Compressed Sensing for Natural Videos: A Survey of Video Compressed Sensing Network (VCSNet)

Shrikant Arvavasu ashri@umich.edu

Abstract

Compressed Sensing (CS) has gained a lot of attention among researchers in information theory. Natural videos, being highly compressible due to the inter-frame redundancies, could be effectively sub-samples at a considerably lower rate. Traditional approaches to compressed sensing for images incorporate the use of random sensing matrices (or operators) and iterative projection algorithms or greedy algorithms for reconstruction. Considering the dimensionality and computation costs for videos, these approaches become infeasible. This project aimed to employ deep-learning techniques in both sampling and reconstruction while employing techniques that exploit the inter-frame correlation to provide an optimal sub-sampling rate. We employed the concept of a Group of Pictures Blocked Compressed Sensing (GOP-BCS) which was implemented using a deep convolutional network. The experiments with a single keyframe setting and double keyframe setting show decent results in the well-known Kitti Dataset.

1. Introduction

Compressive sensing aims at sampling any signal at sub-Nyquist rates and efficiently reconstructing them, provided some assumptions about the signal are satisfied. For instance, if one assumes that the sampled signal is sparse in some transform domain (say Fourier domain), one can formulate the reconstruction of the signal, from its samples, as a variant of the well-known Lasso problem, for which efficient solvers have been developed to achieve state-of-the-art results. The notion of compressed sensing has gathered attraction in complex imaging systems like Magnetic Resonance Imaging (MRI), where sampling at the Nyquist rate is a time-consuming and power-intensive process, and video capture systems, where inter-frame redundancies allow high compression rates.

However, some of the most challenging aspects of an efficient compressive sensing system are the knowledge of the transform domain and the subsampling algorithm. From Nyquist's sampling theory, it is quite conclusive that subsampling the signal at a uniform rate would result in the aliasing effect. Therefore, non-uniform sampling methods like domain-aware sampling for MRI, and block-based sampling for natural images and videos are studied.

This project aimed to explore deep learning techniques in [1] for the sampling and reconstruction of natural videos using a deep convolutional neural network and provide an analysis of the methods used in the paper.

Natural videos in modern days, in demand for superior quality, are captured at high frame rates. However, as the frame rate for video capture increases, the storage space and processing time for the video data increase linearly. Compression schemes like H.26x, and MPEG use the compressibility of frames in the transform domain (e.g. in the wavelet domain) and use some form of the residual of frames for storing the data in a compressed form. However, CS theory [2] aims at collecting fewer measurements in the first place and using the same argument of compressibility to reconstruct the video data effectively.

For natural images, Block-Compressed Sensing (BCS) methods have been quite effective in the reduction of measurements, by sub-sampling an image in a block-based fashion. This notion is naturally extended to videos as a per-frame BCS approach. However, researchers have shown that the straightforward use of BCS

alone for video CS is not sufficient as the BCS only exploits the intra-frame correlations, but not inter-frame correlations. Lam and Wunsch used a 3-D wavelet domain for sparse representation of videos. Mun and Fowler used motion compensation to learn the motion between the previously reconstructed frame and the current frame. In [1], the inter-frame correlation is learned by assigning the first frame of the input video as a keyframe and others as non-key frames. A reconstruction network (convolutional network) is then trained to compensate the non-key frames with a *multi-level feature compensation*. The reconstruction is achieved by solving the non-linear least squares regression problem with the dimension constraints enforced by the deep network itself. Shi et al. also provided an analysis on two approaches, where one and two key-frames were used respectively. Using two-key frames instead of one provides a much better reconstruction, but calls for a higher computation cost than using a single key-frame. We have currently reproduced the results for the single keyframe-based sampling and reconstruction.

2. Related Work

Compressed sensing was first studied and published in [2], stating that natural signals and images can be recovered faithfully by solving a linear-program-based basis pursuit. It suggests that an m pixel image requires only $n = \mathcal{O}(m^{1/4}log(m)^{5/5})$ non-pixel samples to faithfully recover the image. It also suggests that if a signal has an m-sparse representation in some orthonormal basis, we only require $n = \mathcal{O}(Nlog(m))$ samples to reconstruct the signal with accuracy equivalent to having knowledge of the N most important coefficients. In [5], an alternate approach is discussed, where weighted $\ell_p(p \leq 2)$ norm is used in Landweber iterations followed by a thresholding operation. The success of this approach depends highly upon the chosen sampling matrix, which, in most cases, has to be accurately transmitted beforehand, which is highly challenging. Thus several approaches where Bernoulli [6] or Gaussian-based sampling matrices were introduced for the subsampling process.

