


Fall 12-19-2017

Algorithm for Computational Imaging on a Real-Time Hardware

Manish Bhattarai
University of New Mexico

Follow this and additional works at: https://digitalrepository.unm.edu/ece_etds

 Part of the [Signal Processing Commons](#), and the [VLSI and Circuits, Embedded and Hardware Systems Commons](#)

Recommended Citation

Bhattarai, Manish. "Algorithm for Computational Imaging on a Real-Time Hardware." (2017). https://digitalrepository.unm.edu/ece_etds/403

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.

Manish Bhattarai

Candidate

Electrical and Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Majeed M. Hayat , Chairperson

Dr. Payman Zarkesh-Ha

Dr. Marios Pattichis

Algorithm for Computational Imaging on a Real-Time Hardware

by

Manish Bhattarai

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2018

Dedication

To my lord, ever-loving grandmother, my beloved parents and wonderful sisters

Acknowledgments

Firstly, I would like to thank my advisor, Prof. Majeed Hayat for his constant support and guidance. I really feel honored to have had the opportunity to work under your guidance. Without your constant encouragement, it would not have been possible to successfully complete my Master's thesis.

Thank you also to my committee members, Prof. Payman Zarkesh Ha and Prof. Marios Pattichis for their encouragement and kindness.

I would like to thank my parents and sisters for their endless love and support which aided me to accomplish most of the wishes in my life.

Finally, I would also like to thank my colleagues and friends, Javad Ghasemi, Abraham P. Vinod, Anees Abrol, Pankaz Das and Joseph Gleason for all your help, advise and friendship. Also big thanks to my dear friend Esmeralda for her motivation that inspired me to build passion and love for the research. Thank you all for easing my difficulties and giving me support in tough times. I also want to thank other group members from ECE and CHTM research groups.

Algorithm for Computational Imaging on a Real-Time Hardware

by

Manish Bhattarai

M.S., Electrical Engineering, University of New Mexico, 2018

Abstract

DRAMATIC advances in the field of computational and medical imaging over the past decades have enabled many critical applications such as night vision, medical diagnosis, quality control, and remote sensing. The increasing demand for image quality and its fidelity requires an increase in pixel count and a sophisticated post-processing mechanism to efficiently store, transmit, and analyze this massive data. There is an inherent trade-off between the generation of big data by such imaging systems and efficiency in extraction of useful information within real-time constraints, limiting the efficacy of such sensors in real-time decision-making systems. The traditional imaging system gets burdened by the acquisition, transmission, and storage of surplus data, often bearing redundant information for the given application of interest. Transmission of the irrelevant information requires a high bandwidth and results in consuming extra power to store or transmit. Similarly, post-processing imposes extra latency and suffers under the power constraint, which is troublesome for many low-power, real-time applications, and portable devices.

There is a need to address this problem by intelligently acquiring a limited but most important set of data features, and then efficiently processing this abstract information. This, in turn, needs an additional ability of performing computations at

the pixel level, within the readout integrated circuit at the front-end of the imager. The most challenging job in this computational part is devising the mechanism to properly select the right set of data features for adequately representing the information which might be corrupted with noise and therefore potentially increase the reconstruction error.

In this thesis, we work towards development of an efficient bias selection algorithm to configure the computational imaging hardware. This algorithm not only fulfills the fundamental hardware requirements of memory efficiency, low power consumption and minimal latency, but also results in lesser reconstruction error than the previously used naive approach. This algorithm supports compression during the data acquisition time at the pixel level thus enabling compression even in a noisy environment. This work was carried out in joint collaboration and supervision of two pioneers, Prof. Majeed Hayat and Prof. Paymen Zarkesh-Ha, under whom an integrated hardware and algorithm imaging concept called compressed domain imaging device was developed. The hardware used in this study was developed under the supervision of Prof. Payman Zarkesh-Ha by Dr. Javad Ghasemi, a graduate student from his lab. Details of this “Readout Integrated Circuit” (ROIC) based imaging hardware are presented in Dr. Ghasemi’s Ph.D. dissertation. The developed hardware enables compressed-domain image acquisition by means of projecting the input image onto a series of coded apertures, or spatial intensity masks, that are stored in the ROIC. The real-time, pixel-level projection is achieved by the means of designing a set of coded apertures and programming them in the ROIC as a 2D array of analog biases. The stored bias values at each pixel govern the operating point of the photodetector, thereby affecting the spatial intensity modulation of the image. The pixel values are then summed up within the ROIC. The output of this unique camera is therefore the features (or code-words) arising from the projection of the image onto each of the prescribed coded apertures. Hence, the camera does not generate and store the actual image at any point. The image can be reconstructed using

a reconstruction algorithm that employs the features computed by the camera.

However, due to the limited dynamic range and sensitivity of the hardware towards the noise, the overall performance could degrade significantly during computation of the Discrete Cosine Transform (DCT) and Compressed Sensing (CS) coefficients for data compression; hence the significance and applicability of the unique hardware could diminish considerably at times. This specific type of hardware limitation has opened the scope for this thesis work, where we have performed theoretical and experimental systematic study of the hardware by modeling its response and characterizing the noise statistically. Thus, in this work we develop a robust Minimum Mean Square Error (MMSE) based bias selection algorithm to overcome the effect of noise and formally demonstrate how use of superior signal processing techniques on the statistical model of the hardware response enhances the performance of computational imaging hardware systems.

Contents

List of Figures	x
Glossary	xiii
1 Introduction	1
1.1 Overview	1
1.2 History of Computational Imaging Hardware	4
1.3 Concept of Programmable Imaging	7
1.4 Extension to Compressed Domain Hardware	7
1.5 Publications	9
2 Experimental setup	10
2.1 Novelty of Hardware	16
2.1.1 Scalability to large-format arrays	16
2.1.2 Proof of concept for real-time implementation of front-end computational imaging	16

3	Compressed-domain image acquisition	17
3.1	Discrete cosine transform	17
3.1.1	Bias selection algorithm	19
3.1.2	Conditioning the masks for mapping the bias into device dynamic range	23
3.1.3	DCT-based image compression	25
3.1.4	DCT-based image reconstruction	25
3.2	Block based Transform Coding	26
3.3	Reduction of Blocking Effect in Image Coding	27
3.4	Reconstruction Noise Analysis	27
4	Compressive sensing	29
4.1	Introduction to Compressed Sensing Theory	30
4.2	Background	32
4.3	Early CS Imaging Hardware	35
4.4	Implementation	38
5	Results	41
5.1	Performance comparison between naïve DCT, LMS DCT, and CS reconstruction	41
6	Computational Imaging and Non-Uniformity Correction	45
6.1	Non-Uniformity Correction	45

<i>Contents</i>	ix
6.2 Functioning as a stand-alone camera	51
6.3 Region of interest (ROI) enhancement	52
7 Conclusion	55

List of Figures

1.1	a) A system-level block diagram of a conventional imaging system, which includes image acquisition, storage, and post-processing stages. ■	
	b) Block diagram of the intelligent readout integrated circuit we propose for on-chip image acquisition and compression.	2
1.2	Optical Architecture of the Single pixel compressive hyperspectral imaging sensor (SPHIS)	4
1.3	Illustration of Computational Imaging	5
1.4	Implemented Programmable Imaging Hardware as per Javad's work	6
1.5	Scheme of a CMOS Compressed Imager.	8
2.1	a) A microphotograph of the fabricated ROIC, the row, and column select, and the test devices. The unit cell is shown in the extended view. b) A block diagram of the experimental setup, which includes a Raspberry Pi board as the main controller of the system, an ADC, and a DAC to set the bias voltage of the detectors and grabs the readout of the imager. All communication between the controller and a remote machine is over SSH.	11

2.2	Block diagram of the individual pixel bias tunable readout integrated circuit and the CTIA-based unitcell at the extended view. The extra circuitry added to the CTIA-based unitcell enables setting independent bias voltages for each individual pixel while the previously integrated voltage is being read out.	11
2.3	Demonstration of the normalized modulation function of the system to a uniform illumination level. The graph reflects the system's response to the modulation of the detector's bias.	15
3.1	Acquisition and compression processes, which include mapping k mask matrices to their corresponding bias voltages. The mapping is based on the system's response-modulation function shown in Fig. 2.3. Then the bias matrices that are stored in the Raspberry Pi memory are loaded to the imager and projected to the object's reflectance function. The resultant dot product is optionally summed up in the hardware, and the k resulting coefficients are sent to the remote computer for reconstruction.	20
3.2	a) Distribution of 8×8 block-based DCT mask coefficients for naïve method, b) distribution of bias for naïve method, c) distribution of 8×8 block-based DCT mask coefficients for MMSE method, and d) distribution of bias for MMSE method.	24
4.1	Rice Single pixel Camera	35
4.2	Overview of CS imaging hardware	37

5.1	The resulting images reconstructed using a) naïve DCT, b) minimum-mean-square error based DCT, c) compressive sensing, and d) ideal DCT. e) The performance of different method is compared in terms of the mean square error between the reconstructed image and the original image.	44
6.1	a) The result of imaging a white paper with uniform biasing, while the illumination is not uniform. Defects and other sources of nonuniformity also contribute to the variation across the image. The stack of three graphs demonstrates (I) camera output image, (II) illumination contour, and (III) 3D view of the intensities. b) Another white paper is imaged with the same illumination condition using the implemented nonuniformity correction. The graph has the same scale as part (a), and the legend in the middle is for part (II). c) and d) show the histogram for the measured results of part (a) and (b), respectively.	50
6.2	Four images that are taken using iROIC camera in normal mode. a) phantom, b) a cell, c) some rice grains, and d) UNM logo.	52
6.3	a) Original white matter image used for imaging. b) The image is taken using iROIC with a uniform biasing for all of the pixels where some of the pixels are saturated due to the high intensity. In c), d), e), and, f) the same scene is imaged using proper biasing for the different areas that normally are at the noise floor of the imager. . .	54

Glossary

DCT	Discrete Cosine Transform
CS	Compressed Sensing
ROIC	Read-Out Integrated Circuit
CMOS	Complementary metaloxidesemiconductor
I-ROIC	Intelligent ROIC
BW	Bandwidth
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
SNE	Signal to Noise Ratio

Chapter 1

Introduction

1.1 Overview

The big-data problem has introduced many challenges to efficient acquisition, storage, and transmission of data. Usually, most of the current imaging modalities are based on compressing the data after the acquisition, wherein the acquired data stored in memory are projected into basis functions corresponding to compression domain. However, in real-time applications the system's performance degrades when there are constraints in resources, such as power, memory, and speed [23].

The conventional imaging system suffers from inefficient data transmission, additive latency, and large power consumption, which is not desired in real-time and medical applications [18] [27] [12] [5]. If one is able to achieve the compression of data at the acquisition phase itself, rather than undergoing post-processing, and compression of big data, it will result in the saving of power, time, and the efficient usage of the hardware resources [26] [9]. Compressive image acquisition relaxes the requirements dictated by the Nyquist theorem and gives flexibility in representing information with fewer projection coefficients under the assumption that the original

signal is sparse in some domains [8]. Furthermore, instead of reading the values of intensity sampled at different pixels, in the case of a compressive-sampling sensor, a set of orthogonal gain matrices are loaded to the pixel array and the image sensor output would be directly an inner product between gain matrices, and the sample values of the image.

To address the big data problem associated with the computational imaging hardware, we propose an efficient system that shifts the tradition of grabbing and transmitting raw images, to an efficient compressed-domain with the aid of proper bias selection algorithm to the computational hardware. We have demonstrated an integrated hardware-and-algorithm imaging concept, termed the filterless, compressed-data imager, which is designed to sense only the important spatial information while directly outputting only key coefficients (compressed data) that best represent a scene subject to prescribed performance criteria. These coefficients can, in turn, be efficiently stored/transmitted and used by an operator for analysis or, if so desired, to reconstruct the spatial scene (represented by an original image). What is also unique in this imager is that it does not employ any physical spatial filters, and unlike traditional imagers, it does not generate massive amounts of data at any point.

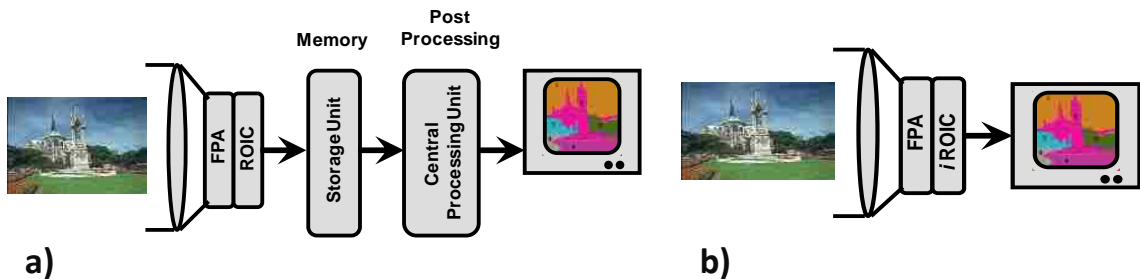


Figure 1.1: a) A system-level block diagram of a conventional imaging system, which includes image acquisition, storage, and post-processing stages. b) Block diagram of the intelligent readout integrated circuit we propose for on-chip image acquisition and compression.

Source: NSF Proposal document by Hayat and Zarkesh-Ha

Additionally, all the computations are performed at the pixel level within the readout circuit, at the front-end of the imager, without any need of large-data transmission and post-processing. As a secondary application of the concept, we will develop the first DCT and CS coefficients generator, where the imager outputs directly the transform coefficients associated with the image, instead of outputting massive amounts of raw data. This is achieved by designing the necessary bias masks that are relevant to the application and performing the necessary calculations in the analog domain within the readout circuit.

An intelligent bias-selection algorithm is presented for the problem of front-end computational imaging at the read-out-circuit level. The algorithm exploits the bias-dependent nature of the imagers responsivity to implement series of coded apertures. A visible-light imager with controllable readout is used to demonstrate the algorithm.

