between the side-information and the original transform coefficients. The random variable X corresponding to transform coefficients, are assumed to be Laplacian distributed with std. dev. σ_X. Further, if Y denotes the corresponding (unquantized) side-information, then we assume $Y = X + Z$ where the noise Z is uncorrelated with X, and modeled as a Gaussian with std. deviation σ_Z. The std. dev pair $\{\sigma_X, \sigma_Z\}$ not only depends on frequency and class, but also on the target quantization parameter QP for the reference frames and the LR layer. They can be estimated offline based on training sequences for a given semi super-resolution operation. In Section 5, we will see how the parameters $Q_{ij}(s)$ and $M_{ij}(s)$ should be chosen given the std. dev. pair $\{\sigma_X, \sigma_Z\}$.

4.3. Decoding

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:

$$\hat{X}_{YC}(y,c) = E(X/Y=y, C=c) = E(X/Y=y, \psi(\phi(X,QP),M)=c) = \frac{\sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx}{\sum_{q\in\Omega_Q:\psi(q,M)=c} \int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx} \quad (4)$$

where $[x_l(q), x_h(q)]$ is the interval corresponding to quantization bin q.

The class index s and the frequency (ij) of a coefficient not only yields the quantization step-size $Q_{ij}(s)$ and coset modulus $M_{ij}(s)$, but also map to the model parameters $\{\sigma_X, \sigma_Z\}$estimated offline to be used for the computation above. Unfortunately, while exact computation of Eq. 4 is difficult based on the noise model, various approximations or interpolation on various pre-computed tables can yield a practical solution. Figure 5 illustrates the decoding principle.

The coefficients that were forced to zero during encoding are reconstructed exactly as they appear in the side-information.

5. PARAMETER CHOICE BASED ON SOURCE AND SIDE-INFORMATION STATISTICS

In this section we study in detail the problem of making the optimal choice of the quantization parameter QP and coset modulus M for coding a source X with known statistics, where the side information Y available only at the decoder is obtained by: $Y = X + Z$, where Z is additive noise uncorrelated with X. Starting from the general formulation of the

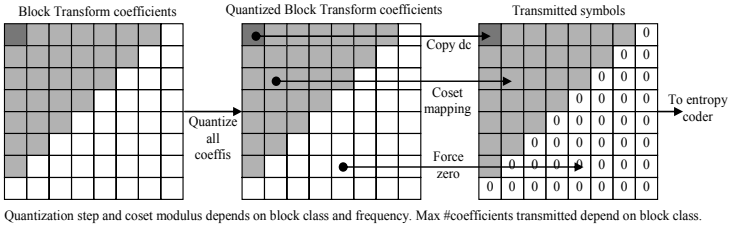


Figure 4. Block transform based WZ coding steps

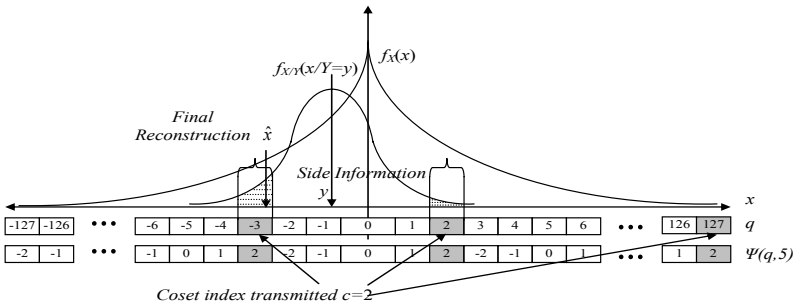


Figure 5. Decoding example

Rate-Distortion characteristics, we will derive the specific characterization for the case where X is Laplacian distributed with zero mean and variance σ_X^2, and Z is Gaussian with zero mean and variance σ_Z^2. Further, we will assume a deadzone quantizer typically used in a practical codec. We believe that this characterization would be very useful in many transform-domain Wyner-Ziv coding scenarios since transform coefficients closely follow the Laplacian distribution. Therefore studying this problem will not only help optimize the coder presented here, but also a variety of other similar coders.

