Joint optimization for compressive video sensing and reconstruction under hardware constraints

Michitaka Yoshida[1][0000−0002−2227−6345], Akihiko Torii[2], Masatoshi Okutomi[2], Kenta Endo[3], Yukinobu Sugiyama[3], Rin-ichiro Taniguchi[1], and Hajime Nagahara[4][0000−0003−1579−8767]

[1] Kyushu University
[2] Tokyo Institute of Technology
[3] Hamamatsu Photonics K. K.
[4] Osaka University

Abstract. Compressive video sensing is the process of encoding multiple sub-frames into a single frame with controlled sensor exposures and reconstructing the sub-frames from the single compressed frame. It is known that spatially and temporally random exposures provide the most balanced compression in terms of signal recovery. However, sensors that achieve a fully random exposure on each pixel cannot be easily realized in practice because the circuit of the sensor becomes complicated and incompatible with the sensitivity and resolution. Therefore, it is necessary to design an exposure pattern by considering the constraints enforced by hardware. In this paper, we propose a method of jointly optimizing the exposure patterns of compressive sensing and the reconstruction framework under hardware constraints. By conducting a simulation and actual experiments, we demonstrated that the proposed framework can reconstruct multiple sub-frame images with higher quality.

Keywords: Compressive sensing, video reconstruction, deep neural network

1 Introduction

Recording a high-frame video with high spatial resolution has various uses in practical and scientific applications because it essentially provides more information to analyze the recorded events. Such video sensing can be achieved by using a high-speed camera [1] that shortens the readout time from the pixel by employing a buffer for each pixel and reducing the analog-to-digital (AD) conversion time by using parallel AD converters. Since he mass production of these special sensors is not unrealistic, several problems remain unresolved with regard to the replacement of standard complementary metal-oxide-semiconductor (CMOS) sensors. As an example of hardware related problems, a fast readout sensor is larger than a standard sensor because it is assembled with additional circuits and transistors. To make a high-frame sensor more compact, a smaller phototransistor must be used to lower the sensitivity.

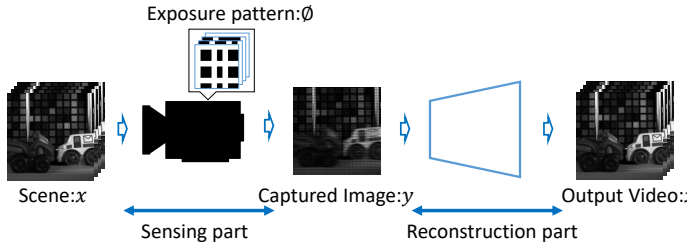


Fig. 1. Compressive video sensing. A process of encoding multiple sub-frames into a single frame with controlled sensor exposures, and reconstructing the sub-frames from a single compressed frame.

A feasible approach consists of capturing video by using compressive sensing techniques [2–6], *i.e.*, by compressing several sub-frames into a single frame at the time of acquisition, while controlling the exposure of each pixel ' s position. In contrast to the standard images captured with a global shutter, where all pixels are exposed concurrently, a compressive video sensor samples temporal information and compresses it into a single image, while randomly changing the exposure pattern for each pixel. This non-continuous exposition enables the recovery of high-quality video. Formally, compressive video sensing is expressed as follows:

$$\mathtt{y} = \phi \mathtt{x} \tag{1}$$

where $\mathtt{x}$ is the high-frame video to be compressed, ϕ is the measurement matrix (exposure patterns), and $\mathtt{y}$ is the compressed single image. The following tasks are included in compressive video sensing: reconstruct a high-frame video $\bar{\mathtt{x}}$ from a single image $\mathtt{y}$ by using pattern ϕ; optimize the pattern that enables high-quality video reconstruction (Figure 1).

Under the assumption that random (theoretically optimal) patterns can be implemented without hardware sensor constraints, numerous studies have investigated a method of reconstructing (decoding) from a single image based on sparse coding [3–5]. In signal recovery theory, the best exposure pattern is random sampling from a uniform distribution. However, this is not an optimal pattern in terms of practical image sensing, because a practical scene does not always maintain the sparsity assumed in compressive sensing theory. Few existing studies [6] have investigated scene adaptive exposure patterns in the context of a target scene.

However, implementing such completely random exposures with a practical CMOS sensor is not realistic, owing to hardware limitations. Achieving compatibility between these special sensors and the sensitivity and resolution is difficult because these sensors typically have more complicated circuits in each pixel, and this decreases the size of the photo-diode [7]. Additionally, standard commercial CMOS sensors, *e.g.*, three-transistor CMOS sensors, do not have a per-pixel frame buffer on the chip. Thus, such sensors are incapable of multiple exposure in

a non-destructive manner [3]. There exists an advanced prototype CMOS sensor [2] that can control the exposure time more flexibly. However, its spatial control is limited to per line (column and row) operations. Therefore, it is necessary to optimize the exposure patterns by recognizing the hardware constraints of actual sensors.

