Wyner-Ziv Coding of Motion Video

Anne Aaron, Rui Zhang, and Bernd Girod
Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford, CA 94305
{amaaron, rui, bgirod}@stanford.edu

ABSTRACT

In current interframe video compression systems, the encoder performs predictive coding to exploit the similarities of successive frames. The Wyner-Ziv Theorem on source coding with side information available only at the decoder suggests that an asymmetric video codec, where individual frames are encoded separately, but decoded conditionally (given temporally adjacent frames) could achieve similar efficiency. We report first results on a Wyner-Ziv coding scheme for motion video that uses intraframe encoding, but interframe decoding.

1: Introduction

Current video compression standards perform interframe predictive coding to exploit the similarities among successive frames. Since predictive coding makes use of motion estimation, the video encoder is typically 5 to 10 times more complex than the decoder. This asymmetry in complexity is desirable for broadcasting or for streaming video-on-demand systems where video is compressed once and decoded many times. However, some future systems may require the dual scenario. For example, we may be interested in compression for mobile wireless cameras uploading video to a fixed base station. Compression must be implemented at the camera where memory and computation are scarce. For this type of system what we desire is a low-complexity encoder, possibly at the expense of a high-complexity decoder, that nevertheless compresses efficiently.

To achieve low-complexity encoding, we propose an asymmetric video compression scheme where individual frames are encoded independently (*intraframe encoding*) but decoded conditionally (*interframe decoding*). Two results from information theory suggest that an intraframe encoder - interframe decoder system can come close to the efficiency of an interframe encoder-decoder system. Consider two statistically dependent discrete signals, X and Y, which are compressed using two independent encoders but are decoded by a joint decoder. The Slepian-Wolf Theorem on distributed source coding states that even if the encoders are independent, the achievable rate region for probability of decoding error to approach zero is $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$ and $R_x + R_y \geq H(X,Y)$ [1]. The counterpart of this theorem for lossy source coding is Wyner and Ziv's work on source coding with side information [2]. Let X and Y be statistically dependent Gaussian random processes, and let Y be known as side information for encoding X. Wyner and Ziv showed that the conditional Rate-Mean Squared Error Distortion function for X is the same whether the side information Y is available only at the decoder, or both at the encoder and the decoder. We refer to lossless distributed source coding as Slepian-Wolf coding and lossy source coding with side information at the decoder as Wyner-Ziv coding.

It has only been recently that practical coding techniques for Slepian-Wolf and Wyner-Ziv coding have been studied. Pradhan and Ramchandran presented a practical framework based on sending the syndrome of the codeword coset to compress the source [3]. Since then similar concepts have been extended to more advanced channel codes. Garcia-Frias and Zhao [4][5], Bajcsy and Mitran [6][7] and Aaron and Girod [8] showed that using turbo codes for compression can come close to the Slepian-Wolf bound. Liveris et al. argued that low-density parity-check codes, another form of iterative channel coding, are also suitable for this problem [9].

In spite of the recent developments in schemes for Wyner-Ziv coding, examples where the codes have been used for practical compression applications are few and limited. Pradhan and Ramchandran applied their syndrome idea to a system where a digital stream provides enhancement to a noisy analog image transmission [10]. The digital stream contains the syndromes representing the codewords of the wavelet coefficients for the original image, and the syndromes are decoded using the analog signal as side information. Similarly, Liveris et al. used turbo codes to encode the pixels of an image which has a noisy version at the decoder [11]. Although the work in [10][11] apply Wyner-Ziv coding to natural images, they define the rela-

This work has been supported in part by a gift from Ericsson, Sweden and a Cisco Systems Stanford Graduate Fellowship.

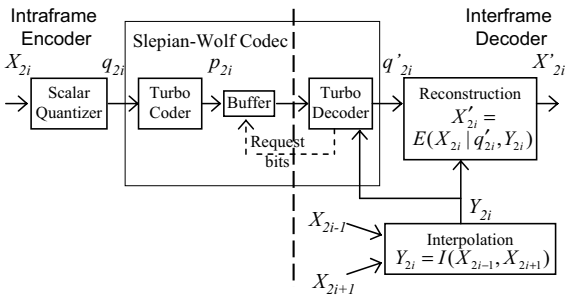

Fig. 1. Wyner-Ziv video codec with intraframe encoding and interframe decoding.

tionship between X and Y to be a simple additive Gaussian noise term, that is, $Y = X + N$ where N is Gaussian independent of X. The problem of using Wyner-Ziv coding for more general statistical models in practical compression scenarios is still open.

