Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering

Basılı/Printed ISSN: 1302-9304. Elektronik/Online ISSN: 2547-958X

Dağıtık Sıkıştırmalı Video Algılama için Basit ve Uyarlanabilir bir Veri Hızı Kontrolü Tasarımı

A Simple and Adaptive Data Rate Control Scheme for Distributed Compressive Video Sensing

Sinem Aslan[1*], E. Turhan Tunalı [2]

[1] Ege Üniversitesi, Uluslararası Bilgisayar Enstitütü, İzmir, TÜRKİYE
[2] İzmir Ekonomi Üniversitesi, Bilgisayar Mühendisliği Bölümü, Balçova, İzmir, TÜRKİYE
Sorumlu Yazar / Corresponding Author *: sinem.aslan@ege.edu.tr

Geliş Tarihi / Received: 05.03.2018 *DOI:10.21205/deufmd.2019216230*
Kabul Tarihi / Accepted: 15.10.2018 *Araştırma Makalesi/Research Article*
Atıf şekli/ How to cite: ASLAN, S., TUNALI, E. T. (2019). A Simple and Adaptive Data Rate Control Scheme for Distributed Compressive Video Sensing. DEUFMD, 21(62), 685-696.

Abstract

Providing fast video encoding, Compressive Sensing has been very suitable for the schemes which require power-constraint devices such as in wireless sensor networks. However, resulted data load on the network is extremely high and accomplishing rate control on these compressive sensing codecs by a simple scheme is crucial. In this paper we propose a simple scheme of data rate control mechanism for an existing compressive sensing codec for distributed video coding which has been reported as an efficient framework in terms of video reconstruction quality. Our approach does not use a feedback channel from decoder which avoids encoder to wait for high-complexity optimization problem. Embedding the proposed scheme to the codec architecture, we have obtained ~43% gain in data rate with an acceptable decrease (6%) in Video PSNR.

Keywords: *Compressive sensing, Distributed video coding, Rate control, Adaptive sampling rate, Video quality.*

Öz

Sıkıştırmalı Algılama, düşük karmaşıklıklı video kodlama imkanı sunduğu için, kablosuz duyarga ağları gibi kaynak-kısıtlı cihazlar gerektiren ortamlar için oldukça uygundur. Ancak, ağ üzerinde oluşan veri yükü geleneksel video kodlama yöntemleri ile kıyaslandığında oldukça fazladır. Dolayısıyla, sıkıştırmalı algılamalı kodlayıcı-kodçözücüler için düşük hesaplama karmaşıklıklı basit bir tasarım ile veri hızı kontrolünün gerçekleştirilmesi önemlidir. Bu makalede, literatürde dağıtık video kodlama uygulamaları için önerilmiş olan video geriçatım kalitesi anlamında etkin bir sıkıştırmalı algılamalı kodlayıcı-kodçözücü kullanılarak düşük karmaşıklı bir veri hızı kontrol mekanizması önerilmiştir. Önerilen tasarım geri bildirim kanalı kullanmamakta ve dolayısıyla kodlayıcının yüksek karmaşıklıklı eniyileme problemi çözümünü beklemesi gerekmemektedir. Önerilen tasarım ile kabul edilebilir ölçüde (6%) Video PSNR kalite kaybına karşılık dikkate çeker ölçüde (~43%) veri hızı kazanımı sağlandığı gösterilmiştir.

Anahtar Kelimeler: *Sıkıştırmalı algılama, Dağıtık video kodlama, Veri hızı kontrolü, Uyarlanabilir veri hızı, Video geriçatım kalitesi*

1. Introduction

The primary concern of video coding is compressing raw video data into relatively small sizes at the encoder side while maintaining acceptable reconstruction quality at the decoder side [1]. The well accepted codecs such as H.26x and MPEG-x use the heavy-encoder and light-decoder architecture which is appropriate for the applications requiring to encode once and decode several times by the end-users. In these codecs, the encoder is responsible for complex implementations such as spatial and temporal correlation analysis and motion compensation and estimation which require a device with high processing power. However, in the last decades, codecs with light-encoder and heavy-decoder architecture are demanded in the applications where resource-constraint devices (sensors) take place, e.g. Wireless Sensor Networks (WSN) [2]. To cope with this lightweight encoder expectation, based on Slepian-Wolf [4] and Wyner-Ziv [5] theorems Distributed Video Coding (DVC) [3] is introduced where complexity is shifted from encoder to decoder.

Another development was the proposal of Compressive Sensing (CS) [6-8] which provides representing signals using only a few non-zero coefficients in an appropriate basis. Such signals are then recovered from a few measurements by solving a sparsity-constraint optimization problem. Established the theory of compressive sensing by the mathematicians in the early 2000s, attracted significant attentions in signal and image processing community [9]. CS has been effectively used in many application areas, examples include but not limited to image denoising, deblurring and inpainting [10], depth-map coding [11], video surveillance [12], foreground extraction [13], radar imaging [14], astronomical data analysis [15], digital watermarking [16], wireless body sensor networks [17], etc. CS is very suitable to be used in DVC architectures since it provides a lightweight encoding scheme, i.e. compressive measurements are acquired by a simple randomly sampling process at the encoder side fastly.

