

Compressive Imaging using RIP-compliant CMOS Imager Architecture and Landweber Reconstruction

Marco Trevisi, Ali Akbari, Maria Trocan, Ángel Rodríguez-Vázquez, *Fellow, IEEE*, and Ricardo Carmona-Galán, *Senior Member, IEEE*

Abstract— In this paper we present a new image sensor architecture for fast and accurate compressive sensing (CS) of natural images. Measurement matrices usually employed in compressive sensing CMOS image sensors (CS-CIS) are recursive pseudo-random binary matrices. We have proved that the restricted isometry property (RIP) of these matrices is limited by a low sparsity constant. The quality of these matrices is also affected by the non-idealities of pseudo-random numbers generators (PRNG). To overcome this limitation, we propose a hardware-friendly pseudo-random ternary measurement matrix generated on-chip by means of a class III elementary cellular automaton (ECA). It presents a chaotic behaviour that emulates random CS measurement matrices better than other PRNG. We have combined this new architecture with a block-based CS smoothed-projected Landweber (BCS-SPL) reconstruction algorithm. By means of single value decomposition (SVD) we have adapted this algorithm to perform fast and precise reconstruction while operating with binary and ternary matrices. Simulations are provided to qualify the approach.

Index Terms—CMOS image sensor architecture, compressive sensing, Landweber reconstruction, power spectral density, random binary matrix RIP proof, single value decomposition, ternary measurement matrix.

I. INTRODUCTION

COMPRESSIVE Sensing (CS) is a data acquisition technique that can be used to represent the content of an image with fewer samples than required by Shannon-Nyquist theorem. These samples are called compressed samples. Reconstruction algorithms exploit the sparsity of the original image in order to recover it by finding the unique solution to an underdetermined linear system [1]. This system can be expressed in matrix notation as:

$$\mathbf{y} = \Phi \mathbf{x} \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^M$ is the set of M compressed samples, $\mathbf{x} \in \mathbb{R}^N$ represents the N pixels values folded in a column vector, i.e. the unknowns of the system of equations, and $\Phi \in \mathbb{R}^{M \times N}$

($M \ll N$) is referred to as measurement matrix. It has been shown in [2] that, if Φ holds the Restricted Isometry Property (RIP), then the solution to this inverse linear problem, by means of convex optimization, is unique and error free [3].

CS is a field of research that is located in-between the areas of analog microelectronics and of signal processing. The link between these areas is given by the feasible yet RIP-compliant design of measurement matrices. Measurement matrices affect both the performance of reconstruction algorithms as well as the architecture of CS CMOS image sensors (CS-CIS).

CS differs from standard acquisition-plus-compression techniques in that the image is not captured and then digitized before compression. To benefit from the characteristics of CS, the image needs to be sensed and compressed simultaneously. For that reason, in CS-CIS, compressed samples are formed in the analog domain and digital conversion occurs just before their delivery.

If the coefficients of a measurement matrix (Φ) were real numbers, since each pixel has different coefficients for each compressed sample, a CS-CIS would require the implementation of in-pixel analog multipliers. This solution would be impractical in terms of pixel sensitivity and spatial resolution. Furthermore, each compressed sample is a linear combination of the weighted readings of all pixels (N), which means it is the sum of N products. The upper bound on the amount of bits required to represent these linear combinations is:

$$B_{CS} = \lceil \log_2 N \rceil + (B_\Phi + B_1) \quad (2)$$

where $\lceil \cdot \rceil$ denotes the smallest integer greater than the argument and B_1 and B_Φ are the number of bits used to describe the pixel values and the coefficients of the measurement matrix, respectively.

It is virtually impossible to design analog to digital converters (ADCs) with such resolutions in standard technologies. In fact, one of the main drawbacks of most CS-CIS implementations [4][5][6] is the lack of dynamic range to properly represent compressed samples.

This paper has been submitted for review on the 18th of Jun 2018.

The First and Second Authors contributed equally to this work.

This work has been funded by the Spanish Government through projects TEC2015-66878-C3-1-R MINECO (European Region Development Fund, ERDF/FEDER), by Junta de Andalucía through project TIC 2338-2013 CEICE and by the Office of Naval Research (USA) through grant N000141410355

Marco Trevisi, Ángel Rodríguez-Vázquez and Ricardo Carmona-Galán are with the Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC-Universidad de Sevilla, Spain (e-mail: {trevisi, angel, rcarmona}@imse-cnm.csic.es)

Ali Akbari and Maria Trocan are with Institut Supérieur d'Electronique de Paris (ISEP), Paris, France (e-mail: {ali.akbari, maria.trocan}@isep.fr).

The use of binary measurement matrices and block based compressed sampling (BCS) [7], which divides a pixel array into smaller sub-arrays that are digitized independently, is essential for the implementation of practical measurement matrices in CS-CISs.

In signal processing literature, there are many works that study deterministic binary measurement matrices; i.e. cyclic matrices [8], sparse bipartite graphs [9] or other forms of deterministic matrices that try to match the performance of Gaussian random matrices [10]. These matrices are usually coined hardware-friendly by their authors, but, even if they are binary, given the technology at our disposal, they cannot be implemented in application specific integrated circuits (ASICs). Being compressive sampling an acquisition technique, the fact that we cannot apply these theories to actively design sensors is, in itself, the limitation of these works. This happens because even if they are binary in nature they still disregards the fact that, on-chip, measurement matrices need hardware resources to be generated, transmitted and/or stored. To store a binary measurement matrix in its integrity a large memory would be required. This exceeds nowadays CS-CIS capabilities in term of area occupation. If, on the contrary, we were to introduce a connection between sensor and reconstruction system to continuously send the rows of the matrix, the amount of exchanged data would void CS benefits as a compression technique. For this reason, measurement matrices designed in analog microelectronics [5][6] solely focus on the use of pseudo random binary numbers generators (PRNG) to recursively create the rows of the matrix sample after sample. This solution employs the minimum possible amount of on-chip resources and does not need data feedback between sensor and reconstruction system. These matrices though do not ascertain the quality of the resulting samples because their primary focus is placed almost entirely on electrical parameters such as power consumption and compactness of the implementation.

PRNG are non-linear spatially discrete and temporally discrete dynamic systems made of binary logic elements that show high sensitivity to initial conditions and evolve in time according to a divergent and fractal behaviour [11]. A new state in their time evolution is derived from their actual state using a feedback mechanism that promotes this instability. Each state corresponds to a row of the binary measurement matrix. Each row of a measurement matrix is generated by the CS-CIS from the previous row and there is no need to store the whole binary matrix on chip or to receive it from outside the chip.

In this paper, the performance of binary measurement matrices obtained using binary PRNG is evaluated in terms of RIP. We introduce a new method for the generation of a hardware-friendly measurement matrix that takes into account both technology limits and quality of the resulting matrix. A differential pixel readout system and an elementary cellular automaton (ECA) are proposed to recursively create a ternary measurement matrix in a row by row fashion. The performance of the generated ternary measurement matrix is analysed and compared with other binary measurement matrices typically found in analog microelectronics. To do so we introduce a new

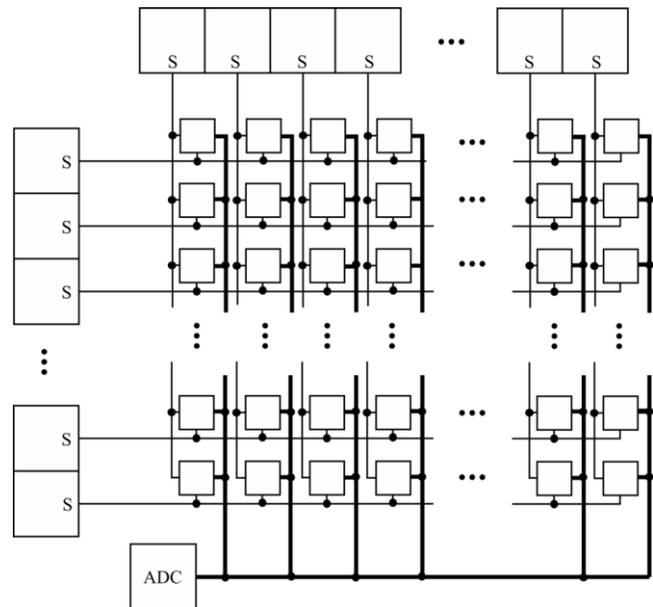


Fig. 1. Conceptual floor plan of a CS-CIS.

reconstruction algorithm that uses this matrix to deliver reconstruction results close to those of the more refined matrices typically found in signal processing.

