A Unified Framework for Compressive Video Recovery from Coded Exposure Techniques

Prasan Shedligeri*
Dept. of EE, IIT Madras
Chennai, India
ee16d409@ee.iitm.ac.in

Anupama S*
Qualcomm India
Bangalore, India

Kaushik Mitra
Dept. of EE, IIT Madras
Chennai, India
kmitra@ee.iitm.ac.in

Abstract

Several coded exposure techniques have been proposed for acquiring high frame rate videos at low bandwidth. Most recently, a Coded-2-Bucket camera has been proposed that can acquire two compressed measurements in a single exposure, unlike previously proposed coded exposure techniques, which can acquire only a single measurement. Although two measurements are better than one for an effective video recovery, we are yet unaware of the clear advantage of two measurements, either quantitatively or qualitatively. Here, we propose a unified learning-based framework to make such a qualitative and quantitative comparison between those which capture only a single coded image (Flutter Shutter, Pixel-wise coded exposure) and those that capture two measurements per exposure (C2B). Our learning-based framework consists of a shift-variant convolutional layer followed by a fully convolutional deep neural network. Our proposed unified framework achieves the state of the art reconstructions in all three sensing techniques. Further analysis shows that when most scene points are static, the C2B sensor has a significant advantage over acquiring a single pixel-wise coded measurement. However, when most scene points undergo motion, the C2B sensor has only a marginal benefit over the single pixel-wise coded exposure measurement.

1. Introduction

Cameras that can acquire high frame rate videos require high light sensitivity and massive data bandwidth increasing their cost significantly. Hence, several methods have been proposed to first acquire a low frame rate video from a low-cost camera and computationally upsample the videos temporally [6, 20, 21, 11]. Computational imaging techniques have used compressive sensing theory to first acquire compressed video measurements at low bandwidth and then computationally reconstruct the high frame rate video signal [3]. For visible light compressive video sensing, coded exposure techniques are the most popular ones with several compressive acquisition systems and reconstruction algorithms proposed over the years [27, 4, 28, 7, 16, 15, 9, 39, 10, 17, 14].

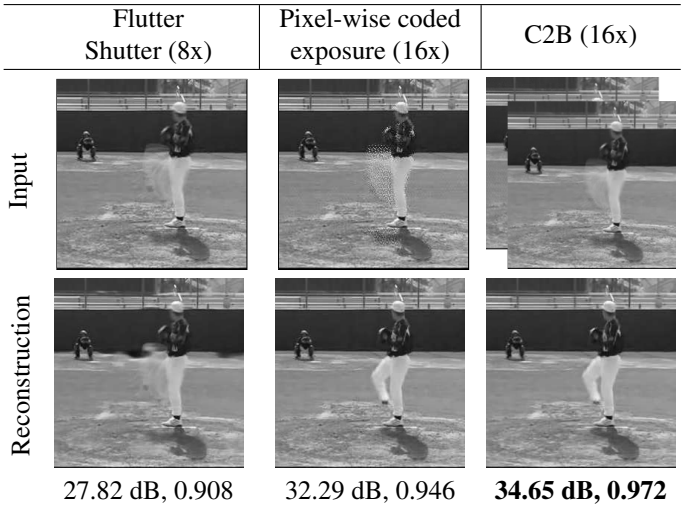

Figure 1: We propose a unified deep learning-based framework that allows us to compare the performance of various coded exposure techniques. The figure shows the input and the middle frame of the reconstructed video for each of the exposure techniques.

In coded exposure techniques, a pre-determined code is used to multiplex the temporal dimension of the video signal into compressed measurements. Recently, a novel prototype sensor based on multi-bucket pixels named Coded-2-Bucket sensor [30, 36] was introduced. While allowing for per-pixel control of the "shutter", this sensor also can acquire two compressed measurements per exposure using the 2 light-collecting buckets per pixel. Based on the number of measurements acquired per exposure, we can classify these sensing techniques into two categories: a) single compressed measurement (such as flutter shutter and pixel-wise coding) [7, 27, 16, 28, 15, 9, 39, 10, 17, 14] and b) two com-

*Equal contribution

pressed measurements per exposure [30]. It is expected that two measurements should lead to better video reconstruction quality compared to a single measurement. However, the performance improvement provided by two compressed measurements over a single compressed measurement is yet to be investigated. As the C2B sensor is recently introduced, no previous algorithm exists, making a quantitative comparison between the single and two compressed measurement techniques. While [14] uses C2B, only a qualitative comparison on a single video sequence is made for single and two measurement cases. An extensive quantitative and qualitative comparison has not been made, and it can help determine how much advantage is gained by acquiring two measurements over just one. This comparison of the different sensing architectures will also provide users with a tool to determine which sensing technique is better for a given scenario.

With this objective, we propose a unified learning-based framework with which we wish to make an extensive evaluation. This learning based framework should be usable for recovering videos from various single and two measurement techniques, particularly, Flutter-Shutter [27], pixel-wise coded exposure [33, 5] and C2B [30]. Most of the previously proposed algorithms for compressive video recovery use fully connected networks, and, ideally, we can use any of those networks for our framework. However, fully connected networks have fallen out of favour for most image processing tasks as they have a large number of trainable parameters and are also hard to scale up for large spatial/temporal resolutions. Hence, we design our framework to be fully-convolutional, enabling reconstruction of the full resolution video sequence in a single forward pass. In [17] it has also been demonstrated that a fully convolutional network provides better reconstruction results than fully connected networks. Later, we provide an intuitive explanation for why a convolutional network with local spatial connectivity is actually more suitable for this problem than fully connected networks with global connectivity. Our framework also uses the recently proposed *shift-variant* convolutional (SVC) layer [22] that has shown to be effective for feature extraction from a coded image input.

