End-to-End Video Compressive Sensing Using Anderson-Accelerated Unrolled Networks

End-to-End Video Compressive Sensing Using Anderson-Accelerated Unrolled Networks

Paper ID 69

Abstract—Compressive imaging systems with spatial-temporal encoding can be used to capture and reconstruct fast-moving objects. The imaging quality highly depends on the choice of encoding masks and reconstruction methods. In this paper, we present a new network architecture to jointly design the encoding masks and the reconstruction method for compressive high-frame-rate imaging. Unlike previous works, the proposed method takes full advantage of a denoising prior to provide a promising frame reconstruction. The network is also flexible enough to optimize full-resolution masks and efficient at reconstructing frames. To this end, we develop a new dense network architecture that embeds Anderson acceleration, known from numerical optimization, directly into the neural network architecture.

Our experiments show the optimized masks and the dense accelerated network respectively achieve 1.5 dB and 1 dB improvements in PSNR without adding training parameters. The proposed method outperforms other state-of-the-art methods both in simulations and on real hardware. In addition, we set up a coded two-bucket camera for compressive high-frame-rate imaging, which is robust to imaging noise and provides promising results when recovering nearly 1,000 frames per second.

Index Terms—high-frame-rate imaging, deep neural network, computational camera

1 Introduction

As a well-developed technique, compressive sensing (CS) is widely applied in reconstructing images with low sampling rates [1], [2]. In particular, a variety of mask-based CS cameras have been demonstrated for capturing high-dimensional image data (e.g., spectra, video, etc.) using a two-dimensional camera with encoding capacity. Compared to conventional cameras employing brute-force sampling strategies, such CS cameras have significant advantages in acquisition efficiency, storage consumption, and potentially cost [3], [4].

High-frame-rate imaging is concerned with recording videos at rates in excess of hundreds of frames per second. However, with bandwidth being a limiting factor, conventional cameras record either a very low spatial resolution with a relatively high frame rate, or at relatively high spatial resolution with a low frame rate. Using mask-based compressive sensing, it becomes feasible to capture high-frame-rate and high-spatial-resolution videos with an efficient spatio-temporal encoding. This approach is a good fit for recently developed image sensors with high-speed per-pixel programmable exposure control [5]. The exposure control can be viewed as an encoding of the captured frames with a set of binary temporal masks. With such cameras, it is possible to encode multiple *subframes* into a captured image and decode them later using frame reconstruction methods (Fig. 1).

Much research has focused on the improvement of the reconstruction techniques, usually by employing optimization-based approaches (see Section 2 for more detail). Less work has concentrated on the derivation of good encoding masks: it can be shown that optimal mask selection in CS is NP-complete, but random (Bernoulli or Gaussian) patterns are satisfactory with high probability [6]. However, the encoding and decoding components of the imaging system are highly interdependent. Based on this observation, we focus on the joint end-to-end design of encoding masks and reconstruction methods for improving both encoding efficiency and reconstruction accuracy. We put forward a compact end-to-end neural network that can handle the mask optimization for the whole image with fewer training parameters. We also show that this network design corresponds to Anderson acceleration, a well-known

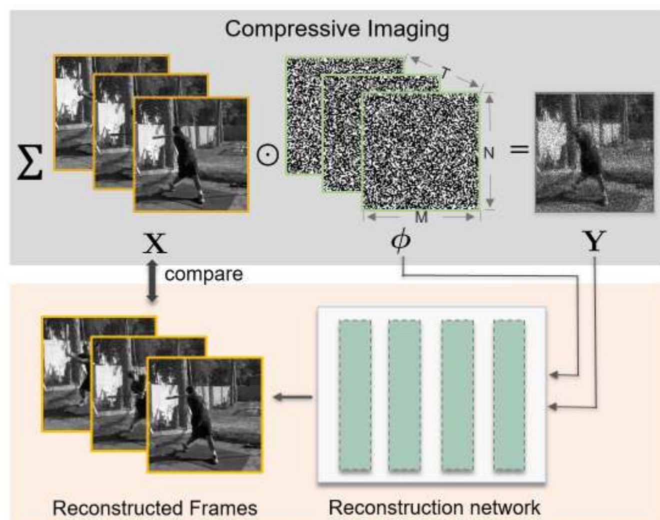


Fig. 1. Illustration of the encoding and reconstruction within the compressive high-frame-rate imaging system. In the system, T subframes with resolution $M \times N$ are encoded with masks ϕ. The reconstruction network reconstructs the frames from the measurement $\mathbf{Y}$ and the known mask ϕ.

- *This paper is under review for ICCP 2020 and the PAMI special issue on computational photography. Do not distribute.*

acceleration technique in numerical optimization [7].

Both simulations and experiments on real hardware show that our network outperforms existing methods. In addition, we show that our masks can also improve the reconstruction quality of existing methods. Our contributions can be summarized as follows:

- We present the *first work* to jointly design full- resolution coding masks and reconstruction methods for compressive high-frame-rate imaging using an end-to-end network. Our approach outperforms state-of-art methods by 2.2dB in PSNR.
- We show that the acceleration of the gradient descent algorithm is equivalent to adding dense skip connections to iterative optimization-unrolling neural networks. This speeds up training convergence and helps to design a compact and efficient *network architecture*.
- Experiments on both simulation and *real hardware* demonstrate the effectiveness of our reconstruction method and the designed masks. The two-bucket design of our camera shows improved noise suppression and can provide promising results in reconstructing video of frame rates up to almost 1,000 frames per second.

2 Related Work

Many approaches have been developed to solve the ill-conditioned inverse problem in CS. The existing methods can be divided into model-based optimization methods, deep discriminative learning methods, and unrolled iterative optimization methods.

Model-based methods.

Model-based methods utilize designed image priors for regularization, which can reduce the number of possible solutions and remove artifacts in frame reconstruction. For example, the Total Variation (TV) prior [4], [8] can simultaneously preserve edges while smooth away noise in flat regions; optical flow [9] can estimate the motion of moving objects and helps to eliminate ghosting effects; Gaussian mixture models [10] and dictionary learning methods [11], [12] take into account image statistics and reconstruct frames using learned atoms; non-local low-rank priors [13] consider correlation between small patches in the frames for denoising. Such model-based methods are straightforward to adapt to different sensing matrices without retraining, and the sensing matrix can be optimized based on the analysis of mutual coherence in dictionary-learning based methods [14]. However, such model-based methods have their respective drawbacks, and none of them is suitable for all scenes. In addition, these methods can be computationally expensive, especially compared to learning-based methods.

Learning-based methods.

In recent years, deep discriminative learning methods have shown drastic improvements in image reconstruction quality. Some deep neural networks (DNNs) have been proposed for compressive imaging as well. Convolutional neural networks [15], [16], [17] and fully-connected networks [18], [19] were developed to reconstruct small image patches. However, none of the convolutiomal networks are capable of simultaneously designing masks and optimizing parameters in the network. Compared to model-based methods, these DNN-based methods are efficient but difficult to adapt to different sensing matrices. These networks usually use random code masks, such as Gaussian or Bernoulli random masks [20], and thus cannot achieve optimal reconstruction quality. On the other hand, fully connected networks, suffer form a large search space, and can in practice only optimize a small repeated mask by preserving the essential connections. While repeated masks significantly reduce the scale of the optimization problem, they may also introduce structured artifacts during reconstruction.

Unrolling iterative optimization methods.

