Research Article | Vol. 30, No. 2 / 17 Jan 2022 / Optics Express 887

Optics EXPRESS

Shuffled rolling shutter for snapshot temporal imaging

ESTEBAN VERA,* FELIPE GUZMÁN, AND NELSON DÍAZ

School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
*esteban.vera@pucv.cl

Abstract: We propose a modification to the rolling shutter mechanism found in CMOS detectors by shuffling the pixels in every scanline. This potential hardware modification improves the sampling of the space-time datacube, allowing the recovery of high-speed videos from a single image using either tensor completion methods or reconstruction algorithms often used for compressive temporal video. We also present a design methodology for optimal sampling schemes and compare them to random shuffling. Simulations, and experimental results obtained by optically emulating the hardware, demonstrate the ability of the shuffled rolling shutter to capture images that allow reconstructing videos, which would otherwise be impossible when using the traditional rolling shutter mechanism.

1. Introduction

CMOS imaging sensors are very popular in consumer and surveillance cameras due to their reduced cost [1], despite their low quality in contrast with CCDs. Nevertheless, as fabrication technologies have improved, CMOS detectors are becoming the new standard in high-end cell phones and professional and scientific cameras. They started to offer higher quantum efficiencies and reduced noise levels even competitive to the most expensive EMCCDs. However, most CMOS sensors are designed with a column-parallel readout circuitry known as the Rolling Shutter (RS). All scanlines share the readout circuitry, so scanlines are readout sequentially at different times. Also, the starting time for each scanline is delayed, accordingly, to maintain the same exposure time per row. As such, the RS unavoidably generates unwanted distortions such a wobble, skew, and other image artifacts [2] when imaging dynamic scenes.

The RS-derived distortions are considered a nuisance to the imaging performance. Hence, there is a variety of research devoted to compensating them by different means [3–7]. On the other hand, some research approaches have also pointed out that if we consider that each row is exposed at different times, then a single RS snapshot has the potential to sample the space-time volume in a very particular and deterministic way. Since dynamic information could potentially be extracted from an RS shot, some recent efforts for compressive temporal imaging have been presented in [8] and [9], where spatial multiplexing is added to the RS deterministic spatio-temporal sampling through the use of diffusers at either the imaging or pupil plane, respectively, achieving remarkable reconstruction frame rates from a single RS image. Nevertheless, additional multiplexing of light comes at the expense of optical calibration and a reduced signal-to-noise. Lately, another approach based on an array of RS cameras placed at different orientations allowed to demonstrate that by sampling different spatial positions simultaneously, then the sampling of the space-time datacube is improved, being able to reconstruct videos from a set of simultaneous shots from the array [10].

In this work, we tackle the inherent lack of sampling diversity given by the RS, proposing a slight hardware-only modification to the RS scanline mechanism. In particular, we propose to shuffle the scanline, so it is not reading out the pixels from a unique row at a time. Instead, it is sampling pixels from different rows at every column, not following a straight line. Then, if feasible, it is a matter of hard-wiring the shuffled scanlines. In the end, the shuffled RS will

#444864 https://doi.org/10.1364/OE.444864
Journal © 2022 Received 6 Oct 2021; revised 12 Dec 2021; accepted 13 Dec 2021; published 3 Jan 2022

Optics EXPRESS

work similar to the original RS, sampling all pixels once a snapshot acquisition is over. However, pixels from diverse spatial positions–belonging to the same scanline–may be sampled at the same time, improving the sampling diversity of the space-time datacube. Although the shuffled RS will most likely generate a new class of distortions when sampling moving scenes, the information contained in the captured image may be more informative regarding motion within the scene, enabling the extraction of videos from a snapshot.

The question that arises from the novel shuffled RS scheme is: how to perform the shuffle? The naïve approach would be to use a random shuffle. As an alternative, we present elements for optimally designing the shuffled coding to boost the ability to recover videos. Given the restrictions of the RS–that no pixel is repeated and only one pixel is sampled per column for every scanline–our coding design is based on solving the Kepler conjecture [11] for sphere packing [12], which leads to finding a solution to a 3-dimensional N^2-Queens problem [13]. Our designed shuffled RS coding solutions lead to a homogeneous sampling of the space-time datacube, potentially improving the recovery of videos from the shuffled RS snapshots.

Apart from computer simulations to model, test, and validate our proposed shuffled RS, we also implement an optical system inspired by what has been used for compressive temporal imaging applications–such as the CACTI [14,15] or similar in [16,17]–making use of dynamic coded apertures to replicate the functioning of the RS. By assuming short exposure times, we can treat the recovery of the space-time datacubes–videos–from the snapshots as a tensor completion problem. We implement three reconstruction alternatives to demonstrate the sampling ability of the shuffled RS and compare the advantages of coding design instead of random shuffling. We use 3-D interpolation [18], a state-of-the-art tensor completion algorithm [19], and an adaptation of the latest deep neural network solution used in CACTI problems [20].

Our contributions can be summarized as follows. Firstly, we propose a novel sampling scheme for the space-time datacube using a modified RS mechanism. Secondly, we also propose a method for optimally coding the shuffled RS that is consistent with the RS restrictions. Finally, through simulations and experiments we validate that–if ever implemented in the CMOS hardware–the proposed idea allows to recover videos from snapshots without the need for optical coding.

2. Observation model

In this section, we review how to model a generalized RS detector and its interaction with the space-time datacube. If we appropriately discretize the time domain, then we can treat the RS, or the proposed shuffled RS, as a dynamic coded aperture applied to the imaging plane.

2.1. Continuous model

In any detector class, the acquisition process is mainly dominated by three temporal factors: the reset time, the exposure time, and the readout time. As shown in Fig. 1, if the three steps are simultaneous across all rows of the sensor, then we are in the presence of a Global–electronic–Shutter. However, in the CMOS Rolling Shutter acquisition mechanism, the timing for every consecutive row is displaced to assure that only one row is being read within the readout time.

The acquisition process for an imaging detector can be modeled as a temporal integral at the sensor (image) plane such that

$$Y(x,y) = \int_0^\infty V(x,y,t) \cdot S(x,y,t)dt, \tag{1}$$

where (x, y) are the discrete pixel indices of the sensor, V is the time-varying optical intensity distributed on the image sensor plane (i.e., space-time datacube), and $S \in \{0, 1\}$ represents the binary shutter function. For the rolling shutter, the characteristics of S are determined by the acquisition parameters: Readout time t_r, the exposure time t_e, and the delay time t_d.

Optics EXPRESS

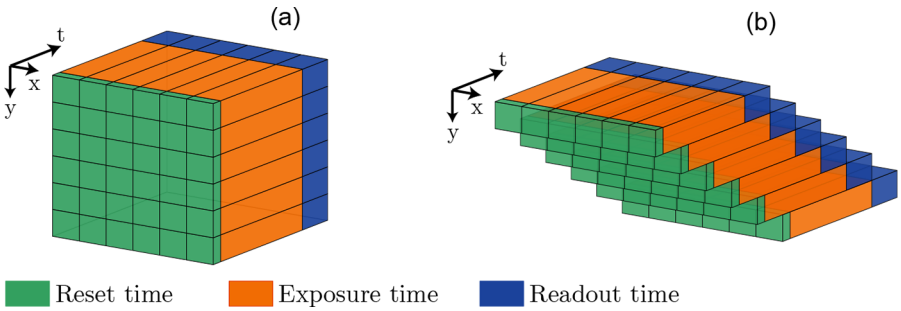

Fig. 1. Electronic shutter comparison. (a) simultaneous readout: Global Shutter; (b) row-by-row time-shifted readout: Rolling Shutter.

