

DISTRIBUTED COMPRESSIVE VIDEO SENSING

Li-Wei Kang and Chun-Shien Lu

Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, ROC

{lwkang, lcs}@iis.sinica.edu.tw

ABSTRACT

Low-complexity video encoding has been applicable to several emerging applications. Recently, distributed video coding (DVC) has been proposed to reduce encoding complexity to the order of that for still image encoding. In addition, compressive sensing (CS) has been applicable to directly capture compressed image data efficiently. In this paper, by integrating the respective characteristics of DVC and CS, a distributed compressive video sensing (DCVS) framework is proposed to simultaneously capture and compress video data, where almost all computation burdens can be shifted to the decoder, resulting in a very low-complexity encoder. At the decoder, compressed video can be efficiently reconstructed using the modified GPSR (gradient projection for sparse reconstruction) algorithm. With the assistance of the proposed initialization and stopping criteria for GRSR, derived from statistical dependencies among successive video frames, our modified GPSR algorithm can terminate faster and reconstruct better video quality. The performance of our DCVS method is demonstrated via simulations to outperform three known CS reconstruction algorithms.

Index Terms—compressive video sensing, (distributed) compressive sampling/sensing, distributed video coding

1. INTRODUCTION

Low-complexity video coding has been potentially applicable for several emerging applications, such as video conferencing with mobile devices and wireless visual sensor networks (WVSN) [1]. Since the low-complexity restriction for a video device, efficient video compression is challenging. Recently, distributed video coding (DVC) [2] based on the principle of distributed source coding (DSC) has been proposed to reduce video encoding complexity to the order of that for still image encoding while preserving a certain coding efficiency. In DVC, the major encoding computation burden can be shifted to the decoder, which is usually allowed to possess powerful computational capability in several real applications (e.g., WVSN).

However, for still image encoding, it is required to capture huge amounts of raw image data first, followed by performing some transformation operator (e.g., discrete wavelet transform, *i.e.*, DWT), which is also computation-intensive [3]. Recently, with the advent of a single-pixel camera [4], compressive sensing (CS) [3]-[8] has been applicable to directly capture compressed image data efficiently. The compressed image can be reconstructed using some CS reconstruction algorithms at the decoder. Similar to DVC, the computation burden can be shifted to the decoder.

However, for compressing huge amounts of video data, it may not be efficient enough to only reduce the encoding complexity or only to individually apply CS to each frame without considering similarities among successive frames. In this paper, by integrating the respective characteristics of DVC and CS, a distributed compressive video sensing (DCVS) framework is proposed to simultaneously capture and compress video data. Almost all

computation burdens can be shifted to the decoder where our modified GPSR (a kind of CS reconstruction algorithm) incorporating with the statistical dependencies among successive frames is exploited to reconstruct video data.

The characteristics of our DCVS includes: **(i) very low-complexity encoder:** only general CS measurement process (described in Sec. 2.3) will be individually applied to each frame; and **(ii) very efficient decoder:** by applying the proposed initialization and stopping criteria for GRSR (described in Sec. 4), the convergence speed and reconstructed video quality using our modified GPSR can be, respectively, faster and better than those using the original GPSR [9], TwIST (two-step iterative shrinkage/thresholding) [10], and OMP (orthogonal matching pursuit) [11].

2. RELATED WORKS

In this section, several related works, including DSC, DVC, CS, compressive image/video sensing, and distributed CS will be reviewed first. Then, the proposed DCVS based on DVC and CS will be addressed in Sec. 3.

2.1. Distributed source coding (DSC)

Assume that W and S are two statistically dependent discrete signals, which are encoded independently but decoded jointly. Slepian-Wolf theorem [2] states the achievable rate region for lossless coding is defined by $R_W \geq H(W|S)$, $R_S \geq H(S|W)$, and $R_W + R_S \geq H(W, S)$, where R_W and R_S are the rates for encoding W and S , respectively, $H(W|S)$ and $H(S|W)$ are the conditional entropies of W and S , respectively, and $H(W, S)$ is the joint entropy of W and S . Then, Wyner-Ziv theorem [2] states DSC with side information (SI) for lossy coding. Assume that S is known as the SI of W . The conditional distortion function for W is unchanged no matter S is available only at the decoder, or both at the encoder and decoder.