More recently, a class of networks constructed by unrolling iterative optimization methods has started to be used in image reconstruction (e.g. LISTA [21]ADMM-net [22], LDAMP [23], IRCNN [24], ISTA-Net [25]). Such network architectures combine the advantages of both model-based methods and deep discriminative learning methods, and provide an efficient and flexible plug-and-play framework to solve inverse problems. Previous works have utilized the multistage iterative network for image restoration [26] and illumination optimization [27]. In this paper, we claim that such networks are effective in jointly optimizing the sensing matrix and reconstruction method if the elements of the sensing matrix are treated as trainable parameters in the network. Crucially, we also show how to improve the design of such unrolled networks to embed Anderson acceleration directly into the network architecture. This improvement will be applicable and useful far beyond our specific application scenario.

Computational video cameras.

Many different prototype designs for computational video cameras have been proposed. Raskar *et al.* modified a conventional DSLR camera and added a control unit for high-speed control of the exposure pattern over the full frame. The camera can then be used for deblurring [28] and video compressive sensing [29]. Liu *et al.* used an LCoS to implement a single exposure mask and applied dictionary learning to reconstruct the scene [30]. To achieve a high-speed encoding, Bub *et al.* used a DMD for high-frame-rate imaging [31]. Llull *et al.* changed from active to passive codes to reduce the power consumption [4]. In their design, the static mask is spatially shifted over time, which provides a very limited design space for the spatio-temporal encoding.

Recently, several image sensor designs have been proposed that can implement the CS mask directly on the sensor. Luo *et al.* [32] invented a CMOS sensor that allows for active control of the exposure pattern in each pixel, and applied this design for image deblurring. Zhang *et al.* [33] a CMOS sensor for both high-speed and high-dynamic-range imaging [34]. However, since there is no charge bucket connect with PD, every pixel can only expose once during a frame. Sonoda *et al.* [35] built a sensor with quasi pixel-wise programmable control, but pixels on the sensor can only be controlled in blocks. Therefore, their camera cannot generate arbitrary mask patterns [36]. Sarhangnejad *et al.* [37] implemented a coded-exposure-pixel camera with two-bucket pixels that has 180 subframes per second. In this camera every pixel is programmable and can exposed many times during a single frame. Wei *et al.* use this system for a one-shot photometric stereo and develop an image formation model for computational video cameras [5].

3 METHOD

Our goal is to jointly learn both the full-resolution masks for encoding and the reconstruction method for decoding that together minimize subframe reconstruction error. We achieve this by training an end-to-end network that consist of K stages with dense skip connections and a mask layer, as shown in Fig. 4. Given a video sequence, the mask layer modulates each subframe using the learned mask and integrates all subframes into a single captured image; the K stages constructed via unrolling the optimization iterations for reconstruction can decode the captured images into multiple subframes.

In the following, we first present the encoding and decoding parts of our neural network architecture along with training details. Then we describe a set of simulations for comparing the proposed method with other existing methods. Lastly, we implement our approach on a real camera and evaluate the effectiveness of our network.

Image formation.

The image formation model for our compressive video capture system is shown in Fig.1, and can be formulated as:

$$\mathbf{Y} = \sum_{i=1}^{T} \boldsymbol{\phi}^{(i)} \odot \mathbf{X}^{(i)} + \mathbf{N}, \tag{1}$$

where $\boldsymbol{\phi}^{(i)} \in \mathbb{R}^{M \times N}$ denotes the i-th binary encoding mask, $\mathbf{X}^{(i)} \in \mathbb{R}^{M \times N}$ represents the i-th subframe we need to reconstruct, $\odot$ denotes the element-wise product, $\mathbf{N} \in \mathbb{R}^{M \times N}$ denotes the imaging noise, and $\mathbf{Y}$ is the $M \times N$ captured image. The system has a compression ratio of $1/T$, i.e. T successive subframes are encoded into a single captured image.

Eq. 1 can be transformed into the following equation:

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{n}, \tag{2}$$

where $\mathbf{\Phi} \in \mathbb{R}^{MN \times TMN}$ is the sensing matrix with diagonal blocks consisting of the masks $\boldsymbol{\phi}$:

$$\mathbf{\Phi} = [diag(Vec(\boldsymbol{\phi}^{(1)})), \cdots, diag(Vec(\boldsymbol{\phi}^{(T)}))], \tag{3}$$

$\mathbf{x}$ represents the $TMN \times 1$ vectorized subframes of $\mathbf{X}$, $\mathbf{y}$ is the $MN \times 1$ vectorized captured image of $\mathbf{Y}$, and $\mathbf{n}$ denotes the vectorized noise of $\mathbf{N}$.

3.1 Mask generation

A layer containing only bias values is constructed to generate the encoding masks ϕ. Since different pixels in the subframes are encoded independently, the operation $\mathbf{\Phi x}$ can be realized by an element-wise multiplication of ϕ and $\mathbf{X}$ and a summation of the multiplication results; the operation $\mathbf{\Phi}^T\mathbf{y}$ can be realized by a repeat copy operation of $\mathbf{Y}$ and an element-wise multiplication, as shown in Fig.3. The two operations are beneficial for efficient calculation, as well as reduced storage requirements. Since the masks used in high-frame-rate imaging are binary, we need to add a constraint that the outputs of the mask layer must be either 0 or 1 during propagation. Inspired by the Binaryconnect method [38], this can be achieved by a simple but efficient deterministic binarization operation:

$$\hat{b} = \begin{cases} 1, & \text{when } \ b > 0, \\ 0, & \text{else.} \end{cases}, \tag{4}$$

where $\hat{b}$ is the binarized value of the mask layer, and b is the real value. The sign function binarizes the values straightforwardly, however it is only activated during the forward and backward propagations but not during the parameter update since it is necessary to maintain good precision weights during the updates.

3.2 Subframe reconstruction

Unrolled network reconstruction.

To present the subframe reconstruction method, we first mathematically formulate the reconstruction procedure as an unconstrained problem, and then loop-unroll the optimization to construct our multi-stage network. Subframe reconstruction is an optimization problem

$$\arg\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi x}||^2 + \lambda J(\mathbf{x}), \tag{5}$$

where $J(\mathbf{x})$ is the denoising prior for regularization weighted by parameter λ. The first data fidelity term guarantees a minimal re-sensing error while the regularization term ¡¡¡¡¡¡ HEAD ensures that the reconstructed frames satisfy the desired prior model. Different from designed priors in model-based method, denoising prior depicts intrinsic statics of images and results in better image reconstruction. ======= ensures that the reconstructed subframes satisfy the desired prior model. Different from the hand-designed priors of model-based methods, the deep image prior captures the intrinsic statistics of images and results in better image reconstructions. ¿¿¿¿¿¿ a71d6251c8ad5f37b400c1e81a871974a6b63fec

By introducing an auxiliary variable $\mathbf{v}$, Eq. 5 can be reformulated as a constrained optimization problem:

$$(\mathbf{x}, \mathbf{v}) = \arg\min_{\mathbf{x},\mathbf{v}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi x}||^2 + \lambda J(\mathbf{v}), st.\ \mathbf{x} = \mathbf{v}. \tag{6}$$