2.2. Discrete model

In the discrete-time regime, the model of the RS mechanism can be described as follows

$$\mathbf{Y} = \sum_{l=1}^{N_t} \mathbf{X}_l \odot \mathbf{C}_l + \mathbf{\Omega}, \tag{2}$$

where N_t is the number of frames in the datacube, $\mathbf{X}_l \in \mathbb{R}^{N_x \times N_y}$ is the l^{th} frame with $N_x \times N_y$ number of pixels, $\mathbf{C}_l \in \mathbb{R}^{N_x \times N_y}$ is the l^{th} coded aperture, and $\odot$ is the Hadamard product. Also, $\mathbf{\Omega}$ is assumed to be a Gaussian readout noise component.

To analyze the different RS versions, we will neglect the reset time and assume that the exposure time is equivalent to the readout time, becoming the discrete-time unity. In this way, we can assure that only one scanline is sampled at a given frame time. In a traditional CMOS with RS, $\mathbf{C}_l$ is defined row by row, as shown in Fig. 2(a).

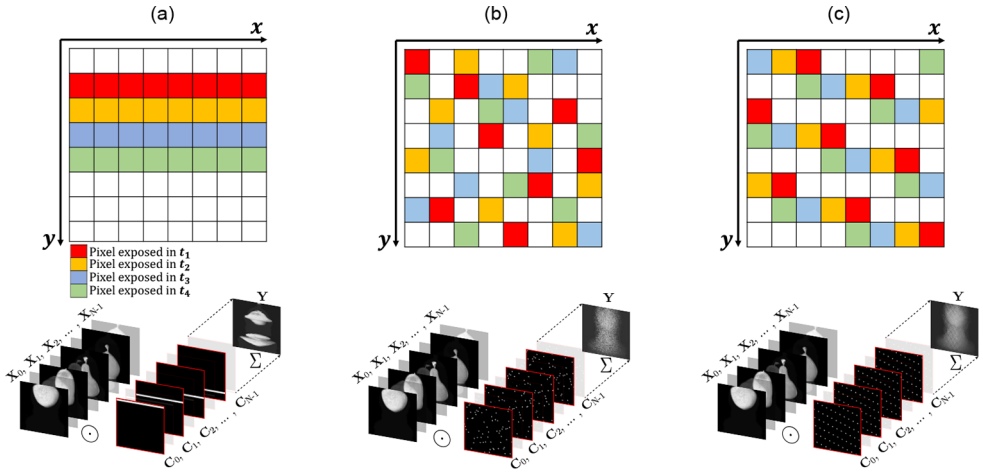

Fig. 2. Rolling shutter sampling. (a) sequential rolling shutter sampling; (b) randomly shuffled rolling shutter; (c) designed shuffled rolling shutter.

3. Shuffled rolling shutter code design

We propose to shuffle the position of the pixels being scanned by the RS mechanism. In this way, a scanline is not a single row anymore, but rather it is distributed at different column positions. A

Optics EXPRESS

randomly shuffled RS is shown in Fig. 2(b), where we can notice that only one pixel is sampled per column at a given time, in the same way as the original straight RS scanline.

We wonder how to choose the shuffling mechanism to make the sampling of the space-time datacube as uniform as possible, in principle in the spatial domain, avoiding clusters of contiguous pixels. If we leave it as a random permutation, we can still end up with solutions where two neighboring pixels are being sampled. For example, in Fig. 2(c) we show a designed scheme that provides a more homogeneous sampling of the spatial domain. However, if we also consider that an extended exposure time will make the sampling for a given pixel persistent to the following frames, then we also have to consider the distance between voxels of the datacube from different frames, making the sampling in all dimensions more even.

3.1. Shuffled RS coding optimization

The design and optimization of the coded aperture for the shuffled RS can be seen as a sphere packing problem in a cubic container [12]. Specifically, let n be the number of spheres to be packed in a cubic container. Assuming a square detector, the size of the cubic container $(N_x + 1)^3$ is determined by N_x, which corresponds to the RS detector's pixels in one direction. Notice that the size of the cubic container is augmented by 1 by assuming points packing, which is an equivalent problem to the sphere packing. According to the Kepler conjecture [11,21,22], no packing of congruent balls in a three-dimensional Euclidean space has a density higher than that of the face-centered cubic packing, which corresponds to $\rho = \frac{\pi}{3\sqrt{2}} = 0.7405$. The RS restriction where scanlines only have one pixel per column and the fact that the sampled pixels are fixed forcing the center of the spheres to be fixed as well, finally constrain the sphere packing problem to be approximately equivalent to a three-dimensional queens problem ($3DN_x^2$QP) [13]. The N_x queens problem is the one of placing N_x chess queens on an $N_x \times N_x$ chessboard so that two queens never attack each other. Similarly, the 3D N_x^2 queens problem deals with placing N_x^2 chess queens on N_x chessboards with size $N_x \times N_x$, so that two queens never attack each other along any dimension. The shuffled RS imposes uniqueness in columns and the temporal axis while providing several degrees of freedom in the diagonals, turning the optimization for the shuffled RS a relaxed version of the $3DN_x^2$QP. Therefore, the proposed RS coded aperture patterns can be found in the solution space for the $3DN_x^2$QP as follows

$$\mathbf{G} = ((a \odot \mathbf{I} + b \odot \mathbf{J}) \mod N_x) + \mathbf{1}, \tag{3}$$

where $\mathbf{I} = \mathbf{x}^T \otimes \mathbf{y}$ such that $\mathbf{I} \in \mathbb{R}^{N_x \times N_x}$, being $\mathbf{x}$ a vector of all ones such as $\mathbf{x} \in \mathbb{R}^{N_x}$, and $\mathbf{y} = [1, \ldots, N_x]^T$ such as $\mathbf{y} \in \mathbb{R}^{N_x}$, $\odot$ denotes the Hadamard product, and $\otimes$ represents the Kronecker product. Matrix $\mathbf{J}$ is given by $\mathbf{J} = \mathbf{x} \otimes \mathbf{y}^T$ such that $\mathbf{J} \in \mathbb{R}^{N_x \times N_x}$, and $\mathbf{1} \in \mathbb{R}^{N_x \times N_x}$ is a all ones matrix. If we want to make the sampling as uniform as possible in both the spatial and temporal directions, then we have to search for the pair of parameters a and b that maximize the distance between the centers of the spheres. Then, the resulting optimal RS coded aperture is:

$$C_{i,j,l} = \begin{cases} 1 & \text{if } l = G_{i,j} \\ 0 & \text{if } l \neq G_{i,j}. \end{cases} \tag{4}$$