Contribution. In this paper, we propose a new pipeline to optimize both the exposure pattern and reconstruction decoder of compressive video sensing by using a deep neural network (DNN) framework [8]. To the best of our knowledge, ours is the first study that considers the actual hardware sensor constraints and jointly optimizes both the exposure patterns and the decoder in an end-to-end manner. The proposed method is a general framework for optimizing the exposure patterns with and without hardware constraints. We demonstrated that the learned exposure pattern can recover high-frame videos with better quality in comparison with existing handcrafted and random patterns. Moreover, we demonstrated the effectiveness of our method with images captured by an actual sensor.

2 Related studies

Compressive video sensing consists of sensing and reconstruction: sensing pertains to the hardware design of the image sensor for compressing video (subframes) to a single image. Reconstruction pertains to the software design for estimating the original subframes from a single compressed image.

Sensing. Ideal compressive video sensing requires a captured image with random exposure, as expressed in Equation 1 and shown in Figure 1. However, conventional charge-coupled device (CCD) and CMOS sensors either have a global or a rolling shutter. A global shutter exposes all of the pixels concurrently, while a rolling shutter exposes every pixel row/column sequentially. A commercial sensor capable of capturing an image with random exposure does not exist. Therefore, most existing studies have only evaluated simulated data [5] or optically emulated implementations [3].

Many studies have investigated the development of sensors for compressive sensing [9]. Robucci et al. [10] proposed the design of a sensor that controls the exposure time by feeding the same signal to pixels located in the same row, *i.e.*, row-wise exposure pattern coding is performed at the sensor level. In an actual sensor implementation, analog computational coding is used before the analog-to-digital (A/D) converter receives the signals. The proposed sensor type is a passive pixel sensor that is not robust to noise, in comparison with an active pixel sensor that is typically used in commercial CMOS image sensors. Dadkhah et al. [11] proposed a sensor with additional exposure control lines connected to the pixel block arrays, each of which was composed of several pixels. The pixel block array shared the same exposure control line. However, each pixel inside the block could be exposed individually. Although the block-wise pattern was repeated, from a global point of view, this sensor could generate a random

exposure pattern locally. Because the number of additional exposure control lines was proportional to the number of all pixels in the sensor, the fill factors remained similar to those of a standard CMOS sensor.

Majidzadeh et al. [12] proposed a CMOS sensor with pixel elements equipped with random pattern generators. Because the generator was constructed from a finite state machine sequence, the fill factor of this sensor was extremely low, and this resulted in lower sensor sensitivity. Oike et al. [13] proposed a sensor wherein all pixels were exposed concurrently, as in a regular CMOS image sensor. The exposure information was read out as a sequential signal, which was cloned and blanched to several multiplexers, in a parallel manner. The sequential signal was encoded by using different random patterns.

Through parallel A/D converters, several random measurements, incoherent to each other, can be obtained with a single shot. Relevant studies [10–13] have mainly focused on super-resolution. Because the measured spatial resolution can be reduced, the frame rate can be increased within a certain bandwidth. High frame rate acquisition has not been demonstrated in any actual experiments conducted by these studies.

There have been fewer attempts to implement sensors for compressive video sensing. Spinoulas et al. [14] have demonstrated on-chip compressive sensing. They used an inexpensive commercial development toolkit with flexible readout settings to perform non-uniform sampling from several captured frames in combination with pixel binning, region of interest (ROI) position shifting, and ROI position flipping. Note that this coding was not performed on the sensor chip, but rather during the readout process.

Sonoda et al. [2] used a prototype CMOS sensor with exposure control capabilities. The basic structure of this sensor was similar to that of a standard CMOS sensor, although separate reset and transfer signals controlled the start and end time of the exposure. Because the pixels in a column and row shared the reset and transfer signal, respectively, the exposure pattern had row and column wise dependency. These researchers also proposed to increase the randomness of the exposure pattern. However, the method could not completely solve the pattern ' s row and column wise dependency.

Reconstruction. There are various methods to reconstruct a video from a single image captured with compressive sensing. Because the video output rank ($\mathtt{x}$ in Equation 1) is higher than the input ($\mathtt{y}$), it is impossible to reconstruct the video deterministically.