The proposed algorithm is divided into two stages, where the first stage uses the SVC layer for an exposure code aware feature extraction. In the second stage, a deep, fully convolutional neural network is used to learn the non-linear mapping to the full resolution video sequence. Extensive comparisons show that our proposed learning framework provides state of the art results on all three sensing techniques. Using our unified framework, we quantitatively evaluate the performance of the various coded exposure techniques. As expected, pixel-wise coded exposure techniques produce much better video reconstructions than global coded exposure technique such as FS. We also confirm that acquiring two compressed measurements as in C2B is better than capturing just a single compressed measurement. The advantage is significant for a largely stationary scene (Fig. 5). However, C2B is only marginally beneficial over a single pixel-wise coded compressed measurement when most scene points undergo motion.

In summary we make the following contributions:

- We provide a deep learning framework using which various coded exposure techniques can be compared.
- Our proposed approach matches or exceeds the state-of-the-art video reconstruction algorithms for each of the sensing techniques.
- We show that C2B has significant advantage over per-pixel exposure coding in reconstructing videos of scenes consisting of significant static regions.

2. Related Work

High speed imaging techniques with conventional sensor: Conventional image sensors capture a sharp video by using exposures shorter than the sampling period of the video. Frame interpolation techniques [6, 20, 11, 21] can be used to interpolate multiple frames between any two acquired frames and thereby increasing the video frame rate. When a long exposure is used, a blurred frame is acquired which encodes the full motion information. Recent learning based methods [26, 12] have been used to decode the motion information from a single blurred frame into multiple video frames.

Computational Imaging techniques: For scenes with little to no depth variations techniques using arrays of low-cost low frame rate cameras have shown to be effective at computationally recovering the high frame rate video [37, 31, 1]. A hybrid imaging system which uses one low-frame rate but high spatial resolution and another high frame rate but low spatial resolution sensors has been proposed for image deblurring [19] and high spatio-temporal resolution video recovery [23]. Recently, a hybrid imaging system consisting of image and event sensor has been proposed for high speed image reconstruction [32, 35, 34].

Motivated from the compressive sensing theory, several imaging architectures have been proposed for video compressive sensing problem [3]. Flutter shutter is a global exposure coding technique which was first introduced for motion deblurring [27] and then extended for video recovery from the compressed measurements [7]. A pixel-wise coded exposure system was proposed in [28] which demonstrated the recovery of high temporal resolution video from measurements compressed using spatial light modulator. A per-pixel control of the exposure was shown in [15], using only a commercially available CMOS image sensor without the need for any other hardware. The recently introduced multi-bucket sensors such as *Coded-2-Bucket* cam-

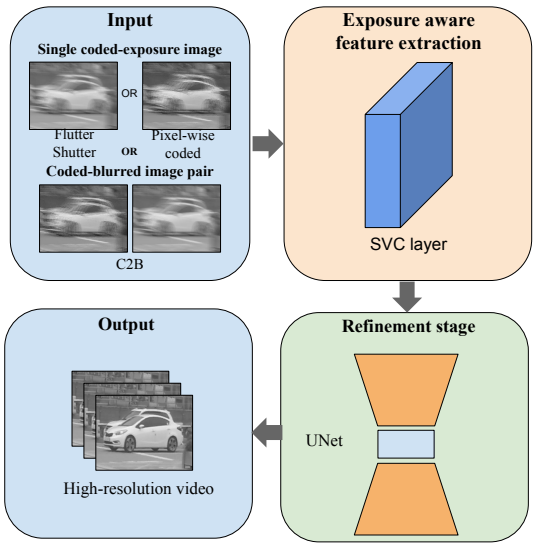

Figure 2: Our proposed algorithm takes in compressed measurements from the different coded exposure techniques as input and output full spatial and temporal resolution video in a single forward pass. Our proposed algorithm is fully convolutional and consists of a feature extraction stage and a refinement stage.

eras [30, 36], have reduced the complexity of per-pixel exposure control to a great extent. As video recovery from the compressed measurements is an ill-posed problem, strong signal priors are necessary for solving the inverse problem. While analytical priors such as wavelet domain sparsity [28, 24], TV-regularization [40] have been used, learning based algorithms such as Gaussian mixture models [38], dictionary learning [15] and neural network based models [9, 10, 39] have shown better performance than analytical priors. While many of the deep learning based methods use fully connected networks for the signal recovery, a very recent paper [14] uses a fully convolutional network to learn a denoising prior to iteratively solve the inverse problem.

3. A Unified Framework for Compressive Video Recovery Using Fully Convolutional Network

In this section, we elaborate on our proposed method to obtain the video signal from its compressed measurements. Our proposed algorithm takes in as input the compressed video measurements and outputs the video sequence at full spatial and temporal resolution in a single forward pass. The proposed architecture consists of two stages, as shown in Fig. 2. First, features are extracted from the compressed measurements using an exposure aware feature extraction stage consisting of shift variant convolutional layer. In the second stage, a deep neural network takes in the extracted features and outputs the full resolution video sequence. Our network architecture is flexible enough that it can be used for video reconstruction from all three coded exposure techniques considered here. All we need to do is train the network for these different inputs.

In Sec. 3.1, we provide motivation for using CNN for extracting relevant features from the compressed measurements. In Sec. 3.2, we elaborate on the use of *shift-variant* convolutional layer for handling pixel-wise coded exposure measurements and in Sec. 3.3 we specify the loss function used in the training our network.

