J Real-Time Image Proc (2019) 16:1525–1540
DOI 10.1007/s11554-016-0658-z

ORIGINAL RESEARCH PAPER

Compressive sensing for images using a variant of Toeplitz matrix for wireless sensor networks

S. Aasha Nandhini[1] · S. Radha[1] · P. Nirmala[1] · R. Kishore[1]

Received: 19 May 2016 / Accepted: 18 November 2016 / Published online: 15 December 2016

Abstract Recent advancement in the field of wireless sensor networks (WSNs) has enabled its use in a variety of multimedia applications where the data to be handled are large that require more memory for storage and high bandwidth for transmission. As WSNs have limited capabilities in terms of computation, memory, energy and bandwidth, compression becomes necessary. The traditional compression methods consume more energy as well as memory which can be overcome by compressive sensing (CS) technique. CS is an emerging technique for efficiently acquiring and reconstructing the signal by processing the reduced number of samples specified by the Nyquist criterion. The objective of this paper is to implement CS for images using the proposed sensing matrix derived from the Toeplitz matrix and its variants. For reconstruction purpose, an existing greedy orthogonal matching pursuit algorithm is used. The measurements obtained from the framework are transmitted in real time using TelosB nodes under Contiki OS platform. The simulation results are compared with the experimental results, and the performance of the CS framework is evaluated in terms of peak signal-to-noise ratio, storage overhead, energy computation, computational time, transmission energy and end-to-end transmission latency. The results show that the performance of the proposed sensing matrix is better in terms of memory requirement, energy computation and computational complexity when compared with an existing Gaussian matrix.

✉ S. Aasha Nandhini
aasha.nandhu@gmail.com

S. Radha
radhas@ssn.edu.in

P. Nirmala
nirmalavp.ece@gmail.com

R. Kishore
kishorer@ssn.edu.in

[1] ECE Department, SSN College of Engineering, Kalavakkam, Chennai 603110, India

Keywords CS · Measurement matrix · Proposed sensing matrix · Orthogonal matching pursuit · WSN · TelosB nodes · Contiki OS

1 Introduction

A WSN consists of several sensor nodes deployed in inaccessible areas to monitor the physical or environmental conditions such as temperature, vibration, sound and pressure. WSNs are used for a variety of applications such as military, civil, health monitoring, fleet monitoring, habitat monitoring, preventing theft, home and industrial automation. A sensor node consists of five basic units such as sensing unit, a central processing unit, a storage unit, a transceiver unit and a power unit [1].

In the case of multimedia applications, the data to be handled by the network are too large that make compression a necessary process [2]. Compression takes into account all the samples in the image which requires more memory for storage and more energy for computation. In traditional image compression, the image is transformed into an appropriate basis and only the significant coefficients are encoded. Instead, the process, in which the data compression can be performed simultaneously along with acquisition, is termed as compressed sensing.

CS is applied to the sparse signal, i.e. the measurement matrix is multiplied with the sparse vector to obtain the measurements which are transmitted for reconstruction [3].

There are many recovery algorithms proposed for CS which reconstructs the original signal from the measurements. In this paper, TelosB nodes [4] are configured as a network using Contiki OS [5] and the measurements acquired are transmitted in real time using TelosB nodes. An existing greedy orthogonal matching pursuit (OMP) algorithm is used for reconstruction, and the research findings are discussed.

The rest of the paper is organized as follows. Section 2 provides a brief description of the CS technique. Section 3 explains the proposed CS framework, and Sect. 4 discusses the research findings. Section 5 provides a brief survey of related works; finally, Sect. 6 gives the conclusions and proposed future work.

2 Compressed sensing

Compressed sensing also called as sub-Nyquist sampling acquires measurements directly without considering all the samples by using a measurement process [6]. This process can be applied to sparse or compressible signals. CS theory asserts that original signal can be reconstructed from very few measurements. Table 1 shows the notations and their meaning as used in this paper. The notations are explained in subsequent sections when they are used. Consider a signal $x = (x_i)_{i=1}^{N} \in R^N$ which by itself is sparse (i.e. having few nonzero coefficients) or it is made sparse by using an orthonormal basis of $N \times 1$ vectors $\{\Psi_i\}_{i=1}^{N}$. Using a basis matrix of dimension $N \times N$ with $\{\Psi_i\}$ as columns, the signal can be expressed as

$$x = \Psi r \tag{1}$$

where r is a $N \times 1$ sparse vector with K nonzero coefficients. The signal x is said to be highly sparse if the representation described in (1) has very few nonzero coefficients.

Measurement matrix also called as sensing matrix is used to obtain the measurements $M \ll N$ by computing the inner product between the signal x and the vectors represented as $\{\Phi_j\}_{j=1}^{M}$. The measurement vector (y) of dimension $M \times 1$ is represented as

$$y = \Phi x \tag{2}$$

Using (1), the measurement vector can be represented as

$$y = \Phi \Psi r$$

$$y = Ar \tag{3}$$

where $A = \Phi\Psi$ represents an $M \times N$ matrix. The minimum number of measurements (M) required to reconstruct the signal is obtained from (4)

$$M \geq K \log(N/K) \tag{4}$$

Table 1 Notations used in this paper

x	Signal
Ψ	Basis matrix
r	Sparse vector
K	Sparsity level
N	Total number of samples
M	Number of measurements
Φ	Measurement matrix
y	Measurement vector
n	Block size
B_N	Total number of blocks
K_1	Sparsity level of block
M_1	Number of measurements in a block
y_1	Measurement vector of block
T_P	Toeplitz matrix of order P, $P = n^2$
H_P	Hankel matrix of order P
C_P	Circulant matrix of order P
S_P	Proposed sensing matrix of order P
d	Distance between nodes
E_G	Energy for Gaussian sensing matrix
E_T	Energy for Toeplitz matrix
E_H	Energy for Hankel matrix
E_C	Energy for circulant matrix
E_S	Energy for proposed sensing matrix
E_{dct}	Energy for DCT computation
E_{csm}	Energy for CS measurements
E_{tot}	Energy for CS framework
E_{enc}	Energy for encoding process
E_{tx}	Theoretical transmission energy
t	Time to transmit a 128-byte packet
I	Current consumption
V	Voltage
E_{mtx}	Practical transmission energy
G_T	Computational time for Gaussian sensing matrix
S_T	Computational time for proposed sensing matrix
$d_{\mathrm{end-end}}$	End-to-end transmission delay
d_{trans}	Transmission delay
d_{prop}	Propagation delay
h	Number of hops

where K represents the sparsity level and N is the total number of samples [3, 7]. There are various types of measurement matrices such as Gaussian matrix, Bernoulli matrix, Fourier matrix, Toeplitz matrix and circulant matrix. The measurement matrix of dimension $M \times N$ is generated by calculating the minimum number of measurements M required for reconstructing the original signal. The measurement process for M is non-adaptive, and hence, Φ is independent of the signal x. In order to reconstruct the original signal, an efficient recovery algorithm and M measurements are needed. For good reconstruction, the measurement matrix must satisfy two important properties such as restricted isometry property (RIP) and incoherence property. The concept of CS is explained in Fig. 1.

The input signal is made sparse using a transform domain, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT). The resulting vector is called as the sparse vector to which the measurement matrix is applied to obtain the measurements. These measurements are transmitted to the receiver side for reconstruction. There are a number of CS recovery algorithms such as basis pursuit, greedy algorithms and iterative algorithms that are used for reconstructing the original signal from the measurements.

3 Related works

Talari and Rahnavard [8] proposed a fully distributed and efficient data storage scheme for WSN. It compresses the natural signals and broadcasts through wireless channels. According to the authors, after dissemination phase each node will have a compressed sample and with the help of data collector all the readings can be recovered. They found the optimal parameters for compressive storage (CStorage) method to achieve the minimum number of required transmissions that is considerably lower than the required transmissions in the existing protocols. The limitation is that the authors have not considered the temporal correlation of the data.

Liu et al. [9] proposed a multiple event detection scheme using CS. The authors focused on the sensed data to estimate the region or boundary of an event. Here, CS can be used to reconstruct the source signal that contains multiple simultaneous events. Moreover, the events may not change often so that the source signals at two adjacent time instants have high redundancy. The temporal correlation is utilized to improve the detection accuracy and the advantage of this method is that it detects not only the positions but also the values of the events. The drawback of the method is that it requires more number of sensors.

