UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

AN ENERGY-EFFICIENT SENSING MATRIX FOR WIRELESS MULTIMEDIA SENSOR NETWORKS

by

Vusi Josias Skosana

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

August 2023

SUMMARY

AN ENERGY-EFFICIENT SENSING MATRIX FOR WIRELESS MULTIMEDIA SENSOR NETWORKS

by

Vusi Josias Skosana

Supervisor:	Prof. A.M. Abu-Mahfouz
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Electronic Engineering)
Keywords:	chaotic sequences, compressive sensing, energy efficiency, image quality, partial canonical identity matrix, sensing matrix, wireless multimedia sensor networks, wireless sensor networks

A Wireless Multimedia Sensor Network (WMSN) make possible new surveillance applications in environments that traditional systems would not handle, including search and rescue operations after a disaster. However, WMSNs ought to perform under energy-constrained conditions that insist on novel compression methods to diminish bandwidth usage and extend network lifespan.

Compressed Sensing (CS) was presented as a means to achieve overcome the challenges faced by WMSNs. A sensing matrix is crucial to the compressed sensing framework. The sensing matrix can maintain the fidelity of a compressed signal, diminish the sampling rate obligation and improve the strength and performance of the recovery algorithm. A great number of measurement matrices have been proposed to either offer reduced computational complexity or good recovery performance, but only some have managed to accomplish both, and even fewer have been proven in a compelling manner. There are images that do not lend themselves to compression, and to maintain Quality of Service (QoS) expectations, adaptive sampling is essential.

Low-performance nodes are essential for making WMSN practical and flexible. Different low-performance nodes have been proposed in the literature, but the Telos Revision B (TelosB) sensor module (mote) can be used as a reference for energy-constrained applications. TelosB is a very low power wireless mote for research and experimentation. The design of sensing matrices has been influenced by practical considerations in WSN. The two major innovations have replaced floating point numbers with bipolar and binary entries and sparse sensing matrices.

The Deterministic Partial Canonical Identity (DPCI) matrix was presented to address the needs of an energy-constrained environment for WMSN. The choices of random number generators were discussed, and criteria were developed for selection. Complexity optimisation was undertaken to improve the time complexity of the construction. The DPCI was outperformed by the Deterministic Binary Block Diagonal (DBBD) and Binary Permuted Block Diagonal (BPBD) in terms of recovery performance but gave a substantial computational cost reduction. The DPCI gives a compelling balance between recovery performance and energy efficiency, benefiting energy-sensitive applications. A recovery performance prediction algorithm was also proposed to be used for an adaptive sampling scheme.

LIST OF ABBREVIATIONS

1D	one dimensional
2D	two dimensional
BCS	Block compressed sensing
BCS-SPL	Block-based CS sampling and smoothed projected Landweber
BP	Basis pursuit
BPBD	Binary permuted block diagonal
CIF	Common intermediate format
CMOS	Complementary metal-oxide-semiconductor
CoSaMP	Compressive Sampling Matching Pursuits
CPU	Central processing unit
CS	Compressed sensing
dB	Decibels
DBBD	Deterministic binary block diagonal
DCT	Discrete cosine transform
DWT	Discrete wavelet transform
DTT	Discrete Tchebichef transform
FFD	Full-function device
FYS	Fisher-Yates shuffle
IRC	Incoherent rotated chaotic
ITMM	Improved Toeplitz measurement matrix
LCG	Linear congruent generator
LR-WPAN	Low-rate wireless personal area network
MCU	Micro-controller unit
MMC	MultiMediaCard
MP	Matching Pursuit
MSE	Mean square error
OMP	Orthogonal Matching Pursuit
PAN	Personal area network
PC	Personal computer
PCI	Partial canonical identity

PSNR	Peak signal to noise ratio
RAM	Random access memory
RF	Radio frequency
RFD	Reduced-function device
RIP	Restricted isometry property
ROMP	Regularized Orthogonal Matching Pursuit
SBCM	Sparse block circulant matrix
SNR	Signal to noise ratio
SRM	Structurally random matrix
SSIM	Structurally similarity index
StOMP	Stagewise Orthogonal Matching Pursuit
SIMD	Single instruction multiple data
QoS	Quality-of-service
WMSN	Wireless multimedia sensor nodes
WSN	Wirelss sensor nodes
VGA	Video graphics array

LIST OF SYMBOLS

E	energy
f	frequency
I	current
t	time
V	voltage
ε	computational complexity
Φ	sensing matrix
Ψ	sparsity basis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

A Wireless Multimedia Sensor Network (WMSN) is an *ad hoc* ensemble of embedded and battery-powered optical sensor devices. This ensemble consists of a large number of sensor nodes placed around an area to be surveyed. Depending on the topology, there is at least one data sink located at the centre of the nodes [2]. The robustness and flexibility of these systems make possible new surveillance applications in environments that do not have the infrastructure to support traditional equipment. These environments can also be dynamic and hazardous; these include search and rescue operations after a disaster.

These WMSNs have to perform in energy-constrained conditions that compel novel compression schemes to diminish the bandwidth requirement and extend network lifespan [3]. Multi-hop routing can be used in the network link between optical nodes and their sinks, necessitating a high compression ratio to reduce transmission energy consumption.

The WMSN encounters unique challenges compared to the traditional Wireless Sensor Network (WSN). This network deals with much larger data packs and multidimensional signals, which have cross-dimensional dependencies and distortions. The other challenge is that WMSN captures data that can be perceived by humans, which must meet a defined quality-of-service (QoS) [2].

Energy conservation is one of three energy management schemes relied upon in WSN; the others take the form of energy transfer and energy harvesting [4]. Despite this, the large data transfers characteristic of WMSN elevate the importance of energy conservation over the others. Energy conservation consists of computational complexity reduction, power management optimisation and power efficiency improvements. This study will focus on reducing the complexity of optical sensor nodes because they make up the majority of the network.

Compressed sensing (CS) was inaugurated by Pudlewski *et al.* [5] as an instrument to tackle the challenges facing WMSNs. The CS method is fitting for WMSN as a consequence of its low encoder complexity, high compression rate and resistance to transmission errors [6].

Even though there have been other proposals for complexity reduction, amongst others, sparsity transforms optimisation [6, 7], these approaches suffer from an intractable compromise in recovery accuracy [8]. However, sensing matrix design has shown that this compromise can be avoided [9, 10, 11, 12, 13, 14].

Initially, during compressed sensing, each measurement was a mapping of an image onto an unrepeated matrix [15]. This imposed an impossible memory footprint when the signal was substantial in size, as encountered with high-resolution photos. In [16], Gan solved this problem by introducing block-compressed sensing (BCS), which splits the image into different blocks. The blocks are then compressed using a repeated measurement vector, which diminishes the dimensions of the required measurement matrix and the resulting memory footprint to a great extent.

In [17], Bajwa *et al.* reduced sensing and storage complexity at a meaningful scale by showing the efficacy of Toeplitz and circulant sensing matrices. These matrices need random entries exclusively for the first row, with the other rows generated using transformations. These are named semi-deterministic matrices, and those without random entries are fully deterministic.

Some researchers have been investigating optimisation-based matrices that use the mutual coherence with the sparsity transform as a cost function [18, 19, 20, 21, 22, 23] but these matrices can not be constructed dynamically. More recent research has been on training the sensing matrix based on a signal dataset [24, 25, 26], but in the same vein, this is not suitable for WMSN nodes.

The application of CS on images still has challenges. However, minimising computational complexity and sampling rate are the most important [27]. In [28], different types of sensing matrices were contrasted under various metrics typical in CS. They conducted their study through experiments on one-dimensional (1D) signals on a system that uses an advanced central processing unit (CPU). Despite this, the duration measurements (and likely energy consumption) are not analogous to WMSN, where low-power microcontroller units (MCUs) are popular.

Deterministic sensing matrices have performed better than dense ones in many studies [9, 10, 11, 12, 13, 14, 29, 30, 31, 32]. Semi-deterministic matrices are being succeeded in popularity by fully deterministic ones, especially matrices based on chaotic sequences [10, 13, 31].

1.1 PROBLEM STATEMENT

The problem that this study addresses is the compression of multimedia in WMSN using low energy with high fidelity. These WMSN will play a significant role in future mission-critical surveillance applications, such as search and rescue. An energy-efficient and high fidelity compression algorithm is a key enabler for wider adoption of WMSN technology.

1.1.1 Context of the problem

The WMSNs are the key to solving many real-world problems that the existing surveillance systems can not solve. The environmental and operational constraints on WMSNs require energy efficiency and high fidelity. This means that developing innovative data compression to diminish the transmission bandwidth utilisation and computational complexity of operations is essential. Compressive sensing has been identified as the best tool to address these challenges. However, work must be done to identify areas of improvement to get closer to the realisation of WMSNs in the operating environment.

1.1.2 Research gap

Although plenty of proposed sensing matrices aim to achieve low computational complexity or high fidelity, only some have addressed both objectives, and even fewer have been convincingly demonstrated. Energy efficiency and recovery performance appear to require conflicting methods. However, there have been innovative schemes in deterministic sensing matrix design that have given encouraging outcomes. These schemes must be identified, analysed and contrasted. The most viable approach must be exploited as a foundation for a new sensing matrix that offers low complexity and high recovery performance.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The proceeding section enumerates the questions that must be answered in the course of conducting this study. The objectives represent the goalposts to direct the execution of the research approach.

1.2.1 Research Questions

This study aims to respond the following questions in order to achieve its objectives.

1. **Which deterministic measurement matrices deliver the highest level of recovery performance**

While comparing fully deterministic and semi-deterministic measurement matrices, which show the highest level of recovery performance? The sampling rate is an essential constraint for a WMSN, which makes it crucial for the measurement matrices to perform well at these low sampling rates.

2. **What is the energy efficiency of deterministic measurement matrices**

   In which manner does computational complexity relate to energy efficiency? Is it that fully deterministic measurement matrices are more energy-efficient than random matrices, and which element of the compressed sensing framework is the most energy-intensive? In order to give direction to future research in WSMNs, it is essential to find out what are the energy properties of both semi-deterministic and fully deterministic measurement matrices and which attributes of their implementation require prioritisation.

3. **How does block size influence energy efficiency**

   What effect does the block size have on the energy efficiency and recovery performance of fully deterministic and semi-deterministic measurement matrices? The block size appears to influence the recovery accuracy for the majority of measurement matrices, which seems to be that a small block size gives rise to worse recovery performance. It must be investigated what this block size translates to in energy efficiency and where the optimal balance between recovery performance and energy efficiency lies.

4. **What are the most effective design tactics for creating measurement matrices**

   What is the most effective tactic for enhancing energy efficiency while avoiding significantly compromising recovery performance? Can recovery performance be enhanced while avoiding adversely affecting energy efficiency? How much can energy efficiency be strengthened before it results in a drastic reduction in recovery performance?

1.3 APPROACH

In order to answer the research questions and achieve the research objectives meaningfully, a deliberate strategy was devised. This strategy was based on the preliminary investigation of the literature. The strategy is outlined in the following steps below:

1. Undertake a literature review on energy-efficient measurement matrices for use in WMSNs.

2. Perform experiments with greyscale images to assess recovery performance and quantify the impact of block size on recovery performance.
3. Analyse the computational complexity of the measurement matrices through mathematical means and establish the connection between the energy consumption and computational complexity on one or more WSN nodes. Identify the elements that consume the most energy in the implementation and investigate how block size can affect energy efficiency.
4. Resolve which measurement matrix and block size offer the optimal balance between energy efficiency and recovery performance.
5. Make sense of the results and validate them using established theories.
6. Propose a new sensing matrix based on the most viable approach.

1.4 RESEARCH GOALS

- The proposed work will first guide future research by analysing and comparing different deterministic sensing matrices for applications in WMSNs.
- The second goal is interpreting and validating the empirical results of the experiments.
- The last goal is to create a sensing matrix that is informed by the theoretical analysis of the outcomes.

1.5 RESEARCH CONTRIBUTION

- We proposed a sparse and simple sensing matrix that has competitive recovery performance.
- We proposed a simple but accurate measure to predict image compression performance for application in adaptive sampling.

1.6 RESEARCH OUTPUTS

Skosana, Vusi, and Adnan M. Abu-Mahfouz. "Energy-Efficient Sensing Matrices for Wireless Multimedia Sensor Networks: A Review." In *2021 International Conference on Forthcoming Networks and Sustainability in AIoT Era (FoNeS-AIoT)*, pp. 51-56. IEEE, 2021.

Skosana, Vusi, and Adnan Abu-Mahfouz. "An Energy-Efficient Sensing Matrix for Wireless Multimedia Sensor Networks." *Sensors* 23, no. 10 (2023): 4843.

1.7 OVERVIEW OF STUDY

The Chapter 2 lays out the review of the WMSN applications and designs, energy efficiency measurement and optimisation and finally, compressive sensing challenges and opportunities. Chapter 3

discusses the proposed sensing matrix and experiments designed to validate the performance. Chapter 4 lists the quantitative and qualitative results of the experimental evaluation. In Chapter 5, the results are discussed, and the trends are generalised. In Chapter 6 the outcomes of the study are concluded, and future work is suggested.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

Choosing a WMSN node is challenging as the node must have sufficient image resolution, computational power, lifetime and cost-effectiveness. These considerations are weighed by the application priorities, and a balance must be struck. Compressive sensing is a powerful tool for WMSN. Still, it has various aspects that need to be understood, and the strengths of the framework should be effectively exploited while diminishing the effects of the weaknesses.

In Section 2.2, the WMSN nodes are discussed, and their applications and design considerations are covered. The most relevant node configuration is identified for further research. The energy analysis for MCU is discussed. The relationship between the instruction cycles and energy consumption is detailed. In Section 2.3, compressive sensing is introduced, and the major design challenges are highlighted. The two major sensing matrix designs are also discussed with the leading research output reviewed. Section2.4 provides a summary of the major findings uncovered in the literature and ends the chapter.

2.2 WIRELESS MULTIMEDIA SENSOR NETWORKS

WMSNs have been developed to address various problems in the real world. In [33], Chang and Huang proposed the use of WMSNs for emergency services such as the execution of search and rescue operations in unknown, dynamic and unsafe environments. The other applications are listed in Table 2.1. The image sensors that have commonly been used are low-resolution, amongst others, common intermediate format (CIF) and video graphics array (VGA) built with complementary metal-oxide-semiconductor (CMOS) sensors. However, there have been high-resolution sensors as well in the high-end nodes, such as 1280x1024. Most applications have been on surveillance of humans and their activities, the most common being in the assisted living environment.

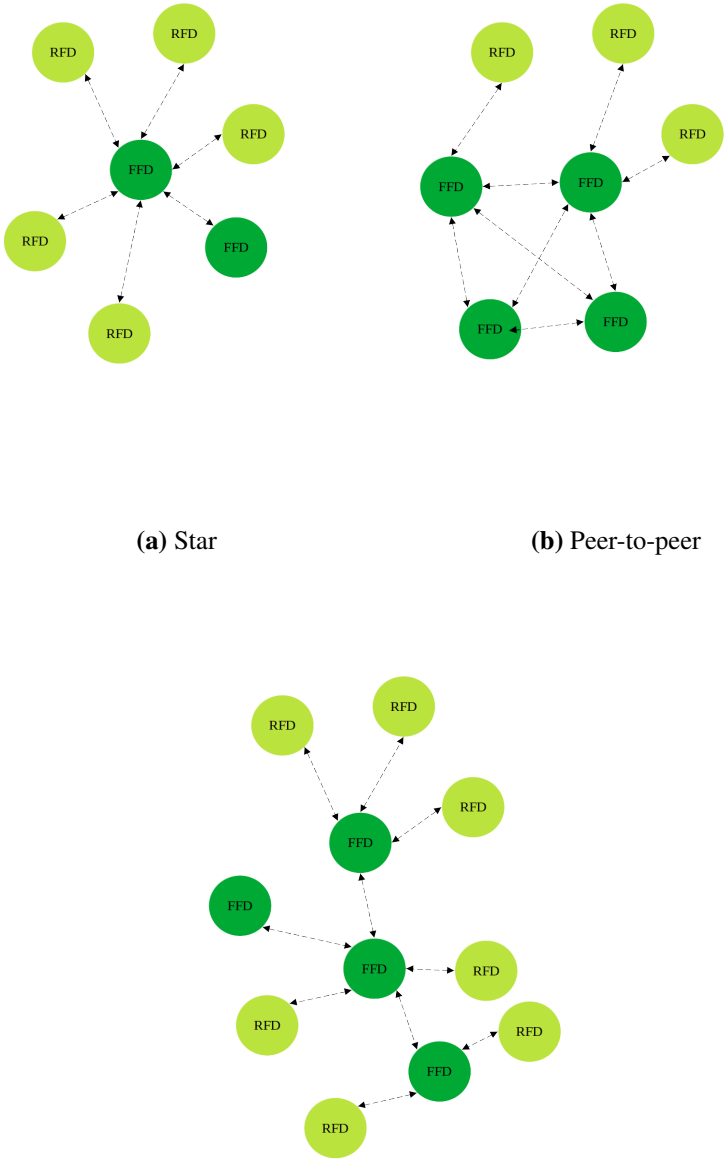
Table 2.1. WMSN applications.

Ref.	Sensor	Network	Application
[34]	VGA CMOS	IEEE 802.15.4	Activity recognition for assisted living
[35]	CIF CMOS	RF transceiver	Object detection and hand gesture recognition
[36]	VGA CMOS	IEEE 802.15.4	Distributed face detection and hand gesture recognition
[37]	VGA CMOS	RF transceiver	Low resolution (320x240) video streaming and high resolution (VGA) snapshots based on event triggers
[38]	VGA CMOS	IEEE 802.15.4	Object detection, tracking and localisation
[39]	CIF CMOS	—	Motion and face detection
[40]	VGA CMOS	IEEE 802.15.1	People detection using optimised image classification
[41]	CIF CMOS	IEEE 802.15.4	Assisted living where multiple cameras are fused to detect and monitor human activities
[42]	1280x1024 CMOS	IEEE 802.15.4	Single object tracking through background subtraction

The most common network interface was the IEEE 802.15.4 standard which describes a low-rate wireless personal area network (LR-WPAN). The IEEE 802.15.4 supports a range of 10 m and a data rate of 250 kb/s. There are two types of network nodes supported by the standard. The first type of node is a full-function device (FFD). This node can operate as a personal area network (PAN) coordinator as well as an elemental node. The second type of node is a reduced-function device (RFD) which can only communicate with an FFD, and these nodes are shown in Figure 2.1 along with the network topologies they can support.

The WMSN has experienced various designs in node configuration; these have entailed processors ranging from the ATmega128L running at 8 MHz to XScale PXA270 at 624 MHz. Many WMSN nodes are available for academic and commercial use [43]. One of the most influential considerations in node design has been whether to adopt the *compress-then-analyse* or *analyse-then-compress* paradigms [44]. Traditionally, the *compress-then-analyse* was favoured, but more recent node designs have incorporated

more computational power to do some analysis to reduce network bandwidth utilisation. However, Redondi *et al.* found that the *compress-then-analyse* paradigm is more energy efficient.



