DISTRIBUTED VIDEO CODING USING COMPRESSIVE SAMPLING

Josep Prades-Nebot[a], *Yi Ma*[b], *and Thomas Huang*[b]

[a]GTS
Universidad Politécnica de Valencia
Valencia, Spain
jprades@dcom.upv.es

[b]IFP
University of Illinois at Urbana-Champaign
Urbana, USA
{yima, huang}@uiuc.edu

ABSTRACT

In this paper, we propose a new Distributed Video Coding algorithm based on Compressive Sampling principles. The encoding algorithm transmits a set of measurements of every frame block. Using these measurements, the decoder finds an approximation of each block as a linear combination of a small number of blocks in previously transmitted frames. Experimental results show that the coding efficiency of our algorithm is close to the efficiency of Distributed Video coders based on Wyner-Ziv techniques.

Index Terms— Compressive sampling, Wyner-Ziv video coding, distributed video coding, sparse representations.

1. INTRODUCTION

Distributed Video Coding (DVC) is a new coding paradigm for those applications where the coding resources are more limited at the encoder than at the decoder [1]. Hence, in DVC, encoders are much less complex systems than their correspondent decoders. This complexity distribution is achieved by performing intra-frame encoding and inter-frame decoding.

The most popular DVC technique is Wyner-Ziv Video (WZV) coding. In WZV coding, frames are intra-frame encoded but are conditionally decoded using side information (SI). The decoder obtains the SI of a frame by extrapolating or interpolating previously decoded frames [1]. In each frame, the WZV encoder applies a channel code (usually a turbo code or a LDPC code) to the frame and transmits a portion of the resulting parity bits. The decoder uses the received parity bits and the SI of the frame to perform its decoding.

In this paper, we propose a DVC scheme based on Compressive Sampling (CS) principles. CS is a novel paradigm that allows the recovery of sparse signals from fewer samples or measurements than traditional methods [2,3]. Our proposal of using CS in video coding is motivated by the fact that, generally, the pixels of a block in a video frame can be accurately predicted by using a linear combination of a small number of blocks in other frames (in many cases, a single block suffices). This property justifies the use of motion compensated prediction in most video coders [4]. Hence, frame blocks are sparse signals when they are represented as linear combinations of blocks in other frames, and according to the CS theory, they can be approximately recovered from a small number of block measurements.

Our CS encoder divides each frame into non-overlapping blocks. Then, it obtains, quantizes, and transmits a proper set of measurements of each block. Using the received measurements, the decoder obtains an approximation of the block as a linear combination of blocks from previously transmitted frames. As our algorithm performs intra-frame encoding but inter-frame decoding, it is a DVC technique that can be used in applications that require low complexity encoding.

2. COMPRESSIVE SAMPLING

In this section, we review the basics of the CS of discrete signals [2, 3]. Let $x \in \mathbb{R}^n$ be a discrete signal and let u be its coefficients in some orthonormal basis $\{\psi_i\}, i \in \{1, \ldots, n\}$. Then, $u = \Psi x$ where the *representation functions* ψ_i are the rows of the $n \times n$ matrix Ψ. If only k of the n coefficients are different to zero, then x is said to be k-sparse with respect to Ψ. A k-sparse signal can be efficiently compressed by encoding the position and the values of the non-zero coefficients if $k \ll n$. However, this acquisition-compression process is inefficient because while n signal samples have to be acquired, only a small number k of coefficients are delivered by the encoder system.