The front-end computational imaging is performed by the means of the inner product between an image and an arbitrarily-specified mask in silicon. The acquisition system is based on an intelligent readout integrated circuit (iROIC) that is capable of providing independent bias voltages to individual detectors, which enables implementation of spatial multiplication with any prescribed mask through a bias-controlled response-modulation mechanism. The modulated pixels are summed up in the image grabber to generate the compressed samples, namely aperture-coded coefficients, of an image. A rigorous bias-selection algorithm is presented to the readout circuit, which exploits the bias-dependent nature of the imagers responsivity. Based on the physical, nonlinear responsivity of the device and its noise characteristics, the algorithm is designed to prescribe the control voltages, for each mask to be realized electronically while minimizing the mean-squared-error.

Before proceeding to the major discussion, we would like to give some relevant background and prior works devoted to this field in the past.

1.2 History of Computational Imaging Hardware

Computational imaging refers to digital image capture and processing techniques that use digital computation instead of optical processes and computations that are inherent in image formation based on Coded Aperture Imaging [2]. An overview of the optical setup of data compression is shown in Figure 1.2. Examples of computational photography include in-camera computation of digital panoramas, [19] high-dynamic-range images, and light field cameras. Computational Processing is the processing of non-optically-coded images to produce new images. Computational sensors are detectors that combine sensing and processing, typically in hardware, like the one demonstrated in this paper. As human vision is immediate, our eyes can instantly recognize and categorize objects as well as the structures of scenes. The vast human computational resources that our brains bring to bear unconsciously to make all of this happen involves more than 50 percent of the cortex and a significant amount of energy (human electricity) on a daily basis[4]. Computational imaging innovation and apps delivered to market as early as next year will close the gap between what mobile devices and our minds eye sees and interprets while awake.

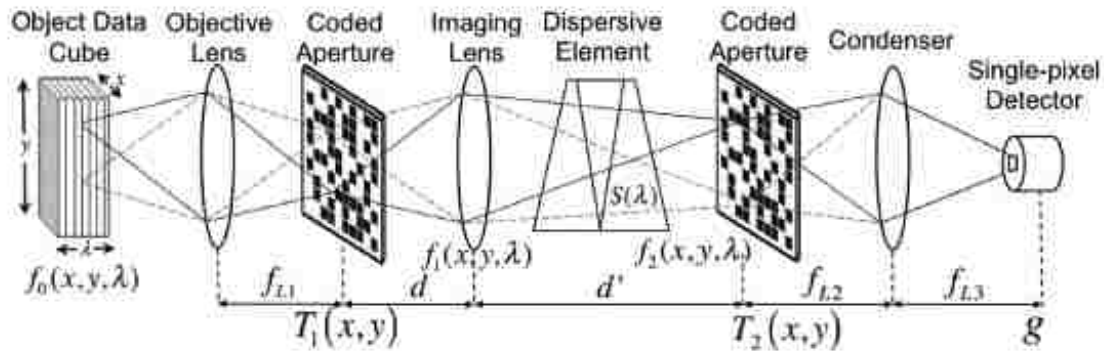


Figure 1.2: Optical Architecture of the Single pixel compressive hyperspectral imaging sensor (SPHIS)

Source: Single-pixel optical sensing architecture for compressive hyperspectral imaging

In traditional film-like digital photography, camera images represent a view of the scene via a 2D array of pixels. Computational Photography attempts to understand and analyze a higher dimensional representation of the scene [21]. Rays are the fundamental primitives. The camera optics encode the scene by bending the rays, the sensor samples the rays over time, and the final ‘picture’ is decoded from these encoded samples. The lighting (scene illumination) follows a similar path from the source to the scene via optional spatiotemporal modulators and optics. In addition, the processing may adaptively control the parameters of the optics, sensor, and illumination.

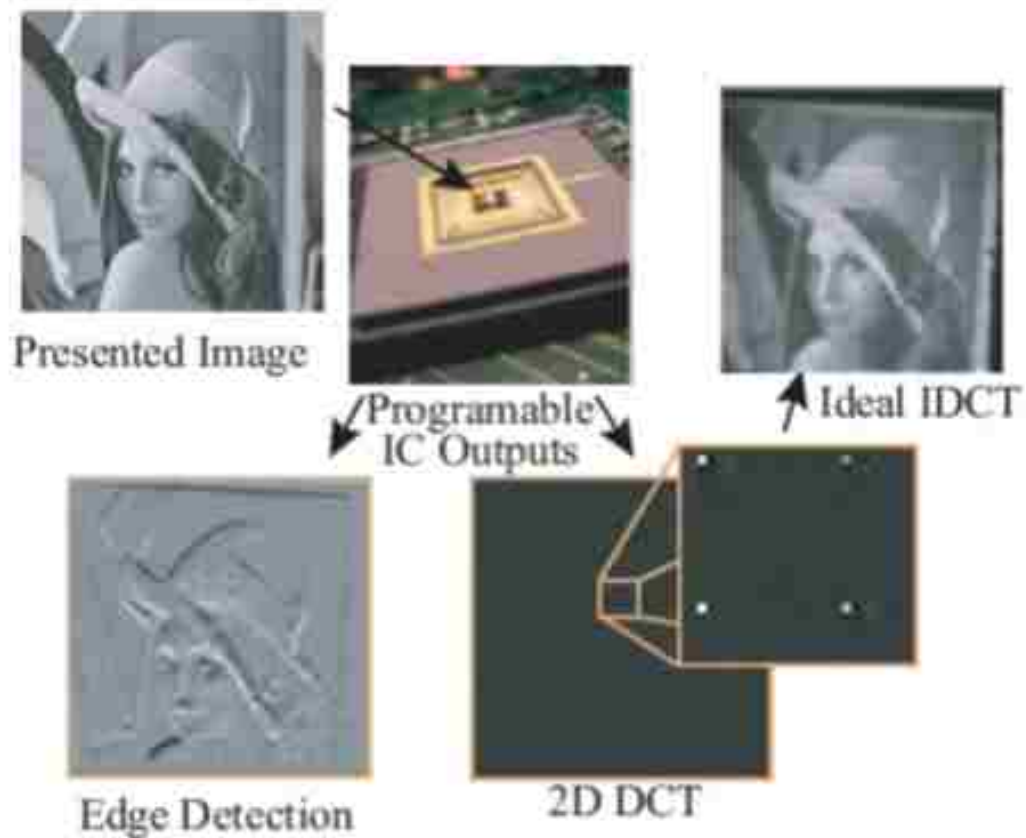


Figure 1.3: Illustration of Computational Imaging

The encoding and decoding process differentiates Computational Photography from traditional ‘film-like digital photography’. With film-like photography, the captured image is a 2D projection of the scene.[3] Due to limited capabilities of the camera, the recorded image is a partial representation of the view. Nevertheless, the captured image is ready for human consumption: what you see is what you almost get in the photo. In Computational Photography, the goal is to achieve a potentially richer representation of the scene during the encoding process. For example, successive images (or neighboring pixels) may have a different exposure, focus, aperture, view, illumination, or instant of capture. Each setting allows recording of partial information about the scene and the final image is reconstructed from these multiple observations. In other cases, Computational Photography techniques lead to ‘Coded Photography’ where the recorded photos capture an encoded representation of the world. In some cases, the raw sensed photos may appear distorted or random to a human observer. But the corresponding decoding recovers valuable information about the scene. Figure 1.3 illustrates the results of computational imaging hardware which are demonstrated in the later chapters.

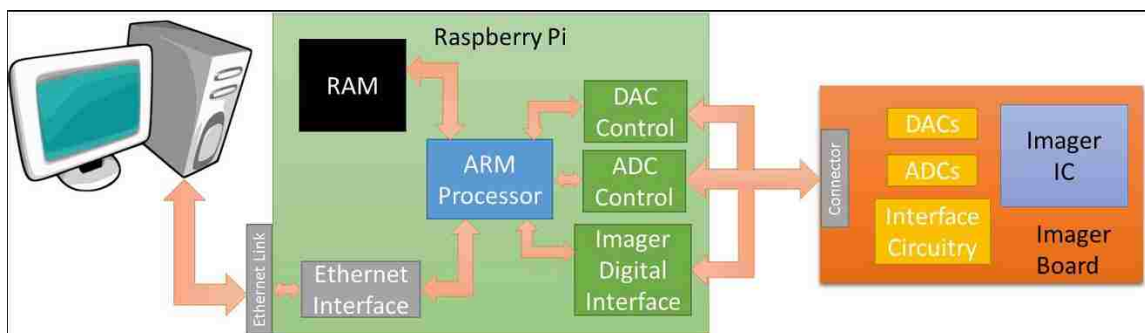


Figure 1.4: Implemented Programmable Imaging Hardware as per Javad’s work

1.3 Concept of Programmable Imaging

As we have seen, computational cameras produce images that are fundamentally different from the traditional perspective image. However, the hardware and software of each of these devices are designed to produce a particular type of image. The nature of this image cannot be altered without significant redesign of the device. This brings us to the notion of a programmable imaging system, which is illustrated in Figure 1.4. It uses an optical system for forming the image that can be varied by a controller in terms of its radiometric and/or geometric properties [1]. When such a change is applied to the optics, the controller also changes the software in the computational module. The result is a single imaging system that can emulate the functionalities of several specialized ones. Such a flexible camera has two major benefits. First, a user is free to change the role of the camera based on his/her needs. Second, it allows us to explore the notion of a purposive camera that, as time progresses, always produces the visual information that is most pertinent to the task.

1.4 Extension to Compressed Domain Hardware

Most of the discussion on this section are based on our paper [13]. For a typical image sensor, imaging involves reading out the values sampled at different pixels [15]; whereas in the case of compressed-domain hardware, a set of gain matrices is loaded to the pixel array, and the image sensor's output would be a linear combination of the projection of the object's reflectance function to the gain matrices [6, 18]. An example of such imaging hardware is shown in Figure 6.3. In the following paragraphs, we make some comparisons among a few other works that have been devoted to the problem of online compression and hardware domain sensing based on matrix projection.

One of the earliest reported hardware implementations to the compressive sensing

is based on a single-pixel camera [11]. The single-pixel imaging utilizes a digital micromirror (DMM) [25] to project the incident light coming from the object to the digital masks. The photodetector samples the integrated light coming from the sample, which is modulated by using the DMM. This method is usually used for far infrared imaging where having an array of low-cost, small size photodetectors is not feasible. The DMM degrades the sensitivity of the imager, and the alignment of different components is a limit to the scaling of this method.

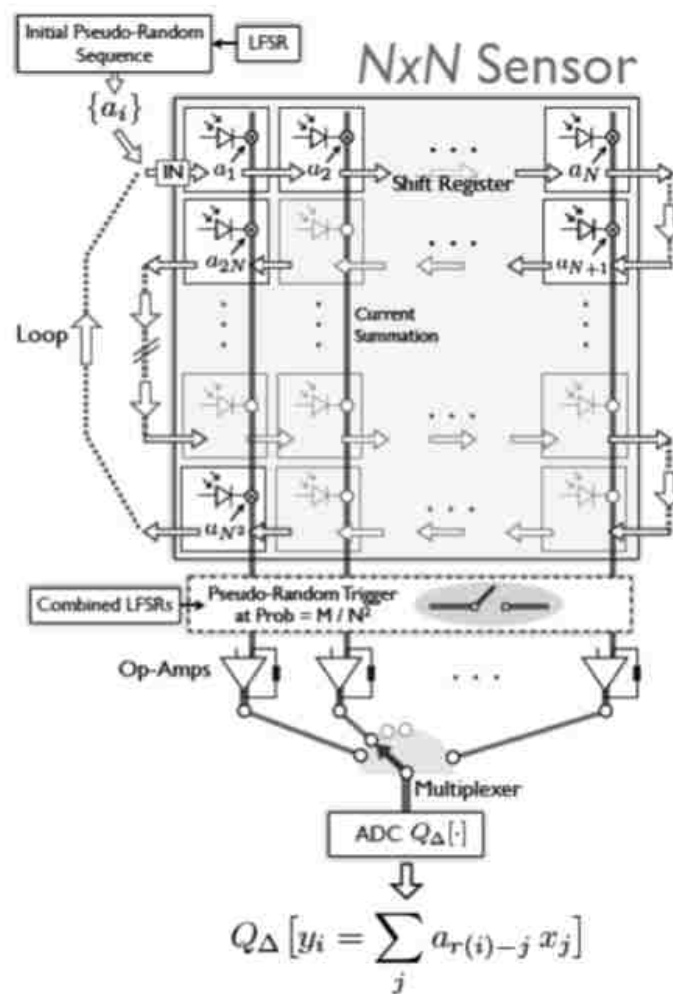


Figure 1.5: Scheme of a CMOS Compressed Imager.

Source: L. Jacques et.al. CMOS Compressed Imaging by Random Convolution

An optical-domain coded-aperture based compressive sensing is demonstrated in [16]. A random phase mask injects the measurement matrices, and the modulated intensities at different pixels are sampled using a low-resolution imager. This technique suffers from the noise added by the optical masks, and the complexity of the alignment setup is a big challenge.

A CMOS imager is demonstrated in [20] that utilizes a flip-flop-based shift-register distributed over the pixel array to hold the random digital patterns. The shift register selectively disconnects the pixels from the readout and implements the measurement matrices. The proposed hardware offers multiplication only by a binary value. This limits the compressive-sensing algorithm to the binary projection matrices, which are composed of only one or zero. Furthermore, there is no control over the bias voltage of the detectors, the result of which many features that are offered by modulation at the detector level are not supported. Finally, because the unit cell does not support integration, the proposed hardware cannot work with the detectors with lower quantum efficiency.

1.5 Publications

A list of publications generated during the development of this thesis work is presented below.

1. **Bhattacharai M**, Ghasemi J, Fiorante GR, Zarkesh-Ha P, Krishna S, Hayat MM. Intelligent bias-selection method for computational imaging on a CMOS imager. In Photonics Conference (IPC), 2016 IEEE 2016 Oct 2 (pp. 244-245). IEEE.
2. Ghasemi J, **Bhattacharai M**, Fiorante GR, Zarkesh-Ha P, Krishna S, Hayat MM. CMOS approach to compressed-domain image acquisition. Optics Express. 2017 Feb 20;25(4):4076-96.