Gan [10] has proposed block-based CS for natural images, in which image acquisition is conducted in a

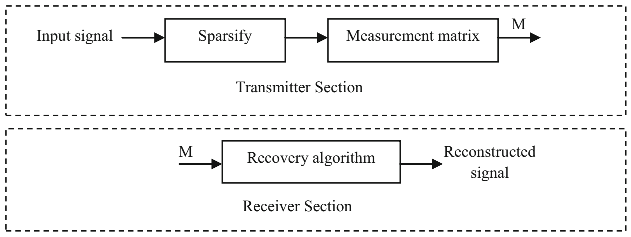


Fig. 1 Block diagram of compressed sending

block-by-block manner. The image reconstruction algorithm includes both linear and nonlinear operations such as Wiener filtering, projection onto the convex set and hard thresholding in the transform domain. The results show comparable performances between block-based CS systems and current CS schemes, but the former have much lower implementation costs.

Kim et al. [11] described a specialized interior point method for solving CS problems that uses a preconditioned conjugate gradient method to compute the search step. This method can efficiently solve large CS problems, by making use of fast algorithms for the signal transform. The method is demonstrated with a medical resonance imaging (MRI) example. The drawback of the proposed method is that the object of interest must be known in order to collect the samples in spatial domain.

Sermwuthisarn et al. [12] proposed the block-based OMP technique. In this method, fast image recovery is achieved by dividing the image information into blocks of $n \times n$ pixels and then applying block-based OMP to each $n \times n$ block. The limitation is that the technique has to search an extremely large dictionary to find the best matching blocks.

Shahidan et al. [13] proposed an image transferring mechanism for realizing JPEG motion data transfer in wireless sensor networks where the data are basically generated from the sequence of compressed images. They implemented the image sequence transfer with data buffering mechanism that overcomes the drawback of serious restriction in transferring multimedia data as it requires huge bandwidth and processing capabilities.

Zhou et al. [14] proposed a non-uniform sampling-based CS for image processing. In this technique, the DCT coefficient of an image block is scanned in zigzag order to form a single column vector. This column vector is divided into important and unimportant components. CS is applied only to the unimportant components, and the important components are transmitted directly without any modification that reduces the CS measurement cost. The drawback of this method is that even though it reduces the measurement cost the compression complexity remains high in real-time process.

Li et al. [15] introduced partial orthogonal symmetric Toeplitz matrices as a sensing matrix and proved that this class of matrices satisfies statistical RIP with high probability. Because of the Toeplitz structure, these new sensing matrices can be applied in channel estimation and signal compression with lower computational and storage complexity.

Yin et al. [16] introduced fast algorithms for reconstructing signals from incomplete Toeplitz and circulant measurements. The computational results show that the Toeplitz and circulant matrices are not only as effective as random matrices for signal encoding, but also permit much faster decoding.

Yu et al. [17] proposed a method to construct a Toeplitz-structured matrix with chaotic sequences for CS and proved that the Toeplitz-structured chaotic sensing matrix retains the restricted isometry property (RIP), which guarantees exact recovery. The results show similar or better performance compared with the chaotic sensing matrix and the Gaussian sensing matrix.

Bajwa et al. [18] numerically compared the performance of Toeplitz and circulant matrices with the Gaussian matrix. The Toeplitz and circulant matrix entries are drawn from the Bernoulli distribution and reconstructed using gradient projection algorithm where matrix multiplications are carried out using FFT. The authors compared the results with IID matrices and proved that the Toeplitz and circulant matrices yield better performance in terms of storage complexity for a large-dimensional signal.

In [19], a non-uniform compressive sensing (NCS) method was proposed to improve the performance of WSNs by exploiting both compressibility and heterogeneity. This method provides a more accurate temporal–spatial profile for a given energy budget compared with other sampling methods. The results show that the implementation of NCS framework introduced very little communication overheads, compared to other approaches based on traditional CS, and also achieved similar signal approximation accuracy with significantly less samples.

In [20], energy dissipation models for CS and conventional approaches are built to construct a mixed integer programming framework that jointly captures the energy costs for computation and communication. A model to quantify the energy dissipation in sensor nodes due to data acquisition, computation and communication is developed using the characteristics of the Mica sensor network platform. The results show that CS prolongs network lifetime for sparse signals and is more advantageous for WSNs with a smaller coverage area.

Han et al. [21] have proposed an image representation scheme using CS because it reduces the computational complexity of the image/video encoder used in the compression process. The encoder first divides the image into two parts, dense and sparse components, where the sparse component alone is encoded using CS. The number of random measurements needed and the decoding complexity are reduced considerably.

In [22], the authors have proposed an energy-efficient image transmission scheme for WSN using compressed sensing. A unique encoding method was developed for the compressed measurements to reduce the number of bits to be transmitted compared with Huffman and LEC measurements. The proposed image transmission scheme was experimentally validated in Mica2 platform, and the results show acceptable PSNR with reduced energy. The encoding technique achieves 66% reduction in bits. However, the proposed work was not implemented for real-time images.

Eleyan et al. [23] have proposed a novel measurement matrix for face recognition problem. The measurement matrix is generated using both zero mean and nonzero mean rows. The authors have also proposed a random measurement matrix generated from the binary entries. The authors have validated the proposed matrix on two databases, namely ORL and FERET databases. The results show promising performance compared to the conventional method.

In [24], the authors have proposed a simple and efficient incoherence rotated chaotic (IRC) matrix which makes use of pseudorandom chaotic sequence for its generation. The matrix was generated using the concept of incoherence and rotation. The proposed matrix was compared with Gaussian measurement matrix and other state-of-the-art methods. The results show that the matrix performs well for both natural images and remote sensing images. However, the computational complexity and storage space required for generating the matrix were not discussed.

4 Proposed CS framework

In this paper, the proposed sensing matrix is a combination of the Toeplitz, Hankel and circulant matrices. This sensing matrix is used in the measurement process to obtain the measurements that are further transmitted for reconstruction using the OMP algorithm.

Consider an input image which is divided into blocks of size $n \times n$ to reduce the complexity of the framework [25]. When the entire image of size $N \times N$ is considered, the CS process requires a large measurement matrix of size $M \times N^2$ for computing the measurements that will consume more memory and energy. In block-based CS framework, the block size is so small that it requires only a small measurement matrix of size $M_1 \times n^2$ for computation. In this framework, each block is converted into a single column vector that is sparsified by using DCT to obtain the sparse vector. The number of nonzero elements (K_1) in the vector represents the sparsity level of the block. The minimum number of measurements (M_1) required to reconstruct the original image is calculated using Eq. (4). A measurement matrix with dimension $M_1 \times n^2$ is generated and multiplied with the sparse vector to obtain the measurement vector (y_1).

The blocks of the original image are reconstructed from the measurements using the OMP algorithm [26–28]. OMP is a greedy algorithm which constructs an approximation by going through an iteration process. During different iterations, the column vector of A that resembles the

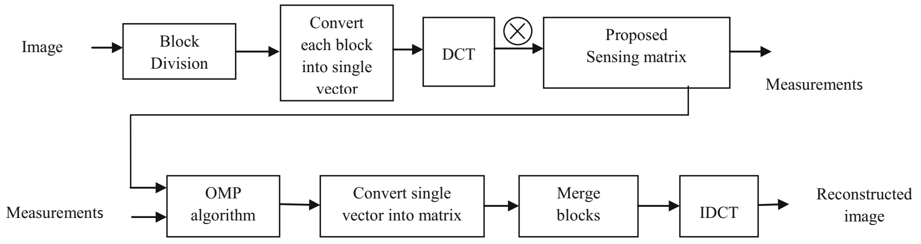


Fig. 2 Block diagram of proposed CS framework

residual vector corresponds to a nonzero entry of the sparse vector. For the first iteration, the residual vector will be the measurement vector. Thus, one nonzero entry of the signal (x) is estimated which is then subtracted from the observation vector (y_1) and repeating the same yields all the nonzero entries of the sparse signal. After reconstruction, an inverse discrete cosine transform (IDCT) is applied and the blocks are merged to obtain the entire image. Figure 2 shows the block diagram of the CS framework that uses the proposed sensing matrix to obtain the measurements.

The proposed matrix was designed using the Toeplitz matrix with Gaussian entries as the basis. The Toeplitz matrix was used to reduce the number of entries to generate the matrix. The Toeplitz matrix was generated using the Gaussian entries to achieve higher PSNR. When the sensing matrix is applied to the sparse vector, the compressive measurements are obtained which is a linear combination of all the pixels in the image. From these measurements, it is possible to reconstruct the image using appropriate compressed sensing-based recovery algorithm [3].