CS improves the acquisition-compression process of sparse signals. In CS, instead of encoding the non-zero k coefficients of a k-sparse signal x, we encode the values of $m < n$ *measurements* of x. The vector of measurements $y \in \mathbb{R}^m$ is obtained through

$$y = \Phi\, x$$

where the rows of the $m \times n$ matrix Φ are called *measurement functions* $\{\phi_i\}$. The recovery of the coefficients from

This work has been supported by the Spanish Ministry of Education and Science and the European Comission (FEDER) under grant TEC2005-07751-C02-01, and by the program PAID-00-08 of the Universidad Politécnica de Valencia.

the measurements can be made by searching for the set of coefficients with the minimum ℓ_0 norm that agrees with the measurements:

$$\min \|u\|_0 \quad \text{subject to} \quad y = \Phi\Psi^T u. \tag{1}$$

Unfortunately, this optimization problem is intractable for typical values of n. CS theory establishes that if $m > ck$ where $c > 1$ is an *overmeasuring factor*, the solution to (1) can be found by solving the problem

$$\min \|u\|_1 \quad \text{subject to} \quad y = \Phi\Psi^T u. \tag{2}$$

This problem can be recast as a linear program that can be efficiently solved. The minimum number of measurements necessary to recover the k non-zero coefficients of x depends on k, n, and the degree of *incoherence* between the sets $\{\phi_i\}$ and $\{\psi_i\}$ [2].

In practice, signals of interest are not sparse but *approximately* sparse, i.e., their coefficients are generally different to zero, although only a small number of them have significant amplitude values. Under some conditions, the solution to problem (2) can still recover the most significant coefficients, and hence, provide a good approximation of the signal [2]. Another problem is that, in practice, all the measurements will be corrupted by noise. To deal with noisy measurements, we can solve the problem

$$\min \|u\|_1 \quad \text{subject to} \quad \|y - \Phi\Psi^T u\|_2 \leq \epsilon \tag{3}$$

where ϵ bounds the amount of noise. Problem (3) can be reformulated as

$$\min \left(\lambda\|u\|_1 + \|y - \Phi\Psi^T u\|_2\right) \tag{4}$$

where $\lambda > 0$ trades off measurement fidelity and sparsity. Both ℓ_1 regularization problems (3) and (4) can be efficiently solved.

The application of CS to the source coding of a sparse signal is straightforward [5]. The encoder first obtains the vector y of measurements of the signal to encode x. Then, it quantizes and encodes the measurements generating a bitstream. The decoder first decodes the bitstream and dequantizes the quantization indexes, which provides a reconstructed measurements vector $\hat{y}$. Finally, the decoder performs ℓ_1 regularization using the measurement vector $\hat{y}$. Nevertheless, this CS-based coder has a worse coding efficiency than the direct encoding of the position and the value of the significant coefficients of x [5].

3. DVC BASED ON COMPRESSIVE SAMPLING

The general CS-based source coder proposed in Section 2 can be applied to video compression. As the CS encoder is simple (measurement and quantization) and the CS decoder is complex (ℓ_1 regularization), a CS video coder is a DVC technique that can be used in video applications that require low complexity encoding.

To reduce the number of measurements to be transmitted, a representation matrix Φ that maximize the sparsity of video signals should be chosen. Frame blocks are approximately sparse signals if Φ is built using the basis vectors of the Discrete Cosine Transform (DCT) or a Discrete Wavelet Transform. This property is the basis of the algorithms used to compress frames in intra mode (I-frames). The degree of sparsity can be improved even more if Φ is made from blocks from other frames. In fact, this explains the success of motion compensated prediction in video coding, where a block can be predicted using one block (as in P-frames), or using the average of two blocks (as in B-frames) or using a linear combination of an arbitrary number of blocks (as in multihypothesis prediction [4]). In our CS video coder, we will build the Φ of each block by picking those blocks from previously transmitted frames that can be more useful in its recovery. Note that the representation functions used in our CS coder do not constitute a fixed and orthonormal basis as in most CS applications, but rather an adaptive and redundant dictionary of signals.

Differently to Φ, matrix Ψ must be the same in all blocks in order to keep the complexity of the CS encoder low. Additionally, Ψ should have a high degree of incoherence with all the Φ matrices of all the blocks. By using a fixed Ψ whose entries are drawn randomly from a distribution, the incoherence will be high with most Φ matrices, and the complexity of the encoder will be kept low.