Specifically, the goal of this characterization would be 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. This criterion will be referred to as *distortion target matching*. The variances of the Laplacian source (σ_X^2) and the additive white Gaussian noise (σ_Z^2), are assumed to be known. For the specific codec described in this work, the variances for each coefficient frequency and potentially each class, are obtained from training data for a given block classification scheme, while QP_t is the quantization step-size corresponding to the target quality.

5.1. Rate-Distortion characterization

We first consider the rate-distortion functions for various Wyner-Ziv coding scenarios. The first is the one adopted in this work. The rest correspond to ideal Slepian-Wolf coding, non-distributed coding and zero-rate coding respectively, used for various comparisons and distortion target matching.

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

The probability of each coset index is known from the probability mass function in Eq. 3. Assuming an ideal entropy coder for the coset indices, the expected rate would be the entropy of the source C:

$$E(R_{YC}) = H(C) = -\sum_{c\in\Omega_C} p_C(c)\log_2 p_C(c) = -\sum_{c\in\Omega_C}\{\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)} f_X(x)dx\}\log_2\{\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)} f_X(x)dx\} \tag{5}$$

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

$$E(R_{YC}) = -\sum_{c\in\Omega_C}\{\sum_{q\in\Omega_Q:\psi(q,M)=c}[m_X^0(x_h(q)) - m_X^0(x_l(q))]\}\log_2\{\sum_{q\in\Omega_Q:\psi(q,M)=c}[m_X^0(x_h(q)) - m_X^0(x_l(q))]\} \tag{6}$$

Assuming the minimum mean-squared-error reconstruction function in Eq. 4, the expected distortion D_{YC} given side information y and coset index c is given by:

$$E(D_{YC}/Y=y, C=c) = E([X-\hat{X}_{YC}(y,c)]^2/Y=y, C=c) = E(X^2/Y=y, C=c) - \hat{X}_{YC}(y,c)^2 \tag{7}$$

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

$$\begin{aligned} E(D_{YC}) &= E(X^2) - \int_{-\infty}^{\infty}\{\sum_{c\in\Omega_C}\hat{X}_{YC}(y,c)^2 p_{C/Y}(C=c/Y=y)\}f_Y(y)dy \\ &= \sigma_X^2 - \int_{-\infty}^{\infty}\{\sum_{c\in\Omega_C}\left(\frac{\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)} x f_{X/Y}(x,y)dx}{\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)} f_{X/Y}(x,y)dx}\right)^2 p_{C/Y}(C=c/Y=y)\}f_Y(y)dy \end{aligned} \tag{8}$$

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

$$p_{C/Y}(C=c/Y=y)=\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)}f_{X/Y}(x,y)dx \tag{9}$$

we have:

$$E(D_{YC})=\sigma_X^2-\int_{-\infty}^{\infty}\{\sum_{c\in\Omega_C}\frac{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)}xf_{X/Y}(x,y)dx\right)^2}{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c}\int_{x_l(q)}^{x_h(q)}f_{X/Y}(x,y)dx\right)}\}f_Y(y)dy \tag{10}$$

Defining:

$$m_{X/Y}^{(i)}(x,y)=\int_{-\infty}^{x}x'^{i}f_{X/Y}(x',y)dx' \tag{11}$$

we can rewrite Eq. 10 as:

$$E(D_{YC})=\sigma_X^2-\int_{-\infty}^{\infty}\{\sum_{c\in\Omega_C}\frac{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c}[m_{X/Y}^{(1)}(x_h(q),y)-m_{X/Y}^{(1)}(x_l(q),y)]\right)^2}{\left(\sum_{q\in\Omega_Q:\psi(q,M)=c}[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)]\right)}\}f_Y(y)dy \tag{12}$$

5.1.2. 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:

$$\begin{aligned}E(R_{YQ})&=H(Q/Y)\\&=-\int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}p_{Q/Y}(Q=q/Y=y)\log_2 p_{Q/Y}(Q=q/Y=y)\}f_Y(y)dy\\&=-\int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)]\log_2[m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)]\}f_Y(y)dy\end{aligned} \tag{13}$$