In [12], Jagmohan et al. discuss how a predictive coding scheme with multiple predictors can be seen as a Wyner-Ziv problem and thus can be solved using coset codes. Specifically, they suggest that Wyner-Ziv codes could be used to prevent prediction mismatch or drift in video systems but do not present an actual implementation.

In this paper we apply Wyner-Ziv coding to a real-world video signal. We take X as the even frames and Y as the odd frames of the sequence. X is compressed by an intraframe encoder that does not know Y. The compressed stream is sent to a decoder which uses Y as side information to conditionally decode X. Note that we do not force a given correlation between X and Y but instead use the inherent temporal similarities between adjacent frames of a video sequence.

In Section 2, we describe in detail the building blocks of our Wyner-Ziv video codec. In Section 3, we compare the performance of the proposed coder to conventional intraframe coding and to conventional interframe coding, using a standard H263+ video coder.

2: Wyner-Ziv Video Codec

We propose an intraframe encoder and interframe decoder system as shown in Fig. 1. Its basic structure is composed of an inner turbo code-based Slepian-Wolf codec and an outer quantization-reconstruction pair. Both the Slepian-Wolf decoder and the reconstruction block make use of the side information available at the decoder.

Let $X_1, X_2, ..., X_N$ be the frames of a video sequence. The odd frames, X_{2i+1}, where $i \in \{0, 1, ..., \frac{N-1}{2}\}$, are the key frames which are available as side information at the decoder. To simplify the problem, we do not consider the compression of the key frames and assume they are known perfectly at the decoder. Each even frame, X_{2i}, is encoded independent of the key frames and the other even frames.

X_{2i} is encoded as follows: First, we scan the frame row by row and quantize each pixel value using 2^M levels to produce the quantized symbol stream q_{2i}. The symbols are fed into the two constituent convolutional encoders of a turbo encoder. Before passing the symbols to the second convolutional encoder, interleaving is performed on the symbol level. The parity bits, p_{2i}, produced by the turbo encoder are stored in a buffer. The buffer transmits a subset of these parity bits to the decoder upon request.

For each frame X_{2i}, the decoder takes the adjacent key frames $X_{2i-1}$and $X_{2i+1}$and performs temporal interpolation $Y_{2i} = I(X_{2i-1}, X_{2i+1})$. The turbo decoder uses the side information Y_{2i} and the received subset of p_{2i} to form the decoded symbol stream q'_{2i}. If the decoder cannot reliably decode the symbols, it requests additional parity bits from the encoder buffer. The request and decode process is repeated until an acceptable probability of symbol error is guaranteed. By using the side information, the decoder needs to request $k \leq M$ bits to decode which of the 2^M bins a pixel belongs to and so compression is achieved. After the receiver decodes q'_{2i} it calculates a reconstruction of the frame X'_{2i} where $X'_{2i} = E(X_{2i}|q'_{2i}, Y_{2i})$.

The main blocks of the Wyner-Ziv video codec are discussed in more detail below.

2.1: Quantization

We use a uniform scalar quantizer with 2^M levels to quantize the pixels of X_{2i}. Each quantizer bin is assigned a unique symbol. For a given frame, we take the symbols and form a block of length L which is then fed into the Slepian-Wolf coder. Unlike the work in [11] which performs modulo encoding before turbo encoding, we do not form cosets in the quantizer domain. In our system the task of grouping the codewords into cosets is left to the turbo coder which operates in a space of much higher dimensionality.

2.2: RCPT-based Slepian-Wolf Coder

For the Slepian-Wolf coding of the quantized symbol stream q_{2i}, we have implemented a turbo encoder-decoder system. The turbo encoder assigns a specific sequence of parity bits to the given input block of symbols. A block of symbols can be seen as a long codeword and the blocks that are assigned the same parity sequence belong to the same coset.

We implement a *rate compatible punctured turbo code* (RCPT) for bit rate flexibility. The RCPT structure is borrowed from channel coding where it is used for handling varying channel statistics [13]. In the case of Slepian-Wolf coding, the rate flexibility of the RCPT helps in adapting to the changing statistics between the side information and the frame to be encoded. With this structure we can ensure that

the encoder only sends the minimum number of parity bits required for the receiver to correctly decode q_{2i}.

To make use of rate compatibility, we employ feedback in our system. The decoder requests parity bits until it can correctly decode the sequence. In terms of coset coding, if the decoder cannot disambiguate which codeword of the coset the current stream belongs to, it requests for more parity bits. The additional bits decrease the number of codewords which map to a given coset and make it easier for the decoder to distinguish the current codeword.