Chapter 2

Experimental setup

The objective of this chapter is to give the reader background of the hardware to which the proposed bias selection algorithm is applied and the performance is evaluated. A major content on this section is based on my colleague Javad Ghasemi's work [14]. We have considered a real-time image-acquisition hardware featuring front-end computational imaging. This work makes use of a unique ROIC that enables compressed-domain image acquisition by means of projecting the input image onto a series of coded-apertures, or spatial intensity masks, that are stored in the ROIC. The real-time, pixel-level projection is achieved by the means of designing a set of coded apertures and programming them in the ROIC as a 2D array of analog biases. The stored bias values at each pixel govern the operating point of the photodetector; thereby affect spatial intensity modulation of the image. The pixel values are then summed up within the ROIC. The output of this unique camera is, therefore, the features (or code-words) arising from the projection of the image onto each of the prescribed code-apertures. The image can be reconstructed using a reconstruction algorithm that employs the features outputted by the camera. Henceforth, at no point, the actual image is generated or stored by the camera.

Figure 2.2 presents our proposed monolithic CMOS image sensor that can run as

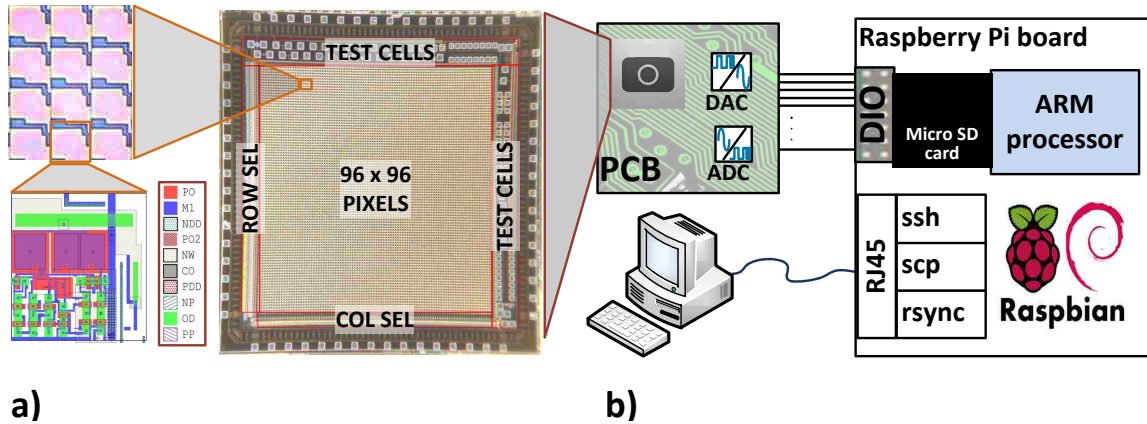


Figure 2.1: a) A microphotograph of the fabricated ROIC, the row, and column select, and the test devices. The unit cell is shown in the extended view. b) A block diagram of the experimental setup, which includes a Raspberry Pi board as the main controller of the system, an ADC, and a DAC to set the bias voltage of the detectors and grabs the readout of the imager. All communication between the controller and a remote machine is over SSH.

Source: based on our second publication

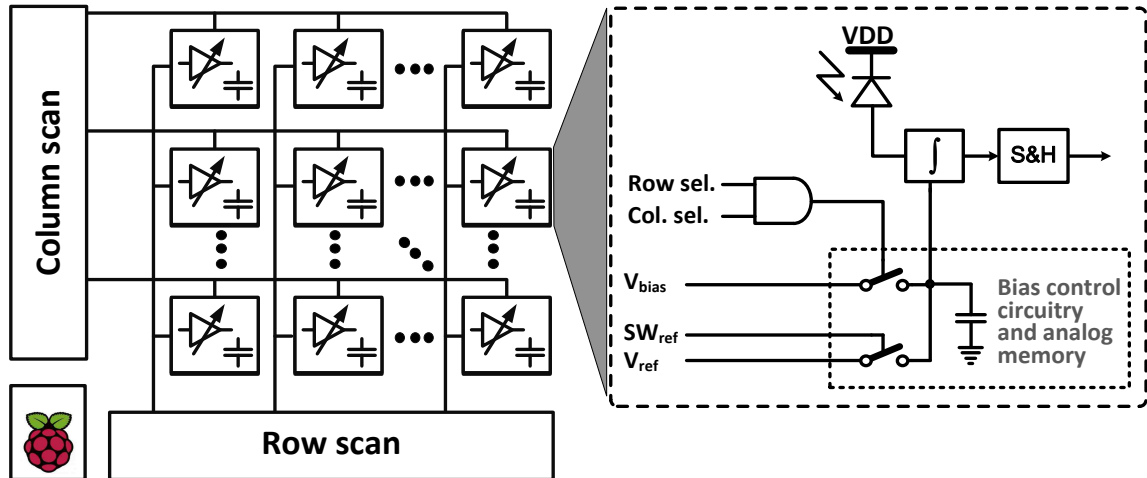


Figure 2.2: Block diagram of the individual pixel bias tunable readout integrated circuit and the CTIA-based unitcell at the extended view. The extra circuitry added to the CTIA-based unitcell enables setting independent bias voltages for each individual pixel while the previously integrated voltage is being read out.

Source: based on our second publication

a stand-alone image sensor and is able to perform spatiotemporal region of interest enhancement.

The hardware is also capable of generating already compressed images as well as canceling the nonuniformity inherent from process variation or other sources such as a voltage drop across the image sensor chip. The main contribution of this hardware is the introduction of control over a per-pixel modulation factor through controlling the photodetector's responsivity that is demonstrated as a controllable gain symbol in the pixels. The capacitor represents the analog memory that is embedded to store and hold the bias information for individual pixels. The AND gate selectively enables different pixels to load the bias voltage to the active pixel, and this selection occurs at the same time that the pixel is being readout; therefore, no delay penalty is associated with the new design. While sampling the integrated voltage to the sample-and-hold (S&H) capacitor, voltage V_{ref} is used as a global reference voltage for all of the preamplifiers. This removes the bias voltage from showing up in the readout and makes the readout value meaningful.

During the readout, the bias information, which is loaded to different pixels, can be different from each other and also from the bias that is loaded to the same pixel in the previous frame. This is what we refer to as the spatiotemporal independence of pixels biasing scheme.

The proposed hardware has the unique feature of performing application-specific transform coding based on a specialized set of bias masks. These sets of bias masks are dictated by a rigorous bias-selection algorithm that is then stored in the memory of the device. The incoming image data is projected into the designated masks to generate the code words used for image reconstruction. Most importantly, our proposed bias-selection algorithm, which has not been reported in the literature, considers the responsivity of the device, resulting in remarkably less reconstruction error. We will discuss the detail implementation of the iROIC in the next section.

As a proof of concept of computational imaging, we have employed a set of DCT matrices as the spatial masks as the code apertures. The bias matrices representing each DCT mask are loaded to the internal memory of a Raspberry PI board (RPIB), which is the main controller of our image-grabbing system. The RPIB loads individual pixels with its corresponding bias voltage and records the pixels response to the bias that has been applied in the previous frame. Finally, the summation of the result of the projection is done in the RPIB, which returns a sample of the image, in the form of a specific code word, for each coded aperture. This work, however, assumed that the responsivities of the detectors were known with some certainty. In this work, we devised an algorithm for the bias selection, assuming some uncertainty in the detector responsivity, which can be due to detector-non-uniformity across the array, temperature variation, inaccuracies in controlling the bias, etc. Neglecting the uncertainty in the detector responsivities results in degradation in the reconstructed image from the code-words (samples) since the code-words that are outputted are inaccurate; the latter being a result of the aperture codes being implemented inaccurately. In this work, we develop an algorithm for the bias-selection assuming uncertainty in the responsivity. The bias-selection algorithm is based on the minimum-mean-square-error (MMSE) criterion. The evaluation of the system is performed under a different number of coded apertures and for a variety of aperture codes. We show that bias selection based on the MMSE criterion aids to reduce image degradation after image.

To implement the spatial aperture-coding algorithm at the pixel-level (front-end) of the FPA, a novel intelligent readout circuit (iROIC) concept is needed. Specifically, the iROIC enables (a) the application of any bias, as instructed by the sensing algorithm, to the individual members of the Pixel array; and (b) on-chip implementation of analog-domain signal-processing operations, such as summing up the individual photocurrents over rows/columns of the array or over the entire array, and performing simple logical operations electronically at the pixel level as required to produce a

compressed output [14]. A key feature of the iROIC is that with very few additional transistors it can be programmed to drive each of the electrically tunable photodetectors in the FPA individually, according to a prescribed dynamic-bias profile. The combination of the above three building blocks (Pixel array, sensing algorithm, and iROIC) allows us to emulate the effect of sensing a scene in the eyes of arbitrarily specified spatial aperture codes, i.e., 2D basis functions, required to perform lossy compression, without using any physical spatial filters and without performing any algebraic multiplications mathematically. Next, by summing up over all pixels in the iROIC and for each 2D aperture code, we will obtain, in effect, the mathematical projection of the Spatial-image scene, performed electronically and instantaneously, onto the 2D aperture code. The projection yields a coefficient containing all the information about a scene as seen with the 2D spatio-spectral basis function. This process is repeated for different spatial patterns through the application of different biases (one for each 2D spatial aperture code), yielding a series of coefficients that represent the spatial scene. The spatial basis collections can be constructed using bias selection algorithm.

We have demonstrated a simplified special case on the spatial domain, where the dynamic bias is selected to implement gray-scale aperture-coding while the signal-to-noise ratio (SNR).

In the implemented hardware, the timing signal and the analog bias for the photodetectors are generated using a Raspberry Pi board (RPB). The main reason for choosing the RPB as the main controller is its extended support for onboard memory in the form of a micro-SD card. The typical FPGAs do not support for high volume storage; this challenges the storage of massive bias information. A DAC converts these digital values to analog and then feed them to the iROIC. The output video signal is sampled using an ADC chip, which is derived by the RPB. The sampled data are both sent to a remote computer for the purpose of online monitoring and also are stored in the local memory of the controller to be processed later. The RPB board

acts as a stand-alone controller for the iROIC and performs all image acquisition details. A custom PCB board is designed to host the test chip, to interface the RPB board, and to deliver high signal integrity. The RPB board is controlled using a desktop over LAN, and test vectors are loaded using Linux’s standard commands such as `rsync`, `ssh`, `scp`, etc. A block diagram of the experimental setup is shown in Fig. 2.1(b). The control over bias information of every pixel’s detector and the flexibility offered by the experimental setup has enabled many different applications that are explained in the following sections.

To have a model for the response-modulation function of the imager, the response of the system to a uniform level of illumination at different bias voltages is measured. The normalized imager’s photoresponse is shown in Fig. 2.3. In the error-bar graph, the mean and standard deviations are based on statistical analysis over all the pixels in the entire 96×96 frame, and each measurement was repeated 10 times to reduce random noises. The mean value and the standard variation shown in this figure are employed as the base for bias selection in a real-time system. The curve infers that the system responds to the bias voltage in a semi-linear fashion as long as the detector’s bias voltage is limited to $\sim [+0.4, +3.5]$.

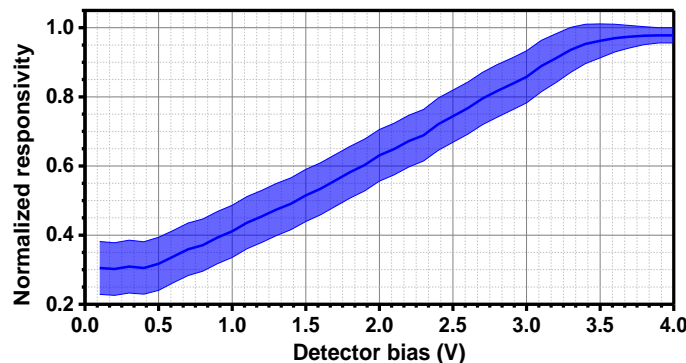


Figure 2.3: Demonstration of the normalized modulation function of the system to a uniform illumination level. The graph reflects the system’s response to the modulation of the detector’s bias.

2.1 Novelty of Hardware

2.1.1 Scalability to large-format arrays

We emphasize that all the compression operations are performed within the iROIC electronically using pre-calculated bias patterns that are applied to (or stored in) the iROIC; hence, the computational sensing is performed instantaneously in real time. In addition, the proposed iROIC hardware requires only a few additional transistors. With such a small area overhead, the given approach is fully scalable as the resolution of the imaging hardware in the spatial domains grows to megapixels and beyond, which is the main source of big data.

2.1.2 Proof of concept for real-time implementation of front-end computational imaging

We have discussed about real-time visible-image acquisition hardware featuring front-end computational imaging [49]. It presented a novel ROIC that has enabled projecting an object in the field of view onto a series of prescribed intensity-coded apertures, or spatial intensity masks (discrete-cosine transform (DCT) and compressive sensing (CS) random spatial masks were used), that were stored in the ROIC.

Chapter 3

Compressed-domain image acquisition

The most important application of the chip is targeted in a compressed-domain imaging framework. The compression is achieved by the hardware by performing a projection of the image to a set of basis masks implemented in the detector's biases. We have considered two different in-hardware compression modalities, which are in-pixel discrete-cosine-transform (DCT) based compressed-domain image acquisition and compressive sensing framework [24, 28].

To implement the compression modalities in hardware, we need to adapt the compressive masks as per device responsivity so that we ensure the mask coefficients are exactly achievable as modulation factors at the pixels.