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:

$$\hat{X}_{YQ}(y,q)=E(X/Y=y,Q=q)=E(X/Y=y,\phi(X,QP)=c)=\frac{\int_{x_l(q)}^{x_h(q)}xf_{X/Y}(x,y)dx}{\int_{x_l(q)}^{x_h(q)}f_{X/Y}(x,y)dx}=\frac{m_{X/Y}^{(1)}(x_h(q),y)-m_{X/Y}^{(1)}(x_l(q),y)}{m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)} \tag{14}$$

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

$$E(D_{YQ})=\sigma_X^2-\int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}\frac{\left(\int_{x_l(q)}^{x_h(q)}xf_{X/Y}(x,y)dx\right)^2}{\left(\int_{x_l(q)}^{x_h(q)}f_{X/Y}(x,y)dx\right)}\}f_Y(y)dy=\sigma_X^2-\int_{-\infty}^{\infty}\{\sum_{q\in\Omega_Q}\frac{\left(m_{X/Y}^{(1)}(x_h(q),y)-m_{X/Y}^{(1)}(x_l(q),y)\right)^2}{\left(m_{X/Y}^{(0)}(x_h(q),y)-m_{X/Y}^{(0)}(x_l(q),y)\right)}\}f_Y(y)dy \tag{15}$$

5.1.3. 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.

$$E(R_Q)=H(Q)=-\sum_{q\in\Omega_Q}p_Q(q)\log_2 p_Q(q)=-\sum_{q\in\Omega_Q}[m_X^{(0)}(x_h(q))-m_X^{(0)}(x_l(q))]\log_2[m_X^{(0)}(x_h(q))-m_X^{(0)}(x_l(q))] \tag{16}$$

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. 14 and Eq. 15 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:

$$\hat{X}_Q(q) = E(X/Q=q) = E(X/\phi(X,QP)=q) = \frac{\int_{x_l(q)}^{x_h(q)} x f_X(x)dx}{\int_{x_l(q)}^{x_h(q)} f_X(x)dx} = \frac{m_X^{(1)}(x_h(q)) - m_X^{(1)}(x_l(q))}{m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q))} \tag{17}$$

while the expected distortion is given by:

$$E(D_Q) = \sigma_X^2 - \sum_{q\in\Omega_Q} \frac{\left(\int_{x_l(q)}^{x_h(q)} x f_X(x)dx\right)^2}{\left(\int_{x_l(q)}^{x_h(q)} f_X(x)dx\right)} = \sigma_X^2 - \sum_{q\in\Omega_Q} \frac{\left(m_X^{(1)}(x_h(q)) - m_X^{(1)}(x_l(q))\right)^2}{\left(m_X^{(0)}(x_h(q)) - m_X^{(0)}(x_l(q))\right)} \tag{18}$$

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. 18. The parameters QP and M for memoryless coset codes should be chosen such that the lowest rate $E(R_{YC})$ given by Eq. 5 is obtained, with the expected distortion $E(D_{YC})$ given by Eq. 12 being equivalent to the target distortion.

5.1.4. 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 $\hat{X}_Y(y)$:

$$\hat{X}_Y(y) = E(X/Y=y) = \int_{-\infty}^{\infty} x f_{X/Y}(x,y)dx = m_{X/Y}^{(1)}(\infty, y) \tag{19}$$

The expected zero-rate distortion D_Y is given by:

$$E(D_Y) = \sigma_X^2 - \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} x f_{X/Y}(x,y)dx\right)^2 f_Y(y)dy = \sigma_X^2 - \int_{-\infty}^{\infty} m_{X/Y}^{(1)}(\infty,y)^2 f_Y(y)dy \tag{20}$$