The first DVC architectures employing CS technique are DCVS (Distributed Compressive Video Sensing) [18] and DISCOS (Distributed Compressed Video Sensing) [19]. In the DCVS architecture [18], merely global compressed measurements are computed on the whole region of each input video frame independently, whereas, in the DISCOS architecture [19] both block-based and globally frame-based compressed measurements are computed independently at the encoder side. The decoding is accomplished jointly in the decoder side at both architectures. Between these, DISCOS attracted considerable attention in the community, since the block-based measurements provide more accurate side information at the decoder side thanks to proposed interframe sparsity model and sparsity-constraint block prediction algorithms in [19]. A significant drawback of DISCOS is no rate-distortion analysis has been accomplished which is significant for practical use of the framework.

The literature studies considering Rate-Distortion problem in distributed compressive video sensing follow feedback-based [20-21] or non-feedback-based [22–24] approaches. Methods adopting feedback-based approach require a communication channel from decoder for the encoder to give decisions on the rate control adaptively. Specifically, rate-distortion optimization is accomplished by discarding some amount of coding measurements at the encoder side regarding to some decision metrics that measure redundancy. In [20], coding measurements of a block are discarded when the Mean Square Error (MSE) between that block and its co-located one at the previous key frame is below a threshold. In [21], target measurement rate of a particular block is determined regarding to degree of sparsity of the reconstruction coefficients of its co-located block on the previously reconstructed frame at the decoder side, i.e., the sparser reconstruction coefficient vector involves lower sampling rate, and vice versa. The reported PSNR results in [20] and [21] are far below the reported PSNR results obtained by DISCOS [19] for the same video data. One reason is that only local block-based information is processed in [20] and [21] whereas [19] employed frame-based measurements as well.

Approaches that are not using feedback accomplish the whole control mechanism at the encoder side by employing some heuristics [22–23]. The advantage of this approach is the encoder does not necessarily wait for decoder's feedback where time-consuming optimization problems are solved at. In [22], the sampling rate to acquire block-based measurements is adjusted according to a deficiency factor, i.e higher degradation yields higher sampling rate and vice versa. Deficiency factor is computed from the reconstructed compressively sensed edge image of the reference frames at the encoder side in [22] which results with high complexity at the encoder side that is not desired for DVC schemes. In [23], Mean Absolute Difference (MAD) of CS measurements for consecutive spatial blocks is computed and the CS measurement for the current block is reused if MAD value is small enough. It is reported that compression rate is improved 20%, but because of the temporal correlations of frames were not exploited, the PSNR results are not good enough compared to DISCOS framework [19]. In [24], adaptive sampling rates are applied by considering normalized l1-norm of residual measurements of consecutive temporal blocks. Rate gain is acceptable but PSNR gain was still marginal compared to DISCOS [19]. Differently from the studies in [20–24], DISCOS benefits from both global frame-based and local-block-based information for an advanced video reconstruction. Although DISCOS has been appealing for DVC in terms of video reconstruction quality, no rate-distortion analysis has been explored for this scheme.

In this paper, we developed a simple, and low-cost rate control strategy for DISCOS framework. We propose to benefit from Euclidean distance between key-frames to infer the degree of temporal sparsity for compressed sensed frames. Lower euclidean distance between blocks of key-frames demonstrates that high temporal redudancy exists, so blocks of CS-frames between these key frames are encoded by lower sampling rates. In order to determine the threshold values of Euclidean distances, we run Equal Frequency Binning (EFB) technique on the training set of key frames. Our approach does not use a feedback channel and does not require the encoder to wait for solving high-complexity optimization problem.

In Section 2, we present preliminaries on compressive sensing, components of the DISCOS framework and we present rate-distortion analysis of the DISCOS framework. In Section 3 we carry out a sensitivity analysis that explores performance of DISCOS under variety of reduced sampling rate scenarios and proposed adaptive sampling rate policy. Experimental works are presented in Section 4 and concluding remarks are made in Section 5.

2. Preliminaries

2.1. Compressive Sensing

Compressive sensing theory [6–8] asserts that certain signals or images can be recovered from much fewer measurements than the traditional methods, which follow Shannon sampling theorem [25]. There are two conditions that make the recovery of a signal $x \in \mathbb{R}^N$ possible according to CS theory:

i. It must be possible to represent the signal $x \in \mathbb{R}^N$ sparsely under some sparsifying basis $\Psi \in \mathbb{R}^{N\times N}$ as $x = \Psi\alpha$ where $\alpha \in \mathbb{R}^N$ is named as reconstruction coefficients vector having K nonzero elements ($K << N$);

ii. The sensing matrix, $\Phi \in \mathbb{R}^{M\times N}$, that is used to take M measurements linearly from the signal x must be incoherent to the sparsifying basis Ψ.

Hence, at the encoder side of CS framework, the input signal x is linearly sensed by taking M measurements ($K < M << N$) as in Equation (1) where $A = \Phi\Psi$ and $y \in \mathbb{R}^M$ is named as measurement vector. The sensed measurements or the measurement vector y is then sent to the decoder side [6–8].

$$y = \Phi\Psi = \Phi\Psi\alpha = A\alpha \qquad (1)$$

At the decoder side, the l$_1$-minimization optimization problem in Equation (2) is solved to estimate the sparse reconstruction coefficients vector $\hat{\alpha}$. Finally, the signal $\hat{x}$ is recovered by $\hat{x} = \Psi\hat{\alpha}$ [6–8].

$$\begin{aligned}\hat{\alpha} = \operatorname{argmin} \left|\left|\alpha\right|\right|_{l_1} \\ \text{subject to } A\alpha = y\end{aligned} \qquad (2)$$

2.2. DISCOS Framework

Proposed by [19] and visualized in Figure 1, DISCOS framework is one of the first architectures employing compressive sensing for distributed video coding. The components of the architecture are briefly explained in this section.