For natural images, block-based sampling methods are highly encouraged, as sparsity of image blocks in wavelet basis [7] or learned dictionaries [8] could be effectively utilized in computationally efficient reconstruction. In, [9], a deep-learning-based stacked denoising autoencoder (SDA) is proposed to capture statistical dependencies between the different elements of signals and improve reconstruction performance. But this approach demands the use of the Orthogonal Matching Pursuit (OMP) [10] as a post-processing step, which becomes infeasible in terms of memory and computation for image and video applications.

The pursuit of deep learning algorithms is explored in this project, which aims at the feasibility of sampling and reconstructing the sequence efficiently.

3. Methodolgy

In this section, we can explore the architecture of the VCSNet model. The architecture consists of a sampling network that performs the block-wise sampling of the frames, a frame-wise initial reconstruction network that provides an initial linear reconstruction of each frame independently, and a deep non-linear reconstruction network that captures the inter-frame correlations to fine-tune the quality of the reconstructed frames. This project only aimed at single-keyframe sampling owing to the computational demands for training on large datasets.

3.1. Sampling Network

The sampling network aims to implement a block-wise weighted sampling process, with a sub-sampling parameter α. Each frame is divided into $B \times B$ blocks. From each of these blocks, αB^2 samples are taken using a convolutional layer with a filter size of B and stride of B. Each filter captures one sample from each block, and thus the sampling layer has αB^2 convolutional kernels (W^s) for sampling the entire image. Figure 1 illustrates an overview of the sampling process.

In frame-wise sampling for videos, we can select a choice of having keyframes and non-keyframes which highlight the importance of the sampled frames. Consequently, the keyframes are sampled with a rate of α_{key} and the remaining frames are sampled at a much smaller sampling rate (say α_k). For the framework mentioned, this notion can be implemented by incorporating frame-wise convolutional layers for keyframes

and non-keyframes.

Thus, the sampling process can be expressed as ($*$ denotes the convolution operator),

$$Y_{key} = W^s_{key} * X_{key} \quad \text{(Keyframe Sampling)}$$
$$Y_k = W^s_k * X_k \quad \text{(Non-Keyframe Sampling)}$$

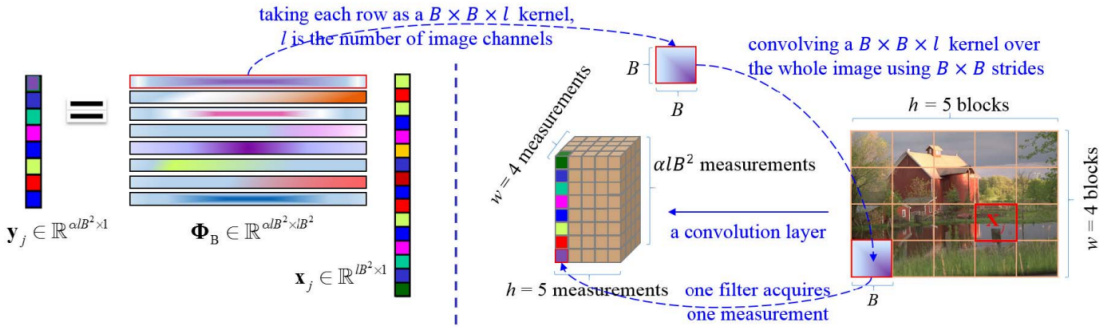

Figure 1. Block-based Sampling Process in [1]

3.2. Frame-wise Initial Reconstruction

The frame-wise initial reconstruction acts as an inverse problem, where given the sampled measurements, we perform linear reconstruction, independently (w.r.t frames). Now, implementing an analytical inverse of the block-based sampling process is highly infeasible and unstable to perturbations, so we incorporate a convolutional layer, similar to that of the sampling process, where we have αB^2 measurements per block. Since the measurements of each block are stored as a vector of size $1 \times 1 \times \alpha B^2$, we convolve these measurements with a $1 \times 1 \times \alpha B^2$ convolutional kernel (W^{ir}), which can recover one pixel in each frame. Thus, the initial reconstruction process can be illustrated as,

$$\tilde{X}_{key} = W^{ir}_{key} * Y_{key} \quad \text{(Keyframe Reconstruction)}$$
$$\tilde{X}_k = W^{ir}_k * Y_k \quad \text{(Non-Keyframe Reconstruction)}$$