(a) Star **(b)** Peer-to-peer

(c) Cluster-tree

Figure 2.1. The IEEE 802.15.4 network topologies.

The choice of processor is a balancing act between power efficiency and node processing capacity. This trade-off, however, is superseded by cost. In order to make WMSN practical, the nodes need to

be cost-effective and to fully leverage the potential of the technology, many of these nodes need to be deployed. Thus costs need to be controlled to make this technology accessible to a broad scope of applications.

2.2.1 Low-Performance Nodes

In [3], WMSN nodes are classified as low and high performance depending on the processor capacity. The low-performance nodes have lower purchase costs and the potential for longer battery lifespans. These properties make them suitable for many environments.

In [35], Rahimi *et al.* proposed the Cyclops. The researchers took advantage of small form-factor, inexpensive and energy-efficient CMOS imaging sensors in order to allow WSN to differentiate objects and evaluate their importance. The authors introduced CMOS imagers to WSN nodes to make possible new types of computer vision applications. The Cyclops had an 8-bit ATmega128L + CPLD processors.

Kleihorst *et al.* [36] proposed the WiCa to allow multiple cameras to view the same scene and communicate to construct high-performance surveillance systems. The researchers took advantage of a single instruction multiple data (SIMD) video analysis processor for onboard processing. The nodes communicate with each other using IEEE 802.15.4 on peer-to-peer connections for surveillance collaboration. The WiCa had 84 MHz Xetal + 8051 ATMEL processors.

In [37], Park and Chou proposed the eCam, which is a very compact, high data transfer rate WMSN node. It comprised a low-resolution digital video camera and the ultra-small self-contained Eco WSN node. The eCam achieved reliable VGA resolution video streaming. The eCam had an OV 528 serial-bridge controller.

Hengstler *et al.* [38] proposed the MeshEye mote as an energy-efficient smart camera mote for intelligent surveillance. The mote had a low-resolution stereo vision system to enable advanced object detection and triangulation. The stereo system worked in concert with a high-resolution module to facilitate object capture for further processing. The MeshEye had a 55 MHz 32-bit ARM7TDMI processor.

In [39], the CMUcam3 is presented as an inexpensive, open-source, embedded vision platform. The

CMUcam3 consists of a CMOS camera, frame buffer, MCU and a MultiMediaCard (MMC) card slot. The CMUcam3 had a 60 MHz 32-bit ARM7TDMI processor. The CMUcam3 also had ports to support applications in robotics.

The MicrelEye was proposed for cooperative distributed video processing applications with classification [40]. The node consists of an inexpensive low-resolution CMOS sensor, a programmable processing engine and a 100 m transceiver. The MiscrelEye had an 8-bit ATMEL FPSLIC processor.

Rowe *et al.* [41] proposed the FireFly Mosaic node as an image processing framework with image processing framework primitives for the development of distributed vision-sensing tasks. The node comprises a FireFly real-time sensor networking platform with a CMUcam3 embedded vision processor. The FireFly had a 60 MHz 32-bit LPC2106ARM7T DMI processor.

The processor is a fundamental component of a WMSN sensor node and accounts for a substantial share of the total energy consumption [45]. In Table 2.2, the energy consumption across the different considered low performance nodes is compared based on processor speed.

Table 2.2. Low performance node comparison.

Ref.	Processor	Clock Frequency	Energy consumption
[35]	8-bit ATmega128L MCU	8 MHz	Low
[36]	8-bit 8051 ATMEL MCU	12 MHz	Low
[37]	OV 528 serial-bridge controller	-	Low
[38]	32-bit ARM7TDMI processor	55 MHz	Medium
[39]	32-bit ARM7TDMI processor	60 MHz	Medium
[40]	8-bit AVR ATMEL MCU	50 Mhz	Medium
[41]	32-bit ARM7TDMI processor	60 Mhz	Medium

The TelosB mote has been widely used as a reference implementation in the WMSN community [6, 9, 12, 46]. This mote consists of a Texas Instrument 16-bit MSP430 MCU that has an 8 MHz frequency with 10 KB of random access memory (RAM) and 1 MB of external flash [47]. The TelosB has an IEEE 802.15.4 complaint RF transceiver capable of a range of 30 m indoors and 100 m outdoors.

This mote is often equipped with 2 AA batteries, each with a capacity of about 2850 mAh, similar to other nodes [48, 49, 50]. The mote and its major components are shown in Figure 2.2.

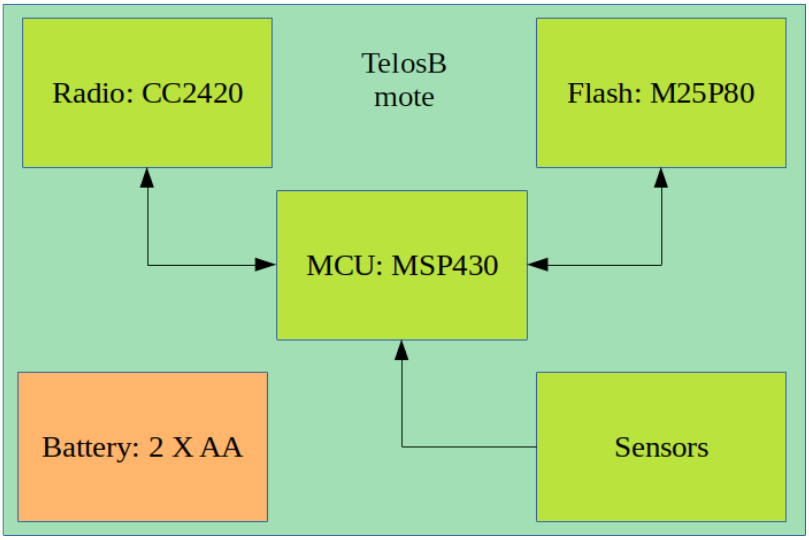


Figure 2.2. The TelosB mote.

The sensing matrix determines the number of needed random numbers for construction and the number of matrix multiplications during sensing. The Bernoulli matrix was found to be more suitable for hardware implementation due to its entries being ± 1 [51]. These entries do not require multiplication operations. Another favoured matrix is the Binary Sparse Random matrix with a few d non-zero entries per column, which eliminates the majority of addition operations [52]. There are two choices for acquiring random entries during matrix construction, reading from memory and random generation. Reading from memory has the disadvantages of ample storage and substantial energy cost to operate non-volatile memory [53]. The random generation has the burden of significant energy consumption from complex mathematical functions like *log* and *sqrt* [52]. Deterministic measurement matrices have been selected as the solution to simplifying hardware design by eliminating random generation and/or ample memory storage [32]. The general CS encoder framework is shown in Figure 2.3.

2.2.2 Energy Consumption of MCU

The current drawn by 8 to 32-bit MCUs is altered only modestly by different operations when in active state [54]. Therefore, in order to predict comparative energy consumption between the various operations, it is enough to measure the duration of each, as seen in Equation (2.1).

$$E_{con} = V_{cc} \times I_{active} \times t_{op} \tag{2.1}$$

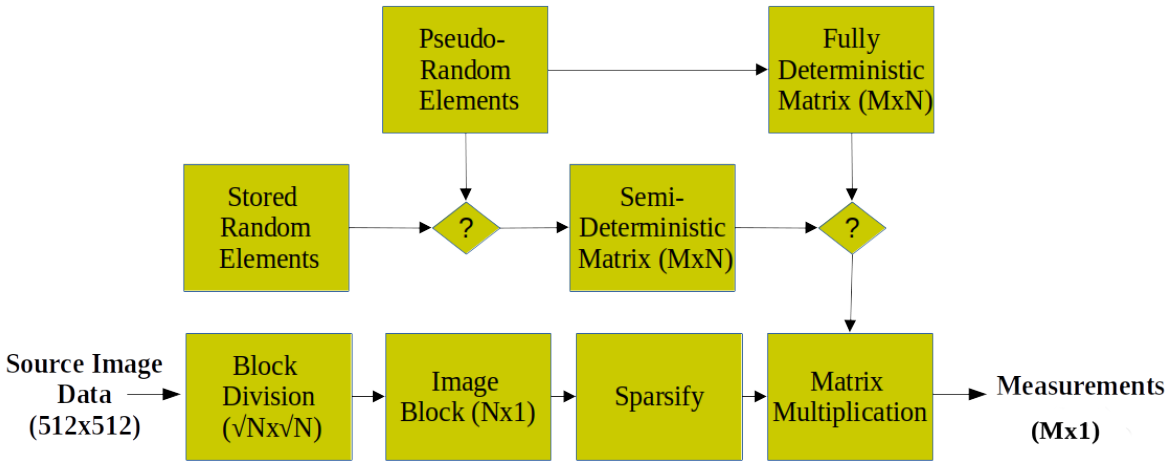


Figure 2.3. The general CS encoder framework.

The time t_{op} can be predicted through the clock frequency f and number of instruction cycles needed for the operation N_{ic}, as shown in Equation (2.2).

$$t_{op} = \frac{N_{ic}}{f} \tag{2.2}$$

The MSP430 excludes a hardware multiplier; therefore, the execution of complex mathematical operations needs multiple instruction cycles. In [55] optimised algorithms were evaluated for the implementation of multiplication and division operations on the MSP430. In [55] the instruction cycle cost of basic operations is discussed, such as *add*, *subtract* and *compare*; these operations cost one cycle each. However, other operations such *multiply* and *divide* require complex algorithms to implement efficiently on the MCU. These algorithms are discussed in [56], where the most efficient implementation of *multiply* and *divide* costs 29 cycles and 22 cycles, respectively.

2.3 COMPRESSIVE SENSING

The CS framework has an encoder that compresses signal $x \in R^N$ into measurements $y \in R^M$, where $M \ll N$. In order for the signal to be recovered from the measurements, there are two conditions that must be met, sparsity and incoherence [57]. Sparsity characterises how few non-zero entries are found in a vector representing the signal. A signal can also be transformed into a domain where it is sparse by using a mathematical transformation, Ψ. Incoherence characterises how low is the correlation of the sparsifying transform with the measurement matrix, Φ [57]. This can be measured through mutual coherence, which is the maximum normalised similarity between the row vectors of Φ, ϕ_i and column vectors of Ψ, ψ_j.

The signal can be recovered from these measurements using numerical optimisation algorithms; this is the decoder half of the framework. The recovery algorithm exploits the sparsity assumption through Equation 2.3

$$\min_{z \in R^N} \|z\|_0 \text{ subject to } Az = y, \tag{2.3}$$

where $A = \Phi\Psi$, however, since this optimisation is an NP-complete problem, the convex relaxation is used instead

$$\min_{z \in R^N} \|z\|_1 \text{ subject to } Az = y. \tag{2.4}$$

The Gaussian measurement matrix and the ℓ^1-minimisation are the most widely studied encoder and decoder pair [58]. This is because they lend themselves to mathematical analysis. However, there are many other combinations since many measurement matrices and reconstruction algorithms have been proposed.

The robustness of the CS framework is founded on the restricted isometry property (RIP) of the measurement matrix. The RIP needs the columns of Φ to be nearly orthogonal. Random measurement matrices adhere to the RIP criteria with a high likelihood if the Equation (2.5) [57] holds. The RIP is also essential in making certain that CS can overcome additive noise gracefully.

$$M \geq WK \log\left(\frac{N}{K}\right), \tag{2.5}$$

where W is a small positive number and K is the sparsity of x. Reconstruction algorithms can be grouped into two broad types, convex relaxation and greedy pursuits [59]. Basis Pursuit (BP) is a convex optimisation algorithm that unpacks a signal into a superposition of dictionary elements with the smallest ℓ^1 norm [60]. The BP has high computational complexity, which undermines feasibility for large-scale applications [61].

2.3.1 Reconstruction Algorithms

The rapid reconstruction and low complexity of mathematical framework have led to the popularity of iterative greedy algorithms in compressive sensing [62]. Currently, the most popular greedy algorithms are Matching Pursuit (MP) and its derivative Orthogonal Matching Pursuits (OMP) due to their ease of implementation and fast recovery [63]. Other greedy algorithms are Stagewise Orthogonal Matching

Pursuit (StOMP), Regularized Orthogonal Matching Pursuit (ROMP) [64] and Compressive Sampling Matching Pursuits (CoSaMP) [65] that seek to deal with lack of sparsity on the input signal.

The MP algorithm [66] iteratively chooses elements from a dictionary that give the closest approximation of the uncompressed signal. The OMP [67] enhances the MP by minimising the number of iterations necessary to recover the signal. This improvement was achieved with the sacrifice of computational complexity by selecting optimal element indexes and updating subspace columns for each iteration [68]. The MP and OMP can take an indeterministic number of iterations to find a signal approximation, and thus, StOMP was proposed to overcome this shortcoming [69]. With StOMP, many elements can be selected per stage, unlike OMP.

In [70], Mun and Fowler proposed the block-based CS sampling and smoothed projected Landweber (BCS-SPL) algorithm. The authors wanted to reduce computational complexity without compromising image quality for convex relaxation algorithms. A directional transform was exploited to increase sparsity and smoothing to realise rapid reconstruction.

There are different minimum sampling requirements for the recovery algorithms discussed in detail in [63]. The Basis Pursuit, OMP, CoSaMP, and Belief Propagation have a minimum sampling requirement of $\mathcal{O}(K \log N)$. While Subspace Pursuits, Expander Matching Pursuits and Sparse Matching Pursuits have $\mathcal{O}(K \log(N/K))$. The other sampling requirements are unique to very few algorithms, such as ROMP with $\mathcal{O}(K \log^2 N)$ and StOMP with $\mathcal{O}(N \log N)$. The reconstruction algorithms are compared in Table 2.3 using minimum sparsity as one of the criteria.

Table 2.3. Reconstruction algorithm comparison.

Ref.	Type	Name	Aim	Minimum Sparsity
[60]	Convex relaxation	Basis Pursuit	Unpacks a signal into a superposition of dictionary elements	$\mathcal{O}(K\log N)$
[66]	Greedy	MP	Chooses elements from a dictionary that give the closest approximation of the uncompressed signal	
[67]	Greedy	OMP	Fewer iterations than MP	
[69]	Greedy	StOMP	Deterministic number of iterations to find a signal approximation	$\mathcal{O}(N\log N)$
[64]	Greedy	ROMP	Combining the strengths of convex relaxation and greedy approaches	$\mathcal{O}(K\log^2 N)$
[65]	Greedy	CoSaMP	Similar recovery performance to the best convex relaxation approaches	$\mathcal{O}(K\log N)$
[70]	Convex relaxation	BCS-SPL	Low computational complexity and high recovery accuracy	

2.3.2 Sparsity Transforms

The compressed signal is only sometimes sparse; therefore, a sparsity transform is often used to achieve sparsity. The most popular sparsity transforms in the WMSN literature have been discrete cosine transform (DCT), and it's variants [7, 12, 15, 71, 72, 73], discrete wavelet transform (DWT) [6, 9, 46], and discrete Tchebichef transform (DTT) and it's variants [74, 75]. The typical approach with the varients was to prune the transforms to make them more computationally efficient, but this always came at the cost of the image quality of the recovered image. The three transforms in their standard form were compared in [8] to evaluate their relative performance in compressed sensing. It was found that the transform offered the best recovery performance and energy efficiency was the DCT.

The DCT divides an image into differing spatial frequencies in order for the minor impactful frequencies to be eliminated. The two-dimensional (2D) DCT is generally computed as two sequential one-dimensional (1D) calculations, first along the rows and then the columns. This is because the DCT is mathematically separable. Efficient exact implementations of the DCT exploit factorisation of the transform into the vectors; P, M and A [76]. The vector A is the additive matrix for the set of butterfly

computations, M is the multiplicative matrix, and P is the permutation matrix for rearranging the output. The implementation that achieves the theoretical limit of multiplications is the Loeffler, Ligtenberg and Moschytz (LLM) or commonly referred to as the Loeffler [77].

The DCT uses orthogonal cosine basis functions to get the transform coefficients. The first coefficient accounts for the first basis function, which is a constant, while each successive basis has a higher frequency. In most images, most of the energy is condensed in the low-frequency coefficient, and the rest have small values [78]. The DCT has good energy compaction but for images with high auto-correlation [79]. This means that when the image has little auto-correlation, the energy is spread out over more frequencies, and fewer coefficients are close to zero. In [15], compressibility was predicted using luminance standard deviation and the sampling rate was increased to compensate for poor compaction. It was assumed that an image with more texture requires higher compression.

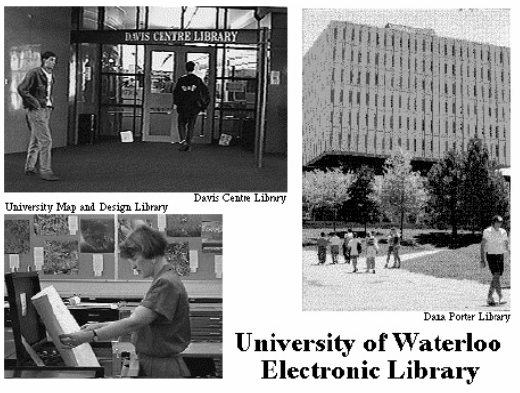
Rauhut *et al.* [80] broadened the scope of CS applications to signals that are not sparse on an orthonormal basis by proposing redundant dictionaries for sparse representation. However, obtaining the optimal coefficients of redundant dictionaries for a given signal is computationally demanding and requires optimisation algorithms [81].

2.3.3 Adaptive Sampling Rates

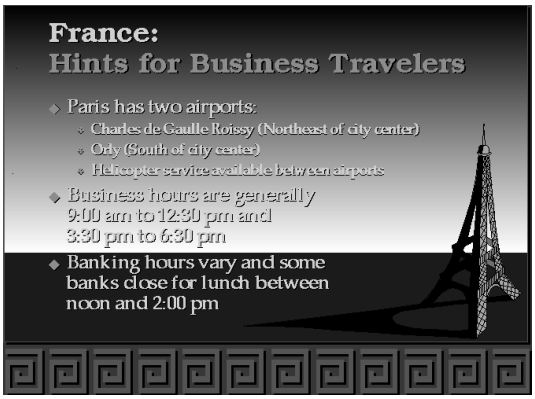
Images have different content that does not always get sufficiently compacted by sparsity transforms. Adaptive sampling rate schemes have been proposed to detect and adapt to these challenging images. Examples of difficult images are shown in Figure 2.4 based on results from [8]. These images were not recovered well for all the sparsity transforms that were utilised, DCT, DWT and DTT.

Luo and Si-Wang [82] proposed an algorithm that unpacks an image through a wavelet packet transform and then exploits expectation and information entropy to separate the coefficients of each block. Each block is then compressed using one of three methods.

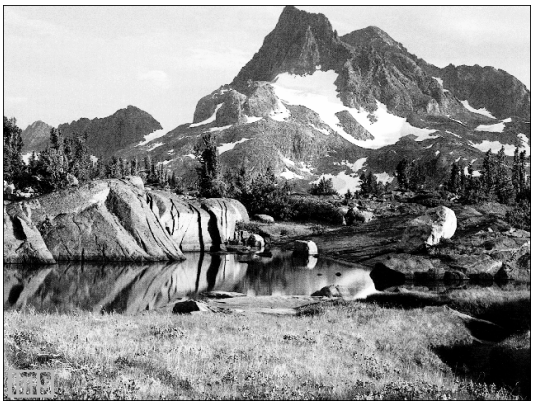
Rong-Fang *et al.* [83] used texture information for each image block to determine a sampling rate. The texture information was measured using 1D entropy, and higher entropy blocks were sampled using a higher rate.