To design the proposed sensing matrix, first a $P \times P$ Toeplitz matrix (T) is generated using Gaussian distribution represented as

$$T_P = T_{i,j} = T_{i+1,j+1} = t_{i-j} \tag{5}$$

where T_P denotes the Toeplitz matrix of order P, $T_{i,j}$ denotes the (i,j)th position of the entries and t denotes the entries of the matrix. The size of P should be equal to n^2. The Toeplitz matrix is a constant diagonal matrix and satisfies the RIP and incoherence property as it uses the Gaussian entries.

A Hankel matrix (H) of size $P \times P$ is derived from the generated Toeplitz matrix which is expressed as

$$H_P = H_{i,j} = H_{i-1,j+1} \tag{6}$$

where H_P denotes the Hankel matrix of order P.

The circulant matrix (C) of dimension $P \times P$ is derived from the Toeplitz matrix by taking the first row of the Toeplitz matrix and rotating it one element to the right relative to the preceding row vector. After generating the Toeplitz matrix and its variants, the basic arithmetic operation such as matrix addition and subtraction is applied on it. First, the matrix addition operation is done on the Hankel and the circulant matrices, and then, the resulting matrix is subtracted from the Toeplitz matrix which gives a matrix (S) of size $P \times P$ that is expressed as

$$S_P = T_P - (H_P + C_P) \tag{7}$$

where S_P denotes the proposed sensing matrix of order P. For different measurements (M_1), the measurement matrix is designed by taking the first M_1 rows of the new matrix. The entries of the resulting matrix also follow Gaussian distribution and hence satisfy the RIP and incoherence property. This matrix requires only $(2P - 1)$ elements to generate the matrix when compared with the Gaussian matrix that requires $M_1 \times P$ elements and also storage, energy and time for generating the matrix are also less when compared with the Gaussian matrix.

An example of a generated Toeplitz matrix of order 8 is given as

$$T_8 = \begin{bmatrix} t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} & t_{-7} \\ t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} \\ t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} \\ t_4 & t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} \\ t_5 & t_4 & t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ t_6 & t_5 & t_4 & t_3 & t_2 & t_1 & t_0 & t_{-1} \\ t_7 & t_6 & t_5 & t_4 & t_3 & t_2 & t_1 & t_0 \end{bmatrix}$$

The Hankel matrix of order 8 which is derived from the generated Toeplitz matrix is given as

$$H_8 = \begin{bmatrix} t_7 & t_6 & t_5 & t_4 & t_3 & t_2 & t_1 & t_0 \\ t_6 & t_5 & t_4 & t_3 & t_2 & t_1 & t_0 & t_{-1} \\ t_5 & t_4 & t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} \\ t_4 & t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} \\ t_3 & t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} \\ t_2 & t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} \\ t_1 & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} \\ t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} & t_{-7} \end{bmatrix}$$

A circulant matrix of order 8 which is derived from the Toeplitz matrix is given as

$$C_8 = \begin{bmatrix} t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} & t_{-7} \\ t_{-7} & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} \\ t_{-6} & t_{-7} & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} \\ t_{-5} & t_{-6} & t_{-7} & t_0 & t_{-1} & t_{-2} & t_{-3} & t_{-4} \\ t_{-4} & t_{-5} & t_{-6} & t_{-7} & t_0 & t_{-1} & t_{-2} & t_{-3} \\ t_{-3} & t_{-4} & t_{-5} & t_{-6} & t_{-7} & t_0 & t_{-1} & t_{-2} \\ t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} & t_{-7} & t_0 & t_{-1} \\ t_{-1} & t_{-2} & t_{-3} & t_{-4} & t_{-5} & t_{-6} & t_{-7} & t_0 \end{bmatrix}$$

The proposed sensing matrix of order 8 is generated by adding the Hankel and circulant matrices and then subtracting the resulting matrix from the Toeplitz matrix which is given as

$$S_8 = \begin{bmatrix} t_0-(t_7+t_0) & t_{-1}-(t_6+t_{-1}) & t_{-2}-(t_5+t_{-2}) & t_{-3}-(t_4+t_{-3}) & t_{-4}-(t_3+t_{-4}) & t_{-5}-(t_2+t_{-5}) & t_{-6}-(t_1+t_{-6}) & t_{-7}-(t_0+t_{-7}) \\ t_1-(t_6+t_{-7}) & t_0-(t_5+t_0) & t_{-1}-(t_4+t_{-1}) & t_{-2}-(t_3+t_{-2}) & t_{-3}-(t_2+t_{-3}) & t_{-4}-(t_1+t_{-4}) & t_{-5}-(t_0+t_{-5}) & t_{-6}-(t_{-1}+t_{-6}) \\ t_2-(t_5+t_{-6}) & t_1-(t_4+t_{-7}) & t_0-(t_3+t_0) & t_{-1}-(t_2+t_{-1}) & t_{-2}-(t_1+t_{-2}) & t_{-3}-(t_0+t_{-3}) & t_{-4}-(t_{-1}+t_{-4}) & t_{-5}-(t_{-2}+t_{-5}) \\ t_3-(t_4+t_{-5}) & t_2-(t_3+t_{-6}) & t_1-(t_2+t_{-7}) & t_0-(t_1+t_0) & t_{-1}-(t_0+t_{-1}) & t_{-2}-(t_{-1}+t_{-2}) & t_{-3}-(t_{-2}+t_{-3}) & t_{-4}-(t_{-3}+t_{-4}) \\ t_4-(t_3+t_{-4}) & t_3-(t_2+t_{-5}) & t_2-(t_1+t_{-6}) & t_1-(t_0+t_{-7}) & t_0-(t_{-1}+t_0) & t_{-1}-(t_{-2}+t_{-1}) & t_{-2}-(t_{-3}+t_{-2}) & t_{-3}-(t_{-4}+t_{-3}) \\ t_5-(t_2+t_{-3}) & t_4-(t_1+t_{-4}) & t_3-(t_0+t_{-5}) & t_2-(t_{-1}+t_{-6}) & t_1-(t_{-2}+t_{-7}) & t_0-(t_{-3}+t_0) & t_{-1}-(t_{-4}+t_{-1}) & t_{-2}-(t_{-5}+t_{-2}) \\ t_6-(t_1+t_{-2}) & t_5-(t_0+t_{-3}) & t_4-(t_{-1}+t_{-4}) & t_3-(t_{-2}+t_{-5}) & t_2-(t_{-3}+t_{-6}) & t_1-(t_{-4}+t_{-7}) & t_0-(t_{-5}+t_0) & t_{-1}-(t_{-6}+t_{-1}) \\ t_7-(t_0+t_{-1}) & t_6-(t_{-1}+t_{-2}) & t_5-(t_{-2}+t_{-3}) & t_4-(t_{-3}+t_{-3}) & t_3-(t_{-4}+t_{-5}) & t_2-(t_{-5}+t_{-6}) & t_1-(t_{-6}+t_{-7}) & t_0-(t_{-7}+t_0) \end{bmatrix}$$

A measurement matrix of size 2×8 is generated by taking the first two rows of the S_8 matrix. In that case, it is observed that only $((2 \times 8) - 1)$ elements are required for generating the measurement matrix. The proposed sensing matrix requires only 15 elements to generate the measurement matrix of size $M_1 \times 8$ for any number of measurements, whereas the Gaussian measurement matrix requires $M_1 \times 8$ elements leading to storage overhead. The use of this proposed sensing matrix is a desirable alternative for WSN applications as it reduces the computational and storage complexity.

5 Performance evaluation

The CS framework with the proposed measurement matrix is simulated in MATLAB, and the measurements obtained from the framework are transmitted in real time using TelosB nodes. The performance of the proposed sensing matrix is compared with the Gaussian sensing matrix in terms of peak signal-to-noise ratio (PSNR), storage complexity, energy complexity, time complexity and end-to-end transmission latency.

