Compressed Sensing Under Poisson Noise

Yizhou Lu *[a] and Akshay Singh Bhadoria[b]

[a]Department of Electrical and Computer Engineering, University of Wisconsin-Madison
[b]Department of Industrial and Systems Engineering, University of Wisconsin-Madison

Abstract

In this project, we investigated the Compressed Sensing method in photon counting system by simulation. As a typical application of convex optimization, Compressed Sensing is able to reconstruct an image without scanning over all the pixels. However, it stays unclear how well it can perform in dark environments, where the light shows more particle nature. In this case, the measured number of photons forms a Poisson Distribution around the ground truth. Though Compressed Sensing is famous for its robustness under Gaussian noise, our results proved its shortcoming when the Poisson noise is significant.

1 Introduction

Conventionally, signals are measured according to Shannon-Nyquist Theoremthe minimum sample rate should be twice of the maximum frequency in the original signal (1).However, a novel sampling strategy called compressed sensing (CS) challenged this convention by reducing a bunch of measurements. The key ideas in CS are the sparsity, a measure about the signal's compressibility, and the incoherence, a measure related to sensing (1).

Imagine a JEPG picture. In Discrete Cosine Transformation (DCT) domain, a picture will have greater numerical values at the left-top corner. By throwing the insignificant values in other regions, we can compress a picture by even 95%. Then this is a JPEG compression. People also noticed the Haar Wavelets domain could be better as in it, a picture will have greater sparsity.

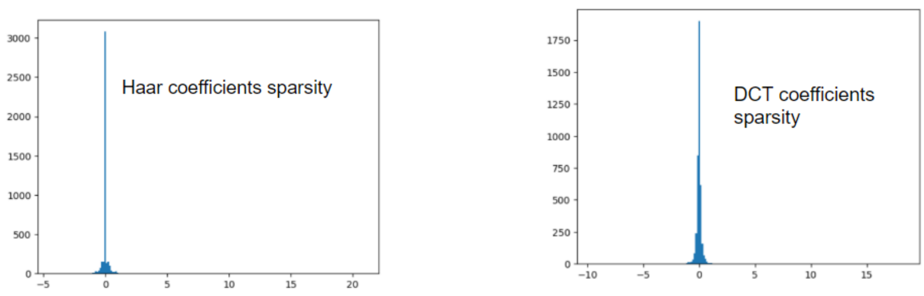

Figure 1: Haar Wavelets and DCT coefficients Histogram

Coherence is a measure of the similarity between two matrices in frequency domain. For measuring an image by CS, we have to make a series of masks with which only photons from half of the image can be collected by the sensor. A common tool is Digital Mirror Device (DMD). DMD is composed of a series of small chips which can rotate to either $+12°$ or $-12°$, so when a beam arrives at these chips, it will be split into two. So via a DMD, your sensor would just "See" parts of the view. In mathematics, we can use a $0-1$ matrix Φ to represent this mask. For each mask, we can measure the sum of photons coming from a part of the object. By changing the mask to different ones, the sensor "See"s different parts of the image and finally figure the image out.

*Corresponding author: ylu289@wisc.edu

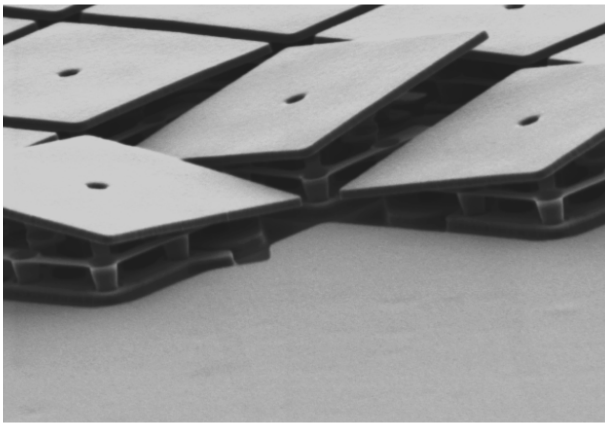

Figure 2: DMD produced by Texas Instruments

If an image is compressible, and we selected good measure strategy with small coherence, it is possible to adopt CS method in image reconstruction. The following equation (1) gives an estimate on at least how many measurements we need to reconstruct this object.

$$m \geq C\mu^2(\Phi, \Psi)S \log n \tag{1}$$

The n here is the number of pixels, and the S represents the level of sparsity. This μ is a measure of the coherence between Φ and Ψ. CS tries to find a series of masks Φ, together with the a new vector space Ψ where the image is sparse. And from this equation we see the smaller the μ, the fewer measurements we need to try.