(a) Library image



(b) France image



(c) Mountain image

Figure 2.4. Difficult images to compress.

In [15], Zhang *et al.* assigned sampling rates for each block based on the standard deviation of the block. The higher the standard deviation, the higher the sampling rate. Li *et al.* [84] used the block-based gradient field to assign the sampling rate, where a block with a higher gradient will be sampled at a higher rate.

Zhang *et al.* [85] used the saliency of each block to assign a sampling rate. The authors measured the saliency using statistical textural distinctiveness. Monika *et al.* [86] used the energy content of a block to determine the sampling rate. The energy was measured as the sum of the square of the mean and variance of low-frequency DCT coefficients. The different sampling strategies are compared in Table 2.4 in terms of their ease of implementation.

Table 2.4. Adaptive sampling strategies.

Ref.	Sampling Strategy	Ease of implementation
[82]	Expectation and information entropy	Low
[83]	1D entropy	High
[15]	Standard deviation	High
[84]	Block-based gradient	High
[85]	Statistical textural distinctiveness	Low
[86]	Sum of the square of the mean and variance of low-frequency coefficients	Low

2.3.4 Sensing Matrices

A properly constructed measurement matrix is essential for the faithful recovery of a compressed signal [87]. The measurement matrix can reduce the sampling rate needed while enhancing the stability and fidelity of the recovery algorithm [88]. Initially, dense (unstructured) random measurement matrices were popular [89, 90]. The Gaussian and Bernoulli matrices are solidly established in the literature and have been theoretically proven to be suitable candidates for sensing matrices [91]. However, the high dimensional vectors that are needed to construct these matrices make them unsuitable for real-world application [92].

In [17], Bajwa *et al.* proved that semi-deterministic measurement matrices, where only the first row of the matrix uses random entries, are viable. Do *et al.* [93], proposed the Structurally Random Matrix (SRM) for large-scale and real-time compressive sensing. The different categories of proposed sensing matrices can be seen in Figure 2.5, as discussed in [1].

The SRM [93] has been influential in the development of structured measurement matrix, but subsequent matrices have had better image quality, such as [94] and have had lower computational complexity, such as [95]. The most efficient matrix was [95], which had the most reduced storage and computational complexity.

Elad [18] directly optimised the sensing matrix to minimise the average measure of the mutual coherence. This approach led to better performance than random measurements. Duarte-Carvajalino and Sapiro [19] demonstrated the utility of jointly optimising a sensing matrix and overcomplete dictionary transform with encouraging results. However, this approach was computationally demanding,

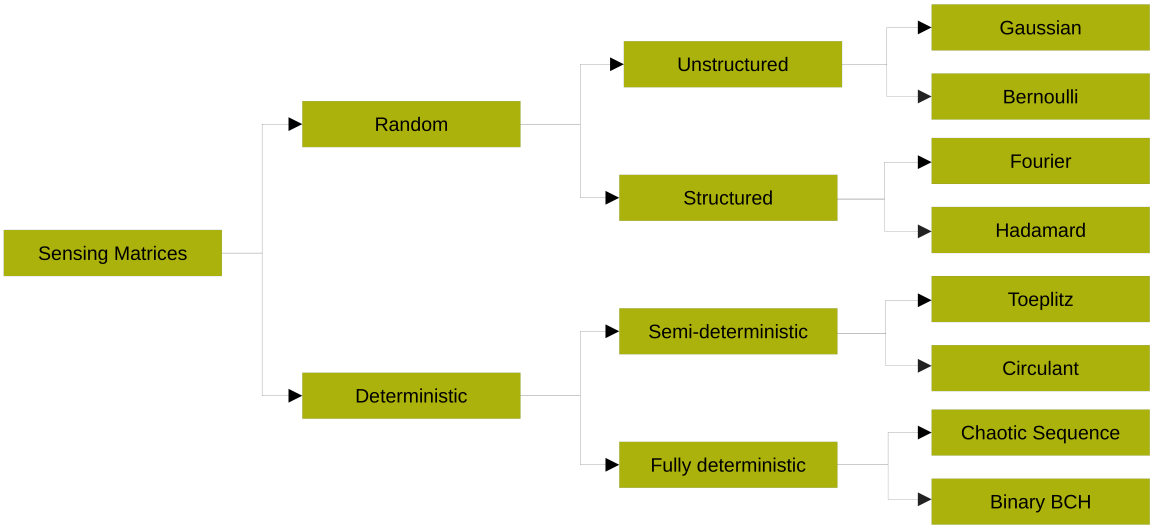


Figure 2.5. The sensing matrix categories. Taken from [1], ©2021 IEEE.

and the authors used image patches for their experiments. Abolghasemi *et al.* [20] proposed an alternating optimisation of the sensing matrix for reducing the mutual coherence. Their minimisation algorithm was based on gradient descent and was able to improve recovery accuracy. There have been many other approaches to mutual coherence minimisation [21, 22, 23], but the construction is infeasible on a WMSN.

In [24], Baldassarre *et al.* used the signal dataset to optimise the sensing matrix. Unlike previous approaches, a structured measurement matrix was developed that offered storage and computational advantages. Ahmed *et al.* [25] trained a binary sensing matrix on a dataset to improve recovery accuracy. For each iteration of the training, the positions of zero and non-zero elements were randomly swapped, and the structural similarity index (SSIM) of the recovered signal was used as a fitness value. More recently, Ahmad and Khan [26] used training to optimise a sensing matrix by identifying regions that have the most energy of the training signals. A selected number of coefficients were chosen, and set the others to zeros. Similarly, with optimisation matrices, these measurement matrices are not ideal for WMSN nodes, the sensing matrices are not universal and need offline training.

In [1], it was discovered that deterministic sensing matrices could offer better recovery accuracy than structured matrices, and the rest of this study will focus on semi-deterministic and fully deterministic matrices.

2.3.4.1 Semi-deterministic Matrices

In dealing with challenges related to hardware implementation and sensing efficiency, He *et al.* [96] suggested the use of a binary permuted block diagonal (BPBD) matrix. It is binary and sparse, making hardware implementation simple and diminishing sensing computations. During construction, it utilises random numbers to permute its columns, which increases computational complexity based on the generation of random numbers. The authors conducted experiments on MATLAB to compare the BPBD matrix with other matrices. They found that the proposed measurement matrix has similar recovery performance to different matrices and the Gaussian. Additionally, the measurement matrix outperforms other matrices at meagre sampling rates.

A study was conducted by Sun *et al.* in [29] aimed to address the computational complexity associated with dense measurement matrices. They introduced the sparse block circulant matrix (SBCM) as a solution, which has the gain of being both sparse and circulant within each block. This is in contrast to other structurally random matrices, which are only circulant across blocks. The SBCM is created using sparse circulant submatrices, with each submatrix constructed from Gaussian numbers used row-wise and block-wise. The researchers compared the storage and computational complexity of the SBCM to dense random matrices and evaluated its performance against the Gaussian matrix in 1D signal experiments. Results showed that the SBCM achieved a similar mean square error (MSE) to the Gaussian matrix but with significantly reduced complexity.

In [30], Su and colleagues aimed to address the challenges associated with implementing a random measurement matrix in hardware. They utilised a Toeplitz matrix and improved its randomness by substituting pseudo-random chaotic entries with random ones. In addition, they reduced the complexity of calculations by splitting the Toeplitz matrix into block-diagonal and skew-circulant components. The outcome of this was an enhanced Toeplitz measurement matrix (ITMM). The researchers evaluated the storage and computational complexity of the matrix and conducted experiments using a DCT and the StOMP recovery algorithm on monochrome images that were divided into 16x16 blocks at high sample rates. The ITMM measurement matrix outperformed the Gaussian and other matrices with regard to signal-to-noise ratio (SNR).

A study by Nandhini et al. (2015) presented two memory-efficient measurement matrices. The first of the two matrices, named the Combination matrix, is created by taking the Kronecker product of the Gaussian and Toeplitz measurement matrices [9]. While this method diminishes the number of random

entries necessary, it increases the complexity of construction due to the multiplication operations. The last matrix, named the Hybrid matrix, is generated by combining Toeplitz and Binary matrices. According to the results, the proposed matrices outperformed the Gaussian matrix with regard to energy consumption and recovery performance. Specifically, the Hybrid matrix performed better than the Combination matrix.

In a study conducted by Nandhini *et al.* (2019), a combination of Toeplitz, Hankel, and circulant matrices was utilised. The Toeplitz matrix was created using random Gaussian entries, while the Hankel and circulant matrices were formed through transformations of the Toeplitz matrix [12]. These three matrices were then combined to generate the sensing matrix. The researchers compared their matrix to the Gaussian matrix and found that their matrix had superior recovery performance and energy efficiency.

2.3.4.2 Fully Deterministic Matrices

In a study by Sun *et al.* [97], an enhanced version of the Hadamard sensing matrix was presented. The standard construction of this matrix involves randomly selecting M rows from an $N \times N$ Hadamard matrix to create an $M \times N$ matrix. However, the researchers opted to choose M rows in order instead of randomly. This deterministic approach ensured that the row vectors satisfied both orthogonality and non-linearity attributes while also reducing computational complexity. The effectiveness of the improved Hadamard measurement matrix was compared to the standard matrix through experiments on monochrome images using the DWT and OMP recovery algorithm. The results showed that the proposed measurement matrix outperformed the standard Hadamard matrix in regard to peak signal-to-noise ratio (PSNR), particularly at low sampling rates.

Authors Ravelomanantsoa *et al.* [32] have introduced a deterministic measurement matrix suitable for hardware implementation called the deterministic binary block diagonal (DBBD). This matrix improves upon the BPBD by eliminating random permutations. The authors conducted experiments using 1D biomedical signals to evaluate the proposed measurement matrix against the Gaussian and BPBD matrices. The DCT and a newly proposed OMP variant reconstruction algorithm were utilised. Results showed that the proposed measurement matrix outperformed both the Gaussian and BPBD matrices with regard to recovery accuracy, measured in SNR, as well as computational complexity.

A research paper by Yao *et al.* [10] utilised the low complexity of Logistic mapping chaotic systems to

create an incoherent rotated chaotic (IRC) sensing matrix. The first 1000 elements of the sequence were disregarded, and the remaining elements were downsampled using an interval d to increase randomness. This process requires a chaotic sequence of length $l = 1000 + dt$ to generate t sampled values. The IRC matrix only needs n rotated factors for each row, thus reducing storage compared to other chaotic matrices. The authors introduced an incoherence factor η, which is multiplied for each rotation and can be separated into a matrix ω. However, this leads to increased computational complexity due to multiple multiplications from the incoherence factor. MATLAB experiments were conducted to compare the IRC matrix with the Gaussian. The study showed that the IRC matrix performs better, but the selection of the incoherence factor presents a challenge for implementation as it heavily impacts reconstruction performance.

In their study, Hong *et al.* [11] created a structured sparse sensing matrix that minimised the distance between the Gram matrix of the equivalent dictionary and the target. This was achieved by ensuring low mutual coherence. To ensure robustness to sparse representation error of the target signal, they regularised the improved structured sensing matrix. The researchers combined a row-wise sparse matrix and a base sensing matrix to create the sensing matrix. They chose a DCT matrix with a dictionary learned using the KSVD algorithm. The computational complexity of the matrix was compared to that of a dense matrix. To contrast their sensing matrix against other types of sensing matrices, such as a dense random matrix, the authors conducted a contrast study. They also included a denoising algorithm called BM3D to minimise blockiness. The researchers evaluated the images and found that their sensing matrix produced better results than the dense random sensing matrix when measured using PSNR.

In a study by Gan *et al.* [13], a measurement matrix was proposed that utilises bipolar chaotic sequences to reduce storage and eliminate multiplication. However, these sequences are generated from the Chebyshev chaotic system, which requires the computationally burdensome *cos* function to be implemented on the MCU. To transform the values into a bipolar matrix, a threshold value is applied to the sequence. The researchers conducted experiments comparing their matrix to the Gaussian, Bernoulli, improved Hadamard, and dense Chebyshev matrices. Their matrix performed similarly to the other matrices but was the most effective at low sampling rates.

In their research paper, Sun and colleagues [31] introduced the Chaos-Bernoulli block circulant matrix as a means to reduce the use of transmission resources. To generate a pseudo-random sequence, they

selected an initial value and sampling interval and used a non-linear Hybrid Chaotic map that integrated the Logistic and Tent maps. They discarded the first 1000 values of the sequence to enhance its randomness and sampled it at an interval to ensure the independence of the sampled values. They then applied the *sign* function to the sampled sequence and used it to create a block circulant matrix, which helped to minimise storage and implementation requirements. To obtain the final measurement matrix, they randomly selected M rows from the block circulant matrix. The authors compared their matrix with other matrices like the Gaussian, Bernoulli, Hybrid chaotic, and Gaussian circulant matrices using numerical experiments. They concluded that their matrix performed better than the others.

2.3.4.3 Matrix Comparison

Various studies have revealed that semi-deterministic and fully deterministic measurement matrices outperform dense random matrices [9, 10, 11, 12, 13, 29, 30, 31, 32, 96, 97]. Table 2.5 presents a comparison of the performance of different matrices. The complexity cost is classified as Low, Medium, or High. Reconstruction accuracy is determined with regard to image quality, which must be better than that obtained using the Gaussian matrix. The matrices with the lowest sensing complexity are [96] and [32] in this case.

Table 2.5. Sensing matrix properties.

Ref.	Type	Construction Cost	Sensing Cost	Image Quality
[96]	Semi	High	Low	Good
[29]	Semi	High	Medium	Competitive
[30]	Semi	High	Medium	Good
[9]	Semi	High	High	Good
[9]	Semi	High	Medium	Good
[12]	Semi	High	High	Good
[97]	Full	High	High	Competitive
[32]	Full	Low	Low	Good
[10]	Full	Medium	High	Good
[11]	Full	High	High	Good
[13]	Full	Medium	Medium	Good
[31]	Full	High	High	Good

2.4 CHAPTER SUMMARY

The networks used in WMSN have limited data rates, typically as low as 250 kb/s. These low rates and the need for energy conservation make compression an essential tool. The most energy-efficient paradigm for image analysis in WMSN is the *compress-then-analyse*. There are many WMSN nodes that have been proposed for different applications. These nodes differ primarily by their computational capacity, namely high and low-performance nodes. Low-performance nodes are essential for making WMSNs practical and flexible. Different low-performance nodes have been proposed, but the TelosB mote can be used as a reference for energy-constrained applications.

Energy consumption on an MCU can be measured using the number of instruction cycles per operation. The most expensive operations are multiplication and division. The design of sensing matrices has been influenced by practical considerations in WSN. The two major innovations have been the replacement of floating point numbers with bipolar or binary entries and sparse sensing matrices. Random numbers have been a source of inefficiency from requiring complex mathematical functions or external storage retrieval.

Compressive sensing is dependent on the sparsity of the signal and incoherence of the sparsity transforms and sensing matrix for ensuring robustness. Different recovery algorithms have been proposed to address recovery performance and speed. The choice of recovery algorithm affects the minimum sampling required on the encoder side, where the sensing matrix is critical.

There are many factors affecting the performance of CS; these include the signal sparsity or sparsity transform, the reconstruction algorithm and the sensing matric design. The most successful sparsity transform has been the DCT which has proven to preserve image quality and require low computational complexity. The sparsity transforms are not universal, and there are images where they fail to compact the energy. Adaptive sampling has been proposed to deal with these challenging images, but an algorithm is still needed to predict compression performance in order to adjust the sampling rate.

The BCS-SPL reconstruction algorithm reduces computational complexity without negatively affecting image quality. Most other compressive sensing aspects impose a compromise between energy efficiency and recovery performance. However, sensing matrix design has been shown to allow the improvement of efficiency and recovery accuracy at the same time.

The most promising sensing matrices for energy efficiency and good recovery performance are deterministic matrices. The matrices with the lowest sensing complexity were based on the BPBD because of the sparsity.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

This chapter introduces the proposed sensing matrix and the experimental setup to validate its performance. The motivation for the presented matrix is offered, and the characteristics of the matrix are discussed. Different alternative components, such as random number generators and construction algorithms, are discussed. A novel adaptive sampling mechanism is proposed in order to keep the sample rate low while maintaining QoS expectations. The experiments conducted in this chapter answer some of the research questions.

Section 3.2 outlines the proposed matrix's design considerations. In Section 3.3, the proposed sensing matrix is presented, and the various aspects of its design are highlighted. In Section 3.4, an adaptive sampling technique is proposed. The technique is compared to similar techniques and evaluated for effectiveness. Lastly, the technique is optimised for computational efficiency. In Section 3.5, the experimental setup is discussed, and all the relevant metrics are presented. Section 3.6 provides a summary of the major elements of the chapter.

3.2 DESIGN CONSIDERATIONS

Matrices that are fully deterministic often rely on chaotic sequences to eliminate the need for random numbers. This results in matrices that are not sparse, unlike the Bernoulli matrix. The DBBD matrix, on the other hand, is a sparse, fully deterministic matrix with fixed entries. However, it does require $N-M$ *add* operations during sensing. Semi-deterministic matrices, such as the partial canonical identity (PCI), have achieved good sparsity without the need for *add* operations during sensing. Additionally, the PCI matrix has a diminished mutual coherence with the majority of sparsity transforms [98]. One drawback of the PCI matrix is that it relies on computationally expensive random numbers, which are consistent with other semi-deterministic matrices. This issue is resolved by replacing random numbers

with a chaotic sequence. To further enhance the construction complexity, random sample positions are used in place of random permutation, which also diminishes the time complexity.

3.3 DETERMINISTIC PARTIAL CANONICAL IDENTITY MATRIX

To create the PCI matrix, a set of M rows is randomly chosen from an identity matrix with dimensions of $N \times N$. While previous studies, including [98, 99, 100], have used the PCI, they have not examined its computational complexity or potential for improved recovery performance. Therefore, the way in which the PCI is constructed has a significant impact on both its energy consumption and recovery performance.

Numerical optimisation has become a common approach to improving the recovery performance of measurement matrices by reducing mutual coherence with sparsity basis Ψ, as evidenced by several sources [18, 19, 20, 21, 22, 23]. However, the DPCI's mutual coherence remains constant, as illustrated in Figure 3.1.