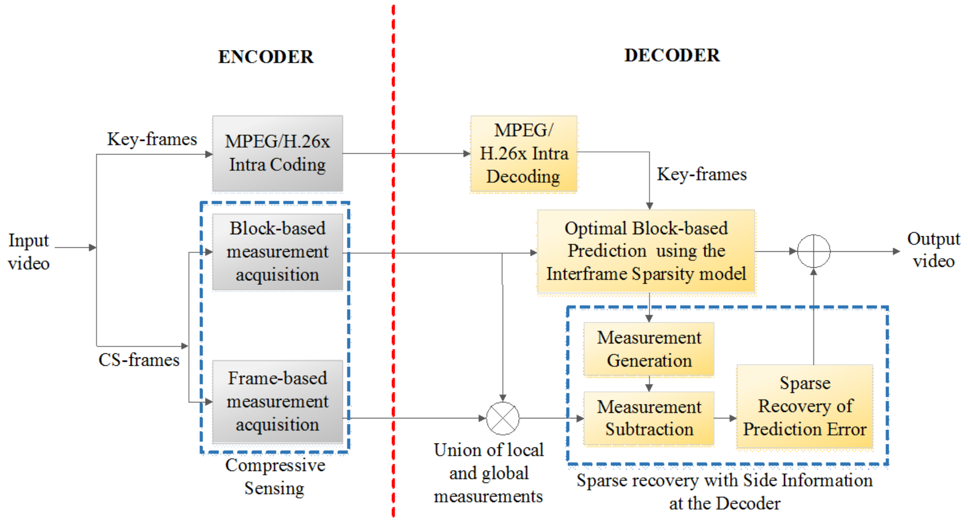
ENCODER
DECODER
Key-frames
MPEG/H.26x Intra Coding
MPEG/ H.26x Intra Decoding
Key-frames
Input video
Block-based measurement acquisition
Optimal Block-based Prediction using the Interframe Sparsity model
Output video
CS-frames
Measurement Generation
Frame-based measurement acquisition
Measurement Subtraction
Sparse Recovery of Prediction Error
Compressive Sensing
Union of local and global measurements
Sparse recovery with Side Information at the Decoder

Figure 1. DISCOS framework [19]

Encoder. Given an input video, i.e., sequence of frames, they are categorised into two categories, *key frames* and *CS-frames* at the encoder. Key frames are intra-coded by a conventional compression standard, i.e., MPEG/H.26x, and compressive sensing is applied to encode the CS-frames. A key frame is sent periodically after a certain number of CS-frames that form a structure similar to well-known Group-of-Pictures (GOP) [29].

Two type of measurements, *frame-based* and *block-based*, are computed for each CS-frame. The frame-based measurement vector, i.e., $y_F \in \mathbb{R}^{M_F}$ where M_F denotes the number of frame-based measurements, is computed by projecting the frame region x_F on a sensing basis Φ_F as in Equation (3). *Structurally Random Matrices* (SRMs) [30] is used as the sensing matrix to acquire frame-based measurements at DISCOS.

$$y_F = \Phi_F \mathrm{x}_F \qquad (3)$$

The block-based measurement vector for each non-overlapping block *i* of the frame is computed by projecting block region on a sensing basis Φ_B as in Equation (4). $y_B^i \in \mathbb{R}^{M_b}$ denotes the measurement vector of block *i*, M_b denotes number of block-based measurements obtained for a block, and n_B denotes the number of non-overlapping blocks that a frame includes in Equation (4). Orthonormal random matrix is used as the sensing matrix to acquire block-based measurements at DISCOS.

$$y_B^i = \Phi_B x_B^i, \quad i = 1, \ldots, n_B \qquad (4)$$

Decoder. Key-frames are decoded by the conventional compression standards at the decoder side. At the decoding stage of a CS frame, non-overlapping blocks are recovered by estimating the sparse reconstruction coefficients, i.e. $\hat{\alpha}_B^i$, $i=1, \ldots, n_B$, by solving the l_1 minimization problem in Equation (5).

$$\hat{\alpha}_B^i = \text{argmin} \left\|\alpha_B^i\right\|_{l_1} \quad \text{subject to } y_B^i = \Phi_B D_i \alpha_B^i \qquad (5)$$

Exploiting the preceding and following keyframes that were decoded, a model named as *Interframe Sparsity Model* is proposed in [19] in order to recover block-based CS measurements (illustrated in Figure 2). Interframe Sparsity Model employs the temporal sparsity of the blocks by referring to the assumption that a block can be sparsely represented by a linear combination of the temporal neighbour blocks within a correlated spatial region on the preceding and following key frames. Thus, the columns of the sparsifiying matrix D_i are constructed by the vectorized overlapping blocks inside a specified area of a temporal neighbour frame of block *i*. *Sparsity Adaptive*

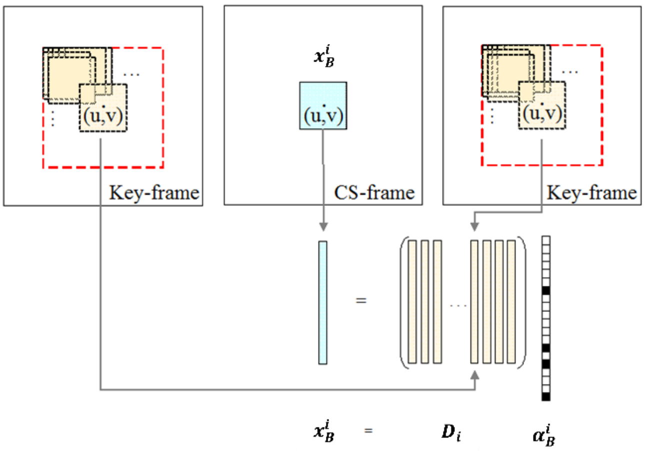

Figure 2. *Interframe Sparsity Model* that is used to estimate block *i*, denoted by x_B^i on a CS-frame by using temporal neighbour blocks on the preceding and following keyframes at the DISCOS decoder. (u, v) denotes the location of the target block, red square denotes the correlated area in keyframes. Image credits to [19].

