VIDEO COMPRESSIVE SENSING

Jiaqian Zhong, Subhrajit Das, Yuetong Liu

Deapartment of Electrical and Computer Engineering,
Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD

ABSTRACT

In this work, we present three video compressive sensing algorithms. The first algorithm GAP-TV considers the total variance (TV) used for compressive sensing and solve it by the generalized alternating projection (GAP) algorithm. It takes a series of compressed frames and reconstuct them. The second algorithm is the end to end neural network based algorithm that takes a video as input and reconstruct the video.The third algorithm is the state-of-the-art iterative optimization method DeSCI, which projects the measurement/ residual to the signal space to fit the sampling process in the SCI system and uses WNNM denoising for video patches iteratively. We compare first two algorithms by running three short videos from UCF101[1]. While the third algorithm, even though promises best results among all, hit a roadblock of lack of resources for computing the reconstruction. Our analysis offers pros and cons of these two algorithms in terms of reconstruction quality and run-time.

Index Terms**—** Compressive sensing, video compressive sensing, generalized alternating projection, total variance,deep denoisning, multi-layer network

1. INTRODUCTION

Compressive sensing (CS) has inspired various compressive imaging systems that capture high-dimensional data, such as videos(Y. Hitomi 2011) and hyperspectral images, in a snapshot fashion. For example, in video CS as shown in Fig. 1, the high-speed frames of a video are modulated at a higher speed than the capture rate of the camera. With knowledge of the modulation, multiple frames can be reconstructed from each single measurement. This type of technique is also termed snapshot compressive imaging (SCI). Capturing high-dimensional data has long been viewed as a challenge in signal processing which can be addressed by SCI. In this project, we will implement reconstruction algorithms to recover the high-speed video. Reconstruction methods are:(i)GapTV (ii)E2E-MLP (iii)DeSCI

2. METHOD

2.1. Snapshot Compressed Sensing(SCI)

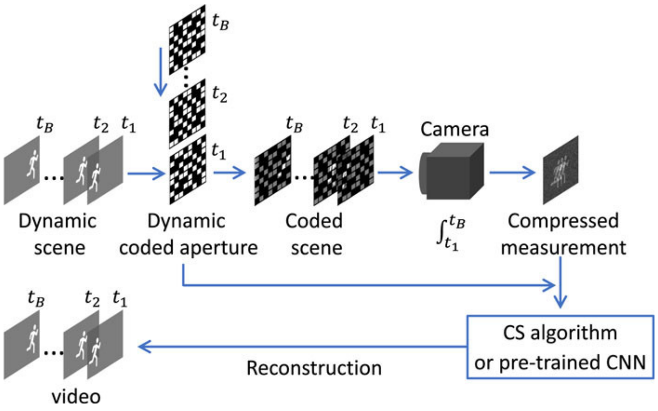


Fig. 1. Principle of video SCI.

The sensing process of video SCI is shown in Fig.2. A dynamic scene, modeled as a time-series of two-dimensional (2D) images, passes through a dynamic aperture which applies timestamp-specified spatial coding. In specific, the value of each timestamp-specified spatial coding is superposed by a random pattern (binary patterns are used in this paper, 0, 1 with 0 denoting blocking the light and 1 meaning the light passing through) and the spatial coding of each two timestamps are different and independent from each other. The coded frames after the aperture are then integrated over time on a camera, forming a compressed coded measurement. Given the coding pattern for each frame, the time series of the scene can be reconstructed from the compressed measurement through iterative algorithms or pre-trained convolutional neural networks (CNNs).

In general, the SCI problem focuses on computational imaging which is quite different from traditional imaging where the users acquire the desired signal directly. In SCI, the captured measurement is not visually interpretable but includes the signal in a carefully designed mechanism. As a result, the reconstruction algorithms are required to recover the signals from the measurement. For SCI problems, the well established algorithms include GAP-TV(X. Yuan,2016)

Thanks to internet for funding.

[1] https://www.crcv.ucf.edu/data/UCF101.php

and GMM(J. Yang, 2014), based algorithms, where different priors are used. Most recently, the DeSCI algorithm has led to the state-of-the-art results of video SCI. DeSCI applies the weighted nuclear norm minimization (WNNM) of nonlocal similar patches in the video frames into the alternating direction method of multiplier (ADMM) regime.Here we will be exploring 2 iterative based algorithm: GapTV and DeSCI alongthwith one Deep learning based algorithm,E2E-MLP.

