SINGLE PIXEL CAMERA USING COMPRESSED SENSING

By

NC Qamar u Zama | NC Saqib Javaid | NC Furqan Ahmed

Submitted to the Faculty of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Telecommunications Engineering

JUNE 2015

ABSTRACT

The aim of the project was to develop a lab table version of single pixel camera. A new camera architecture that employs a digital micro mirror array to perform optical calculations of linear projections of an image onto pseudorandom binary patterns. Compressive Sensing is an emerging field based on the revelation that a small number of linear pseudorandom projections of a compressed signal contain enough information for processing and reconstruction of signal. Instead of recording an image point by point, it only records the brightness of the light reflected from rotatable micro mirror array. Each configuration of the mirror encodes some information about the scene to be captured, which the pixel collects as a single number. The camera produces a picture by pseudo randomly switching the mirrors and measuring the results several thousand times. Unlike other megapixel cameras that record millions of pieces of data and the compress the information to reduce the file sizes. The single pixel camera first compresses the data and records only the compact information. The lab version camera is slow and image quality is rough, but this technique may lead to the single pixel cameras that can collect images outside the visible range, or possibly even megapixel cameras that provide gigapixel resolution.

CERTIFICATE

It is hereby certified that the contents and form of the project report entitled " Single Pixel Camera using Compressed Sensing" submitted By 1) Qamar U Zama 2) Furqan Ahmad 3) Saqib Javaid have been found satisfactory as per the requirement of the B.E. Degree in Electrical (Telecom) Engineering.

Supervisor:

Maj. Faisal Akram

MCS, NUST

____

DECLARATION

We hereby declare that no content of work presented in this thesis has been submitted in support of another award of qualification or degree either in this institution or anywhere else.

DEDICATED TO

Almighty Allah,

Faculty for their help

And our parents for their support

ACKNOWLEDGMENTS

We would like to offer our thanks to Maj. Faisal Akram for his supportive attitude, and complete guidance throughout the completion of the project. Also our special thanks to Mr.Sanjeev and Mr. Trevor from Texas instruments for their valuable help in digital light crafter module.

TABLE OF CONTENTS

TABLE OF FIGURES

List of Abbreviations

CCD – Charge-coupled Device

CS – Compressed Sensing

CMOS - Complementary metal-oxide semiconductor

DLP - Digital Light Processing

DMD – Digital Micro-mirror Device

MDDR – Mobile Double Data Rate

RNDIS - Remote Network Drivers Interface Specification

OMP – Orthogonal Matching Pursuit

1. INTRODUCTION

1.1 Background/Motivation

The conventional digital images acquired a large amount of raw data which they often need to compress in order to store or transmit due to the storage and bandwidth limitations respectively. This compression is mainly based on the fact that an N-pixel image can be well approximated as a sparse linear combination of K<< N wavelets. These wavelet coefficients can be efficiently computed from the N pixel values and then easily stored or transmitted along with their locations. This process has two major potential drawbacks. First, acquiring large amounts of raw image needs large number of sensors or array of sensors that can be expensive, particularly at wavelengths where CMOS or CCD sensing technology is limited. Second, as the amount of raw data increases compressing this data can be computationally demanding. Sample, process, keep the important information, and throw away the rest," there may appear no way around this procedure , a new theory known as Compressive Sensing(CS) has emerged that offers hope for directly acquiring a compressed digital representation of a signal without first sampling that signal [1].

1.2 Problem Statement

The present dogma of signal processing is that a signal must be sampled at a rate at least twice its highest frequency in order to be reconstructed without aliasing (Nyquist Criterion). However, in practice, we often compress the data soon after sensing, trading off signal representation complexity (bits) for some error. For example in digital cameras, JPEG compression algorithm discards some image data when it is compressed, reducing the quality of the final file. The compression algorithm handles sharp edges and abrupt changes poorly. Clearly, this is wasteful of valuable sensing resources.

With the increase in the amount of data Imaging sensors, hardware, and algorithms are under increasing pressure to accommodate ever larger and higher dimensional datasets; Ever faster capture, sampling, and processing rates; ever lower power consumption; communication over more difficult channels. Fortunately, over the past few decades, there has been an enormous increase in computational power and data storage capacity, which provides a new angle to tackle these challenges. We could be on the verge of moving from a "digital signal processing" (DSP) paradigm, where analog signals (including light fields) are sampled periodically to create their digital counterparts for processing, to "computational signal processing" (CSP) paradigm, where analog signals are converted directly to any of a number of intermediate, "condensed" representations for processing using various nonlinear techniques [2].

1.3 Compressive Sensing

Compressive sensing processing builds upon a core principle of signal processing and information theory: that signals, images, and other data often contain some type of structure that enables intelligent representation and processing. The notion of structure has been characterized and exploited in a variety of ways for a variety of purposes. Current compression algorithms use a de-correlating transform to compact a correlated signal's energy into just a few essential coefficients. Such transform coders exploit the fact that many signals have a sparse representation in terms of some basis, meaning that a small number K of adaptively chosen transform coefficients can be transmitted or stored rather than $N \gg K$ signal samples. The commercial coding standards JPEG5 and JPEG20006 directly exploit this sparsity [1] [2].

The standard procedure for transform coding of sparse signal is to 1) gather the full N samples of signal 2) compute the complete set of transform coefficients 3) locate the K largest significant coefficients and discard the(many) small coefficients 4) encode the values and locations of the largest coefficients. This procedure can be quite inefficient if N is very large and K is small. Most of the ADC (analog to digital Conversion) process ends up being discarded. This raises a simple question: For a given signal, is it possible to directly estimate the set of large coefficients that will not be discarded by the transform coder? While this seems improbable, the recent theory of Compressive Sensing demonstrates that a signal that is K-sparse in one basis (call it the sparsity basis) can be recovered from cK non-adaptive linear projections onto a second basis (call it the measurement basis) that is incoherent with the first, where c is a small oversampling constant. While the measurement process is linear, there construction process is decidedly nonlinear [2] [1].

A critical aspect of CS measurements is multiplexing: each measurement is a function of several of the signal samples or image pixels. From this reduced set of measurements, it can still be possible (using CS techniques) to extract the salient signal information. This principle of “sample less, compute later” shifts the technological burden from the sensor to the processing. Thus, CS is an enabling frame work for the CSP paradigm. In this project we developed a system to support Compressive imaging. This system uses a micro-mirror array driven by pseudorandom and other measurement bases and a single photodiode optical sensor [2].

In CS, we do not measure or encode the K significant $\theta(n)$ directly. Rather, we measure and encode M < N projections $y(m)= (x,\varphi^T_m)$ of the signal onto a second set of basis functions $\{\varphi_m\}$, $m \in \{1,2,...,M\}$, where φ^T_m denotes the transpose of φ_m and(-,-) denotes the inner product. In matrix notation, we measure $y = \varphi * x$. Where y is an M x 1 column vector, and the measurement basis matrix φ is M x N with each row a basis vector φ_m. Since M < N, recovery of the signal x from the measurement y is impossible in general, however the additional assumption of signal sparity will makes recovery possible [1].

2. LITERATURE REVIEW

Paper [3] presents a new approach to building simpler, smaller, and cheaper digital cameras that can operate efficiently across a much broader spectral range than conventional silicon-based cameras. The approach fuses a new camera architecture based on a digital micro mirror device (DMD) with the new mathematical theory and algorithms of compressive sampling (CS). CS combines sampling and compression into a single non adaptive linear measurement process. Rather than measuring pixel samples of the scene under view, inner products between the scene and a set of test functions are measured. Interestingly, random test functions play a key role making each measurement a random sum of pixel values taken across the entire image. When the scene under view is compressible by an algorithm like JPEG or JPEG2000, the CS theory enables us to stably reconstruct an image of the scene from fewer measurements than the number of reconstructed pixels. In this manner sub-Nyquist image acquisition is achieved.

In the paper [2], new camera architecture is developed that employs a digital micro mirror array to perform optical calculations of linear projections of an image onto pseudorandom binary patterns. Its hallmarks include the ability to obtain an image with a single detection element while sampling the image fewer times than the number of pixels. Other attractive properties include its universality, robustness, scalability, progressivity, and computational asymmetry. The most intriguing feature of the system is that, since it relies on a single photon detector, it can be adapted to image at wavelengths that are currently impossible with conventional CCD and CMOS imagers.