Since we desire a simple encoder, making the decoder control the bit allocation through feedback is reasonable. This way the encoder does not need to perform any statistical estimation which is necessary for proper rate control.

2.3: Side Information and Statistical Model

In generating the side information Y_{2i}, we take the two adjacent key frames and perform temporal interpolation to get an estimate of X_{2i}.

The first interpolation technique we used was simply averaging the pixel values at the same location from the two key frames. Let $X_{2i-1,j}$ and $X_{2i+1,j}$ be the pixel values at location j from the key frames. We calculate $Y_{2i,j}$ as $\frac{1}{2}(X_{2i-1,j} + X_{2i+1,j})$. We refer to this as *Average Interpolation*.

More sophisticated techniques based on motion compensated (MC) interpolation can be used to generate side information at the decoder. Another scheme we implemented was block-based motion compensated interpolation based on symmetric motion vectors (*SMV Interpolation*). We assume that given a block in X_{2i}, the motion vector of this block from time $2i-1$ to $2i$ is the same as the motion vector from time $2i$ to $2i+1$. We perform block matching to find the best symmetric motion vector for a given block and then take the average of the motion compensated blocks from the two adjacent frames.

In our proposed system the decoder is free to implement any form of interpolation, from the simplest scheme of using the previous frame as side information to more complex schemes such as motion compensated interpolation with dense motion vector fields, intelligent segmentation, or multiple frame predictors. Unlike conventional video compression systems where the encoder sends the motion information, in our system the complexity of the motion compensation at the decoder does not increase the encoding rate. In fact, the better the interpolation, the less bits requested by the decoder. What is interesting with this structure is that we can improve the compression performance by only improving the decoder. Assume we deploy a system of cheap wireless cameras. We can afterwards improve the compression performance by simply upgrading the interpolation algorithm at the base station. Note that this property is the dual to conventional video compression, where decoders are fixed (usually by a standard), but the encoder has considerable flexibility to trade off smart processing vs. bit-rate.

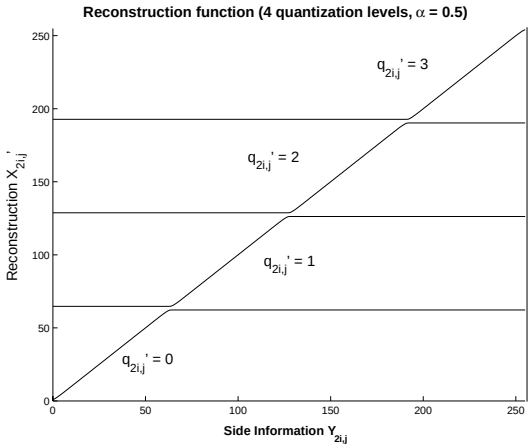

Fig. 2. Sample reconstruction function for a Laplacian model with $\alpha = 0.5$ and number of quantization levels $2^M = 4$.

To make use of the side information, the decoder needs some model for the statistical dependency between X_{2i} and Y_{2i}. The statistical model is necessary for the conditional probability calculations in the turbo decoder as well as for the conditional expectation in the reconstruction block.

For the two interpolation techniques we implemented, we observed that if we take a pixel from the current frame and subtract from it the corresponding side information, the resulting statistics is very close to that of a Laplacian random variable. Given $X_{2i,j}$ and the corresponding side information $Y_{2i,j}$, the distribution of the residual can be approximated as $f(X_{2i,j} - Y_{2i,j}) = \frac{\alpha}{2} e^{-\alpha |X_{2i,j} - Y_{2i,j}|}$. The parameter α can be estimated at the decoder using the residual between the key frame X_{2i-1} and Y_{2i}. We observed that using an estimated α instead of the value with the closest fit to the data does not significantly degrade the system performance.

2.4: Reconstruction Function

Given the decoded symbols and the side information we can calculate the reconstruction for each pixel as $X'_{2i,j} = E(X_{2i,j} | q'_{2i,j}, Y_{2i,j})$. Note that we are reconstructing the pixels independent of the adjacent pixels so the spatial correlation is not being exploited. A plot of the reconstruction function for a Laplacian model with $\alpha = 0.5$ and number of quantization levels $2^M = 4$ is shown in Fig. 2. As it can be seen from the plot, if the side information $Y_{2i,j}$ is within the reconstructed bin $q'_{2i,j}$, then $X'_{2i,j}$ takes the value of $Y_{2i,j}$. If $Y_{2i,j}$ is outside the bin, the function clips the reconstruction towards the boundary of the bin closest to $Y_{2i,j}$.