One of the major approaches consists of adopting sparse optimization, and assuming that the video $\mathtt{x}_\mathtt{p}$ can be expressed by a linear combination of sparse bases $\mathtt{D}$, as follows: $\mathtt{x}_\mathtt{p} = \mathtt{D}\alpha = \alpha_1 D_1 + \alpha_2 D_2 + \cdots + \alpha_k D_k$

where $\alpha = [\alpha_1, .., \alpha_k]^\mathrm{T}$ are the coefficients, and the number of coefficients k is smaller than the dimension of the captured image. In the standard approach, the $\mathtt{D}$ bases are pre-computed, *e.g.*, by performing K-SVD [15] on the training data. From Equation 1, we obtain the following expression:

$$\mathtt{y}_\mathtt{p} = \phi_\mathtt{p}\mathtt{D}\alpha. \tag{2}$$

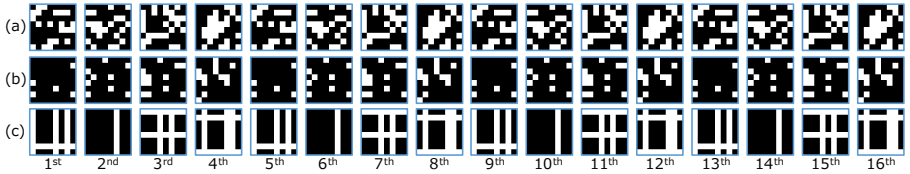


Fig. 2. Examples of exposure patterns under hardware constraints: (a) random exposure sensor, (b) single bump exposure (SBE) sensor [3], (c) row-column wise exposure (RCE) sensor [2]

Because $\mathtt{y_p}$, $\phi_\mathtt{p}$, and $\mathtt{D}$ are known, it is possible to reconstruct videos by solving α, *e.g.*, by using the orthogonal matching pursuit (OMP) algorithm [16, 3] that optimizes the following equation:

$$\alpha = \arg\min_{\alpha} ||\alpha||_0 \text{ subject to } ||\phi\mathtt{D}\alpha - \mathtt{y_p}||_2 \leq \sigma \tag{3}$$

To solve the sparse reconstruction, L_1 relaxation has been used because L_0 optimization is hard to compute and also computationally expensive. LASSO [17] is a solver for the L_1 minimization problem, as expressed in Equation 4, and has also been used in the sparse reconstruction of the video.

$$\min_{\alpha} ||\phi\mathtt{D}\alpha - \mathtt{y_p}||_2 \text{ subject to } ||\phi||_1 \leq \sigma \tag{4}$$

Yang et al. [4] proposed a reconstruction method based on Gaussian Mixture Models (GMM). They assumed that the video patch $\{\mathtt{x}_p\}$ could be represented as follows:

$$\mathtt{x}_p \sim \sum_{k=1}^{K} \lambda_k \mathcal{N}(\mathtt{x}_p \mid \mu_k, \Sigma_k) \tag{5}$$

where $\mathcal{N}$ is the Gaussian distribution and K, Σ_k, and λ_k are the number of GMM components, mean, covariance matrix, and weight of the k_{th} Gaussian component ($\lambda_k > 0$ and $\sum_{k=1}^{K} \lambda_k = 1$). Therefore, the video could be reconstructed by computing the conditional expectation value of $\mathtt{x}_p$.

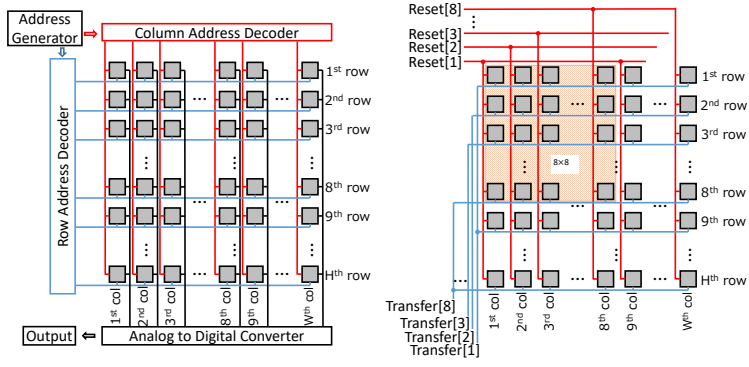
Very recently, Iliadis et al. [5] proposed a decoder based on a DNN. The network was composed by fully connected layers and learned the non-linear mapping between a video sequence and a captured image. The input layer had the size of the captured image, while the hidden and output layers had the size of the video. Because this DNN-based decoder only calculated the convolution with learned weights, the video reconstruction was fast.

3 Hardware constraints of exposure controls

As already discussed in Section 2, there exist hardware constraints that prevent the generation of completely random exposure patterns, which are a theoretical

requirement of compressive video sensing, as shown in Figure 2a. In this paper, we describe two examples of hardware constraints, which have been suggested by [3] and fabricated to control pixel-wise exposures [2] on realistic sensors. In this section, we detail the hardware constraints resulting from sensor architecture.