The task of finding sparse approximations can be very difficult this is because there is no general method guaranteed to work in every situation. In fact, in certain cases there are no efficient methods for finding sparse approximations. This project considers the problems of finding sparse approximations and then examines the two most commonly used algorithms the Lasso and Orthogonal matching pursuit. For our Project we are researching the OMP method [4] [5].

For imaging in the visible spectrum, CCD and CMOS technology has allowed effective and compact airborne sensing devices to be developed for the defense industry. These devices typically acquire pixel samples by means of a full array of photon detectors. However, imaging in other wavelengths can require the need for more exotic detectors, and a vast array of these expensive and bulky detectors may no longer be feasible. This motivates the search for alternative imaging techniques which are capable of delivering good image quality while reducing the cost and bulkiness of the sensing device. Furthermore, there are computation and storage issues: the amount of data generated from a full set of pixel samples may be so large that, for storage or transmission purposes, the data must be compressed. The traditional approach, in which a full set of measurements is acquired, only for much of the information to be subsequently discarded, appears intuitively wasteful. The emerging theory of Compressed Sensing offers a potential solution to all these issues. It says that compressed images can be recovered from a significantly under sampled set of measurements, provided two conditions are met. Firstly, the image must be suitably compressible in some transform domain. Secondly, an appropriate randomized sampling scheme must be used. Central to the theory is the design of suitable CS algorithms to recover the image from the under sampled measurements. This paper concerns a new approach to single-pixel imaging which exploits CS theory. The camera design in question was first proposed by a team at Rice University and they built a proof-of-concept model for visible light [6].

3. DESIGN AND DEVELOPMENT

3.1 Working

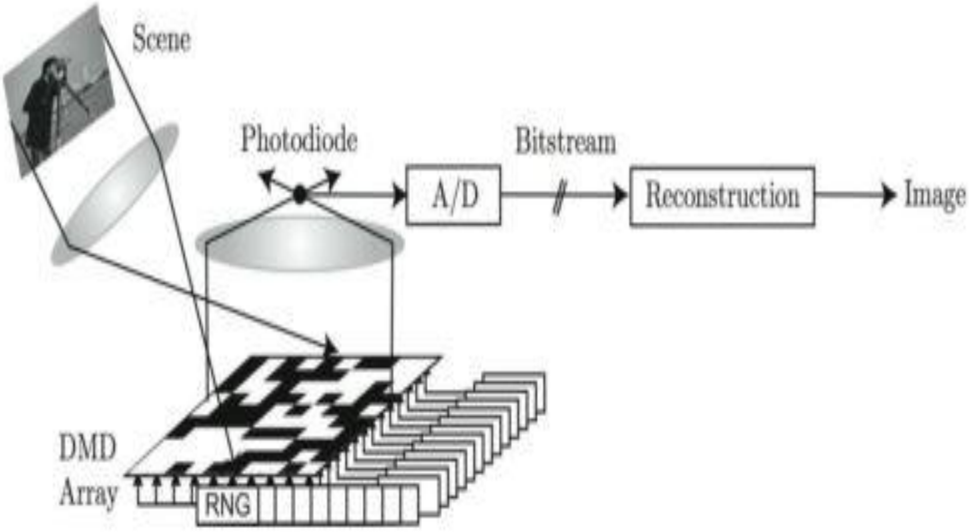


FIGURE 1: Compressive Imaging (CI) camera block diagram. Incident lightfield (corresponding to the desired image x) is reflected off a digital micro mirror device (DMD) array whose mirror orientations are modulated in the pseudorandom pattern φ_m supplied by the random number generators (RNG). Each different mirror pattern produces a voltage at the single photodiode that corresponds to one measurement y(m) [1].

1) First, the incident light from the scene is focused with a biconvex lens on a digital micro-mirror device (DMD).
2) DMD array has mirrors which can rotate to 2 directions only ($+12^o$ or -12^o). Mirrors are set according to the pseudo random binary pattern.
3) Light only reflects from the mirrors which are at $+12^o$ towards the photodiode. The -12^o pattern mirrors reflect the light somewhere else.
4) The voltage at the photodiode is the inner product between the image and the random binary (0, 1) pattern displayed on the DMD array.

$$\mathbf{Y[m] = < x, \O_m > + DC\ offset}$$

Where x is the incident light-field from the scene under view, **$Ø_m$** is a set of two-dimensional (2D) test functions.

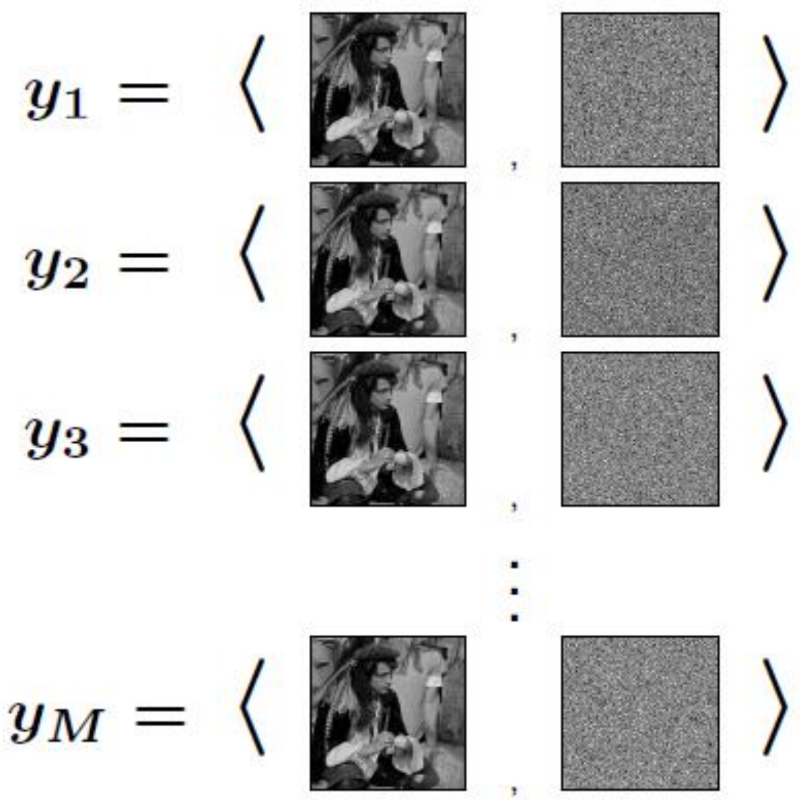


FIG 2*:* The inner product between input image and pseudo-random binary pattern of digital micro mirror device [2]

One set of mirror orientations relates to one particular measurement. Each time, the orientation of the mirrors is shifted, the output voltage differs and a new “sample” is generated. By repeating these measurements M-times with changing mirror patterns, the resulting vector contains M linear measurements of the scene, whereby M is much fewer than the number of pixels. So, this single-pixel camera performs both, sampling and compression, in its measurement process.

According to the Compressive sampling theory random measurement matrices enjoy very useful advantages regarding the incoherence to any sparsity-inducing matrix. The DC Offset can be simply calculated by switching all mirrors to -12°, so that no light is reflected through the lens to the photodiode. Data from this photodiode is sent to a signal processing unit, which manipulates data points to construct an image of the original object. To generate a set of useful data points for image reconstruction, the mirrors of the DMD are flipped into a series of random configurations, in each of which approximately half of the mirrors are in their ‘on’ position.

3.2 Decoding/Reconstruction

We need to solve a matrix equation of the form y = C*x where

• x is the signal we are looking for

• y is the measured vector and

• C is the measurement matrix.

At the expense of slightly more measurements, iterative greedy algorithms have also been developed to recover the signal x from the measurements y. Examples include the iterative Orthogonal Matching Pursuit (OMP), matching pursuit (MP), and tree matching pursuit (TMP) algorithms. OMP, for example, iteratively selects the vectors from ΦΨ that contain most of the energy of the measurement vector y. The selection at each iteration is made based on inner products between the columns of ΦΨ and a residual; the residual reflects the component of y that is orthogonal to the previously selected columns. OMP is guaranteed to converge within a finite number of iterations. In CS applications, OMP requires $c \approx 2*\ln(N)$ to succeed with high probability [2].