This kind of reconstruction function has the advantage of limiting the magnitude of the reconstruction distortion to a maximum value, determined by the quantizer coarseness. Perceptually, this property is desirable since it eliminates the large positive or negative errors which may be very annoying to the viewer.

When the side information is not close to the original signal (i.e. high motion frames, occlusions), the side information will not lie within the reconstructed bin. In these cases, the reconstruction scheme can only rely on the quantized symbol for reconstruction and quantizes towards the bin boundary. Since the quantization is coarse, this could lead to contouring which is visually unpleasing. To remedy this we perform subtractive dithering by shifting the quantizer partitions for every pixel using a pseudo-random pattern. This leads to better subjective quality in the reconstruction.

3: Results

We implemented the proposed system and assessed the performance on sample QCIF video sequences.

To change the rate (and correspondingly, the distortion) we varied the number of quantization levels, where $2^M \in \{2, 4, 16\}$. For every even frame of the sequence, we gathered the quantized symbols to form an input block of length $L = 144\text{x}176 = 25344$.

The turbo encoder was composed of two constituent convolutional encoders of rate $\frac{4}{5}$, identical to those used in [8]. To achieve the rate compatibility for the turbo code, we devised an embedded puncturing scheme, with a puncturing pattern period of 8 parity bits. The simulation set-up assumed ideal error detection at the decoder - we assumed that the decoder can determine whether the current symbol error rate, P_e, is greater than or less than 10^{-3}. If $P_e \geq 10^{-3}$ it requests for additional parity bits. In practical systems, the error rate can be estimated by jointly observing the statistics of the decoded stream and the convergence of the turbo decoder. We implemented Average and SMV interpolation.

We compared the Rate-PSNR performance of our system to three set-ups of the H263+:

1. Intraframe (*I-I-I-I*) - The even frames are intracoded as *I* frames.
2. Interframe, no motion compensation (*I-B-I-B, No MC*)- The even frames are encoded as *B* frames (predicted from the previous and next frame) but the motion vectors are set to zero. For fair comparison, we assume that the key frames are perfectly reconstructed at the decoder.
3. Interframe, with motion compensation (*I-B-I-B*) - Same as Set-up 2 but we allow motion compensation.

The results for the *Carphone* and *Foreman* QCIF sequences are shown in Fig. 3 and Fig. 4. For the plots, we only count the rate and distortion of the luminance of the even frames and consider the even frame rate as 15 frames per second. The zero rate point in our scheme corresponds to using the interpolated frame as the decoded frame.

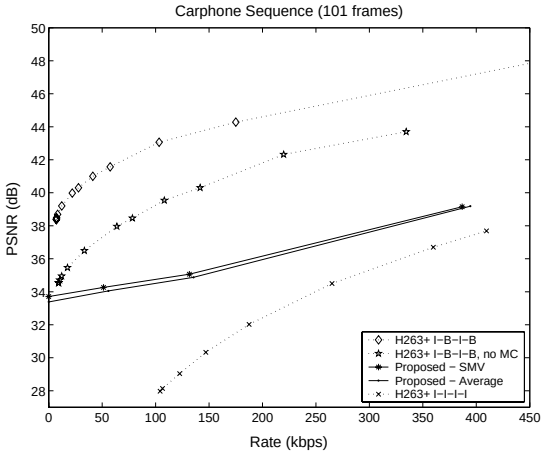

Fig. 3. Rate vs. PSNR for *Carphone* Sequence.

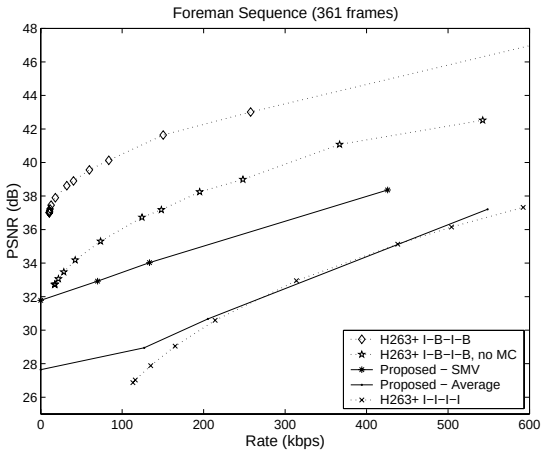

Fig. 4. Rate vs. PSNR for *Foreman* Sequence.

As it can be seen from Fig. 3, the interpolation scheme does not significantly change the performance for the *Carphone* sequence. This is due to the fact that most of the new information in the sequence is caused by the changing scenery in the car window and not by high motion. On the other hand, for the *Foreman* sequence in Fig. 4, using SMV interpolation gives 3 to 4 dB improvement over Average interpolation. For this sequence, simple averaging was not effective since there was high motion throughout the frame.