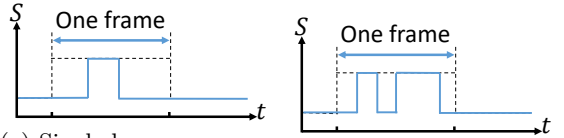
Hitomi et al. [3] suggested that CMOS modification is feasible. However, they did not produce a modified sensor to realize pixel-wise exposure control as shown in Figure 3a. Existing CMOS sensors have row addressing, which provides row-wise exposure such as that of a rolling shutter. These researchers proposed to add a column addressing decoder to provide pixel-wise exposure. However, a typical CMOS sensor does not have a per-pixel buffer, but does have the characteristic of non-destructive readout, which is only a single exposure in a frame, as shown in Figure 4a. The exposure should have the same duration in all pixels because the dynamic range of a pixel is limited. Therefore, we can only control the start time of a single exposure for each pixel, and cannot split the exposure duration to multiple exposures in one frame, even though the exposure time would be controllable. Here, the main hardware restriction is the single bump exposure (SBE) on this sensor, which is termed as the SBE sensor in this paper. Figure 2b shows an example of the SBE sensor ' s space-time exposure pattern.



(a) Architecture of single bump exposure (SBE) CMOS image sensor [3]

(b) Architecture of row-column wise exposure (RCE) CMOS image sensor [2]

Fig. 3. Architecture of single bump exposure (SBE) and row-column wise exposure (RCE) image sensors. The SEB image sensor in (a) has a row and column address decoder and can be read out pixel-wise. However, it does not have a per-pixel buffer and can perform single-bump exposure (Figure 4). The RCE image sensor shown in (b) has an additional transistor and exposure control signal line, and can perform multi-bump exposure. However, it only has row addressing, which provides row wise exposure, such as that of a rolling shutter.



(a) Single-bump exposure (SBE) (b) Multi-bump exposure

Fig. 4. Exposure bump. Single-bump means that the sensor is exposed only once during the exposure. Conversely, multi-bump means that the sensor is exposed multiple times during the exposure.

Sonoda et al. [2] used the prototype CMOS sensor with additional reset and transfer signal lines to control the exposure time. The sensor ' s architecture is shown in Figure 3b. This figure shows the top left of the sensor with a block structure of 8×8 pixels. These signal lines are shared by the pixels in the columns and rows. The reset signal lines are shared every eighth column, and the transfer signal lines are shared every eighth row. Therefore, the exposure pattern is cloned block wise. The sensor had a destructive readout and the exposure was more uniquely controllable such that we could use multiple exposures and their different durations in a frame. However, the exposure patterns depended spatially on the rows or columns of the neighboring pixels. In this paper, we termed this sensor as the row-column wise exposure (RCE) sensor. Figure 2-c shows an example pattern of the RCE sensor.

Few previous methods [6] of designing and optimizing exposure patterns for compressive video sensing have been reported. However, none of them can be applied to realistic CMOS architectures, because all of these previously reported methods have assumed that exposure is fully controllable. Hence, we propose a new method to optimize patterns under hardware constraints, although we also considered unconstrained sensors in this study.

4 Joint optimization for sensing and reconstruction under hardware constraints

In this section, we describe the proposed optimization method of jointly optimizing the exposure pattern of compressive video sensing, and performing reconstruction by using a DNN. The proposed DNN consists of two main parts. The first part is the sensing layer (encoding) that optimizes the exposure pattern (binary weight) under the constraint imposed by the hardware structure, as described in Section 3. The second part is the reconstruction layer that recovers the multiple sub-frames from a single captured image, which was compressed by using the optimized exposure pattern. The overall framework is shown in Figure 5.

Training was carried out in the following steps:

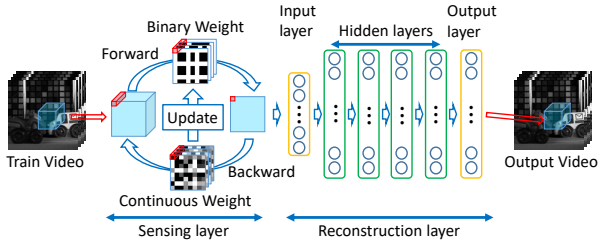


Fig. 5. Network Structure. Proposed network structure to jointly optimize the exposure pattern of compressive video sensing, and the reconstruction. The left side represents the sensing layer that compresses video to an image by using the exposure pattern. The right side represents the reconstruction layer that learns non-linear mapping between the compressed image to video reconstruction.

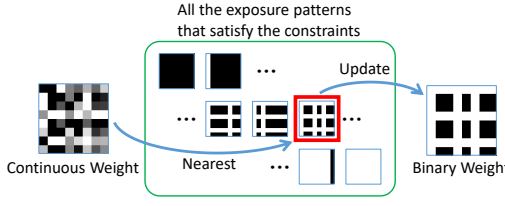


Fig. 6. Binary weight update. Binary weight updated with the most similar patterns in the precomputed binary weights. The similarity between the continuous-value weight and the precomputed binary pattern is computed by the normalized dot product.