Matrix $\mathbf{G}$ can be reorganized as $\mathbf{p}_u = [i, j, G_{i,j}]^T$ such that $\mathbf{p}_u \in \mathbb{R}^3$ denote the centers of $n = N_x N_y$ spheres, where $u \in \{1, \ldots, n\}$. These centers represent active positions in the coded aperture $C_{i,j,l}$, where $i \in \{1, \ldots, N_x\}$, $j \in \{1, \ldots, N_y\}$, and $l \in \{1, \ldots, N_t\}$, being N_t the number of frames. Thus, the distance between a set of n spheres is given by

$$d^*(n) = \min_{1 \leq u < v \leq n} \|\mathbf{p}_u - \mathbf{p}_v\|_2, \tag{5}$$

where $\mathbf{p}_u$ and $\mathbf{p}_v$ are the centers of the u^{th} and v^{th} sphere, respectively. Exploiting the Kepler conjecture and knowing the volume of the spheres, we deduce the theoretical upper bound for the

Optics EXPRESS

RS sphere packing density as

$$d(n) = 2\sqrt[3]{\frac{(\sqrt{n}+1)^3}{4n\sqrt{2}}}, \tag{6}$$

where the optimal density of sphere packing is $\rho = 0.7405 = \frac{\pi}{\sqrt{18}} = \frac{4n\pi r^3}{3(\sqrt{n}+1)^3}$, noting that $N_x = \sqrt{n}$. The optimal known unconstrained sphere packing solutions (available in the magentaPackomania web site.) are compared with the RS sphere packing in Fig. 3. Clearly, the RS restrictions lead to lower densities, though we postulate that the optimal sphere packing density for the shuffled RS tends to $\rho_{RS} \approx 0.696$ for higher dimensions.

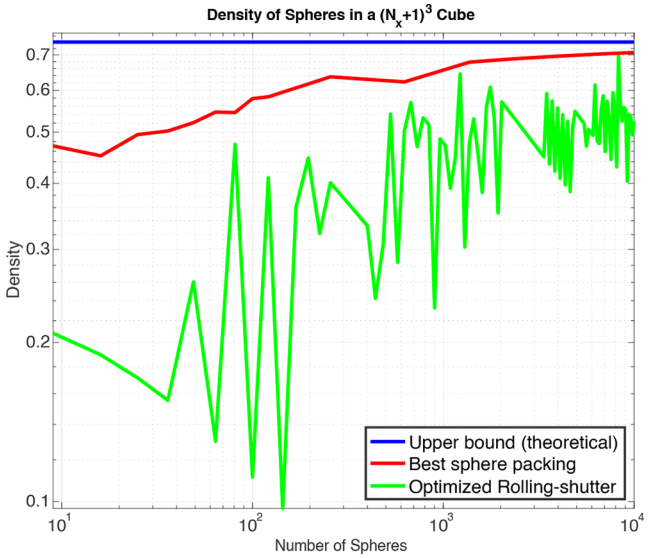

Fig. 3. Sphere packing density comparison for different dimensions. (blue line) theoretical upper bound where the cubic container has size $(N_x + 1)^3$, and the total number of spheres is $n = N_x N_y$; (red line) best known optimal sphere packing without the RS restriction; (green line) sphere packing density with RS restriction.

As an example, Fig. 4 depicts a comparison of random and designed low-dimensional sampling codes $\mathbf{G}$ for a $9 \times 9 \times 9$ datacube obtained using Eq. (3). The sampling scheme for every scanline is shown at the left while the 3D representation for the sampling points in $\mathbf{G}$ are shown as spheres in the middle. Note that the colors in the codes and spheres represent different instants in time, and the spheres' radius are related to the minimum distance between all of them. The minimum distance between every sphere and its nearest neighbor is shown on the right, where the designed codes provide a uniform distance between the sampled voxels within the datacube. In fact, the minimum distance between the center of the spheres for the random pattern of 81 balls is $\min(\mathbf{D}) = 1.4142$, while for the designed pattern is $\min(\mathbf{D}) = 2.2361$, as shown by the larger touching spheres depicted in the sphere packing.

3.2. *Parameters selection algorithm*

We propose Algorithm 1 to find the optimal parameters a and b for the shuffled RS code design. For a square detector, the search space for a and b is symmetric and it has a size of $N_x \times N_x$. An example of the search space of size 100×100 is depicted in Fig. 5(a). To reduce the search space, the algorithm exploits the symmetries as seen in Fig. 5(b). The color on both search spaces in Fig. 5 denotes the minimum distance between the center of the spheres, obtained from matrix $\mathbf{D}$.

Optics EXPRESS

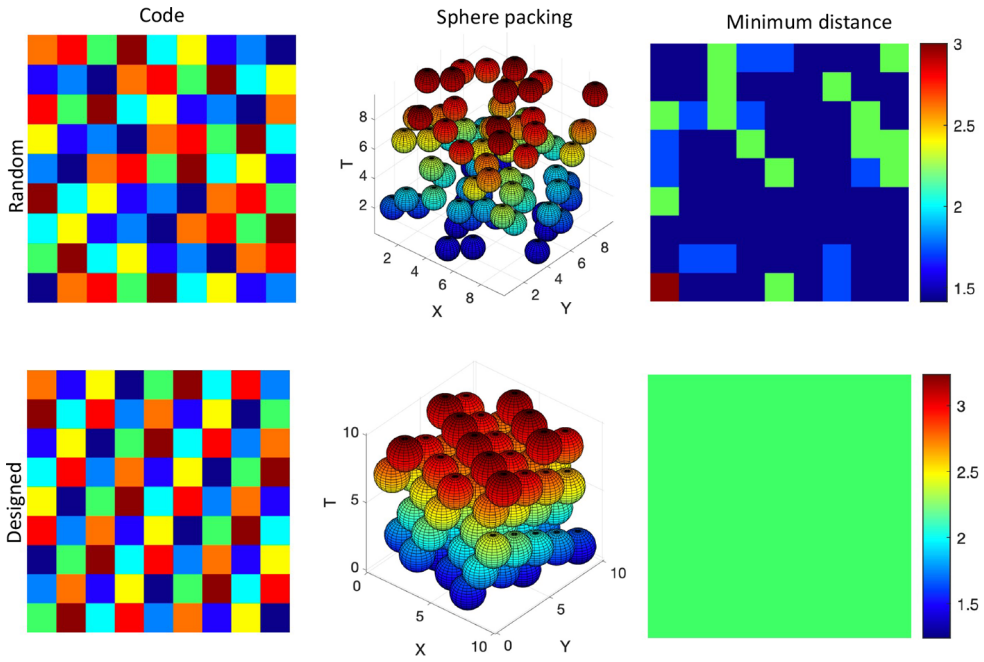

Fig. 4. Comparison between random and designed shuffled RS sampling codes. (upper row) random shuffle; (lower row) designed shuffle; (left) code design for the shuffled RS scanlines represented by different colors; (middle) sphere packing using a radius $d = \frac{D_{a,b}}{2}$ that depends on the minimum distance between voxels; (right) minimum distance matrix between sampled voxels.