II. RIP OF BINARY MATRICES PRODUCED BY PSEUDO RANDOM BINARY GENERATORS

PRNG commonly used to implement measurement matrices into CS-CIS are Linear Feedback Shift Registers (LFSRs) [12][13] LFSRs consist of a series of sequentially connected flip-flops. Some of the outputs of these flip-flops, besides being connected to the input of the flip-flop that they precede, are also connected, by means of XOR logic gates, to the input of the very first one. When considered individually, the output of each flip-flop in this sequence evolves in time with a behaviour that resembles a symmetric probability distribution. Using one flip-flop for each pixel it is possible to sequentially create the pseudo-random binary coefficients of the measurement matrix one row after the other. The resulting matrix is an approximation of a very efficient type of measurement matrix popular in CS: a matrix in which each element is derived from the normalized Gaussian distribution. Early implementations of CS-CIS binary measurement matrices were carried out incorporating a flip-flop inside each pixel. These flip-flops were then connected sequentially to form LFSRs. This solution though posed a heavy burden on the design specifications of image sensors due to the large area required at pixel level. Recent solutions have overcome this problem placing LFSRs around the pixel array as row and column drivers (Fig. 1) [6][14][15].

Inside of each pixel the flip-flops are replaced by simple and more compact selectors. These selectors are driven by LFSRs surrounding the array. Selectors can be implemented with few transistors using for instance switches or gated inverters. This simplification was possible because, at pixel level, we can consider random row and column selection as the multiplication of two independent binary random variables. The multiplication

of random variables still delivers random outputs. Using PRNG, such as LFSRs, the recursive row and column drivers also reduce the amount of on-chip memory needed to store their elements to $N_g = 2\sqrt{N}$ bits. Since this recursive pseudo-random sequence can be reproduced from an initial N_g bits seed, it allows the independent generation of the same measurement matrix in the system in charge of reconstruction, without continuous transmission of the measurement matrix from the sensor. It is important to notice that LFSRs produce binary matrices that are neither normalized nor orthogonalized. To understand the quality of the compressed samples that they produce, it is necessary to analyse them in terms of RIP.

Suppose that an area of size $\sqrt{N} \times \sqrt{N}$ pixels is sampled by a measurement matrix generated row by row as described above. Let us consider the simplest possible case in which an element of the pseudo random generator is assigned to each pixel of the imager so that each row of the measurement matrix Φ corresponds to a time step in the discrete evolution of the generator itself. Mathematically, this sampling process is formulated as Eq.(1), where $\mathbf{x} \in \mathbb{R}^N$ is the $\sqrt{N} \times \sqrt{N}$ array of pixels folded in vector form. The reconstruction of the original signal \mathbf{x} from the compressed samples \mathbf{y} is an ill-posed problem. By making use of sparsity, a well-known characteristic existing in natural signals, a k -sparse¹ signal, like images and videos, can be reconstructed from a few measurements by using an appropriate measurement matrix.

A sufficient condition for the unique and exact recovery of the signal is the RIP of the measurement matrix Φ [2]. A matrix $\Phi \in \mathbb{R}^{M \times N}$ satisfies the RIP of order k if there is constant δ_k ($0 < \delta_k < 1$) such that, for all vectors $\mathbf{x} \in \mathbb{R}^N$ with $\|\mathbf{x}\|_0 \leq k$ (i.e. k -sparse signals²), it holds:

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\Phi\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2 \quad (3)$$

The smallest non-negative value for δ_k is called restricted isometry constant (RIC) of order k . The construction of a measurement matrix which satisfies the RIP is a central problem in CS. If sparse signals with maximum possible sparsity level k can be recovered exactly and stably, it is said the measurement matrix Φ has sparsity order k . An upper bound for the sparsity level is $k \leq CM/\log(N/M)$ where C is a constant [16]. The signals with k above this bound can only be approximately reconstructed. In practice, to recover a signal \mathbf{x} with a large k , is desirable to have a measurement matrix with a δ_k as small as possible. The limit imposed on the RIP of random binary measurement matrices has already been studied in [17] Such binary matrices satisfy the RIP with:

$$\delta_k = \frac{(3k-2)m}{(k-2)m+4d} \quad \text{if } 3 \leq d \leq M/2 \text{ and } 2 \leq m \leq d \quad (4)$$

where d is the estimated number of non-zero elements in each column of Φ and m is the maximum inner product between two distinct columns. The distribution of 1's among the elements of random binary matrices, generated by pseudo random generators, can be associated with a Bernoulli probability distribution in which each element has a probability P of being 1 and $\bar{P} = 1 - P$ of being 0. Since these pseudo random

measurement matrices try to emulate the normalized Gaussian distribution, which is a symmetric probability distribution, usually, circuit designers choose LFSR because their distribution is akin to $P = \bar{P} = 0.5$.

Let us consider the ideal case of an ideal pseudo random generator capable of recursively producing independent binary elements with a symmetric Bernoulli probability distribution having $P = 0.5$. If we consider a large enough pseudo random binary generator (a large enough number of pixels N), since each of its elements, by construction, is independent from the others and the probability distribution is symmetrical, we will obtain a measurement matrix Φ with an equal amount of 0s and of 1s row-wise. If the acquisition process is long enough (a large enough number of acquired compressed samples M), we will have an equal amount of 0s and of 1s column-wise as well. Such measurement matrix, whose entries are randomly drawn from pseudo random generators, satisfies the RIP with Eq.(4). Following [17], let us define the mutual coherence of Φ as:

$$\mu(\Phi) = m/d \quad (5)$$

We can derive m and d as functions of the number of pixels in the image N , the sampling substrate of the sensor $S = M/N$, and the probability of each element of being different from 0, P . From the definition of substrate, it is straight forward that $M = S \times N$. Moreover, using the probability that each element of the matrix has of being different from 0, we can define the number of non-zero elements in a column of Φ as:

$$d = S \cdot N \cdot P \quad (6)$$

Mutual coherence $\mu(\Phi)$ of a matrix represents the maximum absolute value of the cross-correlations among its normalized columns and is defined as:

$$\mu(\Phi) = \max_{j \neq k} |\boldsymbol{\varphi}_j^H \boldsymbol{\varphi}_k| \quad (7)$$

where $\boldsymbol{\varphi}_k$ is the k -th normalized column of matrix Φ and $\boldsymbol{\varphi}_j^H$ is the conjugate transpose of the j -th normalized column of the same matrix. For a matrix to be column-wise normalized it means that for each column: $\boldsymbol{\varphi}_j^H \boldsymbol{\varphi}_j = 1$. Given the definition of mutual coherence and the randomness of $\boldsymbol{\varphi}_j^H$ and $\boldsymbol{\varphi}_k$, to approximate m , we will make use of the most probable outcome of a dot product between normalized columns of Φ :

$$m = E[\boldsymbol{\varphi}_j^H \boldsymbol{\varphi}_k]d \quad (8)$$

Remember that $\boldsymbol{\varphi}_j^H \boldsymbol{\varphi}_k$ is the sum of M element by element products. Each one of these M products will be different from zero if and only if the two elements that are being multiplied differ from 0 as well. By construction, we are multiplying independent random variables; the probability that their product be different from 0 is equal to the joint probability of the single elements. For this reason, the expected amount of non-zeros among the M products is:

$$E[\text{Tr}(\boldsymbol{\varphi}_j^H \cdot \boldsymbol{\varphi}_k)] = P^2 \cdot S \cdot N \quad (9)$$

where $\text{Tr}(\cdot)$ is the trace of the matrix obtained by the vector multiplication of $\boldsymbol{\varphi}_j^H$ and $\boldsymbol{\varphi}_k$. Furthermore we can derive the value of each product in terms of P , N and S by using the definition of normalization and remembering that Φ is binary.

¹ \mathbf{x} is called k -sparse or a signal with sparsity level k , if it has no more than k non-zero components with $k \ll N$.

² $\|\mathbf{x}\|_0$ norm denotes the number of non-zero components in \mathbf{x} .