1. At the time of forward propagation, the binary weight is used for the sensing layer, while the reconstruction layer uses the continuous weights.
2. The gradients are computed by backward propagation.
3. The continuous weights of sensing and reconstruction layers are updated according to the computed gradients.
4. The binary weights of the sensing layer are updated with the continuous weights of the sensing layer.

4.1 Compressive sensing layer

We sought an exposure pattern that would be capable of reconstructing video frames with high quality when trained along with the reconstruction (decoding) layer. More importantly, the compressive sensing layer had to be capable of handling the exposure pattern constraints imposed by actual hardware architectures. Because implementing nested spatial pattern constraints (Section 3) in the DNN layer was not trivial, we used a binary pattern (weight) chosen from the precomputed binary weights at forward propagation in the training. The binary weight was relaxed to a continuous value [18] to make the network

differentiable by backward computation. Next, the weight was binarized for the next forward computation by choosing the most similar patterns in the precomputed binary weights. The similarity between the continuous-value weight and the precomputed binary pattern was computed by the normalized dot product (Figure 6).

The binary patterns can be readily derived from the hardware constraints. For the SBE sensor [3], we precomputed the patterns from all possible combinations of the single bump exposures with time starting at $t = 0, 1, 2, \cdots, T - d$, where d is the exposure duration.

For the RCE sensor, the possible patterns were computed as follows: (1) generate the possible sets by choosing the reset combinations (8 bits) and transfer (8 bits) signals; (2) simulate the exposure pattern for all signal sets.

For the unconstraint sensor, we applied the same approach to prepare all possible patterns, and then chose the nearest pattern. We used simple thresholding to generate binary patterns, as has been done by Iliadis et al. [6] in experiments, seeing as this approach is not computationally effective.

4.2 Reconstruction layer

The reconstruction layer decodes high-frame videos from a single image compressed by using the learned exposure pattern, as was described in the previous section. This decoding expands the single image to multiple sub-frames by nonlinear mapping, which can be modeled and learned by a multi-layer perceptron (MLP). As illustrated in Figure 5, the MLP consisted of four hidden layers and each layer was truncated by rectified linear unit (ReLU). The network was trained by minimizing the errors between the training videos and the reconstructed videos. We used the mean squared error (MSE) as the loss function because it was directly related with the peak signal-to-noise ratio (PSNR).

5 Experiments

5.1 Experimental and training setup

The network size was determined based on the size of the patch volume to be reconstructed. We used the controllable exposure sensor [2], which exposes the 8 pixel block. Therefore, the volume size of $W_p \times H_p \times T$ was set to $8 \times 8 \times 16$ in the experiments. The reconstruction network had four hidden layers. We trained our network by using the SumMe dataset, which is a public benchmarking video summarization dataset that includes 25 videos. We choose 20 videos out of the available 25. The selected videos contained a relative variety of motion. We randomly cropped the patch volumes from the videos and augmented the directional variety of motions and textures by rotating and flipping the cropped patches. This resulted in 829,440 patch volumes. Subsequently, we used these patches in the end-to-end training of the proposed network to jointly train the sensing and reconstruction layers. In the training, we used 500 epochs with a minibatch size of 200.

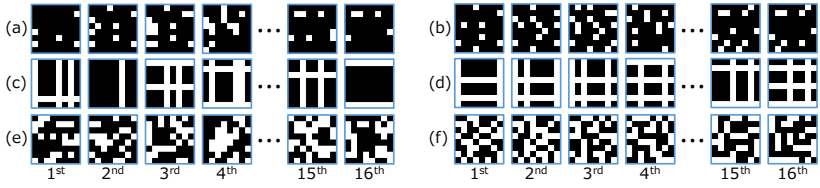


Fig. 7. Handcraft and optimized exposure pattern. (a)(b) single bump exposure (SBE) sensor [3] (c)(d) row-column wise exposure (RCE) sensor [2] (e)(f) unconstraint sensor

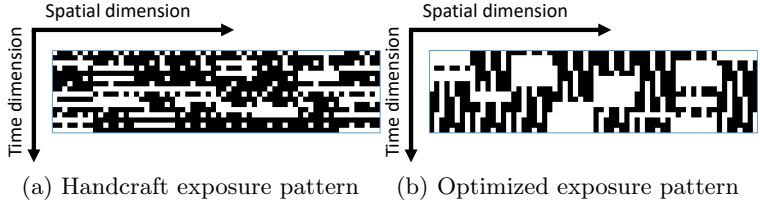


Fig. 8. Comparison of exposure patterns. The optimized exposure pattern indicates more smooth and continuous exposures after the training.

5.2 Simulation experiments

We carried out simulation experiments to evaluate our method. We assumed three different types of hardware constraints for the SBE, RCE, and unconstraint sensors. The details of the SBE and RCE sensor constraints are described in Section 3. The exposure pattern for an unconstrained sensor can independently control the exposure for each pixel and achieve perfect random exposure, which is ideal in signal recovery. The handcrafted pattern for the unconstrained sensors was random.