Inspired by previous image restoration works [24], we adopt the half-quadratic splitting method to convert the constrained optimization problem into an unconstrained one:

$$(\mathbf{x}, \mathbf{v}) = \arg\min_{\mathbf{x},\mathbf{v}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi x}||^2 + \frac{\tau}{2}||\mathbf{x} - \mathbf{v}||^2 + \lambda J(\mathbf{v}), \tag{7}$$

where τ is a weight term. Then, Eq. 7 can be solved by alternatively optimizing the two sub-problems with respect to $\mathbf{z}$ and $\mathbf{x}$, respectively:

$$\begin{cases} \mathbf{x}^{i+1} & = \arg\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{\Phi x}||^2 + \frac{\tau}{2}||\mathbf{x} - \mathbf{v}^i||^2 \\ \mathbf{v}^{i+1} & = \arg\min_{\mathbf{v}} \frac{\tau}{2}||\mathbf{x}^{i+1} - \mathbf{v}||^2 + \lambda J(\mathbf{v}) \end{cases} \tag{8}$$

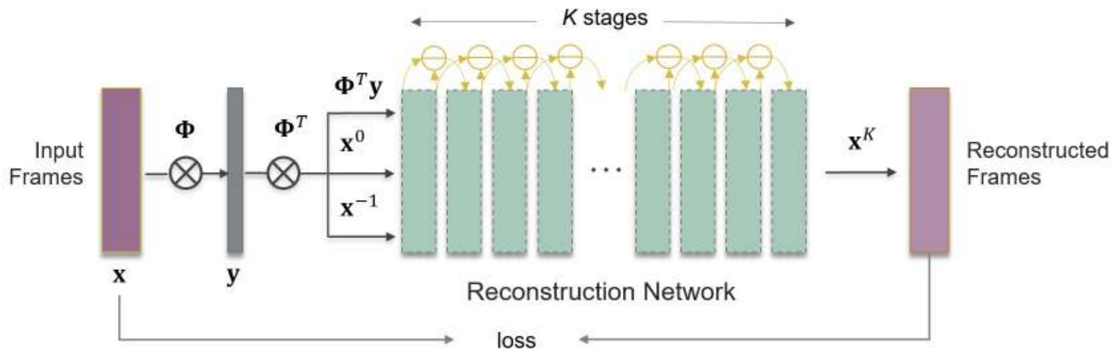


Fig. 2. Our deep network architecture. The overall network consists of a mask layer for generating masks and K stages for reconstruction. Note that the skip connections of residuals among stages make the network denser and more compact. (Here show is the case where the number of skip connections of each stage is $m = 1$.)

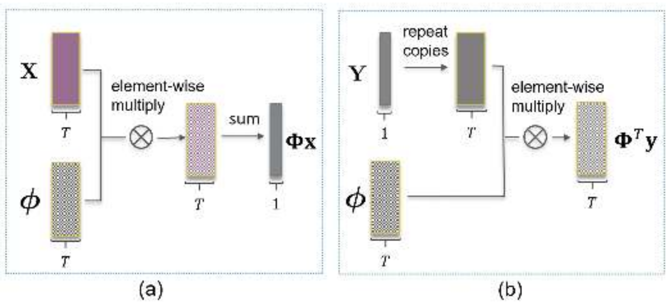


Fig. 3. Two matrix-vector multiplication operations: (a) $\mathbf{\Phi x}$ and (b) $\mathbf{\Phi}^T\mathbf{y}$.

By analyzing Eq. 8, it is evident that the optimization of $\mathbf{x}$ in the first line is a quadratic problem, while optimization of $\mathbf{v}$ in the second line is actually a denoising problem. To solve the first problem, we can calculate the closed-form solution

$$\mathbf{x}^{i+1} = (\mathbf{\Phi}^T\mathbf{\Phi} + \tau\mathbf{I})^{-1}(\mathbf{\Phi}^T\mathbf{y} + \tau\mathbf{v}^i). \tag{9}$$

However, the matrix inversion is time consuming. More importantly, such inverse models consisting of the trainable sensing matrix $\mathbf{\Phi}$ are harder to train, compared to a forward model of $\mathbf{\Phi}$. Previous work [26] suggests that using gradient descent algorithms to obtain an inexact solution in each step can also effectively and efficiently optimize the problem. In the general gradient descent method, the update step of $\mathbf{x}$ can be performed as:

$$\begin{aligned}\mathbf{x}^{i+1} &= \mathbf{x}^i - \alpha^i g(\mathbf{x}^i) \\ &= \mathbf{x}^i - \alpha^i(\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{x}^i - \mathbf{\Phi}^T\mathbf{y} + \tau(\mathbf{x}^i - \mathbf{v}^i))\end{aligned} \tag{10}$$

where $g(.)$ is the gradient function of $\mathbf{x}$, and α^i is the length of the gradient descent step.

Anderson acceleration

Many efforts have been devoted to developing acceleration methods for the gradient descent algorithm [39]. For example, the wildly used Momentum acceleration method takes into account the previous gradients in the update step at each iteration [40]; Anderson acceleration uses the residuals of previous m iterations to adjust the current iteration point [7]. We claim that acceleration methods not only speed up convergence but can also inform the *network's architecture*. Specifically, we use the general acceleration form:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \sum_{j=1}^{m'} w_j^i \mathbf{d}^{i-j} - \alpha_i g(\mathbf{x}^i - \sum_{j=1}^{m'} w_j^i \mathbf{d}^{i-j}), \tag{11}$$

where $\mathbf{d}^{i-j}$ is the descent direction in the j-th iteration prior to iteration i, and w_j^i is the weight of the descent direction in iteration i. We choose $m' = \min(m, i)$ to ensure that $i - m'$ is a non-negative integer in the early layers.

Note that the form of Eq. 11 is exactly that of Anderson acceleration [7], [41], except that the parameters of Anderson acceleration are manually estimated while ours are learned from the network. Specifically, when $m = 1$, our acceleration becomes Nesterov's accelerated gradient method [42].

Since the norm of the residual in each iteration can be absorbed by its weights w_j^i, without loss of generality, we directly let

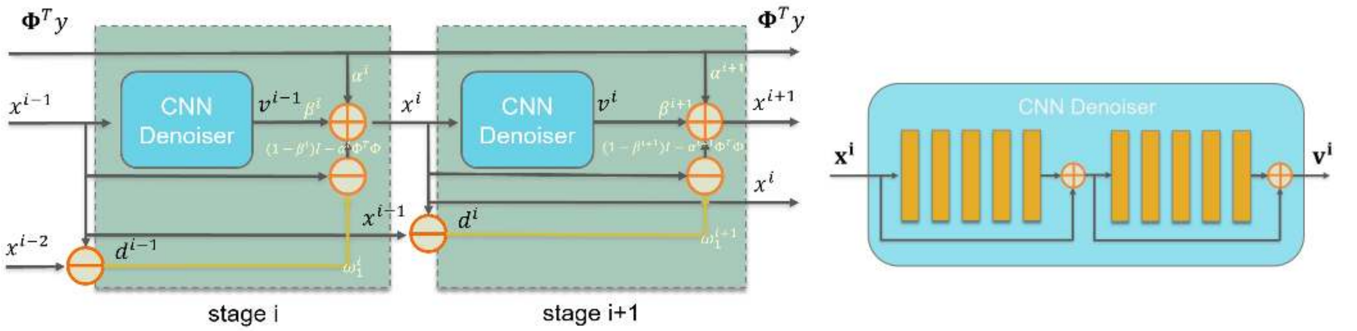
$$\mathbf{d}^i = \mathbf{x}^i - \mathbf{x}^{i-1}. \tag{12}$$

Combining Eq. 11 and the definition of $g(.)$ in Eq. 10, the update step of $\mathbf{x}$ can be rewritten as:

$$\mathbf{x}^{i+1} = [(1-\beta^i)\mathbf{I} - \alpha^i\mathbf{\Phi}^T\mathbf{\Phi}](\mathbf{x}^i - \sum_{j=1}^{m'} w_j^i\mathbf{d}^{i-j}) + \alpha^i\mathbf{\Phi}^T\mathbf{y} + \beta^i\mathbf{v}^i, \tag{13}$$

where $\alpha^i\tau$ is denoted as β^i. We show the detailed operations and connections in and between stages in Fig.4 (a). Compared to general unrolling networks, the skip connections between stages in our model make the network denser and more compact, and transform it from a Resnet to a Densenet.