We used the Orthogonal Matching Pursuit for the recovery.

3.3 OMP for Signal Recovery

This section describes how to apply a fundamental algorithm from sparse approximation to the signal recovery problem. Suppose that s is an arbitrary m-sparse signal in R^d, and let $\{x_{1,\ldots\ldots\ldots,}x_N\}$ be a family of N measurement vectors. Form an N x d matrix Ø whose rows are the measurement vectors, and observe that the N measurements of the signal can be collected in an N-dimensional data vector v = Øs . We refer to Ø as the measurement matrix and denote its columns by $\varphi_{1,\ldots\ldots,}\ \varphi_d$. As we mentioned, it is natural to think of signal recovery as a problem dual to sparse approximation. Since s has only m nonzero components, the data vector v = Øs is a linear combination of columns from Ø. In the language of sparse approximation, we say that v has an m-term representation over the dictionary Ø. Therefore, sparse approximation algorithms can be used for recovering sparse signals. To identify the ideal signal, we need to determine which columns of Ø participate in the measurement vector. The idea behind the algorithm is to pick columns in a greedy fashion. At each iteration, we choose the column of Ø that is most strongly correlated with the remaining part of v. Then we subtract off its contribution to v and iterate on the residual. One hopes that, after m iterations, the algorithm will have identified the correct set of columns [7].

Algorithm (OMP for Signal Recovery): [7]

INPUT:

- An N x d measurement matrix
- An -dimensional data vector v
- The sparsity level m of the ideal signal

OUTPUT:

- An estimate s in R^d for the ideal signal
- A set Λ_m containing m elements from {1,,,,,,,,,,,d}
- An N-dimensional approximation a_m of the data v
- An N-dimensional residual $r_m = v - a_m$

Procedure:

1) Initialize the residual $r_0 = v$, the index set $\Lambda_0 = \emptyset$, and the iteration counter t=1
2) Find the index λ_t that solves the easy optimization problem
   a. $\lambda_t = \arg\max_{j=1....d} |\langle r_{t-1}, \varphi_j \rangle|$
3) If the maximum occurs for multiple indices, break the tie deterministically.
4) Augment the index set and the matrix of chosen atoms: $\Lambda_t = \Lambda_{t-1} \cup \{ \lambda_t \}$ and $\emptyset_t = [\emptyset_{t-1} \ \ \varphi_{\lambda t}]$. We use the convention that $\emptyset_0$ is an empty matrix.
5) Solve a least squares problem to obtain a new signal estimate:
   i. $x_t = \text{argmin}_x ||v - \emptyset_t x||_2$
6) Calculate the new approximation of the data and the new residual
   i. $A_t = \emptyset_t x_t$, $r_t = v - a_t$
7) Increment t, and return to Step 2 if t< m.
8) 7) The estimate s for the ideal signal has nonzero indices at the components listed in Λ_m. The value of the estimate s in component λ_j equals the jth component of x_t. [7]

3.4 Hardware Realization & Specifications

1) Digital Micro-mirror Device (DMD)

A digital micro mirror device, or DMD, is an optical semiconductor that is the core of DLP projection technology and was invented by Dr. Larry Hornbeck and Dr. William E. "Ed" Nelson of Texas Instruments (TI) in 1987. A DMD chip has on its surface several hundred thousand microscopic mirrors arranged in a rectangular array which correspond to the pixels in the image to be displayed. The mirrors can be individually rotated ±10-12°, to an on or off state.

- For 12° or 10°, mirrors correspond to on or 1 state.
  For -12°or -10°, mirrors correspond to off or 0 state.

In the on state, light from the projector bulb is reflected into the lens making the pixel appear bright on the screen. In the off state, the light is directed elsewhere (usually onto a heat sink), making the pixel appear dark.

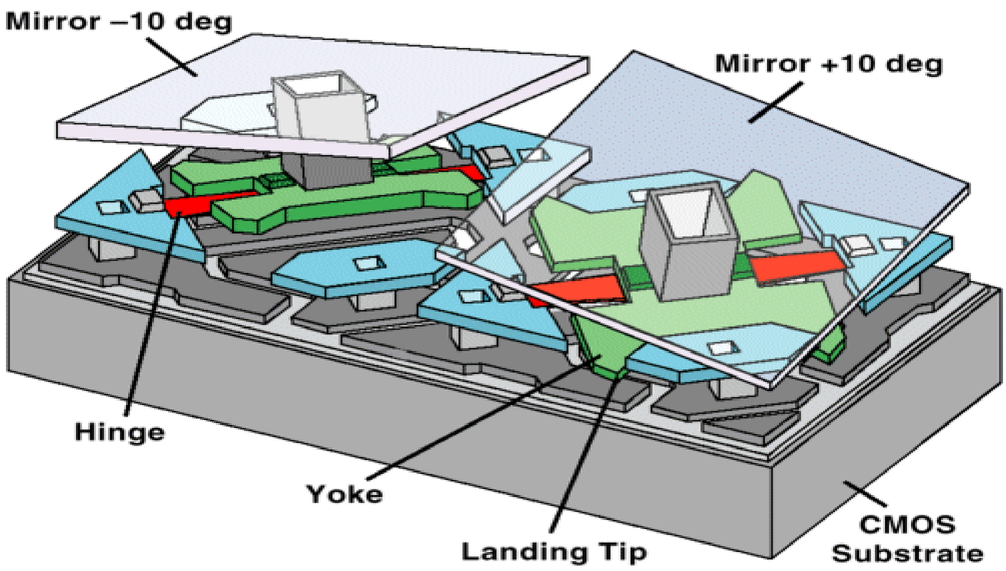


Figure 3: A portion of DMD Showing the rotation of mirrors

To move the mirrors, the required state is first loaded into an SRAM cell located beneath each pixel, which is also connected to the electrodes. Once all the SRAM cells have been loaded, the bias voltage is removed, allowing the charges from the SRAM cell to prevail, moving the mirror. When the bias is restored, the mirror is once again held in position, and the next required movement can be loaded into the memory cell.[8]

2) Digital Light Crafter Module (DMD Controller)

A DMD array is processor controlled device. The random binary pattern displayed on the DMD array is controlled by the controller which is also from the Texas instruments. The following diagram explains the interfacing of DMD array with controller.

Figure 4 Digital Light Crafter Module

The DLP Light Crafter is a third party implementation of a next generation DLP 0.3-inch WVGA chipset reference design to enable faster development cycles for applications requiring small form factor and intelligent pattern display [8].

The DLP Light Crafter module consists of three subsystems.

- Light engine – includes the optics, red, green, and blue LEDs, and the 608 × 684 pixel 0.3-inch WVGA DMD.
- Driver board – includes the LED driver circuits, DLPC300 DMD Controller, power management circuits and MSP430.
- System board – includes TMS320DM365, FPGA, and several connectors for external inputs.