2.2. GAP-TV

The purpose of GAP-TV is to minimized the total variation (TV) using the generalized alternating projection (GAP) algorithm. It solves the following problem:

$$min_{x,C} \text{ subject to } TV(x) \leq C \text{ and } \phi x = y$$

where C is the radius of the TV of the signal.
We regard it as a series of alternating projection problem:

$$(x^{(t)}, \theta^{(t)}) = argmin_{x,\theta} \frac{1}{2} \parallel x - \theta \parallel_2^2 + \lambda \parallel TV(\theta) \parallel$$

$$\text{subject to } \phi x = y$$

To solve it, we update θ and x alternatively:

$$x^{(t)} = \theta^{(t-1)} + \phi^T(\phi\phi^T)^{-1}(y - \phi\theta^{(t-1)})$$

$$\theta^{(t)} = x^{(t)} - D^T z^{(t)}$$

where

$$z^{(t)} = clip(z^{(t-1)} + \frac{1}{\alpha} D\theta^{(t-1)}, \frac{\lambda}{2}$$

$$clip(b, T) := \begin{cases} b & \text{if } |b| \leq T \\ Tsign(b) & \text{otherwise} \end{cases} \quad (1)$$

The iteration starts from $z^{(0)} = 0$, then updates θ and x sequentially.

2.3. E2E-MLP

A.Patch Extraction

First we let a set of unknown signals as $X : N_f \times 1$ and the captured signals as $Y : M_f \times 1$ and $M \ll N$. Each sampling mask is denoted by $\Phi 1, ...\Phi t$. Specifically, $\Phi = [diag(\Phi 1), ...diag(\Phi t)]$, $M_f \times N_f$ and diag() denotes the diagonal matrix from each input vector. Then

$$Y = \Phi X$$

If we apply the linear mapping as $Y = WX$, then W would be too large to compute and store. To avoid this complexity, we apply block to the videos. Then the collected video blocks are $X_i, i \in N$, with the size of $W_p \times h_p \times t$. Now the linearing mapping is

$$Y = \Phi_p X$$

where $Y = [Y_1, Y_2, ...Y_N], X = [X_1, X_2, ...X_N]$ and Φ_p is the same for all blocks. Then the linear mapping becomes $Y = W_p X$. Therefore, we want to

$$\min \|X - W_p Y\|_2^2, \text{ with respect to } W_p$$

which is equivalent to

$$W_p = (XY^T)(TY^T)^{-1}$$

The intuition behind this method is that if the number of frames of desired signals is sufficiently large and the sensing matrix Φ_p has at least one nonzero entry in each row, the recovered videos can be surprisingly good.

For this study, we selected the blocking size to be $8 \times 8 \times 16$, with 50 percent of non-zero random binary elements. We set $N_p = 1024$ and $M_p = 64$, therefore the CR is 1/16.

B.Multi-layer Network

We applied an end-to-end Multi-layer Network that takes a measured frame patch X_i and maps it to a video block Y_i via 7 nonlinear layers. The input layer transforms the compressed 2D measurements to 3D signals. For each fully connected hidden layer FC_K, the non-linear function is $FC_K(Y) = \sigma(b_k + W_k Y)$, where b is the bias term and $W_k$is the weight matrix. Each hidden layer is followed by a Relu activation function σ to avoid easy saturation. The model weight W is updated during the backpropagation. We used mean squared error as the loss function, which is defined as

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{Y}_i)^2$$

, where $\hat{Y}_i$ is the reconstructed video block and X_i is the corresponding video block.

2.4. DeSCI

Mathematically, the measurement in the SCI systems can be modeled as

$$Y = \Phi x + g$$

where ϕ is the sensing matrix and x is the desired signal.In video snapshot compressive imagers, i.e., the CACTI system [5] uses the sensing matrix ϕ which can be written as:

$$\phi = [D_1, D_2, ...D_B]$$

where D are the diagonal matrices. For integrating Weighted Nuclear Norm Minimization(WNNM) to SCI, all video frames are divided into N overlapping patches of size $\sqrt{d} \times \sqrt{d}$ and each patch is denoted by a vector z_i where i=1,2...N.After this all the patches are stacked together into a matrix Z_i

$$Z_i = [z_1, z_2, ...z_N]$$

This matrix Z_i consisting of patches with similar structures is thus called a group. Since all patches in each data matrix have similar structures, the constructed data matrix Z_i is of low rank.So by using rank minimization as a constraint, the SCI reconstruction can be formulated as

$$\hat{x} = argmin_x \frac{1}{2} \parallel y - \phi x \parallel_2^2 + \lambda \sum_i \parallel Z_i \parallel_*$$

where λ is a parameter to balance these two terms, and recall that Z_i is constructed from x.