3.3. Deep Reconstruction

The deep reconstruction network aims to capture the inter-frame correlations using a stacked-deep convolutional network. The initial reconstructions $\tilde{X}_{key}$ and $\tilde{X}_k$ are passed through convolutions W_d and W_{ref} as,

$$D^1_{key}(\tilde{X}_{key}) = \text{Act}(W^1_{d\,key} * \tilde{X}_{key})$$
$$D^{(i)}_{key} = \text{Act}(W^i_{dkey} * D^{(i-1)}_{key}) \quad i = 2, ..., N-1$$
$$D^{(i)}_k = \text{Act}(W^i_d * \tilde{X}_k + W_{ref} * D^{(i)}_{key}(\tilde{X}_{key})) \quad i = 1, ..., N$$
$$\text{where,} \quad \text{Act}(x) = \text{ReLU}(x) = \max(0, x)$$
$$\hat{X}_{key} = D^{(N)}_{key}$$
$$\hat{X}_k = D^{(N)}_k$$

where, W_d and W_{ref} are $d \times d$ convolutional filters which capture the interframe correlations. Figure 2 gives a pictorial overview of the deep reconstruction process.

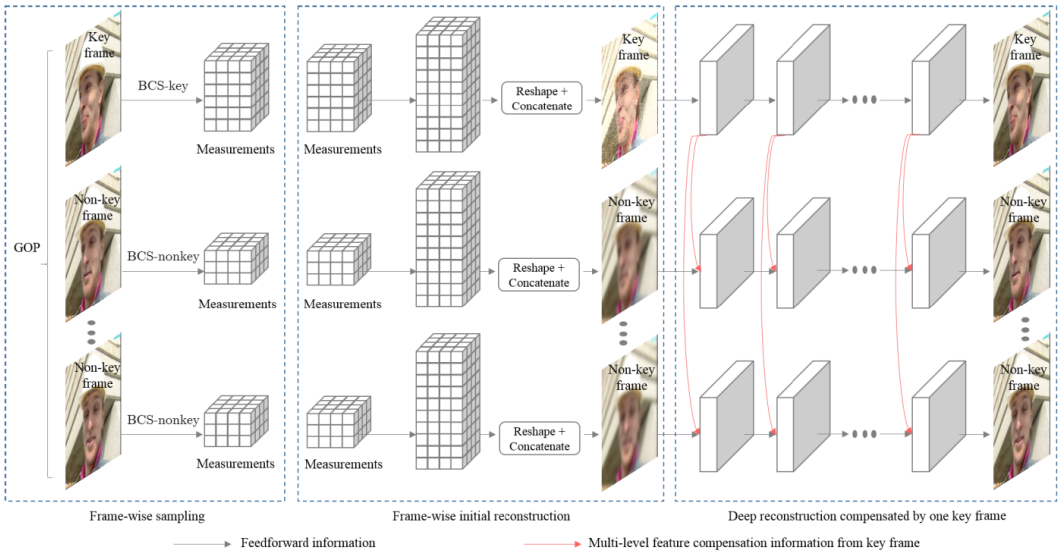

Figure 2. Deep Reconstruction in [1]

4. Experimentation and Results

4.1. Experimental Setup

The sampling and reconstruction networks were implemented in Pytorch and the training was accomplished using Google Colab Pro Premium GPUs and Nvidia A40 GPUs. For training the model, Kitti Dataset [11] was used, which consisted of 160 videos and 40 testing videos each containing 21 frames. The parameters α_{key} and α_k were chosen to be 0.5 and 0.25 for compressed sampling. The images were sampled using a block size of 32, with a single keyframe setting having one keyframe and 20 non-keyframes and a double keyframe setting with 2 keyframes and 19 non-keyframes.