Figure 7a shows the handcraft exposure pattern of the SBE sensor. The exposure patterns indicates an exposed pixel in white color and an unexposed pixel in black color. Note that [3] used a patch volume size of $7 \times 7 \times 36$, and an exposure pattern. Instead, we used a size of $8 \times 8 \times 16$ to make a fair comparison with [2] under the same conditions. Figure 7b shows the optimized exposure pattern of the SBE sensor after training. This pattern still satisfies the constraint by which each pixel has a single bump with the same duration as that of other pixels.

Figure 7c shows the handcrafted exposure pattern of the RCE sensor. Figure 7d shows the optimized exposure pattern after training. The optimized pattern satisfied the constraints. Figure 8 compares the exposure patterns. We reshaped the $8 \times 8 \times 16$ exposure patterns to 64×16 to better visualize the space vs. time dimensions. The horizontal and vertical axes represent the spatial and temporal dimension, respectively. The original handcrafted pattern of the RCE sensor

indicates that the exposure was not smooth in the temporal direction, while the optimized exposure pattern indicates more temporary, smooth, and continuous exposures after the training. Similar results have been reported by [6], even though our study considered the hardware constraints in pattern optimization.

Figure 7e shows the random exposure pattern, and Figure 7f shows the optimized exposure pattern of the unconstraint sensor. The optimized patterns were updated by the training and generated differently than the random exposure patterns, which were used as the initial optimization patterns.

We generated a captured image simulated for the SBE, RCE, and unconstraint sensors. We input the simulated images to the reconstruction network to recover the video. We quantitatively evaluated the reconstruction quality by using the peak signal to noise ratio (PSNR). In the evaluation, we used 14 256×256 pixel videos with 16 sub-frames. Figure 9 shows two example results, which are named Car and Crushed can. The upper row (Car) of Figure 9 shows that, in our result, the edges of the letter mark were reconstructed sharper than in the result of the handcrafted exposure pattern. Additionally, the bottom row (Crushed can) shows that the characters were clearer in the optimized exposure pattern results, in comparison with the results of the handcrafted exposure pattern. The reconstruction qualities were different in each scene. However, the qualities in the optimized exposure pattern were always better than those of the handcrafted exposure pattern, regardless of whether SBE, RCE, or unconstraint sensors were assumed. Hence, the proposed framework effectively determined better exposure patterns under different hardware constraints and jointly optimized the reconstruction layer to suit these patterns. Table 1 shows the average PSNRs of the handcrafted and optimized results for the SBE, RCE, and unconstraint sensors. Owing to the pattern ' s joint optimization and the reconstruction layers, the proposed method always outperformed the original handcrafted patterns.

Table 1. Average peak signal-to-noise ratio (PSNR) of video reconstruction with different noise levels.

	Noise Level	Handcraft SBE	Optimized SBE	Handcraft RCE	Optimized RCE	Ramdom	Unconstraint
DNN (Ours)	0	29.41	30.05	28.51	29.45	29.17	29.99
	0.01	29.09	29.70	26.76	27.46	28.47	28.88
	0.05	25.61	25.95	19.85	20.22	23.08	22.05
GMM [4]	0	27.69	29.63	28.18	28.82	29.05	29.81
	0.01	27.54	29.29	26.27	26.57	28.13	28.09
	0.05	24.58	25.50	19.18	19.23	22.13	21.25
OMP [3]	0	24.66	26.22	22.96	24.22	24.27	25.83
	0.01	24.46	26.02	22.46	23.46	24.09	25.39
	0.05	21.56	23.32	17.59	17.42	21.21	20.54

We compared our DNN approach with the dictionary-based (OMP) [3] and GMM based [4] approaches. We trained the dictionary for OMP and GMM with

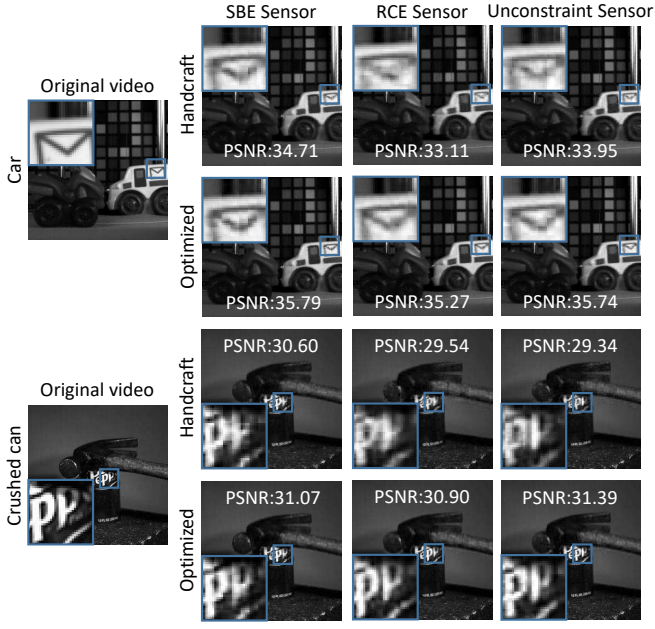


Fig. 9. Reconstruction results of 3rd sub-frame (DNN).