For solving this SCI-WNNM problem ADMM solver is used where an auxillary variable θ is introduced to the problem

$$\hat{x} = argmin_x \frac{1}{2} \parallel y - \phi \theta \parallel_2^2 + \lambda \sum_i \parallel Z_i \parallel_*$$

$$\text{subject to } x = \theta$$

This optimization problem can be further broken down into 3 sub problems for solving θ and x seperately by iterative approach. Finally Z_i are aggregated over i=1,..N to recover the video(hyperspectral images) x.

2.5. FFDNet

To improve the reconstruction result, we used deep denoising priors (K. Zhang, 2018) to solve $z^{(t)}$. This algorithm doesn't need to retrain the model, enabling the algorithm's flexibility. In our work, we chose FFDNet (K. Zhang, 2017) which has the best performance in our videos among different denoising algorithms. However, FFDNet was trained by Gaussian noise, while the noise distribution in our video SCI is different in every iteration. Therefore, we used a joint denoising strategy as follows:

$$z_1^{t+1} = FFDNet(x^{t+1} - u^t, \sigma^t)$$

$$z_2^{t+1} = TV(x^{t+1} - u^t, \sigma^t)$$

where σ^t is the estimated Gaussian noise level in each iteration.
The final z^{t+1} in each iteration is achieved by

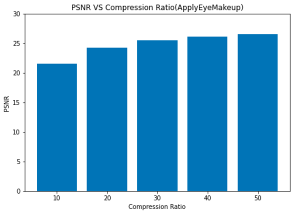
$$z^{t+1} = \alpha z_1^{t+1} + (1 - \alpha) z_2^{t+1}$$

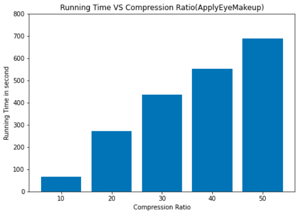
where $0 \leq \alpha \leq 1$ is a weighted parameter.

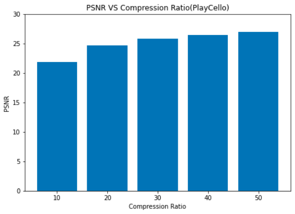
3. RESULT

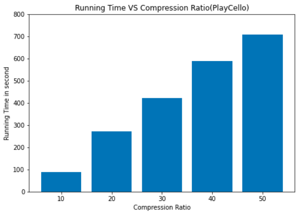
3.1. GAP-TV

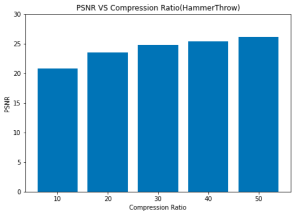
The origin videos are compressed with different compression ratios (CR) to see how CR could influence the reconstruction result. We use PSNR as our evaluation metrics, which is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Below are plots of the PSNR and running time w.r.t. CR for three videos (Apply Makeup, Play Cello and HammerThrow).

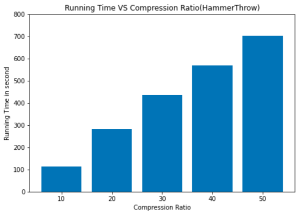




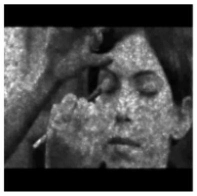


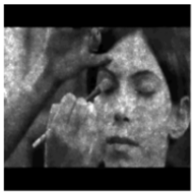


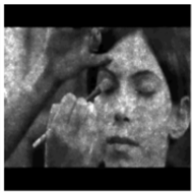


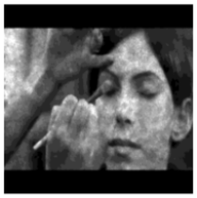


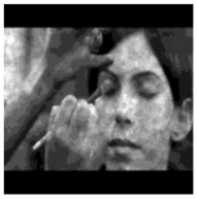
According to the plots, when CR is getting higher, PSNR gets improve with the cost of running time. However, PSNR might not be the only indicator to evaluate the output. The below plot consists of the first frame in the video "ApplyMakeup" with different CR.