Normalizing a vector involves dividing each one of its non-zero elements by the Euclidean norm of the vector itself. Since each non-zero element in a binary vector has to take the value of 1, all elements of a normalized binary vector will be equal to the inverse of the vector Euclidean norm itself. For this reason, if the i -th elements in column $\boldsymbol{\varphi}_j$ and column $\boldsymbol{\varphi}_k$ are non-zeros, than their normalized product will be:

$$\boldsymbol{\varphi}_j^H \boldsymbol{\varphi}_k = \frac{1}{P \cdot N \cdot S} \quad (10)$$

Combining Eq.(9) and Eq.(10), we can deduce the most probable cross-correlation between two columns of a binary matrix $\boldsymbol{\Phi}$. This correlation only depends on P and it is independent from the number of pixels and the sampling substrate:

$$E[\boldsymbol{\varphi}_j^H \boldsymbol{\varphi}_k] = \frac{P^2 \cdot S \cdot N}{P \cdot N \cdot S} = P \quad (11)$$

Joining Eq.(4), Eq.(5) and Eq. (11) we obtain:

$$\delta_k = \frac{(3k-2)}{(k-2)+4/P} \text{ if } 3/(S \cdot N) \leq P \leq 1/2 \text{ and } 2/(S \cdot N) \leq P^2 \leq P \quad (12)$$

For $P = 0.5$ and $N \geq 6$, it is possible to see that all of the conditions hold and that random binary matrices that follow a symmetric probability distribution will have RIP ($\delta_k \leq 1$) only if $k \leq 4$.

A recovery algorithm can deliver error free reconstructions only if the measurement matrix holds RIP. Eq. (12) poses a harsh limitation on matrices that need to be implemented in CS-CISs. Block based CS-CIS, in many occasions, need small blocks in order to maintain acceptable errors during ADC. ADC dynamic range is determined by noise and the maximum signal range. It is very difficult to allocate large dynamic range. Small blocks reduce the requirement for signal range and consequently making it feasible for implementation. This in turn introduces asymmetries in the pixel array. These asymmetries create artefacts in the captured images in correspondence of the borders of the blocks. Reconstruction of block based compressive sampled images usually entails smoothing of those regions. When using a CS-CIS the smoothing process needs to take care of physical discontinuities as well. Because these two effects combine themselves they can potentially deteriorate the quality of the recovered images. It would be desirable to implement a measurement matrix that did not add errors during reconstruction to facilitate the smoothing process. But, since the RIP of random binary matrices holds only if $k \leq 4$, errors during reconstruction are to be expected.

III. TERNARY MATRICES GENERATED USING CLASS III ELEMENTARY CELLULAR AUTOMATA

We have bound the RIP of binary measurement matrices usually implemented in a CS-CIS with a limit ($k \leq 4$) that, in practice, is too small to grant the error free reconstruction of natural images. As explained in the introduction, PRNG generated measurement matrices are necessary to design feasible CS-CIS architectures. However, to overcome the limitation imposed by Eq.(12), we propose to use them to generate ternary measurement matrices instead of the commonly used binary ones.

The resulting ternary matrices present smaller coherence than the aforementioned binary ones of same size thus

improving RIP and diminishing reconstruction errors. Similar results have been obtained using binary matrices with binary correlations between columns applied to CS in [18].

Ternary measurement matrices can be generated by delivering two independent binary coefficients to each pixel: one to define the number either 0 or 1 and the other to define the sign of the contribution. As mentioned in section II, if one set of row/column selectors (or driving signal) can be used to produce a binary matrix having one bit per element we could use two sets (or driving signals) to produce a ternary matrix that needs two bits per element. If each pixel received two driving signals, it would be possible to use one of them to determine if the pixel took part in a compressed sample and, in case it did, use the other to select the sign of its contribution. Since recent CS-CIS examples [6][14][15] have shown that as few as $N_g = 2\sqrt{N}$ bits of information are needed to generate binary measurement matrices on-chip, the amount of resources (number of transistors) needed to incorporate a ternary matrix would still be implementable in a CS-CIS design. Each coefficient of the resulting matrix would have a probability distribution of:

$$\begin{cases} P_{+1} = 0.25 \\ P_0 = 0.5 \\ P_{-1} = 0.25 \end{cases} \quad (13)$$

where P_{-1} represents the probability that a pixel has of contributing negatively in a compress sample, P_{+1} represents the probability that a pixel has of contributing positively and P_0 represents the probability that a pixel has of not contributing at all. Furthermore, a ternary measurement matrix would have a positive effect on the limit, imposed by Eq. (2), on the amount of bits required to represent a compressed sample. In hardware, the sign associated to a pixel contribution can be implemented by means of a differential readout system that routes the output of the pixels through one of two output lines [5]. These lines, outside of the pixel array can then be digitized separately or used as input of differential circuits, such as analog subtractors or transimpedance amplifiers. Using a differential readout system to divide the pixel contribution to different output lines to be treated separately would in fact reduce the amount of bits needed for ADC thus relaxing the converter design parameters.

Another important factor to study in order to improve the quality of CS-CISs measurement matrices are LFSRs performances. Even if LFSRs are good candidates to emulate random probability distributions thanks to their low impact on on-chip area consumption, ultimately the matrices that they produce still follow a row-by-row deterministic pattern. For this reason it is necessary to analyse this pattern to understand how well the generated measurement matrix approximates the functionality of one that is extracted directly from a real binary probability distribution. Moreover, LFSRs are not the only example of circuits that can serve as PRNG in a CS-CIS. A valid alternative are elementary cellular automata (ECA) [19]. An ECA is a spatially and temporally discrete dynamic system made of identical interconnected cells. Each ECA cell is a triple-input single-output binary unit. Based on this setup there are 256 possible ECA configurations. Each configuration, also known as rule, corresponds to a particular logic

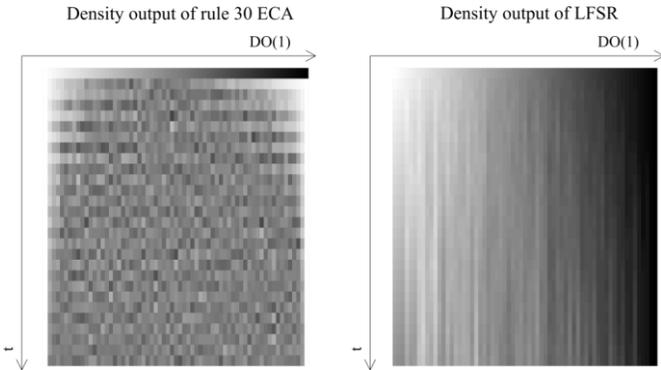


Fig. 3. Density of the output over a varying random initial seed for rule-30 ECA and LFSR.

implementation. We will focus our attention on rules that belong to class III, in particular rule 30 [20]. We will study rule 30 hardware implementation and the matrices that it produces in order to establish if it is better suited than LFSRs to be used as PRNG in the design of CS-CIS.

Let us consider the simple case of binary measurement matrices. This simplification poses no loss of generality because the elements of the aforementioned ternary matrices are generated following the same pattern used for binary measurement matrices and the bits are grouped in sets of two only at pixel level. Let us define $\Phi_G \in \{0,1\}^{M \times N}$ as a binary measurement matrix obtained using a PRNG and $\Phi_R \in \{0,1\}^{M \times N}$ as a binary measurement matrix obtained selecting each element at random from a binary probability distribution with $P = \bar{P} = 0.5$. Φ_G will be a good approximation of Φ_R if it holds three characteristics: the number of non-zero elements in a row of Φ_G must approximate $P \approx 0.5$; the temporal evolution of the elements of the PRNG do not present repeating patterns; and the temporal evolution of the elements of the PRNG present no correlation with one another. The first and second requirements bind the number of non-zero elements in Φ_G with the average of the probability distribution used to create Φ_R . The second and third requirements can be used to bind the mutual coherence of Φ_G as:

$$E[\mu(\Phi_G)] \approx E[\mu(\Phi_R)] = P \quad (14)$$

We will analyse how binary measurement matrices generated using LFSRs and ECA fulfil these criteria in order to compare the performances of the two types of PRNGs in CS imagers.

A. Density of the PRNG Output

To study the number of non-zero elements in a row of Φ_G let us define the density of a PRNG output ($DO(t)$) as the average of all the states of its binary elements at a given discrete time step t (following a notation similar to [21]):

$$DO(t) = \frac{1}{N} \sum_{h=1}^N \varphi_h(t) \text{ where } t = 1, 2, \dots, M \quad (15)$$

being $\varphi_h(t)$ the state of the h -th element of the PRNG at time t as well as the t -th element of the h -th column of Φ_G , N the number of elements of the PRNG as well as the number of columns in Φ_G and M the number of discrete time steps of the PRNG as well as the number of rows in Φ_G . Let $t = 1$ be the initial condition when the first row of Φ_G coincide with the seed that has been loaded in the memory on-chip. Since the elements of Φ_R are extracted from a symmetric probability distribution,

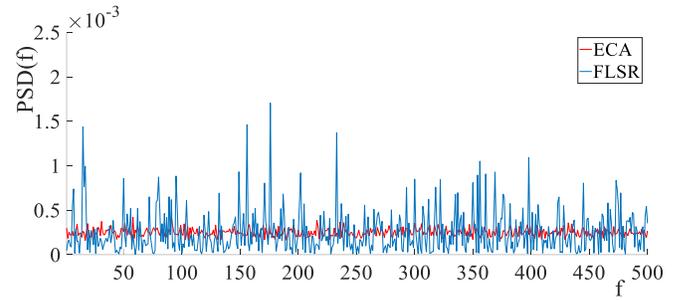


Fig. 2. Power spectral density of rule-30 ECA and LFSR.

an optimal seed should have half of its elements set to 0 and half set to 1 and as such $DO(1) = 0.5$.