Figure 1 shows the block diagram of our CS video coder. In our coder, video frames are organized into K-frames and CS-frames. The K-frames are coded using a conventional intra-frame coder while the CS-frames are coded using CS principles. In the encoding of a CS-frame, the frame is first divided into non-overlapping square blocks of pixels. In each block x, the encoder first obtains a vector y of m measurements by using a matrix Ψ ($y = \Psi x$). Then, the measurements are quantized using a fixed-rate uniform quantizer and the resulting bits are transmitted. To decode a block, the decoder first dequantizes the measurements obtaining a reconstructed measurement vector $\hat{y}$. Then, the decoder builds a matrix Φ whose rows are the vectors of a dictionary. The dictionary contains those decoded blocks from previously decoded frames that lie in a window. The window is a square region centered in the position of x. Using Φ and $\hat{y}$, the decoder recovers an approximation $\hat{x}$ of x. Finally, all the recovered blocks of the frame are put together providing the decoded CS-frame. Note that, similarly to WZV coding, neither the CS encoder nor the CS decoder can compute the decoding error of each block. In our CS coder, the transmitted measurements are used to *estimate* the quality of the decoded video and to make some coding mode decisions.

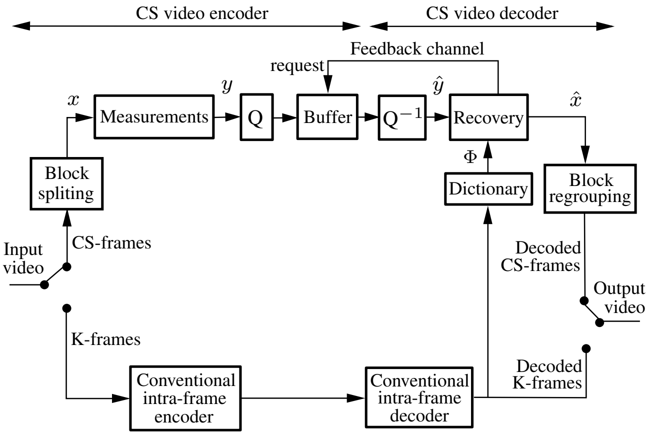


Fig. 1. Block diagram of our CS-based video coder.

To achieve a good rate-distortion performance, the encoder should transmit the minimum number of measurements of each block that guarantees an accurate recovery. Unfortunately, an accurate measurement allocation would highly increase the complexity of the encoder, which is not possible in DVC. Similarly to most DVC algorithms [1, 6], we solve the allocation problem by using a buffer together with a feedback channel (Figure 1). In this setup, the encoder first saves in a buffer the maximum number of quantized measurements of each block that it is able to transmit. Then, for each block, the encoder sends a small group of encoded measurements to the decoder, which then performs recovery. If the decoder estimates that the decoding error is too large, then it requests the transmission of another group of measurements using the feedback channel. This transmission-request process is repeated until the estimated decoding error is small enough. The feedback channel solves the allocation problem but increases the decoding latency and cannot be used in unidirectional or non-real time applications.

There are two situations where blocks are recovered without performing l_1 regularization. If a block x has changed very little with respect to its co-located block in the previous decoded K-frame, our encoder does not transmit any measurement and the recovery is done by copying the co-located block. We refer to this coding mode as SKIP mode. Before transmitting any information, the encoder sends a bit indicating whether or not the block is encoded in this mode. Although deciding if a block must be skipped increases the complexity of the encoder, the use of this coding mode have several advantages. Thus, in skipped blocks, the rate is very small (1 bit/block), the decoding complexity is drastically reduced, and the feedback channel is not used. To decide if a block x must be skipped, the encoder computes the mean absolute difference d_0 between x and the co-located block in the previously decoded K-frame. Then, if d_0 is smaller than a threshold t_0, the block is skipped.