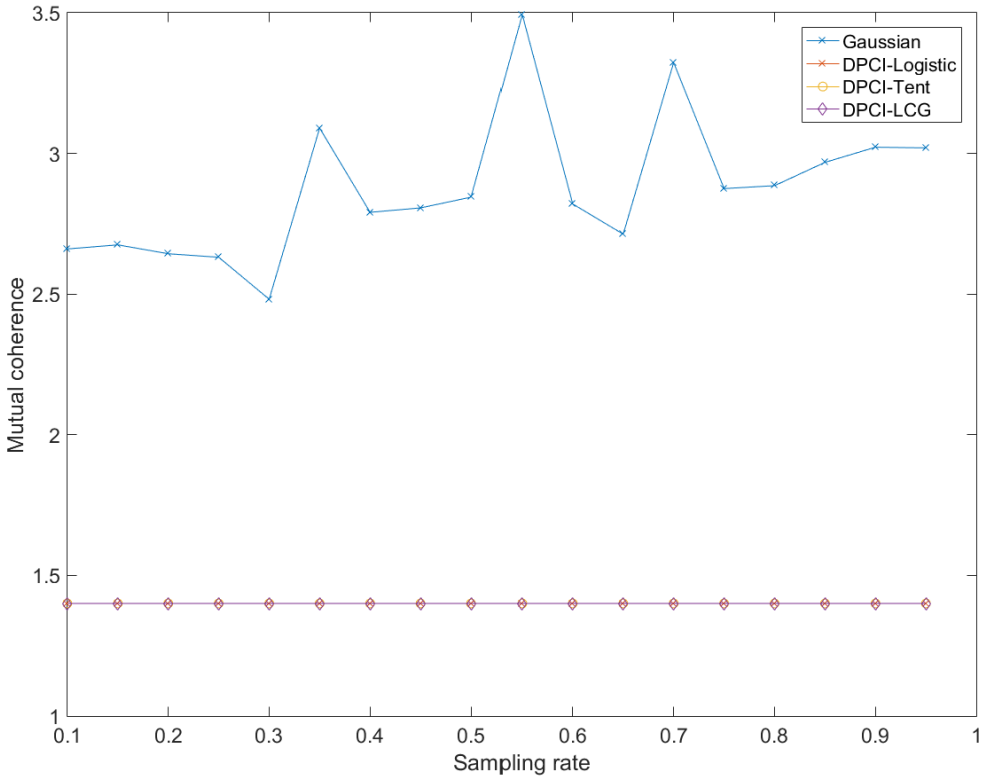


Figure 3.1. The mutual coherence of matrices using random number generators.

3.3.1 Random Number Generation

Obtaining random numbers is a crucial aspect of constructing the sensing matrix. The computational complexity can be reduced, and recovery performance can be improved depending on the method used for generating the random numbers. We analysed three different algorithms to determine the most suitable one based on their simplicity and impact on recovery performance.

3.3.1.1 Linear Congruent Generator

When it comes to generating pseudo-random numbers, the Linear Congruential Generator (LCG) is one of the earliest and most well-known algorithms. With low complexity with regards to computation and storage, this generator utilises recurrence to produce random numbers that can be seen in Equation (3.1)

$$\begin{aligned} X_{n+1} &= \Big(aX_n + c\Big) \mod m, \\ x_n &= \frac{X_n}{m}. \end{aligned} \tag{3.1}$$

The equation presented involves the use of X as the state of the algorithm and x as a random number that falls within the range of $[0,1]$. The variables a, c and m are known as the multiplier, increment and modulus, respectively, and the selection of values for these variables greatly affects the algorithm's performance. In this case, the values chosen were $a = 16807$, $c = 0$, and $m = 2147483647$, as stated in [101]. The resulting energy consumption is expressed in Equation (3.2) with regards to instruction cycles

$$\varepsilon_{LCG} = \big(\varepsilon_{mul} + 2\varepsilon_{div}\big). \tag{3.2}$$

The LCG has two *divide* and one *multiply* operations, which do not bode well for energy efficiency.

The frequency of occurrence of the random numbers is shown using a histogram with 100 bins and 10^6 trials in Figure 3.2. The histogram is almost flat, suggesting that each number has the same probability of occurrence as the next.

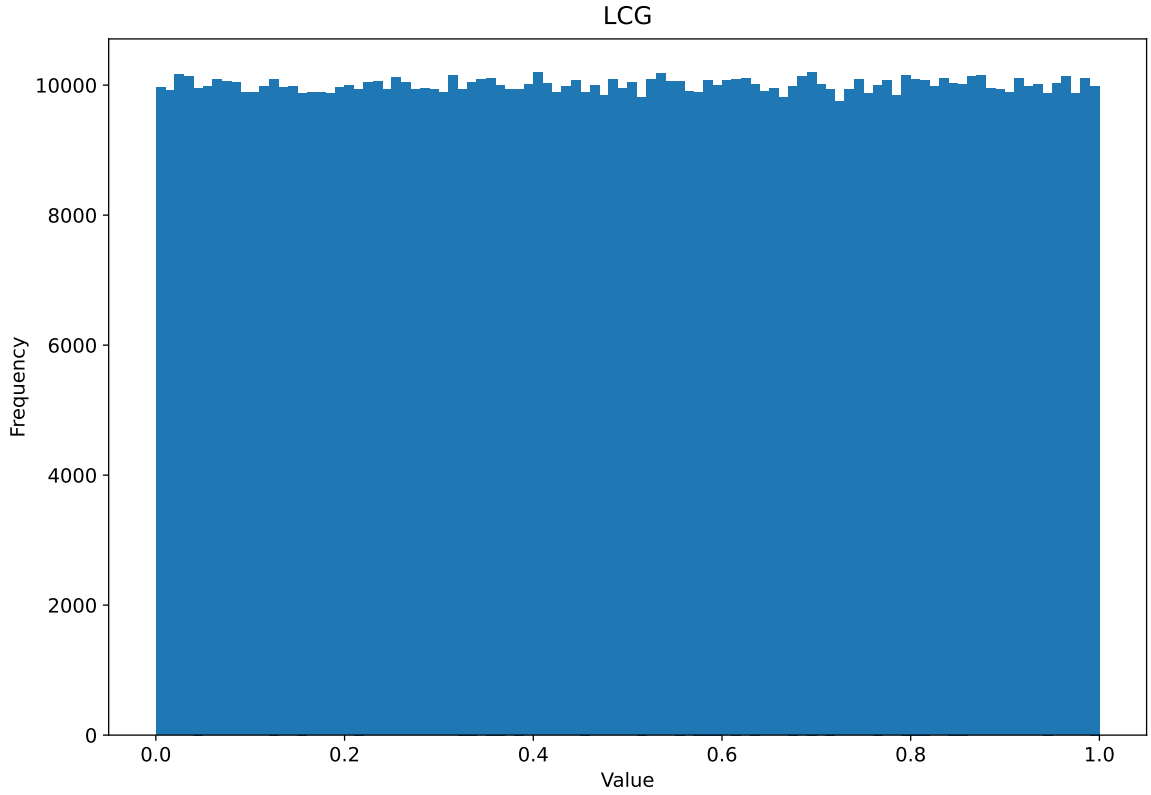


Figure 3.2. The LCG Histogram with 100 bins.

3.3.1.2 Logistic Map

One of the least complex chaotic systems is the Logistic map. It is often used as a reliable source for generating random numbers [102]. The equation for the system is presented in Equation (3.3). The variable μ represents the system parameter, while x_n is the value of each n, which falls within the range of $[0,1]$. To create the system, a value of $\mu = 4.0$ and an initial state of $x_0 = 0.4$ were utilised.

$$x_{n+1} = \mu x_n(1 - x_n) \tag{3.3}$$

The energy consumption for producing each number in the sequence is formulated in Equation (3.4) using instruction cycles. The Logistic map has two *multiply* operations, which is not ideal but much better than the LCG.

$$\varepsilon_{logistic} = \left(\varepsilon_{sub} + 2\varepsilon_{mul}\right) \tag{3.4}$$

The frequency of the Logistic random numbers is shown using a histogram with 100 bins and 10^6 trials in Figure 3.3. The histogram has two peaks, one at each end of the scale, suggesting that the probability is not evenly distributed and most values will be greater than 0.8 or less than 0.2.

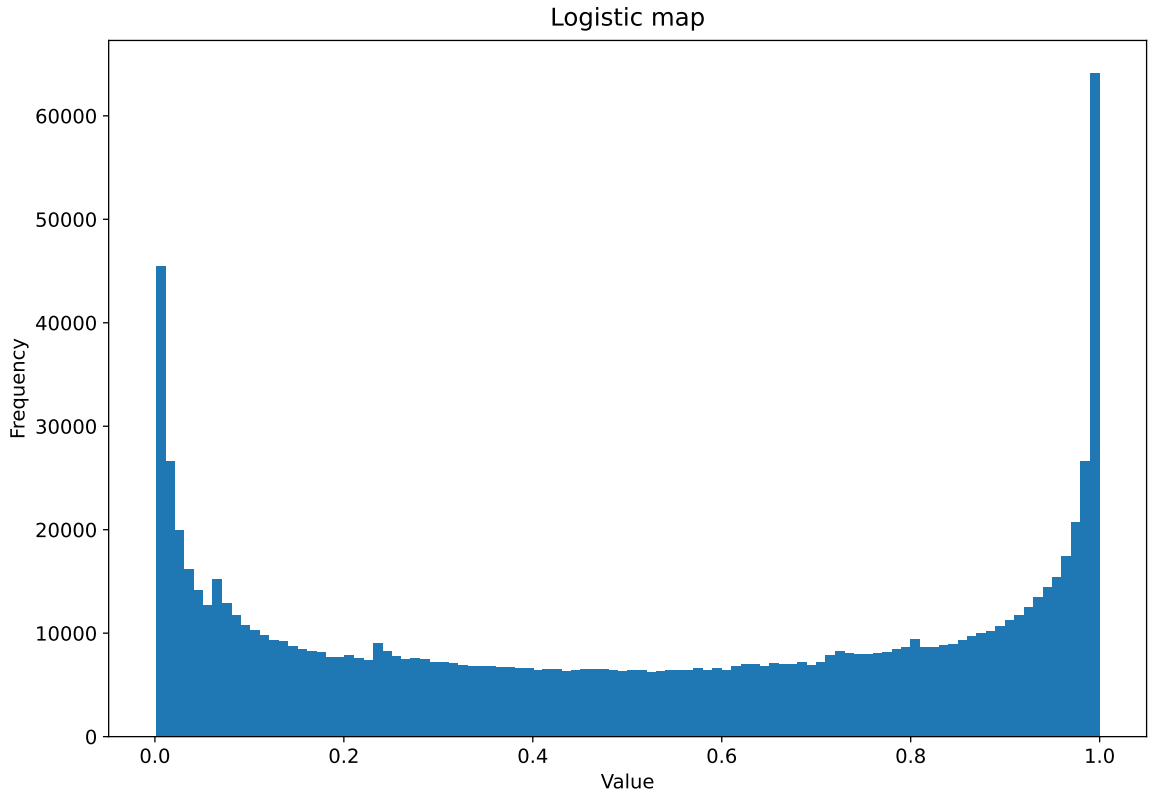


Figure 3.3. The Logistic Histogram with 100 bins.

3.3.1.3 Tent Map

A graph resembling a tent, the Tent map is a difference equation that is iterative, piece-wise linear, and characterised by chaotic behaviour. This map is known for its uniform distribution of random numbers and low computational complexity [103]. The equation for this map is shown in Equation (3.5), where μ represents the system parameter and every value of n is $x_n \in [0,1]$. In this particular instance, the values utilised were $\mu = 1.99$ for the system parameter and $x_0 = 0.4$ as the initial state.

$$x_{n+1} = \begin{cases} \mu x_n, & \text{if } 0 \leq x_n < \frac{1}{2} \\ \mu(1-x_n) & \text{if } \frac{1}{2} \leq x_n \leq 1 \end{cases} \tag{3.5}$$

The energy consumption for producing each number in the sequence is formulated in Equation (3.6). The Tent map has only one *multiply* operation, which is at least one order of magnitude better than the other random number generators.

$$\varepsilon_{tent} \approx \left(\varepsilon_{comp} + \frac{1}{2}\varepsilon_{sub} + \varepsilon_{mul}\right) \tag{3.6}$$

The frequency of the Tent random numbers is shown using a histogram with 100 bins and 10^6 trials in Figure 3.4. The histogram is relatively flat but values less than 0.2 are less likely to occur than the rest.

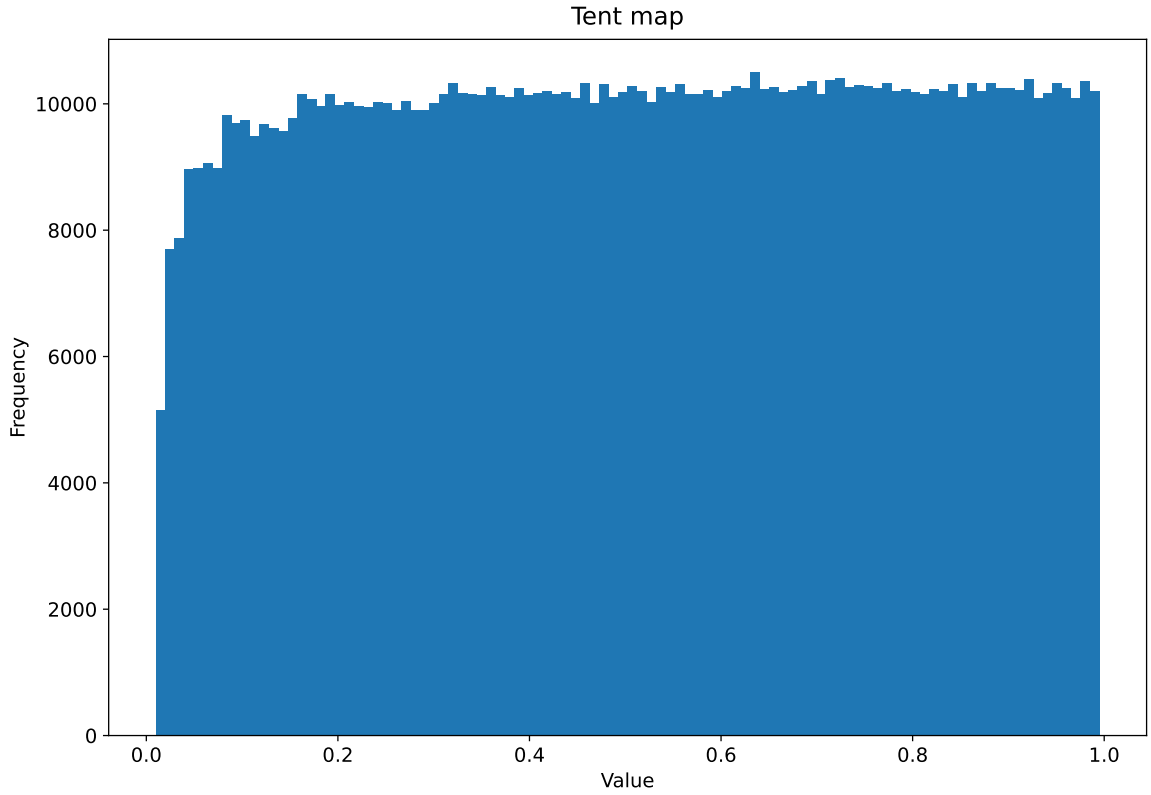


Figure 3.4. The Tent Histogram with 100 bins.

3.3.2 Complexity Optimisation

To simplify the creation of the PCI, we employed deterministic random numbers and replaced random permutations with random sample positions. This optimisation step significantly diminished both the computation and time complexity.

3.3.2.1 Random Permutation

To randomly permute the columns of the PCI matrix, the Fisher-Yates Shuffle (FYS) algorithm was used. This algorithm offers two advantages: unbiased permutations and time complexity $\mathcal{O}(n)$ [104]. The Tent map chaotic sequence was employed to generate random values, as outlined in Algorithm 1.

Algorithm 1 Random Permutation

procedure SHUFFLE(A)
  $X_0 = 0.400$
  $U = 1.99$
  for index i from $N-1$ to 1 in A **do**
    if $X_{i-1} < 0.5$ **then**
      $X_i = UX_{i-1}$
    else
      $X_i = U(1-X_{i-1})$
    end if
    $j = (i-1)X_i$+1
    Swap $A[i]$ and $A[j]$
  end for
  Return A
end procedure

The energy consumption of the algorithm is formulated in Equation (3.7). The permutation requires $N-1$ iterations to shuffle N columns. Each iteration needs two expensive computations, a Tent random number and a *multiply* operation.

$$\varepsilon_{fys} = \left(N-1\right)\left(\varepsilon_{tent} + \varepsilon_{sub} + \varepsilon_{mul} + \varepsilon_{add}\right) \tag{3.7}$$

3.3.2.2 Random Sample Position

In the PCI matrix, each row only has a single non-zero entry. This characteristic has been utilised to lessen the complexity of the construction of the DPCI matrix. To implement the random sample position algorithm, random column sample positions are generated for every row using Algorithm 2.

Algorithm 2 Random Sample Position

```
procedure SAMPLE(A)
    X_0 = 0.400
    for row i from 1 to M in A do
        X_i = random(X_{i-1})
        j = N × X_i
        A[i, j] = 1.0
    end for
    Return A
end procedure
```

The columns in the novel construction algorithm remain orthogonal, while the non-zero row entries differ between them. This design retains the RIP properties of the original shuffling algorithm. However, the effectiveness of the new algorithm is dependent on the quality of the random number generator, which must be sufficient to spread the non-zero entries evenly across columns. Additionally, the randomness of the distribution may be affected by the block size N and the sample size M.

Equation (3.8) is used to calculate the amount of energy consumed by the algorithm. In order to simulate the shuffling of N columns, the permutation requires M iterations. Each iteration only involves one complex computation, which is a Tent random number. This significantly lessens the computational complexity of the shuffling process. Additionally, the algorithm lessens the number of required random numbers and iterations from $N-1$ to M.

$$\varepsilon_{rsp} = M\varepsilon_{tent} \tag{3.8}$$

Figure 3.5 and 3.6 were used to evaluate the algorithm with all three random number generators, which included LCG, Logistic, and Tent. The algorithm was then compared to the conventional implementation that used random permutation. It was found that the conventional implementation with FYS had the best recovery performance, but LCG and Tent were also competitive. Due to its lower computational complexity, the Tent implementation was selected for further evaluation.

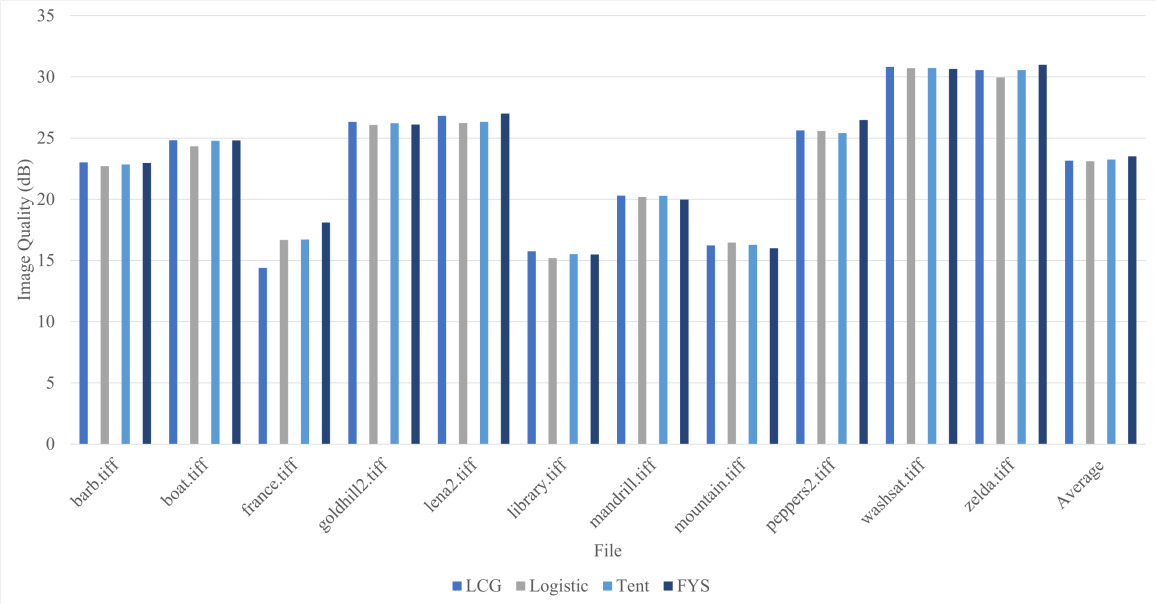


Figure 3.5. Quantitative recovery performance of different variants.