2.2. Distributed video coding (DVC)

In DVC [1]-[2], based on Wyner-Ziv theorem, the statistical dependency between a frame W and its SI S is modeled as a virtual correlation channel, where S can be viewed as a noisy version of W . The correlation between W and S can be modeled as a Laplacian distribution as follows:

$$P(W(a, b) - S(a, b)) = \frac{\alpha}{2} e^{-\alpha|W(a, b) - S(a, b)|}, \quad (1)$$

where $W(a, b)$ and $S(a, b)$ are the (a, b) -th pixel in W and S , respectively, and $\alpha \geq 0$ is the model parameter, where $\alpha = \sqrt{2}/\sigma$, and σ is the standard deviation of $(W(a, b) - S(a, b))$. At the encoder, without performing motion estimation, the compression of W can be achieved by transmitting only part of the parity bits (Wyner-Ziv bits) derived from the channel-encoded version of W according to the request from the decoder via the feedback channel. The decoder uses the received Wyner-Ziv bits and the SI S derived from previous decoded frames to perform channel decoding to correct some “errors” in S for the reconstruction of W . In transform-domain DVC, the Wyner-Ziv bits are generated by performing some transformation operator, followed by performing

scalar quantization and channel encoding. Hence, the complexity of the DVC encoder is similar to that of still image encoder consisting of transformation, quantization, and entropy encoding.

2.3. Compressive sensing (CS)

Assume that a sparse basis matrix Ψ with size $N \times N$ can provide a K sparse representation for a real value signal x with length N . That is, x can be represented as $x = \Psi\theta$ and θ with length N can be well approximated using only $K \ll N$ non-zero entries. CS [3]-[8] states that x can be accurately reconstructed by taking only:

$$M = O(K \log(N/K)), \quad (2)$$

where $K < M \ll N$, linear and non-adaptive measurements from:

$$y = \Phi x = \Phi \Psi \theta = A \theta, \quad (3)$$

where y is an $M \times 1$ vector, Φ is an $M \times N$ measurement matrix that is incoherent with Ψ , and $A = \Phi \Psi$. More specifically, the M measurements in y are random linear combinations of the entries of θ , which can be viewed as the compressed and encrypted version of x . Currently, it is unclear that how to efficiently quantize and entropy-encoded the M measurements [5], which will be left for the future research. To reconstruct θ from y , CS is based on solving the convex optimization problem [3]-[10] (e.g., linear programming or GPSR [9]) or some iterative greedy algorithms (e.g., OMP [11]). Finally, x can be reconstructed via $\tilde{x} = \Psi \tilde{\theta}$, where $\tilde{\theta}$ is the reconstructed θ .

2.4. Compressive image/video sensing

In compressive image sensing, if an image x can be sparsely represented using a basis Ψ (e.g., DWT), x can be compressed via the CS technique in Eq. (3) and reconstructed via some CS recovery algorithms [3]-[11]. On the other hand, compressive video sensing has been first proposed in [8], where each video block, at the encoder, is classified to be either sparse or non-sparse via a CS test. Each sparse block is compressed via CS, whereas each non-sparse block is fully sampled.

2.5. Distributed compressive sensing (DCS)

Distributed compressive sensing (DCS) [6] exploits both intra-signal and inter-signal correlation structures. Consider a sensor network scenario, several sensors measure signals that are each individually sparse in a certain basis and also correlated among sensors. In DCS, each signal is independently measured via a CS technique and jointly reconstructed at a collection point collecting measurements from multiple sensors.

3. DISTRIBUTED COMPRESSIVE VIDEO SENSING (DCVS)

In this section, the joint sparsity model of our DCVS is first described in Sec. 3.1 to show the guideline for exploiting the statistical dependencies among successive frames. In our DCVS encoder described in Sec. 3.2, each frame is independently compressed via a CS measurement process. In our DCVS decoder, each frame is jointly reconstructed using our modified GPSR incorporating the proposed initialization derived from DVC side information generation and the proposed stopping criteria derived from statistical dependencies among successive frames, described in Secs. 3.3-3.6. Note that our DCVS, at its current status, is only designed for single-view videos, which can be extended to multiview video scenario, and can be applicable in a WVSN.