Some blocks are accurately approximated by just copying one block of the dictionary. This speeds up the decoding since the search for the best block is less complex than the ℓ_1 regularization. Since the original block is not available at the decoder, our algorithm uses its measurements to perform the search for the best block. Specifically, if the encoder has decided not to encode a block in SKIP mode, it transmits its first m_1 measurements. Then, the decoder compares the received measurements with the m_1 first measurements of each block in the dictionary and selects the block with the minimum mean square error (MMSE). If the MMSE is below a threshold d_1, then the selected block is considered to be the decoded block, and the decoder informs the encoder not to transmit more measurements for this block. We refer to this coding mode as SINGLE mode.

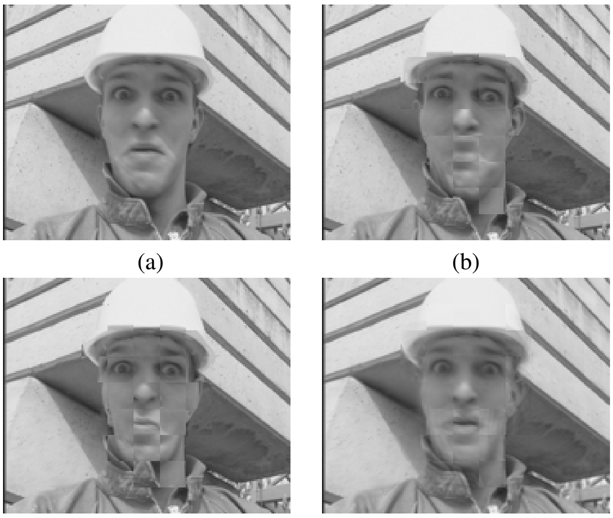
If a block is not encoded in either SKIP or SINGLE mode, then it is encoded in L1 mode. In this mode, the encoder iteratively request groups of measurements through the feedback channel and performs ℓ_1 regularization until the estimated decoded quality is high enough.

4. EXPERIMENTAL RESULTS

In this section, we test the coding efficiency of our CS-based coding algorithm and compare the results with another WZV coding algorithm. We implemented a CS-based video coder with the structure shown in Figure 1. As in [6], in this coder, the odd frames were considered as K-frames and the even frames were considered CS-frames. In our experiments, we assumed K-frames are losslessly available at the decoder. Note this configuration is the same as the one used in [6] where odd frames are encoded using WZV coding instead of CS principles. In our CS encoder, CS-frames were divided into blocks of 16 × 16 pixels. The rows of Φ were samples of an i.i.d. symmetric Bernoulli distribution ($\text{Prob}\{\Phi_{i,j} = \pm 1\} = 1/2$). Therefore, the measurement process only implied pixel additions and subtractions. Each measurement was quantized using a uniform quantizer of $b = 8$ bits. In blocks encoded in L1 mode, the decoder was allowed to make only *one* request of m_2 additional measurements. In the blocks coded in this mode, the ℓ_1 regularization problem in (4) width $m_1 + m_2$ measurements was solved. The dictionary of each block included those blocks in the two closest K-frames that were lying in a square window with a width of $w = 21$ pixels.

Two test sequences with QCIF resolution (176 × 144 pixels/frame) and a frame rate of 30 frames/second were used. In all the encodings, only the luminance component was considered. As in [6], we encoded the first 101 frames (51 K-frames and 50 WZ-frames) of the sequences *Foreman*, and *Mother and Daughter* using our CS coder. To obtain several rate-distortion points, each sequence was encoded using different values of the parameters d_0, d_1, m_1, and m_2. The PSNR values (in dB) and the rate values of the CS-frames were computed and averaged (the rate values were computed considering that the WZ-frame rate was 15 frames/second). The values are shown in Figures 3 and 4. These figures also show the performance of the WZV coding algorithm in [6], with the same sequences and coding setup.