3.1 Discrete cosine transform

In this part, we present the mathematical formulations for compression and reconstruction of the image using the DCT. In order to realize any sort of transform coding

on the computational imaging hardware, one needs to be able to project the acquired image into the designated mask where the transform coefficients need to be realized at each of the pixels as multiplication factors. Considering R is the responsivity of the image sensor, which is a function of the object's reflectance function I and the detector's bias voltage V , then:

$$R = g(I, V), \quad (3.1)$$

where g is some nonlinear function of I and V . Here, if I is the object reflectance function in spatial domain, then its frequency domain transform is given by:

$$y^{uv} = \frac{2}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left[C(u)C(v)I_{ij} \cos \frac{\pi(2i+1)u}{2N} \cos \frac{\pi(2j+1)v}{2N} \right], \quad (3.2)$$

where i and j are integers in the range of $[0, N - 1]$, which are used to address different pixels, and $C(u)$ and $C(v)$ are defined in the following equation:

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u, v = 0 \\ 1 & \text{otherwise.} \end{cases} \quad (3.3)$$

The inverse of the DCT transform function is defined as:

$$I_{ij} = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \left[y^{uv} \cos \frac{\pi(2i+1)u}{2N} \cos \frac{\pi(2j+1)v}{2N} \right] \quad (3.4)$$

In order to implement the computationally intensive DCT transform in hardware, we have reordered Eq. (3.2) and decoupled the bias (mask) matrices from the image sensor responses, which is shown in the equation below:

$$y^{uv} = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_{ij} Mask^{uv}(i, j)], \quad (3.5)$$

where:

$$u, v = 0, 1, \dots, N - 1.$$

In the above equation, $Mask^{uv}(i, j)$ is the mask set that is to be loaded to the image sensor as the bias information. If we assume N equals M , for exact reconstruction

the total number of masks would be $N \times N$. The mask matrices can be represented as,

$$Mask^{uv}(i, j) = \cos \frac{\pi(2m+1)u}{2N} \cos \frac{\pi(2n+1)v}{2N}. \quad (3.6)$$

In the calculation of the mask matrices, because $C(u)$ and $C(v)$ are not a function of m and n , they are treated as constants and are not included in Eq. (3.6). Because all of the coefficients are limited to the same range of $[-1, +1]$, we could efficiently use the limited dynamic range of the analog memory to store the bias voltage; otherwise, the DCT coefficient would need a greater number of bits to deliver the same SNR.

The discussion above works fine as long as the system is noise free; however, the system's response-modulation function shown in Fig. 2.3 triggers the need for a more intelligent bias-selection algorithm. Due to the device's limited dynamic range and noise behavior of the system, it is a must to have a bias-selection algorithm. This algorithm efficiently prescribes the optimal bias to each pixel, which leads to minimization of the effect of noise. Also, some linear transformation is used to map all coefficients over the given implementable dynamic range. The next section is devoted to the mathematical model of the device-response and bias-selection algorithms.

3.1.1 Bias selection algorithm

In this section, we will describe a novel bias-selection algorithm based on the MMSE approach, which tends to address the issue of image reconstruction when noise comes into play in the responsivity of the device. When the bias corresponding to a basis coefficient is computed without considering the effect of noise in the responsivity of the device, then we call it a naïve technique. This term will be used frequently in the rest of paper to consider such cases.

The projection and reconstruction are exact as long as the device behaves deterministically for the applied mask. However, the complexity rises as its behavior

tends to be random and there exists a finite uncertainty in its response. In this case, the common reconstruction method does not lead to exact recovery as it is difficult to find a unique bias that is able to achieve the designated gain factor. Next, we discuss a technique that enables us to optimally choose the bias for the given mask coefficient.

To begin describing the bias-selection method, as shown in Fig. 3.1, we consider a set of basis masks, $\{B^k\}_{k=1}^N$, each of which is to be implemented by a 2D array of biases to be determined later. Each of these masks consists of a 2D array of coefficients, given by $\{\{b_{ij}^k\}\}_{i,j=1}^N$. The objective is to map each of these b_{ij}^k coefficients into achievable responsivity values by means of the application of appropriate bias drawn from the responsivity function given by $\tilde{R}(v)$. Here, $\tilde{R}(v)$ is the noisy responsivity of the device as a function of applied bias. This bias assignment is carried out according to the optimization criterion stated in Eq. (3.12).

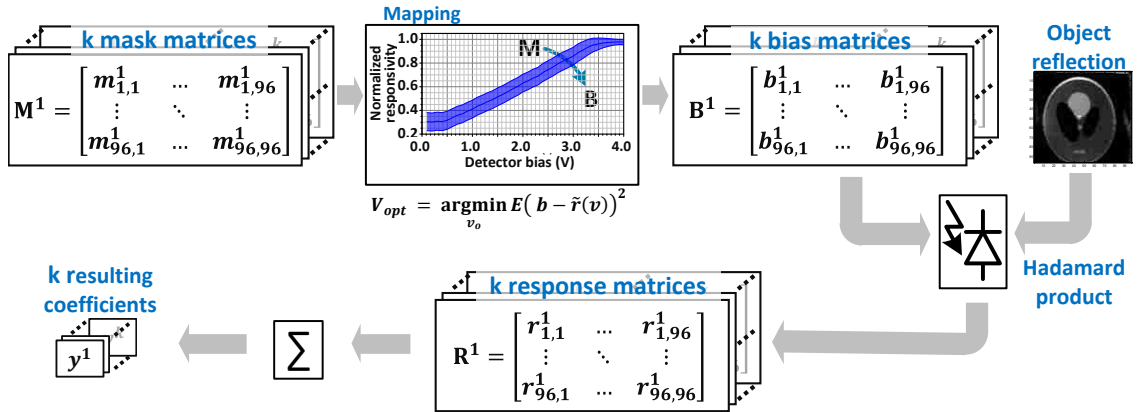


Figure 3.1: Acquisition and compression processes, which include mapping k mask matrices to their corresponding bias voltages. The mapping is based on the system's response-modulation function shown in Fig. 2.3. Then the bias matrices that are stored in the Raspberry Pi memory are loaded to the imager and projected to the object's reflectance function. The resultant dot product is optionally summed up in the hardware, and the k resulting coefficients are sent to the remote computer for reconstruction.

For an imaging system of resolution $N = 96 \times 96$ pixels, the image captured by the system I , the matrix of DCT coefficients Y , and the k -th ideal DCT mask $B^{(k)}$, is represented by

$$I = \begin{bmatrix} I_{1,1} & \dots & I_{1,96} \\ \vdots & \ddots & \vdots \\ I_{96,1} & \dots & I_{96,96} \end{bmatrix}, Y = \begin{bmatrix} y^{(1)} & \dots & y^{(96)} \\ \vdots & \ddots & \vdots \\ y^{(96^2-96)} & \dots & y^{(96^2)} \end{bmatrix}$$

and

$$B^{(k)} = \begin{bmatrix} \tilde{b}_{1,1}^{(k)} & \dots & \tilde{b}_{1,96}^{(k)} \\ \vdots & \ddots & \vdots \\ \tilde{b}_{96,1}^{(k)} & \dots & \tilde{b}_{96,96}^{(k)} \end{bmatrix}.$$

The k -th practical mask based on noisy responsivity is:

$$\tilde{R}^{(k)} = \begin{bmatrix} \tilde{r}_{1,1}^{(k)} & \dots & \tilde{r}_{1,96}^{(k)} \\ \vdots & \vdots & \vdots \\ \tilde{r}_{96,1}^{(k)} & \dots & \tilde{r}_{96,96}^{(k)} \end{bmatrix},$$

and

$$\tilde{r}(v) = r(v) + \eta(\mu, \sigma_v^2), \quad (3.7)$$

where $R(v)$ is the implementable k^{th} mask based on ideal responsivity when the system is noise free. Now the expression for computing the individual DCT coefficients corresponding to the noisy responsivity mask and corresponding error are given by

$$y_{\tilde{R}}^{(k)} = \sum_{i=1}^{96} \sum_{j=1}^{96} I_{i,j} \tilde{r}_{i,j}^{(k)}(v), \quad (3.8)$$

and

$$y_{err}^{(k)} = y_{idl}^{(k)} - y_{\tilde{R}}^{(k)} = \sum_{i=1}^{96} \sum_{j=1}^{96} I_{i,j} \left(b_{i,j}^{(k)} - \tilde{r}_{i,j}^{(k)}(v) \right), \quad (3.9)$$

where the k -th DCT coefficient corresponding to the ideal mask is denoted by

$$y_{idl}^{(k)} = \sum_i \sum_j I_{i,j} b_{i,j}^{(k)}. \quad (3.10)$$

For a specific pixel at (i, j) position, if b is the mask coefficient to be achieved and $\tilde{r}(v)$ is the realizable coefficient from responsivity, then the objective function for bias selection for that specific pixel is given by

$$f(v) = (b - \tilde{r}(v))^2, \quad (3.11)$$

and the optimization problem is given by

$$\begin{aligned} & \underset{v}{\text{minimize}} && f(v) \\ & \text{subject to} && E(f(v)) = 0, \end{aligned} \quad (3.12)$$

where

- $E(f(v))$ stands for the expected value of the entity $f(v)$, which is a function of v .
- $f(v) : \mathbb{R}^n \rightarrow \mathbb{R}$ to be minimized over variable v .
- $E(f(v)) = 0$ is the equality constraint.

Equivalently, the problem can be reformulated as

$$v_{opt} = \underset{v}{\operatorname{argmin}} E(f(v)),$$

where

$$E(f(v)) = (b - r(v))^2 - 2\mu(b - r(v)) + \mu^2 + \sigma_v^2,$$

then to find the optimum v_{opt} , we differentiate the objective with respect to v such that $\frac{d}{dv} E(f(v_{opt})) = 0$, and thus we obtain

$$r(v_{opt}) = b - \mu - \sigma \frac{\frac{d}{dv} \sigma_{v_{opt}}}{\frac{d}{dv} r(v_{opt})}, \quad (3.13)$$

and

$$v_{opt} = \sigma_{v_{opt}}^2 \left[\left(\frac{\frac{d}{dv} \sigma_{v_{opt}}}{\frac{d}{dv} r(v_{opt})} \right)^2 + 1 \right], \quad (3.14)$$

where, $r(v_{opt})$ corresponds to the optimal realizable gain coefficient for a given ideal value of mask coefficient b , and v_{opt} stands for the optimal bias to be applied to realize gain $r(v_{opt})$. The above expressions explained the optimal bias-selection rule driving the corresponding gain coefficients to be implemented on the pixel to realize the optimal mask coefficient b .

The bias-selection algorithm works fine as long as the variance of noise in the responsivity lies within some limit and the lighting condition does not change drastically. This is because for different operating light conditions, the responsivity might change and the designed bias in the memory will not be able to suffice the objective.

3.1.2 Conditioning the masks for mapping the bias into device dynamic range

For an image $\{\{I_{ij}\}\}_{i,j=1}^N$ and basis masks given by $B^k = \{\{b_{ij}^k\}\}_{i,j=1}^N$, the DCT coefficient for the ideal case is achieved as

$$y_{ij}^k = \sum_i \sum_j I_{ij} b_{ij}^k.$$

However, due to the device's limited operating dynamic range and memory, there is a need to appropriately condition the mask coefficients such that they are realizable as per the device responsivity. Once the projection is obtained, an equivalent transform needs to be applied to retrieve the actual DCT coefficients.

Now for any linear transformation given by $r = mb + c$, where m is the gain, c is the offset and r is the entity equivalent to b in the transform domain. Hence, this transformation is identically applied to all of the basis coefficients, to accommodate all of them into the working dynamic range of the device responsivity. If $r_{ij} = mb_{ij} + c$,

then

$$\begin{aligned}
y_{ij}^{k'} &= \sum_i \sum_j I_{ij} r_{ij}^k \\
&= \sum_i \sum_j I_{ij} (mb_{ij} + c) \\
&= m \sum_i \sum_j I_{ij} b_{ij} + c \sum_i \sum_j I_{ij} \\
&= m y_{ij}^k + c \sum_i \sum_j I_{ij} .
\end{aligned} \tag{3.15}$$

Then, for each projection coefficient $y_{ij}^{k'}$, we can condition as follow so to retrieve the actual projection coefficient:

$$\frac{y_{ij}^{k'}}{m} - c \sum_i \sum_j I_{ij} = y_{ij}^k . \tag{3.16}$$

This conditioning is responsible for mapping of the target mask coefficients into the realizable region, the distribution of which is shown in Figs. 3.2(a) and 3.2(c) corresponding to naïve and MMSE methods, respectively.

Also, as observed from the distribution of bias from Figs. 3.2(b) and 3.2(d), the MMSE spreads out the bias to ensure the quantization effects on implementation are

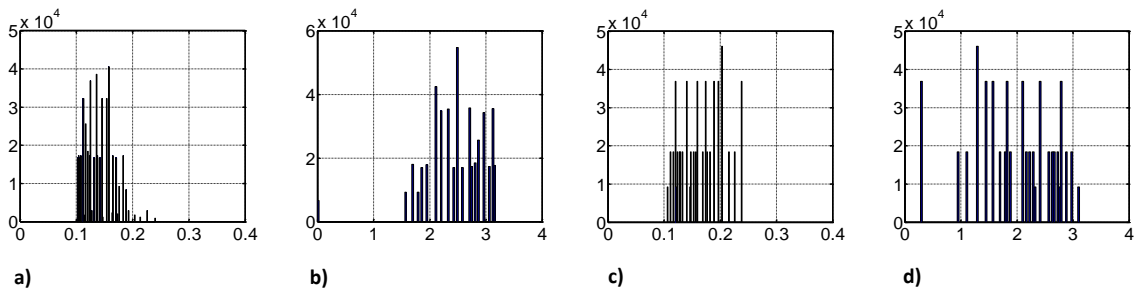


Figure 3.2: a) Distribution of 8×8 block-based DCT mask coefficients for naïve method, b) distribution of bias for naïve method, c) distribution of 8×8 block-based DCT mask coefficients for MMSE method, and d) distribution of bias for MMSE method.

minimized. As MMSE considers the effects of noise while bias is prescribed for the given mask, the variance is added on the realizable mask coefficients, which leads to their spread when compared to that designed without considering the effect of noise.

3.1.3 DCT-based image compression

Once the optimal masks and gain are designed with the aid of the bias-selection algorithm, the biases are then applied to the hardware, which in turn, results in achieving the desired coefficients as modulation factors at each pixel. Finally, the DCT coefficient corresponding to each mask is achieved by

$$y_{opt}^k = \sum_{i=1}^{96} \sum_{j=1}^{96} I_{i,j} \tilde{r}_{i,j}^k(v_{opt}) . \quad (3.17)$$

3.1.4 DCT-based image reconstruction

Image reconstruction is achieved by simply applying the linear combination of the masks to which the image was projected. The reconstruction is achieved by the following equation:

$$I_{rec} = \sum_k R_{opt}^k y_{opt}^k . \quad (3.18)$$

Following the discussion above, we performed DCT-based image compression optimally on the hardware. However, some error still exists in the projection coefficients that propagate during the reconstruction, which is mainly due to the limited dynamic range of the pixels and different random uncharacterized noise present in the hardware.