Consider an image of size 128×128 for simulation and experimental evaluation. The image is divided into 256 blocks of size 8×8. Each block is converted into a single vector that is sparsified using DCT resulting in sparse vector. The sparsity level (i.e. number of nonzero elements in the sparse vector) is fixed at $K_1 = 10$ per block. From the sparsity level, the minimum number of measurements required to reconstruct the image is calculated using (4). As the number of measurements increases, the quality of the reconstructed image also increases. The proposed sensing matrix that is derived from the Toeplitz matrix and its variants is generated as follows:

1. Toeplitz matrix of size 64×64 is generated with Gaussian entries using (5).
2. Hankel matrix and circulant matrix of size 64×64 are derived from Toeplitz matrix.
3. Hankel matrix is added with the circulant matrix.
4. Subtract the resulting matrix from the Toeplitz matrix which results in a matrix of size 64×64.
5. Select M_1 rows from the final matrix which is used as the sensing matrix of size $M_1 \times 64$.

After designing the sensing matrix, multiply each sparse vector with the sensing matrix to obtain the measurement vector y_1 of size $M_1 \times 1$. Finally, OMP recovery algorithm is employed to estimate the sparse vector from the measurements. The original image is reconstructed by applying IDCT to the estimated sparse vector. The simulation parameters used in this section are provided in Table 2

5.1 Simulation analysis

The simulation is carried out with four standard images at 30, 40 and 50% sampling rates. The simulation is done

Table 2 Simulation and experimental parameters used in this paper

n	8
M_1 (%)	[20, 30, 40]
P	64
K_1	[3, 6, 10]
d	[1, 5, 10]
t	1/250 s
I	19.5 mA
V	3 V

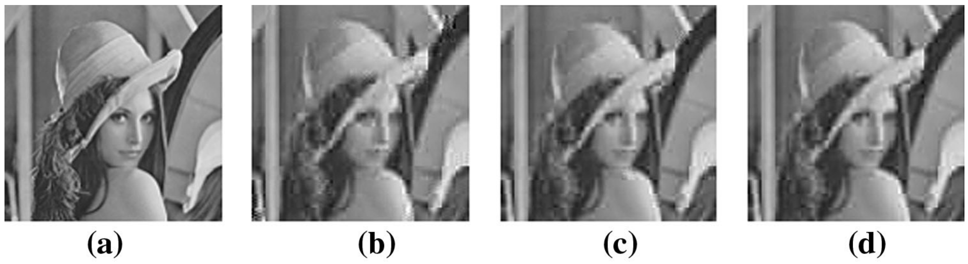


Fig. 3 **a** Original image, **b** reconstructed with sampling rate = 30%, **c** reconstructed with sampling rate = 40%, **d** reconstructed with sampling rate = 50%

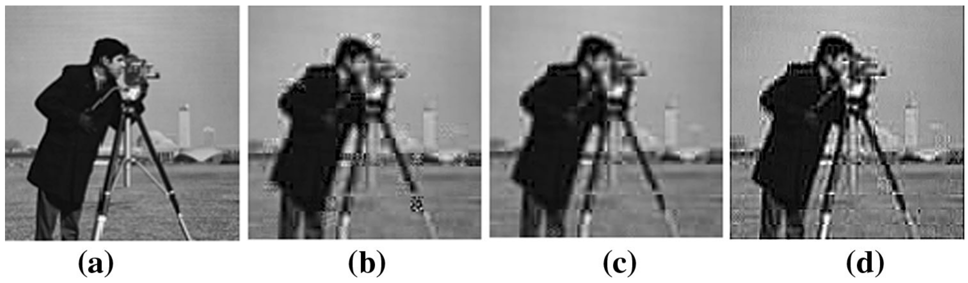


Fig. 4 **a** Original image, **b** reconstructed with sampling rate = 30%, **c** reconstructed with sampling rate = 40%, **d** reconstructed with sampling rate = 50%

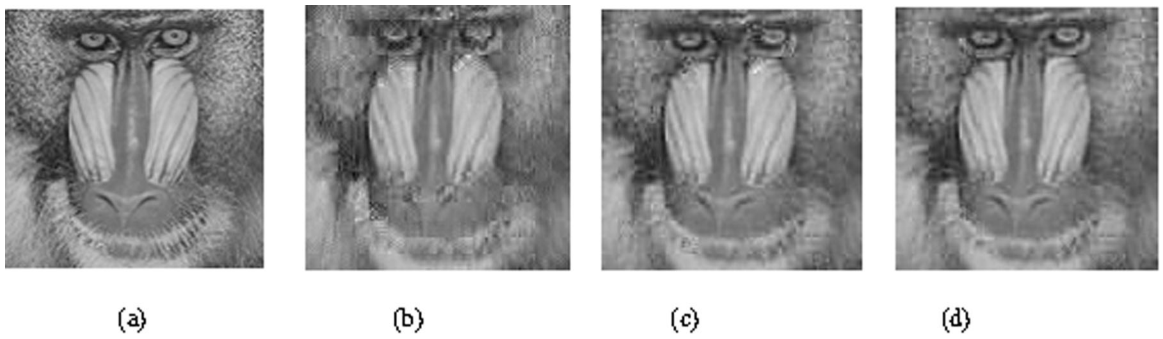


Fig. 5 **a** Original image, **b** reconstructed with sampling rate = 30%, **c** reconstructed with sampling rate = 40%, **d** reconstructed with sampling rate = 50%

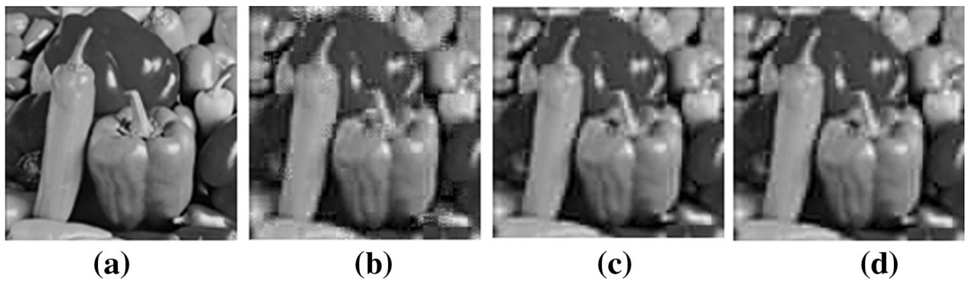


Fig. 6 **a** Original image, **b** reconstructed with sampling rate = 30%, **c** reconstructed with sampling rate = 40%, **d** reconstructed with sampling rate = 50%

using both the Gaussian sensing matrix and proposed sensing matrix. Figures 3, 4, 5 and 6 show the original and reconstructed image using the proposed sensing matrix at different sampling rates for Lena, Cameraman, Mandrill and Peppers images, respectively. It is observed that as the sampling rate increases the quality of the reconstructed image approaches the quality of the original image.

Table 3 shows the comparison between the Gaussian matrix and the proposed sensing matrix in terms of PSNR for different sampling rates. It is observed that as the sampling rate increases the PSNR also increases. The results also show that the proposed matrix yields better PSNR compared with the Gaussian matrix.

Table 4 depicts the comparative analysis of Gaussian sensing matrix and proposed sensing matrix in terms of PSNR by varying the sparsity level per block at $K_1 = 3, 6, 10$.

It is observed that the PSNR increases with increase in the sparsity level and the proposed matrix yields better PSNR compared with the Gaussian matrix. It is also observed that for a small number of measurements the proposed matrix

Table 3 Comparison between Gaussian sensing matrix and proposed sensing matrix for different sampling rates

Image (128 × 128)	Sample rate per block (%)	PSNR (dB)	
		Gaussian matrix	Proposed matrix
Lena	30	23.6	25.3
	40	24.6	26.2
	50	25.5	26.5
Cameraman	30	20.4	23.6
	40	22.2	24.8
	50	23.5	26
Mandrill	30	22.5	24.1
	40	24.1	26
	50	25.7	26.8
Peppers	30	21.5	24.2
	40	23.6	25.7
	50	25.9	26

Table 4 Comparison between Gaussian sensing matrix and proposed sensing matrix for different sparsity levels

Image (128 × 128)	Sparsity level per block (K_1)	Measurements (M)	PSNR (dB)	
			Gaussian matrix	Proposed matrix
Lena	3	3072	20.18	22.2
	6	6400	23.6	24.6
	10	10,752	25.8	26.5
Cameraman	3	3072	18.5	20.9
	6	6400	21.8	23.1
	10	10,752	24.3	24.8
Mandrill	3	3072	22	23.5
	6	6400	23.9	24.7
	10	10,752	25.5	25.9
Peppers	3	3072	19.5	21.9
	6	6400	22	23.9
	10	10,752	25.3	26

shows better PSNR than the Gaussian matrix. The proposed matrix shows better performance compared with the state-of-the-art methods such as IRC matrix, Toeplitz matrix, Bernoulli matrix and Kronecker matrix for $M/N = 0.4$ [28]. When the number of measurements approaches the total number of samples, the PSNR of the proposed sensing matrix becomes similar to the PSNR of the Gaussian matrix because the measurement matrix will have no effect on the quality of reconstruction. Since the proposed matrix yields better PSNR even for small measurements, the energy for transmitting the measurements gets reduced, increasing the lifetime of the network.