CR = 10


CR = 20


CR = 30


CR = 40


CR = 50

Even though the frame with higher CR contains less noise, the resolution is lower. Moreover, video with high CR is less fluent than videos with low CR. Therefore, when choosing CR, we need to make a trade-off between image quality and video fluency.

3.2. Deep Denoising

We integrated pre-trained denoising networks into the iteration-based algorithms,Gap-TV which is dubbed as Plug-and-Play (PnP) framework.From Fig. 2 it can be seen it has significantly improved the image quality.This has resulted in a higher psnr than GAP-TV reconstruction. Although the speed of the PnP framework is not comparable to that of the end-to-end deep learning framework, its overall performance on quality, flexibility, ease-of-use, cost, and speed makes it a good baseline for SCI reconstruction.

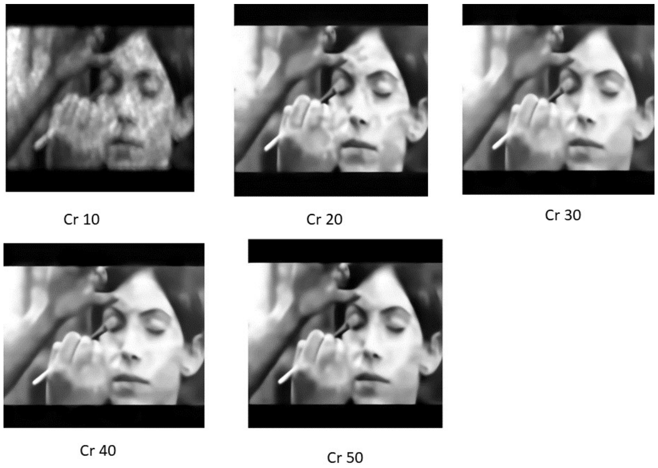
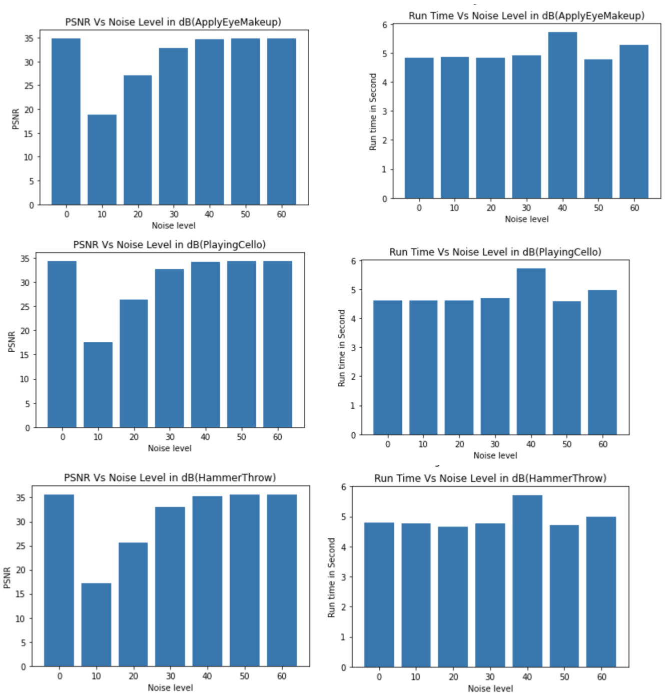

Cr 10
Cr 20
Cr 30
Cr 40
Cr 50

Fig. 2. Gap-TV+DD

3.3. E2E-MLP

We added random Gaussian noise to compare the noise level from 0dB to 60 dB and see if there is a linear relation between noise level and PSNR. Since changing the CR requires to modify the model architecture and train a new model. The model is trained on 10 million video patches of size 41GB, therefore we used the pre-trained model and the CR is fixed. Below are plots of the PSNR and running time w.r.t. noise level for three videos (Apply Makeup, Play Cello and HammerThrow). As shown in the graph below, the reconstruction qualities are robust as noise level increase. We also compared the model run time as noise level increase and the model run time is robust as noise level increased as well.