3.1. Joint sparsity model

To exploit the correlation among successive frames, similar to [6], the joint sparsity model in our DCVS can be described as follows. Assume two successive frames, x_t and x_{t+1} , in the same scene are visually similar, where t is the time instant. That is, x_t and x_{t+1} should have similar common portion and respective unique portions. Conceptually, the two frames can be expressed as:

$$x_t = x_C + x_{t_U}, \quad (4)$$

$$x_{t+1} = x_C + x_{t+1_U}, \quad (5)$$

where x_C is the similar/common portion between x_t and x_{t+1} , x_{t_U} and x_{t+1_U} are the unique portions of x_t and x_{t+1} , respectively. By treating x_t as a reference frame for x_{t+1} , in conventional video coding, the encoder will perform motion estimation to find the predictor (similar to x_C) for x_{t+1} and encode the difference (similar to x_{t+1_U}) between x_{t+1} and its predictor. Hence, the compression of x_t and x_{t+1} can be achieved by compressing x_t and x_{t+1_U} . Assume a sparse basis matrix Ψ can provide K_t and K_{t+1_U} sparse representations for x_t and x_{t+1_U} , respectively, as:

$$x_t = \Psi \theta_t, \|\theta_t\|_0 = K_t, \quad (6)$$

$$x_{t+1_U} = \Psi \theta_{t+1_U}, \|\theta_{t+1_U}\|_0 = K_{t+1_U}, \quad (7)$$

where θ_t and θ_{t+1_U} are the sparse representations of x_t and x_{t+1_U} , respectively, and $\|\theta\|_0$ is the number of nonzero entries in θ , i.e., ℓ_0 norm of θ . Usually, $K_t \geq K_{t+1_U}$, and based on Eq. (2), $M_t \geq M_{t+1_U}$, where M_t and M_{t+1_U} are the number of measurements of x_t and x_{t+1_U} , respectively. Although, in DCVS, it is impossible to find x_{t+1_U} at the encoder due to low-complexity restriction, it can be confirmed that the number of measurements (M_{t+1}) of x_{t+1} can be smaller than that (M_t) of x_t if the correlation between them can be adequately exploited by treating x_t as a reference frame for x_{t+1} .

3.2. DCVS encoder

In DCVS, a video sequence consists of several GOPs (group of pictures), where a GOP consists of a key frame followed by some non-key frames. Conceptually, each key frame serves as a reference frame for its neighboring non-key frames. At our DCVS encoder shown in Fig. 1, without performing motion estimation, without needing any prior knowledge about correlation among successive frames, and without performing extra tasks (no additional burden) in the CS process described in Sec. 2.3, each frame x_t (key frame or non-key frame) with size N is compressed via the CS measurement process (Eq. (3)) as:

$$y_t = \Phi x_t, \quad (8)$$

where y_t is the measurement vector with size $M_t \times 1$ and Φ is the $M_t \times N$ measurement matrix described later. Based on the joint sparsity model in Sec. 3.1, the measurement rate (MR) of a key frame should be larger than that of a non-key frame. The MR for a frame X_t can be defined as $MR_t = M_t/N$.

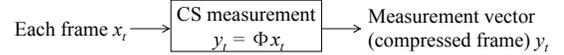


Fig. 1. Our DCVS encoder.

Here, the exploited measurement matrix Φ is the scrambled block Hadamard ensemble (SBHE) matrix [7], which takes the partial block Hadamard transform, followed by randomly permuting its columns. SBHE has been shown to satisfy the five requirements, including near optimal performance, universality, fast computation, memory efficient, and hardware friendly, and outperform several existing ones (e.g., the Gaussian i.i.d matrix and the binary sparse matrix) [7]. The sparse basis matrix Ψ used in this paper is DWT basis.