the same data used by the DNN, and set the number of dictionary elements to 5,000 for OMP, and the number of components in GMM to 20. These parameters were selected based on preliminary experiments. Additionally, we evaluated the video recovery from a noisy input to validate robustness. We added white Gaussian noise to the simulated captured image with different variances (the mean value was 0). Table 1 shows the average PSNR value between the ground truth video and the reconstructed video for the variances of 0, 0.01, and 0.05. Figure 10 shows the reconstruction results with different noise levels, as obtained by the DNN, GMM, and OMP. We did not add noise to the training of the DNN. Figure 10 shows that the images were degraded, while the PSNRs decreased when the noise increased by any method. The proposed DNN decoder was affected by the noise, but still achieved the best performance in comparison with the other decoders.

5.3 Real experiments

We conducted real experiments by using the real compressive image captured by the camera with the sensor reported by [19, 2]. Figure 11 shows the camera image used in the real experiment. The compressed video was captured with 15 fps. We set 16 exposure patterns per frame. Thus, the reconstructed video was equivalent to 240 fps after recovering all of the 16 sub-frames. We set the exposure

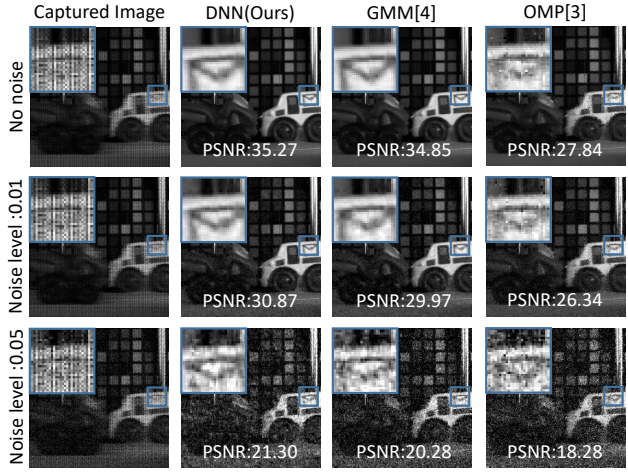


Fig. 10. Reconstruction results of 3rd sub-frame with different noise levels by the deep neural network (DNN), Gaussian mixture models (GMM), and orthogonal matching pursuit (OMP) (exposure pattern: optimized RCE)

Fig. 11. Camera used in real experiment.

pattern obtained by the sensing layer of the proposed network after the training. Moreover, we reconstructed the video from the captured image by reconstructing the layer of the proposed network. The sensor had a rolling shutter readout and temporal exposure patterns, which were temporally shifted according to the position of the image ' s row. The shifted exposure pattern was applied every 32 rows (four blocks with a 8 patch), in the case where the resolution of the sensor was 672×512 pixels and the number of exposure patterns was 16 in one frame. For example, the actual sub-exposure pattern was applied to the first four blocks as the 0-15 sub-exposure pattern, the second four blocks were applied as the 1-15, 0 pattern, the third four blocks were applied as the 2-15, 0, 1 pattern, and so on. Hence, we trained 16 different reconstruction networks to apply the variety of shifted exposure patterns. We used these different reconstruction networks every 32 rows in an image. Figure 12 shows the single frame of the real captured image and three of the 16 reconstructed sub-frames. The upper row shows that a moving pendulum appeared at a different position in the reconstructed sub-

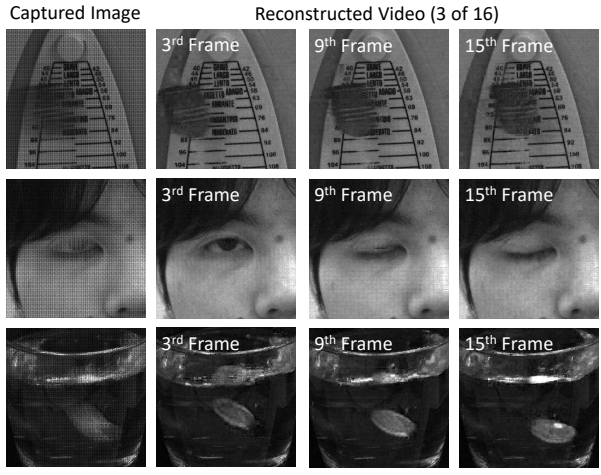


Fig. 12. Reconstruction results. The left column shows the captured image; left of the second column are the 3rd, 9th, and 15th frames of the reconstructed video.

frames, and the motion and shape were recovered. The second row shows the blinking of an eye, and the bottom row shows a coin dropped into water. Our method successfully recovered very different appearances; namely, the swinging pendulum, closing eye, and moving coin. Because the scene was significantly different from the videos included in the training dataset, these results also demonstrate the generalization of the trained network.