To analyse the performance of LFSRs and ECA it is necessary to evaluate how fast their states can reach $DO(t) = 0.5$ in the eventuality of having a suboptimal initial seed in which the number of elements set to 0 differs from the number of those set to 1 (Fig. 3). To do so we devise a MATLAB experiment in which we evolve a 64-cells rule-30 ECA and a 64-flip-flops LFSR using 65 different initial seeds having a density that varies from $DO(1) = 0$ all the way to $DO(1) = 1$.

In the result of the experiment we represent $DO(t)$ using greyscale elements in which $DO(t) = 0$ is coloured black and $DO(t) = 1$ is coloured white. In Fig. 3 the top line represents these initial seeds with increasing density, from the leftmost $DO(1) = 0$ to the rightmost $DO(1) = 1$. Each step underneath shows the temporal evolution of the PRNG output density evolving from each initial seed configuration.

Fig. 3 shows that LFSRs density changes in time at a much slower rate than class III ECA. This implies that in the presence of sub-optimal initial seed, the DO of LFSRs would take several time steps longer than ECAs to achieve a stable situation centred around $DO(t) = 0.5$. As reported in [22], the behaviour of LFSRs is more similar to class IV ECA, where randomness in the evolution is linked to the randomness of the initial configuration, rather than class III ECA where the chaotic evolution is introduced by the rule itself. For this reason, in order to approximate Φ_R using LFSRs much care should be placed in the selection of an appropriate initial seed. ECA appears to be a safer choice to generate a Φ_G rather than LFSRs at least in terms of number of distribution of non-zero elements starting from seeds that are not ideal.

Fig. 3 also shows an interesting behaviour for ECA when its seed is either $DO(1) = 0$ or $DO(1) = 1$ and for LFSR when its seed is $DO(1) = 0$. These particular seeds represent the forbidden states of these PRNGs. All PRNGs have forbidden states. A forbidden state is a state in which the PRNG continuously delivers a fixed output. These states though do not pose any issues in real-world implementations because it is impossible for a PRNG to stumble upon them accidentally. Even though these states do not pose any problem during standard operation, to use PRNGs in CS-CIS, it is important to be aware of their presence in order to avoid sending them accidentally. The only particular concern regarding forbidden states happens when power is first applied to a circuit. Since each register can randomly start up containing either a zero or

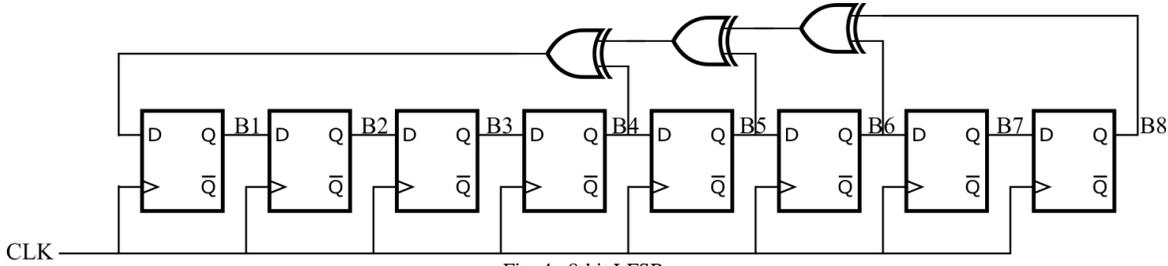


Fig. 4. 8-bit LFSR.

a one, a PRNG could power up containing its forbidden state, but this can be quickly taken care of initializing the PRNG with an opportune seed value.

B. Power Spectral Density

To understand if the rows in Φ_G present repeating patterns and to compare dynamical behaviours of LFSRs and rule-30 ECAs we will use Power Spectral Density (PSD) analysis. This technique has been extensively applied to analyse discrete dynamical systems such as ECA [21] or LFSRs [13]. For a spatially discrete and temporally discrete dynamic system, the DFT can be expressed as:

$$\phi_h(f) = \frac{1}{M} \sum_{t=1}^M \varphi_h(t) e^{-i2\pi t f / M} \text{ where } t = 1, 2, \dots, M \quad (16)$$

where, following a notation similar to [21], $\phi_h(f)$ is the DFT value of the h -th element of the PRNG at frequency f , M once again is the number of discrete time steps of the PRNG as well as the number of rows in Φ_G and $\varphi_h(t)$ is the state of the h -th element of the PRNG at time t as well as the t -th element of the h -th column of Φ_G .

PSD expresses the distribution of the energy of a waveform among its different frequency components. Any peak in a graphic of $\text{PSD}(f)$ over f at a given frequency f_p would represent a strong repeating pattern of period $1/f_p$ among the rows of Φ_G . Given all $\phi_h(f)$ PSD can be computed as:

$$\text{PSD}(f) = \frac{1}{N} \sum_{h=1}^N |\phi_h(f)|^2 \quad (17)$$

where N is the number of elements of the PRNG as well as the number of columns in Φ_G .

The PSD profile of a suitable and efficient PRNG for CS should closely resemble white noise since its energy should be equally spread throughout the entire spectrum. Once again we devise a MATLAB experiment in which we evolve a 64-cells rule-30 ECA and a 64-flip-flop LFSR. Fig. 2 shows the PSD profiles of rule-30 ECA and LFSR. These graphics have been produced for $t = 1000$ time steps starting from an initial random binary seed with $\text{DO}(1) = 0.5$.

The first thing that we can notice is that both PSD profiles are similar to white noise. That was to be expected because otherwise neither of them would be suitable for CS but, the amplitude of the pattern of the LFSR presents a greater variation than that of rule-30 ECA as well as some unwanted fluctuation.

Rule-30 ECA appears to be a better choice over LFSRs to extract quasi-independent coefficients from a random like distribution. Furthermore, as we said in the introduction, a necessary requirement to design a measurement matrix for CS is that said matrix respects the RIP which means that it must be

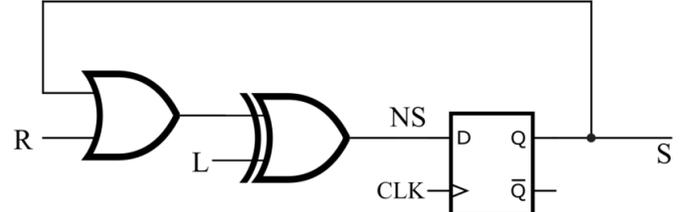


Fig. 5. ECA cell implementation of rule 30.

nearly orthonormal, at least when operating on sparse vectors [2]. Orthogonal matrices have uncorrelated columns and rows. For this reason, even though RIP and mutual coherence are two different concepts, a matrix Φ_G , compliant with RIP, and its transpose Φ_G^T will both have low mutual coherence by definition [23]. Class III ECA PSD profile offers better guarantees over LFSRs that the mutual coherence of the generated matrix will be low. We opted to use a single class III rule elementary cellular automaton over other solutions like multi-rule hybrid cellular automata for two reasons: to facilitate the circuit design and because we are interested in the patterns that the individual cells produce and not the generated number in its decimal representation.

C. Hardware Implementation

To analyse the correlations among temporal evolutions of the elements of a PRNG is equivalent to analyse mutual coherence of Φ_G^T . To do so it is necessary to study how these patterns are generated by the feedback mechanism that promotes the PRNG's instability.

Fig. 4 is an example that represents the feedback needed to evolve an 8-bit LFSR. This feedback is imposed on the first flip-flop of the sequence through XOR gates tapping flip-flops 4, 5, 6 and 8. The flip-flops that must be tapped to allow an N -bit LFSR to cycle through all $2^N - 1$ combinations are tied to primitive polynomials with binary coefficients and degree equal to N [24].