The common single pixel measure strategies includes Raster Scan (RS), Basis scan (BS) and Compressed sensing (CS). When compared with RS, CS shows its robustness under Gaussian noise, the most common addictive noise on the original signals. The signal-to-noise-ratio (SNR) can be expressed as the ratio between the signal's intensity and its standard deviation. The Gaussian noise usually is independent on the expected measured values. In each measurement, half of the Φ is 1, and half of the image is measured. In this scenario, the intensity is greater than RS, which scans over pixel by pixel, so the SNR of CS is better than RS. Also,during the same time span, we need to measure n times to reconstruct the object by BS which measures along different masks Φ, but reconstructing directly by solving $\Phi x = y$. However, CS optimizes the coefficients in other domains such as DCT or Haar.

$$x_{n\times 1} = \Psi_{n\times n} a_{n\times 1} \tag{2}$$
$$y_{m\times 1} = \Phi_{m\times n} \Psi_{n\times n} a_{n\times 1} \tag{3}$$
$$\textbf{s.t.} \quad m \ll n \tag{4}$$

So by CS we only need m measurements. If we do CS and BS during the same time span, we will collect roughly the same number of photons. But CS only needs m measurements, so each measurement will have greater mean value, which renders a better SNR.

The applications of CS under common room brightness seems to have a bright future, but if CS can be applied onto fluorescence imaging is still unclear. Different from common imaging, fluorescence imaging requires very dark environment as a result of the low intensity from natural fluorescent agents. Nevertheless, this constraint renders Poisson noise significant. Poisson noise has different properties from Gaussian noise, such as its SNR always equals $\sqrt{\mu}$, where μ is the mean value, and it can never be added or removed by formula. To our curiosity, we would like to see if CS still the most robust strategy in the dark environments under Poisson noise.

2 Model

We simulated a process of measuring an image and reconstruct it, and the reconstructed quality can be used as an evaluation of how well CS and other strategies can perform. Julia programming and JuMP packages are used in further optimization.

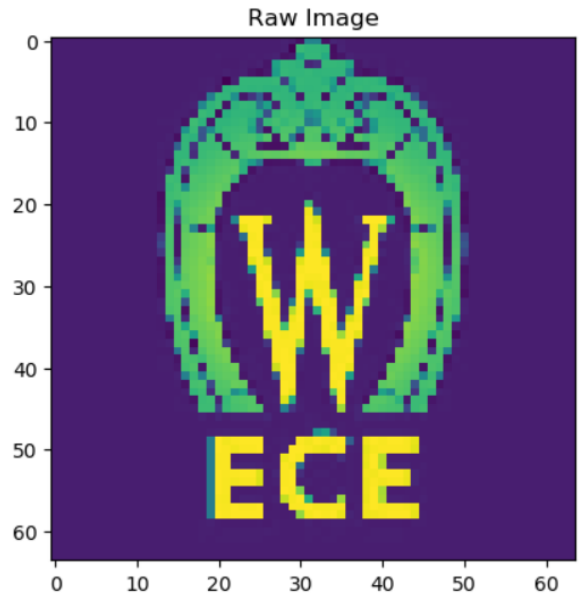

Figure 3: ECE department badge

We adopted a 64×64 pixel image of ECE department badge as the object and reshape it into a vector. Then we generated the masks Φ and get the measured intensity y. Then we got the intensity under Poisson noise, $\hat{y}_i \sim \mathcal{P}(y_i) \quad \forall 1 \leq i \leq m$. To reconstruct this image, we need to find the optimal of a satisfying the following equation:

$$\arg\min_a \quad (\hat{y} - \Phi\Psi a)^2 + \lambda \|a\|_1 \tag{5}$$

The reason for using Lasso Regression here is, in Ψ (such as inverse DCT and inverse Haar Wavelets Transform) domain, we would like to see the sparsity of this image. By Lasso regression, insignificant coefficients will thus ignored and this model won't be overfitted. In the following figure, we can see exactly that most coefficients are much smaller. Compared with Ridge regression, which tends to fit smoothly, Lasso regression can have better performance in the case of sparsity.