In essence, it is only necessary to select the entries of $\mathbf{f} \in \mathbb{R}^M$ such that f_m and N_x are relatively prime, where $M = \lfloor \frac{N_x}{2} \rfloor$. Two integer numbers f_m and N_x are relatively prime, or also called co-prime, if the greatest common divisor between f_m and N_x is 1. The resulting vector is $\mathbf{s} \in \mathbb{R}^K$, and only contains the co-primes with respect to N_x. The coded aperture is computed using Eq. (3)

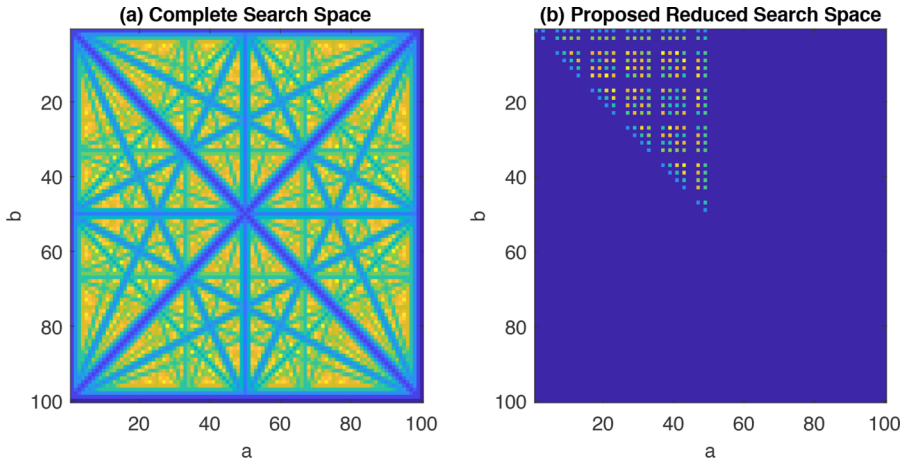

Fig. 5. Comparison of the complete search space against the reduced search space using the proposed algorithm to select the parameters a and b for the shuffled RS code design.

Optics EXPRESS

Algorithm 1 Selection of a and b based on search space reduction using the co-prime assumption.

Input: N_x ▷ Inputs of the algorithm.
Output: a and b. ▷ Output of the algorithm.

function PARAMETER SELECTION(N_x)
  $M = \lfloor \frac{N_x}{2} \rfloor$
  $\mathbf{f} \leftarrow [1, \ldots, M]$
  $k = 0$
  for $m \leftarrow 1, M$ **do**
    if GCD(f_m, N_x)=1 is true **then** ▷ To select only entries of $\mathbf{f}$ such that GCD(f_m,N_x)=1
      $k = k + 1$
      $s_k \leftarrow f_m$
  $K \leftarrow$ **Size of s** ▷ $\mathbf{s} \in \mathbb{R}^K$
  for $i \leftarrow 1, K$ **do**
    for $j \leftarrow i, K$ **do**
      $\mathbf{G} \leftarrow$ **Coded aperture RS**(s_i, s_j) ▷ Computes $\mathbf{G}$ by solving Eq. (3).
      $q \leftarrow$ **Verify RS restriction**($\mathbf{G}$) ▷ Verify Rolling Shutter restriction.
      if q is true **then**
        $D_{s_i,s_j} \leftarrow$ **Compute distance**($\mathbf{G}, N_x$) ▷ Computes the minimum distance matrix in a patch of size $\sqrt{N_x} \times \sqrt{N_x}$ of the matrix $\mathbf{G}$.
  max $\mathbf{D}$ ▷ Select $a = \bar{s}_i$ and $b = \bar{s}_j$ in $\mathbf{D}$ such that $D_{a,b}$ has the maximum value.
  return a and b ▷ Output of the algorithm.

and 4 according to the selected $a = s_i$ and $b = s_j$ at each iteration. To verify the RS restriction, the designed entries in the RS matrix, $G_{i,j}$, should have an unique value along the corresponding column. The minimum distance between the centers of the spheres is computed for a patch of size $\sqrt{N_x} \times \sqrt{N_x}$, which is stored in D_{s_i,s_j}. The coordinates $D_{\bar{s}_i,\bar{s}_j}$ that correspond to the column and row with the maximum value in the distance matrix $\mathbf{D}$ denotes the optimal values for the parameters a and b, $a = \bar{s}_i$ and $b = \bar{s}_j$, becoming the output of the algorithm.

4. Video reconstruction

If we select a short exposure time such that there is no temporal overlap between scanlines, we can consider the sampling of the shuffled RS as a random–or pseudo-random, if designed–sampling of the discrete video datacube. Therefore, every pixel of the shuffled RS is sampling one voxel, making the recovery of the video from a single snapshot equivalent to a tensor completion problem, i.e., filling the holes not sampled in the datacube by using the information provided by the sampled values. Since the amount of frames is proportional to the number of pixels N_x in one of the spatial domains (rows), we may end up with only $\frac{100N_x^2}{N^3} = \frac{100}{N_x}\%$ of valid samples from the whole datacube, becoming a tough tensor completion problem from a sparse set of samples as the dimensionality increases. Despite losing temporal resolution, and to ease the problem, we designed two approaches to shorten the datacube in the temporal domain by combining frames. We start by expanding the measurements $\mathbf{Y}$ into the datacube $\hat{\mathbf{X}}_l$ by using the coded apertures C_l such that

$$\hat{\mathbf{X}}_l = C_l \odot \mathbf{Y}. \tag{7}$$

Then, the first approach combines $V = 2r$ frames into the j^{th} output frame according to

$$\bar{\mathbf{X}}_j = \begin{cases} \sum_{j=0}^{l+r-1} \hat{\mathbf{X}}_l & l<r+1 \\ \sum_{j=l-r}^{l+r-1} \hat{\mathbf{X}}_l & r+1 \le l \le N_t-(r+1) \\ \sum_{j=l-r}^{N_t} \hat{\mathbf{X}}_l & l>N_t-(r+1). \end{cases}$$

Optics EXPRESS

This overlapped method is used to perform Nearest Neighbor Interpolation (NNI) [18], recovering up to N_t frames. Although NNI does not implicitly exploit prior information to perform the reconstruction, the inherent spatial-temporal correlation of videos of natural scenes may contribute to the reconstruction quality. Our second approach consists in combining r non-overlapped frames of $\hat{\mathbf{X}}_l$, increasing the number of samples per frame from $\frac{1}{N_r}$ to $\frac{r}{N_r}$, where $r<<N_r$ is

$$\bar{\mathbf{X}}_j = \sum_{l=(jr)}^{(j+1)(r-1)} \hat{\mathbf{X}}_l. \tag{8}$$

This approach is used in the tensor completion method, able to recover $\frac{N_t}{r}$ frames. It is well-known that tensor completion methods barely work when having less than a 10% of sampling density, so this is important to define the number of frames to be combined. For this work, we choose the Low-Rank Tensor Completion using Total Variation (LRTC-TV-II) method [19] since it is the one that demonstrated to work with the smallest densities, while the TV regularizer exploits spatial correlations to improve the reconstruction quality of the frames.

Finally, RevSCI-net [20] was chosen, being one of the latest deep learning-based methods designed for the CACTI system. In this case, there is no need to expand the tensor before entering the deep neuronal network. On the other hand, the network has to be trained end-to-end with the combined 3D coded aperture at the desired target density per frame, while the measured image is used as the input to the neural net. Due to the 3D-convolutional layers of the network, both temporal and spatial correlations of the videos are exploited by the 3D-kernels.