Matching Pursuit (SAMP) algorithm in [28] is used to solve the optimization problem that is given in Equation (5).

Once the $\hat{\alpha}_B^i$ reconstruction coefficients are estimated, the block $\hat{x}_B^i$ recovered by $\hat{x}_B^i = D_i \hat{\alpha}_B^i$. After recovering each non-overlapping block $\hat{x}_B^i$, *i* = *1*, ..., *nB*, the frame region, i.e., $\hat{x}_F^i$, is constructed. Then, prediction of frame-based measurements is computed by $\hat{y}_F = \Phi_F \hat{x}_F$.

Since the residual between the predicted and received frame-based measurements, i.e., $\hat{y}_F - y_F$, would be sparse, it is recovered by the GPSR algorithm [29]. Finally, the recovered prediction error of intensity values are added on the values of the predicted frame $\hat{x}_F$.

2.3. Rate Distortion Characteristics of DISCOS

Number of block-based (*MB*) and frame-based (*MF*) measurements computed for a CS encoded frame at the DISCOS framework is given in Equations (6) and (7), respectively. In Equation (6), *nB* is the number of non-overlapping blocks that a frame consists of, *N* is the number of pixels in the frame, *rB* is the block-based sampling rate that is the ratio of number of acquired samples to the total possible samples of a block, i.e., the number of pixels of a block., and *rF* is the frame-based sampling rate that is the ratio of number of acquired samples to the total possible samples of the frame, i.e., the number of pixels of the frame.

$$\begin{aligned} M_B &= n_B \times \left(r_B \times \frac{N}{n_B}\right) \\ &= r_B \times N \end{aligned} \qquad (6)$$

$$M_F = r_F \times N \qquad (7)$$

The total number of measurements obtained from both block-based and frame-based measure- ments for a CS-frame is equal to N × (rB + rF) as shown in Equation (8).

$$\begin{aligned} M &= M_B + M_F \\ &= r_B \times N + r_F \times N \\ &= N \times (r_B + r_F) \end{aligned} \qquad (8)$$

It is claimed in [19] that the budget is divided equally between block-based and frame-based measurements, so rF = rB = 0.5. Thus, number of measurements acquired for each CS-frame is equal to the number of pixels that the frame consists of, i.e., *M* = *N*, in DISCOS. This means that the DISCOS encoder may send the frame region without employing any encoding procedure which, in turn, may yield extremely high traffic load.

By applying the default *rF* = *rB* = 0.5 sampling rates to encode the first 37 frames of six 352×288 sized CIF videos by the DISCOS framework [19] we obtain the Video PSNR results presented in Table 1. In [19], the authors computed PSNR for each frame and present the average PSNR over all frames for each video. We prefer to present video PSNR computed over merged frames of the whole sequence rather than average PSNR of individual frames since this is accepted as a more reliable video quality assessment metric [1]. We use the default GOP structure employed in DISCOS, which consists of three compressed sensing frames with an I-frame that is H.264 coded. We worked on the Luma channel and used 30 fps, 8 bit/pixels to compute the data rate for CS-frames. The average and standard deviation of video PSNR values obtained from six videos are presented to make an overall comparison in the following sections. We name the default coding scheme of DISCOS in Table 1 as the *benchmark model*.

Table 1: Benchmark Model ($r_F = r_B = 0.5$)

Video Name	Video PSNR (dB)
Coastguard	30.38
Foreman	35.26
Hall	35.66
Highway	39.16
Mother-Daughter	39.83
News	35.16
Video PSNR (dB) per video	35.91 ± 3.4
Data Rate (kbps) per video	24330

We reduced the block-based and frame-based sampling rates to observe their effect on the data-rate and video PSNR in Table 2. We also present the average PSNR and data rate of six videos in Table 3. Note that the average PSNR values in Table 3 are computed from Table 2.

In Table 2, we see that decrease in r_B has minor effect on Video PSNR for all of the videos, i.e., Coastguard, Foreman, Hall, Highway, Mother-Daughter, and News, whereas decrease in r_F has a more significant effect for all in terms of Video PSNR.

Table 3 demonstrates that reducing r_B by 80% (from 0.5 to 0.1) yields ~4% decrease in average Video PSNR (from 35.91 ± 3.4 to 34.51 ± 2.7), whereas reducing r_F by 80% (from 0.5 to 0.1) yields ~10% decrease in average Video PSNR (from 35.91 ± 3.4 to 26.23 ± 3.1). Thus, it is plausible to apply adaptive sampling strategy to acquisition of merely block-based measurements by fixing the frame-based sampling rate to an acceptable value that provides adequate video quality.