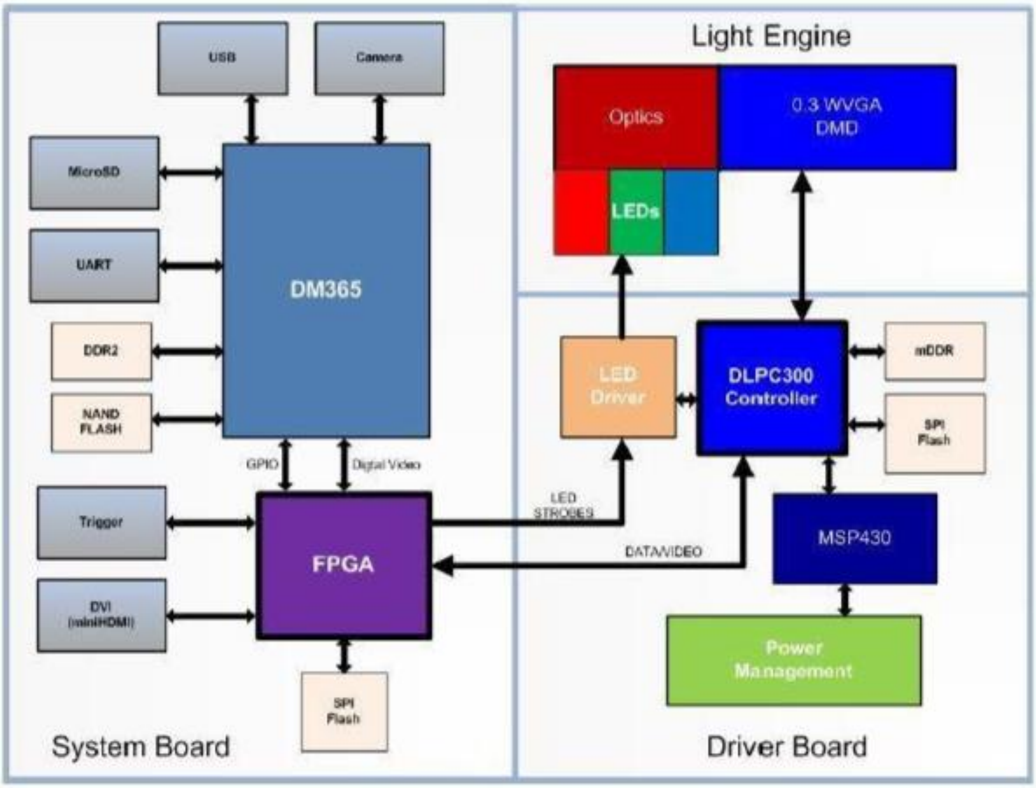


Figure 5: Different blocks of components inside the DLP light crafter evaluation module [8]

Light Engine

The DLP3000 0.3-inch DMD contains 415872 mirrors arranged in a 608 by 684 with the diamond pattern geometry shown in Figure 6

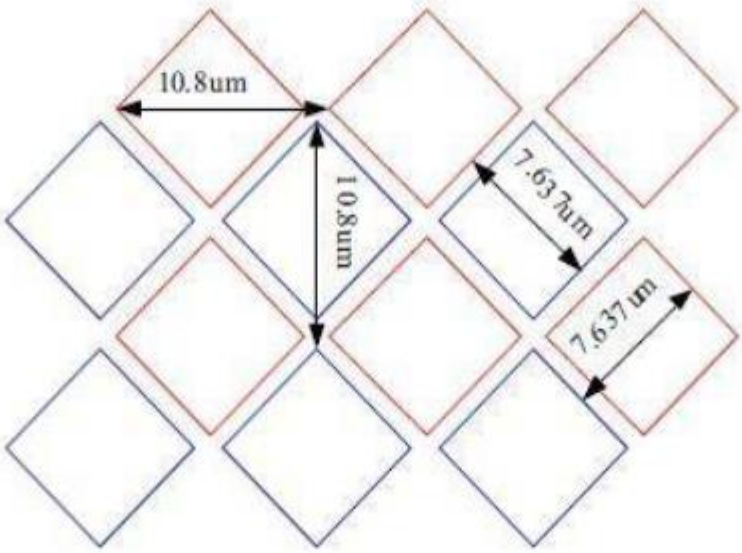


Figure 6: DMD mirrors arrangement [8]

The MSP430 monitors the light engine's thermistor to shut down the EVM if excessive heat is measured on the green LED. Passively cooled systems (no extra heat sinks or fans) have a thermal limit resulting in LED currents less than 633 mA. Actively cooled systems (extra heat sink and fan) have a thermal limit resulting in LED currents under 1.5 A [8].

Driver Board of DLP Light crafter

The major components of the DLP Light Crafter’s driver board are:

- DLP3000: 0.3-inch WVGA chipset DMD
- DLPC300: 0.3-inch WVGA chipset controller for DLP3000.
  - 2MB SPI flash that contains DLPC300 firmware
  - 32MB MDDR that buffers images for the DLP3000
- MSP430:
  - Controls power supply sequencing and system initialization
  - Shuts down system upon detection of low or high input voltage
  - Shuts down system if LED anode voltages exceed maximum limit
  - Measures thermistor and shuts down system when maximum temperature
- ratings are exceeded
- LED driver circuitry
- Power management regulators [8]

• DM365: Embedded main processor that controls camera interface, connectivity with PC,

Non-volatile storage, FPGA control, video output, and video buffer in DDR2.

– 128MB DDR2 memory

– Micro-SD connector

– Mini-USB connector and UART mini plug

• Mini HDMI connector (DVI-D compliant)

• Power management:

– TPS650531: 2 step-down converter for FPGA’s and DM365’s 1.2-V and 1.8-V supplies with three LDOs for FPGA’s 2.5-V supply and camera interface optional 2.8-V supply that contains DLPC300 firmware

– 32MB MDDR that buffers images for the DLP3000

• MSP430:

– Controls power supply sequencing and system initialization

– Shuts down system upon detection of low or high input voltage

– Shuts down system if LED anode voltages exceed maximum limit

– Measures thermistor and shuts down system when maximum temperature

ratings are exceeded

• LED driver circuitry

• Power management regulators

System Board of DLP Light crafter

DLP Light Crafter system board is based on a powerful drive TMS320DM365 digital media processor, DM365 based on TI DaVinci technology, the processor core for ARM9, speeds up to 300MHz. DM365 supports multiple video accelerator H.264 / MPEG4 / MJPEG, accessible 720p @ 30fps H.264 and MPEG4 video formats as well as meet other various resolutions. DM365 can run the Linux operating system, making it possible to focus on the development of a complete DLP 0.3WVGA chipset embedded systems. The figure is its system diagram [8].

The major components of the system board are:

- Altera Cyclone IV FPGA:
    - Controls video MUXing (external mini-HDMI or DM365)
    - Controls LEDs enables
    - Generates programmable camera triggers
    - Manages four internal buffers for fast pattern display
- DM365: Embedded main processor that controls camera interface, connectivity with PC,

- Non volatile storage, FPGA control, video output, and video buffer in DDR2.
  - 128MB DDR2 memory
  - Micro-SD connector
  - Mini-USB connector and UART mini plug
- Mini HDMI connector (DVI-D compliant)
- Power management:
  - TPS650531: 2 step-down converters for FPGA's and DM365's 1.2-V and 1.8-V supplies with three LDOs for FPGA's 2.5-V supply and camera interface optional 2.8-V supply [8].

DMD Controller

The controller for controlling the tilting of mirrors in DMD (DLP3000 in our case) is DLPC300 by Texas Instruments. Figure below illustrates the connectivity between the individual components in the chipset, which include the following internal chipset interfaces [8]:

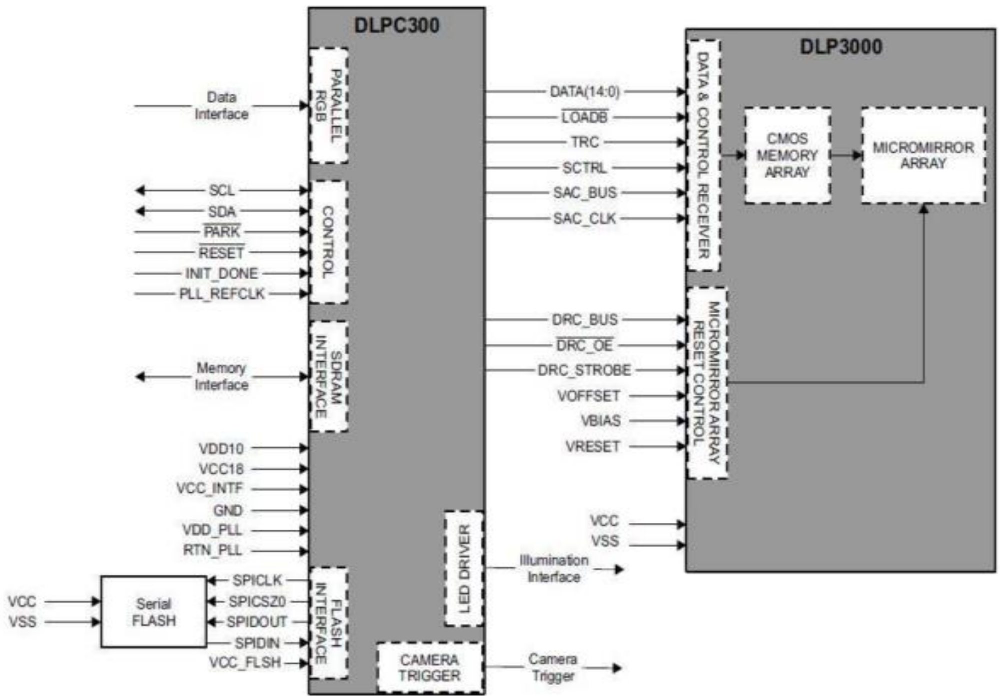


Figure 7: Illustrates the connectivity between the individual components of Chipset [8]

The figure shows the connectivity between following individual components.