3.1. Motivation for Using CNN

Several previous learning-based algorithms for compressive video recovery from coded exposure techniques have used fully connected networks [39, 10]. In [17], it has been shown that a fully convolutional network provides better reconstruction than fully connected networks for compressive video sensing. This section shows that a fully convolutional network is a better choice for solving our problem than a fully connected network.

For coded exposure techniques, each pixel in the compressed measurement is a linear combination of the underlying video sequence at that pixel alone. As there is no spatial multiplexing involved, it is possible to recover the video sequence at each pixel independently of the neighboring pixels. However, by using the information in a small neighborhood of a pixel, we can exploit the spatio-temporal redundancy inherent in natural video signals. Fully connected networks that are used in previous works provide global connectivity at the cost of much larger computational complexity and learning parameters. Thus, they should be used for solving inverse problems where global multiplexing occurs in the forward model, such as FlatCam [2]. With a toy example and elementary mathematical operations, we demonstrate next that fully connected networks with global connectivity are overkill and fully convolutional network with local spatial connectivity is a better design choice for our problem.

3.1.1 Toy example demonstration

Consider a video signal x of size $H \times W \times T$ with x_t representing each of the T frames of the video signal. A binary exposure sequence ϕ of dimension $H \times W \times T$ is used for temporally multiplexing the signal x into the measurement I. Mathematically, we can write the forward model as:

$$I = \sum_{t=1}^{T} \phi_t \odot x_t \ , \tag{1}$$

where ϕ_t represents the code corresponding to each frame of ϕ and $\odot$ represents element-wise multiplication.

The linear system in Eq. (1) can be represented in the matrix-vector form as follows:

$$I = \Phi X \ , \tag{2}$$

where Φ is a matrix representation of ϕ and X is a column vector obtained by vectorizing x. The minimum L_2-norm solution for the signal X can be obtained by:

$$\min_X \|X\|_2 \tag{3}$$

$$s.t.\ I = \Phi X \ . \tag{4}$$

Note that there are better reconstruction techniques such as dictionary learning which uses L_0 or L_1 norm on sparse transform coefficients of X [15]. But here our main goal is to show that CNN is appropriate for solving our inverse problem and hence we only provide a justification with L_2-norm, that has a closed-form solution. The approximate solution $\tilde{X}$ for Eq. (3) is given by,

$$\tilde{X} = \Phi^\dagger I \ , \tag{5}$$

$$\Phi^\dagger = \Phi^T(\Phi\Phi^T)^{-1}. \tag{6}$$

We notice that the matrix $\Phi\Phi^T$ is a diagonal matrix of dimension $HW \times HW$, and so is the matrix $(\Phi\Phi^T)^{-1}$. As shown in Fig. 3, the matrix $\Phi^\dagger$ is the matrix Φ^T whose columns are scaled by the entries of the diagonal matrix $(\Phi\Phi^T)^{-1}$. From the solution shown in Fig 3, it is clear that the temporal sequence at each pixel of the video is recovered only from the compressed measurement captured at that pixel. For example, if we consider the j^{th} pixel location, then the estimated temporal sequence $(\hat{x}_1^j, \hat{x}_2^j, \hat{x}_3^j)$ corresponding to the j^{th} pixel location depends only on the compressed measurement at the same pixel location I^j. This shows that CNN with local connectivity is more than sufficient for video reconstruction from coded exposure images.

3.2. Feature Extraction Using Shift-Variant Convolution (SVC)

In Sec. 3.1, we determined that to recover a video at a particular pixel, only that pixel's compressed measurements are necessary. Hence, the local connectivity offered by CNNs can be efficiently used for the task of recovering the underlying video signal. However, CNNs share the same weights across the whole input image. In pixel-wise coded exposure, the compressed measurement can be encoded using a different exposure sequence at each pixel. From Eq. (5) and Fig. 3, we see that the estimated video sequence at a particular pixel is dependent on the exposure sequence at that particular pixel. Hence, for pixels with different exposure sequence, using a different set of weights in the convolutional layer is desirable.

In flutter shutter video camera, each pixel in the image shares the same coded exposure sequence. Hence, the same learned convolutional weights w can be used to recover the underlying video signal for all the pixels. Thus, for recovering video sequences from the flutter shutter camera, we build our inversion stage as a standard convolutional layer as it achieves the functions mentioned above: local connectivity and shared weights across the whole image.

In pixel-wise coded exposure and C2B architectures, the underlying coded exposure sequence can change from one pixel to the next. In practice, a predetermined code of size $m \times n \times T$ is repeated over the entire image with a stride of $m \times n$ pixels. Hence, a standard convolutional layer cannot be directly used as it shares the same set of weights across the whole image. Instead, a convolutional layer, which can share weights for every $m \times n^{th}$ pixel, is desirable. Such a convolutional layer whose weights *vary* in a local neighborhood of $m \times n$ pixels was proposed in [22] called *shift-variant* convolutional (SVC) layer. This layer allows the network the freedom to learn different weights to invert the linear system when the underlying exposure sequence is different. Hence, we use this layer to extract adaptive features from the input compressed measurement. The exact implementation of our SVC layer is shown in Algorithm 1. These extracted features are input to the next stage of the network, which predicts the full resolution video sequence.