5.2 Experimental analysis

For experimental evaluation, the measurements obtained from the CS framework are transmitted from the source node to the sink node. These measurements are transmitted using TelosB nodes under Contiki OS platform. The measurements of each block are transmitted in a packet which consists of the sequence number of the block followed by the measurements.

5.2.1 Network model

The network is modelled using TelosB nodes and is operated under Contiki OS platform. For experimental evaluation, two types of deployment scenarios (i.e. line deployment and random deployment) are considered.

(a) Line deployment

In line deployment, the transmission is carried out in multihop manner. Figure 7 shows the network with line deployment in which the source and the sink nodes are

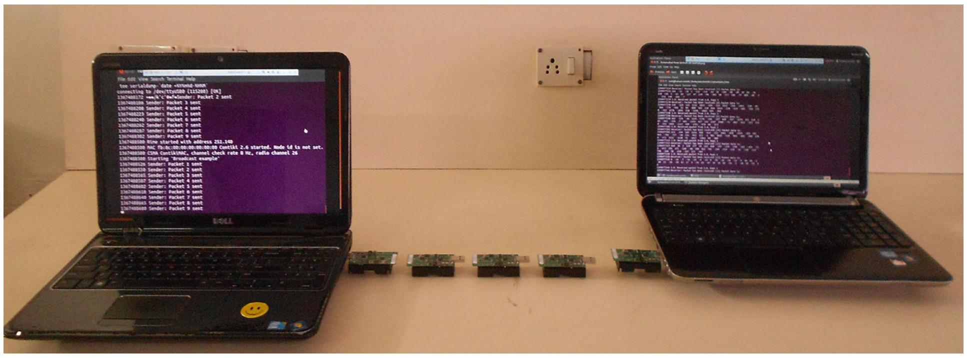


Fig. 7 Line deployment

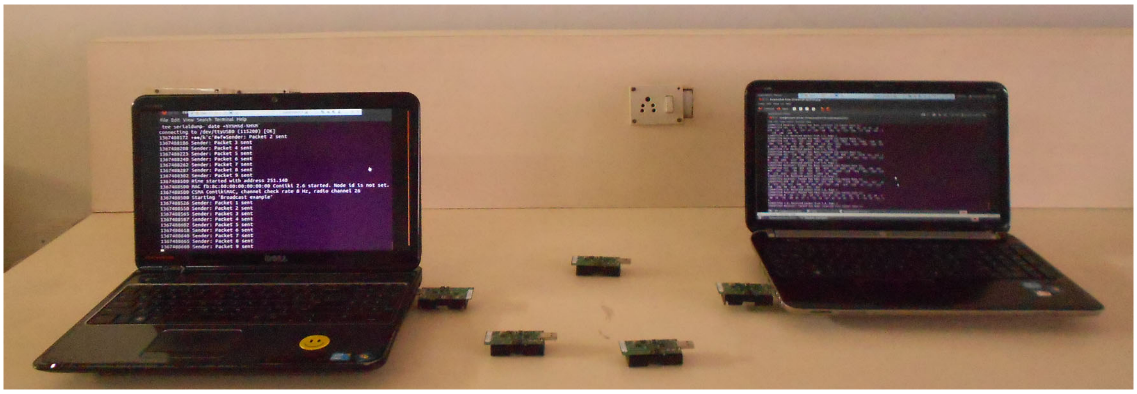


Fig. 8 Random deployment

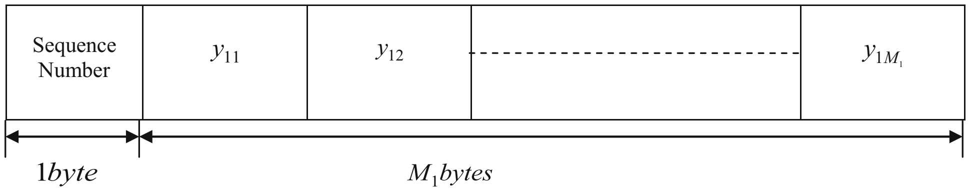


Fig. 9 Structure of the packet

connected to the PC. The node 1 acts as the source node, node 5 acts as the sink node, and nodes 2, 3 and 4 act as the relay nodes. The distance between the nodes is fixed at $d = 1, 5$ and 10 m. The indoor transmission range is 20–30 m, and outdoor transmission range is 75–100 m [4]. The CS process is carried out in the PC which will be replaced with the customized hardware in the near future. The packets of measurement obtained from the CS framework are transmitted from the source node to the sink node through the relay nodes. The measurements received by the sink node are viewed in the PC from which the original image can be reconstructed.

(b) Random deployment

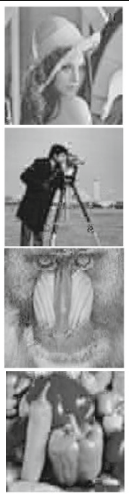
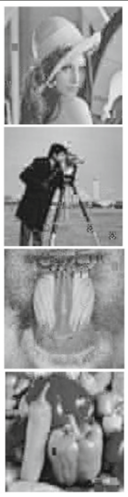
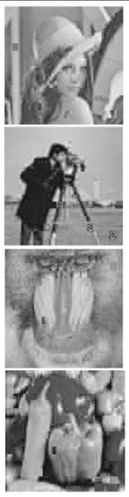
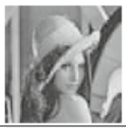
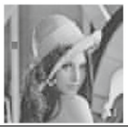
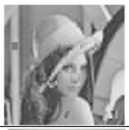
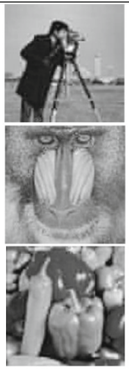
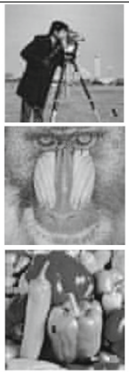
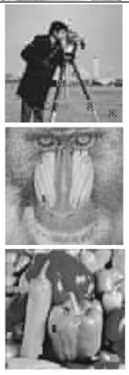
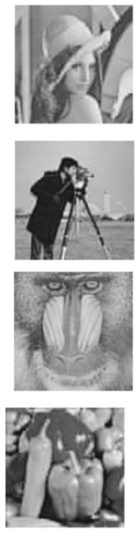
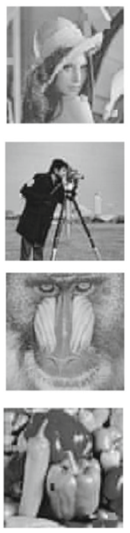
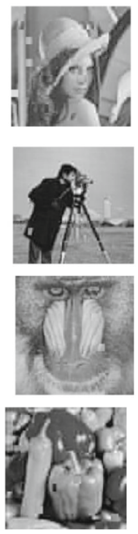
In random deployment scenario, the source node (node 1) and the sink node (node 5) are connected to the PC, and the relay nodes 2, 3 and 4 are deployed randomly as shown in Fig. 8. In this network model, the relay nodes will forward the data to a random neighbour node using multihop transmission until it reaches the sink node. The measurements are transmitted from the source node to the sink node and are displayed in the PC.

5.2.2 *Experimental results*

The measurements obtained for the sampling rate of 30, 40 and 50% are transmitted from the source node to the sink node using TelosB nodes. The experiments are carried out for both the line deployment and random deployment scenarios. The measurements are transmitted in a packet of size $1 + M_1$ bytes. The structure of the packet that is transmitted is shown in Fig. 9, where the first byte represents the sequence number of the packet and M_1 bytes of data represent the measurements $y_1 = \{y_{11}, y_{12}, y_{13}, \ldots, y_{1M_1}\}$ of the block.