• DLPC300 to DLP3000 data and control interface (DMD pattern data)

• DLPC300 to DLP3000 micro mirror array reset control interface

• DLPC300 to mobile DDR SDRAM

• DLPC300 to SPI serial flash

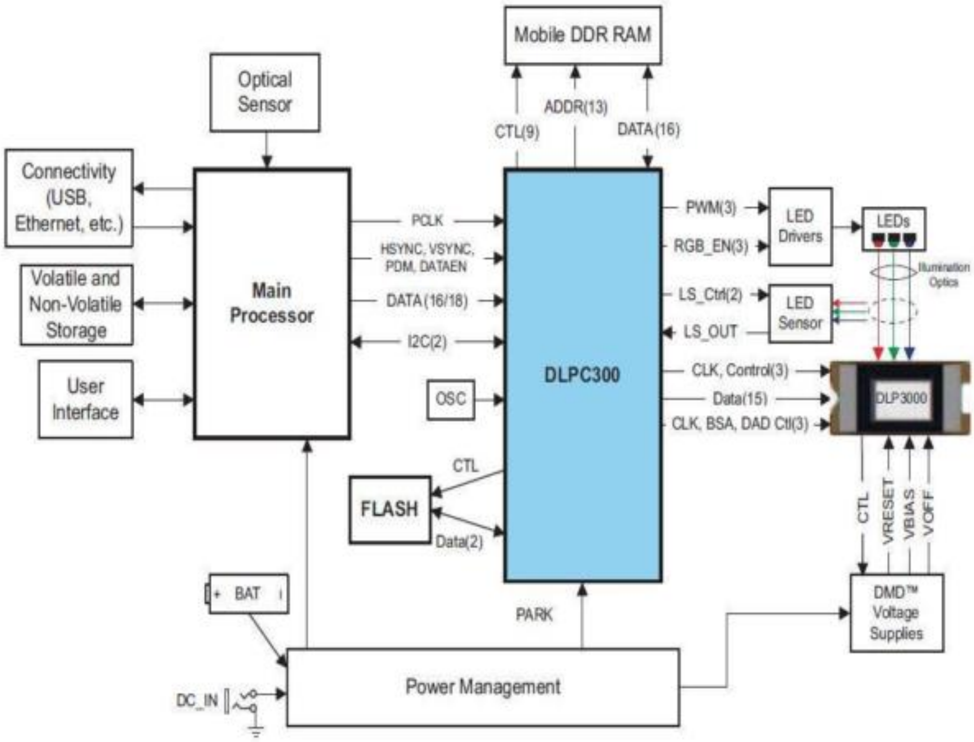


Figure 8: Illustrates the connectivity between the chipset and the other key-level components [8].

Data Interface

The data Interface is a digital video input port with up to 24-bit RGB, and has a nominal I/O voltage of 3.3V. The data interface also supports a 24-bit BT656 video interface [8].

It consists of

– 24-bit data bus (PDATA [23:0])

– Vertical sync signal (VSYNC) and horizontal sync signal (HSYNC)

– Data valid signal (DATAEN)

– Data clock signal (PCLK)

– Data mask (PDM)

Control Interface

The 0.3 WVGA chipset is supported by a set of I2C commands to control its operation. The I2C commands allow users to control in real-time or configure the 0.3 WVGA chipset. For example, the I2C commands have functions to set the LED drive current or display splash screens stored in serial flash memory [8].

It consists of

– I2C signals (SCL and SDA)

– Park signal (PARK)

– Reset signal (RESET)

– Oscillator signals (PLL_REFCLK)

System Support Interfaces

There are three system support interfaces provided by the 0.3 WVGA chipset:

- Mobile DDR synchronous DRAM (mDDR)
- Serial configuration non-volatile FLASH
- System reference clock

System Input Interface

The 0.3 WVGA Chipset supports a single 24-bit parallel RGB interface for data transfers from another device. The system input also requires that proper configuration of the PARK and RESET inputs to ensure reliable operation [8].

Controlling Of DMD

The DMD (DLP 3000) which is placed inside the DLP light crafter module is controlled through Matlab. The DLP light crafter communicates through PC by making a TCP/IP network connection on IP address “192.168.1.100” and a port of “21645” which transmits and receives the TCP/IP packets. In Matlab, a TCP/IP object is created of the same IP address and same port which allows us to communicate using Matlab with DLP light crafter. Using this connection, No of images can be transmitted on to the light crafter to be displayed on the DMD. (Code is available in appendix).

3) Photodiode And ADC

The light from a given configuration of the DMD mirrors is summed at the photodiode to yield an absolute voltage that yields a coefficient y(m) for that configuration. The output is digitized by a 10 bit analog-to-digital converter (ADC). An Arduino Uno is used for this purpose.

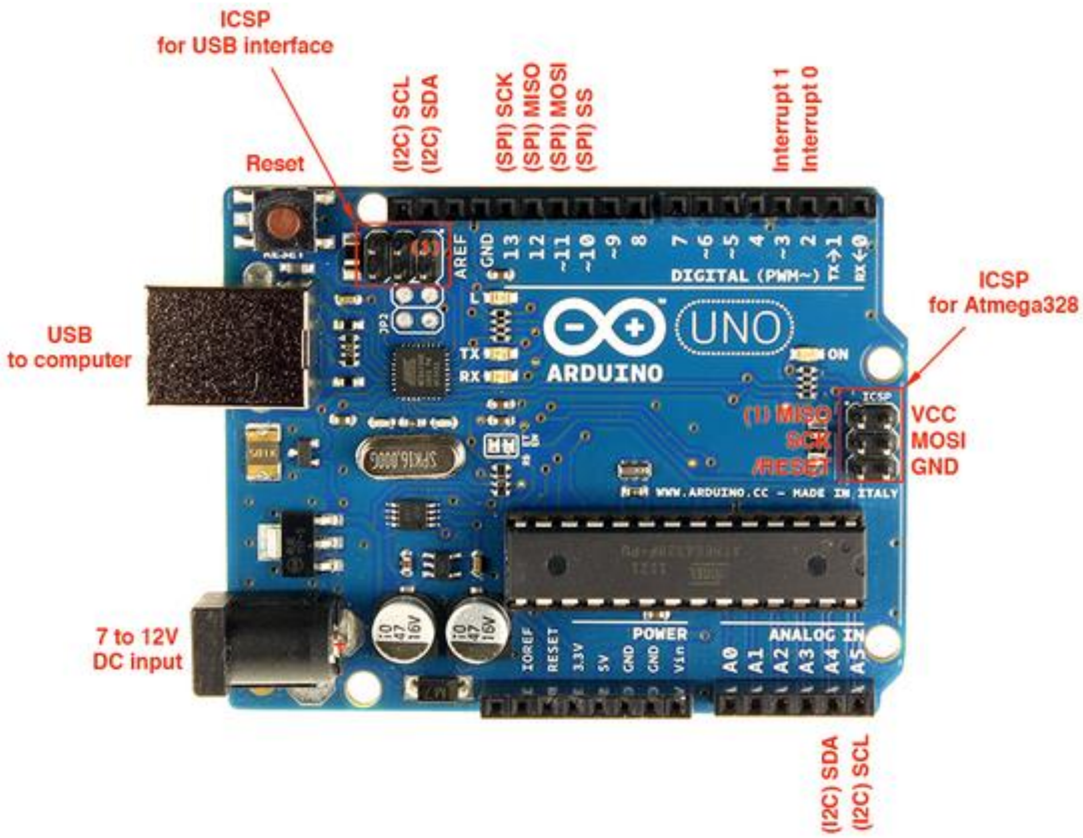


Figure 9 AURDINO UNO BOARD

The Arduino board contains 6 channels (10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and

1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit. It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

4) Biconvex lens

Two biconvex lens of 12cm focal length are used to focus the light from the image onto the DMD and from DMD to photodiode.