Algorithm 1 SVC implementation

Input: Single channel coded image I ▷ Size: $[1, H, W]$
Mask size: [m,n,T]
Output: Extracted feature maps ▷ Size: $[C, H, W]$

procedure SVC(I)

For each pixel, extract $k \times k$ pixel neighborhood and arrange them as the third dimension ▷ output tensor of size $[k^2, H, W]$

Reverse pixel shuffle with block size $m \times n$ ▷ output mn tensors of size $[k^2, H/m, W/n]$

for i = 1 to $m \times n$ **do**

2D convolutions for each i^{th} tensor of size $[k^2, H/m, W/n]$ with input channels k^2 and output C channels ▷ Output size [C,H/m,W/n]

end for

Pixel shuffle for the $m \times n$ channels to upsample the tensors spatially ▷ output tensor of size $[C, H, W]$

end procedure

3.3. Refinement Stage

The refinement stage takes as input the features extracted from the *shift-variant* convolutional layer and outputs a re-

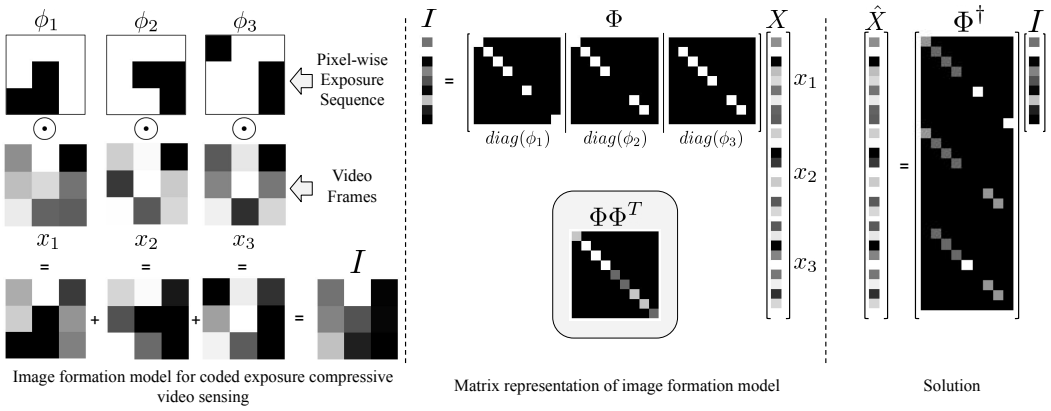

Figure 3: We show a toy example of pixel-wise coded exposure technique for compressing a video sequence of size $3 \times 3 \times 3$. Φ and X are the matrix and vector representation of the exposure sequence ϕ and the video sequence x, respectively. From the pseudo-inverse solution we see that the temporal video sequence reconstruction at any pixel depends only on the measurement and the code at that pixel alone. This motivates our choice of a fully convolutional design.

fined video sequence $\hat{X}$. Our refinement stage consists of a UNet [29] like deep neural network. Our proposed Unet model consists of 3 encoder stages followed by a bottleneck layer and 3 decoder stages. In each of the encoder stages, the feature maps are downsampled spatially by a factor of 2 and upsampled by the same factor in corresponding decoder stage. The output of this network is supervised using L_1 loss function. We also add a TV-smoothness loss on the final predicted video sequence. Our overall loss function then becomes,

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{ref} + \lambda_{tv}\mathcal{L}_{tv} \\ \mathcal{L}_{ref} &= \|\hat{X} - X\|_1 \\ \mathcal{L}_{tv} &= \|\nabla\hat{X}\|_1 \end{aligned} \tag{7}$$

where ∇ is the gradient operator in the x-y directions and λ_{tv} weights the smoothness term in the overall loss function.

4. Experimental Results

4.1. Experimental and Training Setup

Ground truth data preparation: We trained our proposed network using GoPro dataset [18] consisting of 22 video sequences at a frame rate of 240 fps and spatial resolution of 720×1280. The first 512 frames from each of the 22 sequences are spatially downsampled by 2 for preparing the training data. Overlapping video patches of size $64 \times 64 \times 16$ (height×width×frames) are extracted from the video sequences by using a sliding 3D window of $(32, 32, 8)$ pixels resulting in $263,340$ training patches. Similarly, for 8-frame reconstruction, we extracted video patches of size $64 \times 64 \times 8$ and shifting the window by $(32, 32, 4)$ pixels. The network was trained in PyTorch [25] using Adam optimizer [13] with a learning rate of 0.0001, λ_{tv} of 0.1 and batch size of 50 for 500 epochs[1].

Network architecture for each sensing technique: We trained our network separately for each of the different coded exposure techniques - *Flutter Shutter (FS), Pixel-wise coded exposure*, and *Coded-2-Bucket*. For FS, we trained our proposed network for 16-frame reconstruction and 8-frame reconstruction. As FS uses global code, a standard convolutional layer is used as a feature extraction layer in place of the SVC layer. We use the SVC layer as described in Sec. 3.2 as a feature extraction stage for pixel-wise coded exposure and C2B.

Input to the network: In the case of FS, the input to the network is a single coded exposure image obtained by multiplexing with a global exposure code. We used the exposure code obtained by maximizing the minimum of the DFT values' magnitude and minimizing the variance of the DFT values [27], over all possible binary codes. For the case of pixel-wise coded exposure, the coded mask of size $8 \times 8 \times 16$ is repeated spatially to make it the same dimension as input, which is then used for multiplexing. We used the *optimized SBE mask* exposure code proposed in [39] for this purpose. In the case of C2B exposure, the input to the network can either be a pair of coded and complement-coded images or a pair of coded and fully-exposed images. The output of the C2B sensor is two images that are coded using complementary exposure sequences (i.e., ϕ and $1 - \phi$). We used the same exposure pattern *optimized SBE mask* from [39] for C2B exposure as well. The fully-exposed or blurred image is obtained by adding the coded and complementary coded images. The image pair for the C2B sensor are stacked as two channels and provided as input to the proposed algorithm.