These measurements are transmitted from the source node to the sink node for all 256 blocks, and the image is

Table 5 Comparison of experimental results with the simulation results for Lena, Cameraman and Mandrill

Sampling Rate (%)	Simulation results	Experimental results	
		Line Deployment	Random Deployment
30			
40			
			
50			

reconstructed using MATLAB. Table 5 shows the comparison of experimental results and simulation results for Lena, Cameraman, Mandrill and Peppers images, respectively, at different sampling rates.

Table 6 shows the comparison between experimental and simulation results in terms of PSNR for different sampling rates.

From the table, it is inferred that the PSNR is better for line deployment when compared with random deployment due to packet loss. Table 7 shows the comparison between line deployment and random deployment in terms of packet loss at different distances. It is inferred that as the distance increases the percentage of recovery decreases due to the increase in the packet loss. The percentage of packet loss is calculated using formula

$$\text{Packet loss } (\%) = \left(\sum \text{lostpackets} \Big/ \sum \text{packets}\right) \times 100$$

In line deployment, the nodes have a predefined path to reach the destination, whereas in the random deployment, the nodes select a random neighbour to reach the destination. Hence comparing both the deployment scenarios, it is observed that the packet loss for line deployment is less.

5.3 Complexity analysis

The complexity is analysed in terms of storage, energy and time for generating the measurement matrix. The energy computation for the overall CS framework, for Huffman encoding and for transmitting the measurements is also analysed.

5.3.1 Storage complexity

For generating a proposed sensing matrix of size $M_1 \times 64$, it requires only $((2 \times 64) - 1)$ elements, whereas for generating a Gaussian matrix of size $M_1 \times 64$ totally $M_1 \times 64$ elements are required to be stored. This shows that the proposed matrix requires less storage capacity and is irrespective of the size of the measurement matrix when compared with the Gaussian matrix. Table 8 shows the elements to be stored for the Gaussian matrix and the proposed matrix for different measurements. It is observed that in the case of the Gaussian sensing matrix, as the measurement increases the elements required in generating the matrix also increase whereas the proposed matrix requires only 127 elements which shows a reduction of more than 90% of the elements. Even though the proposed sensing matrix is a combination of Toeplitz matrix and its variants, it is sufficient to generate the Toeplitz matrix alone as the other two variants, i.e. Hankel and circulant matrices are derived from the Toeplitz matrix, thereby reducing the overall computational overhead.

Table 6 Comparison of experimental results with the simulation results for different sampling rates

Image (128 × 128)	Sampling rate (%)	PSNR (dB)		
		Simulation results	Experimental results	
			Line deployment	Random deployment
Lena	30	25.3	24.5	24.1
	40	26.2	25.1	24.8
	50	26.5	25.6	25.3
Cameraman	30	23.6	23.1	22.3
	40	24.8	24.2	23.6
	50	26	25.9	25.8
Mandrill	30	24.1	23.7	23.3
	40	26	25.4	24.8
	50	26.8	26.7	26
Peppers	30	24.2	23.4	23.1
	40	25.7	25.2	24.9
	50	25.9	25.7	25.3

Table 7 Comparison between line deployment and random deployment at different distances

Image (128 × 128)	Distance (m)	Line deployment		Random deployment	
		Packet loss (%)	Percentage of recovery (%)	Packet loss (%)	Percentage of recovery (%)
Lena	1	0	100	0	100
	5	1.2	98.8	2.3	97.7
	10	2.7	97.3	3.5	96.5
Cameraman	1	0	100	0	100
	5	0.4	99.6	1.2	98.8
	10	2	98	3	97
Mandrill	1	0	100	0	100
	5	0.4	99.6	1.6	98.4
	10	1.6	98.4	0.8	99.2
Peppers	1	0	100	0	100
	5	0.6	99.4	1.8	98.2
	10	1.9	98.1	2.6	97.4

5.3.2 *Energy complexity*

In this paper, energy complexity is analysed for computing the measurement matrix, CS framework, Huffman encoding and transmission of measurements.

(a) Energy computation for generating the measurement matrix

The energy consumed for generating the measurement matrix is computed using (8) and (12)

(i) Gaussian sensing matrix

To generate a Gaussian matrix of size $M_1 \times 64$, $M_1 \times 64$ read and write operations are required.

$$E_G = (M_1 \times 64) \times e_{\text{read}} + (M_1 \times 64) \times e_{\text{write}} \tag{8}$$

(ii) Proposed sensing matrix

To generate the Toeplitz matrix of size 64×64, 127 read and write operations are required. Energy consumed for generating the Toeplitz matrix is given in (9).

$$E_T = 127 \times e_{\text{read}} + 127 \times e_{\text{write}} \tag{9}$$

To derive the circulant matrix from the Toeplitz matrix, 4032 shift operations are required. Energy consumed for deriving the circulant matrix is given in (10).

$$E_C = 4032 \times e_{\text{shift}} \tag{10}$$

To derive the Hankel matrix from the Toeplitz matrix, 4096 shift operations are required. Energy consumed for deriving the Hankel matrix is given in (11).

Table 8 Comparison of Gaussian sensing matrix and proposed matrix in terms of elements to be stored

Size of measurement matrix	No. of elements to be stored (bytes)		Percentage of reduction of elements (%)
	Gaussian sensing matrix	Proposed sensing matrix	
20 × 64	1280	127	90
30 × 64	1920	127	93.4
40 × 64	2560	127	95

$$E_H = 4096 \times e_{\text{shift}} \tag{11}$$

The energy consumed for generating the proposed matrix is given in (12)

$$E_S = (E_T + E_C + E_H) + (4096 \times e_{\text{add}} + 4096 \times e_{\text{sub}}) \tag{12}$$

The energy consumption analysis is performed by using the characteristics of TelosB nodes. The energy consumed for one processing cycle in TelosB is 0.73 nJ. The addition, subtraction and shift operations can be performed in one cycle, and multiplication operation can be performed in four cycles. The energy consumed for reading one byte from flash is 8.2 μJ and writing one byte to flash is 34.9 μJ [29]. The energy consumption for the basic operations involved in the computation is provided in Table 9.

The energy consumed for generating the measurement matrix is computed by substituting the values given in Table 9 in (8) and (12). Energy consumption of the Gaussian matrix and the proposed sensing matrix for $M_1 = 20, 30, 40$ is given in Table 10.

Table 10 clearly shows that the energy consumed for generating the proposed matrix is 90% less when compared with the existing Gaussian measurement matrix and does not depend on the size of the measurement matrix.

(b) Energy computation for the overall CS framework and Huffman encoding

Energy computation for the overall CS framework is the sum of the energy consumed for all the processes involved in CS.

Table 9 Energy consumed by the MSP430 microcontroller

Variable	Operation performed	Energy consumed
e_{add}	Addition over 1 byte	0.73 nJ
e_{sub}	Subtraction over 1 byte	0.73 nJ
e_{mul}	Multiplication over 1 byte	2.92 nJ
e_{shift}	Shift over 1 byte	0.73 nJ
e_{read}	Reads 1 byte from the flash memory	8.2 μJ
e_{write}	Writes 1 byte to the flash memory	34.9 μJ
e_{ent}	Encodes 1 byte	160 nJ

1. DCT energy computation

Two-dimensional DCT of $k \times k$ matrix needs two $k \times k$ matrix multiplications. Each matrix product $k \times k$ *needs* k^2 coefficients to be computed, each coefficient necessitating k multiplications and $(k - 1)$ additions. Therefore, the energy dissipated for the two matrix products is $2k^2(k\ e_{\text{mult}} + (k-1)\ e_{\text{add}})$, where e_{mult} and e_{add} represent the energy consumption for multiplication and addition instructions, respectively. The energy consumption for DCT computation is given in (13) where $k = 8$, $N_1 = n^2 = 64$ and e_{read} and e_{write} represent the energy consumption for reading and writing a byte in the flash memory, respectively.

$$E_{\text{dct}} = N_1 \times e_{\text{read}} + 2N_1((k-1) \times e_{\text{add}} + k \times e_{\text{mult}}) + N_1 \times e_{\text{write}} \tag{13}$$

2. Energy computation for CS measurements

The sparse vector is multiplied by the measurement matrix to get the measurement vector. The size of the measurement matrix is decided based on the minimum number of measurements required for reconstruction. The measurement matrix of size $M_1 \times 64$ is used for obtaining M_1 measurements. The energy related to the measurement process is given in (14).

$$E_{\text{csm}} = ((N_1 - 1) \times e_{\text{add}} + N_1 \times e_{\text{mult}}) \times M_1 \tag{14}$$

3. Overall energy consumption

The overall energy consumption of the CS framework is given as

$$E_{\text{tot}} = ((E_{\text{dct}} + E_{\text{csm}}) \times B_N) \tag{15}$$

where B_N represents the number of blocks in the image. The energy consumed for the overall CS framework with measurement matrix of size 20 × 64 is computed by substituting the values from Table 9 in (15) as 0.71 J.