3.4 ADAPTIVE SAMPLING

Adaptive sampling is the primary means to ensure a QoS level in image quality while maintaining a low average sampling rate. A detection algorithm was developed for challenging images for DCT. The algorithm was also optimised to reduce the complexity while not compromising accuracy.

3.4.1 Median Coefficient Magnitude

The median magnitude (absolute value) of the 1D DCT coefficients was used to predict sparsity for each image. The DCT needs to compact the energy in an image to a few coefficients, and the median magnitude of these coefficients is a good proxy for the success of the compaction. The median magnitude will be high for energy that was not successfully compacted. This algorithm was compared with another simple algorithm based on luminance standard deviation [15]. The images are listed in Table 3.1 along with their values for standard deviation, median magnitude and image quality. The image quality results are from the study in [8], where images were compressed at 10% using a DCT.

The median magnitude and standard deviation can predict recovery performance. The standard deviation has a stronger Pearson correlation with the image quality of 0.92 than the median magnitude of 0.91. However, the median magnitude is less computationally expensive than the standard deviation.

The standard deviation has a time complexity of $\mathcal{O}(2n)$ and expensive mathematical operations such as *pow* and *sqrt*.

Table 3.1. Test images and their performance predictions.

File	Standard Deviation	Median Magnitude	Image Quality (dB)
barb.tiff	47.20	340.23	20.89
boat.tiff	52.32	201.51	21.83
france.tiff	77.21	405.57	13.04
goldhill2.tiff	49.23	161.25	23.70
lena2.tiff	47.85	160.60	23.82
library.tiff	89.33	634.27	10.27
mandrill.tiff	42.30	346.45	18.62
mountain.tiff	80.64	757.24	12.25
peppers2.tiff	57.40	193.73	21.19
washsat.tiff	10.31	117.01	29.48
zelda.tiff	40.53	112.32	27.74

The median is computed by first sorting all the values in order of magnitude and then selecting the middle point, or the average of the two closest values to the middle point, for odd and even-sized values. The sorting of the values is the most expensive operation and dominates the computation cost of the median. The best-performing algorithms have a log-linear time complexity, which makes calculating the median over the entire image expensive.

(a) LCG

(b) Original

(c) FYS

(d) Logistic

(e) Tent

Figure 3.6. Qualitative recovery performance from different variants.

3.4.2 Region Sampling

The image was randomly sampled into regions to reduce the computational load while sacrificing the prediction accuracy, shown in Figure 3.7. The median coefficient magnitude was computed for each region, and the results were averaged to predict performance over the entire image.

Figure 3.7. The region sampling.

The sampling was tested on two strategies to see if acceptable outcomes could be achieved. In the first strategy, a block size of 16 $\times$ 16 was used on 11 regions. The second strategy was a block size of 32 $\times$ 32 on 11 regions. The results are shown in Table 3.2. The second strategy gives better results than the first based on correlation with the full-image baseline. The second strategy had a correlation of 0.95, while the first had 0.92. Increasing the block size increases the accuracy of the spatial sampling, while increasing the number of regions enhances the overall representation of textures in the image.

Table 3.2. Region sampling results.

File	Full Image	Sampling Strategy 1	Sampling Strategy 2
barb.tiff	340.23	47.14	79.20
boat.tiff	201.51	40.52	43.75
france.tiff	405.57	38.94	111.26
goldhill2.tiff	161.25	32.23	56.57
lena2.tiff	160.60	28.05	40.37
library.tiff	634.27	64.48	124.73
mandrill.tiff	346.45	63.11	77.20
mountain.tiff	757.24	106.53	132.43
peppers2.tiff	193.73	34.48	34.60
washsat.tiff	117.01	22.08	27.46
zelda.tiff	112.32	19.74	30.97

3.5 EXPERIMENTAL EVALUATION

The compressed sensing experiments were conducted using MATLAB on a personal computer running Windows 10, which had an Intel Core i7 CPU and 16 GB RAM. In order to ensure that the results were representative, a large number of images were utilised during the experiments. The Waterloo Repertoire GreySet2 collection of monochrome images, which includes images of people, landscapes, animals, objects, and posters, was used in 512×512 sizes. The images in the collection have high-frequency textures and low-frequency structures. DCT was used to make each block of 8×8, 16×16, and 32×32 pixel images sparse, and they were then sampled using various measurement matrices. The block size of 8×8 is popular [74, 8] and offers the least memory footprint during sensing, while the other block size were added to see what influence size plays on energy consumption and recovery performance. The BCS-SPL algorithm was used to recover the images. The measurement matrices were evaluated using experiments that graded them at a relatively reduced sampling rate of 10% to determine their most energy-efficient level. In order to account for stochastic deviation in the results, the experiments were repeated five times. The average image quality in the form of PSNR was taken to evaluate the recovery performance, while mathematical analysis was used to evaluate the energy consumption of the matrices.

3.5.1 Peak Signal to Noise Ratio

In order to gauge the effectiveness of compressed sensing, the quality of images will be evaluated using the PSNR metric. Numerous studies have shown that PSNR is a reliable measure for low sampling rates and high distortion [8]. It is a widely used objective image quality measure. PSNR values are expressed in decibels (dB) and correspond to image quality. The mathematical expression for PSNR is provided in Equation (3.9) [73]

$$\begin{aligned} MSE &= \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\rho_r(i,j)-\rho_p(i,j))^2, \\ PSNR &= 10\log\left(\frac{L^2}{MSE}\right). \end{aligned} \tag{3.9}$$

The PSNR has an inverse, nonlinear mathematical relationship to MSE, which is a cumulative squared error of the reference $\rho_r(i,j)$ against the processed image $\rho_p(i,j)$, L is the dynamic range of the intensity.

3.6 CHAPTER SUMMARY

The DPCI matrix was presented to solve the issues faced by a WMSN in an energy-constrained environment. This matrix promises high energy efficiency and high fidelity. The primary advantage of DPCI is that it uses deterministic random numbers that can be computed with low computational overhead.

Different random number generators were evaluated, and criteria were developed for selection. The Tent map chaotic system was selected because it gave the best combination of computational efficiency and statistical soundness.

The time complexity of constructing the matrix was optimised by developing a novel construction algorithm. The algorithm directly assigns sample positions for the matrix for each row instead of random shuffling.

A novel adaptive sampling mechanism was proposed to deal with challenging images. The adaptive sampling mechanism consists of a prediction algorithm for compression performance to allow for increasing the sampling rate to maintain QoS.

Experiments were designed to demonstrate the efficacy of the proposed matrix in addressing the identified problems. In each of the experiments, several trials were used to evaluate the performance of each matrix.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

The quantitative and qualitative recovery performance and energy consumption results are provided in this chapter of the experiments conducted in the previous chapter.

In Section 4.2, the recovery performance is presented in detail. The various measurement matrices and block sizes are covered in the results. In Section 4.3, the energy consumption analysis is presented for all the sensing matrices and block sizes. Section 4.4 provides a summary of the main aspects of the results.

4.2 RECOVERY PERFORMANCE

The quantitative recovery performance of the matrices is measured utilising image quality (PSNR) in Table 4.1, 4.2 and lastly 4.3. These tables capture the performance under block sizes of 8×8, 16×16 and 32×32. Every recovered image is listed with the image quality for all the sensing matrices. On the bottom of each table is the average image quality over all the images to represent the average performance of each measurement matrix. The qualitative performance is shown alongside each table with examples for each of the sensing matrices in Figure 4.3 to 4.9. For each figure, four images are shown that were compressed utilising the Gaussian, DBBD, BPBD and DPCI along with the original image as a benchmark.

4.2.1 Block size 8×8

In Table 4.1, the outcomes for the block size of 8×8 are presented. The best recovery performance was from the Zelda image with 32.19 dB utilising the DBBD matrix, seen in Figure 4.1. The worst performance was from the Library image at 15.13 dB compressed with the DPCI matrix; see Figure 4.2. The DBBD matrix had the highest average at 24.42 dB. The BPBD matrix had the second-highest

performance, with 24.01 dB. The DPCI averaged 22.94 dB, representing the third-best performance. Finally, the Gaussian matrix had the lowest average performance, with 21.84 dB.

Table 4.1. Recovery performance for 8×8.

	PSNR (dB)			
File	Gaussian	BPBD	DBBD	DPCI
barb.tiff	21.62	23.40	23.47	22.63
boat.tiff	24.16	25.32	25.84	24.64
france.tiff	15.65	17.78	18.32	16.89
goldhill2.tiff	25.47	26.71	26.56	26.25
lena2.tiff	25.57	27.51	28.13	26.28
library.tiff	15.58	16.15	16.74	15.13
mandrill.tiff	19.80	20.57	20.67	19.79
mountain.tiff	15.87	16.97	17.44	16.10
peppers2.tiff	25.33	27.16	27.57	25.08
washsat.tiff	22.46	31.33	31.75	29.66
zelda.tiff	28.67	31.23	32.19	29.89
Average	**21.84**	**24.01**	**24.42**	**22.94**

In Figure 4.1, the Zelda images are presented. The images look close in performance, and it is difficult to say which image has the best quality overall objectively. Despite this, quantitatively, there is a clear difference in performance, where there is $\approx$ 1 dB difference between DBBD and BPBD, BPBD and DPCI, and finally, DPCI and Gaussian, in order of decreasing performance.

Figure 4.2 shows the Library images. The BPBD preserved much of the text, and the sub-images have some detail to discern the objects. The Gaussian had blurred a lot of detail, but the structural content was intact. The DBBD had sharp detail, but the text was unrecognisable, and small details were lost to blocking artefacts. The DPCI retained the content of the sub-images but distorted the text. Quantitatively, the DBBD and BPBD images had a comparable performance with less than a 1 dB difference. The Gaussian and DPCI had comparable quantitative performance, which was $\approx$ 1 dB less than the DBBD and less than 1 dB between them.

(a) BPBD

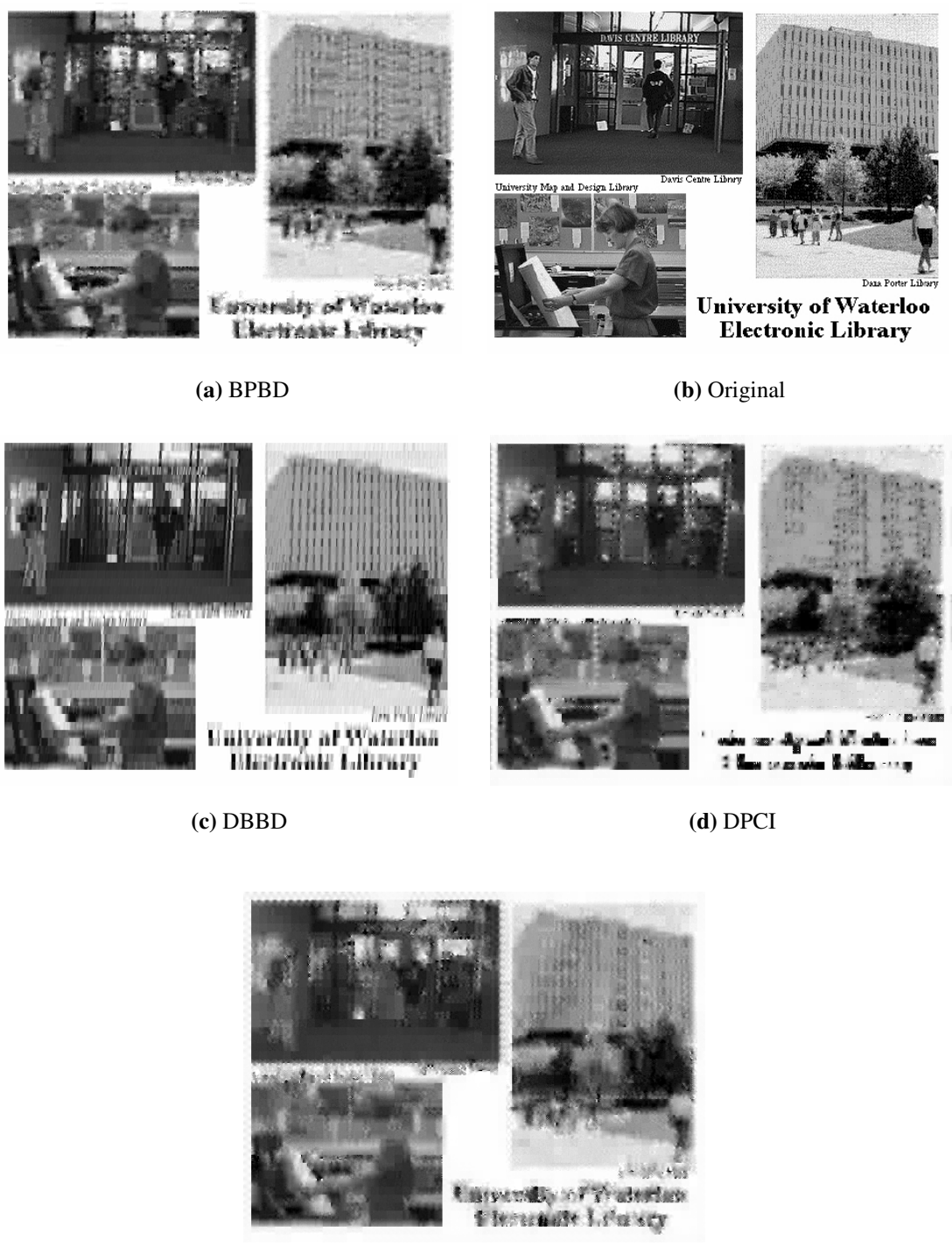
(b) Original

(c) DBBD

(d) DPCI

(e) Gaussian

Figure 4.1. Recovered Zelda images for 8×8.



(a) BPBD

(b) Original

(c) DBBD

(d) DPCI

(e) Gaussian

Figure 4.2. Recovered Library images for 8×8.

(a) BPBD

(b) Original

(c) DBBD

(d) DPCI

(e) Gaussian

Figure 4.3. Recovered Boat images for 8×8.

The recovery of Boat images is displayed in Figure 4.3. Notably, the DBBD exhibited the sharpest detail but with visible pixelation artefacts. The DPCI came in second place with great attention to small details, such as the ship rigging, but some pixelation artefacts were present. The BPBD followed closely with the third highest detail, revealing certain sections of the ship rigging. Meanwhile, the Gaussian method had no pixelation artefacts but produced the lowest level of detail. These findings are supported by Table 4.1.

4.2.2 Block size 16×16

Based on the results presented in Table 4.2, the Zelda image had the highest image quality, achieving compression recovery of 33.36 dB when utilising the DBBD matrix. This can be seen in Figure 4.4. On the other hand, the Library file had the poorest image quality, with compression of 15.65 dB when utilising both the DPCI and Gaussian matrices. This is demonstrated in Figure 4.5. When considering the average performance, the DBBD matrix was found to be the best, achieving 25.28 dB. The BPBD came in second with 24.03 dB, followed by the Gaussian matrix with 23.53 dB, and finally, the DPCI with 23.35 dB.

Table 4.2. Recovery performance for 16×16.

	PSNR (dB)			
File	Gaussian	BPBD	DBBD	DPCI
barb.tiff	22.97	23.19	23.90	23.03
boat.tiff	24.73	25.21	26.89	24.89
france.tiff	16.82	18.02	19.00	16.08
goldhill2.tiff	26.05	26.61	27.99	26.41
lena2.tiff	27.06	27.62	29.39	26.92
library.tiff	15.65	16.01	17.37	15.65
mandrill.tiff	20.08	20.42	21.02	20.00
mountain.tiff	16.74	16.84	17.81	16.19
peppers2.tiff	27.10	27.80	29.09	26.05
washsat.tiff	30.93	31.08	32.28	30.72
zelda.tiff	30.76	31.53	33.36	30.90
Average	**23.53**	**24.03**	**25.28**	**23.35**

Figure 4.4 shows the Zelda images compressed utilising a block size of 16 $\times$ 16. The DBBD image is noticeably better than the other matrices, but the rest of the images had a close performance. The results are corroborated by Table 4.2, where the DBBD was higher than the other matrices by more than 1 dB.

Figure 4.5 shows the Library images. The DBBD had the best performance, the detail is sharp, and some of the text is legible. The DPCI had some sharp details, but smaller objects were smoothed out with unrecognisable text. The BPBD and Gaussian had comparable performance, characterised by blurred details. The results are corroborated by Table 4.2, where the DBBD was higher than the other matrices by more than 1 dB.

The images of GoldHill2 are visible in Figure 4.6. The DBBD showed the highest level of visual detail, but there were some colour distortions. The BPBD and DPCI had similar performance levels. The Gaussian, on the other hand, had the least amount of detail but no distortions. These findings align with the quantitative results in Table 4.2.

4.2.3 Block size 32$\times$32

The results presented in Table 4.3 indicate that the Zelda file provided the best image quality, achieving 32.71 dB when compressed with the DBBD matrix. Figure 4.7 showcases this result. On the other hand, the Library file produced the worst image quality, with only 14.91 dB when compressed utilising the DPCI matrix. This result is illustrated in Figure 4.8. The DBBD matrix demonstrated the highest average performance, delivering 24.73 dB, followed by the BPBD matrix at 23.96 dB. The Gaussian matrix achieved the third-best performance, providing 22.94 dB, while the DPCI matrix performed the worst, with only 22.81 dB.