Table 3: Effect of reduction in block-based (r_B) and frame-based (r_F) sampling rates on the video reconstruction quality (video PSNR) and Data rate, both averaged over six videos. Benchmark model is presented in bolded fonts.

r_B	r_F	Avr. Video PSNR (dB)	*Avr. Data Rate* (kbps)
0.5	**0.5**	**35.91 ± 3.4**	**24330**
0.4	0.5	35.80 ± 3.3	21897
0.3	0.5	35.63 ± 3.2	19464
0.2	0.5	35.29 ± 3.0	17031
0.1	0.5	34.51 ± 2.7	14598
0.5	0.4	33.61 ± 3.4	21897
0.5	0.3	31.23 ± 3.5	19464
0.5	0.2	28.77 ± 3.3	17031
0.5	0.1	26.23 ± 3.1	14598

3. Proposed adaptive block-based sampling scheme for DISCOS

According to CS theory, if the sparsity of the signal increases (which means that the number of nonzero elements (K) in the reconstruction vector becomes smaller), then less number of measurements (M) would be sufficient to reconstruct the sensed signal according to the constraint $M \geq cK\ logN$ [6–8]. The proposed *Interframe Sparsity Model* in the DISCOS decoder [19] (that is formalized in Equation (5) and visualized in Figure 2) can ensure to adjust the block-based sampling rate adaptively with respect to the degree of temporal sparsity. In a similar manner, some literature works adaptively adjust the sampling rate of the CS encoder regarding to the measured degree of sparsity of the signal [21–23]. However, at these works it is required to solve an optimization problem previosuly either at the encoder or the decoder side and such a high- complexity operation slows down the speed of the encoder significantly.

In this work, we bypass the reconstruction of the signal to give decisions on sampling rates and we propose to infer the degree of temporal sparsity at the encoder side via simple and low-complexity heuristic and accordingly decide to the block based sampling rates. Thus, in our scheme encoding is performed fastly and with less overhead on the network traffic, since neither the encoder requires implementing a high complexity operation to arrange the sampling rates, nor waits for the decoder's feedback to give decisions which would result with some delay at encoder response and additional overhead on network traffic.

Algorithm 1. Algorithm of Adaptive Block-based Sampling Rate Assignment

```
Input: Coordinates of the center of the macro block ($u$,$v$), preceeding and following key frames I1, I2.
Output: Sampling rate ${r_B}_{u,v}^{I_1-I_2}$

  • Compute Euclidean distance ($d_{u,v}^{I_1-I_2}$) between macro-blocks at the position of ($u$, $v$) of the preceding and following keyframes.
  • Compute sampling rate for the blocks of CS-frames between keyframes I1 and I2 and centered at ($u$,$v$) by following heuristic:
        ${r_B}_{u,v}^{I_1-I_2} = \begin{cases} R1, & \text{if } d_{u,v}^{I_1-I_2} < T1 \\ R2, & \text{elseif } T1 \leq d_{u,v}^{I_1-I_2} < T2 \\ R3, & \text{otherwise} \end{cases}$
Output: ${r_B}_{u,v}^{I_1-I_2}$, the sampling rate for the blocks centered at (u,v).
```

The basis of our scheme is on investigating Euclidean similarity distance between co-located blocks on the successive keyframes. More similar co-blocks on the successive keyframes demonstrate the existence of higher temporal sparsity. Thus, these blocks can be sensed by smaller sampling rates and vice versa. The proposed framework is presented in Figure 3 and we present the implementational details at Algorithm 1. As mentioned in Algorithm 1, we use different sampling rates to sense measurements from a block on a CS-frame regarding to Euclidean distance between its co-located blocks at the preceding and following key-frames I$_1$ and I$_2$. Two threshold values, i.e. *T1* and *T2*, of Euclidean distance are used to decide on the sampling rate (${r_B}_{u,v}^{I_1-I_2}$) for a macro-block of a CS-frame in Algorithm 1 where $d_{u,v}^{I_1-I_2}$ denotes the Euclidean distance between its co-located blocks at the preceding and following key-frames and (u,v) denote the coordinates of the center of the macro-block. One could use a single threshold value, e.g mean or median of the Euclidean distance range, and assign sampling

Table 2. Effect of reduction in block-based (r$_B$) and frame-based (r$_F$) sampling rates on the video reconstruction quality (video PSNR) for six videos.

	Sampling rates		Video PSNR (dB)					
	r_B	r_F	Coastguard	Foreman	Hall	Highway	Mother-Daughter	*News*
Benchmark model (equal budget)	0.5	0.5	30.38	35.26	35.66	39.16	39.83	35.16
Decrease in r_B	0.4	0.5	30.34	35.15	35.58	38.81	39.77	35.17
	0.3	0.5	30.28	35.04	35.49	38.33	39.54	35.07
	0.2	0.5	30.16	34.74	35.25	37.52	39.12	34.92
	0.1	0.5	29.87	33.99	34.56	35.84	38.01	34.81
Decrease in r_F	0.5	0.4	28.74	32.88	32.59	37.84	37.38	32.23
	0.5	0.3	27.23	30.53	29.45	36.06	34.86	29.26
	0.5	0.2	25.78	28.03	26.58	33.68	32.18	26.40
	0.5	0.1	24.26	25.35	23.86	30.84	29.27	23.79