On the contrary, the feedbacks needed for the evolution of a cell in a cellular automaton are spatially local. The block diagram of Fig. 5 is an example of a cell of a rule-30 ECA, in it we have defined as S the actual state of the cell. L and R are the actual states of its two closest neighbours, to the left and to the right respectively. NS corresponds to the cell's next state.

This is in sharp contrast with the feedback of LFSRs which is unique for all its elements. The only drawback that ECA present with respect to LFSRs is that while the basic element of a LFSR is a flip-flop, to implement a cell of a rule-30 ECA it takes two extra logic gates.

Even though ECA require more resources (transistors) to be

Table 1: **BCS-SVD Algorithm**

Input: $\Phi, [y_i]_{i=1}^H, \Psi(\cdot)$
Output: $\hat{\mathbf{x}}$
Initialize: $k = 0$ and $\hat{\mathbf{x}}^{[0]} = 0$
SVD decomposition: $\Phi = \mathbf{U}\mathbf{D}\mathbf{V}^T$
While $\|\hat{\mathbf{x}}^{[k+1]} - \hat{\mathbf{x}}^{[k]}\| \leq \varepsilon$:
 $[\hat{\mathbf{x}}_i^{[k]}]_{i=1}^H = \mathbf{R}(\hat{\mathbf{x}}^{[k]})$
 For $i = 1$ **to** H
 $\hat{\mathbf{x}}_i^{[k+1]} = \hat{\mathbf{x}}_i^{[k]} + \Phi^T \mathbf{U} \mathbf{S} \mathbf{U}^T (y_i - \Phi \hat{\mathbf{x}}_i^{[k]})$
 End For
 $\hat{\mathbf{x}}^{[k+1]} = \mathbf{R}^{-1}([\hat{\mathbf{x}}_i^{[k+1]}]_{i=1}^H)$
 $\hat{\mathbf{x}}^{[k+1]} = \mathbf{Wiener}(\hat{\mathbf{x}}^{[k+1]})$
 $\hat{\mathbf{x}}^{[k+1]} = \Psi(\hat{\mathbf{x}}^{[k+1]})$
 $\hat{\mathbf{x}}^{[k+1]} = \mathbf{Threshold}(\hat{\mathbf{x}}^{[k+1]})$
 $\hat{\mathbf{x}}^{[k+1]} = \mathbf{R}^{-1}(\hat{\mathbf{x}}^{[k+1]})$
End While

implemented than LFSRs it has been shown in [22] that, due to the shifting of data, inside large LFSRs, the outputs of the individual flip-flops have strong correlation between each other. A consequence of this fact is that the mutual coherence of Φ_G could inadvertently increase, thus reducing k , the maximum amount of non-zero coefficients, in an appropriate domain, that the signal to be sampled can have before incurring in reconstruction errors.

IV. RECOVERY ALGORITHM FOR PRNG GENERATED MEASUREMENT MATRICES

When the signal to be acquired is a large image, the receiver needs a large memory to store the generated measurement matrix. Furthermore, the reconstruction algorithm undergoes a huge computational burden due to this large size. BCS [7][25][26] solves these issues by dividing the scene to be acquired into H non-overlapping blocks of size $\sqrt{N_b} \times \sqrt{N_b}$. With BCS, the acquisition is achieved independently, block by block, using the same measurement matrix $\Phi \in \mathbb{R}^{M_b \times N_b}$. For each block we can reformulate Eq. (1) as:

$$\mathbf{y}_i = \Phi \mathbf{x}_i \quad (18)$$

where $\mathbf{x}_i \in \mathbb{R}^{N_b}$ is the vector representation of the i -th block of the image and $\mathbf{y}_i \in \mathbb{R}^{M_b}$ are the corresponding samples. At the receiver side, each block can be reconstructed separately.

In order to avoid creating unpleasant blocking artefacts, recently, several BCS reconstruction algorithms have been designed [25][27][28][29]. BCS with smoothed-projected Landweber reconstruction (BCS-SPL) is a fast and efficient reconstruction algorithm [25]. It is based on successive projections and thresholding in the transform domain and provides a good trade-off between computational complexity and reconstruction quality. While this method is simple to derive, convergence is only guaranteed for a limited class of matrices that hold the strong RIP condition, e.g. Gaussian random measurement matrices with real coefficients. Recently, authors in [26] proposed a way to improve the performance of the BCS-SPL algorithm to a wide range of random measurement matrices, including Bernoulli and Gaussian measurement matrices.

Binary PRNG such as those used in hardware

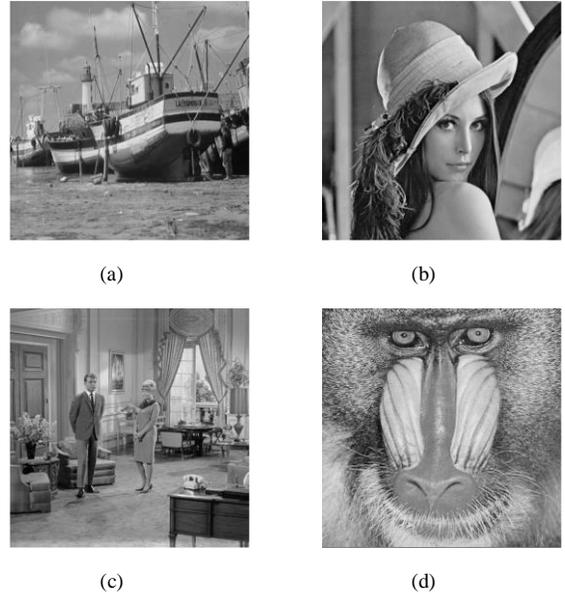


Fig. 6. Test images: (a)Boat, (b)Lena, (c) Livingroom, (d)Baboon.

implementations deliver matrices that, a priori, would destabilize the BCS-SPL algorithm. In this paper, we propose one way to adapt this algorithm to work with these generators. To this purpose, we combine the BCS-SPL algorithm with singular value decomposition in order to design a more robust and stable reconstruction algorithm, called BCS-SVD. Let $\hat{\mathbf{x}}_i^{[k]}$ be the estimated image at the k -th iteration, partitioned into H blocks. Furthermore, suppose that the SVD decomposition of the measurement matrix Φ is given by:

$$\Phi = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (19)$$

where \mathbf{U} and \mathbf{V} are orthogonal matrices of size $M_b \times M_b$ and $N_b \times N_b$ respectively, and their column are known as the singular vectors of the matrix Φ . \mathbf{D} is a diagonal matrix of size $M_b \times N_b$ and its element along its diagonal are known as the singular values [30]. The proposed algorithm updates the i -th block $\hat{\mathbf{x}}_i^{[k]}$ iteratively by means of the following approximating operator:

$$\hat{\mathbf{x}}_i^{[k+1]} = \hat{\mathbf{x}}_i^{[k]} + \Phi^T \mathbf{U} \mathbf{S} \mathbf{U}^T (y_i - \Phi \hat{\mathbf{x}}_i^{[k]}) \quad (20)$$

where \mathbf{S} is a diagonal matrix of size $M_b \times M_b$ whose entries are obtained by taking the reciprocal of the corresponding squared non-zero elements of the matrix \mathbf{D} . The most useful feature of the approximating operator in BCS-SVD is that it partially generalizes matrix inversion of the measurement matrix Φ . This operator is based on the error between the original samples of each block and the corresponding approximated samples, obtained by the estimated block at the previous iteration. Using this residual, it forms a correction equation for which this process is repeated.