5. Simulations

Using the linear model of Eq. (2) and three algorithmic options for video reconstruction, we perform simulations of the acquisition and reconstruction process for a variety of short image sequences. We use a total of 38 datasets with a spatial resolution fixed at 256×256 pixels and a temporal resolution of 256 frames. For the reconstruction, we use the NNI [18] interpolation algorithm, LRTC-TV-II [19] tensor completion algorithm, and the RevSCI-net [20] compressive temporal imaging reconstruction adapted to the shuffled RS equivalent coded aperture masks. The chosen reconstruction datacube dimension has 256×256 pixels, but only 32 frames. Therefore the original datacubes are reduced to the same size integrating over every eight frames to perform fair comparisons against the reconstructed videos.

We trained what we call Tensor-RevSCI-net (TRevSCI-net), since it is a slight modification to the original RevSCI-net adapted to perform tensor completion instead of compressive reconstruction. Details of the network are shown in Supplement 1. We collected a dataset of 50 different grayscale videos of 512×512×400 used for data augmentation with temporal translation, frame rotations, and magnification to increase the training set to a total of 5.000 datacubes (4950 for training and 50 for validation) of 256×256×32. Five additional videos were used for testing. The training parameters were 100 epochs, a learning rate of 0.0001, and a batch size of 2. We used a desktop computer equipped with a dual RTX3090 Nvidia GPU. Since this network is not capable of generalizing an arbitrary input mask, two networks were trained: one for the random coded aperture and the other for the designed coded aperture.

We use two performance metrics to evaluate the quality of every reconstructed frame: the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM). For the shuffled RS, we use two coding options: a random shuffle and a designed shuffled pattern maximizing the distance between sample voxels within the datacube. The random shuffled coding is obtained through permutations of the original scanlines in the RS. The designed patterns are computed using Eq. (4) once selecting an optimal combination of parameters a and b for the 3D-N^2-queens problem following the optimization procedure in Algorithm 1. In this case, a pair of optimal a

Optics EXPRESS

and b parameters are selected for the experiments, corresponding to $a = 39$ and $b = 7$ for the $256 \times 256 \times 256$ datacube.

5.1. Results

For each sampling scheme, one snapshot acquisition is obtained using the forward model in Eq. (2). Examples for the traditional RS and the random and designed shuffled RS measurements are depicted in Fig. 6(a), while Fig. 6(b) displays the ground-truth video. Figure 6(c,d,e) show the frames for the reconstructed videos from the different methods. Complimentary results can also be seen in Supplement 1.

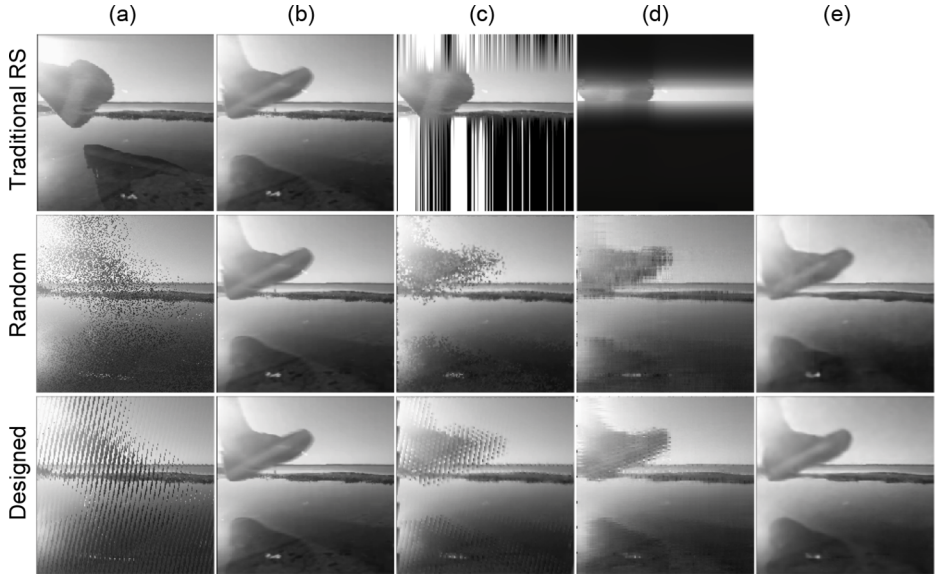

Fig. 6. Simulation results using different reconstruction algorithms for the traditional RS, random and designed shuffled RS coded apertures (see Visualization 1). (a) shuffled RS measurement; (b) ground-truth frames; (c) NNI reconstructed frames; (d) LRTC-TV-II reconstructed frames; (e) TRevSCI-net reconstructed frames.

As expected, notice that when using the traditional RS, only partial parts of the scene can be reconstructed in every frame. The comparison in terms of the PSNR and SSIM for all the 38 datasets is shown in Table 1.

Although the tensor completion method seems to be outperformed by the interpolation in quantitative terms, from the videos in Visualization 1 and Visualization 5 we may have the perception that it is the opposite. Nonetheless, the neural network approach clearly outperforms both mathematical-driven methods by large. This fact is confirmed when inspecting the videos in Visualization 6 and Visualization 7, which are reconstructed using different temporal resolutions. Also, we can observe from Table.1 that the designed shuffled RS slightly wins over the random shuffle, although visually, the designed clearly brings smoother videos for all reconstruction methods.

Finally, Fig. 7 depicts measurements from 12 out of the 38 datasets solely using the designed shuffled RS, and the corresponding video frames reconstructed using the neural network approach alongside. From the video associated with Fig. 7, we can notice that motion within the scene generates a new class of artifacts and distortions in the sampled images, while reconstructing fairly smooth videos from them.

Optics EXPRESS

Table 1. A comparison of the video reconstruction quality in terms of PSNR and SSIM. From the average values obtained for the 38 datasets, we display the mean, minimum, and maximum performance obtained by every method and coding. The designed and random coded apertures are tested with three reconstruction algorithms.

	NNI			LRTC TV-II			TRevSCI-net		
	PNSR [dB]								
	mean	min	max	mean	min	max	mean	min	max
Random	20.38	14.10	26.07	15.18	9.71	22.10	28.87	24.50	34.05
Designed	20.30	14.19	25.86	18.13	10.20	26.07	29.05	24.47	34.93
	SSIM								
	mean	min	max	mean	min	max	mean	min	max
Random	0.664	0.310	0.890	0.539	0.147	0.859	0.899	0.741	0.961
Designed	0.685	0.339	0.894	0.652	0.180	0.876	0.900	0.740	0.963