The denoising network we used to solve the second subproblem in Eq 8 consists of two cascaded residual blocks. The architecture of the denoising network is as shown in Fig. 4 (b). The number of used residual blocks is chosen empirically. Previous work [43] gave some convergence analysis and also showed that two residual blocks provide the best results for learning the proximal operator. Note that we can also apply non-local attention [44] and a multi-scale architecture [45], [46]. But to ensure the decoding network has a limited parameter count to prevent overfitting, each residual block in the denoising network contains only five convolutional layers, and all layers generate feature maps with 3×3 kernels.



(a) (b)

Fig. 4. (a)Illustration of two stages in our network.(here we show the case $m = 1$) (b) The architecture of our denoising network.

Algorithm 1 Accelerated subframe reconstruction

Input: Sensing matrix Φ, captured image $\mathbf{y}$, number m
Output: Reconstructed subframes $\mathbf{x}$

Initialize $\mathbf{x}^0 = \mathbf{\Phi}^T\mathbf{y}$, $\mathbf{x}^{-1} = \mathbf{x}^0$ $(i = 1, ..., m)$, $\mathbf{d}^0 = 0$
for $i = 1, 2, \ldots, K$ **do**
  $\mathbf{v}^{i-1} = D(\mathbf{x}^{i-1})$
  $m' = min(m, i)$
  $\mathbf{z}^i = \mathbf{x}^{i-1} - \sum_{j=1}^{m'} w_j^i \mathbf{d}^{i-j}$
  $\mathbf{x}^i = [(1 - \beta^i)\mathbf{I} - \alpha^i \mathbf{\Phi}^T \mathbf{\Phi}]\mathbf{z}^i + \alpha^i \mathbf{\Phi^T y} + \beta^i \mathbf{v}^{i-1}$
  $\mathbf{d}^i = \mathbf{x}^i - \mathbf{x}^{i-1}$
end for

3.3 Training

We constructed an end-end network by unrolling the algorithm shown in Algorithm 1. The proposed model mainly consists of a mask layer and a K-stage reconstruction network using convolutional layers. The input subframes $\mathbf{x}$ are encoded using a trainable mask layer ϕ. We multiply the transpose of the mask $\mathbf{\Phi}^T$ and the captured images $\mathbf{y}$ to generate an initial guess $\mathbf{x}^0 = \mathbf{\Phi}^T\mathbf{y}$. We then feed the initial image into the reconstruction. All layers use ReLU as their activation function, except the output layer, which uses a sigmoid. We choose the mean square error (MSE) as our loss function, expressed as

$$\mathcal{L}(\boldsymbol{\phi}, w; \alpha; \beta; \theta) = \frac{1}{k}\sum_{i=1}^{k} ||f(\mathbf{x}; \boldsymbol{\phi}; w; \alpha; \beta; \theta) - \mathbf{x}||^2, \quad (14)$$

where k is the number of the training samples, θ are the denoising network weights, ϕ are the mask layer weights, and $(w; \alpha; \beta)$ are the optimization parameters. We trained the proposed network to learn these parameters simultaneously. The parameters of each stage are set to be different, and the α are set to be channel-wise.

The model was trained on an Intel Xeon E5 workstation with an NVIDIA GeForce RTX 2080 Ti GPU and 512 GB main memory. Our network is implemented using Keras 2.2.5 and trained using the Adam optimizer [47]. The initial learning rate is set to 10^{-4} and decayed by a factor of 10 at the 20th iteration. We train the model for 80 iterations with a batch size of 1, which takes about two days to complete.

4 Simulations

In this section, we conduct numerical simulations to show the effectiveness of our proposed network and compare our method with other state-of-the-art compressive reconstruction methods.

Datasets and Training. The data we used for the simulations are two popular databases: the SumMe database from https://gyglim.github.io/me/vsum/index.html [48] and the "Sports Videos in the Wild" database from http://cvlab.cse.msu.edu/project-svw.html [49]. We randomly cropped and selected 3,000 video sequences of size $256 \times 256 \times 32$ to train our network, and selected 800 video sequences of the same size for testing.

TABLE 1
Ablation Studies.

Methods	Noiseless		Noisy ($\sigma = 0.01$)	
	PSNR	SSIM	PSNR	SSIM
Unopt [26]	30.68	0.896	28.52	0.861
Opt	32.35	0.921	30.52	0.897
Opt + SC (m=1)	33.18	0.930	31.24	0.905
Opt + SC (m=2)	33.30	0.932	31.43	0.908
Opt + SC (m=3)	33.32	0.932	31.46	0.909

Ablation studies. To clearly understand the effect of each component as well as choosing an appropriate m in our end-to-end network, we carried out five ablation simulations. We present our observations and quantitative results in Table 1. For all the simulations in the ablation study, we used the architecture shown in Fig. 4 with 39 stages for frame reconstruction, and calculated the average PSNR and SSIM of the reconstructed results in the presence and absence of noise. The baseline for comparison is model Unopt, a multistage network without mask optimization and dense skip connections, which is the same network architecture as in previous work [26]. Compared to this baseline, our method leads to a significant improvement in reconstruction quality as well as to a reduction of the number of training epochs needed for the same accuracy.

Optimized vs. fixed mask: For the Unopt model, we used a randomly shifted Bernoulli binary masks as shown in Fig. 5(a) while in other Opt models we used optimized

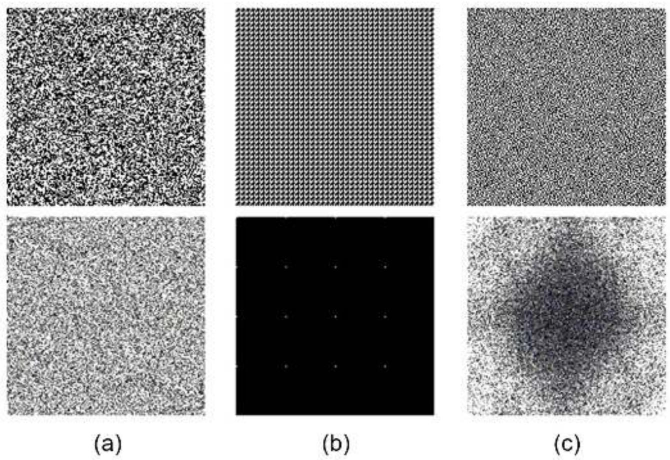


Fig. 5. The comparison of the first binary pattern(upper) and their spectrum distribution(bottom) of the three used masks sequences. (a) Bernoulli pattern used in [50] and [8]. (b) Optimized repeated pattern of [18]. (c) Our optimized pattern. Note that the patterns were cropped into 160×160 for visulization.

masks as shown in Fig. 5(c). PSNRs can be improved by nearly 1dB when replacing the random masks by the optimized masks. It is worth noting that the loss of Unopt is relatively low in the initial few epochs since random Bernoulli masks are suitable for compressive reconstruction [51]. However, Opt models catch up with and surpass the Unopt model as the number of epochs increases, as shown in Fig. 7. The results indicate that our network has learned more efficient masks after several epochs of training.