[1] https://github.com/asprasan/unified_framework

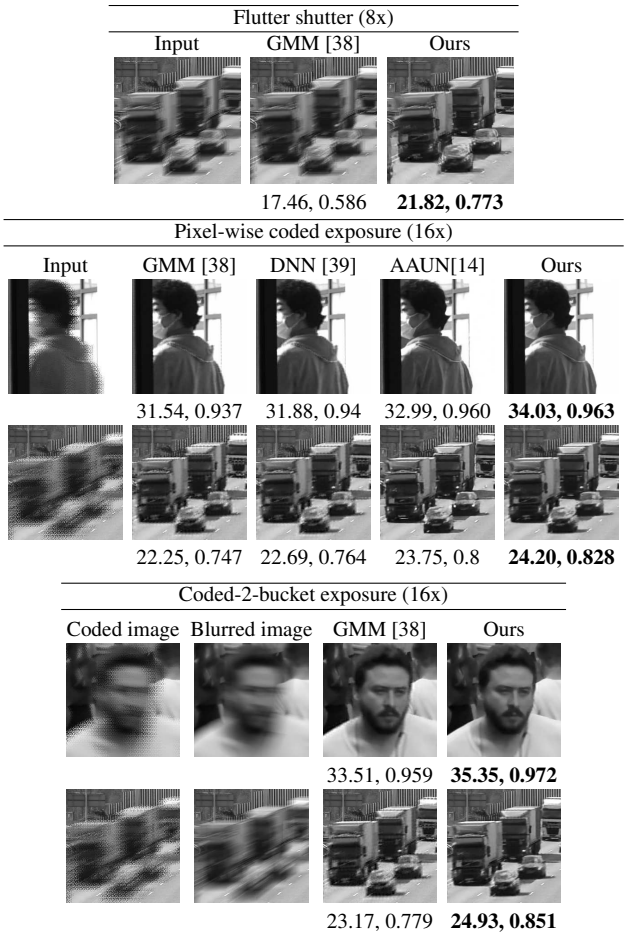

Figure 4: Visual comparison of middle frame from the reconstructed video sequences from various reconstruction algorithms. Our proposed method performs better than the existing methods GMM [38], DNN [39], and also doesn't suffer from block artifacts caused by patch-wise reconstruction. As expected, C2B produces better results than pixel-wise coded imaging. FS lags far behind.

4.2. Analysis of Video Reconstruction for Various Compressive Sensing Systems

In this section, we qualitatively and quantitatively assess video reconstruction from compressed measurements captured by different coded exposure techniques - *FS*, *pixel-wise coded exposure*, and *C2B*. We compared our proposed method with existing state-of-the-art algorithms for video reconstruction such as GMM-based inversion [38], DNN [39] and AAUN [14]. We used two sets of test videos with a different spatial resolution to perform this analysis. First, we used the test set that was used for evaluation in *DNN [39]*, consisting of 14 videos of spatial resolution 256×256 and 16 frames each. For the second set, we randomly selected 15 videos of resolution 720×1280 and 16 frames each, from the *GoPro test dataset [18]*.

Exposure	Algorithm	Test data	
		DNN set [39]	GoPro set [18]
FS 8x	GMM [38]	23.90, 0.818	23.30, 0.766
	Ours	**24.06, 0.833**	**25.03, 0.811**
FS 16x	GMM [38]	21.50, 0.738	21.45, 0.697
	Ours	**21.69, 0.752**	**21.61, 0.710**
		DNN set [39]	GoPro set [18]
Pixel-wise coded 16x	GMM [38]	29.31, 0.898	29.94, 0.887
	DNN [39]	30.21, 0.905	30.27, 0.890
	AAUN [14]	28.5, 0.882	31.6, 0.910
	Ours	**31.14, 0.925**	**31.76, 0.914**
		DNN set [39]	GoPro set [18]
C2B 16x	GMM [38]	30.94, 0.914	30.84, 0.898
	Ours	**32.23, 0.935**	**32.34, 0.920**

Table 1: Quantitative results for different coded exposure techniques and reconstruction algorithms. The table lists average PSNR(dB) and SSIM of reconstructed videos from *DNN set* [39] and *GoPro set* [18].

Exposure	CPU run-time (GPU run-time) in seconds			
	GMM [38]	DNN [39]	AAUN [14]	Ours
Pixel-wise	78.7	4.6 (2.7)	11.1 (0.3)	3.6 (0.011)
C2B	96.4	–	–	4.1 (0.013)

Table 2: Run time for various algorithms to reconstruct a single $256 \times 256 \times 16$ frame sequence. For algorithms that are accelerated by GPU, the run times are provided in parentheses. The run times are for an Intel i7 CPU and Nvidia GeForce 2080 Ti GPU.

For FS, we compared our proposed method with the GMM-based video reconstruction method [38] for 8-frame and 16-frame reconstructions. For single pixel-wise coded exposure sensing, we compare with GMM-based inversion [38] and state-of-the-art deep learning based methods, DNN [39] and AAUN [14], for 16-frame reconstruction. For C2B exposure, we compare with GMM-based inversion [38] for 16-frame reconstruction from a pair of coded and blurred images. We trained the GMM [38] model with 20 components using the same training dataset as described in Sec. 4.1. We used $8\times8\times8$ patches to train the GMM [38] for 8-frame reconstruction and $8 \times 8 \times 16$ patches for 16-frame reconstruction. We used the pre-trained model for DNN proposed in [39]. We trained the AAUN [14] algorithm on the same training dataset as described in Sec. 4.1. The model was trained for 80 epochs on patches of size 128×128 for 16-frame reconstruction.