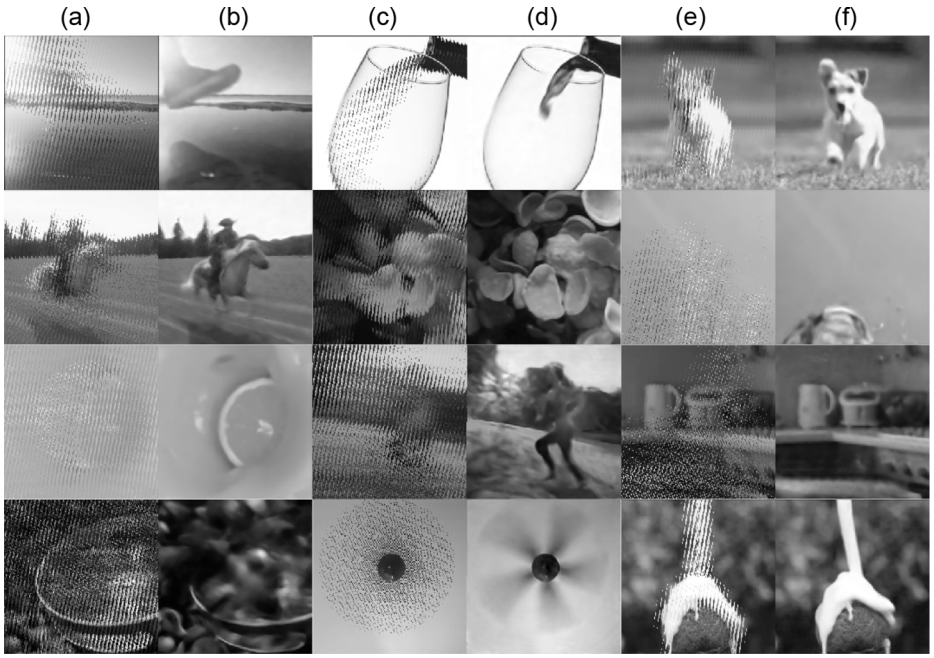

Fig. 7. Simulation results for 12 of the test datasets using the designed shuffled RS and TRevSCI-net reconstructions (see Visualization 2). Every pair of images shows: (a,c,e) the RS measurement; (b,d,f) the corresponding reconstructed video frames.

6. Experimental results

To demonstrate the effectiveness of the proposed system and the use of optimal shuffled RS codes, we implemented an optical setup capable of emulating the spatial and temporal functioning offered by the shuffled RS. The system is based on the CACTI design that uses a digital micromirror device (DMD) for dynamic coding, as presented in [15]. As depicted in the scheme and picture in Fig. 8, the proof-of-concept setup is a 4f system composed of a coded aperture implemented using a DMD located at an imaging relay plane, which is relayed again to be integrated by a global shutter detector. The objective lens is a 50mm Ricoh lens, the DMD is a high-speed Vialux V-9501 with a resolution of 1920×1080, and the microscopic objective lens is a 4X Olympus apochromatic. The detector is a global shutter Grasshopper CMOS sensor GS3-U3-23S6M with

Optics EXPRESS

1920×1200 pixels. To correct the depth of focus given by the DMD orientation relative to the objectives lens (a), a tilt of 22° was added to the objective lens. Also, a 45° rotation was used in both DMD and detector planes to compensate for the orientations of the DMD mirror actuators. Furthermore, the presented results are obtained considering a 4×4 super-pixel to avoid any DMD-sensor aliasing.

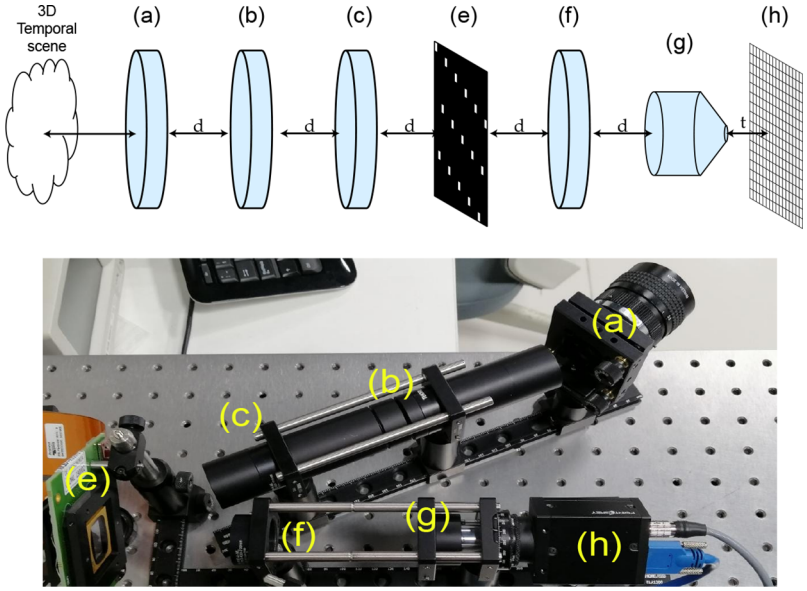

Fig. 8. Experimental setup scheme and picture of the CACTI system used for emulating the shuffled RS. (a) Objective lens; (b), c) and (f) Relay lenses; (e) DMD; (g) Objective lens. (h) Detector. $d = 100mm$.

6.1. Results

We performed a series of experimental tests capturing images using the same designed shuffled RS patterns used in the simulations, but now sequentially displayed on the DMD during the integration time of a global shutter acquisition, which finally emulates the RS acquisition. Also, we used the TRevSCI-net neural network for the video reconstructions using 32 frames, considering every frame to be sampled by a combination of 8 temporal frames of the coding datacube. For the detector, we choose a measurement ROI of 1024×1024 pixels subsampled to the matched coding resolution of 256×256. Also, we use an integration time of 52ms, while for the DMD, every code pattern (shuffled scanline) is projected every $203\mu s$, scanning the 256 scanlines within the integration time.

The measurements and the reconstructed frames are shown alongside in Fig. 9. From the videos, we can inspect the 32 frames reconstructed from every snapshot. Therefore, as the measurements were recorded at an equivalent frame rate of 17fps, the presented videos are equivalent to being captured at 544fps. Nevertheless, if the scene is well illuminated, then we can reduce even further the acquisition time and thus the fps of the reconstructed videos, although we are not sure if this would be compatible with the actual scanline speeds found in current RS CMOS sensors. From the measured snapshots in Fig. 9, we can notice that wherever there is motion, there is a sort of a coded blur. This is where the motion information resides and can be recovered during the reconstruction. We can notice the oscillatory motion of a pendulum, the rotation of a spinning wheel, an opening hand, the drop of dices, the details of a person speaking, and cars crashing. It is also noticeable that–for the static parts in the videos–there is a loss in

Optics EXPRESS

spatial resolution that is due to the subsampling of the feature extraction part of the deep neural network, which on the other hand helps the network with an improved generalization and a better memory usage.