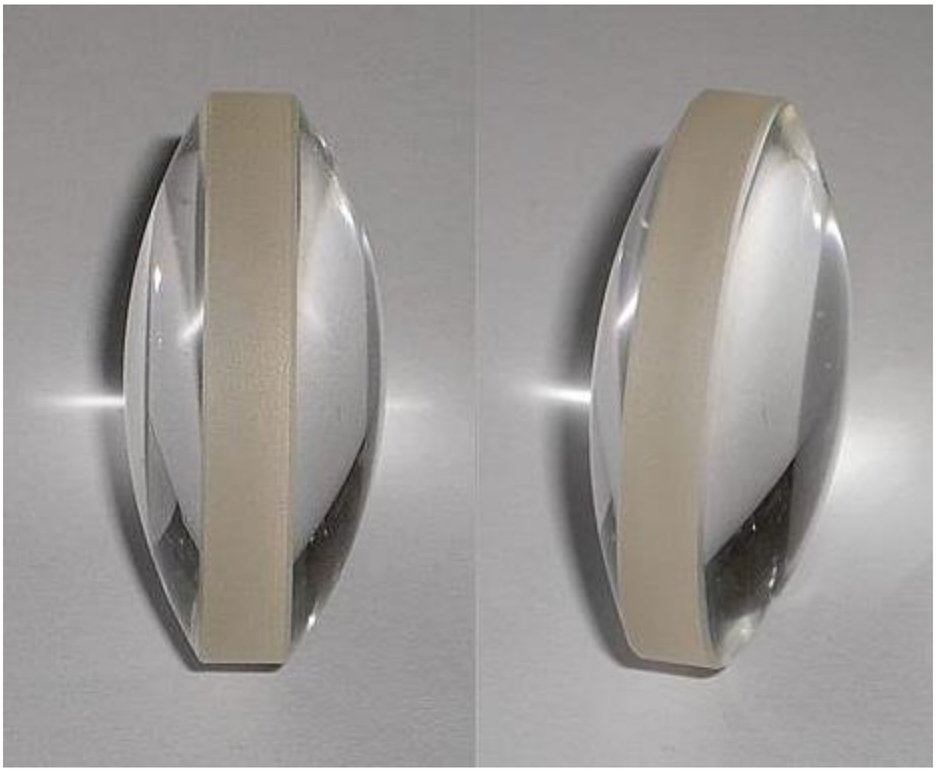


Figure 10 Biconvex lens

3.5 Problems Faced in Hardware

This project was implemented in Rice University Texas and Stanford University but they did not disclose the specific hardware they used so we ordered the DMD (DLP3000) and DMD controller (DLPC300) and we thought it is already interfaced as described by the datasheet but these were just small chips (smaller than the nail of thumb). So again we find DLP light crafter evaluation module which provided interfacing of these two chips plus the processing board DM365 for optical processing.

Since this module is expensive, we found an alternative solution that advanced projectors use DLP (Digital Light Processing) technology which we are using in our project. So instead of purchasing it, we decided on DLP board kits inside the DLP projectors and trying to use it according to our requirements.

Figure 11 Internal view of a DLP projector having DLP technology from Texas Instruments

We issued the projector, opened it and analyzed it, we found out that the controller for tilting the DMD mirrors in that projector (DDP 2000) is part of an agreement between Texas Instruments and 3rd. party projector manufacturing companies. So we were unable to access its data sheet from Texas Instruments official forum and thus unable to turn this board according to our ways. So we again proceeded with DLP light crafter evaluation module.

4. ANALYSIS AND EVALUATION

4.1- Interfacing of DLP light crafter with PC

DLP light crafter is used to control DMD mirrors in our project. It works on 5V, 2A power and transmits and receive data packets for communication through pc by using RNDIS (uses remote network drivers interface specification). It was connected to PC and a local area network was created which exchanged packets after sometime of connection and different .bmp extension images were projected on wall.

```
for K=1:100
Tm=randperm (684*608);
im1{K}=zeros (684,608);
im1{K}(Tm(1:207000))=1;
imwrite (im1{K},'im1{K}.bmp');
%load file
imFile{K} = fopen( 'im1{K}.bmp' );
imData{K} = fread( imFile{K}, inf, 'uchar' );
fclose( imFile{K} );
%test light crafter
L=LightCrafter()
%L.connect()
tcpObject = tcpip('192.168.1.100',21845)
tcpObject.BytesAvailableFcn = @instrcallback
tcpObject.BytesAvailableFcnCount = 7;
tcpObject.BytesAvailableFcnMode = 'byte';
fopen(tcpObject)
L.setBMPImage( imData{K}, tcpObject )
```

```
L.setStaticColor( 'FF', 'FF', 'FF', tcpObject )
L.setPattern('0A', tcpObject)
%data = fread(tcpObject,tcpObject.BytesAvailable);
```

4.2 – Pseudorandom Binary Pattern Generation

In the project, DLP light crafter is used to display a random pattern of '0s' (represent black) and '1s' (represent white). The resolution of the DLP 3000 is 608 by 684 mirrors. We created the image of same resolution in which more than half pixels were black (set to zero pattern) and rest were white (set to one pattern) through Matlab and projected them on DMD. A few samples of the different patterns that will appear on DMD are also provided below. The code is attached in the appendix.

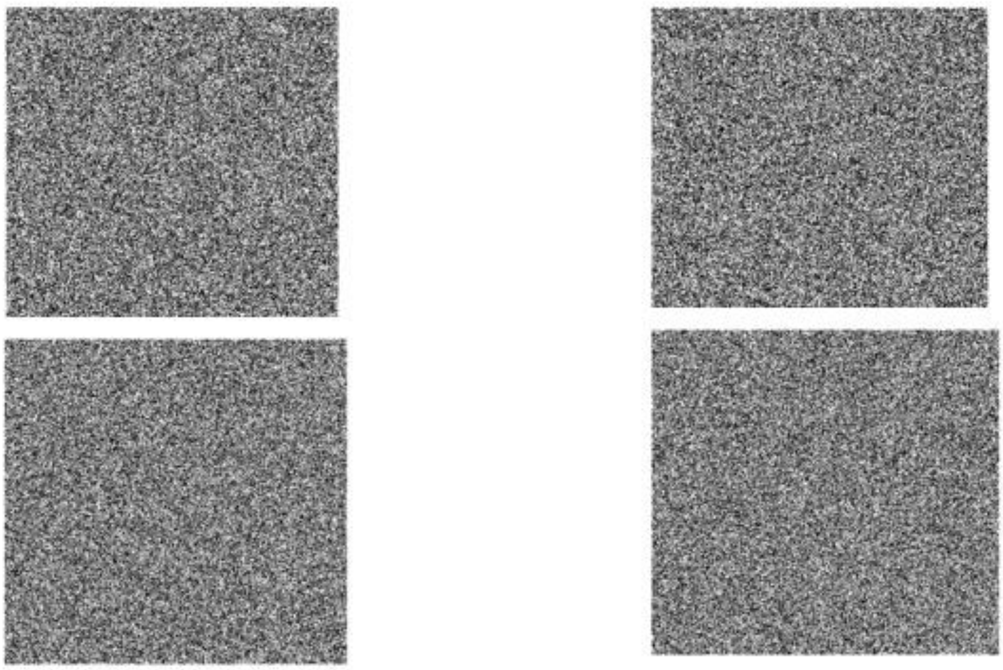


Figure 12: Pseudorandom binary patterns on DMD

4.3 – Removing the Light Engine

Light crafter module is a projector which projects the light on to the wall but in this project, we have to project the light on to it. It is only possible when DMD of the light crafter is exposed and all the light from the source fall on it.