After performing the CS process, the measurement vector y_t for each frame x_t (the compressed version for x_t) will be transmitted to the decoder. At the decoder, each key frame is reconstructed via the GPSR algorithm [9] while each non-key frame is reconstructed via GPSR incorporating with the statistical dependencies among successive frames, to be addressed in Secs. 3.3-3.6.

3.3. Gradient projection for sparse reconstruction (GPSR)

At the decoder, each key frame $x_t = \Psi \theta_t$ with size N is reconstructed via GPSR [9], which solves the convex unconstrained optimization problem as:

$$\min_{\theta_i} \frac{1}{2} \|y_i - A\theta_i\|_2^2 + \tau \|\theta_i\|_1, \quad (9)$$

where y_i is a $M_i \times 1$ vector, $y_i = \Phi x_i$, $A = \Phi\Psi$ is a $M_i \times N$ matrix, $\|v\|_2$ is the Euclidean norm (ℓ_2 norm) of v , $\|v\|_1$ is the ℓ_1 norm of v , *i.e.*, the sum of the absolute value of each component in v , and τ is a non-negative parameter. GPSR is essentially a gradient projection (GP) algorithm applied to a quadratic programming formulation of Eq. (9), in which the search path for each iteration is obtained by projecting the negative-gradient direction onto the feasible set [9].

Each key frame is reconstructed via GPSR [9] with default settings in the public GPSR code included in the ‘‘Fast CS using SRM’’ tool [7]. In GPSR, the default initial solution for θ_i is a zero vector. The default stopping criterion of GPSR is that when the relative change in the number of nonzero components in θ_i is smaller than a threshold T_A (default $T_A = 0.01$), the algorithm will stop. Finally, the key frame x_i can be reconstructed via $\tilde{x}_i = \Psi\tilde{\theta}_i$, where $\tilde{\theta}_i$ is the final solution obtained by GPSR. Note that the used GPSR included in [7] is an older version. Recently, the latest version called GPSR 5.0 [9] providing a novel default stopping criterion has been released. However, our simulations show that the older version can provide a better tradeoff between reconstructed video quality and reconstruction complexity.

3.4. Side information (SI) generation

In DCVS, each non-key frame is reconstructed via GPSR with the proposed initialization and stopping criteria, derived from the statistical dependencies among successive frames. Before reconstructing a non-key frame x_i , the decoder will generate its SI S_i first, which can be viewed as a noisy version of x_i . Similar to DVC [2], SI can be generated by motion-compensated interpolation from previous reconstructed neighboring key frames. In DCVS, a very efficient frame rate up-conversion tool [12] is exploited to generate the SI for each non-key frame.

3.5. Initialization at DCVS decoder

In our modified GPSR for reconstructing each non-key frame $x_i = \Psi\theta_i$, the initial solution for θ_i is set to be its SI θ_{S_i} as follows: $\tilde{\theta}_i^{(0)} = \theta_{S_i}$, *i.e.*, $\tilde{x}_i^{(0)} = S_i$, where $\tilde{\theta}_i^{(0)}$ is the initial solution (at the 0-th iteration) for θ_i , $S_i = \Psi\theta_{S_i}$, and S_i is the SI of x_i . In the same scene, successive frames should have a certain similarity. Hence, the SI derived from the neighboring key frames for a non-key frame should be similar to this frame, even though the SI may be coarse due to fast motions, poor SI generation, or poor neighboring reconstructed key frames. Based on Sec. 3.1, the measurement rate for a non-key frame is usually set to be smaller than that of a key frame. To get a good reconstructed non-key frame, it is required to have a good initialization, followed by GPSR optimization where proper stopping criteria are required to get optimal or near-optimal solution after a small number of iterations.

3.6. Stopping criteria at DCVS decoder

It is usually difficult to decide when GPSR can stop without incurring excessive computation [9] and with sufficient reconstruction quality. At DCVS decoder, the stopping criteria for GPSR are designed based on the statistical correlation between the current non-key frame and its SI. Consider a non-key frame x_i , its SI S_i , and the reconstructed x_i at the i -th iteration, denoted by $\tilde{x}_i^{(i)}$.