Skip connections (SC) vs. no skip-connections: We tested the effect of skip connections in our network. It is obvious that skip connections can enhance reconstruction quality and accelerate the convergence of training loss. The PSNRs are improved by nearly 1dB when three skip connections for a single stage ($m = 3$) are applied. However, denser skip connections require more memory, so we need to choose an appropriate m for the best trade-off between memory consumption and reconstruction accuracy. As shown in Table 1, the model with $m = 3$ outperforms the one with $m = 2$, but only by a small margin in both PSNR and SSIM. Therefore, we choose $m = 2$ as an empirical setting for our reconstruction network.

Comparison methods. We compared the proposed method with two representative DNN-based methods: DeepMask [18] and Deep Tensor ADMM-Net (DTAN) [50]; and two state-of-the-art traditional methods: GAP-TV [8] and GMM [10]. Following previous literature, we used masks to modulated every eighth consecutive frame. Thus we reconstructed 32 subframes from 4 measurements in the simulations. To be specific, DeepMask is the only existing method which can jointly optimize masks and reconstruction method; it learns $4 \times 4 \times 8$ repeated masks for encoding and reconstructs frames via a fully-connected network. The other three methods use a $256 \times 256 \times 8$ shifting Bernoulli binary masks. The masks of different methods and their frequency spectra are shown in Fig.5. It can be observed that our masks perform as a 'high-pass filter' that blocks low-frequency spatial content.

Quantitative results. The PSNR and SSIM results of different methods with different masks are shown in Table 2. As an optimization method, GAP-TV is effective and efficient in reconstructing subframes, but the reconstruction quality is not competitive compared to ours due to the used handcrafted priors. The GMM approach reconstructs frames patch-by-patch, and also cannot produce competitive results. To our surprise, DTAN performs worst among these methods, although it works well on its 'NBA' dataset. This might be because the non-local low-rank prior fails in reconstructing spatial high-frequency content. Due to the joint design of masks and reconstruction, the average PSNR and SSIM of DeepMask exceed 31dB and 0.9, respectively. However, we finds serious structured artifacts in the reconstructed images of DeepMask (see Fig.6) caused by the use of repeated masks. Our method outperforms state-of-art methods by more than 2.2dB in PSNR and more than 0.03 in SSIM. This is further confirmed by visual comparison of the reconstructed images in Fig. 6, where we show ground truth and the reconstructed results of four frames. Our method generates much more visually pleasant images with more accurate detail information. We also compared our method with GAP-TV, GMM, and DeepMask with T=32 subframes. In this simulation, 64 frames are reconstructed from two encoded images. The results are shown in ¡¡¡¡¡¡¡ HEAD Table3. Compression ratios of 1:32 are very ======= Table 3. Compression ratios of 1:32 are very ¿¿¿¿¿¿¿ a71d6251c8ad5f37b400c1e81a871974a6b63fec challenging for compressive sensing algorithms in general, so the results are worse than for 8 subframes, however our

TABLE 2
The comparison of reconstruction quality of the five methods with T=8 subframes.

Methods	Noiseless		Noisy($\sigma = 0.01$)	
	PSNR	SSIM	PSNR	SSIM
GAP-TV [8] + random	29.82	0.857	27.99	0.835
GAP-TV + optimized	30.72	0.884	29.04	0.843
GMM [10] + random	27.24	0.797	27.00	0.774
GMM + optimized	27.35	0.807	27.10	0.785
DTAN [50] + random	26.08	0.803	25.12	0.799
DTAN + optimized	27.28	0.816	26.45	0.813
DeepMask [18]	31.05	0.905	29.28	0.882
Ours	**33.32**	**0.932**	**31.43**	**0.908**

TABLE 3
The comparison of reconstruction quality of the four methods with T=32 subframes.

Methods	Noiseless		Noisy($\sigma = 0.01$)	
	PSNR	SSIM	PSNR	SSIM
GAP-TV [8] + random	23.44	0.725	23.15	0.700
GMM [10] + random	22.19	0.589	22.16	0.583
DeepMask [18]	27.58	0.814	25.46	0.792
Ours	**28.01**	**0.840**	**26.15**	**0.810**

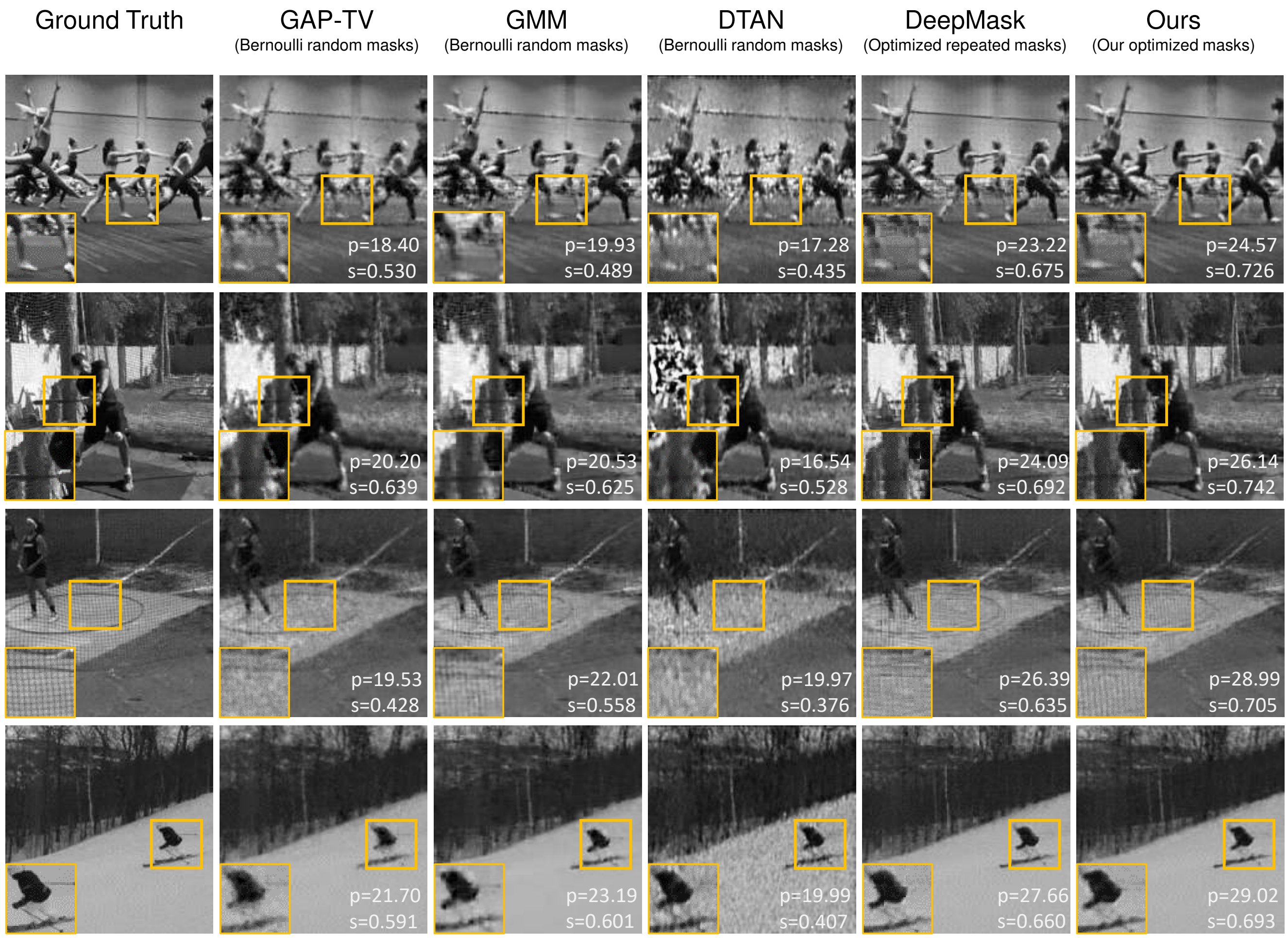


Fig. 6. The comparison of reconstructed frames and the statistics on the PSNR and SSIM. From top to bottom: ground truth;reconstructed results of GAP-TV, GMM, Deep Tensor Admm-net, DeepMask, and ours.