3.2 Block based Transform Coding

The analog imaging sensor basically has a limited memory which enforces the device to operate over a limited dynamic range. That constrains the device to rely on block based transform coding instead of a large kernel mask. This is because, for large block size of mask, there are more number of mask coefficients which are also denser. This gives raise to quantization issue as most of the neighboring coefficient values are rounded to their nearby realizable coefficients. As a result, the realized mask loses its orthonormal property and the implemented mask is no longer equivalent to the targeted mask leading into reconstruction errors.

Block-based transform coding has the ability of energy compacting property and relative ease of implementation [17]. To save the vital resources like power, time and memory for real-time application, block-based transform coding provides an efficient way to achieve the same compressibility. Block based transform coding leads to smaller number of coefficients which are spread in distribution preserving orthonormal property of basis even after being quantized in the hardware. With smaller block size, although the total number of coefficients gets increased by factor of $\frac{Total\ Image\ Size}{block\ size^2}$, but the total number masks to be implemented is reduced to $\{block\ size\}^2$ from $\{Total\ Image\ Size\}^2$ that enables to save time and other resources.

The idea of block-based transform coding implemented over here is to divide the image into blocks of the desired size and then perform transform coding on each block independently. However, such process results in a noticeable degradation known as “blocking artifact”. This is mainly due to independent processing of the blocks, which is unable to take account of existing correlation between adjacent block pixels. Existing techniques use DCT based filtering, spatial filtering or hybrid techniques to overcome it.

3.3 Reduction of Blocking Effect in Image Coding

We adopt a spatial Gaussian filtering based image enhancement procedure after decoding to minimize the blocking artifacts caused due to the block based transform coding [22]. It is well known that any sharp edges in an image represent high-frequency content. Boundary region discontinuities caused by the segmentation procedure are similar to very sharp edges. Furthermore, we also know the location of these edges exactly, since the segmentation procedure is predetermined based on block size. Thus, we can reliably expect that low-pass filtering of the image at or near the sub-image boundaries would tend to smooth the unwanted discontinuities resulted due to blocking artifacts. This is the basis of the filtering method. We have considered a variety of low-pass filters for test considering the criteria for simplicity and relative effectiveness. A 3x3 Gaussian spatial domain filter was chosen, after some brief test which was found optimal. The specs of the filter is given by

$$h(n_1, n_2) = .2042 \times \exp(-.5(n_1^2 + n_2^2)) \quad (3.19)$$

where $n_1, n_2 = -1, 0, 1$

Only pixels directly adjacent to sub-image boundaries are processed by the filter after decoding with the least damage to image content.

3.4 Reconstruction Noise Analysis

In order to model the noise in reconstruction, we let the k -th DCT coefficient for the ideal mask $y_B^{(k)}$ and the estimated/computed mask $\tilde{y}_B^{(k)}$ be given by

$$y_B^{(k)} = \sum_i \sum_j I_{i,j} b_{i,j}^{(k)}, \quad \text{and} \quad \tilde{y}_B^{(k)} = \sum_i \sum_j I_{i,j} \tilde{b}_{i,j}^{(k)}.$$

Then, if the reconstruction is given by $\tilde{I}_{rec} = \sum_j \tilde{y}_B^{(k)} B^{(k)}$, the reconstruction error is

$$I_{err} = \tilde{I}_{rec} - I = \sum_j \tilde{y}_B^{(k)} B^{(k)} - \sum_j y_B^{(k)} B^{(k)} = \sum_j \left(\tilde{y}_B^{(k)} - y_B^{(k)} \right) B^{(k)}.$$

Next, we use the frequency domain error $\left(\tilde{y}_B^{(k)} - y_B^{(k)} \right)$ to model the corresponding error in spatial domain. We notice that the covariance matrix in the frequency domain

$$K_{e_B} = E \left\{ \left(\tilde{y}_B^{(k)} - y_B^{(k)} \right) \left(\tilde{y}_B^{(k)} - y_B^{(k)} \right)^T \right\}$$

has only non-zero elements on the diagonal, and these elements correspond to the perturbation in each coefficient by variance $\sigma_{e_B}^2$. Then we wish to find the covariance matrix in spatial domain as follows

$$K_{e_I} = E \left\{ \left(\tilde{I} - I \right) \left(\tilde{I} - I \right)^T \right\} = \Gamma^T K_{e_B} \Gamma$$

where the $N^2 \times N^2$ matrix Γ is constructed by stacking $N^2 \times 1$ vectors β_k hence, $\Gamma = [\beta_1 | \beta_2 | \dots | \beta_{N^2}]$. Here each β_k is the vectorized form of each DCT mask $B^{(k)}$. Now to find the individual covariance element $\sigma_{e_I}^2$ and the vector of error variance λ , both in the spatial domain, we let $M_{k,l}(m,n) = \Gamma^2(l,m)\Gamma^2(l,n)$ and whence

$$\sigma_{e_I}^2(m,n) = \sum_k \sum_l \Gamma^2(l,m)\Gamma^2(l,n) \sigma_{e_B}^2(k,n), \quad \lambda = \sum_k \sum_l M_{k,l}(m,n) \sigma_{e_B}^2(k,l)$$

Lastly, we model the noise in the spatial domain as a zero-mean Gaussian random variable, with auto-covariance K_{e_I} , hence it has a probability density function:

$$p(e_I) = \frac{1}{(2\pi)^{-N/2} \|K_{e_I}\|^2} \exp \left(-\frac{1}{2} e_I^T K_{e_I}^{-1} e_I \right)$$

Chapter 4

Compressive sensing

The well established Nyquist Criteria requires a signal to be sampled at a rate at least twice of its frequency content to be able to faithfully reproduced from its samples. Well, the Nyquist criteria refer not to the frequency, but to the bandwidth, which is related to information density in a signal. A very high-frequency signal, of approximately known frequency, with a sufficiently small bandwidth, will still be aliased or folded down with baseband frequencies by undersampling. But if the bandwidth (or other known characteristics) of the signal is known to rule out other aliased frequencies (such as the non-existence of baseband spectrum content), that knowledge in conjunction with the undersampled samples may still provide enough information to reconstruct the signal. Yes, it is possible to sample and reconstruct a signal at a sampling frequency lower than the Nyquist Criteria. For that, the signal has to be sparse in some representation basis. Then it is absolutely possible to reconstruct the signal with a certain probability of having an error in reconstruction.

Compressed sensing (also known as compressive sensing, compressive sampling, or sparse sampling) is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. This is based on the principle that, through optimization, the sparsity of a signal can be

exploited to recover it from far fewer samples than required by the Shannon-Nyquist sampling theorem. There are two conditions under which recovery is possible.[1] The first one is sparsity which requires the signal to be sparse in some domain. The second one is incoherence which is applied through the isometric property which is sufficient for sparse signals

Signal sampled at the Nyquist frequency can be perfectly reconstructed. But that does not mean that the information content of that signal is always evenly distributed across the bandwidth (which is limited by Nyquist frequency). Compressed sensing tries to exploit this trade-off. Signal encoded with compressed sensing is able to encode enough information to be able to reconstruct the desired signal which may or may not be as perfect as signal encoded by sampling at the Nyquist frequency.

Compressed sensing depends on the sparsity of the signal and not its highest frequency, which might seem to violate the sampling theorem. This is a misconception because the sampling theorem guarantees perfect reconstruction given sufficient, not necessary, conditions. A sampling method different from the classical fixed-rate sampling, therefore, cannot violate the sampling theorem. Sparse signals with high-frequency components can be highly under-sampled using compressed sensing compared to classical fixed-rate sampling.

4.1 Introduction to Compressed Sensing Theory

CS is based on the principle that one can recover certain signals and images from far fewer samples or measurements than traditional methods do and suggest that for a given focal plane array, a scene can be recovered at higher resolution than as dictated by the pitch of FPA which limits the reconstruction in naïve method as specified by Nyquist criteria. The well celebrated Nyquist theorem states that signal that has no frequency outside the interval $[-B/2, B/2]$ can be precisely recovered

from their samples taken with a sampling interval $1/B$ where B is called the Nyquist rate. Precisely, it says that a signal can be uniquely restored from sampled data collected with rate B samples per unit of the signal area given that the signal has occupied B area in its spectrum. However, as one can not find ideal bandlimited signals in practice, sampling theorem for such signals corresponds to band limited approximation and optimal in terms of mean squared approximation error. To satisfy the sampling theorem for an image, the sampling intervals are usually considered on the fact that how many pixels are sufficient for sampling smallest objects or object boundary so as to ensure their resolvability, localization, and recognition. And as such sampled pixels comprise a small part of the entire image where the given image may have many of such redundant pixels present throughout the given image as a result of which Fourier, DCT, DFT and other transform power spectra of images decay quite rapidly at a higher frequency. We call such property as signal sparsity and this provides the tool for various transforms to compress the image energy in few spectral coefficients called as energy compaction capability. Such ability of transforms replace the images by their bandlimited or sparse approximations by representing the information into fewer non zero components which are actually less than the original image. Compressed sensing is also based on the assumption that signal is sparse in some transform domain and suggests another approach for signal discretization that avoids the need for further compression. For a given signal, if we consider that N is the actual number of the signal samples present and for some $M \ll N$ where M be the number of certain measurements to be done to obtain signal discrete representation and reconstruct N samples with $L1$ -norm minimization. Here to measure the degree of compression, we use a compression sensing compression factor which is given by $CSCF = N/M$. Besides, Data compressibility is evaluated wrt signal sparsity which is given as $DC = K/N$ where K is the number of non zero transform coefficients of the signal sparse approximation for a given N total number of signal samples. According to discrete sampling theorem (DST), if only K out of N discrete signal spectral components is non zero then all signal N

samples can be precisely restored from its K samples provided that indices of the non zero spectral components are known and restrictions on their position as defined by the signal transforms are met. The condition being the theoretical upper bound of signal compression achieved by sparse approximation is given by $CUB = 1/SPRS$ where CUB stands for compressibility upper bound and $SPRS$ stands for sparse signal approximation given by $SPRS = K/N$. Thus on an important note, compressed sensing approach is based on the assumption that signals can be approximated by their sparse or bandlimited copies which are based on energy compaction capability of transforms

Compressed sensing technique has an ability to restore signals with few spectral components within baseband defined by the component of the highest frequency from their samples taken with a rate lower than twice this frequency. This assumption doesn't necessarily overcome the Nyquist sampling principle for such signals as the Nyquist requirement of twice the component of highest frequency may not be the required Nyquist rate for such signals. Besides Nyquist deals with signals with few spectral components by dividing the signal baseband into sub-bands of the width of those spectral components and carry out the sampling of the sub-bands that have non zero energy, thus not necessarily sampling with a rate if twice the highest frequency component. This sort of optimal signal sampling may be considered as sinusoidal modulation-demodulation so as to shift the frequency subbands to lower frequency band before sampling and then to shift them back for the signal reconstruction. Whereas compressed sensing replaces the signal sin modulation-demodulation by signal blind modulation-demodulation using a set of pseudo-random masks.

4.2 Background

CS is based on the fundamental principle that a small number of linear projections of a compressible signal suffices to represent the sufficient information for reconstruction

and processing. The earliest work demonstrating the CS work was Rice Single-pixel camera which basically use a digital micromirror array to perform optical calculations of linear projections of an image onto pseudorandom binary patterns. There is a growing pressure on imaging sensors, hardware and algorithms to accommodate humongous data sets of several dimensions over constraints on acquisition mechanism, power, bandwidth availability and efficient methodologies. However, amid of such issues, there has been a resolution in the field of computational imaging due to the breakthrough of in the field of processing capability and storage capacity providing an angle to tackle the prudent issues. This provides a motivation for moving from Digital signal processing paradigm to a computational signal processing(CSP) paradigm where the idea is to convert analog signals directly to a number of intermediate condensed representation. CSP basically exploits the structure of the data for intelligent representation and processing. The existing state of art compression modalities is based on the principle of decorrelating transform to condense a correlated signals energy into few representative coefficients. Considering the principal signal is sparse in some basis, the key idea is to seek a small number K for adaptively chosen transform coefficients to be transmitted or stored rather than $N \gg K$ being actually acquired as classical acquisition techniques. For eg, most of the smooth images are sparse in Fourier or Discrete Cosine basis and that motivates most of the existing commercial compression modalities like JPEG .

The conventional approach of the transform coding applied to image compression involves the following principal steps. I) acquiring the full N -sample signal ii) computing the complete set of transform coefficients, iii) locating the K largest, significant coefficients and then discarding the rest insignificant coefficients and finally encode the values and locations of the largest coefficients. The complexity is aroused when the size of data ie $N \gg K$. Then the reconstruction error goes significantly high when K goes far low compared to N as one may not be able to encode the most significant coefficients which primarily contribute to the structure of the image.

Whereas in the innovative compressed sensing technique, a signal that is K -sparse in some sparsity basis can be recovered from cK nonadaptive linear projections onto a second basis called measurement basis which is incoherent with the first basis for a given c oversampling constant. This is feasible due to non-linear reconstruction process based on optimization technique through the acquisition involves the linear projection mechanism. Another strong property of CS that makes it superior compared to conventional imaging modality is its principle of “sample less, compute later” which shifts the technological burden from the sensor to the processing which enables CS to provide a framework the computational signal processing paradigm. The work features a unique computational imaging hardware which supports universality where the random and pseudorandom measurement bases are universal in the sense that they can be paired with any sparse basis and this allows same encoding strategy to be applied in a variety of sensing environments as considering continuous random and binary random specific to this work. In addition to this, the hardware is open to any future implementations of any sort of random measurements which can be used to reconstruct an even better quality image with enhanced image processing techniques yielding better sparsity-inducing basis. Also in this compression technique involving CS as a tool, we can adaptively choose the selective number of measurements to compute the tradeoff between compression of the acquired image versus the acquisition time, unlike a conventional camera where the scalability is defined as a trade-off between resolution versus the number of pixel sensors. At last but not at least, the Computational imager places most of the computational complexity in the decoder reducing the computational burden for the encoder which is a very simple module that computes the incoherent projections and makes no decision. Here the projection refers to the arithmetic addition and multiplications.