(a) BPBD

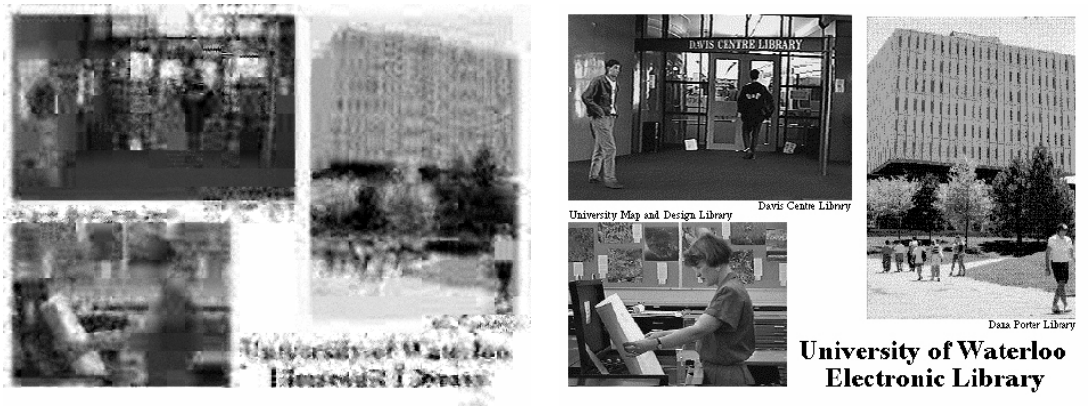
(b) Original

(c) DBBD

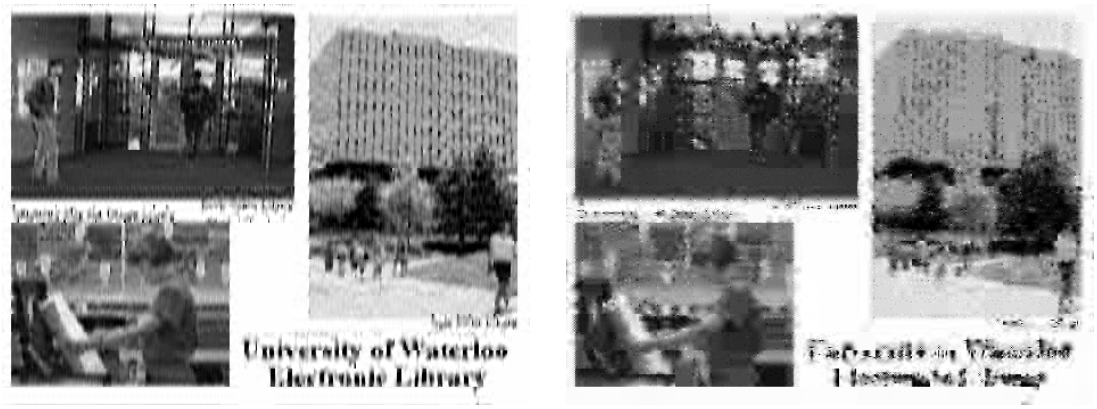
(d) DPCI

(e) Gaussian

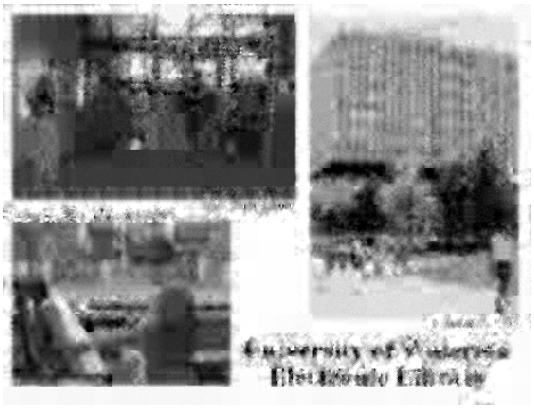
Figure 4.4. Recovered Zelda images for 16×16.



(a) BPBD **(b)** Original



(c) DBBD **(d)** DPCI



(e) Gaussian

Figure 4.5. Recovered Library images for 16×16.

(a) BPBD

(b) Original

(c) DBBD

(d) DPCI

(e) Gaussian

Figure 4.6. Recovered Goldhill2 images for 16×16.

Table 4.3. Recovery performance for 32×32.

	PSNR (dB)			
File	Gaussian	BPBD	DBBD	DPCI
barb.tiff	22.94	23.11	23.56	22.72
boat.tiff	24.84	25.36	26.04	24.49
france.tiff	17.50	17.85	18.86	16.45
goldhill2.tiff	24.93	26.46	27.23	24.62
lena.tiff	26.95	27.77	28.49	26.12
library.tiff	15.72	15.85	17.21	14.91
mandrill.tiff	20.13	20.19	20.50	20.06
mountain.tiff	16.65	16.75	17.33	15.87
peppers2.tiff	26.36	27.85	28.19	25.07
washsat.tiff	30.82	30.89	31.87	30.72
zelda.tiff	30.78	31.48	32.71	29.84
Average	**23.42**	**23.96**	**24.73**	**22.81**

Figure 4.7 shows the Zelda images compressed utilising a block size of 32 × 32. The quantitative results in Table 4.3 are slightly different from the qualitative evaluation, the DBBD still had the best performance, but it was followed by the BPBD, Gaussian and then DPCI. The DBBD performed better than the other matrices. The DBBD image had sharp detail and high contrast. The DPCI also had some sharp detail but had blocking artefacts. The Gaussian and BPBD had a comparable performance with smoothed details.

Figure 4.8 shows the Library images. The DBBD performed the best; the detail was sharp, the text visible, and the letters recognisable. The BPBD had the second-best performance, where some of the text is visible but not recognisable. The DPCI and Gaussian had comparable performances characterised by blurred details. These are analogous to the quantitative results in Table 4.3.

The Mandrill images that have been restored can be seen in Figure 4.9. Among the different techniques used, the DBBD method displayed the most visual detail but had noticeable lines running through the

image. The DPCI and Gaussian methods had comparable performance and displayed the second most amount of detail. On the other hand, the BPBD method produced the worst results with visible blocking artefacts. The findings are consistent with the data presented in Table 4.3, where the performance metrics are fairly close, except for the DBBD method, which stood out.

4.3 ENERGY CONSUMPTION

The energy consumption was evaluated using the MSP430x1xx Family MCU as a reference hardware platform. The most energy-efficient operating region of the MCU is at 1MHz, using a 2.2V supply. The MCU draws a current of 220 μA in the active mode. The instruction cycles needed for mathematical operations are converted to energy consumption in Table 4.4. The most expensive operations are multiplication and division, with the other operations equal in energy cost.

Table 4.4. Energy in joules.

Operation	Energy Consumption (nJ)
Add	0.484
Subtract	0.484
Multiply	14.036
Divide	10.648
Compare	0.484

In order to create the DPCI, one must have a total of M Tent sequence numbers. Sensing is possible with the DPCI without the need for any additions or multiplications, unlike the BPBD and DBBD, which require $N-M$ additions. While the DPCI solely uses energy for construction, the BPBD and DBBD consume energy for both construction and sensing. Table 4.5 provides a breakdown of the energy costs associated with constructing and sensing for all block sizes at a 10% sampling rate. The total energy used for sensing the image is not dependent on the block size but rather on the size of the image itself. In this particular case, the energy used for sensing was 114.2 μJ.

(a) BPBD

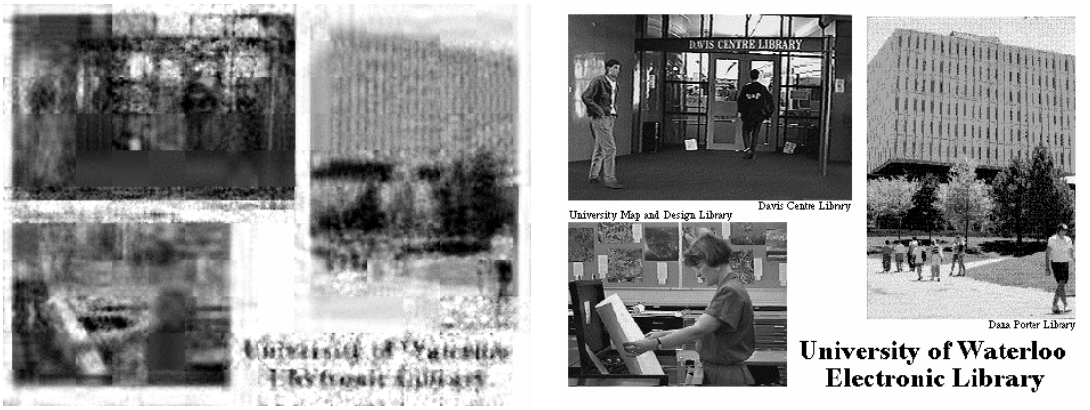
(b) Original

(c) DBBD

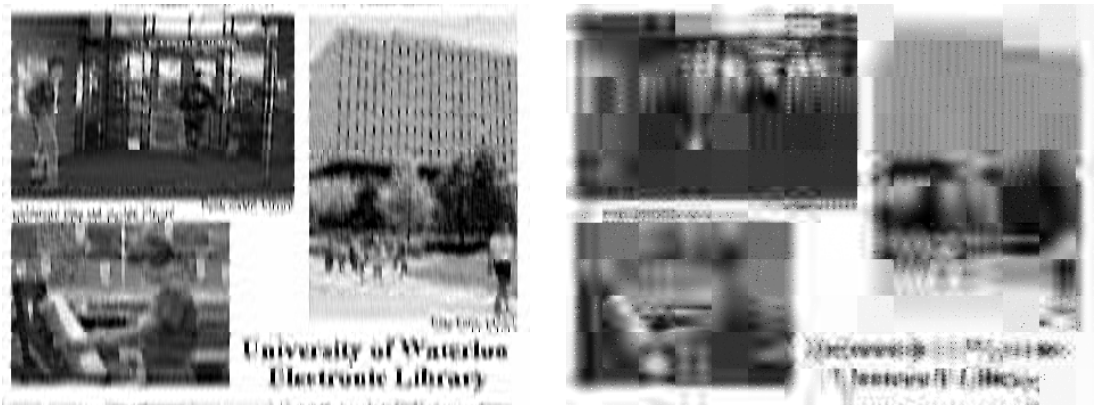
(d) DPCI

(e) Gaussian

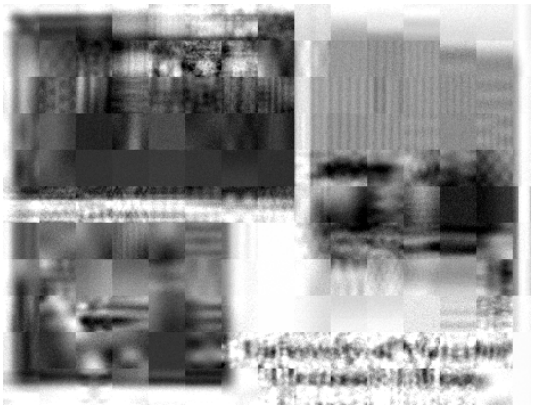
Figure 4.7. Recovered Zelda images for 32×32.



(a) BPBD **(b)** Original

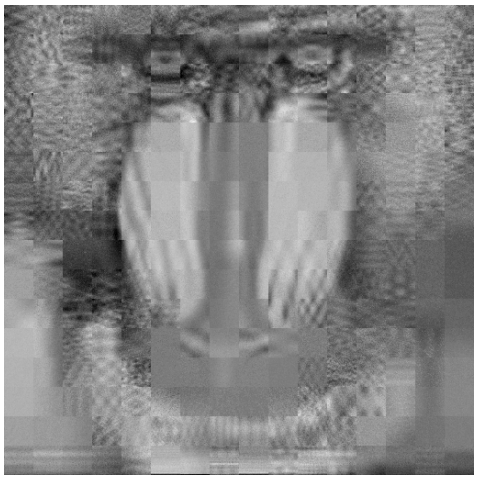


(c) DBBD **(d)** DPCI

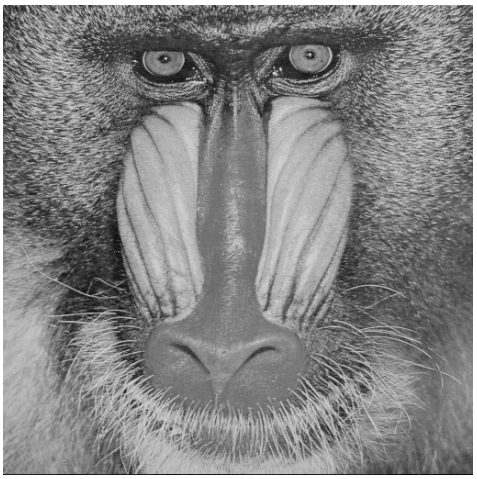


(e) Gaussian

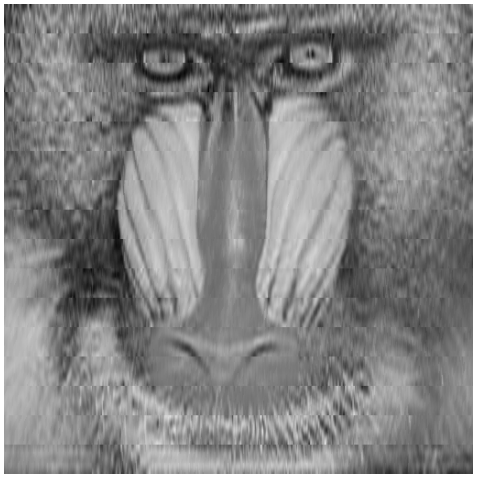
Figure 4.8. Recovered Library images for 32×32.



(a) BPBD

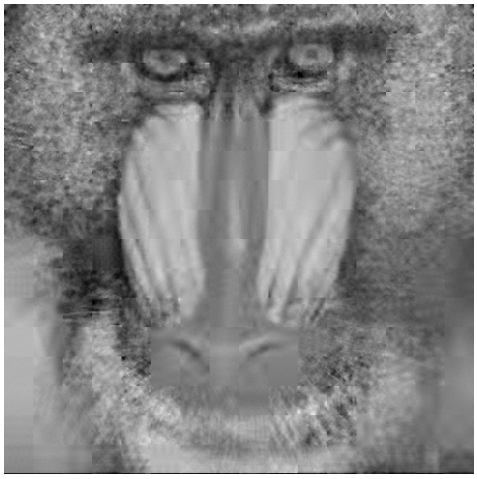


(b) Original



(c) DBBD

(d) DPCI



(e) Gaussian

Figure 4.9. Recovered Mandrill images for 32×32.

Table 4.5. Energy consumption for matrices.

	Energy per block (μJ)			
Process	Gaussian	BPBD	DBBD	DPCI
Constructing 8×8	High	High	0	0.089
Constructing 16×16	High	High	0	0.369
Constructing 32×32	High	High	0	1.506
Sensing 8×8	High	0.028	0.028	0
Sensing 16×16	High	0.112	0.112	0
Sensing 32×32	High	0.446	0.446	0

4.4 CHAPTER SUMMARY

The results of the experimental evaluation were presented. The results were divided into recovery performance and energy consumption.

Recovery performance was evaluated for 8×8 block size using quantitative measures as well as qualitative assessment. The quantitative measures were assessed using PSNR and presented in a table, while the qualitative assessment was carried out by visual inspection of the different images. The same was done for 16×16 and 32×32 block sizes.

Energy consumption was analysed by first converting instruction cycles to energy in Joules using the ideal operating region of the MCU. The influence of block size was analysed on the construction and sensing cost.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

This chapter discusses the results of the experiments conducted in the previous chapter in order to critically evaluate and identify patterns.

In Section 5.2, the recovery performance results for each block size are analysed and summarised. The general pattern across block sizes is then identified and discussed. In Section 5.3, the energy consumption results are analysed and discussed. The influence of block size is discussed for all the evaluated block sizes.

5.2 RECOVERY PERFORMANCE

The recovery performance of all the sensing matrices is discussed in the following subsections. The performance is organised into different block sizes.

5.2.1 Block size 8×8

France, Library, and Mountain were the three images that performed the worst. Across all the measurement matrices, their image quality was below 20 dB. These images present a challenge for compression as they are not easily compacted by most sparsity transforms.

The DBBD had the best quantitative performance, but the qualitative performance was not clear. For some images, the DBBD had a similar performance to the other matrices. In general, the DBBD was sharp but suffered from pixelation artefacts.

The BPBD had the second-best quantitative performance. The qualitative performance was not consistent; the matrix performed second best only on one image where the performance was evaluated. The BPBD preserved the structural content of the images but did not have sharp detail.

The DPCI had the third-best quantitative performance but the second qualitative performance for one image. The DPCI had sharp detail but distorted some of the structural content of the images.

The Gaussian had the worst quantitative and qualitative performance. The Gaussian did not have sharp detail, but the structural content was intact, and there were no distortions.

Of the three qualitatively analysed images, only the Boat image had clear performance outcomes from the different sensing matrices. The Zelda image had performance results that were too similar to discern. The Library image highlighted the strengths and weaknesses of the sensing matrices. However, the overall best performance was subjective.

5.2.2 Block size 16×16

The three images, France, Library and Mountain, still had a much lower performance than the other images with less than 20 dB.

The DBBD had the best quantitative performance and qualitative performance. The DBBD had sharp detail but suffered from colour distortion artefacts.

The BPBD had the second-best quantitative performance. The qualitative performance was not clear. The BPBD preserved the structural content for most of the images but had blurred detail.

The DPCI had the worst quantitative performance, but the difference with the Gaussian is very small. The DPCI had the third-best qualitative performance for one of the images. The DPCI had sharp detail but distorted the structural content of the images.

The Gaussian had the third-best quantitative and qualitative performance. The Gaussian did not have sharp detail, but the structural content was intact, and there were no distortions.

The DBBD was the only sensing matrix that performed qualitatively well for all three images. The Zelda and Goldhill2 images had a similar performance to the remaining sensing matrices. The Library image highlighted the different strengths and weaknesses of the remaining matrices.

5.2.3 Block size 32×32

The DBBD had the best quantitative performance and qualitative performance. The DBBD had sharp detail and high contrast but suffered from distortions, such as lines across the image.

The BPBD had the second-best quantitative performance. The BPBD smoothed details and had blocking artefacts. The qualitative performance was not consistent.

The DPCI had the worst quantitative performance. The DPCI qualitative performance was not consistent between the images. The DPCI had blocking artefacts for one image, sharp detail on the second and smoothed details on the last.

The Gaussian had the third-best quantitative performance. The Gaussian did not have sharp detail, but the structural content was intact, and there were no distortions.

The DBBD was the only sensing matrix that performed qualitatively well for all three images. The Zelda and Goldhill2 images had similar performance for the remaining sensing matrices. The Library image highlighted the different strengths and weaknesses of the remaining matrices.

5.2.4 All block sizes

Three images were particularly challenging for all the measurement matrices for all the block sizes. These were the France, Library and Mountain. However, there were differences between sensing matrices, such as the DBBD almost exceeded 20 dB at 16×16 block size. There was different performance between the different block sizes, and most of the sensing matrices achieved their peak performance at 16×16. The content of these images presents a challenge for compression.

The DBBD had the best quantitative performance for all the block sizes. The matrix generally produced sharp detail in its processed images but would suffer some distortion from pixelation, colour or lines across the image.

The BPBD had the second-highest quantitative performance for all the block sizes. The qualitative performance was not consistent, but generally, the matrix produced smoothed details and, for large block sizes, blocking artefacts.

When it came to a block size of 8 × 8, the DPCI had the third-best quantitative outcome. However, it had similar results to the Gaussian at 16× 16 and inferior results at 32×32. For certain images, the DPCI had the second and third-best qualitative performance. The DPCI typically generated precise features but was vulnerable to distortions such as pixelation and blocking. At 32 × 32, the DPCI had the poorest quantitative performance, and it struggled with the more challenging images, experiencing a comparative decrease in performance of almost 1 dB compared to the Gaussian. Further research is needed to determine if this drop was caused by the statistical distribution of the non-zero values when the sample size, N, is large.