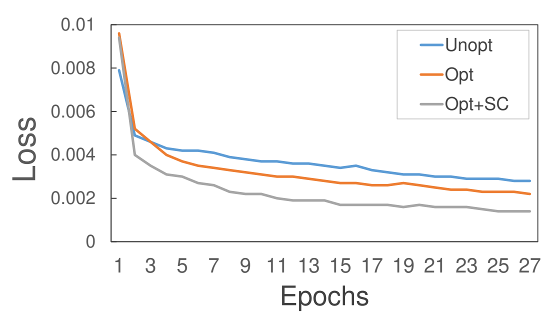


Fig. 7. Training loss vs number of epochs on the neural network models in ablation study.

approach still dominates the comparison methods.

Mask evaluation. We also evaluated our optimized mask by comparing it with random masks using the same reconstruction method. Since GAP-TV is a model-based optimization method which does not memorize data, we reconstructed frames using GAP-TV with random masks and our proposed masks respectively to present the behavior of the two masks. Fig. 8 shows the reconstructed results. The frames reconstructed from the image encoded by our masks are significantly better than those by random masks, especially around the edges. We also observed the improvement brought by the optimized masks using other existing methods (e.g. [10] and [50]). Note that the improvement for the GMM method is not significant since it reconstructs frames patch-by-patch while our masks are optimized as a whole.

5 Real Experiments

Previous work on mask-based video compressive sensing uses either a static mask that is shifted over time, or a setup with some form of spatial light modulator, such as a DMD or LCOS, which can be controlled with high temporal resolution. However, the drawback of these methods is that they are difficult to align and rather bulky due to the need for re-imaging optics [52].

Fortunately, recent developments in image sensor technology allow us to directly implement the CS mask on the sensor itself. Specifically, there are now several prototypes of image sensors with per-pixel programmable exposure control [5], [37]. In this paper, we use the Coded two-Bucket (C2B) camera from Wei *et al.* [5]. In this camera, each pixel has two charge-collection sites (i.e. two buckets). The exposure control signal for each pixel can select which of the two buckets integrates incident light at any given point in

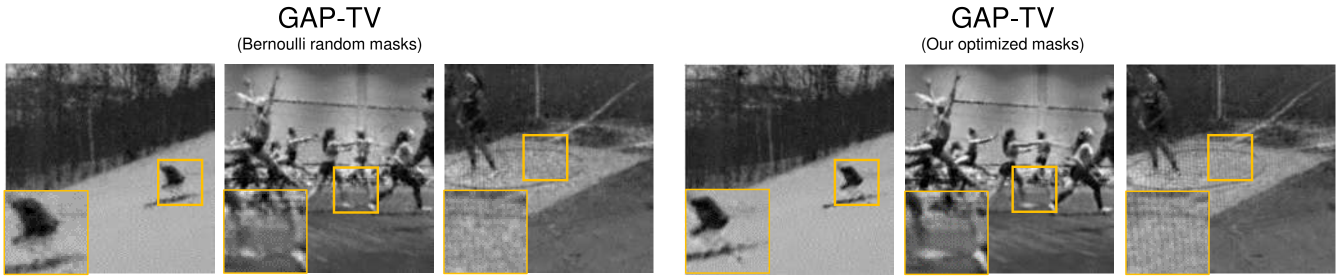


Fig. 8. The comparison of reconstructed results using GAP-TV method with different encoding masks.

time. The major advantage of this design is that it makes use of all incident photons and simultaneously encodes subframes with a pair of complementary masks. Using this camera, subframes are reconstructed from the pair of captured complementary images. The spatial resolution of the camera is 312×320, and the frame rate can reach 30 frames per second with over 100 different masks per frame. In our experiments we use only up to 32 masks per frame since a compression ration of 1:32 is already extremely challenging for all compressive sensing approaches.

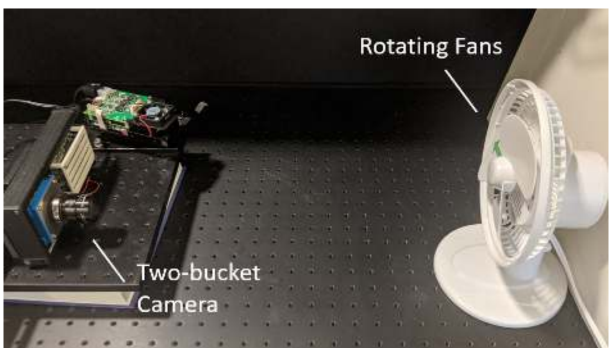


Fig. 9. The setup of our experiments.

We captured several dynamic scenes using the camera to compare the reconstruction quality of four different methods: GAP-TV [8], GMM [10], DeepMask [18], and ours. The setup for our experiments is shown in Fig. 9. Unlike the simulation, here, the number of subframes we used is 32 to explore the limits of the four methods; thus, a high-frame-rate ($32 \times 30 = 960$) imaging can be achieved. In the experiment, the first two methods used $312 \times 320 \times 32$ random masks, DeepMask used optimized repeated $4 \times 4 \times 32$ masks, and our method used $312 \times 320 \times 32$ optimized masks. We reconstructed 64 subframes from two successively captured images. Fig. 10 shows two examples of the reconstructed results. It can be seen that the GAP-TV method created watercolor-like artifacts due to the drawbacks of the handcrafted prior; GMM and DeepMask introduced significant structured artifacts in the patch-by-patch reconstruction. The proposed methods, on the other hand, can produce better results with fewer artifacts, clearer contents, and higher contrast compared with the other three methods (please zoom in for details).

We also investigated the improvement brought by the two bucket mechanism of the camera. With the two-bucket mechanism each subframe is encoded by a pair of complementary masks, so that the number of measurements is doubled when compared to the one-bucket mechanism. To demonstrate the improvements due to thex two-bucket design, we captured a fan with varying rotation speeds and reconstructed 64 subframes from two one-bucket images and four two-bucket images respectively. The results are shown in Fig.11. It can be seen that the reconstructed results from two-bucket images are significantly better than those from one-bucket images. We can also observe that the advantages of our method over the state of the art are even more compelling in real experiments than in simulation. That is because our method depends on a deep image prior rather than handcrafted priors and thus can better handle complicated video content found in real scenes.

6 Conclusion and Future Works

We have presented a new end-to-end learned method and prototype system for video reconstruction from mask-based compressive sensing cameras. Unlike existing approaches, the proposed method is suited for optimizing full-resolution masks, and can reconstruct subframes efficiently. The reconstruction quality of the proposed method significantly outperforms that of previous methods due to the utilized deep image prior. We implemented a two-bucket camera for high-frame-rate imaging; the frame rate can reach close to 1,000 frames with superior image quality compared to other CS video approaches.