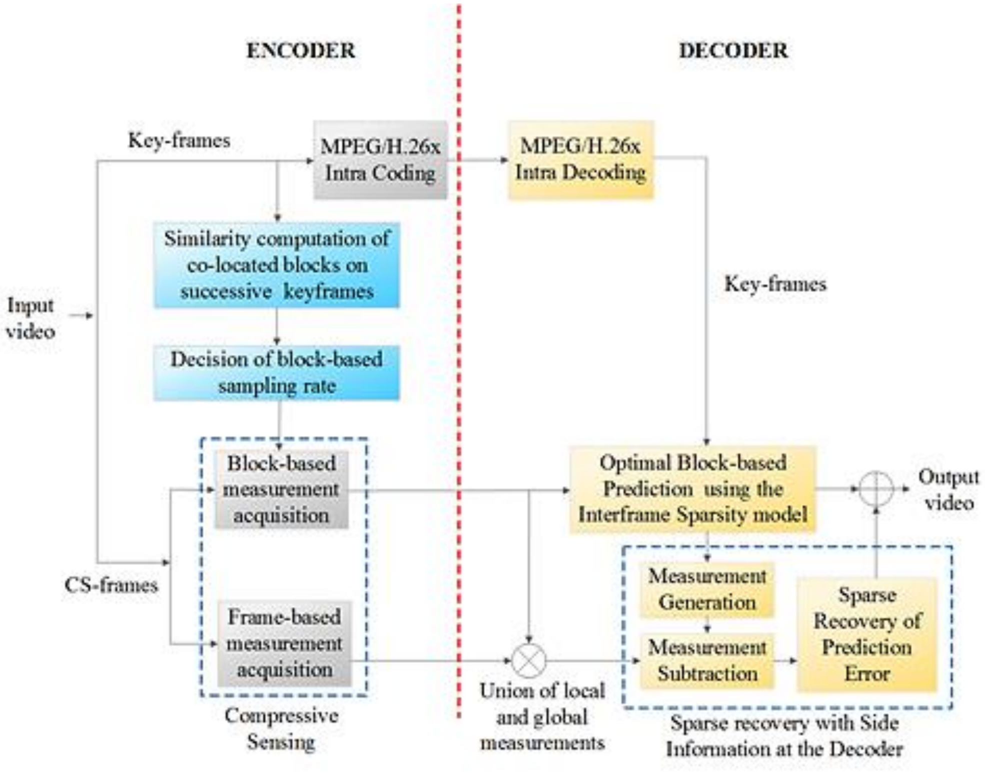
ENCODER
DECODER
Key-frames
MPEG/H.26x Intra Coding
MPEG/H.26x Intra Decoding
Similarity computation of co-located blocks on successive keyframes
Key-frames
Input video
Decision of block-based sampling rate
Block-based measurement acquisition
Optimal Block-based Prediction using the Interframe Sparsity model
Output video
CS-frames
Measurement Generation
Sparse Recovery of Prediction Error
Frame-based measurement acquisition
Measurement Subtraction
Union of local and global measurements
Compressive Sensing
Sparse recovery with Side Information at the Decoder

Figure 3. Proposed scheme for adaptive block-based sampling rate assignment.

rates of two levels. However, that would be coarser decision and we intended to make a finer analysis. We decide on the threshold values of T1 and T2 by getting clues from data with the techniques that will be explained in Section 4.1. While we arrange the block-based sampling rates adaptively as described in Algorithm 1, we use 0.5 for frame based sampling rate, i.e., $r_F = 0.5$ since the video reconstruction quality was too sensitive to changes in frame-based sampling rate.

4. Experiments

4.1. Rate Distortion Characteristics of DISCOS

Various techniques of feature discretization in the literature [30–33] can be used to decide on the threshold values of *T1* and *T2*. We chose the unsupervised methods called *Equal Frequency Binning* (EFB) and *Equal Width Binning* (EWB) described in [30, 31] that do not make use of class membership information which is suitable for our case. EFB computes threshold values so that approximately equal number of samples fall in each threshold interval, while EWB divides the Euclidean distance range into equal intervals. We benefit from training data in order to decide on the technique, i.e., either EFB or EWB, to be used in *T1* and *T2* decision.

We randomly chose five key frames from each of the six video sequences to form the training set, and used the key frames and the CS frames at the remaining GOPs as the testing set. By fixing the block-based sampling rates at *R1* = 0.02, *R2* = 0.06, *R3* = 0.1 in Algorithm 1, we observed the video reconstruction quality obtained by each method, i.e., EFB and EWB, on the testing set. The *T1* and *T2* values determined by EWB were same for all videos, that are *T1* = 5.32 and T2 = 10.65. The *T1* and *T2* values determined by EFB are as

Table 4: T1 and T2 values that are determined by *Equal Frequency Binning* (*EFB*) technique on the training set of each video sequences.

Video Name	T1	T2
Coastguard	8.94	12.95
Foreman	8.70	12.79
Hall	9.10	13.06
Highway	8.48	12.68
Mother-Daughter	9.74	13.78
News	9.74	13.78

in Table 4. The performance of average Video PSNR obtained by *T1* and *T2* determined by EFB was slightly better, i.e., ~0.05, than EWB implementation, thus we continued by the *T1* and *T2* values in Table 4 that are determined by EFB.

4.2. Results

Once we decided on the method to determine *T1* and *T2* values, we investigated the video reconstruction quality and data rate obtained by all sampling rate combinations from the set of r_B = 0.02, 0.04, 0.06, 0.08, 0.1 by brute force search presented in Tables 5 and 6. From Table 3, we observe that decreasing r_B to 0.1 provides 40% gain in data rate with a slightly decrease, i.e., 4%, in Video PSNR. We aimed to provide more data rate gain with an acceptable decrease in Video PSNR, thus we chose 0.1 as the upper bound for r_B in these experiments.

Table 5. Data rate resulted of a variety of block based sampling rate values assigned to the blocks adaptively. The frame-based sampling rate employed in the experiments are r_F = 0.5.