Comparison analysis: Qualitative reconstruction results are shown in Fig. 4 and quantitative results are summarized in Table 1. FS produces satisfactory results for 8-frame reconstruction but struggles to reconstruct 16 frames. Pixel-wise coded exposure can perform 16-frame reconstruction with good fidelity. For natural images, the intensities in a small spatial neighborhood are correlated. Intuitively, using different exposure sequences for different pix-

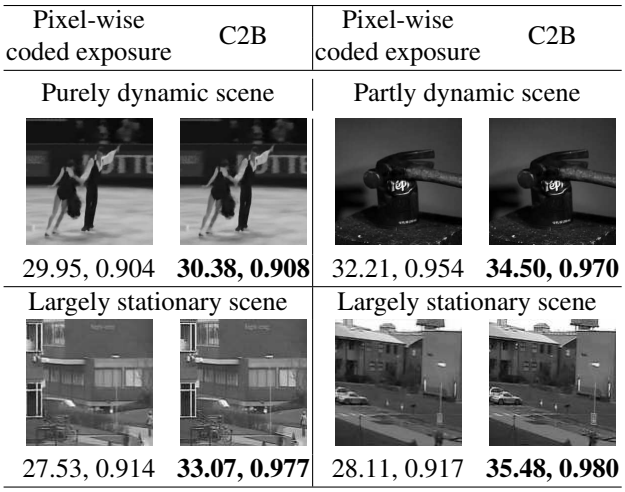

Figure 5: Qualitative comparison of cropped middle frames from the reconstruced video sequences. When majority of the pixels do not see any motion C2B has a significant advantage, while being only marginally beneficial in the case where majority of the pixels see motion.

els, is equivalent to making multiple measurements, which helps in recovering the information better. As our algorithm exploits the spatial correlation structure, the pixel-wise coded exposure technique will have an advantage over the global, flutter shutter imaging technique in the fidelity of the reconstructed video. The C2B exposure provides an additional advantage by capturing information that is lost by the pixel-wise coded exposure and hence produces better reconstruction than pixel-wise coded exposure. Overall, we observe a similar trend in the reconstruction performance of different sensing techniques in both GMM [38] our proposed and model. We see that, overall, C2B provides the best reconstruction and FS performs the worst, while there is only a slight quantitative advantage for C2B when compared to pixel-wise exposure. We further compare the performance of pixel-wise coded exposure with C2B exposure in the following section.

Our proposed fully-convolutional model performs better than the existing methods, GMM [38], DNN [39] and AAUN [14], for all the sensing techniques. Since we reconstruct the full video, our proposed method doesn't suffer from block artifacts, which is seen in patch-wise reconstruction methods such as GMM and DNN. A comparison of run times of various algorithms on CPU as well as GPU has also been provided in Table 2. Patch-based reconstruction methods such as GMM and DNN require a significantly longer time to reconstruct a single video sequence compared to AAUN [14] and our algorithm. Being an iterative deep learning algorithm, AAUN [14] takes 3x and 10x longer time than our proposed algorithm on CPU and GPU, respectively.

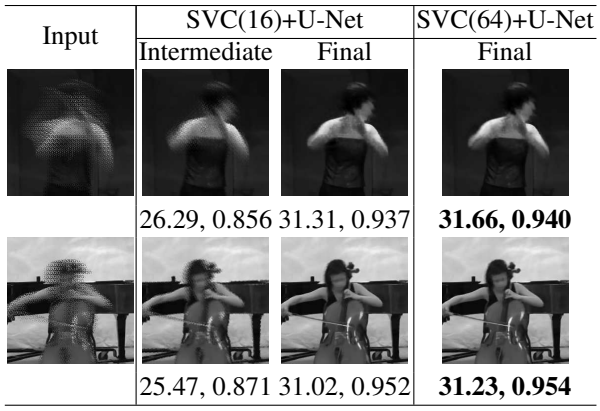

Figure 6: The figure compares the middle frames from the reconstructed video sequences from two different architectural choices. It can be seen that SVC(64)+U-Net performs better than SVC(16)+U-Net in terms of the PSNR and SSIM.

4.3. When Does C2B Have a Significant Advantage over Pixel-wise Coded Exposure?

In Sec. 4.2, we observe that C2B based sensing provides only a slight advantage compared to pixel-wise coded exposure technique. To analyze and identify the cases where C2B provides a significant advantage over pixel-wise coded exposure, we conduct experiments on different kinds of videos: purely dynamic sequences, partly-dynamic-partly-static sequences, and largely static sequences. We use our proposed method to compare video reconstruction from a pixel-wise coded exposure image and from a coded-blurred image pair obtained from C2B. We explain why we use a blurred image with the coded image as input through an ablation study in Sec 4.4. Fig. 5 shows reconstructed results for the different cases of video sequences mentioned above. For purely dynamic scenes, C2B does not show a notable performance improvement over pixel-wise coded exposure. However, for videos containing significant static regions, C2B produces much better reconstruction results than pixel-wise coded exposure. If we consider a scene composed of both stationary and dynamic regions, the dynamic regions are better captured by the coded exposure image, while the stationary regions are better captured by the fully-exposed image. Therefore, it follows that videos containing stationary regions can be better recovered by using the additional information captured by C2B.