In addition to providing a superior solution to the compressive sensing video reconstruction problem, we also make a fundamental improvement to loop-unrolled neural network architectures for image reconstruction problems in general: we demonstrate that dense skip connections can implement Anderson acceleration directly in the neural network to make it compact and efficient. The proposed dense network is not limited to CS problems, but can be applied to solve other inverse problems directly.

We believe that the frames in the near future can be predicted from previously reconstructed frames. Therefore, in future work, we plan to explore more efficient frame reconstruction and adaptively optimize masks in real-time for even better results.

References

[1] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

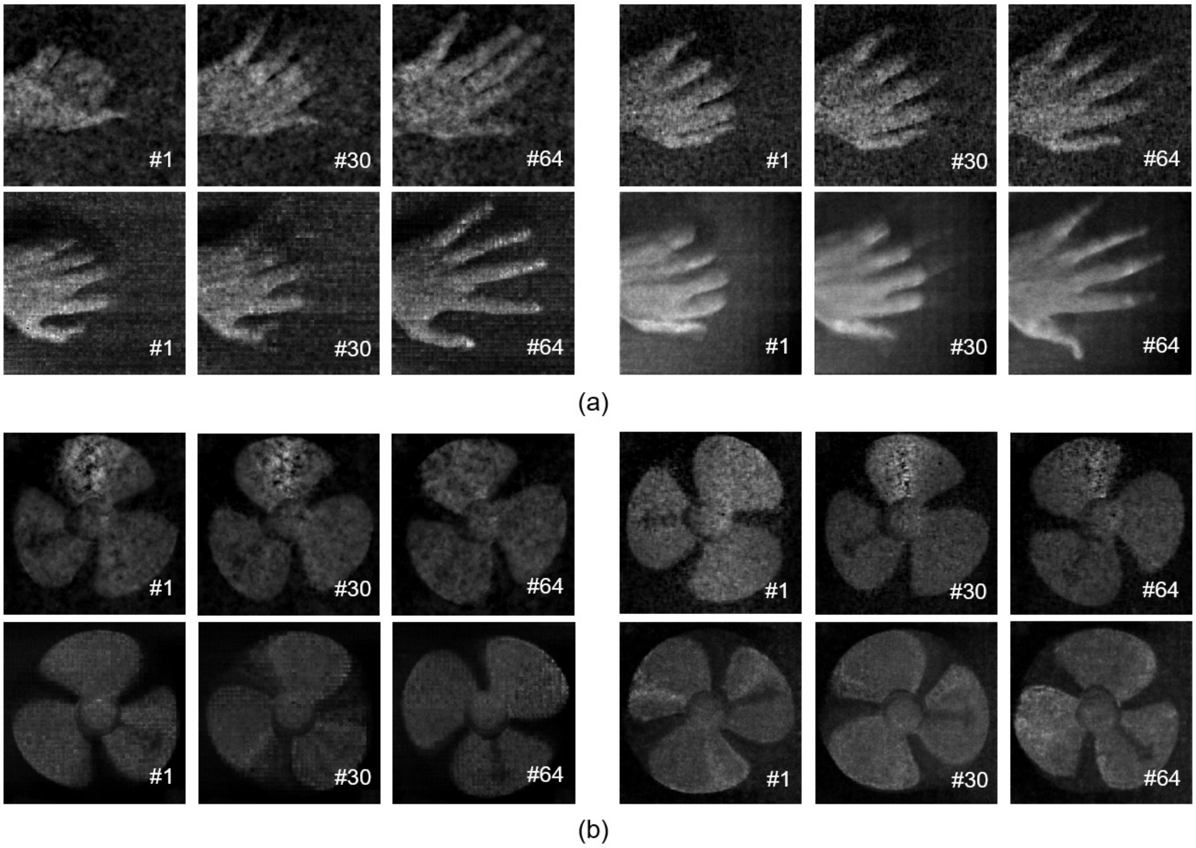


Fig. 10. The reconstructed results of (a)an opening hand and (b) a rotating fans using four methods. Left-top: GAP-TV; Right-top: GMM; Left-bottom: DeepMask; Right-bottom: our method. Here shows the 1st, 30th, and the 64th subframes reconstructed from two one-bucket images. The rotating speed of the fans is 2.5 rounds pre second. Note thayt the reconstructed subframes are scaled by the maximum intensity for visualization.

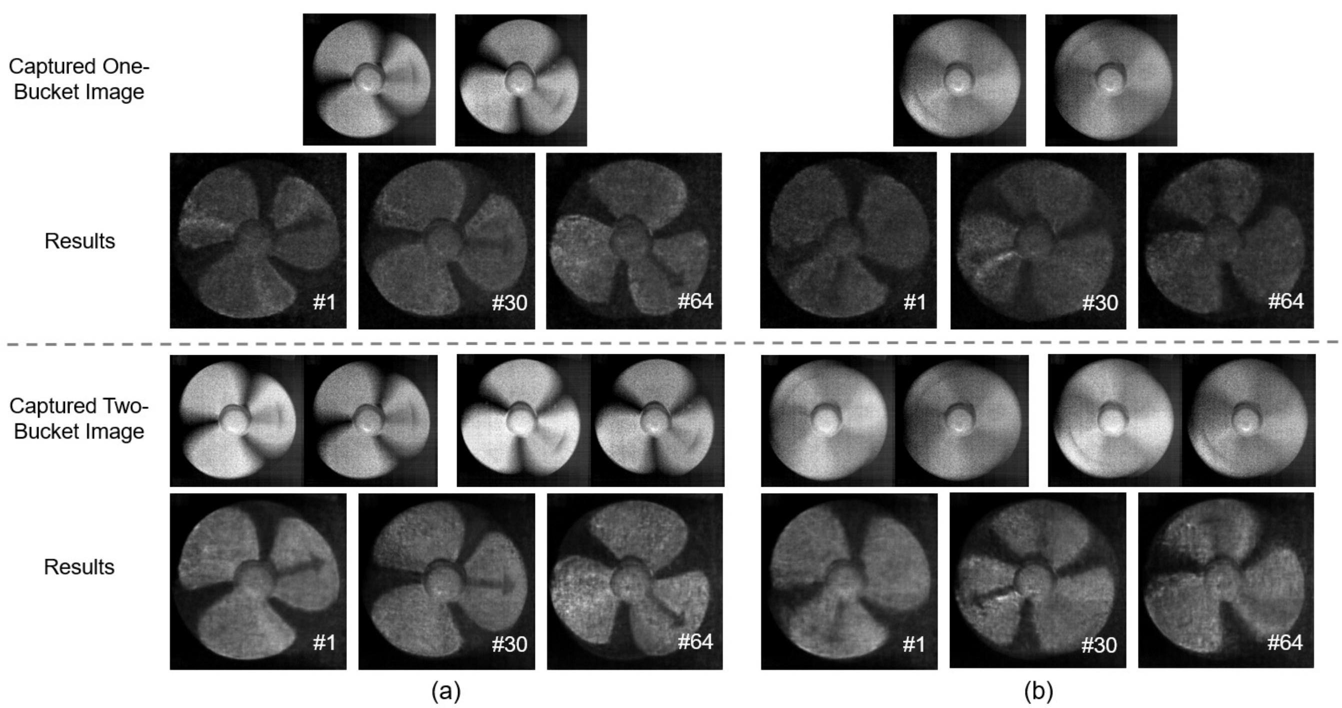


Fig. 11. The 1st, 30th, and 64th subframes of a rotating fans reconstructed from two one-bucket encoded images and four two-bucket encoded images. The fans are captured under the rotating speeds (a) 2.5 rounds and (b) 7 rounds per second. Note that our method can reconstruct clear results from the two-bucket encoded images with heavy motion-blur.