5.2. Laplacian Source with additive Gaussian noise

5.2.1. Expressions

While the expressions in Section 5.1 are generic, we now specialize for the case of Laplacian X and Gaussian Z, *i.e.*:

$$f_X(x) = \frac{1}{\sqrt{2}\sigma_X} e^{-\left|\frac{x\sqrt{2}}{\sigma_X}\right|}, \quad f_Z(z) = \frac{1}{\sqrt{2\pi}\sigma_Z} e^{-\frac{1}{2}\left|\frac{z}{\sigma_Z}\right|^2} \tag{21}$$

In the following, we assume:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{22}$$

Then, defining

$$\beta(x) = e^{\frac{\sqrt{2}x}{\sigma_X}} \tag{23}$$

we have

$$m_X^{(0)}(x) = \begin{cases} \dfrac{\beta(x)}{2}, & x \le 0 \\ 1 - \dfrac{1}{2\beta(x)}, & x > 0 \end{cases} \qquad m_X^{(1)}(x) = \begin{cases} \dfrac{\beta(x)}{2\sqrt{2}}(\sqrt{2}x - \sigma_X), & x \le 0 \\ -\dfrac{1}{2\sqrt{2}\beta(x)}(\sqrt{2}x + \sigma_X), & x > 0 \end{cases} \tag{24}$$

Further defining:

$$\gamma_1(x) = erf\left(\frac{\sigma_X x - \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}\right), \quad \gamma_2(x) = erf\left(\frac{\sigma_X x + \sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z}\right) \tag{25}$$

and using $Y=X+Z$, we have:

$$\begin{aligned} f_{XY}(x,y) &= \frac{1}{2\sqrt{\pi}\sigma_X\sigma_Z} e^{-\left|\frac{x\sqrt{2}}{\sigma_X}\right|} e^{-\frac{1}{2}\left(\frac{y-x}{\sigma_Z}\right)^2} \\ f_Y(y) &= \int_{-\infty}^{\infty} f_{XY}(x,y)dx = \frac{1}{2\sqrt{2}\beta(y)\sigma_X} e^{\sigma_X^2/\sigma_Z^2}[\gamma_1(y)+1.0-\beta(y)^2(\gamma_2(y)-1.0)] \\ f_{X/Y}(x,y) &= \frac{f_{XY}(x,y)}{f_Y(y)} = \frac{\sqrt{2}\beta(y)}{\sqrt{\pi}\sigma_Z} \frac{e^{-\left|\frac{x\sqrt{2}}{\sigma_X}\right| - \frac{1}{2}\left(\frac{y-x}{\sigma_Z}\right)^2 - \frac{\sigma_X^2}{\sigma_Z^2}}}{[\gamma_1(y)+1.0-\beta(y)^2(\gamma_2(y)-1.0)]} \end{aligned} \tag{26}$$

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

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

$$m_{X/Y}^{(1)}(x,y) = \begin{cases} \frac{\beta(y)^2[y+\sqrt{2}\frac{\sigma_Z^2}{\sigma_X}][erf(\frac{\sigma_X(y-x)+\sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z})-1]+\frac{\sqrt{2}}{\sqrt{\pi}}\sigma_Z\beta(x)^2 e^{\frac{-\frac{1}{2}(\sigma_X(y-x)-\sqrt{2}\sigma_Z^2)^2}{\sigma_X^2\sigma_Z^2}}}{[\gamma_1(y)+1-\beta(y)^2(\gamma_2(y)-1)]}, & x \le 0 \\ \frac{\beta(y)^2[y+\sqrt{2}\frac{\sigma_Z^2}{\sigma_X}](\gamma_2(y)-1)+[y-\sqrt{2}\frac{\sigma_Z^2}{\sigma_X}][erf(\frac{\sigma_X(y-x)-\sqrt{2}\sigma_Z^2}{\sqrt{2}\sigma_X\sigma_Z})-\gamma_1(y)]+\frac{\sqrt{2}}{\sqrt{\pi}}\sigma_Z e^{\frac{-\frac{1}{2}(\sigma_X(y-x)-\sqrt{2}\sigma_Z^2)^2}{\sigma_X^2\sigma_Z^2}}}{[\gamma_1(y)+1-\beta(y)^2(\gamma_2(y)-1)]}, & x > 0 \end{cases} \tag{27}$$

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* [16] . 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 .