With good interpolation at the decoder, our system performs much better than H263+ intraframe coding. For the *Carphone* sequence, the gain compared to H263+ intraframe coding ranged from 2 to 6 dB. For the *Foreman* sequence with SMV interpolation, the gain above intraframe coding was about 4 to 7 dB.

(a) Interpolated frame (b) 16-level encoded

Fig. 5. Interpolated and encoded frame from *Foreman*.

As expected the performance of our Wyner-Ziv codec is below that of H263+ interframe coding. For *Carphone*, the gap from the corresponding interframe plots ranges from 1 to 8 dB, with a smaller gap in lower bit rates. For *Foreman* with SMV, our system performance is about 5 to 7 dB lower than interframe coding. This is partly due to the fact that the H263+ coder exploits both the spatial and temporal redundancy in the signal. For our codec, the spatial correlation has not yet been incorporated into the decoding process.

Even if a sophisticated MC interpolation scheme is implemented, the content of the video may be such that it is difficult to have a good estimate of the current frame from the adjacent frames. In Fig. 5 we see how our coding scheme can fix the MC-interpolation artifacts in cases of occlusions and high motion. The image on the left is the interpolated frame (zero rate case) and the frame on the right is encoded with $2^M = 16$ levels (average bit rate for sequence = 400 kbps). As we can see, the encoding sharpens the image and closely reconstructs the hand even if the interpolation is bad. The dithering of the quantizer also improves the visual quality in the areas where motion compensation fails and coarse quantization dominates. Comparing this sequence to that of H263+ intraframe coding with the same sequence bit rate, we observe that the intraframe decoded sequence has obvious blocking artifacts which are not present in our system.

It is important to note that one artifact introduced by our scheme is the presence of residual errors from the Slepian-Wolf decoder. Visually, this may result in isolated blinking pixels at random locations or clustered error specks in a part of the image where the side information is not reliable. In our simulations we fixed the maximum error to be less than 10^{-3} or about 25 pixels per frame. Determining a visually acceptable error rate is left to future investigation.

4: Conclusion

In this paper we propose a Wyner-Ziv video codec which uses intraframe encoding and interframe decoding. This type of codec is useful for systems which require simple encoders but can handle more complex decoders. The encoder is composed of a scalar quantizer and a rate compatible turbo encoder. The decoder performs turbo decoding using an interpolated frame as side information.

We showed that our proposed scheme performs 2 to 7 dB better than H263+ intraframe encoding and decoding. The scheme has not yet reached the compression efficiency of a H263+ interframe coder but this gap could be reduced in the future by exploiting spatial correlation in the proposed codec.

References

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

[2] D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Inform. Theory*, vol. IT-22, pp. 1–10, Jan. 1976.

[3] S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. DCC '99*, Snowbird, UT, Mar. 1999, pp. 158 –167.

[4] J. Garcia-Frias, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417–419, Oct. 2001.

[5] Y. Zhao and J. Garcia-Frias, "Data compression of correlated non-binary sources using punctured turbo codes," in *Proc. DCC '02*, Snowbird, UT, Apr. 2002, pp. 242–251.

[6] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem using turbo codes," in *Proc. Globecom '01*, San Antonio, TX, Nov. 2001, pp. 1400–1404.

[7] P. Mitran and J. Bajcsy, "Coding for the Wyner-Ziv problem with turbo-like codes," in *Proc. ISIT '02*, Lausanne, Switzerland, July 2002, p. 91.

[8] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. DCC '02*, Snowbird, UT, Apr. 2002, pp. 252 –261.

[9] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, pp. 440–442, Oct. 2002.

[10] S.S. Pradhan and K. Ramchandran, "Enhancing analog image transmission systems using digital side information: a new wavelet based image coding paradigm," in *Proc. DCC '01*, Snowbird, UT, Mar. 2001, pp. 305 –309.

[11] A. Liveris, Z. Xiong, and C. Georghiades, "A distributed source coding technique for correlated images using turbo codes," *IEEE Commun. Lett.*, vol. 6, pp. 379–381, Sept. 2002.

[12] A. Jagmohan, A. Sehgal, and N. Ahuja, "Predictive encoding using coset codes," in *Proc. ICIP '02*, Rochester, NY, Sept. 2002, pp. 29–32.

[13] D. Rowitch and L. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes," *IEEE Trans. on Commun.*, vol. 48, no. 6, pp. 948–959, June 2000.