4. DISCUSSION

As shown in the figures above, the PSNR with CR=50 achieves the best reconstructed quality around 22dB via GAP-TV. However, E2E-MLP outperform the best recovery result of GAP-TV around 35dB. In terms of run-time, GAP-TV takes longer as the CR increases and the minimum run-time is around 100 seconds. As mentioned earlier, the E2E-MLP is pre-trained on 10 million video patches[2] of size 41GB and we don't know the exact training time. The reconstruction time for each video is around 4.7 second. Therefore, in terms of the reconstruction run-time, E2E-MLP is faster than GAP-TV.

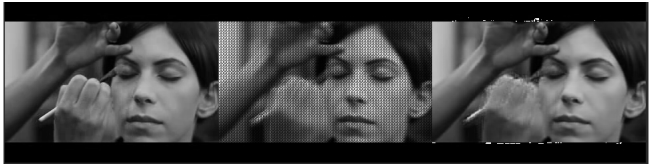
	PSNR_EyeMakeup	PSNR_PlayCello	PSNR_HammerThrow
GAP-TV	21.59 dB	21.86 dB	20.81 dB
E2E-MLP	34.89 dB	34.38 dB	35.65 dB
	Runtime_EyeMakeup	Runtime_PlayCello	Runtime_HammerThrow
GAP-TV	65 s	88 s	114 s
E2E-MLP	4.82 s	4.61 s	4.79 s

Table1. GAP-TV with CR=10, E2E-MLP with no noise

In terms of training cost, E2E-MLP requires 41GB videos to achieve promising result. GAP-TV does not need any training data. One drawback of the E2E-MLP model is it lacks of flexibility. The pre-trained network only works with specific mask patterns, image size and cpmrpession ration. Another drawback is that it is less good at reconstructing the still object than the moving object. As shown in the ApplyEyeMakeup figure with CR=1/16 below, the moving hand in the

[2] https://drive.google.com/file/d/1NyMIyp9N-UCYCobcTv7CFYd1zfXG9wJn/view?usp=sharing

reconstructed frame is corrupted.



The figures from left to right are the original frame, compressed frame and reconstructed frame

Overall, there is a tradeoff between training cost, reconstruction speed, image quality and flexibility. E2E-MLP suffers from expensive training cost and flexibility, but have fast and high reconstruction quality. GAP-TV suffers from blurry artifacts and have relatively long reconstruction time, but does not need any training process and is flexible. The deep denoising method that we used over reconstructed images generate decent results without task-specific pre-training and is faster than conventional iterative algorithms. Considering speed, accuracy, and flexibility, this deep denoising method will serve as a baseline in video SCI reconstruction

Even though we explored DeSCI, since it is a state-of-the-art iterative optimization method which has given the best reconstruction results on multiple datasets, unfortunately it hit a roadblock due to lack of the computation power available to run it.

5. FUTURE WORK

We used fully connected dense layers in the E2E-MLP. For future work, we can try on more neural network architecture such as CNNs or RNNs since CNNs are good for processing image and RNNs do good on time series data. In addition, we can train on deeper architecture. Also, fast deep denoising priors enables the algorithms' flexibility, Therefore, deep denoising priors can be added to the iteration based algorithms. Regarding future work, we expect to exploit the correlation between video frames to train a video-wise deep denoising prior, rather than the frame-wise prior presented in this paper, for the PnP framework to improve the reconstruction quality.

6. REFERENCES

[1] Iliadis, Michael, Leonidas Spinoulas, and Aggelos K. Katsaggelos. ”Deep fully-connected networks for video compressive sensing.” Digital Signal Processing 72 (2018): 9-18.

[2] X. Yuan, ”Generalized alternating projection based total variation minimization for compressive sensing,” IEEE International Conference on Image Processing (2016): 2539-2543.

[3] Qiao, Mu, et al. ”Deep learning for video compressive sensing.” APL Photonics 5.3 (2020): 030801.

[4] K. Zhang, W. Zuo, and L. Zhang, “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” IEEE Trans. Image Process. 27 (2018): 4608–4622.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,” IEEE Trans. Image Process. 26 (2017): 3142–3155.

[6] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, “Video from a single coded exposure photograph using a learned over-complete dictionary,” in 2011 International Conference on Computer Vision (IEEE, 2011), pp. 287–294.

[7] J. Yang, X. Yuan, X. Liao, P. Llull, G. Sapiro, D. J. Brady, and L. Carin, “Video compressive sensing using Gaussian mixture models,” IEEE Trans. Image Process. 23, 4863–4878 (2014). https://doi.org/10.1109/tip.2014.2344294,