4.4. Ablation Study

Ablation study on proposed architecture: We explain some of the architectural choices that we made in developing our proposed network. We experimented with two different architectures for pixel-wise coded exposure - U-Net only, SVC(16) + U-Net, and SVC(64) + U-Net. SVC denotes the shift-variant convolution layer [22], and the

following value in bracket specifies the number of output channels of the SVC layer. In U-Net only framework, we input the coded image directly to the standard U-Net architecture, which learns the mapping to the full resolution video sequence. In SVC(16)+U-Net, we implemented the SVC layer to produce an intermediate reconstruction from the input, followed by U-Net [29] to refine the intermediate reconstruction and produce the final high-quality video. While training the network, we supervise both the intermediate and final reconstructions using ground truth with a 0.5 weightage for intermediate reconstruction. In SVC(64)+U-Net, we modified the number of output channels of the SVC layer from 16 to 64. Therefore, instead of producing an intermediate reconstruction, the SVC layer extracts the features required to reconstruct the video. Here, we supervise the final reconstruction using ground truth while training. From Table 3, we observe that using SVC(64)+U-Net gives the best reconstruction results. It can also be observed that using an SVC layer instead of a standard convolutional layer provides a significant improvement in performance. The SVC layer also does not add significantly to the computational overhead. While, SVC(64)+U-Net model takes 0.011s, Unet-only model takes 0.009s per forward pass on a GPU for a $256 \times 256 \times 16$ video sequence. Therefore, we choose SVC(64)+U-Net architecture as our proposed method.

Ablation Study on C2B Input: The advantage of using C2B exposure is that it captures the complementary information otherwise lost in pixel-wise coded exposure. C2B captures two coded exposure images: coded image and complement-coded image. We can obtain a fully-exposed or blurred image by adding the coded and complementary coded images. There are two ways of representing the C2B input: a coded-complement image pair or coded-blurred image pair. We evaluated both the cases and determined that video reconstruction from a coded-blurred image pair performs marginally better than reconstruction from a coded-complement pair. The results are summarized in Table 3.

4.5. Learning the mask

Jointly learning the coded mask ϕ and the reconstruction algorithm has been shown to provide better reconstruction results [14, 22, 10]. To demonstrate this, we jointly learn the coded mask ϕ along with our proposed learning-based reconstruction algorithm. We add the weights of the mask ϕ also as trainable parameters along with the other trainable network parameters. As the hardware sensors can use only binary mask patterns, we restrict the mask weights to be binary. Binarization is done via thresholding the weights before each forward pass through the network. As thresholding is non-differentiable, we follow [8] and use the *straight-through estimator* for computing gradients. We use a similar training scheme and training dataset as described in Sec. 4.1. The mask ϕ and the network are jointly trained for 16x reconstruction for the case of FS, pixel-wise exposure, and C2B. The trained network is evaluated on the GoPro test set, and the results are summarized in Table 4. We observe that for both the noiseless and the noisy cases, joint optimization of the coded mask and the reconstruction algorithm provides better performance. The gap between the fixed and optimized code is bigger for the noisy case.

Exposure		DNN set [39]		GoPro set [18]	
		PSNR	SSIM	PSNR	SSIM
Pixel-wise coded	U-Net only	30.68	0.919	31.27	0.902
	SVC(16)+U-Net	30.89	0.921	31.56	0.910
	SVC(64)+U-Net	**31.14**	**0.925**	**31.76**	**0.914**
C2B	coded+complement	32.19	0.935	32.31	0.919
	coded+blurred	**32.23**	**0.935**	**32.34**	**0.920**

Table 3: Ablation studies on proposed architecture and C2B input. The table lists average PSNR(dB) and SSIM of reconstructed videos from *DNN set [39]* and *GoPro set [18]*.

	Noiseless		Noisy($\sigma = 0.01$)	
Model	PSNR	SSIM	PSNR	SSIM
FS (fixed)	21.61	0.752	21.28	0.707
FS (optimized)	**21.72**	**0.756**	**21.42**	**0.722**
Pixel-wise(fixed)	31.76	0.914	27.58	0.845
Pixel-wise(optimized)	**32.13**	**0.953**	**29.58**	**0.912**
C2B(fixed)	32.34	0.920	28.22	0.860
C2B(optimized)	**32.59**	**0.961**	**30.06**	**0.912**

Table 4: PSNR, SSIM comparison of reconstructed videos for FS, pixel-wise and C2B for *fixed* and *optimized* coded mask ϕ. We observe better reconstruction performance for *optimized* mask for both the noisy and noiseless cases.

5. Conclusion

We propose a unified deep learning-based framework to make a fair comparison of the video reconstruction performance of various coded exposure techniques. We make a mathematically informed choice for our framework that leads to the use of fully convolutional architecture over a fully connected one. Extensive experiments show that the proposed algorithm performs better than previous video reconstruction algorithms across all coded exposure techniques. The proposed unified learning framework is used to make an extensive quantitative and qualitative evaluation of the different coded exposure techniques. From this, we observe that C2B provides the best reconstruction performance, closely followed by the single pixel-wise coded exposure technique, while FS lags far behind. Our further analysis of C2B shows that a significant advantage is gained over pixel-wise coded exposure only when the scenes are largely static. However, when the majority of scene points undergo motion, C2B shows only a marginal benefit over acquiring a single pixel-wise coded exposure measurement.

References

[1] Amit Agrawal, Mohit Gupta, Ashok Veeraraghavan, and Srinivasa G Narasimhan. Optimal coded sampling for temporal super-resolution. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 599–606. IEEE, 2010.