In the next step, all the reconstructed blocks at iteration k are put back into the full-image plane to update the reconstructed image $\hat{\mathbf{x}}^{[k+1]}$ using the same smoothing and thresholding operations, proposed in the BCS-SPL algorithm [25]. The smoothing operation is done by Wiener filtering which reduces the blocking artefacts. The aim of thresholding process is

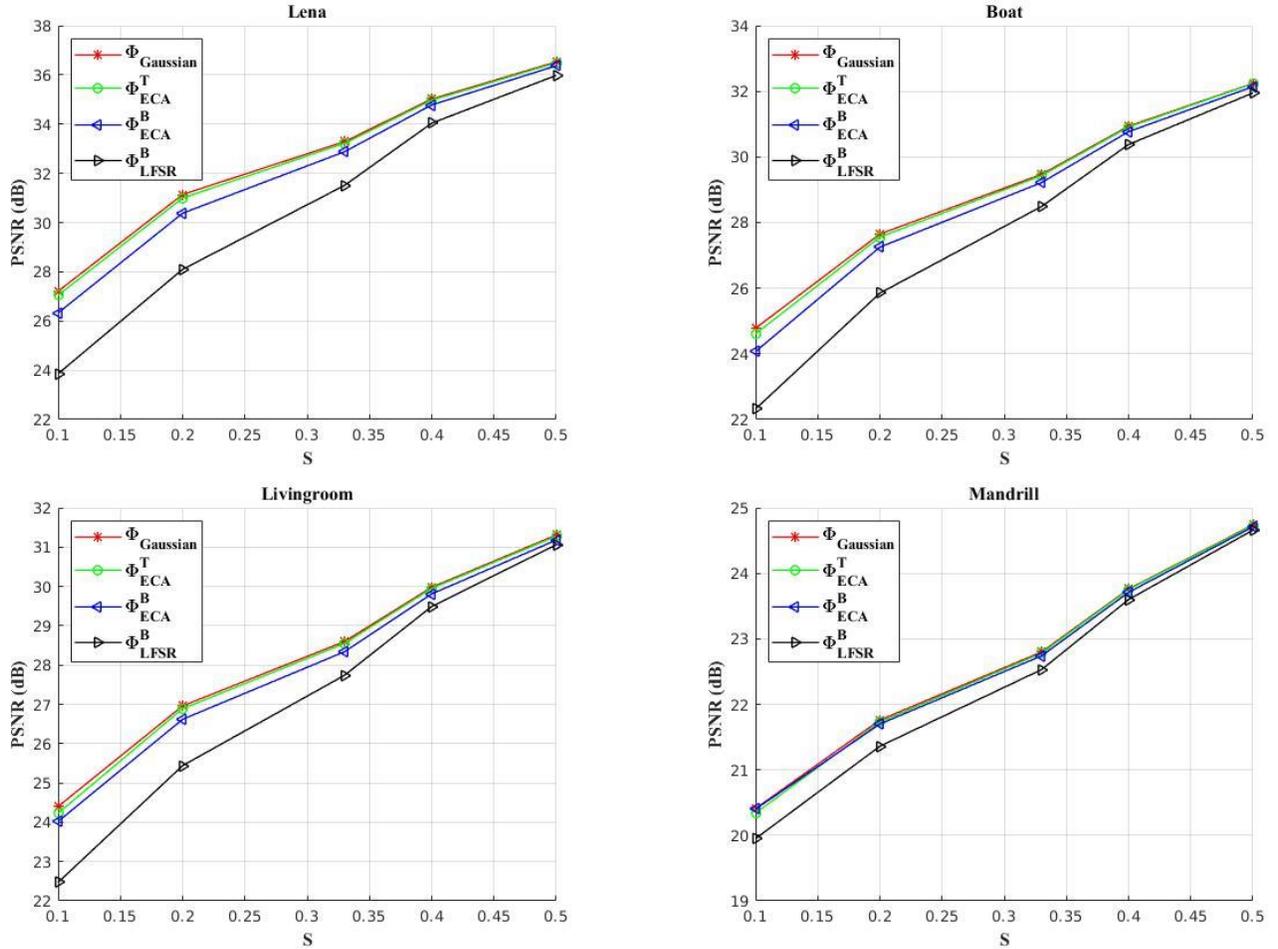


Fig. 7. PSNR of the reconstructed images as a function of substrate (S).

controlling the local sparsity that is well-known in the natural image and video signals. To this purpose, soft thresholding is done in the transform domain Ψ by:

$$\tilde{\mathbf{x}}^{[k+1]} = \begin{cases} \tilde{\mathbf{x}}^{[k+1]} & |\tilde{\mathbf{x}}^{[k+1]}| \geq \tau \\ 0 & \text{else} \end{cases} \quad (21)$$

where τ is a predetermined threshold and $\tilde{\mathbf{x}}^{[k+1]}$ is the transformed image. A wide range of sparse transforms can be used, such as discrete cosine and wavelet transforms. The most effective one, as proposed in [25], is the dual-tree discrete wavelet transform (DDWT). Such a procedure is iterated until the difference of successive approximations of the image is a very small value, *i.e.* $\|\tilde{\mathbf{x}}^{[k+1]} - \tilde{\mathbf{x}}^{[k]}\| \leq \varepsilon$. The details of the proposed BCS-SVD algorithm are summarized in Table 1. The operator $\mathbf{R}(\cdot)$ splits the image into H non-overlapping blocks and the operator $\mathbf{R}^{-1}(\cdot)$ puts back the reconstructed blocks into the corresponding positions in the reconstructed image, padded with zeros elsewhere.

Eq. (20) is derived to solve the following minimization problem

$$\operatorname{argmin}_{\mathbf{x}_i} \|\mathbf{y}_i - \Phi \mathbf{x}_i\|_2 + \lambda \|\mathbf{x}_i\|_1 \quad (22)$$

where λ is the regularization parameter. Eq. (20) is the enhanced version of the projected Landweber step proposed in BCS-SPL [25] which, with the thresholding operator of Eq. (21), iteratively improves the reconstruction quality. This iterative threshing process has been analysed extensively in the literature. Different theoretical analyses of the BCS-SPL algorithm applied to the CS recovery problem have been given in [31][32][33]. BCS-SPL is guaranteed to converge to a local minimum of Eq. (22). The argument for BCS-SVD follows the same line as that in [31].

V. EXPERIMENTAL RESULTS

In this section, the performance of the CS measurement matrix, described in sections III, and the BCS-SVD algorithm, proposed in the section IV, will be evaluated via a suite of simulations carried out on 8-bit grayscale standard images of size 512×512 pixels. Here, as shown in Fig. 6, we report the results of four classic images often used in image processing: Boat, Lena, Livingroom, and Baboon³. The Peak Signal-to-Noise Ratio (PSNR) is chosen as objective quality measure for our experimental framework. PSNR is an approximation to

³ The MATLAB implementation of the ternary measurement matrix obtained using ECA and the BCS-SVD recovery algorithm can be found, along with more test at the address: www2.imse-cnm.csic.es/icaveats/tcsvt

human perception of reconstruction quality from lossy compression. When used for this purpose the original data is

accuracy of the reconstruction result.