BENCHMARK MODEL

r_B	Data Rate (kbps)						Mean
	Coastguard	Foreman	Hall	Highway	Mother-Daughter	*News*	
0.5	24330	24330	24330	24330	24330	24330	24330

ADAPTIVE STRATEGY

r_B			Data Rate (kbps)						Mean
R1	*R2*	*R3*	Coastguard	Foreman	Hall	Highway	Mother-Daughter	*News*	
0.06	0.08	0.1	14324	14307	14314	14288	14349	14108	14282± 87
0.04	0.08	0.1	14268	14240	14250	14210	14289	13924	14197± 136
0.04	0.06	0.1	14106	14082	14092	14054	14161	13801	14049 ± 127
0.04	**0.06**	**0.08**	**13837**	**13820**	**13827**	**13801**	**13863**	**13621**	**13795 ± 88**
0.02	0.08	0.1	14212	14173	14187	14133	14228	13741	14112 ± 185
0.02	0.06	0.1	14050	14015	14029	13977	14100	13618	13965 ± 175
0.02	0.06	0.08	13782	13753	13764	13724	13802	13438	13711 ± 136
0.02	0.04	0.1	13887	13858	13871	13821	13972	13494	13817 ± 166
0.02	0.04	0.08	13619	13596	13606	13567	13674	13314	13563 ± 127
0.02	0.04	0.06	13351	13334	13340	13314	13376	13135	13308 ± 87

Table 6. Video reconstruction quality obtained by a variety of block based sampling rate values assigned to the blocks adaptively. The frame-based sampling rate employed in the experiments are r_F = 0.5

BENCHMARK MODEL									
r_B			**Data Rate (kbps)**						**Mean**
			Coastguard	Foreman	Hall	Highway	Mother-Daughter	*News*	
0.5			30.38	35.26	35.66	39.16	39.83	35.16	35.91±3.4
ADAPTIVE STRATEGY									
r_B			**Video PSNR (dB)**						**Mean**
R1	*R2*	*R3*	Coastguard	Foreman	Hall	Highway	Mother-Daughter	*News*	
0.06	0.08	0.1	29.77	33.78	34.41	35.32	37.36	34.66	34.21±2.5
0.04	0.08	0.1	29.74	33.55	34.22	34.98	37.43	34.58	34.08±2.5
0.04	0.06	0.1	29.67	33.36	34.07	34.68	37.22	34.57	33.93±2.5
0.04	**0.06**	**0.08**	**29.58**	**33.12**	**33.90**	**34.33**	**36.87**	**34.48**	**33.71±2.4**
0.02	0.08	0.1	29.62	33.33	33.98	34.39	37.07	34.33	33.79±2.4
0.02	0.06	0.1	29.54	33.10	33.81	31.11	36.90	34.28	33.62±2.4
0.02	0.06	0.08	29.45	32.85	33.58	33.74	36.52	34.23	33.40±2.3
0.02	0.04	0.1	29.42	32.83	33.61	33.61	36.34	34.23	33.34±2.3
0.02	0.04	0.08	29.33	32.59	33.37	33.23	35.97	34.17	33.11±2.2
0.02	0.04	0.06	29.21	32.24	33.15	32.81	35.56	34.12	32.85±2.1

We ran the experiments for the parameter settings that are mentioned in Table 5 and 6 on the whole sequences of videos. In Table 5, we present the data rate averaged over CS- frames for videos. The values presented in Table 5 also include the data load of frame-based measurements that is resulted when we use r_F = 0.5. In Table 6 we present the video reconstruction quality obtained with each parameter setting. When we observe the averaged video PSNR quality at Table 6, we see that it changes from 34.21 to 32.85 dB while the benchmark model gives 35.91 dB. So, at the worst case scenario in terms of video reconstruction quality, the PSNR loss of 3db (~10%) is obtained when the setting of R1 = 0.02, R2 = 0.04, and R3 = 0.06 was used. When we check the data rate resulted with the same setting (R1 = 0.02, R2 = 0.04, and R3 = 0.06) of worst case on Table 5, we see that the data rate gain is 11022 kbps (~45%) regarding to the benchmark model that yielded to 24330 kbps. While the Tables 5 and 6 can be used as lookup tables to give decision on the block based sampling rate, we suggest that the setting of R1 = 0.04, R2 = 0.06, and R3 = 0.08, which yields ~2db (~6%) decrease in Video PSNR and 10535 kbps (~43%) gain in data rate would be a good decision.

5. Conclusions

In this paper, we demonstrate the data-rate deficiencies of a distributed compressive video sensing architecture reported as an efficient system in terms of video reconstruction quality. We analysed the rate-distortion characteristics of the current system and then we embedded an adaptive block-based sampling scheme at the encoder in order to obtain data-rate gain with an acceptable video quality. Accomplishing the whole process at the encoder side without decoder's feedback provides fast encoding and avoids extra overhead on the network traffic. The proposed scheme benefits from a simple strategy that is based on inferring sparsity of blocks regarding to Euclidean distances between co-located blocks on the key-frames. We obtained encouraging results, i.e., ~43% gain in data rate with 2 dB decrease in Video PSNR.

References

[1] Richardson I.E. 2011. The H. 264 advanced video compression standard, John Wiley & Sons.

[2] Akyildiz, I.F., Melodia, T., Chowdhury, K.R. 2007. A survey on wireless multimedia sensor networks, Computer networks Vol. 51, no. 4, 921–960.