[2] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing mri," *IEEE signal processing magazine*, vol. 25, no. 2, p. 72, 2008.

[3] A. Wagadarikar, R. John, R. Willett, and D. Brady, "Single disperser design for coded aperture snapshot spectral imaging," *Appl. Opt.*, vol. 47, no. 10, pp. B44–B51, Apr 2008.

[4] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, "Coded aperture compressive temporal imaging," *Optics Express*, vol. 21, no. 9, p. 10526, May 2013. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=oe-21-9-10526

[5] M. Wei, N. Sarhangnejad, Z. Xia, N. Gusev, N. Katic, R. Genov, and K. N. Kutulakos, "Coded two-bucket cameras for computer vision," in *Computer Vision ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Springer International Publishing, 2018, vol. 11207, pp. 55–73. [Online]. Available: http://link.springer.com/10.1007/978-3-030-01219-9_4

[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*, June 2010, pp. 455–460.

[7] H. F. Walker and P. Ni, "Anderson acceleration for fixed-point iterations," *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1715–1735, 2011.

[8] X. Yuan, "Generalized alternating projection based total variation minimization for compressive sensing," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 2539–2543.

[9] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2c2: Programmable pixel compressive camera for high speed imaging," in *CVPR 2011*, June 2011, pp. 329–336.

[10] J. Yang, X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro, and L. Carin, "Video compressive sensing using gaussian mixture models," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4863–4878, Nov 2014.

[11] 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*, Nov 2011, pp. 287–294.

[12] D. Liu, J. Gu, Y. Hitomi, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Efficient space-time sampling with pixel-wise coded exposure for high-speed imaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 248–260, Feb 2014.

[13] Y. Liu, X. Yuan, J. Suo, D. Brady, and Q. Dai, "Rank minimization for snapshot compressive imaging," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[14] R. Obermeier and J. A. Martinez-Lorenzo, "Sensing matrix design via mutual coherence minimization for electromagnetic compressive imaging applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 2, pp. 217–229, June 2017.

[15] S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, and A. Ashok, "Convolutional neural networks for noniterative reconstruction of compressively sensed images," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 326–340, Sep. 2018.

[16] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, "Dr2-net: Deep residual reconstruction network for image compressive sensing," *Neurocomputing*, vol. 359, pp. 483 – 493, 2019.

[17] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 449–458. [Online]. Available: http://ieeexplore.ieee.org/document/7780424/

[18] M. Iliadis, L. Spinoulas, and A. Katsaggelos, "Deep fully-connected networks for video compressive sensing," *Digital Signal Processing: A Review Journal*, vol. 72, pp. 9–18, 1 2018.

[19] M. Yoshida, A. Torii, M. Okutomi, K. Endo, Y. Sugiyama, R.-i. Taniguchi, and H. Nagahara, "Joint optimization for compressive video sensing and reconstruction under hardware constraints," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 649–663.

[20] J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep Tensor ADMM-Net for Snapshot Compressive Imaging," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, p. 10.

[21] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.

[22] J. Sun, H. Li, Z. Xu *et al.*, "Deep admm-net for compressive sensing mri," in *Advances in neural information processing systems*, 2016, pp. 10–18.

[23] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned d-amp: Principled neural network based compressive image recovery," in *Advances in Neural Information Processing Systems*, 2017, pp. 1772–1783.

[24] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.

[25] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1828–1837.

[26] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2305–2318, 2018.

[27] M. Kellman, E. Bostan, N. Repina, and L. Waller, "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging," *IEEE Transactions on Computational Imaging*, 2019.

[28] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: motion deblurring using fluttered shutter," in *ACM transactions on graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 795–804.

[29] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded strobing photography: Compressive sensing of high speed periodic videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 671–686, 2010.

[30] D. Liu, J. Gu, Y. Hitomi, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Efficient Space-Time Sampling with Pixel-Wise Coded Exposure for High-Speed Imaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 248–260, Feb. 2014.

[31] G. Bub, M. Tecza, M. Helmes, P. Lee, and P. Kohl, "Temporal pixel multiplexing for simultaneous high-speed, high-resolution imaging," *Nature methods*, vol. 7, no. 3, p. 209, 2010.

[32] Y. Luo, D. Ho, and S. Mirabbasi, "Exposure-programmable cmos pixel with selective charge storage and code memory for computational imaging," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 5, pp. 1555–1566, 2017.

[33] J. Zhang, T. Xiong, T. Tran, S. Chin, and R. Etienne-Cummings, "Compact all-cmos spatiotemporal compressive sensing video camera with pixel-wise coded exposure," *Optics express*, vol. 24, no. 8, pp. 9013–9024, 2016.

[34] J. P. Newman, X. Wang, C. S. Thakur, J. Rattray, R. Etienne-Cummings, M. A. Wilson *et al.*, "A closed-loop all-electronic pixel-wise adaptive imaging system for high dynamic range video," *arXiv preprint arXiv:1906.10045*, 2019.

[35] T. Sonoda, H. Nagahara, K. Endo, Y. Sugiyama, and R.-i. Taniguchi, "High-speed imaging using cmos image sensor with quasi pixel-wise exposure," in *2016 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2016, pp. 1–11.

[36] M. Yoshida, A. Torii, M. Okutomi, K. Endo, Y. Sugiyama, R.-i. Taniguchi, and H. Nagahara, "Joint optimization for compressive video sensing and reconstruction under hardware constraints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 634–649.

[37] N. Sarhangnejad, N. Katic, Z. Xia, M. Wei, N. Gusev, G. Dutta, R. Gulve, H. Haim, M. M. Garcia, D. Stoppa *et al.*, "5.5 dual-tap pipelined-code-memory coded-exposure-pixel cmos image sensor for multi-exposure single-frame computational imaging," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 102–104.

[38] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3123–3131.

[39] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[40] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145 – 151, 1999.

[41] J. Zhang, Y. Peng, W. Ouyang, and B. Deng, "Accelerating admm

for efficient simulation and optimization," in *Siggraph Asia*, Nov. 2019.

[42] Y. Nesterov, "A method for solving a convex programming problem with convergence rate o(1/k2)," *Soviet Mathematics Doklady*, no. 27, pp. 372–367, 1983.

[43] M. Mardani, Q. Sun, D. Donoho, V. Papyan, H. Monajemi, S. Vasanawala, and J. Pauly, "Neural proximal gradient descent for compressive imaging," in *Advances in Neural Information Processing Systems*, 2018, pp. 9573–9583.

[44] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019.

[45] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891.

[46] T. Rott Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image," in *Computer Vision (ICCV), IEEE International Conference on*, 2019.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *ECCV*, 2014.

[49] S. M. Safdarnejad, X. Liu, L. Udpa, B. Andrus, J. Wood, and D. Craven, "Sports videos in the wild (svw): A video dataset for sports analysis," in *Proc. International Conference on Automatic Face and Gesture Recognition*, Ljubljana, Slovenia, May 2015.

[50] J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep tensor admm-net for snapshot compressive imaging," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 223–10 232.

[51] 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*. IEEE, 2010, pp. 455–460.

[52] Q. Sun, X. Dun, Y. Peng, and W. Heidrich, "Depth and transient imaging with compressive spad array cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 273–282.