The Gaussian generally performed better than the DPCI on the quantitative measures but had the worst qualitative performance overall. The Gaussian was quite stable in that it never produced any distortions, but the images lacked details.

When it came to recovering images, the impact of block size was inconsistent. The worst performance was demonstrated by the Gaussian when the block size was 8 × 8, but it performed better for 16 × 16 and 32 × 32. The recovery performance of BPBD was comparable across all block sizes, whereas the DBBD and DPCI performed the best at 16 × 16.

5.3 ENERGY CONSUMPTION

Reducing the block size can help to decrease the energy required for construction in DPCI, unlike in DBBD. DPCI does not require any energy for sensing, which distinguishes it from other matrices. Sensing energy is a crucial factor that affects the overall energy consumption of the node. The costs associated with sensing can increase exponentially depending on the sensor's spectral and spatial resolution, as well as the type and duration of events being observed.

5.4 SUMMARY

The results across all the performance metrics are summarised in Table 5.1.

Table 5.1. Summary of all the results.

	Recovery Performance		Energy Consumption (μJ)	
Matrix	Quantitative (dB)	Qualitative	Construction	Sensing
Gaussian 8×8	21.84	Blurred details	High	High
BPBD 8×8	24.01	Smoothing and distortion	High	0.028
DBBD 8×8	24.42	Pixelation and colour distortion	0	0.028
DPCI 8×8	22.94	Pixelation and blocking	0.089	0
Gaussian 16×16	23.53	Blurred details	High	High
BPBD 16×16	24.03	Smoothing and distortion	High	0.112
DBBD 16×16	25.28	Pixelation and colour distortion	0	0.112
DPCI 16×16	23.35	Pixelation and blocking	0.369	0
Gaussian 32×32	23.42	Blurred details	High	High
BPBD 32×32	23.96	Smoothing and distortion	High	0.446
DBBD 32×32	24.73	Pixelation and colour distortion	0	0.446
DPCI 32×32	22.81	Pixelation and blocking	1.506	0

CHAPTER 6 CONCLUSION AND FUTURE RESEARCH

6.1 CHAPTER OVERVIEW

This chapter summarises the results and presents recommendations for future work. The conclusions are presented in respect of the research questions and objectives. Future research recommendations are made based on the challenges encountered during the execution of the research as well as opportunities that need to be pursued.

6.2 CONCLUSION

There recovery performance of semi-deterministic and fully deterministic was comparable. Except for the DBBD, there has not been a significant leap in recovery performance. The DBBD performs significantly better than other sensing matrices and is the deterministic version of the BPBD, but this is an exception, not a general trend.

Computational complexity affects energy consumption in terms of the complexity of the mathematical and logical operations as well as the time complexity of the compressive sensing. These factors are influential when the sensing matrix is constructed and when the signal is sensed. Construction is less important than sensing complexity because it is not a core operation that has to be carried out for each signal. Fully deterministic matrices tend to be less sparse than semi-deterministic matrices because they have been constructed directly from chaotic sequences. The sparsity of the matrices affects the sensing complexity, and this has favoured semi-deterministic matrices. Semi-deterministic matrices have suffered from their reliance on random entries, which makes them expensive to construct, but this problem can be solved by using chaotic matrices, as was done in this study.

The block size affects only the construction cost of the sensing matrices in the case of semi-deterministic

and some fully deterministic matrices. The relationship is that the larger the block size, the higher the construction cost.

The most successful design in the sensing matrix is the BPBD; this matrix was successfully improved even further in the DBBD. This matrix design gives the best recovery performance but suffers one drawback, which is sensing complexity. The matrix requires $N - M$ addition operation for sensing, which places a significant energy draw over time. The most energy-efficient sensing matrix design is the PCI which does not need addition operations for sensing, unlike the BPBD. The PCI had a drawback in that it had high construction costs because of a dependence on random numbers. This was solved in the proposed sensing matrix, the DPCI, by replacing random numbers with chaotic sequences. However, the quality of the random number generator is important for recovery performance, and the Tent map gave the optimal trade-off between even distribution and energy efficiency. Properties of the PCI were exploited to improve the time complexity of the DPCI even further by replacing random permutation with random sample positions.

Some images are not compacted well by sparsity transforms. These images can only be addressed by increasing the sample rate to maintain QoS. There have been many attempts at predicting compression performance for images, but most of these prediction methods are complex. The luminance standard deviation is simple to implement, but an even more simple algorithm was proposed. The coefficient median magnitude is simple to implement, with complexity only stemming from ordering the values, but this prediction method can be further simplified by randomly sampling smaller regions of the image.

6.3 FUTURE RESEARCH

Some images don't perform well during compression; these images need to be studied further to find suitable sparsity transforms that can compact their content. But for now, this type of image can be handled by using adaptive sampling, which needs to be studied further to allocate sample rates efficiently.

The block size affects recovery performance. The effect is not the same for each sensing matrix, but most sensing matrices had a peak performance at the intermediate size. One matrix was unaffected, and another had a peak outside the tested region. This relationship between block size and recovery performance needs to be studied further to understand how this peak operating region can be identified

and exploited for other sensing matrices.

Improving sensing matrix performance through mutual coherence is ineffective for the DPCI, as it remains constant. Further research should focus on improving the recovery performance of the DPCI. This can be done by analyzing sample positions that yield better results through image dataset training. Once the results are obtained, they can be used to develop a universal sensing matrix that can be applied across different datasets.

REFERENCES

[1] V. Skosana and A. M. Abu-Mahfouz, "Energy-efficient sensing matrices for wireless multimedia sensor networks: A review," in *2021 International Conference on Forthcoming Networks and Sustainability in AIoT Era (FoNeS-AIoT)*, 2021, pp. 51–56.

[2] P. Wang, R. Dai, and I. F. Akyildiz, "A spatial correlation-based image compression framework for wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 388–401, 2011.

[3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921 – 960, 2007.

[4] F. Engmann, F. A. Katsriku, J.-D. Abdulai, K. S. Adu-Manu, and F. K. Banaseka, "Prolonging the lifetime of wireless sensor networks: a review of current techniques," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8035065, pp. 1–23, 2018.

[5] S. Pudlewski, A. Prasanna, and T. Melodia, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, 2012.

[6] V. Angayarkanni and S. Radha, "Design of bandwidth efficient compressed sensing based prediction measurement encoder for video transmission in wireless sensor networks," *Wireless Personal Communications*, vol. 88, pp. 553–573, 2016.

[7] R. Banerjee and S. D. Bit, "Low-overhead video compression combining partial discrete

cosine transform and compressed sensing in WMSNs," *Wireless Networks*, vol. 25, no. 8, pp. 5113–5135, 2019.

[8] V. Skosana and A. M. Abu-Mahfouz, "Performance comparison of video encoding at low sampling rates," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, 2021, pp. 1–7.

[9] S. A. Nandhini, R. Sankararajan, and K. Rajendiran, "Video compressed sensing framework for wireless multimedia sensor networks using a combination of multiple matrices," *Computers & Electrical Engineering*, vol. 44, no. C, pp. 51–66, 2015.

[10] S. Yao, T. Wang, W. Shen, P. Shaoming, and Y. Chong, "Research of incoherence rotated chaotic measurement matrix in compressed sensing," *Multimedia Tools and Applications*, vol. 76, no. 17, pp. 17 699–17 717, 2017.

[11] T. Hong, X. Li, Z. Zhu, and Q. Li, "Optimized structured sparse sensing matrices for compressive sensing," *Signal Processing*, vol. 159, no. C, pp. 119–129, 2019.

[12] S. A. Nandhini, S. Radha, P. Nirmala, and R. Kishore, "Compressive sensing for images using a variant of Toeplitz matrix for wireless sensor networks," *Journal of Real-Time Image Processing*, vol. 16, no. 5, pp. 1525–1540, 2019.

[13] H. Gan, S. Xiao, T. Zhang, and F. Liu, "Bipolar measurement matrix using chaotic sequence," *Communications in Nonlinear Science and Numerical Simulation*, vol. 72, pp. 139–151, 2019.

[14] S. Hanumanthu *et al.*, "Universal measurement matrix design for sparse and co-sparse signal recovery," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 404–411, 2021.

[15] J. Zhang, Q. Xiang, Y. Yin, C. Chen, and X. Luo, "Adaptive compressed sensing for wireless image sensor networks," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4227–4242, 2017.

[16] Lu Gan, "Block compressed sensing of natural images," in *2007 15th International Conference on Digital Signal Processing*, 2007, pp. 403–406.

[17] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, 2007, pp. 294–298.

[18] M. Elad, "Optimized projections for compressed sensing," *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, 2007.

[19] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1395–1408, 2009.

[20] V. Abolghasemi, S. Ferdowsi, and S. Sanei, "A gradient-based alternating minimization approach for optimization of the measurement matrix in compressive sensing," *Signal Processing*, vol. 92, no. 4, pp. 999–1009, 2012.

[21] H. Huang and A. Makur, "Optimized measurement matrix for compressive sensing," *Sampling Theory in Signal and Image Processing*, vol. 12, no. 1, pp. 71–86, 2013.

[22] W. Yan, Q. Wang, and Y. Shen, "Shrinkage-based alternating projection algorithm for efficient measurement matrix construction in compressive sensing," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 5, pp. 1073–1084, 2014.

[23] Z. Wei, J. Zhang, Z. Xu, Y. Liu, and K. Okarma, "Measurement matrix optimization via mutual coherence minimization for compressively sensed signals reconstruction," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7979606, pp. 1–18, 2020.

[24] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 809–822, 2016.

[25] I. Ahmed, A. Khan, N. Ahmad, and H. Ali, "Speech signal recovery using block sparse bayesian learning," *Arabian Journal for Science and Engineering*, vol. 45, no. 3, pp. 1567–1579, 2020.

[26] I. Ahmed and A. Khan, "Learning based speech compressive subsampling," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15 327–15 343, 2023.

[27] Zongxin Yu, Rui Wang, Haiyan Zhang, Yanliang Jin, and Yixing Fu, "Distributed compressed sensing for image signals," in *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2014, pp. 1–5.

[28] Y. Arjoune, N. Kaabouch, H. El Ghazi, and A. Tamtaoui, "A performance comparison of measurement matrices in compressive sensing," *International Journal of Communication Systems*, vol. 31:e3576, 2018.

[29] J. Sun, S. Wang, and Y. Dong, "Sparse block circulant matrices for compressed sensing," *IET Communications*, vol. 7, no. 13, pp. 1412–1418, 2013.

[30] X. Su, Y. Hongpeng, C. Yi, X. Yushu, and T. Xue, "An improved Toeplitz measurement matrix for compressive sensing," *International Journal of Distributed Sensor Networks*, vol. 10, no. 6, p. 846757, 2014.

[31] Y. Sun, G. Han, L. Huang, S. Wang, and J. Xiang, "Construction of block circulant measurement matrix based on hybrid chaos: Bernoulli sequences," in *Proceedings of the 2020 4th International Conference on Digital Signal Processing*, ser. ICDSP 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–6.

[32] A. Ravelomanantsoa, H. Rabah, and A. Rouane, "Compressed sensing: A simple deterministic measurement matrix and a fast recovery algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405–3413, 2015.

[33] C.-K. Chang and J. Huang, "Video surveillance for hazardous conditions using sensor networks," in *IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 2, 2004, pp. 1008–1013.

[34] T. Teixeira, D. Lymberopoulos, E. Culurciello, Y. Aloimonos, and A. Savvides, "A lightweight camera sensor network operating on symbolic information," in *Proceedings of the 1st Workshop on Distributed Smart Cameras*, 2006, pp. 76–81.

[35] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In situ image sensing and interpretation in wireless sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '05. New York, NY, USA: Association for Computing Machinery, 2005, pp. 192–204.

[36] R. Kleihorst, B. Schueler, A. Danilin, and M. Heijligers, "Smart camera mote with high performance vision system," in *ACM SenSys 2006 Workshop on Distributed Smart Cameras (DSC 2006)*, 2006, pp. 22–23.

[37] C. Park and P. H. Chou, "ECAM: Ultra compact, high data-rate wireless sensor node with a miniature camera," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, ser. SenSys '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 359–360.

[38] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, ser. IPSN '07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 360–369.

[39] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh, "CMUcam3: an open programmable embedded vision sensor," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-07-13, 2007.

[40] A. Kerhet, M. Magno, F. Leonardi, A. Boni, and L. Benini, "A low-power wireless video sensor node for distributed object detection," *Journal of Real-Time Image Processing*, vol. 2, no. 4, pp. 331–342, 2007.

[41] A. Rowe, D. Goel, and R. Rajkumar, "FireFly Mosaic: A vision-enabled wireless sensor networking system," in *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*,

2007, pp. 459–468.

[42] P. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Y. Yang, C. Yeo, L.-C. Chang, J. Tygar, and S. S. Sastry, "CITRIC: A low-bandwidth wireless camera network platform," in *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1–10.

[43] H. Shi, K. M. Hou, X. Diao, L. Xing, J.-J. Li, and C. De Vaulx, "A wireless multimedia sensor network platform for environmental event detection dedicated to precision agriculture," *arXiv:1806.03237*, 2018. [Online]. Available: https://arxiv.org/abs/1806.03237

[44] A. Redondi, L. Baroffio, L. Bianchi, M. Cesana, and M. Tagliasacchi, "Compress-then-analyze versus analyze-then-compress: What is best in visual sensor networks?" *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 3000–3013, 2016.

[45] D. Singh, S. Sai Prashanth, S. Kundu, and A. Pal, "Low-power microcontroller for wireless sensor networks," in *TENCON 2009 - 2009 IEEE Region 10 Conference*, 2009, pp. 1–6.

[46] T. Sheltami, M. Musaddiq, and E. Shakshuki, "Data compression techniques in wireless sensor networks," *Future Generation Computer Systems*, vol. 64, pp. 151 – 162, 2016.

[47] "TelosB datasheet," Crossbow Technology, San Jose, California, USA, Tech. Rep. 020-0094-01 Rev B, Jan. 2022. [Online]. Available: https://www.willow.co.uk/TelosB_Datasheet.pdf

[48] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 2005, pp. 497–502.

[49] P. Vicaire, T. He, Q. Cao, T. Yan, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. F. Abdelzaher, "Achieving long-term surveillance in VigilNet," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 1–39, 2009.

[50] D. Jung, T. Teixeira, A. Barton-Sweeney, and A. Savvides, "Model-based design exploration of wireless sensor node lifetimes," in *Wireless Sensor Networks*, K. Langendoen and T. Voigt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 277–292.

[51] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[52] C. Caione, D. Brunelli, and L. Benini, "Compressive sensing optimization for signal ensembles in WSNs," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 382–392, 2014.

[53] M. Jones and J. Scott, "The energy efficiency of 8-bit low-power microcontrollers," in *Proceedings of the 18th Electronics New Zealand Conference*, 2011, pp. 87–90.

[54] P. Ruberg, E. Liiv, K. Lass, and P. Ellervee, "Data analysis for embedded software performance and energy consumption estimation," in *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2019, pp. 928–933.

[55] *MSP430x5xx MSP430x6xx Family User's Guide Rev. Q*, Texas Instruments, Dallas, Texas, USA, 2018. [Online]. Available: https://www.ti.com/lit/pdf/slau208

[56] "Efficient multiplication and division using MSP430," Texas Instruments, Dallas, Texas, USA, Tech. Rep. SLAA329A, Sep. 2006. [Online]. Available: https://www.ti.com/lit/pdf/slaa329

[57] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[58] J. D. Blanchard, C. Cartis, and J. Tanner, "Compressed sensing: How sharp is the restricted isometry property?" *SIAM Review*, vol. 53, no. 1, pp. 105–125, 2011.

[59] S. Narayanan, S. K. Sahoo, and A. Makur, "Greedy pursuits assisted basis pursuit for compressive sensing," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 694–698.

[60] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, 1994, pp. 41–44.

[61] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, 2008, pp. 581–587.

[62] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[63] A. L. Pilastri and J. M. R. Tavares, "Reconstruction algorithms in compressive sensing: An overview," in *11th edition of the Doctoral Symposium in Informatics Engineering (DSIE-16)*, 2016, pp. 127–137.

[64] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, vol. 9, pp. 317–334, 2009.

[65] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[66] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[67] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[68] R. M. Gibson, A. Amira, N. Ramzan, P. C. de-la Higuera, and Z. Pervez, "Matching pursuit-based compressive sensing in a wearable biomedical accelerometer fall diagnosis device," *Biomedical Signal Processing and Control*, vol. 33, pp. 96–108, 2017.

[69] D. L. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[70] S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 3021–3024.

[71] K. Mechouek, N. Kouadria, N. Doghmane, and N. Kaddeche, "Low complexity DCT approximation for image compression in wireless image sensor networks," *Journal of Circuits, Systems and Computers*, vol. 25, no. 08, p. 1650088, 2016.

[72] C. Araar, S. Ghanemi, M. Benmohammed, and E. Bourennane, "Low complexity image compression using pruned 8-point DCT approximation in wireless visual sensor networks," in *2017 International Conference on Mathematics and Information Technology (ICMIT)*, 2017, pp. 279–285.

[73] C. Araar, S. Ghanemi, M. Benmohammed, and H. Atoui, "Pruned improved eight-point approximate DCT for image encoding in visual sensor networks requiring only ten additions," *Journal of Real-Time Image Processing*, vol. 17, no. 5, pp. 1597–1608, 2020.

[74] N. Kouadria, K. Mechouek, D. Messadeg, and N. Doghmane, "Pruned discrete Tchebichef transform for image coding in wireless multimedia sensor networks," *AEU - International Journal of Electronics and Communications*, vol. 74, pp. 123 – 127, 2017.

[75] V. A. Coutinho, R. J. Cintra, F. M. Bayer, P. A. Oliveira, R. S. Oliveira, and A. Madanayake, "Pruned discrete Tchebichef transform approximation for image compression," *Circuits, Systems, and Signal Processing*, vol. 37, no. 10, pp. 4363–4383, 2018.

[76] D. F. Coelho, R. J. Cintra, F. M. Bayer, S. Kulasekera, A. Madanayake, P. Martinez, T. L. Silveira, R. S. Oliveira, and V. S. Dimitrov, "Low-complexity Loeffler DCT approximations for image and video coding," *Journal of Low Power Electronics and Applications*, vol. 8, no. 4, p. 46, 2018.

[77] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *International Conference on Acoustics, Speech, and Signal Processing,*, vol. 2, 1989, pp. 988–991.

[78] D. Nister and C. Christopoulos, "An embedded DCT-based still image coding algorithm," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 5, 1998, pp. 2617–2620.

[79] S. A. Khayam, "The discrete cosine transform (DCT): theory and application," *Michigan State University*, vol. 114, no. 1, p. 31, 2003.

[80] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2210–2219, 2008.

[81] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, pp. 57–98, 1997.

[82] M.-R. Luo and S.-W. Zhou, "Adaptive wavelet packet image compressed sensing," *Journal of Electronics and Information Technology*, vol. 35, no. 10, pp. 2371–2377, 2013.

[83] R.-F. Wang, L.-C. Jiao, F. Liu, and S.-Y. Yang, "Block-based adaptive compressed sensing of image using texture information," *ACTA ELECTONICA SINICA*, vol. 41, no. 8, p. 1506, 2013.