Based on Sec. 2.2, the correlation between x_i and S_i can be modeled as a Laplacian distribution with the parameter $\alpha(x_i, S_i)$. The more similar x_i and S_i are, the larger $\alpha(x_i, S_i)$ is. Similarly, $\tilde{x}_i^{(i)}$ and S_i can be modeled by $\alpha(\tilde{x}_i^{(i)}, S_i)$ while $\tilde{x}_i^{(i)}$ and x_i can be

modeled by $\alpha(\tilde{x}_i^{(i)}, x_i)$. Obviously, if x_i can be perfectly reconstructed by $\tilde{x}_i^{(i)}$, $\alpha(\tilde{x}_i^{(i)}, x_i) = \infty$. Hence, if $\tilde{x}_i^{(i)}$ can be found to maximize $\alpha(\tilde{x}_i^{(i)}, x_i)$, $\tilde{x}_i^{(i)}$ should be very similar to x_i . Initially, $\tilde{x}_i^{(0)} = S_i$ and hence $\alpha(\tilde{x}_i^{(0)}, S_i) = \infty \gg \alpha(\tilde{x}_i^{(0)}, x_i)$ for $i = 0$. When i increases, $\alpha(\tilde{x}_i^{(i)}, S_i)$ will first decrease rapidly and then slowly decrease while $\alpha(\tilde{x}_i^{(i)}, x_i)$ will slowly increase. If excess iterations (i becomes larger) are performed, $\alpha(\tilde{x}_i^{(i)}, x_i)$ may decrease, *i.e.*, $\tilde{x}_i^{(i)}$ may begin to be distant from x_i . However, at the decoder, x_i is unknown and only $\alpha(\tilde{x}_i^{(i)}, S_i)$ can be known. Under this circumstance, it is not guaranteed that when $\alpha(\tilde{x}_i^{(i)}, S_i)$ decreases and $\alpha(\tilde{x}_i^{(i)}, x_i)$ increases. Hence, **the first stopping criterion** can be determined as follows. When the relative change in the Laplacian parameter $\alpha(\tilde{x}_i^{(i)}, S_i)$ is smaller than a threshold T_a , *i.e.*, if

$$|\alpha(\tilde{x}_i^{(i)}, S_i) - \alpha(\tilde{x}_i^{(i-1)}, S_i)| / \alpha(\tilde{x}_i^{(i-1)}, S_i) \leq T_a, \quad (10)$$

the algorithm will stop.

On the other hand, the major goal of GPSR is to find the optimal $\tilde{\theta}_i^{(i)}$ by minimizing Eq. (9), where $\tilde{x}_i^{(i)} = \Psi\tilde{\theta}_i^{(i)}$. Without considering video characteristics, the solution obtained by minimizing Eq. (9) may be over-sparse, leading to lower visual quality. To preserve the video characteristic for a non-key frame, its SI, *i.e.*, the correlations among this frame and its neighboring frames, can be exploited. By adding an extra term, a quality-preserving fitness function can be derived as:

$$F(\tilde{\theta}_i^{(i)}) = W_1 \times F_1(\tilde{\theta}_i^{(i)}) + W_2 \times F_2(\tilde{\theta}_i^{(i)}), \quad (11)$$

where $F_1(\tilde{\theta}_i^{(i)})$ is defined by Eq. (9) and $F_2(\tilde{\theta}_i^{(i)})$ is defined as:

$$F_2(\tilde{\theta}_i^{(i)}) = \|\tilde{\theta}_i^{(i)} - \theta_{S_i}\|_2, \quad (12)$$

where $S_i = \Psi\theta_{S_i}$, S_i is the SI of $x_i = \Psi\theta_i$, and W_1 and W_2 are weighting coefficients, empirically set by 0.9 and 0.1, respectively. Initially, $\tilde{\theta}_i^{(0)} = \theta_{S_i}$, $F_2(\tilde{\theta}_i^{(0)}) = 0$, and $i = 0$. When i increases, $F_2(\tilde{\theta}_i^{(i)})$ will increase while $F_1(\tilde{\theta}_i^{(i)})$, *i.e.*, Eq. (9), will decrease. The major goal to evaluate Eq. (11) is that while GPSR attempts to minimize $F_1(\tilde{\theta}_i^{(i)})$, the similarity between $\tilde{\theta}_i^{(i)}$ and θ_{S_i} should be preserved to a certain degree. Hence, **the second stopping criterion** can be determined as follows. If $F(\tilde{\theta}_i^{(i)})$ in Eq. (11), when compared with the one obtained in the previous iteration, is increased, *i.e.*, if