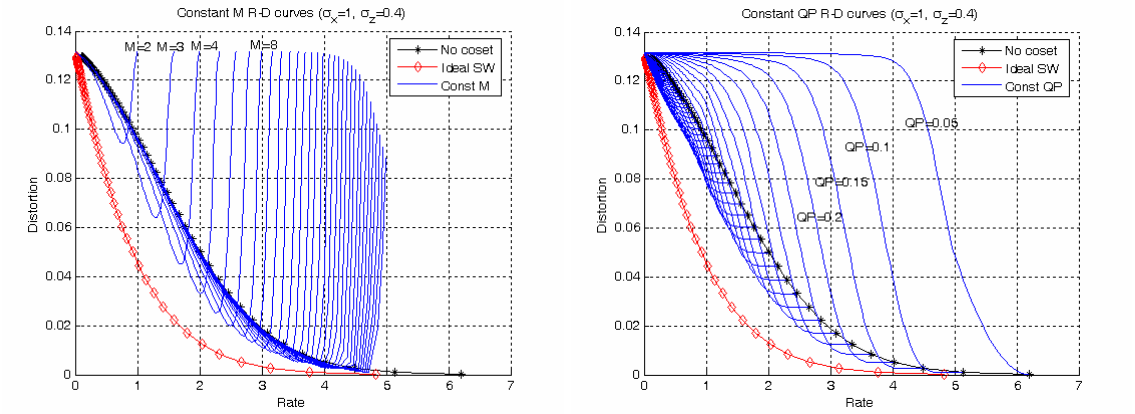
5.2.2. R-D curves for deadzone quantizer and optimal parameter choice

We next present the R-D curves for a deadzone quantizer given by:

$$\phi(X, QP) = sign(X) \times \left\lfloor |X| / QP \right\rfloor \tag{28}$$

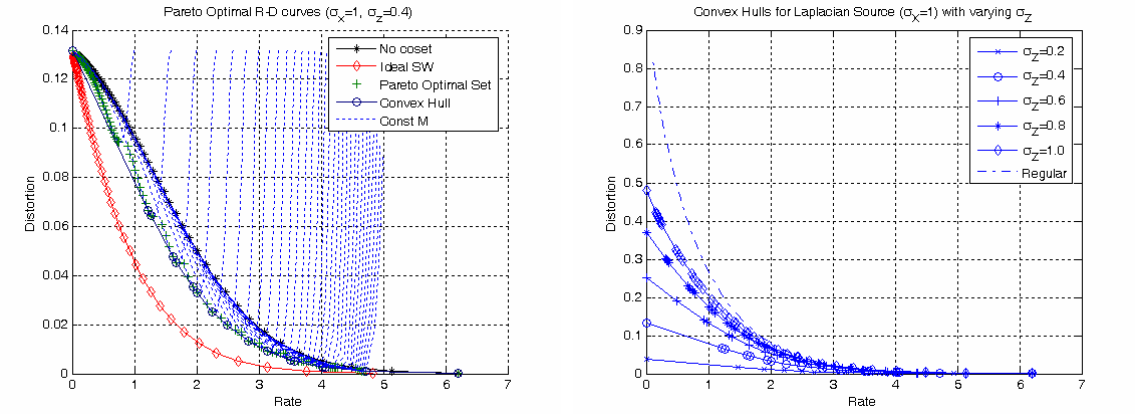
and the coset modulus function given by Eq. 2, obtained by changing the parameters *QP* and *M*. Note that while *M* is always discrete, *QP* can in general be continuous. However we have sampled it at regular intervals in the R-D curves presented below. On the other hand, for most real codecs, the *QP* is indeed discrete.