Experiments are designed with two goals: first, the

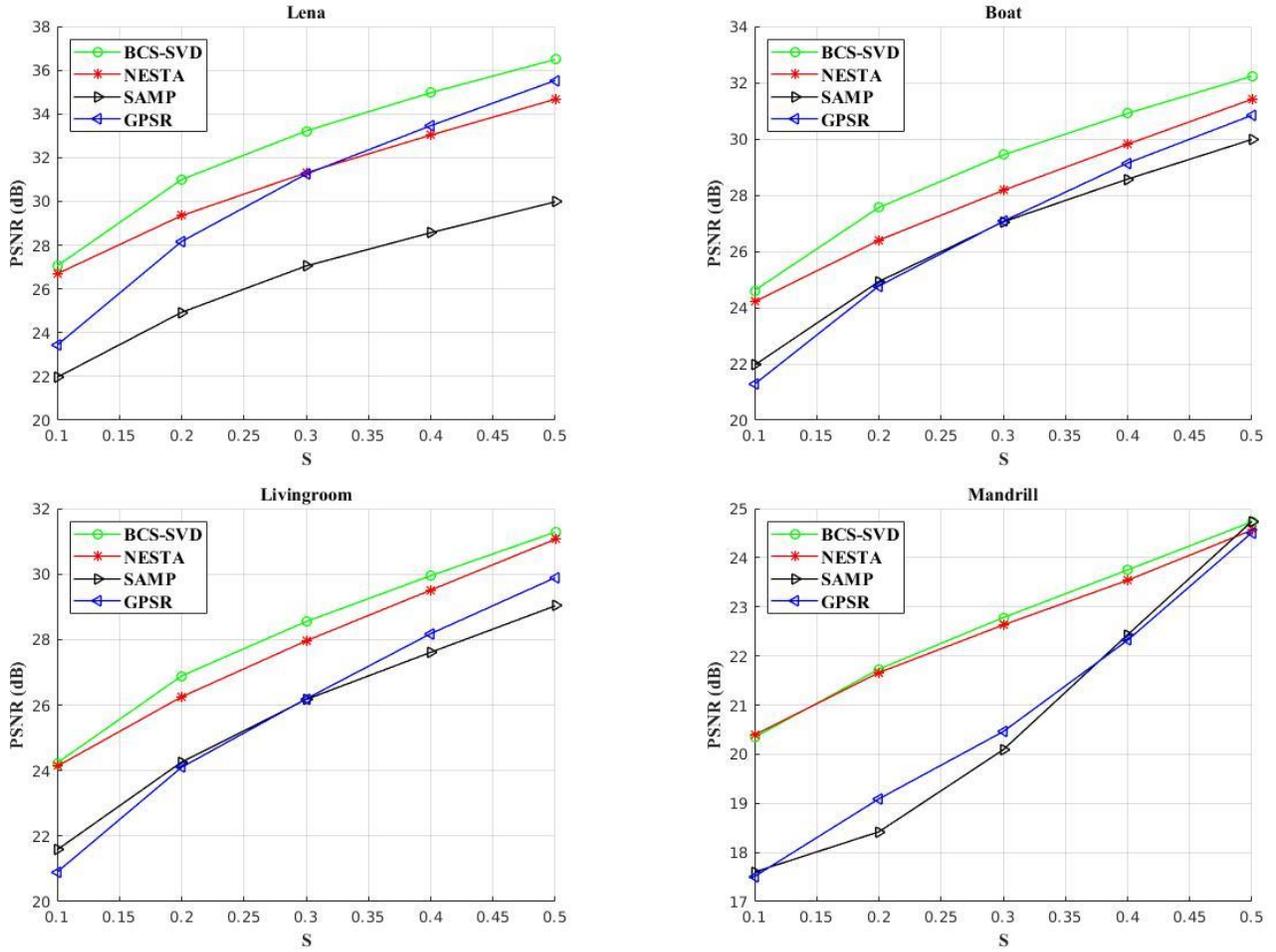


Fig. 8. PSNR of the reconstructed images using different recovery algorithms as a function of subrate (S).

considered the signal and the error introduced by compression is the noise:

$$\text{PSNR} = 10 \log_{10} \frac{\max(\mathbf{x})^2}{\frac{1}{MN} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2} \quad (23)$$

being $\max(\mathbf{x})$ the highest possible pixel value. The simulations were carried out in MATLAB. In each set of simulations, 30 trials are performed due to the random nature of generated measurement matrices and the average PSNR is computed. For the BCS sampling, we use the block dimension of size 32×32 , *i.e.* $N_b = 1024$. The value of threshold in the reconstruction algorithm is set to $\varepsilon = 10^{-4}$. We remind the reader that dividing the pixel array into smaller sub-arrays that can be digitized independently is essential for two reasons. It helps the implementation of practical measurement matrices in CS-CISs and it reduces software resources needed during reconstruction. Usually fast and accurate reconstruction algorithms opt for small blocks, typically 8×8 . On the other hands, blocks that are too small deteriorate the quality of the sensor introducing asymmetries within the pixel array. We chose a block size of 32×32 as a compromise between the two divergent necessities. We have set the value of threshold in the reconstruction algorithm to be $\varepsilon = 10^{-4}$ because we saw that it was a good compromise between reconstruction time and

performance of ECA-based ternary matrices for CS should be compared with other types of matrices; second, the stability and robustness of the proposed measurement matrices with different recovery algorithms should be evaluated. This test set up includes 480 trials. More examples can be found in our online repository³.

A. Validation of the Proposed Measurement Matrices

In this subsection, the performance of the proposed ternary and binary pseudo random measurement matrices based on ECA is compared to performance of the pseudo random measurement matrices based on LFSR. In addition, we have included the Gaussian random matrices, whose entries are randomly selected from a Gaussian distribution, as a reference. We conducted a series of test during which we applied the CS using four matrices: Φ_{Gaussian} a Gaussian random matrix chosen *i.i.d.* from $\mathcal{N}(0,1)$ and then normalized; Φ_{LFSR}^B a binary pseudo random measurement matrix based on LFSR; Φ_{ECA}^B a Binary pseudo random measurement matrix based on ECA; and Φ_{ECA}^T a ternary pseudo random measurement matrix based on ECA.

First, the test images are divided into non-overlapping blocks of size 32×32 pixels and each block is compressed with the

abovementioned measurement matrices. Then, the test images



Fig. 9. The reconstructed image *Lena* at subrate = 0.1: (a)BCS-SVD, (b) SAMP, (c)NESTA, (d)GPSR.

are recovered using the proposed recovery algorithm in section IV. This process is repeated several times per image and then the results are averaged.

Fig. 7 demonstrates the quality of reconstructed images in terms of PSNR as a function of subrate (S), where S varies from 0.1 to 0.5. As shown in Fig. 7, ternary matrices can be used to compressive sample the images and their performances degrade when the subrate decreases. An interesting observation is that the ternary matrix Φ_{ECA}^T outperforms binary matrices almost reaching the Gaussian random matrix $\Phi_{Gaussian}$. Fig. 9 represent *Lena* reconstructed using a ternary measurement matrix generated using ECA with a subrate of 0.1. We will provide more images and results along with the MATLAB code used to obtain them in an online repository³. Depending on the subrate, on average, the proposed ternary matrix improves the PSNR of the reconstructed images from 0.08 dB to 0.35 dB when compared with the binary measurement matrix Φ_{ECA}^B and from 0.27 dB to 1.90 dB when compared with the binary measurement matrix Φ_{ECA}^B . As can be seen, the performance the binary measurement matrix Φ_{ECA}^B is lower than that of the ternary matrix Φ_{ECA}^T at lower subrates. However, it shows a good performance when higher subrates are considered. For example, at $S = 0.5$, the PSNR of reconstructed images using the samples obtained from the matrix Φ_{ECA}^B matrix is almost 0.08 dB lower on average than that of the matrix Φ_{ECA}^T . Finally, it should be particularly noted that the ternary matrix Φ_{ECA}^T competes with the performance of the Gaussian random matrix, which is an optima theoretical measurement matrix. It should be reminded that the proposed ternary matrix is much easier to implement in hardware and that adds a valuable advantage to it.

B. Stability of the Ternary Measurement Matrices

The aim of the simulation in this section is to show that the

images can be recovered almost perfectly from the compressed

Table 2: *Lena* reconstruction time For Subrate=0.3

| Algorithm | NESTA | SAMP | GPSR | BCS-SVD |
|-----------|-------|------|------|---------|
| Time (s) | 65 | 41 | 97 | 2.5 |

samples obtained by the proposed ternary measurement matrix, when different recovery algorithms are used. To this purpose we use NESTA algorithm [34], gradient projection for sparse reconstruction (GPSR) [35], and sparsity adaptive matching pursuit (SAMP) [36] and we compare their performances with that of BCS-SVD algorithm. We chose these algorithms due to the fact that they are robust and perform well for recovering images sampled using sub-optimal matrices obtained from PRNGs. During this simulation, the test images are divided into non-overlapping blocks of size 32×32 pixels. We applied the measurement process of CS with the ternary measurement matrix Φ_{ECA}^T . Then, the test images are recovered using NESTA and BCS-SVD respectively. The same measurement matrix is used for both algorithms.

Fig. 8 shows the quality of reconstructed images in terms of PSNR at different subrates. It can be observed that both algorithms can recover the images, although the proposed BCS-SVD recovery algorithm has better performance in terms of PSNR compared with the other algorithms. As another result, at the same subrate S , the test images like *Boat* and *Lena* are recovered with ~ 1 dB gain w.r.t. NESTA. This is due the fact that these images are more compressible, i.e. have a higher sparsity level in the wavelet domain. A visual comparison of the various algorithms is shown in Fig. 9.

C. Complexity

Finally, it should be noted that the proposed scheme is valuable because it reduces the complexity of both the measurement and recovery processes of CS; at the sensor side, encoding is achieved without a huge memory requirement and, at the decoder side, the proposed BCS-SVD recovery algorithm is able to provide a fast and accurate solutions. The BCS-SVD algorithm is very simple. Each iteration involves the application of two operators, Φ and Φ^T , as well as two vector additions. It should be noted that SVD decomposition is carried out once for the whole recovery and its complexity is ignored. The thresholding process Eq. (21) also involves a partial ordering of the elements $\hat{x}_i^{[k]}$ in magnitude and has ignorable complexity. Apart from the storage of samples y_i , we only require the storage of vector $\hat{x}_i^{[k]}$ for each block, which is of length N_b . The bottle neck, both in terms of storage and computation time, is due to the operators Φ and $\Phi^T \mathbf{U} \mathbf{S} \mathbf{U}^T$. So, the computational complexity of each iteration is $O(M_b \times N_b)$. To justify our claim regarding the fast implementation recovery using the BCS-SVD algorithm, we have compared the required time for reconstructing the test image *Lena* using the above-mentioned algorithms at the subrate 0.3 in Table 2. As can be seen in Table 2, the proposed algorithm indicates a lower order of complexity.