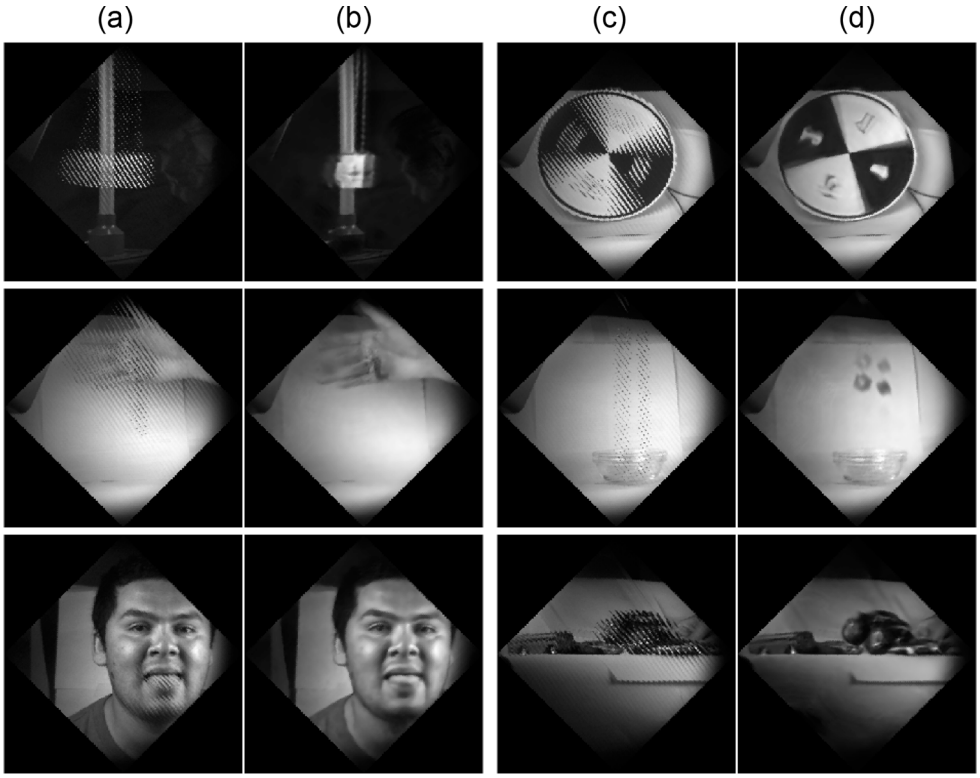

Fig. 9. Experimental results of diverse real scenes (see Visualization 3). Every pair of images shows: (a,c) the shuffled RS measurement; (b,d) the reconstructed video frame with the TRevSCI-net. Note that images are rotated by 45° to compensate for the DMD/detector rotation.

Finally, to demonstrate even further the functionality of the proposed idea, we simultaneously recorded with a high-speed camera the motion captured by the experimental shuffled RS scheme. For that, we placed an HB-500SM camera (Emergent Vision) that has a Sony IMX426 detector with 800×616 pixels and is capable of operating at up to 1594fps. In this case, the frame rate was set to 1000fps and then adjusted to match the 544 fps from the reconstructed videos. The results from these experiments can be seen in Fig. 10, where we observe a miniature car crash, the motion of a splash cymbal, and a falling balloon filled with water. Since we took a sequence of shuffled RS measurements, we can recover an extended video sequence, although we are not able to replicate a fully continuous sequence such as in a normal RS sensor given to the DMD/detector synchronization needed in our experiment. From the three dynamic scenes, we can observe one of the captured measurements, a sample of the reconstructed frame, and one of the relatively equivalent (with a distinct perspective) frames from the high-speed camera. From the associated video in Fig. 10, we can observe a significant correlation between the real motion recorded in the high-speed camera and the reconstructed videos from the shuffled RS measurements, despite the lags between every reconstructed short scene.

Optics EXPRESS

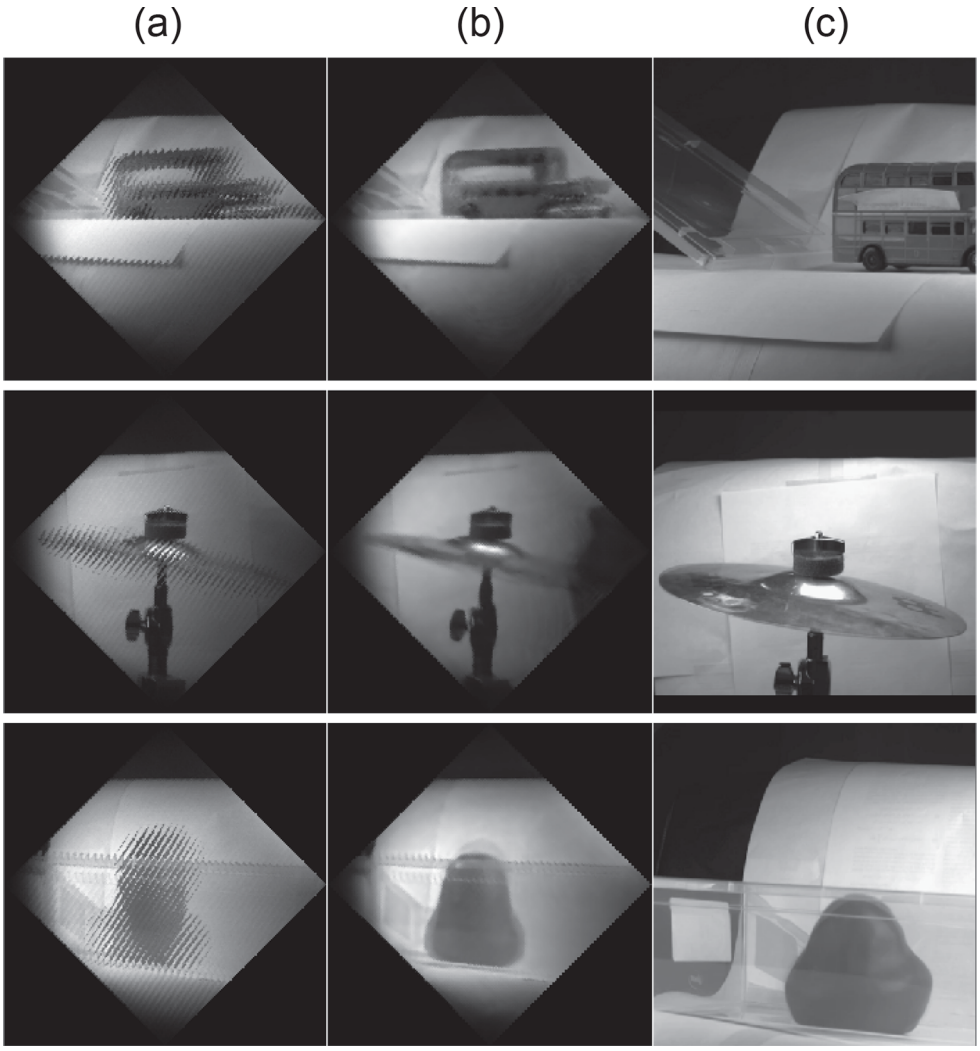

Fig. 10. Experimental results for diverse real scenes (see Visualization 4). Every set of images shows: (a) the shuffled RS measurement; (b) the reconstructed video frame with the TRevSCI-net; (c) the high-speed video reference. Images are rotated by 45° to compensate for the DMD/detector rotation.

7. Conclusion

We proposed a modification to the RS scanline mechanism found in CMOS detectors to better sample the space-time datacube, enabling snapshot temporal imaging. The main idea resides in shuffling the pixels in the scanline, allowing to sample spatial pixels from different rows. We also presented a design methodology for optimal sampling schemes and compared them to simple random shuffling. If constraining the exposure time to finish before acquiring the next scanline, we can see the shuffled RS as a particular sampling of the space-time datacube tensor. Therefore, in contrast with compressive temporal imaging schemes that also code and multiplex the temporal dimension, we can treat the reconstruction problem as a tensor completion problem. We implemented three reconstruction methods based on 3D interpolation, a state-of-the-art

Optics EXPRESS

tensor completion algorithm for videos, and a deep neural network used for snapshot compressive imaging.