[2] M Salman Asif, Ali Ayremlou, Aswin Sankaranarayanan, Ashok Veeraraghavan, and Richard G Baraniuk. Flatcam: Thin, lensless cameras using coded aperture and computation. *IEEE Transactions on Computational Imaging*, 3(3):384–397, 2016.

[3] Richard G Baraniuk, Thomas Goldstein, Aswin C Sankaranarayanan, Christoph Studer, Ashok Veeraraghavan, and Michael B Wakin. Compressive video sensing: algorithms, architectures, and applications. *IEEE Signal Processing Magazine*, 34(1):52–66, 2017.

[4] Jinwei Gu, Yasunobu Hitomi, Tomoo Mitsunaga, and Shree Nayar. Coded rolling shutter photography: Flexible space-time sampling. In *2010 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2010.

[5] Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa G Narasimhan. Flexible voxels for motion-aware videography. In *European Conference on Computer Vision*, pages 100–114. Springer, 2010.

[6] Evan Herbst, Steve Seitz, and Simon Baker. Occlusion reasoning for temporal interpolation using optical flow. *Department of Computer Science and Engineering, University of Washington, Tech. Rep. UW-CSE-09-08-01*, 2009.

[7] Jason Holloway, Aswin C Sankaranarayanan, Ashok Veeraraghavan, and Salil Tambe. Flutter shutter video camera for compressive sensing of videos. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9. IEEE, 2012.

[8] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in neural information processing systems*, pages 4107–4115, 2016.

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

[10] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deepbinarymask: Learning a binary mask for video compressive sensing. *Digital Signal Processing*, 96:102591, 2020.

[11] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018.

[12] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6334–6342, 2018.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[14] Y. Li, M. Qi, R. Gulve, M. Wei, R. Genov, K. N. Kutulakos, and W. Heidrich. End-to-end video compressive sensing using anderson-accelerated unrolled networks. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020.

[15] Dengyu Liu, Jinwei Gu, Yasunobu Hitomi, Mohit Gupta, Tomoo Mitsunaga, and Shree K Nayar. Efficient space-time sampling with pixel-wise coded exposure for high-speed imaging. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):248–260, 2013.

[16] Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle, Lawrence Carin, Guillermo Sapiro, and David J Brady. Coded aperture compressive temporal imaging. *Optics express*, 21(9):10526–10545, 2013.

[17] J. N. P. Martel, L. K. Müller, S. J. Carey, P. Dudek, and G. Wetzstein. Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1642–1653, 2020.

[18] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] Shree K Nayar and Moshe Ben-Ezra. Motion-based motion deblurring. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):689–698, 2004.

[20] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017.

[21] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017.

[22] T. Okawara, M. Yoshida, H. Nagahara, and Y. Yagi. Action recognition from a single coded image. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, 2020.

[23] Avinash Paliwal and Nima Khademi Kalantari. Deep slow motion video reconstruction with hybrid imaging system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[24] Jae Young Park and Michael B Wakin. A multiscale framework for compressive sensing of video. In *2009 Picture Coding Symposium*, pages 1–4. IEEE, 2009.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[26] Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2019.

[27] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *ACM transactions on graphics (TOG)*, volume 25, pages 795–804. ACM, 2006.

[28] Dikpal Reddy, Ashok Veeraraghavan, and Rama Chellappa. P2c2: Programmable pixel compressive camera for high speed imaging. In *CVPR 2011*, pages 329–336. IEEE, 2011.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[30] Navid Sarhangnejad, Nikola Katic, Zhengfan Xia, Mian Wei, Nikita Gusev, Gairik Dutta, Rahul Gulve, Harel Haim, Manuel Moreno Garcia, David 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)*, pages 102–104. IEEE, 2019.

[31] Eli Shechtman, Yaron Caspi, and Michal Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, 2005.

[32] Prasan Shedligeri and Kaushik Mitra. Photorealistic image reconstruction from hybrid intensity and event-based sensor. *Journal of Electronic Imaging*, 28(6):063012, 2019.

[33] Ashok Veeraraghavan, Dikpal Reddy, and Ramesh Raskar. Coded strobing photography: Compressive sensing of high speed periodic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):671–686, 2010.

[34] Zihao W Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi. Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1609–1619, 2020.

[35] Zihao W Wang, Weixin Jiang, Kuan He, Boxin Shi, Aggelos Katsaggelos, and Oliver Cossairt. Event-driven video frame synthesis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[36] Mian Wei, Navid Sarhangnejad, Zhengfan Xia, Nikita Gusev, Nikola Katic, Roman Genov, and Kiriakos N Kutulakos. Coded two-bucket cameras for computer vision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 54–71, 2018.

[37] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. In *ACM SIGGRAPH 2005 Papers*, pages 765–776. ACM, 2005.

[38] Jianbo Yang, Xin Yuan, Xuejun Liao, Patrick Llull, David J Brady, Guillermo Sapiro, and Lawrence Carin. Video compressive sensing using gaussian mixture models. *IEEE Transactions on Image Processing*, 23(11):4863–4878, 2014.

[39] Michitaka Yoshida, Akihiko Torii, Masatoshi Okutomi, Kenta Endo, Yukinobu Sugiyama, Rin-ichiro Taniguchi, and Hajime Nagahara. Joint optimization for compressive video sensing and reconstruction under hardware constraints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 634–649, 2018.

[40] Xin Yuan. Generalized alternating projection based total variation minimization for compressive sensing. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2539–2543. IEEE, 2016.