[3] Girod, B., Aaron, A.M., Rane, S. et al. 2005. Distributed video coding, Proceedings of the IEEE, vol 93, no. 1, 71–83.

[4] Slepian D. and Wolf J. 1973. Noiseless coding of correlated information sources, IEEE Transactions on information Theory vol.19, no.4, 471–480.

[5] Wyner, A. 1974. Recent results in the shannon theory, IEEE Transactions on information Theory, vol. 20, no. 1, 2–10.

[6] Donoho D.L. 2006. Compressed sensing, IEEE Transactions on information theory, vol. 52, no. 4, 1289–1306.

[7] Candes, E.J., Romberg, J. K. and Tao T. 2006. Stable signal recovery from incomplete and inaccurate measurements, Communications on pure and applied mathematics vol.59, no.8, 1207–1223.

[8] Candes E.J. et al. 2006. Compressive sampling, International congress of mathematicians, vol.3, 1433–1452, Madrid, Spain.

[9] Carron, I. 2016. Compressive Sensing: The Big Picture. http://sites.google.com/site/igorcarron2/cs, (Accessed Date: 05.12.2016).

[10] Elad M., Figueiredo M. A., and Ma Y. 2010. On the role of sparse and redundant representations in image processing, Proceedings of the IEEE, vol. 98, no.6, 972–982.

[11] Liu Y., Vijayanagar R. K., and Kim J. 2014. Quad-tree partitioned compressed sensing for depth map coding, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[12] Liu Y., and Pados D.A. 2016. Compressed-Sensed-Domain L 1-PCA Video Surveillance, IEEE Transactions on Multimedia, vol.18, no.3, 351-363.

[13] Pierantozzi M., Liu Y., Pados D.A., Colonnese S., 2016. Video background tracking and foreground extraction via L1-subspace updates, In SPIE Commercial Scientific Sensing and Imaging, pp. 985707-985707.

[14] Potter, L. C., Ertin, E., Parker, J. T. et al. 2010. Sparsity and compressed sensing in radar imaging, Proceedings of the IEEE, vol. 98, no. 6, 1006–1020.

[15] Starck, J.-L. and Bobin, J. 2010. Astronomical data analysis and sparsity: from wavelets to compressed sensing, Proceedings of the IEEE, vol. 98, no.6, 1021–1030.

[16] Zhang X., Qian Z., Ren Y., et al. 2011. Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction, IEEE Transactions on Information Forensics and Security, vol. 6, no. 4, 1223–1232.

[17] Mamaghanian, H., Khaled, N., Atienza D., et al. 2011. Compressed sensing for real-time energyefficient ecg compression on wireless body sensor nodes, IEEE Transactions on Biomedical Engineering, vol. 58, no. 9, 2456–2466.

[18] Kang L.-W. and Lu C.-S. 2009. Distributed compressive video sensing, IEEE International Conference on Acoustics, Speech and Signal Processing, 1169–1172.

[19] Do, T. T., Chen, Y., Nguyen, D. T. et al. 2009. Distributed compressed video sensing, 16th IEEE International Conference on Image Processing (ICIP), 1393–1396.

[20] Prades-Nebot J., Ma Y., and Huang, T. 2009. Distributed video coding using compressive sampling, IEEE Picture Coding Symposium, 1–4.

[21] Chen, H.W., Kang L.W., and Lu, C.S. 2010. Dynamic measurement rate allocation for distributed compressive video sensing, in Visual Communications and Image Processing, 77440I–77440I, International Society for Optics and Photonics.

[22] Azghani, M., Aghagolzadeh, A., and Aghagolzadeh, M. 2010. Compressed video sensing using adaptive sampling rate, 5th IEEE International Symposium on Telecommunications (IST), 710–714.

[23] Wang, Z. and Lee, I., 2010. A study of video coding by reusing compressive sensing measurements. In Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 7th International Conference on (pp. 64-69).

[24] Liu, Z., Zhao, H.V. and Elezzabi, A. Y. 2010. Block-based adaptive compressed sensing for video, IEEE International Conference on Image Processing, 1649–1652.

[25] Unser, M. 2000. Sampling-50 years after shannon, Proceedings of the IEEE, vol. 88, no. 4, 569–587.

[26] Le Gall, D. 1991. Mpeg: A video compression standard for multimedia applications, Communications of the ACM, vol. 34, no. 4, 46–58.

[27] Do, T. T., Tran, T. D. and Gan L. 2008. Fast compressive sampling with structurally random matrices, IEEE International Conference on Acoustics, Speech and Signal Processing, 3369–3372.

[28] Do, T. T., Gan, L., Nguyen, N. et al. 2008. Sparsity adaptive matching pursuit algorithm for practical compressed sensing, tech. rep., DTIC Document.

[29] Nowak, R. D., Wright, S. J. et al. 2007. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, IEEE Journal of selected topics in signal processing, vol. 1, no. 4, 586–597.

[30] Kotsiantis S. and Kanellopoulos D. 2006. Discretization techniques: A recent survey, GESTS International Transactions on Computer Science and Engineering, vol. 32, no. 1, 47–58.

[31] Dougherty J., Kohavi R., Sahami M., et al. 1995. Supervised and unsupervised discretization of continuous features, 12th International Conference on Machine learning, 12, 194–202.

[32] Holte, R. C. 1993. Very simple classification rules perform well on most commonly used datasets, Machine learning, vol. 11, no. 1, 63–90.

[33] Kerber, R. 1992. Chimerge: Discretization of numeric attributes, 10th International conference on Artificial intelligence, 123–128, Aaai Press.