4. Encoding energy

The encoding scheme used is Huffman coding. The energy is given by the number of bytes of information multiplied by the e_{ent} energy for entropy encoding per byte.

Table 10 Energy consumed for generating the measurement matrix

Size of measurement matrix	Energy consumed (mJ)		Percentage of reduction in energy (%)
	Gaussian matrix	Proposed matrix	
20×64	55	5.5	90
30×64	83	5.5	93.4
40×64	110	5.5	95

Table 11 Energy per bit transmit in TelosB mote

S. No.	Mote	Theoretical value of energy for 1-bit transmit (μJ)	Practical value of energy for 1-bit transmit (μJ)
1.	TelosB	0.23	0.27

$$E_{\text{enc}} = \text{no. of bytes} \times e_{\text{ent}} \tag{16}$$

The energy consumed for the encoding process is computed by substituting the values from Table 9 in (16) as 0.8 mJ.

(c) Transmission energy

In the case of multimedia applications, the data to be transmitted in the network are large which can be reduced significantly by employing CS in the network that requires only the measurements to be transmitted in the network, thereby reducing the transmission energy. The transmission energy per bit is analysed theoretically by considering the current, voltage specifications of the TelosB mote and the time to transmit a 128-byte packet in a 250-kbit/s IEEE 802.15.4 link using Eq. (17)

$$E_{\text{tx}} = (t \times I \times V)/1024 \text{ J} \tag{17}$$

where t represents the time to transmit a 128-byte packet which is approximately 1/250 s, I represents the current consumption which is 19.5 mA taken from the TelosB datasheet and V represents the voltage which is 3 V [30]. The time taken to transmit a 128-byte packet in real time is computed using the Energest component given in power-trace tool in Contiki OS [31]. The current consumed for transmitting a packet is measured using an oscilloscope by setting the voltage at 3 V. The transmission energy per bit is computed practically as

$$Em_{\text{tx}} = (t_m \times I_m \times V)/1024 \text{ J} \tag{18}$$

where t_m represents time that is measured to be 4.6 ms, I_m represents the current that is measured to be 20 mA and V represents the voltage which is 3 V. The theoretical and practical value of the energy for one-bit transmission is given in Table 11, and the energy comparison for different measurements is given in Table 12.

The transmission energy with and without encoding for different measurements is computed using Table 11, and Table 12 shows that as the number of measurements increases the transmission energy also increases. The transmission energy can be reduced further by encoding the measurements.

The encoding technique employed is Huffman encoding which is a lossless coding technique, and hence, it does not affect the quality of the reconstructed image. The bits used to represent the measurements are reduced after encoding which in turn reduces the transmission energy. The number of bits to represent the measurement after encoding and its corresponding transmission energy is also given in Table 12. It is inferred that the energy consumed for transmission of measurements after encoding is reduced to 40% of the energy consumed for transmission before encoding.

5.3.3 *Time complexity*

The computational time for generating the measurement matrix is calculated using (19) and (20). The time consumed for computing the Gaussian matrix of size $M_1 \times 64$ is given as

$$G_T = (M_1 \times 64) \times t_{\text{read}} + (M_1 \times 64) \times t_{\text{write}} \tag{19}$$

The time consumed for computing the proposed matrix of size $M_1 \times 64$ is given as

$$S_T = ((127 \times t_{\text{read}} + 127 \times t_{\text{write}}) + (4032 \times t_{\text{shift}} + 4096 \times t_{\text{shift}})) + (4096 \times t_{\text{add}} + 4096 \times t_{\text{sub}}) \tag{20}$$

The time consumed for one processing cycle in TelosB node is 1 μs. The time consumed for reading one byte from flash is 2.69 ms and for writing one byte to flash is 4.49 ms [29]. The computational time for generating the matrix is calculated by using the values given in Table 13. Table 14 shows the computational time for both the Gaussian measurement matrix and the proposed sensing matrix.

Table 14 shows that the computational time for the proposed matrix is very less when compared with the

Table 12 Transmission energy with and without encoding

Number of measurements	Transmission energy (mJ)					
	Without encoding			With encoding		
	Bits per measurement	Theoretical values	Practical values	Bits per measurement	Theoretical values	Practical values
5120	8	9.4	11	4.4	5.2	6
6400	8	11.8	13.8	4.72	6.95	8.2
7680	8	14.1	16.6	4.93	8.71	10.22
8960	8	16.5	19.4	4.97	10.2	12

Table 13 Time consumed by the MSP430 microcontroller for different operations

Variable	Operation performed	Time consumed
t_{add}	Addition over 1 byte	1 µs
t_{sub}	Subtraction over 1 byte	1 µs
t_{shift}	Shift over 1 byte	1 µs
t_{read}	Reads 1 byte from the flash memory	2.69 ms
t_{write}	Writes 1 byte to the flash memory	4.49 ms

Table 14 Time consumed for generating the measurement matrix

Size of the measurement matrix	Computational time (seconds)	
	Gaussian matrix	Proposed matrix
20×64	9	0.93
30×64	14	0.93
40×64	18	0.93

Gaussian matrix. These observations clearly show that the quality of the image can be further enhanced by increasing the measurements without increasing the storage, energy and time complexity.

5.4 End-to-end transmission latency

The end-to-end transmission latency in the network is calculated by computing the time taken to transmit the measurements and propagation delay of the link. The transmission delay is obtained from the powertrace tool in Contiki OS platform [31]. In WSN, the sensor nodes are only short distance apart and hence the propagation delay is negligible. The end-to-end transmission latency ($d_{\text{end-end}}$) for a single hop is calculated using (21)

$$d_{\text{end-end}} = d_{\text{trans}} + d_{\text{prop}} \tag{21}$$

where d_{trans} represents the transmission delay and d_{prop} represents the propagation delay. In the case of multihop network, the end-to-end delay in the network is the sum of the delay of each link

Table 15 End-to-end transmission latency for different measurements

Measurements	End-to-end transmission latency (seconds)
5120	0.8
7680	1.2
10,240	1.6

$$\text{End-to-end transmission latency in multihop network} = h \times d_{\text{end-end}}$$

where h represents the number of hops between source node and destination node. Table 15 shows the end-to-end transmission latency for different measurements with distance between nodes = 5 m.

The end-to-end transmission latency increases with the increase in the number of measurements transmitted in the network. From the above results, it is observed that the proposed sensing matrix is efficient compared with the Gaussian matrix as the storage requirement, energy consumption and computational time for the proposed matrix are independent of the size of the matrix and it is also clear that the transmission energy is reduced significantly while maintaining the acceptable range of PSNR.

6 Conclusions and future work

A memory and energy-efficient sensing matrix is proposed for the CS technique to overcome the storage, energy and time complexity of the existing Gaussian sensing matrix. The simulation results of the proposed framework show that it yields better PSNR even for fewer measurements when compared with the Gaussian measurement matrix. In this work, an experimental evaluation is done by transmitting the measurements from the source node to the sink node using TelosB nodes under Contiki OS platform. The network is formed by using the TelosB nodes for the line deployment scenario and random deployment scenario.

The results show that in line deployment, less number of packets is lost yielding better PSNR when compared with the random deployment. This proposed matrix is best suitable for WSN, as it yields better PSNR for a small number of measurements and the memory, energy and time for generating the matrix are also less that prolong the lifetime of the network. The results also show that the storage, energy computation and computational time for the proposed matrix are irrespective of the size of the measurement matrix when compared with the Gaussian matrix. The results also show that the proposed sensing matrix reduces the transmission energy significantly for image transmission while maintaining the acceptable range of PSNR. It is seen that the transmission energy can be reduced further by encoding the measurements with the help of a lossless encoding technique.

In the near future, the PC used at the transmitting end will be replaced with the customized hardware in which the CS technique can be incorporated and the measurements alone can be transmitted using TelosB nodes. The customized hardware will have a camera module to capture the scene, high processor and large memory to implement the CS technique. In order to capture and process only the scene of interest or to detect anomalies, the camera module must be associated with a PIR sensor which triggers the camera. The future work aims in designing the customized hardware setup that incorporates the CS technique, which can be used for surveillance applications. An efficient encoding technique can also be used to further reduce the bandwidth and transmission energy by reducing the bit rate of the measurements for multimedia applications.