[84] R. Li, X. Duan, X. Li, W. He, and Y. Li, "An energy-efficient compressive image coding for green internet of things (IoT)," *Sensors*, vol. 18, no. 4, 2018.

[85] Z. Zhang, H. Bi, X. Kong, N. Li, and D. Lu, "Adaptive compressed sensing of color images based on salient region detection," *Multimedia Tools and Applications*, vol. 79, pp. 14 777–14 791, 2020.

[86] R. Monika, D. Samiappan, and R. Kumar, "Underwater image compression using energy based adaptive block compressive sensing for IoUT applications," *The Visual Computer*, vol. 37, pp. 1499–1515, 2021.

[87] J. Bigot, C. Boyer, and P. Weiss, "An analysis of block sampling strategies in compressed sensing," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 2125–2139, 2016.

[88] X. Sui, Q. Chen, G. Gu, and X. Shen, "Infrared super-resolution imaging based on compressed sensing," *Infrared Physics and Technology*, vol. 63, pp. 119 – 124, 2014.

[89] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[90] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[91] S. Hong, H. Park, B. Shin, J. No, and H. Chung, "A new performance measure using k-set correlation for compressed sensing matrices," *IEEE Signal Processing Letters*, vol. 19, no. 3, pp. 143–146, 2012.

[92] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, 2011.

[93] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 139–154, 2012.

[94] J. E. Fowler, S. Mun, and E. W. Tramel, "Multiscale block compressed sensing with smoothed projected Landweber reconstruction," in *2011 19th European Signal Processing Conference*, 2011, pp. 564–568.

[95] L. Quan, S. Xiao, X. Xue, and C. Lu, "Semi-deterministic sparse matrix for low complexity compressive sampling." *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 5, pp. 2468–2483, 2017.

[96] Z. He, T. Ogawa, and M. Haseyama, "The simplest measurement matrix for compressed sensing of natural images," in *2010 IEEE International Conference on Image Processing*, 2010, pp. 4301–4304.

[97] R. Sun, H. Zhao, and H. Xu, "The application of improved Hadamard measurement matrix in compressed sensing," in *2012 International Conference on Systems and Informatics (ICSAI2012)*, 2012, pp. 1994–1997.

[98] N. Jain, A. Gupta, and V. A. Bohara, "PCI-MDR: Missing data recovery in wireless sensor networks using partial canonical identity matrix," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 673–676, 2019.

[99] N. Ansari and A. Gupta, "Image reconstruction using matched wavelet estimated from data sensed compressively using partial canonical identity matrix," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3680–3695, 2017.

[100] N. Jain, V. A. Bohara, and A. Gupta, "PCI-MF: Partial canonical identity and matrix factorization framework for channel estimation in mmwave massive MIMO systems," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 135–145, 2020.

[101] P. A. W. Lewis, A. S. Goodman, and J. M. Miller, "A pseudo-random number generator for the System/360," *IBM Systems Journal*, vol. 8, no. 2, pp. 136–146, 1969.

[102] S. Phatak and S. Rao, "Logistic map: A possible random-number generator," *Physical review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 51, no. 4, pp. 3670–3678, 1995.

[103] L. A. Demidova and A. V. Gorchakov, "A study of chaotic maps producing symmetric distributions in the fish school search optimization algorithm with exponential step decay," *Symmetry*, vol. 12, no. 5, 2020.

[104] F. P. Juniawan, H. A. Pradana, Laurentinus, and D. Y. Sylfania, "Performance comparison of linear congruent method and Fisher-Yates shuffle for data randomization," *Journal of Physics:*

Conference Series, vol. 1196, no. 1, p. 012035, 2019.

[105] University of Waterloo, "Waterloo Repertoire GreySet2," University of Waterloo Links http://links.uwaterloo.ca/oldwebsite/greyset2.base.html, (accessed Oct. 13, 2020).

APPENDIX A DATASETS

A.1 THE WATERLOO REPERTOIRE GREYSET2

The selected images from the University of Waterloo [105] are shown in high resolution from Figure A.1 to A.11. These images are popular in the image processing literature. The image and their subject are described in Table A.1.

Table A.1. Selected Waterloo Repertoire Greyset2 images.

File	Description
barb.tiff	A person is photographed sitting on the floor with their face and most of the body visible.
boat.tiff	A small grounded sailboat is captured with a person standing next to it.
france.tiff	Travel poster with smooth background and text.
goldhill2.tiff	The image captures conjoined houses built up a hill, and there is a person walking down the hill.
lena2.tiff	This image is a portrait of a person at close range and captures mostly the face.
library.tiff	This is a poster with three subfigures; each subfigure has people and has small and large text.
mandrill.tiff	The image is a close-up of the face of a mandrill.
mountain.tiff	The image is a landscape of a mountain range with a body of water also visible.
peppers2.tiff	This is a close-up image of peppers.
washsat.tiff	This is a satellite image of a city where there are structures from streets.
zelda.tiff	This is a portrait of a person where the background is out of focus and the face dominates the frame.

Figure A.1. The barb image.

Figure A.2. The boat image.

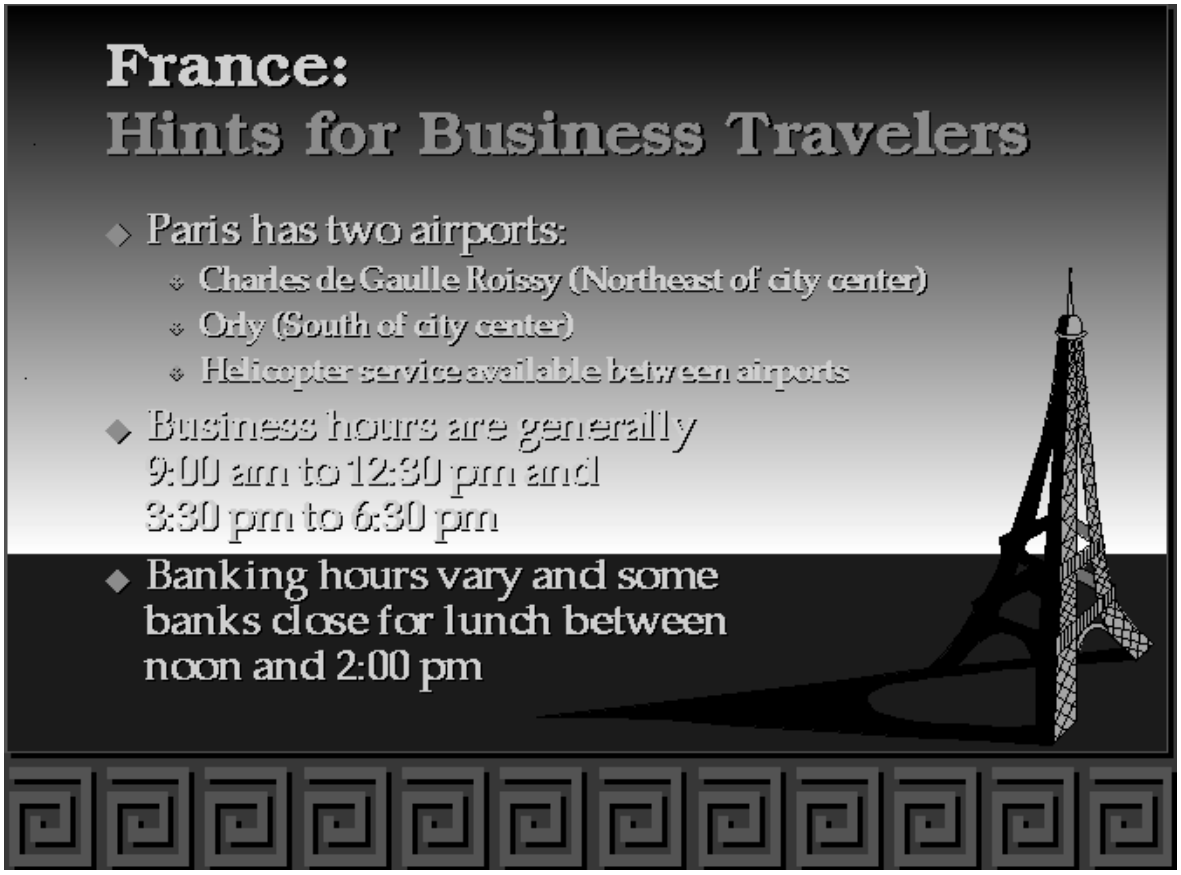


Figure A.3. The france image.

Figure A.4. The goldhill2 image.

Figure A.5. The lena2 image.

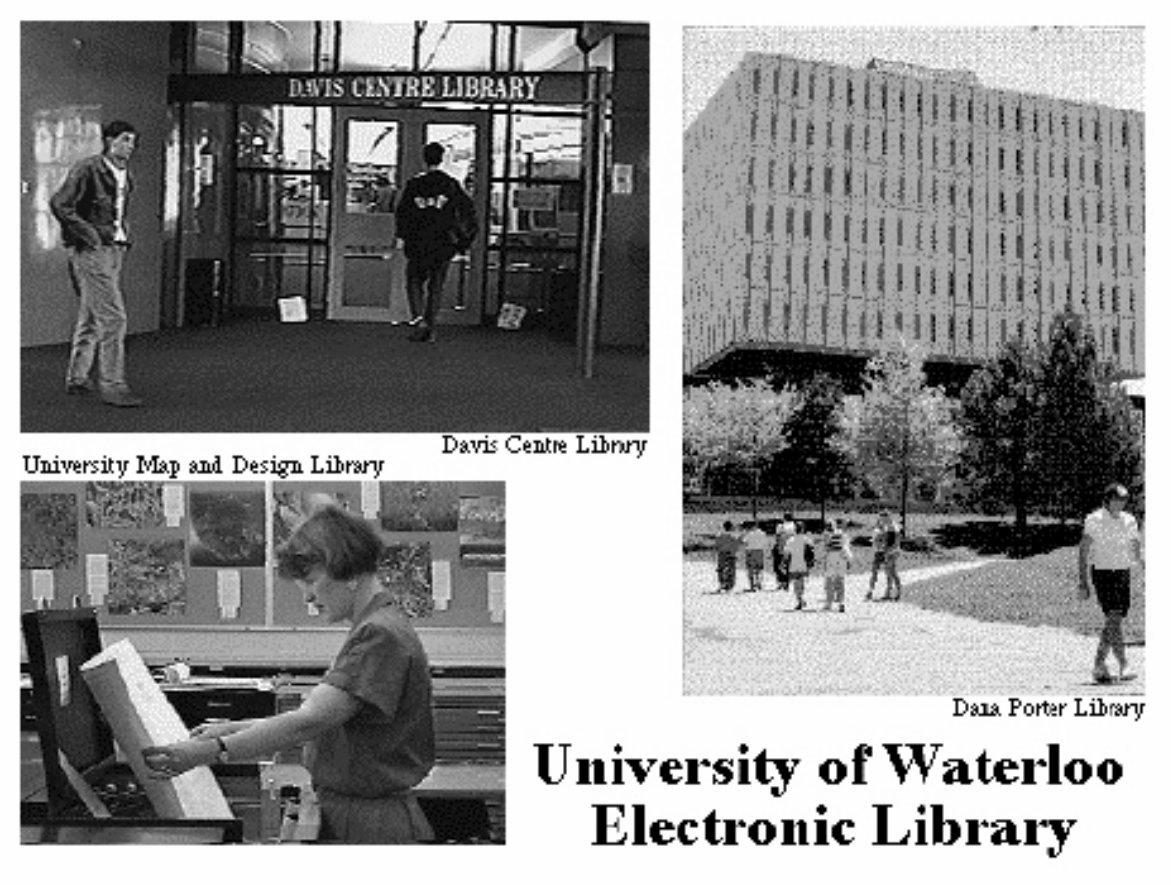


Figure A.6. The library image.

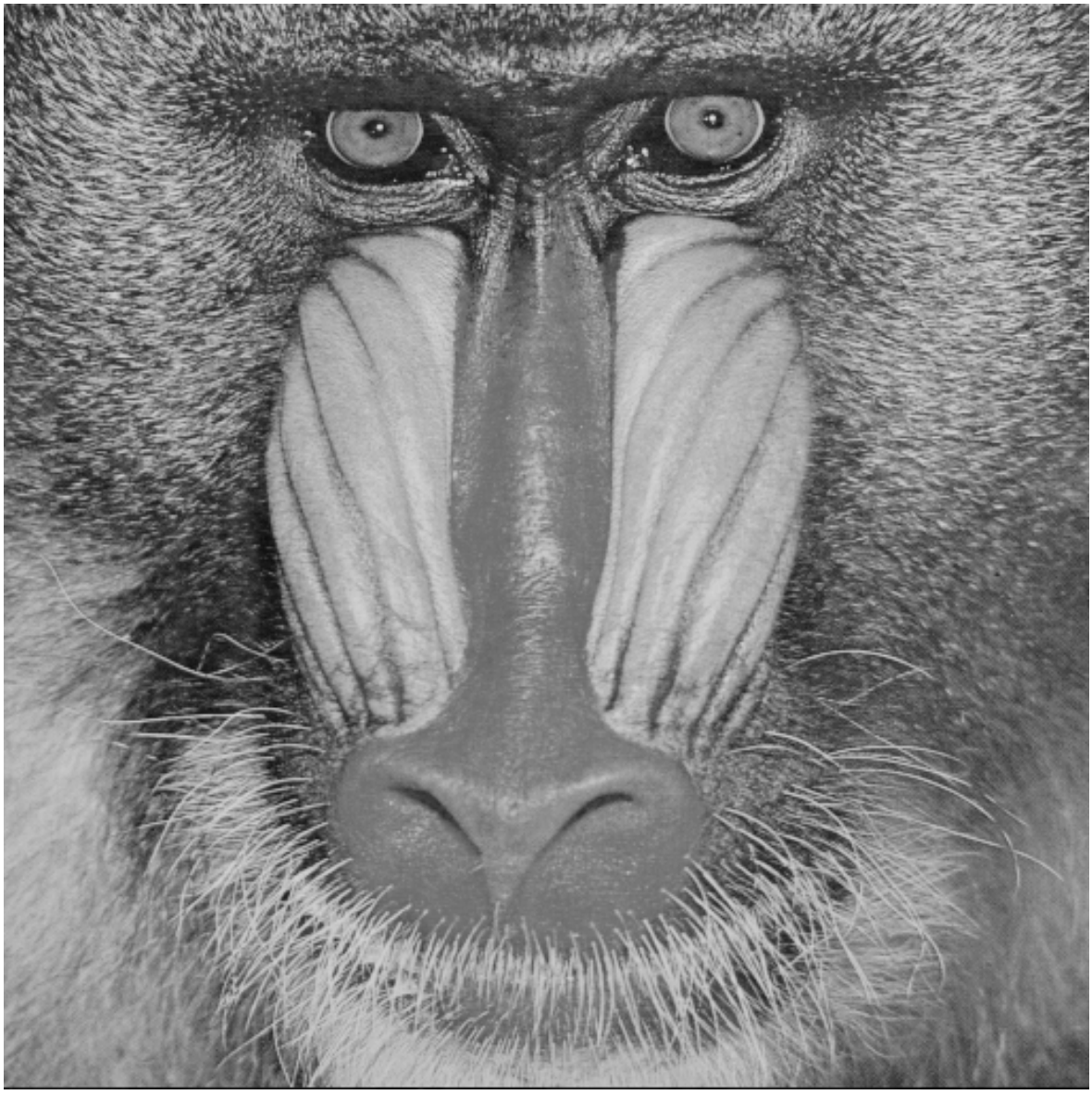


Figure A.7. The mandrill image.

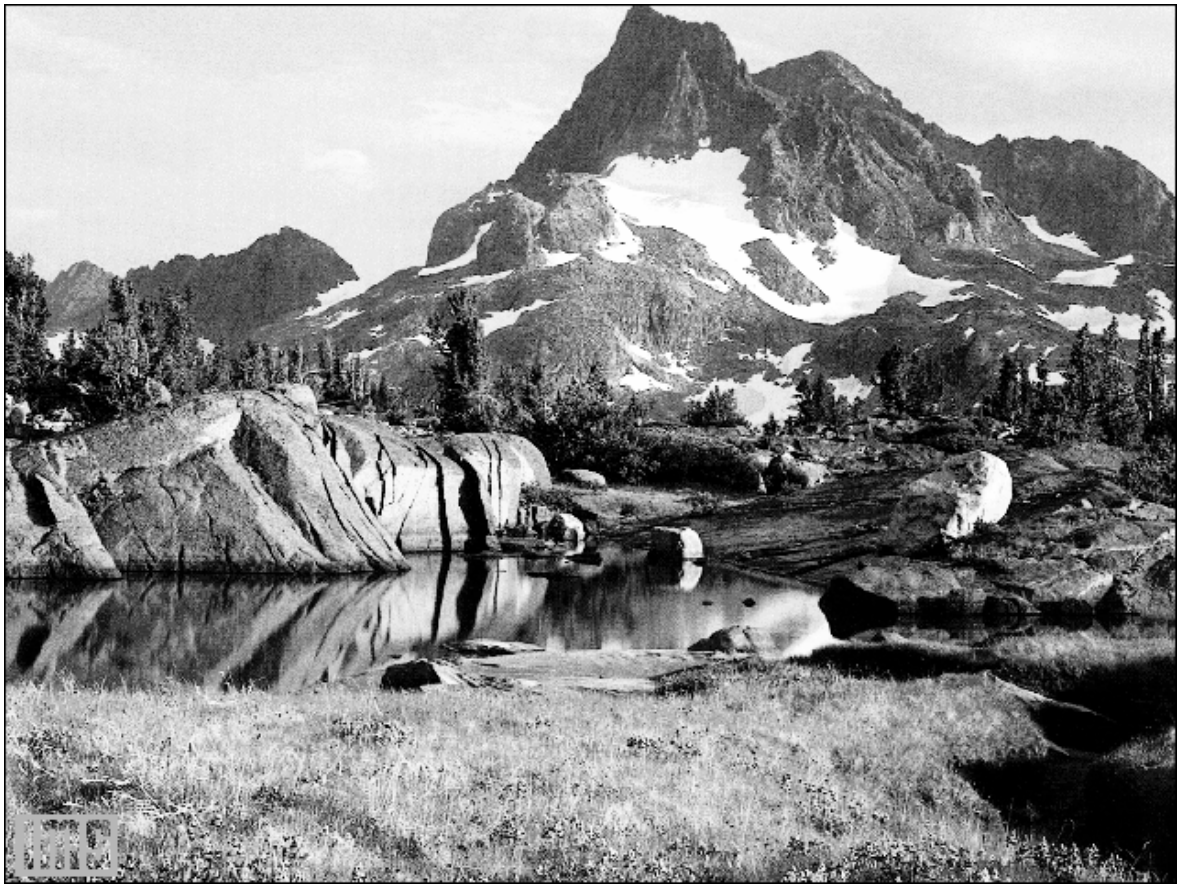


Figure A.8. The mountain image.

Figure A.9. The peppers2 image.

Figure A.10. The washsat image.

Figure A.11. The zelda image.