Figure 6(a) and (b) shows two ways of presenting the curves for the specific case of σ_X=1, and σ_Z=0.4. In Figure 6(a) each R-D curve is generated by fixing *M* and changing *QP* at finely sampled intervals of 0.05. However, the following discussion assumes *QP* to be continuous. The case $QP \to \infty$ 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. 20. Alternatively, this point can also be viewed as the M=1 curve which degenerates to a point. The other extreme is the case where $QP \to 0+$. 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 '*' correspond to the non-distributed coding case with minimum MSE reconstruction using side-information given by Eq. 16 and Eq. 15 respectively, while the line with diamonds correspond to ideal Slepian-Wolfe coding followed by minimum MSE reconstruction. Figure 6(b) shows the same results but now using constant *QP* curves. Each curve in the 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 $QP \to \infty$ curve that degenerates to a point. As $M \to \infty$ 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.



(a) Constant M curves

(b) Constant QP curves



(c) Pareto Optimal and Set and Convex Hull

(d) Convex Hulls for varying σ_Z

Figure 6. R-D curves obtained by changing QP and M

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 6(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. 18, 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:

$$k^* = \underset{k=n_P+1,\ldots,|P|-1}{\arg\min} \{(D_{H[n_H-1]} - D_{P[k]})/(R_{H[n_H-1]} - R_{P[k]})\}, \quad H[n_H] = P[k^*], \quad n_H = n_H + 1, \quad n_P = k^* \tag{29}$$

   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.

Figure 6(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–α 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– α) 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 σ_X=1, σ_Z=0.4 is shown in Table 1, where the QP are 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.

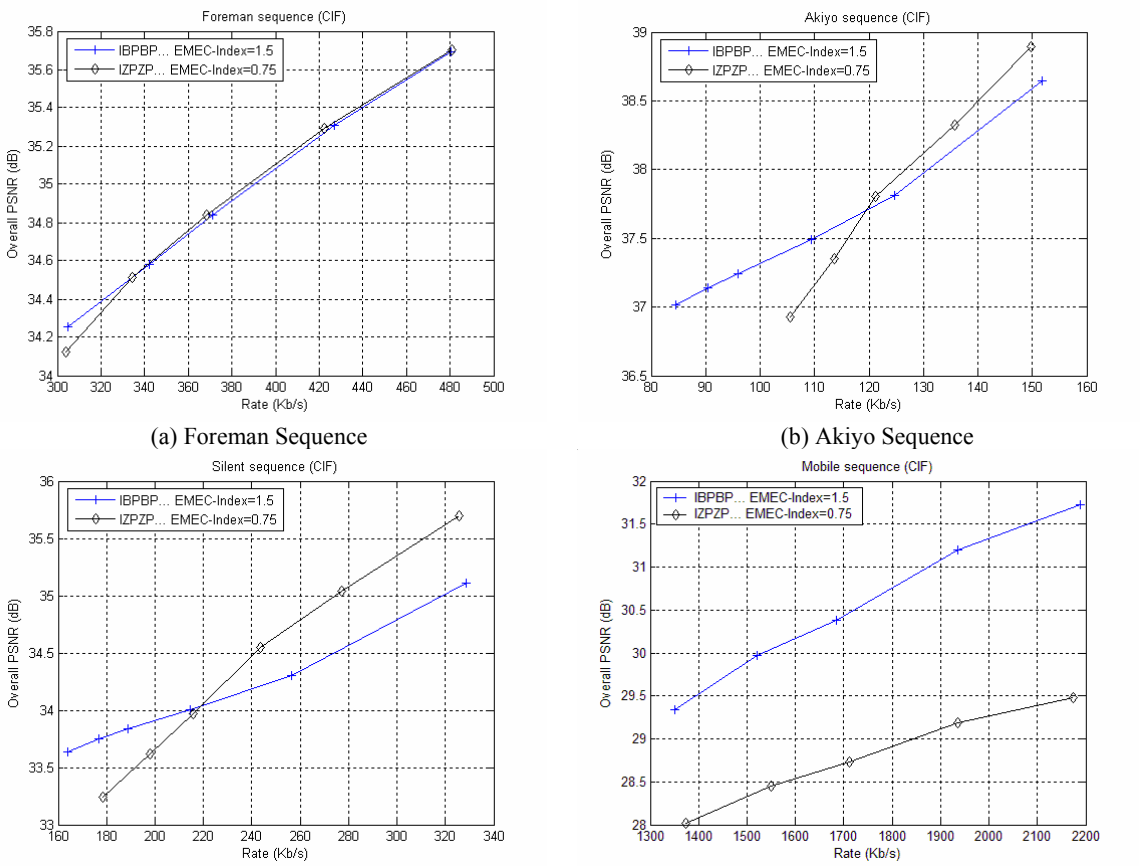
Table 1. Look-up table from target QP_t to 5-tuple parameters for σ_X=1, σ_Z=0.4