From our simulated results over a dataset of 38 videos, we demonstrated that all reconstruction methods are able to recover videos that show the motion that happened in the scenes while keeping most of the static parts untouched. Nonetheless, the neural network approach, which was specially adapted to the shuffled RS coding scheme, surpasses by far the reconstruction quality of the other two methods. Although the designed shuffled RS only shows a marginal gain over the random shuffling RS approach in terms of performance, we can notice from the visual inspection that the reconstructed videos are smoother.

We implemented an experimental test-bed based on the CACTI scheme that uses a DMD to dynamically code the coded apertures emulation the RS functioning. However, instead of using random codes that multiplex the temporal dimension, we constrained the coded apertures to accommodate the demanded RS restrictions, allowing a pixel to sample a single spatial pixel at a given time frame only once. From the experimental results, we validated the ability of the shuffled RS to capture images that allow reconstructing videos that would otherwise be impossible if using the traditional RS mechanism.

In conclusion, the proposed hardware modification to the RS mechanism, if implemented, can dramatically improve the sampling diversity of the space-time datacube in contrast with the traditional RS, allowing to recover high-speed videos from a single image using either tensor completion methods or sophisticated neural networks. Moreover, we noticed that the proposed methodology for the shuffled patterns leads to a regular sampling lattice within the datacube, similarly to how many video compression methods work [23], and at the end, the reconstructed videos confirmed that the results were visually superior. At a first glance, this can become counter-intuitive for people working in compressed sensing, in particular for compressive temporal video, where random realizations are often preferred. Nonetheless, the problems are slightly different since the proposed system is not coding and multiplexing the temporal domain, so no temporal deconvolution is needed, and that is why tensor completion or interpolation also work. However, if we increase the exposure time, we start observing temporal multiplexing, although not coded since it merges contiguous frames from the datacube.

Finally, if ever implemented in hardware, the shuffled RS can be seen as an alternative for snapshot temporal imaging that do not require optical calibration nor the use of any sophisticated computational imaging hardware, although it would never perform as well as the CACTI given the sparsity of the sampling imposed by the RS mechanism. Future work will consider the reconstruction issues when dealing with arbitrary exposure times and how to implement it on CMOS hardware. For instance, longer exposure times will bring temporal multiplexing that will turn the problem into a compressive temporal imaging problem–most likely harder to disambiguate–while the advantages from the shuffled RS sampling may start to vanish. On the other hand, we believe that the proposed hardware modification should be feasible to be implemented in CMOS since even more complicated modifications were firstly demonstrated for computational photography applications in [24], more than ten years ago.

Funding. Air Force Office of Scientific Research (FA9550-19-1-0293); Fondo Nacional de Desarrollo Científico y Tecnológico (1181943).

Acknowledgements. We are grateful to Mr. Bastian Romero, Research and Development Engineer at our group (Optolab) for assisting with capturing the high-speed scenes using a high-speed CMOS camera.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Supplemental document. See Supplement 1 for supporting content.

Optics EXPRESS

References

1. B. S. Carlson, "Comparison of modern ccd and cmos image sensor technologies and systems for low resolution imaging," IEEE Sensors **1**, 171–176 (2002).
2. C. Geyer, M. Meingast, and S. Sastry, "Geometric models of rolling-shutter cameras," 6th OmniVis WS **1**, 4 (2005).
3. S. Baker, E. Bennett, S. B. Kang, and R. Szeliski, "Removing rolling shutter wobble," in *IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2010), pp. 2392–2399.
4. M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *IEEE international conference on computational photography*, (IEEE, 2012), pp. 1–8.
5. L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *IEEE Conference on Computer Vision and Pattern Recognition*, (2013), pp. 1360–1367.
6. P. Purkait, C. Zach, and A. Leonardis, "Rolling shutter correction in manhattan world," in *IEEE International Conference on Computer Vision*, (2017), pp. 882–890.
7. B. Zhuang, Q.-H. Tran, P. Ji, L.-F. Cheong, and M. Chandraker, "Learning structure-and-motion-aware rolling shutter correction," in *IEEE Conference on Computer Vision and Pattern Recognition*, (2019), pp. 4551–4560.
8. N. Antipa, P. Oare, E. Bostan, R. Ng, and L. Waller, "Video from stills: Lensless imaging with rolling shutter," in *IEEE International Conference on Computational Photography*, (IEEE, 2019), pp. 1–8.
9. G. Weinberg and O. Katz, "100, 000 frames-per-second compressive imaging with a conventional rolling-shutter camera by random point-spread-function engineering," Opt. Express **28**(21), 30616–30625 (2020).
10. F. Guzmán, P. Meza, and E. Vera, "Compressive temporal imaging using a rolling shutter camera array," Opt. Express **29**(9), 12787–12800 (2021).
11. J. Kepler, *The six-cornered snowflake* (Paul Dry Books, 2010).
12. M. Goldberg, "On the densest packing of equal spheres in a cube," Mathematics Magazine **44**(4), 199–208 (1971).
13. L. Allison, C. Yee, and M. McGaughey, *Three-dimensional Queens Problems* (Monash University, Department of Computer Science, 1989).
14. P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady, "Coded aperture compressive temporal imaging," Opt. Express **21**(9), 10526–10545 (2013).
15. M. Qiao, X. Liu, and X. Yuan, "Snapshot spatial–temporal compressive imaging," Opt. Lett. **45**(7), 1659–1662 (2020).
16. A. Matin and X. Wang, "Compressive coded rotating mirror camera for high-speed imaging," Photonics **8**(2), 34 (2021).
17. X. Liu, J. Liu, C. Jiang, F. Vetrone, and J. Liang, "Single-shot compressed optical-streaking ultra-high-speed photography," Opt. Lett. **44**(6), 1387–1390 (2019).
18. I. Amidror, "Scattered data interpolation methods for electronic imaging systems: a survey," Journal of Electronic Imaging **11**(2), 157–176 (2002).
19. X. Li, Y. Ye, and X. Xu, "Low-rank tensor completion with total variation for visual data inpainting," in *Thirty-First AAAI Conference on Artificial Intelligence*, (AAAI Press, 2017), AAAI'17, p. 2210–2216.
20. Z. Cheng, B. Chen, G. Liu, H. Zhang, R. Lu, Z. Wang, and X. Yuan, "Memory-efficient network for large-scale video compressive sensing," in *IEEE Conference on Computer Vision and Pattern Recognition*, (2021), pp. 16246–16255.
21. T. C. Hales, J. Harrison, S. McLaughlin, T. Nipkow, S. Obua, and R. Zumkeller, "A revision of the proof of the kepler conjecture," in *The Kepler Conjecture*, (Springer, 2011), pp. 341–376.
22. T. Hales, M. Adams, G. Bauer, D. Dang, J. Harrison, T. Hoang, C. Kaliszyk, V. Magron, S. McLaughlin, T. Nguyen, T. Nguyen, T. Nipkow, S. Obua, J. Pleso, J. Rute, A. Solovyev, A. Ta, T. Trung, D. Trieu, and R. Zumkeller, "A formal proof of the kepler conjecture," in *Forum of Mathematics, Pi*, vol. 5 (Cambridge University, 2017).
23. A. M. Tekalp, *Digital video processing* (Prentice Hall Press, 2015).
24. J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar, "Coded rolling shutter photography: Flexible space-time sampling," in *IEEE International Conference on Computational Photography*, (IEEE, 2010), pp. 1–8.