For the deep reconstruction, N=5 convolution layers, each having 64 filters, each of size 3×3 were chosen with bias included. The loss function for the i^{th} keyframe and j^{th} non-keyframe was given by

$$l_{(i,j)}\left(X_{key}^{(i)}, \hat{X}_{key}^{(i)}, X_k^{(j)}, \hat{X}_k^{(j)}\right) = \|X_{key}^{(i)} - \hat{X}_{key}^{(i)}\|_F^2 + \|X_k^{(j)} - \hat{X}_k^{(j)}\|_F^2$$

To evaluate the performance of the model, the Peak Signal-to-Noise Ratio(PSNR) and Structural Similarity Index (SSIM) were used. These metrics capture the reconstruction quality and statistical similarity with the ground truth video frames.

5. Results

The model was trained for 50 epochs for both single and double keyframe setting, which gave average PSNRs of 34.5dB and 40.6dB respectively for the single and the double keyframe setting respectively and structural similarity index of 0.79 and 0.87 respectively. Some example case results are tabulated in Table 1 and the reconstruction is shown in Figure 7. The complete implementation with the prepared datasets and the weights can be found here.

6. Conclusion

As of now, just the primary set of experiments was carried out, which suggests that including a few more keyframes could improve the reconstruction. There are several blocking artifacts in the output, which was duly noted in [1], suggesting the need for larger filter sizes or possibly a few dilated convolutions. Using the limited computational resources and data at hand, the model was trained to perform with a decent accuracy of reconstruction.

Video Sequence #	Single		Double	
	PSNR	**SSIM**	**PSNR**	**SSIM**
Video Seq1	33.84	0.7688	35.23	0.8322
Video Seq2	35.76	0.7811	35.33	0.7766
Video Seq3	38.33	0.8003	40.88	0.8912
Video Seq4	36.42	0.7781	34.26	0.7524

Table 1. Table of Metrics: Tabulating the performance metrics for Single Keyframe setting and Double Keyframe Setting

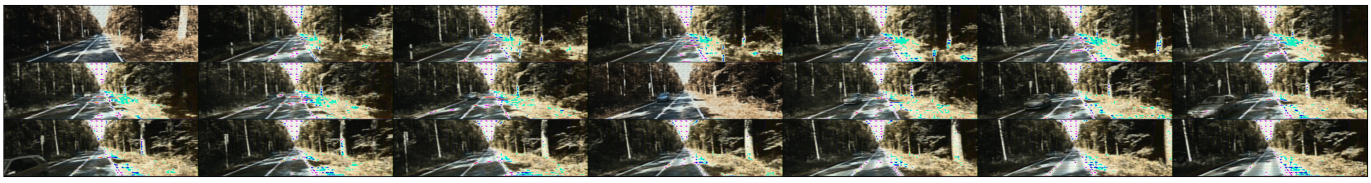

Figure 3. Reconstruction: Video Seq1 with Single Keyframe Setting

Figure 4. Reconstruction: Video Seq2 with Single Keyframe Setting

Figure 5. Reconstruction: Video Seq1 with Double Keyframe Setting

Figure 6. Reconstruction: Video Seq2 with Double Keyframe Setting

Figure 7. Reconstruction of a few example cases

References

[1] W. Shi, S. Liu, F. Jiang, and D. Zhao, "Video compressed sensing using a convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 425–438, 2021.

[2] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] D. Lam and D. Wunsch, "Video compressive sensing with 3-d wavelet and 3-d noiselet," in *2012 19th IEEE International Conference on Image Processing*, 2012, pp. 893–896.

[4] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in *2011 Data Compression Conference*, 2011, pp. 183–192.

[5] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, p. 1413–1457, 2004. [Online]. Available: http://dx.doi.org/10.1002/CPA.20042

[6] G. Zhang, S. Jiao, X. Xu, and L. Wang, "Compressed sensing and reconstruction with bernoulli matrices," in *The 2010 IEEE International Conference on Information and Automation*, 2010, pp. 455–460.

[7] J. E. Fowler, S. Mun, and E. W. Tramel, "Multiscale block compressed sensing with smoothed projected landweber reconstruction," in *2011 19th European Signal Processing Conference*, 2011, pp. 564–568. [Online]. Available: https://ieeexplore.ieee.org/document/7073994

[8] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, 2014.

[9] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep 2015. [Online]. Available: http://dx.doi.org/10.1109/ALLERTON.2015.7447163

[10] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.