QP_t	QP_1	M_1	QP_2	M_2	α
0.05	0.10	32	0.05	∞	0.93314
0.10	0.15	21	0.10	32	0.90638
0.15	0.20	15	0.15	20	0.98211
0.20	0.20	14	0.20	15	0.39819
0.25	0.30	9	0.25	11	0.96786
0.30	0.35	7	0.30	9	0.87608
0.35	0.40	6	0.35	7	0.92355
0.40	0.45	5	0.40	6	0.74711
0.45	0.55	4	0.50	5	0.97749
0.50	0.55	4	0.50	5	0.03730
0.55	0.70	3	0.60	4	0.54183
0.60	∞	1	0.75	3	0.99238
0.65	∞	1	0.75	3	0.80090
0.70	∞	1	0.75	3	0.59556
0.75	∞	1	0.75	3	0.37739
0.80	∞	1	0.75	3	0.14747
0.85	∞	1	∞	1	0
0.90	∞	1	∞	1	0
0.95	∞	1	∞	1	0
1.00	∞	1	∞	1	0

Figure 6(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. 16 and Eq. 18). 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 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.

5.2.3. 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.



(a) Foreman Sequence (b) Akiyo Sequence

(c) Silent Sequence (d) Mobile Sequence

Figure 7. R-D results for various sequences

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.

6. RESULTS ON H.263+

A reversed complexity coding mode based on the above principles has been integrated within the H.263+ video codec. In this mode, the B-frames in the regular codec are replaced by NRWZ-B frames. The base layers of the NRWZ-B frames are coded at quarter resolution. In order to handle the Direct-B prediction modes in NRWZ-B frames, the motion vectors and modes from the full-resolution P-frames, are transformed appropriately.

The coding performance of a reversed complexity codec operating in IZPZPZPZPZ…. mode with 'Z' frames indicating NRWZ-B frames, is compared against a H.263+ coder, operating in IBPBPBPBPB… mode, in Figure 7 for the *Foreman, Akiyo, Silent* and *Mobile* CIF sequences. The *encoder motion estimation complexity (EMEC) index* shown compares the average per frame complexity due to motion estimation of each encoder in relation to that of a regular P-frame. The results are quite comparable for three of the sequences, especially at higher rates even though the IZPZP… codec has an EMEC index half that of the IBPBP... codec. Interestingly, at some rates, the proposed coder actually performs better than the regular codec, because the side-information generation operation has an effect of post-processing, even though the exact component in the side-information generation operation that is equivalent to post-

processing cannot be separated. For the *Mobile* sequence however, there is substantial quality degradation apparently due to failure of the side-information generation process.

7. CONCLUSION

The design principles and preliminary results for a reversed complexity coding mode based on Spatial reduction, as applied to H.263+ is presented. However the methodology is generic enough to allow incorporation of a similar mode in other codecs, notably H.264/AVC. Future work would involve improving the side-information generation process which in fact holds the most potential for improving the overall performance, using better entropy coding of the Wyner-Ziv layer, and using more powerful channel codes for the Wyner-Ziv layer.

8. 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] G. Cote, B. Erol, M. Gallant, F. Kossentini, "H.263+: Video coding at low bit-rates," IEEE Trans. *Circuits Syst. Video Technology*, vol. 8, no. 7, pp. 849–866, Nov. 1998.

[15] T. Wiegand, G. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVCvideo coding standard," *IEEE Trans. Circuits Syst. Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

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