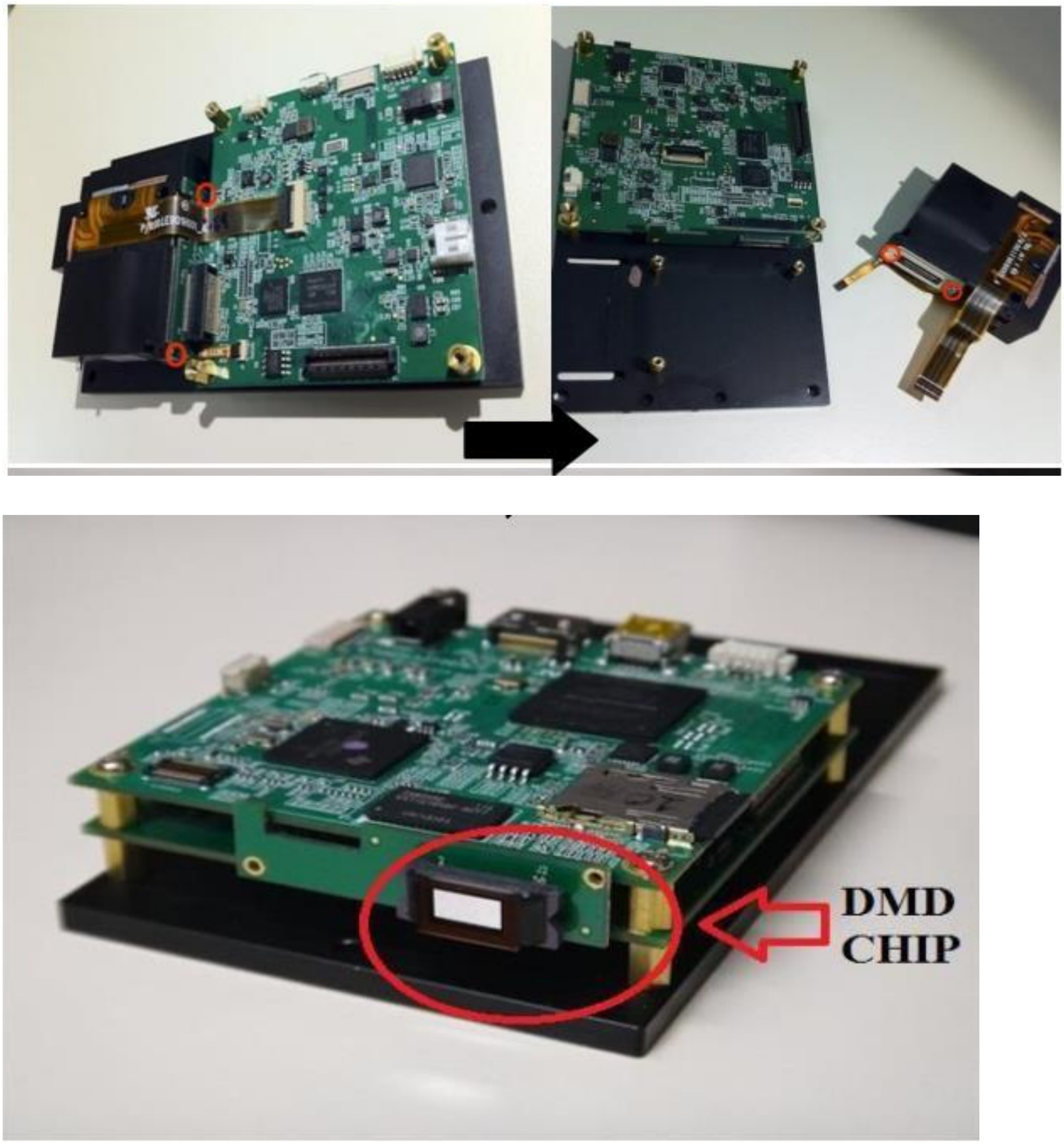


Figure 13: Removing the light engine

4.4 OMP RESULTS

- Number of samples per image is 100 (10x10)
- The total number of measurements to be carried out is selected as 10
- The dictionary matrix thus created will be (10,100)
- Error threshold is 10^{-4}

```
>> [y,d]=rvects(10,100);
>> [x,ey,err]=OMPfunc(d,y,10,1e-5);
```

```
function [x,esty,error] = OMPfunc(D,Y,M,ethresh)

dim_m = size(D,1);
dim_n = size(D,2);

r = Y;                                  %% step 1
lamb = zeros(1,dim_n);                  %% step 2
tempd = zeros(dim_m,dim_n);
x = zeros(dim_n,1);
approx_a = zeros(dim_m,1);

t =1;
threshold = 0;

while (t<=M)&&(threshold==0)
   temp_lamb = transpose(D)*r;          %% step 4
   [C,I] = max(abs(temp_lamb));
   lamb(1,t) = I;                       %% step 5


   tempd(:,I) = D(:,I);
    for j=1:1:dim_m
        tempd(j,I) = D(j,I);            %% step 6
    end

   x = tempd\Y;                        %% step 7
   approx_a = tempd*x;                 %% step 8
   r = Y - approx_a;                   %% step 9
   t=t+1;

   normr = norm(r);
   if normr <= ethresh
      threshold=1;
   end
```

OMPCODE

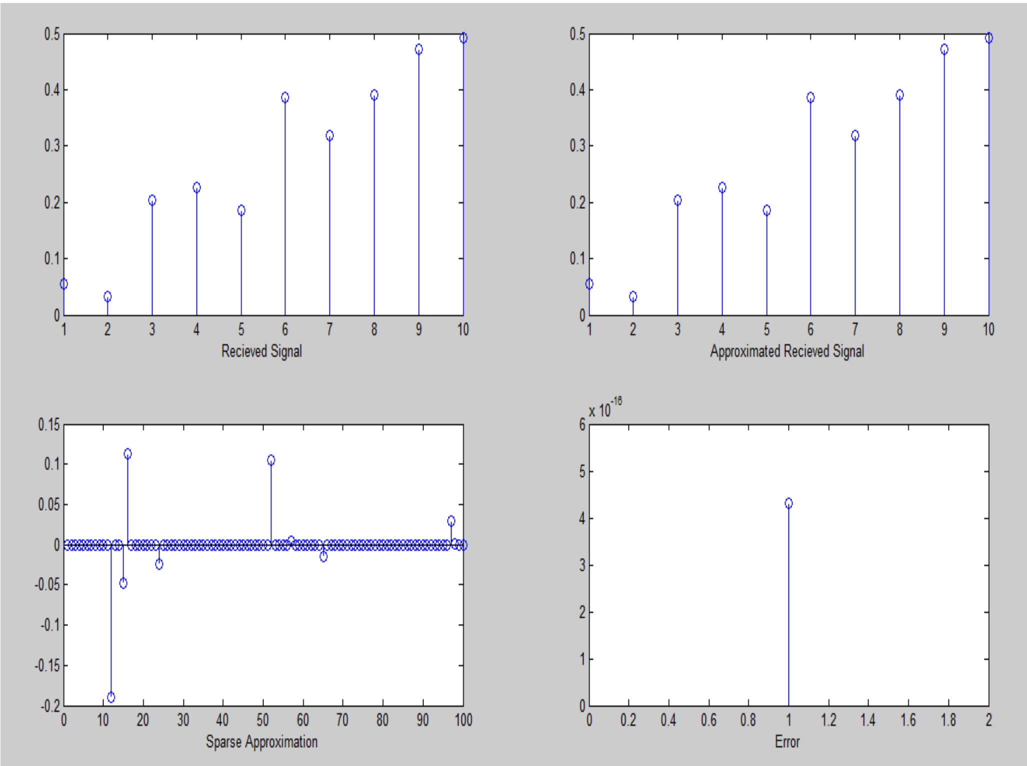
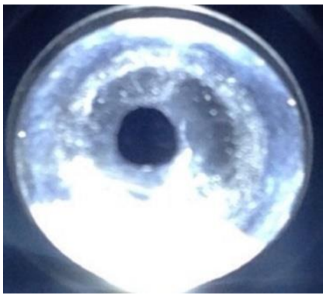


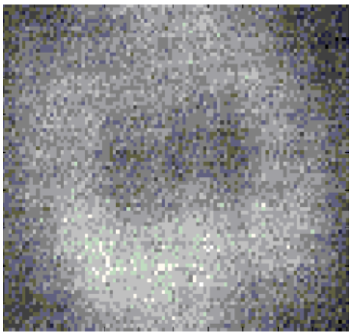
Figure 14 OMP Results

4.5 RESULTS

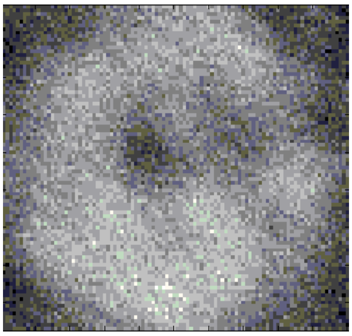
We first tried to reconstruct the letter R for our project but the photodiode values for this were very low for further processing. So we made a black dot in the middle of the led torch and used it as our image.