$$F(\tilde{\theta}_i^{(i)}) - F(\tilde{\theta}_i^{(i-1)}) > 0, \quad (13)$$

the algorithm will stop. In addition, when the relative change in Eq. (11) is smaller than a threshold T_F (default $T_F = 0.001$), *i.e.*, if

$$|F(\tilde{\theta}_i^{(i)}) - F(\tilde{\theta}_i^{(i-1)})| / F(\tilde{\theta}_i^{(i-1)}) \leq T_F, \quad (14)$$

the algorithm will stop. This is **the third stopping criterion**.

Based on our simulations, when the measurement rate (MR) for a non-key frame is low, the initial solution (initialized by its SI) is already very close to the optimal solution. The algorithm can usually stop in few iterations, and the first stopping criterion is very suitable. When MR is high, the other two criteria should be also exploited. The stopping criteria at DCVS decoder for a non-key frame $\tilde{x}_i^{(i)}$ at the i -th iteration can be summarized as follows:

(a) MR is low ($MR \leq 20\%$): if Eq. (10) with $T_a = 0.9$ is satisfied, the algorithm will stop.

(b) MR is middle ($20\% < MR \leq 70\%$): if Eq. (10) with $T_a = 0.05$ or Eq. (13) is satisfied, the algorithm will stop.

(c) MR is high ($MR > 70\%$): if Eq. (14) is satisfied, the algorithm will stop.

The above-mentioned thresholds, T_a and T_F , are empirically decided, and fixed for all test video sequences. Finally, x_t can be reconstructed via $\tilde{x}_t = \Psi \tilde{\theta}_t$, where $\tilde{\theta}_t$ is the final solution obtained by GPSR. Our DCVS decoding procedure is summarized in Fig. 2.

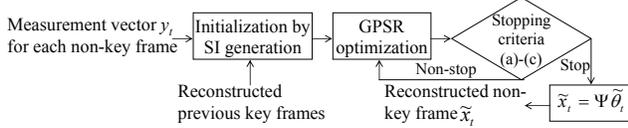


Fig. 2. Our DCVS decoder.

4. SIMULATION RESULTS

In this paper, two CIF (frame size: 352×288) video sequences (300 Y frames for each), *Coastguard* and *Foreman*, with GOP size = 3, and different measurement rates (MR s) were employed to evaluate the proposed DCVS method. For example, the average $MR = 30\%$ means that the MR s for each key and non-key frames are 50% and 20%, respectively. The three known sparse signal reconstruction algorithms, GPSR [9], TwiST [10], and OMP [11], with default settings were used for comparisons with our DCVS. The three algorithms were applied to each frame individually. For OMP [11], the reconstruction complexity will be too expensive if it is directly applied to a whole frame. As suggested by [8], OMP can be individually applied to each 32×32 block with good trade-off between CS efficiency and reconstruction complexity. The four evaluated algorithms used the same measurement matrix, SBHE [7] and the same basis matrix, DWT. The four algorithms possess the same low-complexity encoder (the same CS measurement process).

The average PSNR performances at different average MR s for the two sequences are shown in Fig. 3(a) and (b), respectively. The average reconstruction complexities (in seconds) for obtaining Fig. 3(b) are shown in Fig. 4(a). The average PSNR performances at different reconstruction complexities at $MR = 30\%$ for the *Foreman* sequence are shown in Fig. 4(b). It can be observed from Figs. 3 and 4(a) that the PSNR performances of our DCVS can outperform or be comparable with the three known algorithms, especially at low MR s, with lower or comparable reconstruction complexities. At lower MR s, initializing by SI in our method can achieve good performances while at higher MR s, all the four algorithms can achieve similar performances. For the *Coastguard* sequence with slower motions, the SI is more accurate than that of the *Foreman* sequence, and better performance can be achieved. Based on Fig. 4(b), the PSNR performances of our DCVS can significantly outperform the three known algorithms at the same reconstruction complexities.