VI. CONCLUSIONS

In this paper we have presented a new hardware-friendly ternary measurement matrix for CS acquisition of natural

images and a reconstruction algorithm capable of exploiting this matrix for fast and accurate recovery. We have modelled through a set of equations the challenges and limitations that arise during the design of compressive sampling CMOS image sensors and we have established why PRNG are essentials for the practical implementation of this type of circuits. We have bound the RIP of the binary measurement matrices that these generators produce with a constant that is very strict. We overcame this shortcoming presenting a novel sensor's topology capable of delivering ternary pseudo-random matrices by means of ECA and a differential readout system. The proposed measurement matrix approaches closely the ideal case of normalized Gaussian random measurement matrix. Further, a fast and accurate acquisition and reconstruction system is proposed with the BCS-SVD algorithm. The experiments are promising and show the possibility of using this system in real time applications such as video surveillance or data transmission with channels that have a high noise and/or limited bandwidth.

REFERENCES

- [1] J. A. Tropp, J. S. Wright. "Computational Methods for Sparse Solution of Linear Inverse Problems". *Proc. of the IEEE*, Vol. 98, No. 6, pp. 948-958, Jun 2010.
- [2] E. Candès. "Compressive sampling". *Int. Congress of Mathematics*, pp. 1433-1452. Madrid, Spain. August, 2006.
- [3] R. G. Baraniuk, V. Cevher, M. B. Wakin. "Low-Dimensional Models for Dimensionality Reduction and Signal Recovery: A Geometric Perspective". *Proc. of the IEEE*, Vol. 98, No. 6, pp. 959-971, Jun 2010.
- [4] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, R. G. Baraniuk. "Single-Pixel Imaging via Compressive Sampling". *IEEE signal Processing Magazine*, Vol. 25, No. 2, pp. 83-91. March, 2008.
- [5] V. Majidzadeh, L. Jacques, A. Schmid, P. Vanderghyest and Y. Leblebici. "A (256x256) Pixel 76.7mW CMOS Imager/Compressor Based on Real-Time In-Pixel Compressive Sensing". *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2956-2959. Paris, France. May, 2010.
- [6] Y. Oike, A. El Gamal. "CMOS Image Sensor With Per-Column $\Sigma\Delta$ ADC and Programmable Compressed Sensing". *IEEE Journal of Solid-State Circuits*, Vol. 48, No. 1, pp. 318 - 328, January. 2013.
- [7] Lu Gan "Block Compressed Sensing of Natural Images", 15th International Conference on Digital Signal Processing, pp. 403-406. Singapore, Republic of Singapore. July, 2007.
- [8] R. A. DeVore. "Deterministic constructions of compressed sensing matrices," *J. Complexity*, vol. 23, pp. 918-925, 2007.
- [9] S.-T. Xia, X.-J. Liu, Y. Jiang and H.-T. Zheng. "Deterministic Constructions of Binary Measurement Matrices from Finite Geometry". *IEEE Transactions on Signal Processing*, Vol. 63, No. 4, pp. 1017-1029, February 2015.
- [10] S. Jafarpour, M. F. Duarte, R. Calderbank, "Beyond worst-case reconstruction in deterministic compressed sensing," *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. pp. 1852-1856. Cambridge, MA, USA. July, 2012.
- [11] G. Boeing. "Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction". *Systems*, Vol. 4, No. 37. Nov, 2016.
- [12] S. Saravanan, M. Lavanya, R. V. Sai, R. Kumar. "Design and Analysis of Linear Feedback Shift Register Based on Various Tap Connections". *Elsevier, Procedia Engineering*, Vol. 38, pp. 640-646. 2012.
- [13] G. Mazzini, R. Rovatti, G. Setti. "Randomly-Flipped Linear Feedback Shift Registers: Spectral Analysis and Conjectures". *Proceedings of 2008 IEEE International Symposium on Spread Spectrum Techniques and Applications*, pp. 751-755. Bologna, Italy. August, 2008.
- [14] B. Kaliannan, V. S. Rao Pasupureddi. "A Low Power CMOS Imager Based on Distributed Compressed Sensing". *27th International Conference on VLSI Design and 13th International Conference on Embedded Systems*, pp. 534 - 538. Mumbai, India. January. 2014.
- [15] M. Dadkhah, M. Jamal Deen, S. Shirani. "CMOS Image Sensor With Area-Efficient Block-Based Compressive Sensing". *IEEE Sensor Journal*, Vol. 15 No. 7, pp 3699-3710, July, 2015.
- [16] P. Indyk, "Explicit constructions for compressed sensing matrices," in *Proc. 19th Annu. ACM-SIAM Symp. Discr. Algorithms*, Jan. 2008, pp. 30-33.
- [17] W. Lu, W. Li, K. Kpalma, J. Ronsin. "Near-optimal Binary Compressed Sensing Matrix". Manuscript submitted to *IEEE Transaction on Information Theory*, March 2013.
- [18] X. Shu-Tao, L. Weizhi. "Construction of ternary matrices with small coherence for compressed sensing". *Electronic Letters*, Vol. 52, No. 6, pp. 447-448, Mar 2016.
- [19] E. Jen. "Aperiodicity in one-dimensional cellular automata". *Physica D: Nonlinear Phenomena*, Vol. 45 No. 1-3, pp 3-18. September, 1990.
- [20] S. Wolfram. "Statistical Mechanics of Cellular Automata". *Review of Modern Physics*, Vol. 5, No. 3, pp. 601-644. July, 1983.
- [21] S. Ninagawa. "Power Spectral Analysis of Elementary Cellular Automata". *Complex Systems Publications, Inc.*, Vol. 17, No. 4, pp. 399-411. Dec, 2008.
- [22] S. Dhingra. "Comparison of LFSR and CA for BIST". Dept of Electrical and Computer Engineering Auburn University, [online] Available: <http://www.researchgate.net/publication/245717746>.
- [23] R. Obermeier, J.A: Martínez-Lorenzo. "Sensing Matrix Design via Mutual Coherence Minimization for Electromagnetic Compressive Imaging Applications". *IEEE Transactions on Computational Imaging*, Vol. 1 No. 2, pp 217-229. Jun, 2017.
- [24] B. A. Ndaw, D. Sow and M. Sanghare " Construction of Maximum Period Linear Feedback Shift Registers (LFSR) (Primitive Polynomials and Linear Recurring Relations)", *British Journal of Mathematics & Computer Science*, pp. 403-406. Vol. 11, No. 4, pp. 1-24. 2015.
- [25] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms", *Proceedings of IEEE International Conference on Image Processing*, Cairo, Egypt, Nov. 2009, pp. 3021-3024.
- [26] A. Akbari and M. Trocan, "Robust image reconstruction for block-based compressed sensing using a binary measurement matrix," *IEEE International Conference on Image Processing*, Athens, Greece, 2018.
- [27] A. Akbari, D. Mandache, M. Trocan, and B. Granado, "Adaptive saliency-based compressive sensing image reconstruction," *Proceedings of IEEE International Conference on Multimedia Expo Workshops*, Seattle, WA, July 2016, pp. 1-6.
- [28] M. Trocan, T. Maugey, J. Fowler, and B. Pesquet-Popescu, "Disparity-compensated compressed-sensing reconstruction for multiview images," *Proceedings of IEEE Conference on Multimedia and Expo*, Suntec City, Singapore, July 2010, pp. 1225-1229.
- [29] A. Akbari, M. Trocan, and B. Granado, "Residual based compressed sensing recovery using sparse representations over a trained dictionary," *Proceedings of International ITG Conference on Systems, Communications and Coding*, Hamburg, Germany, Feb. 2017, pp. 1-6.
- [30] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, Technical University of Denmark, Nov. 2012.
- [31] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications, special issue on sparsity*, vol. 14, no. 5, pp. 629-654, 2008.
- [32] K. Bredies and D. A. Lorenz, "Linear Convergence of Iterative Soft-Thresholding," *Journal of Fourier Analysis and Applications, special issue on sparsity*, vol. 14, no. 5, pp. 812-837, 2008.
- [33] A. Akbari, M. Trocan, and B. Granado, "Sparse recovery-based error concealment," *IEEE Transactions on Multimedia*, vol. 19, no. 99, pp. 1339-1350, 2017.
- [34] A. Becker, J. Bobin, and W. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1-39, 2011.
- [35] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application compressed sensing and other inverse problems," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 4, pp. 586-597, 2007

- [36] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, October 2008, pp. 581–587.