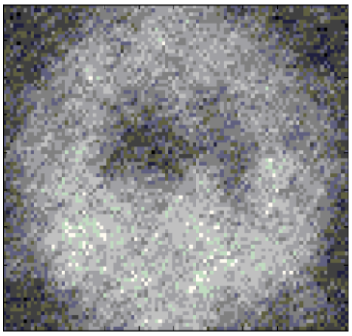
Actual Image



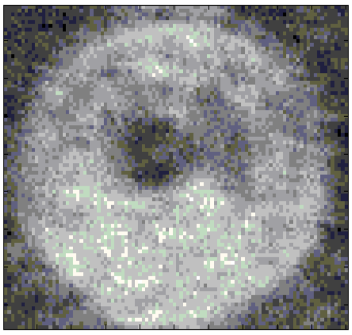
500 Measurements



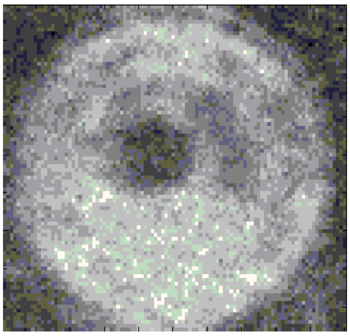
750 Measurements



1000 Measurements



1500 Measurements



2000 Measurements

Figure 15 Results

By increasing the measurements on DMD the results will be further clear. But due to the UN-optimize code it takes almost 24 hours in 2000 measurements for producing the result. So we stop on 2000 measurements as the result is quite clear to identify the original image.

The original image was of 608x684 pixels i.e. 415872 pixels. We can see in the results that only at 2000 measurements we are getting a picture that is giving us a fair idea of the original image. So we have proved here that you don't need to follow the nyquist rate for reconstruction. By following the Compressive sensing we can reconstruct even at sub nyquist rate.

5. FUTURE WORK RECOMMENDATION

- The digital micro mirror device (DMD) of higher dimensions can be used which will ensure better projection of image on itself which will lead to good results even with more less measurements. The DMD we used have the dimensions of 684 X 608. DMD up to 1920 X 1080p is available from Texas Instruments.
- Compressive Sensing is a time consuming process at the present. Recovery algorithm can take up to two days on highly advanced core i5 processor for 1500 or 2000 measurements. So in the future, recovery algorithms can be optimized to give faster and more accurate results.
- Reconstruction algorithms basis pursuit, chaining pursuit etc other than OMP can be used.
- Extending the work from greyscale to RGB domain.
- Video files can be acquired using compressive sensing in the future.

6. CONCLUSION

After analyzing all the results it can be concluded that the scope defined for this project has been successfully achieved. We can even reconstruct at sub nyquist rate by following the compressive sensing theory. Some interesting observations were observed during the project.

6.1 - Overview

Compressive sensing has some nice features.

- First, as the number of measurements increases, the reconstruction error decreases almost at optimal rate.
- It is a democratic procedure: all measurements are equally important, which makes it Robust, since losing a few measurements does not matter.
- The recovery does not depend on the acquisition, if we find a better reconstruction algorithm; we can use it on previously acquired data. Eventually, from a technological point of view, the complexity has been shifted from the sensor to the processing.

6.2 – Objectives achieved

The major objectives achieved are

- Sub-nyquist reconstruction
- Pattern generation for DMD chip
- Data acquisition from photodiode of the light coming from DMD
- Recovered the acquired data using OMP recovery algorithm on MATLAB

BIBLIOGRAPHY

[1] Michael B. Wakin, Jason N. Laska, Marco F. Duarte, Dror Baron Shriram Sarvotham, Dharmpal Takhar, Kevin F. Kelly, Richard G. Baraniuk "**AN ARCHITECTURE FOR COMPRESSIVE IMAGING**" Department of Electrical and Computer Engineering, Rice University

[2] Dharmpal Takhar, Jason N. Laska, Michael B. Wakin, Marco F. Duarte, Dror Baron Shriram Sarvotham, Kevin F. Kelly, Richard G. Baraniuk "**A New Compressive Imaging Camera Architecture using Optical-Domain Compression"** Department of Electrical and Computer Engineering Rice University

[3] Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, Richard G. Baraniuk "**Single-Pixel Imaging via Compressive Sampling**" Rice University

[4] MarkA.Davenport, Marco F. Duarte, Gitta Kutyniok "**Introduction to Compressed Sensing",** Stanford University, Department of Statistics , Duke University, Department of Computer Science, University of Osnabrueck, Institute for Mathematics

[5] D. Donoho, Y. Tsaig, I. Drori, J.L. Starck "**Sparse Solution of Underdetermined Linear Equations by Orthogonal Matching Pursuit**"

[6] Andrew Thompson "**Compressive Single-Pixel Imaging**" School of Mathematics and the Maxwell Institute University of Edinburgh, Edinburgh, EH9 3JZ

[7] J. Tropp and A.C. Gilbert, "**Signal recovery from partial information via orthogonal matching pursuit**," Apr. 2005, Preprint.

[8] DLP3000, DLPC300 DLP® 0.3 WVGA Chipset Data Manual Texas Instruments

[9] Tropp, J. A. "**Greed is good: Algorithmic results for sparse approximation**", IEEE Trans. Inform. Theory, vol 50, no 10, pp. 2231-2242

APPENDICES

APPENDIX A–CODES

The code for the single pixel camera using Compressed Sensing is given below

```
for K=1:2
Tm = randperm(684*608);
im3{K} = zeros(684,608);
im3{K}(Tm(1:(207000))) = 1;
 %   figure(K)
 imwrite( im3{K}, 'im1.bmp' );
 imshow (im3{K});
 imFile1 = fopen( 'im1.bmp' );
 imData1 = fread( imFile1, inf, 'uchar' );
 fclose( imFile1 );


 %%---------------------------------------------------
 %test light crafter
L=LightCrafter()

%L.connect()
tcpObject = tcpip('192.168.1.100',21845)
tcpObject.BytesAvailableFcn = @instrcallback
tcpObject.BytesAvailableFcnCount = 7;
tcpObject.BytesAvailableFcnMode = 'byte';
fopen(tcpObject)

L.setBMPImage( imData1, tcpObject )
L.setStaticColor( 'FF', 'FF', 'FF', tcpObject )
L.setPattern( '0A', tcpObject)


%data = fread(tcpObject,tcpObject.BytesAvailable);
%%------------------------------------------------------------------


 %% Create serial object for Arduino
s = serial('COM9'); % change the COM Port number as needed
%% Connect the serial port to Arduino
s.InputBufferSize = 1; % read only one byte every time
try
    fopen(s);
catch err
    fclose(instrfind);
    error('Make sure you select the correct COM Port where the
Arduino is connected.');
end

data(K,1) = fread(s);
fclose(s)

end
```

```
n=684*608;

y=data;
m=length(y);

Phi = randn(m,n);

Theta = zeros(m,n);
for ii = 1:n
    ii;
    ek = zeros(1,n);
    ek(ii) = 1;
    psi = idct(ek)';
    Theta(:,ii) = Phi*psi;
end


[s1,ey,er]= OMPfunc(Theta,y,m,5e-3);

x1 = zeros(n,1);
for ii = 1:n
    ii;
    ek = zeros(1,n);
    ek(ii) = 1;
    psi = idct(ek)';
    x1 = x1+psi*s1(ii);
end

figure('name','Compressive sensing image reconstructions')
IMAGE(reshape(x1,684,608))

colormap gray
```