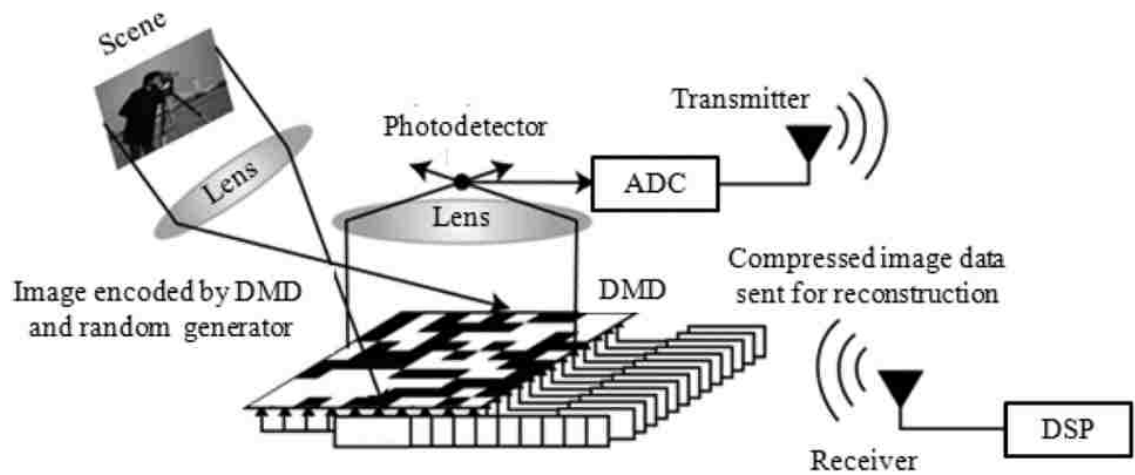


Figure 4.1: Rice Single pixel Camera
Source: Rice CS paper

4.3 Early CS Imaging Hardware

Here, the core idea is to be able to infer a high-resolution image from a relatively small number of measurements using some simple computational encoding method.

In the past, a most celebrated matter of debate in the field of research and consumer imaging technology used to regarding the optimal size of the sensor to meet the required higher resolution as for the conventional imaging systems, there was a need of one sensor (i.e. FPA element) per image pixel. However, CS systems tend to lessen the constraint by taking a smaller number of measurements and projecting the scene into an appropriate set of pseudo-random basis.

The underlying intuition regarding the feasibility of application of CS on most of the images is due to their inherent compressibility. With the development of several compression standards, which stores images using information fewer than one bit per pixel, most of them rely on the framework principle that the critical information content of the image is much less than the number of pixels times the bit depth of each pixel. Rather than measuring each pixel and then seeking compressed

representations, CS tends to measure the compressed representations directly.

As a consequence, only few image sensors based on CS have been presented in the literature. The optical domain CS imager in [2] suffers from a large camera size and limited resolution. The CS imager in [3] is very sensitive to mismatch and process variation effects and has limited frame rate since the pixel output currents are responsible for charging and discharging the column-line parasitic capacitance. The separable-transform image sensor in [4] allows high readout speed and camera resolution but suffers from device mismatch effects in the analog-matrix generation as well as high readout noise.

The history of CS systems is rooted in the rice single-pixel camera. The design is based on the digital micromirror array which is used to represent a pseudorandom binary array and the projection of the scene is done onto the array before the reflected light is aggregated and measured by the single photodetector. This hardware had a major limitation of portability and mobility as the setup needs to be focused to the scene of interest until enough samples are collected for reconstruction, which might take a long time for the setup itself and any sort of discrepancy in alignments of setup may fail to do reconstruction. To address some of the issues unresolved by the single pixel camera, a group of researchers came with an idea of coded aperture imagers, which basically rely on the principle of usage of coded apertures that are designed using pseudorandom construction, the observation model satisfying RIP criteria. One of the types of such imager is the random lens imaging optical system, which is a parallel architecture that provides a snapshot image and for which all M measurements are collected simultaneously without requiring complex and large, imaging apparatuses.

The development of aperture coded imagers was then followed by CMOS CS imagers. Where, Robucci et al proposed of achieving analog, random convolution process in CMOS device. Such imagers removed the early requirements of complex optical setup and making the design compact and efficient. Such devices are efficient

in terms of faster processing capability, space requirements and ability to perform block-wise inner product between incoming data and predefined random basis in silicon. Following the intuition from the early CMOS imagers, weve developed the CS enabled CMOS device.

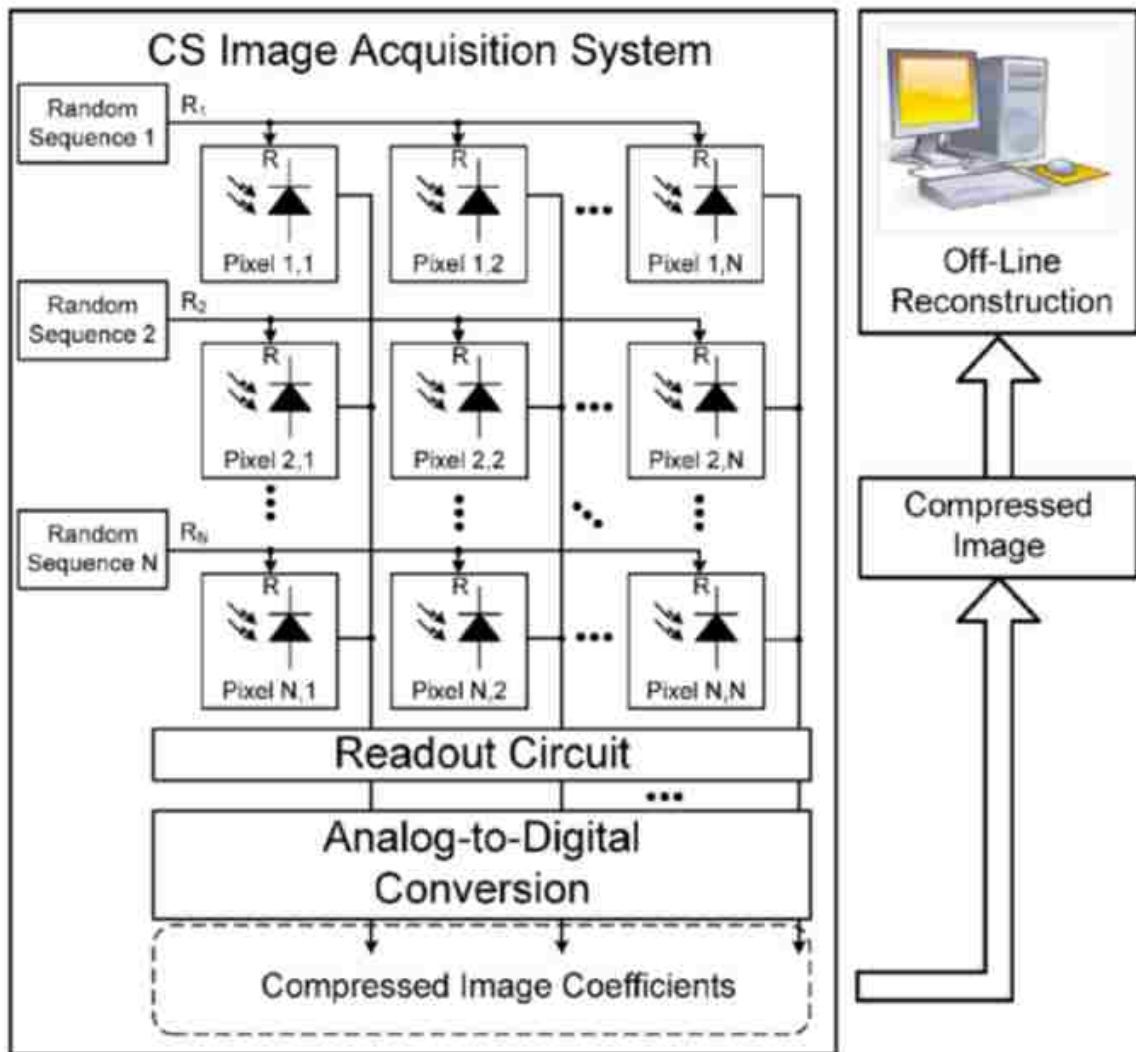


Figure 4.2: Overview of CS imaging hardware

4.4 Implementation

The second type of in-pixel compressed-domain acquisition we have explored is compressive sensing (CS). While in the DCT transform coding, the gain vectors vary continuously, which leads to the maximal exploitation of device dynamic range; the CS implementation simplifies the complexity by making use of only zeros and ones, which makes the system more resilient to noise. Here, we present some background regarding CS and implementation methodology on the proposed hardware.

CS is based on the principle of achieving a larger and more efficient compression, provided that the desired data is sparse in some basis. Sparsity is the primary condition here, which will lead to efficient reconstruction of data if it is sampled in a proper domain. We consider the input image as a discrete-time column vector $x \in R^P$ with elements $x[n]$ where $n = 1, 2, \dots, P$ and $P = 96 \times 96$. Then, x can be represented as a linear combination of elements from an orthonormal basis $\{\phi_i\}_{i=1}^P$ and coefficients s_i . Here,

$$x = \sum_{i=1}^P s_i \phi_i, \quad (4.1)$$

or

$$x = \phi s .$$

We assume that s is sparse with K nonzero coefficients. Now, by selecting an efficient binary random sensing matrix ψ , we can represent the reduced data set as $y = \psi x$ where ψ is a binary matrix of size $M \times P$ and $M \ll P$. In this way, the dimension of data set is reduced from P to M . However, the size M also needs to be properly determined for stable reconstruction. The standard expressions for computing M are given as

$$M \geq cK \log\left(\frac{P}{K}\right),$$

where c is a constant. Here, the matrix ψ is composed of M basis functions in P dimension to which data x is projected, i.e. $\psi = [\psi_1|\psi_2|\dots|\psi_M]^T$, where ψ_1 is of size $P \times 1$. The matrix was designed with the restricted isometry property (RIP) [8] given below:

$$(1 - \sigma_k) |x|_2^2 \leq |\psi x|_2^2 \leq (1 + \sigma_k) |x|_2^2, \quad (4.2)$$

where $\sigma_K \in [0, 1)$. Moreover, each ψ_i is converted to an equivalent 2D data set and then subjected to be implemented on hardware as a measurement mask. Because this mask is composed of binary elements, it is easier to achieve projections as the detector tends to switch on or off depending upon the bias applied for the acquisition. After we have obtained coefficients from the projection of the image to the reduced basis, the challenging problem is to reconstruct the image out of its dimensionally reduced format. Specifically, in this problem, we look forward to reconstruct image vector x by only using the M measurements in the vector y , the random measurement matrix ψ , and the orthonormal basis ϕ . Equivalently, we could reconstruct the sparse coefficient vector s . The estimate is given by the ℓ_1 minimization criteria, which uses a convex relaxation of the ℓ_0 norm given as

$$\hat{x} = \min \left\| x' \right\|_1, \quad (4.3)$$

such that

$$\psi x' = y,$$

and

$$\left\| x' \right\|_1 = \sum_i x_i.$$

The reconstruction was performed with the aid of ℓ_1 -magic algorithm, where the same random basis was considered for reconstruction, which was used for the projection during the hardware implementation [7].

Hence, in this work, weve presented a first prototype imaging system that successfully employs a compressive sensing principle and has attractive features of simplicity, universality, robustness, scalability which enable the usage of the device for different applications. The computational imaging device provides a promising feature of offload processing from data collection to data reconstruction.

Chapter 5

Results

5.1 Performance comparison between naïve DCT, LMS DCT, and CS reconstruction

For a prescribed response-modulation factor, mandated by the DCT masks, for example, we analytically calculated the required voltage using the bias-selection algorithm as discussed in Section 3.1.1. Note that without such a statistical calculation of the voltage, the implementation of the modulation level would be inexact and would result in errors in the image reconstruction. Figure 5.1 shows reconstructed images for different compression methods with a different number of projection coefficients taken into account. The criticality of the statistical calculation of the voltages is evidenced by the presence of noise in the reconstructed images using the naïve approach, which uses bias voltages that are calculated without considering uncertainty in ROIC's implementation of the masks, as shown in Fig. 5.1(a). In contrast, the reconstruction based on a bias-selection algorithm tends to achieve a better reconstruction, as seen in Fig. 5.1(b). In addition, the CS reconstruction, as shown in Fig. 5.1(c), outperforms the DCT-based approach. For the given results, we can see

that naïve based reconstruction fails to retrieve the details of the image as well as contrast levels due to the presence of noise. However, MMSE-based results suggest that they achieve a better contrast result as well as reproduce most of the details of the original image. Note that the CS gives almost exact reconstruction when a sufficient number of coefficients is used. This is due to the fact that CS exploits randomness as a tool to extract information with fewer coefficients, and the uncertainty in responsivity has less of an implication on it compared to the DCT approach, which relies on the exact implantation of the masks. Also, for the DCT transform, a linear combination of projection coefficients, with the corresponding basis masks results in reconstruction where an error in projection is propagated during the reconstruction. CS reconstruction uses ℓ_1 minimization-based optimization, which tends to keep the reconstruction noise as low as possible. Hence, the CS-based reconstruction is more tolerant of uncertainty in electronic mask implementation due to its robust ℓ_1 optimization, whereas the DCT approach uses a ℓ_2 optimization, which is known for its inferior performance compared to ℓ_1 optimization.

A reconstruction based on ideal DCT is depicted in Fig. 5.1(d) where the given input image was projected on a set of ideal DCT masks, and the reconstruction was performed with different projection coefficients. The results demonstrated in Fig. 5.1(d) are entirely carried out at a simulation level.