6 Conclusion

In this paper, we first argued that real sensor architectures for developing controllable exposure have various hardware constraints that make non-practical the implementation of compressive video sensing based on completely random exposure patterns. To address this issue, we proposed a general framework that consists of sensing and reconstruction layers by using a DNN. Additionally, we jointly optimized the encoding and decoding models under the hardware constraints. We presented examples of applying the proposed framework to two different constraints of SBE, RCE, and unconstraint sensors. We demonstrated that our optimal patterns and decoding network realized the reconstruction of higher quality video in comparison with handcrafted patterns in simulation and real experiments.

Acknowledgement

This work was supported by JSPS KAKENHI (Grant Number 18K19818).

References

1. Kleinfelder, S., Lim, S., Liu, X., El Gamal, A.: "A 10000 frames/s CMOS digital pixel sensor." IEEE Journal of Solid-State Circuits 36.12 (2001) 2049-2059
2. Sonoda, T., Nagahara, H., Endo, K., Sugiyama, Y., Taniguchi, R.: "High-speed imaging using CMOS image sensor with quasi pixel-wise exposure." International Conference on Computational Photography (ICCP). (2016) 1-11
3. Hitomi, Y., Gu, J., Gupta, M., Mitsunaga, T., Nayar, S. K.: "Video from a single coded exposure photograph using a learned over-complete dictionary." International Conference on Computer Vision (ICCV). (2011) 287-294
4. Yang, J., Yuan, X., Liao, X., Llull, P., Brady, D. J., Sapiro, G., Carin, L.: "Video compressive sensing using Gaussian mixture models." IEEE Transactions on Image Processing 23.11 (2014) 4863-4878
5. Iliadis, M., Spinoulas, L., Katsaggelos, A. K.: "Deep fully-connected networks for video compressive sensing." Digital Signal Processing 72 (2018) 9-18
6. Iliadis, M., Spinoulas, L., Katsaggelos, A. K.: "Deepbinarymask: Learning a binary mask for video compressive sensing." arXiv preprint arXiv:1607.03343 (2016)
7. Sarhangnejad, N., Lee, H., Katic, N., O ' Toole, M., Kutulakos, K., Genov, R.: "CMOS Image Sensor Architecture for Primal-Dual Coding." International Image Sensor Workshop. (2017)
8. LeCun, Y., Bengio, Y., Hinton, G.: "Deep learning." nature 521.7553 (2015) 436
9. Dadkhah, M., Deen, M. J., Shirani, S.: "Compressive sensing image sensors-hardware implementation." Sensors 13.4 (2013) 4961-4978
10. Robucci, R., Gray, J. D., Chiu, L. K., Romberg, J., Hasler, P.: "Compressive sensing on a CMOS separable-transform image sensor." Proceedings of the IEEE 98.6 (2010) 1089-1101
11. Dadkhah, M., Deen, M. J., Shirani, S.: "Block-based CS in a CMOS image sensor." IEEE Sensors Journal 14.8 (2014) 2897-2909
12. Majidzadeh, V., Jacques, L., Schmid, A., Vandergheynst, P., Leblebici, Y.: "A (256 × 256) pixel 76.7 mW CMOS imager/compressor based on real-time In-pixel compressive sensing." International Symposium on Circuits and Systems (ISCAS). (2010)
13. Oike, Y., El Gamal, A.: "CMOS image sensor with per-column $\Sigma\Delta$ ADC and programmable compressed sensing." IEEE Journal of Solid-State Circuits 48.1 (2013) 318-328
14. Spinoulas, L., He, K., Cossairt, O., Katsaggelos, A.: "Video compressive sensing with on-chip programmable subsampling." IEEE Conference on Computer Vision and Pattern Recognition Workshops (2015)
15. Aharon, M., Elad, M., Bruckstein, A.: "K-SVD: An algorithm for designing over-complete dictionaries for sparse representation." IEEE Transactions on signal processing 54.11 (2006) 4311-4322
16. Pati, Y. C., Rezaiifar, R., Krishnaprasad, P. S.: "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition." The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers (1993)
17. Tibshirani, R.: "Regression shrinkage and selection via the lasso." J ournal of the Royal Statistical Society. Series B (Methodological) (1996) 267-288
18. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1." arXiv preprint arXiv:1602.02830 (2016)

19. Hamamatsu Photonics K.K.: "Imaging device" Japan patent JP2015-216594A (2015)