Figure 2 shows three different encodings of a frame of *Foreman*. Figure 2 (b) shows the result of selecting the dictionary block with the MMSE for each frame block. This decoding cannot be done in a real decoder since it does not have access to the original frame blocks. Figure 2 (c) shows the result of encoding all the blocks in SINGLE mode using $m_1 = 15$ measurements. This encoding suffers a loss in quality with respect to the previous one since the MMSE block is not always selected when $m_1 = 15$ measurements are used to perform the search. Note that while the quality in the moving background is good (these blocks have translational motion and, hence, the SINGLE coding mode provides good recoveries), the blocks of the head were poorly decoded. Figure 2 (d) shows the decoded frame using our CS coder with $m_1 = 15$ and $m_2 = 20$. In this frame, most blocks of the head were encoded in L1 mode, which considerably improved the quality in this region.



(a)

(b)

(c)

(d)

Fig. 2. Different decodings of a frameof *Foreman*: (a) original, (b) using the MMSE block (30.3 dB), (c) using SINGLE mode with $m_1 = 15$ (28.9 dB), (d) using our coder with $m_1 = 15$ and $m_2 = 20$ (34.7 dB).

Note in Figure 3 that, in *Foreman*, our algorithm performs much worse than the WZV coding algorithm of [6] at very low rates. This is due to the fact that, in the WZV coder, the starting quality when no parity bit is transmitted (i.e., the quality of the SI) is high. In our CS coder, however, no SI is explicitly available at the decoder, and the decoder only performs well when a large enough rate is used. Our algorithm performs well at low and mid rates in *Mother and Daughter* (Figure 4) because in this sequence, most blocks are encoded in SKIP mode. Note that, in Figures 3 and 4 that, the slope of the rate-distortion function at high rates is smaller in our algorithm than in the WZV coder. This is a consequence of the model-based nature of our coder which limits the maximum video quality that can be achieved.

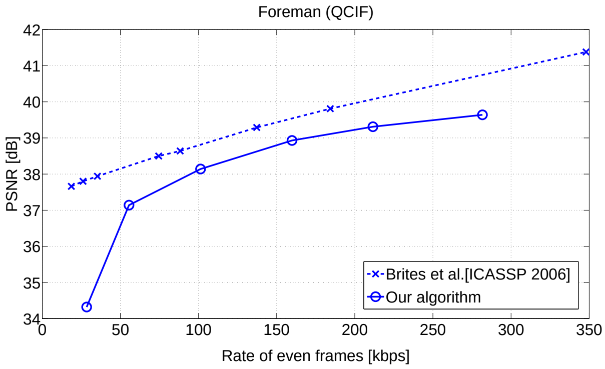


Fig. 3. Rate-distortion performance with *Foreman*

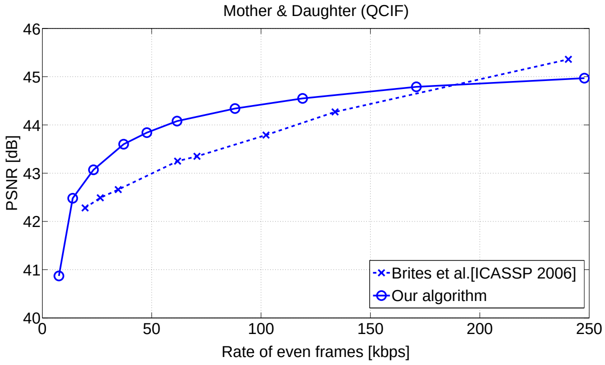


Fig. 4. Rate-distortion performance with *Mother and Daughter*

5. REFERENCES

[1] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[2] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[3] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[4] M. Flierl, Thomas Wiegand, and Bernd Girod, “Rate-constrained multihypothesis prediction for motion compensated video compression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 957–969, Nov. 2002.

[5] V. K. Goyal, A. K. Fletcher, and S. Rangan, “Compressive sampling and lossy compression,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 48–56, March 2008.

[6] C. Brites, J. Ascenso, and F. Pereira, “Improving transform domain Wyner-Ziv video coding performance,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006.