The reconstruction errors which are shown in Fig. 5.1(e) are computed with respect to the ground-truth image. For MMSE and naïve methods, although the visual results are better with the higher percent of coefficients considered to lower percent for binary CS in the reconstruction process; the individual pixel values were off from the original pixels, whereas the difference was less for CS. This is because the correlation between the pixels retains the image structure and looks better for the user. Thus, the correlation of pixels for larger projection coefficients for reconstruction in naïve and MMSE are higher when compared to a lower number of projection coefficients for binary CS. However, compared to CS, because the coefficients are more

sensitive to noise for MMSE and naïve, the reconstruction error is higher. In this context, considering more coefficients in reconstruction leads to more propagation of projection error. This error is less for MMSE when compared to naïve.

Further, an efficient tool such as Cramer – Rao bound [10] can be used to assess the quality of reconstruction with the stated techniques. It can be shown that our proposed MMSE based bias selection algorithm tends to achieve the lowest possible mean squared error among all methods, and is therefore the minimum variance unbiased (MVU) estimator.

The analog image sensor has a limited memory, which forces the device to operate over a limited dynamic range; this constrains the device to rely on small, block-sized transform coding instead of a large kernel mask. This is due to, for a large block size, the mask coefficients being significantly large in number and denser. This gives rise to quantization issues as most of the neighboring coefficient values are rounded to their nearby realizable coefficients. As a result, the realized mask loses its orthonormal property, and the implemented mask is no longer equivalent to the targeted mask, leading to reconstruction errors.

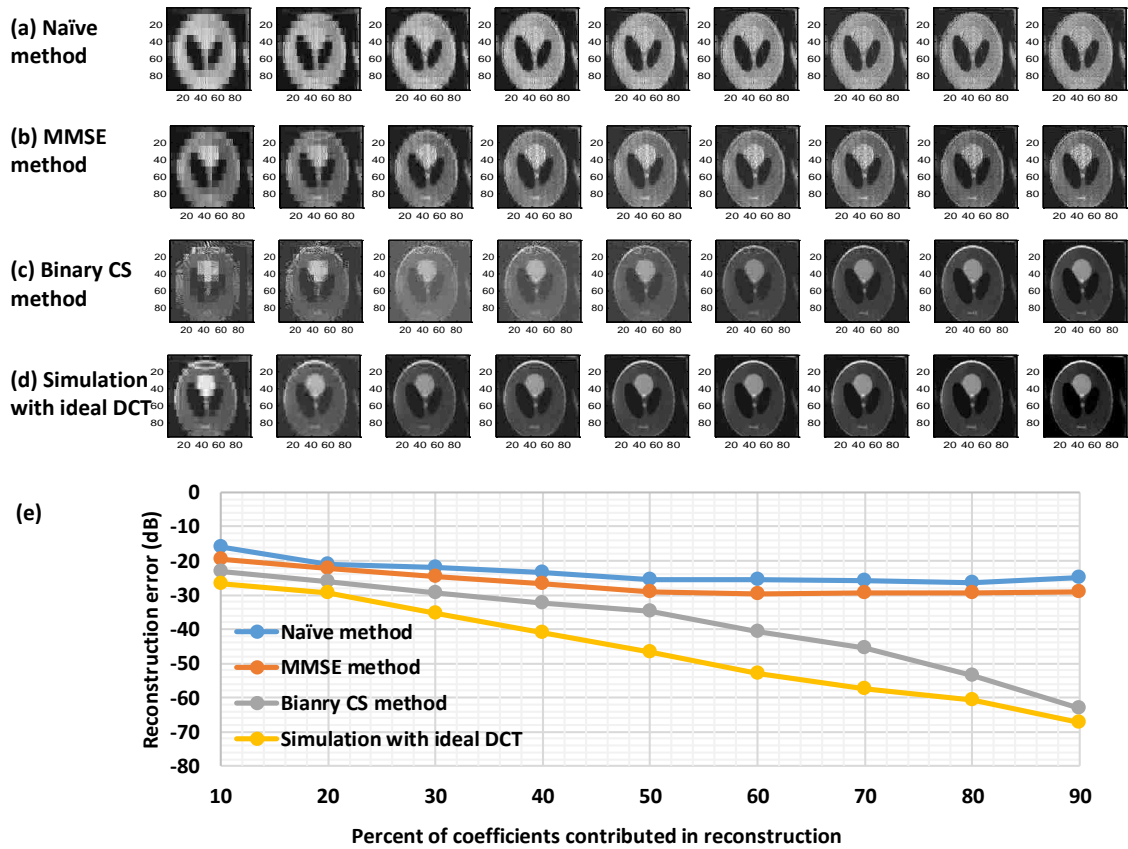


Figure 5.1: The resulting images reconstructed using a) naïve DCT, b) minimum-mean-square error based DCT, c) compressive sensing, and d) ideal DCT. e) The performance of different method is compared in terms of the mean square error between the reconstructed image and the original image.

Chapter 6

Computational Imaging and Non-Uniformity Correction

6.1 Non-Uniformity Correction

The pixels are designed to maximize the sensitivity to the photoresponse. However, the overall performance of the sensor is limited by noise, which comes from many different sources and contributes to the output signal. Random noise is a temporal variation in the signal that is not constant and changes over time, from frame to frame.

The problem of the CMOS based ROIC device is that the response of the detector drifts with time lapse. Also with external conditions such as ambient temperature and variation in transistor bias voltage results in a fixed pattern noise and it tends to degrade the image quality. So a periodic calibration is needed to overcome such issues in the imaging process. To address the problem, we can use one of the existing non-uniformity correction(NUC) algorithms such as 1) based on constant statistics,(2) based on Neural networks or (3) based on registration. However, in this work, we've

considered a scene based NUC algorithm which aims to find optimized correction parameters so that the mean square error between true object reflectance function and corrected value gets minimized.

Non-uniformity or defects in the process of manufacturing the sensor create spatial regions in the sensor with different dark currents and different sensitivity to the input radiation and this refrigeration requirement is tied to the non-uniformity as the dark current raises with temperature and the sensor essentially has a biased noisy signal that depends on temperature. Non-uniformity usually create regions with different quantum efficiency and the difference in transistor gains in readout circuitry. This creates a non-uniform gain effect so that even with no non-uniformity in the dark current, a uniformly illuminated sensor will result in a mottled, stippled or otherwise non-uniform image. The non-uniformities found in the sensor arrays can vary both from an array to array and within a given array, pixel to pixel.

The implementation of non-uniformity correction must be preceded in some manner by a non-uniformity measurement. Whereby, if the array is spatially uniform, then it is just simple to compute a single set of gain and offset values for the entire array. A non-uniform sensor array of size 96x96 requires test and measurement for each of the 9216 pixels, which needs a lot of resources like memory, power and time. And also these set of values are valid for a given ambient temperature and scene temperature. As a result of which the output coefficients may not be valid and hence may be required to periodically recalibrate the cameras.

The offset coefficients of imaging sensors are temperature dependent as discussed previously and they are usually caused due to dark currents that provide a temperature dependent signal even in the absence of input radiation. Besides this, also the sensor integration time plays a vital role in variation in offset. Here the output is directly proportional to the integration time of the dark current. Hence the corrector, even for a spatially uniform array, may have to implement a complex temperature and frame-dependent algorithm to compute the required offset. Though gain is less

dependent on temperature, some sensors may require this correction also. Hence it is advisable to develop an adaptive non- uniformity corrector that does not depend on measurements made in the lab but make the correction to pixel values in real time based solely on scene contents.

Mathematical Formulation

Approach 1: Static Correction/One time Correction

The linear model of the imaging device is given by

$$Y_n(i, j) = g_n(i, j).X_n(i, j) + o_n(i, j) \quad (6.1)$$

Here subscript n is the frame index. $g_n(i, j)$ and $o_n(i, j)$ are the real gain and offset of the (i,j)th detector. Here $X_n(i, j)$ is the actual object reflection function that is incidence on the imaging sensor and the observed pixel value is given by $Y_n(i, j)$. Considering the drift of both gains and offset slow with time, they share the same subscript n. Here Non-uniformity correction is carried out by means of a linear transformation of the observed pixel values $Y_n(i, j)$ so as to provide an estimate of the true scene $X_n(i, j)$ so that all detectors appear to be performed uniformly. And this correction is given by

$$X_n(i, j) = w_n(i, j).Y_n(i, j) + b_n(i, j) \quad (6.2)$$

where $w_n(i, j)$ and $b_n(i, j)$ are the gain and offset of the linear correction model of the (i,j) th detector and the relation to that of actual gain and offset is given by solving (6.1) and (6.2) as

$$w_n(i, j) = 1/g_n(i, j), \quad (6.3)$$

$$b_n(i, j) = -o_n(i, j)/g_n(i, j). \quad (6.4)$$

once we estimate the parameters $w_n(i, j)$ and $b_n(i, j)$ or $g_n(i, j)$ and $o_n(i, j)$, the NUC can be achieved as per equation [2].

Approach 2: Dynamic Correction

Here we used Scribner's algorithm to carry out the NUC. Here, the linear model is given by

$$Y_n(i, j) = g_n(i, j) \cdot X_n(i, j) + o_n(i, j) \quad (6.5)$$

Here, $Y_n(i, j)$ is the corrected pixel of the (i, j) th detector, $X_n(i, j)$ is the uncorrected pixel, n is the frame number, and $g_n(i, j)$ and $o_n(i, j)$ are the gain and offset correction coefficients respectively. Then with the widrow's method of gradient descent algorithm, the updates of these correction coefficients is given by

$$g_{n+1}(i, j) = g_n(i, j) - 2\mu(Y_n(i, j) - d_n(i, j))X_n(i, j) \quad (6.6)$$

$$o_{n+1}(i, j) = o_n(i, j) - 2\mu(Y_n(i, j) - d_n(i, j)) \quad (6.7)$$

Here μ is the learning constant or step size and $d_n(i, j)$ is the desired value that is taken as to be average of local neighbourhood pixel outputs. Here the pixel gain and the offset correction factor is performed by dynamically updating the gain coefficient $g_{n+1}(i, j)$ and offset coefficient $o_{n+1}(i, j)$ so that detector response is uniform.

To minimize the hardware complexity, we can reframe the equation [6] as

$$g_{n+1}(i, j) = g_n(i, j) - 2\mu \frac{(Y_n(i, j) - d_n(i, j))}{M} \frac{(Y_n(i, j) - o_n(i, j))}{Mg_n(i, j)} \quad (6.8)$$

Here M is the normalization constant. Finally, the results obtained are evaluated in terms of global NUC by means of offset estimation.

The need for pixel correction arises from the fact that the individual detectors in the FPA -array usually don't respond perfectly uniformly to a given illumination. Here, the response of any unit detector (as a function of input illumination and bias) may generally be parameterized by an offset and a slope parameter (i.e. photoresponse or responsivity). Here the offset factor is also recognized as a dark signal of the element, as it coincides with the signal produced by the element in the absence of light. As the linear model tends to be a good approximation of the physical behavior of the detector, however, complexity raises as dark signal and responsivity differ among individual pixels of the array. Therefore, the calibration of the detector is carried out offline to obtain these parameters for each detector element individually. Assuming the array consists of n detectors, ie $n=9216$ for this project, $2n$ parameters are sent to the pixel corrector before the image processing starts. Here, in order to achieve the best compression results, we need to make sure to remove any sort of systematic error, normally considered as an additive noise signal, from the source image. The pixel correction function on bias design eliminates the error introduced by the non-uniformity of the imaging sensor. To achieve NUC across all the detectors, every input signal value is multiplied by the gain factor and then increased by an individual offset parameter.

Usually, non-uniformity is produced by the mismatches during the fabrication process of the imaging sensors which can considerably degrade the spatial resolution and temperature resolvability, as it results in a fixed pattern noise (FPN) which is superimposed on the observed image. Scene correction based algorithms based on statistics usually make some spatiotemporal assumptions on the irradiance collected by each detector in the array. With these assumptions, we tend to extract some quantities to estimate the correction coefficients to correct the fixed pattern noise. Such NUC method is used because of their relatively lower computational complexity,

small storage demands as the number of correction coefficients are less and better real-time performance. Since we are mainly carrying the computation for static images, we don't worry about the motion-dependent scenes which might result in ghosting artifacts if implemented with this technique. Hence for our application, we'll show later that this technique guarantees both the convergence speed and stability.

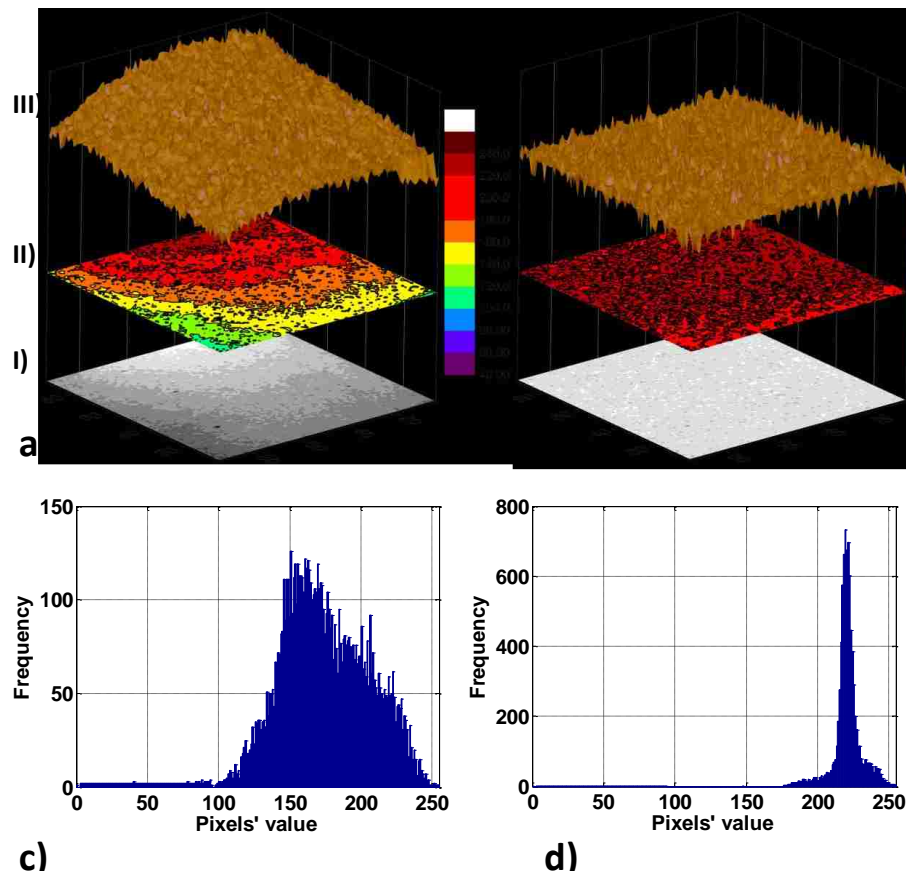


Figure 6.1: a) The result of imaging a white paper with uniform biasing, while the illumination is not uniform. Defects and other sources of nonuniformity also contribute to the variation across the image. The stack of three graphs demonstrates (I) camera output image, (II) illumination contour, and (III) 3D view of the intensities. b) Another white paper is imaged with the same illumination condition using the implemented nonuniformity correction. The graph has the same scale as part (a), and the legend in the middle is for part (II). c) and d) show the histogram for the measured results of part (a) and (b), respectively.