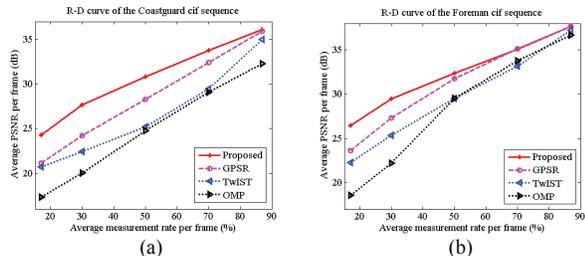


Fig. 3. The MR-PSNR performances for the: (a) *Coastguard* and (b) *Foreman* sequences.

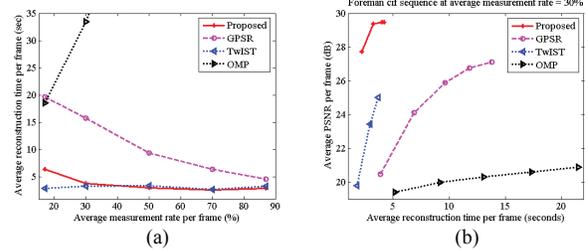


Fig. 4. (a) The reconstruction complexities for the *Foreman* sequence. (b) The PSNR performance at different reconstruction complexities for the *Foreman* sequence.

5. CONCLUSIONS

In this paper, a distributed compressive video sensing (DCVS) framework is proposed to simultaneously capture and compress videos at the low-complexity encoder and efficiently reconstruct videos at the decoder. For future researches, the key components, such as measurement matrix and reconstruction algorithm, in compressive video sensing should be designed based on video characteristics. The theoretical number of measurements for signal perfect reconstruction in Eq. (2) should also be further reduced with side information incorporated. In addition, efficient quantization and entropy coding techniques for CS measurements should be investigated to achieve complete video compression.

ACKNOWLEDGEMENT

This work was supported in part by National Science Council, ROC, under Grants NSC 95-2422-H-001-031 and NSC 97-2628-E-001-011-MY3.

REFERENCES

- [1] F. Pereira et al., "Distributed video coding: selecting the most promising application scenarios," *Signal Processing: Image Communication*, vol. 23, pp. 339-352, 2008.
- [2] C. Guillemot et al., "Distributed monoview and multiview video coding: basics, problems and recent advances," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 67-76, Sept. 2007.
- [3] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14-20, Mar 2008.
- [4] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Mag.*, vol. 25, pp. 83-91, 2008.
- [5] V. K. Goyal, A. K. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," *IEEE Signal Processing Mag.*, vol. 25, 2008.
- [6] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Universal distributed sensing via random projections," *Proc. ACM/IEEE Int. Conf. on Information Processing in Sensor Networks*, 2006.
- [7] L. Gan, T. T. Do, and T. D. Tran, "Fast compressive imaging using scrambled hadamard ensemble," *Proc. EUSIPCO*, 2008 (Matlab code available from <http://thanglong.ece.jhu.edu/CS/>).
- [8] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive video sampling," *Proc. EUSIPCO*, Lausanne, Switzerland, August 2008.
- [9] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "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, pp. 586-597, Dec. 2007 (Matlab code available from <http://www.lx.it.pt/~mtf/GPSR>).
- [10] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwiST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. on Image Processing*, vol. 16, Dec. 2007 (Matlab code available from <http://www.lx.it.pt/~bioucas/TwiST/TwiST.htm>).
- [11] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Trans. on Signal Processing*, vol. 56, June 2008 (Matlab code available from <http://www.see.ed.ac.uk/~tblumens/sparsify/sparsify.html>).
- [12] "AviSynth MSU frame rate conversion filter," http://www.compression.ru/video/frame_rate_conversion/index_en_msu.html.