References

1. Akyildiz, I.F., Su, Z., SankaraSubramaniyam, Y., Cayirei, E.: A survey on sensor networks. IEEE Commun. Mag. **40**, 102–114 (2002)
2. Hussain, S.A., Razzak, M.I., Mirhas, A.A., Sher, M.A., Tahir, G.R.: Energy efficient image compression in wireless sensor networks. Int. J. Recent Trends Eng. **2**(1), 2–5 (2009)
3. Candes, E.J.: Compressive sampling. In: Proceedings of the International Congress of Mathematicians, Madrid, Spain, European Mathematical Society (2006)
4. 'TelosB'. http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf
5. 'ContikiOS'. http://www.contiki-os.org
6. Madni, A.M.: A systems perspective on compressed sensing and its use in reconstructing sparse networks. IEEE Syst. J. **8**(1), 23–27 (2014)
7. Zhang, J., Xiang, Q., Yin, Y., Chen, C., Luo, X.: Adaptive compressed sensing for wireless image sensor networks. Multimed. Tools Appl. 1–16 (2016). doi:10.1007/s11042-016-3496-x
8. Talari, A., Rahnavard, N.: CStorage: distributed data storage in wireless sensor networks employing compressive sensing. In: IEEE Global Telecommunications Conference (GLOBECOM), Dec. 2011, pp. 5–9 (2011)
9. Liu, Y., Zhu, X., Ma, C., Zhang, L.: Multiple event detection in wireless sensor networks using Compressed Sensing. In: 18th International Conference on Telecommunication, Ayia Napa, 8–11 May (2011)
10. Gan, L.: Block compressed sensing of natural images. In: 15th International Conference on Digital Signal Processing, 1–4 July, 2007, pp. 403–406 (2007)
11. Kim, S.-J., Koh, K., Lustig, M., Boyd, S.: An efficient method for compressed sensing. In: IEEE International Conference on Image Processing (ICIP), vol. 3, Sept. 16 2007–Oct. 19 2007, pp. III-117–III-120 (2007)
12. Sermwuthisarn, P., Aurthavekint, S., Patanavijit, V.: A fast image recovery using compressive sensing technique with block based orthogonal matching pursuit. In: Proceedings of Intelligent Signal Processing and Communication Systems, Kanazawa, 7–9 Jan. 2009 (2009)
13. Shahidan, A.A., Fisal, N., Fikri, A.H., Ismail, N.S.N., Farizahyances: Image transfer in wireless sensor networks. In: International Conference on Communication Engineering and Networks, IPCSIT, vol. 19, Singapore (2011)
14. Zhou, C., Xiong, C., Mao, R., Gong, J.: Compressed sensing of images using nonuniform sampling. In: Intelligent Computation Technology and Automation (ICICTA), 28–29 March, 2011, pp. 483–486 (2011)
15. Li, K., Ling, C., Gan, L.: Statistical restricted isometry property of orthogonal symmetric Toeplitz matrices. In: IEEE Information Theory Workshop (ITW), 11–16 Oct. 2009, pp. 183–187 (2009)
16. Yin, W., Morgan, S., Yang, J., Zhang, Y.: Practical compressive sensing with Toeplitz and circulant matrices. In: Proceedings of the SPIE7744, Visual Communications and Image Processing, 7744K (2010)
17. Yu, L., Barbot, J.P., Zheng, G., Sun, H.: Toeplitz-structured chaotic sensing matrix for compressive sensing. In: 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP), 21–23 July, 2010, pp. 229–233 (2010)
18. Bajwa, W.U., Haupt, J.D., Raz, Gil M, Wright, S.J., Nowak, R.D.: Toeplitz-structured compressed sensing matrices. In: IEEE/SP 14th Workshop on Statistical Signal Processing (SSP), 26–29 Aug. 2007, pp. 294, 298 (2007)
19. Shen, Y., Hu, W., Rana, R., Chou, C.T.: Non-uniform compressive sensing for heterogeneous wireless sensor networks. IEEE Sens. J. **13**(6), 2120–2128 (2013)
20. Karakus, C., Gurbuz, A.C., Tavli, B.: Analysis of energy efficiency of compressive sensing in wireless sensor networks. IEEE Sens. J. **13**(5), 1999–2008 (2013). doi:10.1109/JSEN.2013.2244036
21. Han, B., Wu, F., Wu, D.: Image representation by compressive sensing for visual sensor networks. J. Vis. Commun. Image Represent. **21**(4), 325–333 (2010)
22. Hemalatha, R., Radha, S., Sudharsan, S.: Energy-efficient image transmission in wireless multimedia sensor networks using block-based compressive sensing. Comput. Electr. Eng. **44**, 67–79 (2015)
23. Eleyan, A., Kose, K., Cetin, A.E.: Image feature extraction using compressive sensing. In: Image Processing and Communications Challenges, vol. 5, pp. 177–184. Springer International Publishing (2014)

24 Yao, S., Wang, T., Shen, W., Pan, S., Chong. Y.: The chaotic measurement matrix for compressed sensing. In: International Conference on Intelligent Computing, pp. 58–64. Springer International Publishing (2015)

25. Xie, D., Haipeng P., Lixiang L., Yixian Y.: Semi-tensor compressed sensing. Digit. Signal Process. **58**, 85–92 (2016)

26. Huang, A.M., Gui, G., Wan, Q., Mehbodniya, A.: A block orthogonal matching pursuit algorithm based on Sensing dictionary. Int. J. Phys. Sci. **6**(5), 992–999 (2011)
27. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory **53**, 4655–4666 (2007)
28. Gui, G., Mehbodniya, A., Wan, Q., Adachi, F.: Sparse signal recovery with OMP algorithm using sensing measurement matrix. IEICE Electron. Expr **8**(5), 285–290 (2011)
29. Amiri, M.: Measurements of energy consumption and execution time of different operations on Tmote Sky sensor nodes (2010)
30. http://www.ti.com/product/msp430f1611
31. Dunkels, A., Eriksson, J., Finne, N., Tsiftes, N.: Powertrace: Network-Level Power Profiling for Lowpower Wireless Networks. Technical Report T2011:05, SICS (2011)

S. Aasha Nandhini is a full-time Research Scholar at SSN College of Engineering, India. She received the B.E. degree in Electronics and Communication Engineering from Rajalakshmi Engineering College, India, in 2010 and the M.E. degree in Communication Systems from SSN College of Engineering, India, in 2012. Her research interests include security issues and compressive sensing in wireless sensor networks.

S. Radha graduated from Madurai Kamaraj University in Electronics and Communication Engineering during the year 1989. She obtained her master degree in Applied Electronics with First Rank from Government College of Technology, Coimbatore, and Ph.D. degree from College of Engineering, Guindy, Anna University, Chennai. She has 22 years of teaching and 11 years of research experience in the area of mobile ad hoc networks. She also worked as a Visiting Researcher at Carnegie Mellon University for a period of six months with Prof. Rajkumar in the area of wireless sensor networks. She has 92 publications in international and national journals and conferences in the area of mobile ad hoc network and wireless sensor networks. Her current areas of research include security and architecture issues of mobile ad hoc networks and wireless sensor networks and cognitive radios. She is a Member of IEEE, Fellow of IETE and Life member of ISTE. She has received IETE—SK Mitra Memorial Award in October 2006 from IETE Council of India, Best Paper Awards in various conferences and CTS-SSN Best Faculty Award (2007 and 2009) for the outstanding performance. Currently, she is a Principal Investigator for sponsored projects worth of Rs. 62 lakhs which is sponsored by DST, INTEL Bangalore SSN.

P. Nirmala is a Junior Research Fellow at SSN College of Engineering, India. She received the M.E. degree in Communication Systems from SSN College of Engineering, India, in 2013. Her research interests include compressive sensing in wireless sensor networks and multisensor fusion.

R. Kishore graduated from Vellore Engineering College, Madras University, in Electronics and Communication Engineering during the year 1998. He obtained his master degree in Communication Systems from Pondicherry Engineering College, Pondicherry University, Pondicherry, during the year 2003 and Ph.D. degree from Anna University during the year 2012. At present, he is working as Associate Professor in the Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, Tamil Nadu, India. His research interests include security issues in wireless networks, compressive sensing and image fusion.