Figures 6.1(c) and (d) depict the histogram of the images shown in Figs. 6.1(a) and 6.1(b), respectively. While the histogram on Fig. 6.1(c) shows that it is flat as for the given non-uniform illumination, the camera results in an image with a wide range of pixel intensity level, while at the same time our NUC method resulted in a narrow histogram as shown in Fig. 6.1(d). Here, the point is that the hardware is able to cancel the integrated nonuniformity that stems in the pixels, the ROIC, and also in the illumination.

The nonuniformity correction also aided in the fine-tuning of the responsivity curves. Because the responsivity is based on the calibration of pixels under different bias conditions and different lighting conditions, enabling nonuniformity correction before this calibration process allowed a uniform behavior of responsivity through all of the pixels and less invariant toward any form of noise. This also guaranteed that the SNR of responsivity is above a certain threshold, which enabled the bias-selection technique to have superior performance as discussed over results.

6.2 Functioning as a stand-alone camera

Depending on the modulation scheme applied to the chip, different applications could be delivered. In the simplest scenario, if all of the pixels are biased with the same voltage, the iROIC camera can be used as a stand-alone camera. In this mode of operation, V_{bias} should remain constant, and as a result, the modulation factor that is used for different pixels is the same.

The extra benefit of this hardware over the conventional CTIA is that in stand-alone mode because the reference voltage for the readout is different from the detector's bias voltage, a $V_{ref} - V_{bias}$ offset is applied to the measured values, which means a level shifter is embedded in every pixel. This method is beneficial if there is a constant offset at the output of the imager. Figure 6.2 shows four images that

are taken by the iROIC camera in stand-alone mode.

6.3 Region of interest (ROI) enhancement

Another imaging dimension that is of great importance is dynamic range. While digital cameras have improved by leaps and bounds with respect to spatial resolution, they remain limited in terms of the number of discrete brightness values they can measure. Consider a scene that includes a person indoors lit by room lamps and standing next to an open window in which the scene outdoors is brightly lit by the sun. If one increases the exposure time of the camera to ensure the person appears well lit in the image, the scene outside the window would be washed out or saturated. Conversely, if the exposure time is lowered to capture the bright outdoors, the person will appear dark in the image. This is because digital cameras typically measure 256 levels (8 bits) of brightness in each color channel, which is simply not enough to capture the rich brightness variations in most real scenes. A popular way to increase the dynamic range of a camera is to capture many images of the scene using different exposures and then use software to combine the best parts of the differently exposed images. Unfortunately, this method requires the scene to be more or less static as there is no reliable way to combine the different images if they include fast moving objects. Ideally, we would like to have the benefits of combining multiple exposures

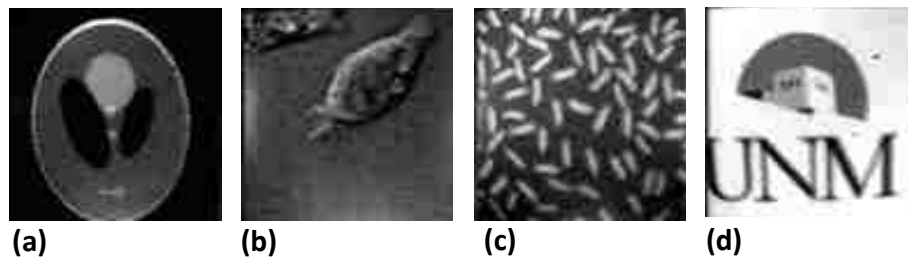


Figure 6.2: Four images that are taken using iROIC camera in normal mode. a) phantom, b) a cell, c) some rice grains, and d) UNM logo.

of a scene, but with the capture of a single image. In a conventional camera, all pixels on the image detector are made equally sensitive to light. The existing solution is to create pixels with different sensitivities either by placing an optical mask with cells of different transmittances on the detector or by having interspersed sets of pixels on the detector exposed to the scene over different integration times. Our approach would be to use the well designed electronic biases to selectively enhance the intensity levels.

The support for continuous spatiotemporal control over the bias voltage applied to each photodetector enables ROI enhancement achieved by means of selectively modulating responsivity of detectors located in the region of interest. Different applications are advantaged from this, and some are briefly discussed below:

- It aids in enhancing the contrast of image over a given region, which is originally poor due to limited dynamic range of the sensor. This is also a solution to the challenge of finding an optimum bias for a high contrast image where part of it saturated and some other part is at the noise level. A smart selection of bias voltages enforces all pixels to operate in the linear region.
- This method facilitates in achieving different resolutions for different regions of a given image by using sub-masks corresponding to low pass and high pass response. This is useful in the surveillance and medical applications, where the user may be interested in a specific region and wants to ignore the information in the rest of the image.
- Spectral selectivity in different areas of the image is another application of the hardware; however, the requirement is to have support for multispectral tunability at the photodetectors.

Figure 6.3(a) shows the original image of the white matter, which we have used at the input of iROIC in the image segmentation experiment. Figure 6.3(b) depicts the

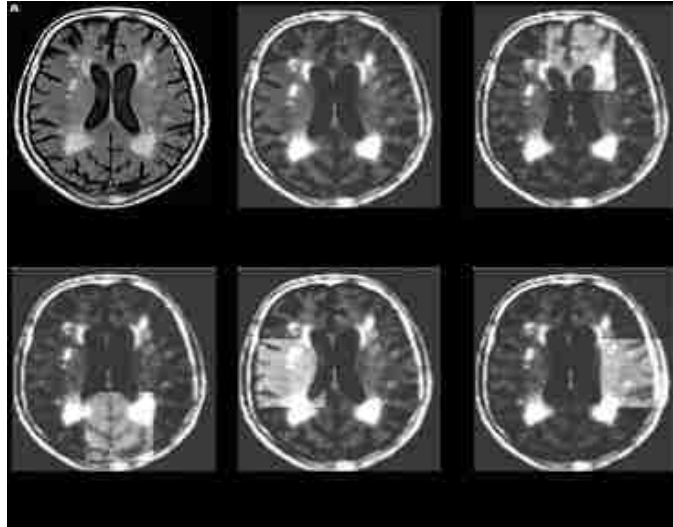


Figure 6.3: a) Original white matter image used for imaging. b) The image is taken using iROIC with a uniform biasing for all of the pixels where some of the pixels are saturated due to the high intensity. In c), d), e), and, f) the same scene is imaged using proper biasing for the different areas that normally are at the noise floor of the imager.

white matter image we have taken with iROIC when a uniform bias is applied to all of the pixels, and Figs. 6.3(c)-6.3(f) present the same scene with the exception of applying different bias to some selected area, which is referred to as region of interest.

Chapter 7

Conclusion

This work sets the stage for the implementation of compressive image acquisition where all computations are performed within the analog ROIC circuit. It enables developing cameras that directly output a reduced set of compression coefficients of an image, thereby avoiding the generation of big data. Detector-bias information is the knob we employed to control the modulation factor of each individual pixel. The reported hardware outputs a reduced set of compression coefficients of an image, thereby avoiding the generation of big data. A flexible image retrieval setup enables fine control over the matrix that is to be projected to the image. The enhanced acquisition technique, which utilizes a statistical detector biasing scheme, offers many different applications, such as in-place nonuniformity correction, sensor level region of interest enhancement, transform coding embedded in ROIC, and compressive sampling, where all of them are proven using the selection of proper biasing matrices.

Additionally, for the case of transform coding, an intelligent bias-selection algorithm is proposed, and the result is compared against the naive method. The ability of bias selection algorithm to produce robust results even in noisy environments has promoted the applicability of the hardware for real time applications. For

a prescribed intensity modulation factor, mandated by the DCT masks for example, we analytically calculated the required voltage using the bias-selection algorithm. We noticed that without such statistical calculation of the voltage, the implementation of the modulation level would be inexact, which will result in errors in the image reconstruction. The criticality of the statistical calculation of the voltages is evidenced by the presence of noise in the reconstructed images using the naïve approach, which uses bias voltages that are calculated without considering uncertainty in ROIC's implementation of the masks. In contrast, the reconstruction based on bias-selection algorithm tends to achieve a better reconstruction. In addition, the CS reconstruction outperforms the DCT-based approach. As discussed earlier, this is due to the fact that CS exploits randomness as a tool to extract information with fewer coefficients and the uncertainty in responsivity has less of an implication on it compared to the DCT approach which relies on the exact implantation of the masks. Also, the CS-based reconstruction is more tolerant to uncertainty in electronic mask implementation due to its robust ℓ_1 optimization, whereas the DCT approach uses an ℓ_2 optimization, which is known for its inferior performance compared to ℓ_1 optimization.

This work enables developing cameras that directly output a reduced set of robust compression coefficients of an image for any operating environment, thereby minimizing reconstruction errors and also avoiding the generation of big data.

References

- [1] Cave — projects: What is a computational camera?
- [2] Computational imaging: The next mobile battlefield.
- [3] Raskar, mit media lab, computational photography, imaging and video.
- [4] Wired:computational photography.
- [5] GJ Bansal. Digital radiography. A comparison with modern conventional imaging. *Postgrad. Med. J.*, 82(969):425–428, 2006.
- [6] Richard G Baraniuk. Compressive sensing. *IEEE Sig. Proc. Mag.*, 24(4), 2007.
- [7] Emmanuel Candes and Justin Romberg. l1-magic: Recovery of sparse signals via convex programming. *URL: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf*, 4:14, 2005.
- [8] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE Sig. Proc. Mag.*, 25(2):21–30, 2008.
- [9] Fred Chen, Anantha P Chandrakasan, and Vladimir Stojanović. A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors. In *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, pages 1–4. IEEE, 2010.
- [10] Harald Cramr. *Mathematical methods of statistics*. princeton, nj, 1946.

- [11] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin E Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Sig. Proc. Mag.*, 25(2):83, 2008.
- [12] A Dupret, B Dupont, M Vasiliu, B Dierickx, and A Defernez. Cmos image sensor architecture for high-speed sparse image content readout. In *IEEE International Image Sensor Workshop*, pages 26–28, 2009.
- [13] Javad Ghasemi, Manish Bhattarai, Glauco R. C. Fiorante, Payman Zarkesh-Ha, Sanjay Krishna, and Majeed M. Hayat. Cmos approach to compressed-domain image acquisition. *Opt. Express*, 25(4):4076–4096, Feb 2017.
- [14] Mohammad Javad GhasemiBenhnagi. *Intelligent ROIC for Real-time In-pixel Image Processing*. PhD thesis, The University of New Mexico, 2017.
- [15] Jeonghwan Lee, Seunghyun Lim, and Gunhee Han. A 10b column-wise two-step single-slope adc for high-speed cmos image sensor. In *Proc. IEEE Int. Image sensor Workshop, Ogunquit, ME*, pages 196–199. Citeseer, 2007.
- [16] Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle, Lawrence Carin, Guillermo Sapiro, and David J Brady. Coded aperture compressive temporal imaging. *Optics express*, 21(9):10526–10545, 2013.
- [17] Ying Luo and Rabab Kreidieh Ward. Removing the blocking artifacts of block-based dct compressed images. *IEEE transactions on Image Processing*, 12(7):838–842, 2003.
- [18] Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.
- [19] Steve Mann and Rosalind W Picard. Virtual bellows: Constructing high quality stills from video. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 363–367. IEEE, 1994.

- [20] Yusuke Oike and Abbas El Gamal. A 256×256 CMOS image sensor with $\Delta\Sigma$ -based single-shot compressed sensing. In *2012 IEEE International Solid-State Circuits Conference*, pages 386–388. IEEE, 2012.
- [21] Ramesh Raskar. Computational photography: Epsilon to coded photography. *Emerging Trends in Visual Computing*, pages 238–253, 2009.
- [22] Howard C Reeve and Jae S Lim. Reduction of blocking effects in image coding. *Optical Engineering*, 23(1):230134, 1984.
- [23] Justin Romberg. Imaging via compressive sampling [introduction to compressive sampling and recovery via convex programming]. *IEEE Sig. Proc. Mag.*, 25(2):14–20, 2008.
- [24] Subhasis Saha. Image compression-from DCT to wavelets: a review. *Crossroads*, 6(3):12–21, 2000.
- [25] Jeffrey B Sampsell. Digital micromirror device and its application to projection displays. *J. Vac. Sci. Technol. B*, 12(6):3242–3246, 1994.
- [26] Mohammed Shoaib. *Design of energy-efficient sensing systems with direct computations on compressively-sensed data*. PhD thesis, Princeton University, 2013.
- [27] Rebecca M Willett, Marco F Duarte, Mark A Davenport, and Richard G Baraniuk. Sparsity and structure in hyperspectral imaging: Sensing, reconstruction, and target detection. *IEEE signal processing magazine*, 31(1):116–126, 2014.
- [28] Yi-Ming Zhou, Chao Zhang, and Zeng-Ke Zhang. An efficient fractal image coding algorithm using unified feature and DCT. *Chaos, Solitons & Fractals*, 39(4):1823–1830, 2009.