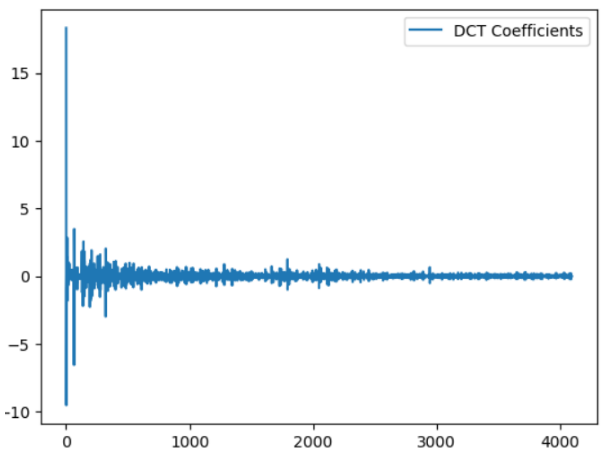

Figure 4: Typical image DCT coefficients distribution

However, this optimization is too expensive for JuMP optimizer. So in order to save the CPU and memory, we chaneged this objective into another form (2).

$$\hat{a} = \arg\min \|a\|_1 \tag{6}$$
$$\textbf{s.t.} \quad (\hat{y} - \Phi\Psi a)^2 \le \epsilon \tag{7}$$

Finally we decided to the define our model in the following form.

$$\hat{a} = \arg\min \|a\|_1 \tag{8}$$
$$\textbf{s.t.} \quad |\hat{y}_i - \Phi_i \Psi a| \le \epsilon \quad \forall_{1 \le i \le m} \tag{9}$$

In this project, we will try on 2 domains, the Haar wavelets and also the DCT, and compare the their performance on CS.

3 Objectives

We are trying to find the optimal between the following two targets during a certain time period. The first one is we want to measure as few as possible, and the second one is we want to have better quality in reconstruction. If you measure too few, each measurement can have longer exposure time and better SNR, but as a result, reconstructed a may not be accurate. However, if we measure too much, the each measurement may not have enough exposure time to encounter Poisson noise. So, if we set a certain time period, we can find for this optimal value. The bad news is, even reconstructing one image may take too long by JuMP, thus we just set the number of measurements integer variables to find the best one.

1. **Optimal compression rate**
   Assume we have a certain time t, and we can measure 10^6 photons per second. Then we try with different compression rate to find the quasi optimal. The compression rate here can be explained as measurements number per pixel. Normally we have to measure each pixel once, but by CS, we can see this measurement numbers could be smaller than one in average.

$$\textbf{variable} \quad m \tag{10}$$
$$\textbf{s.t.} \quad 1 \le m \le n, \quad m \in \mathbb{N} \tag{11}$$
$$\textbf{s.t.} \quad |\hat{y}_i - \Phi_i \Psi a| \le \epsilon \quad \forall_{1 \le i \le m} \tag{12}$$
$$\arg\min_{m} \quad \|a\|_1 \tag{13}$$

2. **Optimal scan strategy**
   People used to compare the reconstruction between RS, BS and CS under Gaussian noise versus time. Their results shows CS has the fastest mean square error convergence. Here we will try this on Poisson noise to see if CS is still the most robust one.

4 Virtual measurement

1. **Decision on Φ**
   To acquire the measured data $\hat{y}$, we need to decide on what masks to choose. The popular mask is "Walsh 0-1 matrix".

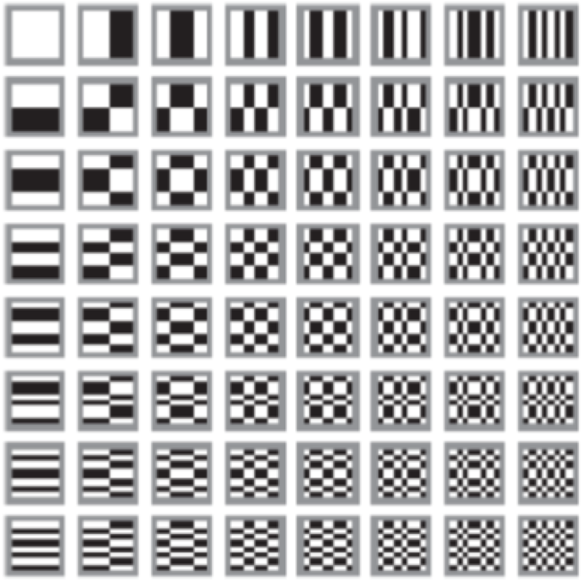

Figure 5: Walsh-Hadamard basis

For each mask in the figure above, we get get a measured y_i. This value cannot be used, as the Poisson noise hasn't been taken into consideration. So what we exactly get in this virtual imaging experiment is $\hat{y}_i \sim \mathcal{P}(y_i)$. Also, we don't have to select all the masks in this case, or we cannot pursue a compressed strategy. It is suggested to randomly select parts from these masks, and there is another saying that we can just randomly generate masks from Gaussian distribution and apply the **sgn** function.

2. **Decision on $\mathbf{\Psi}$**
Two of the popular Ψ domains are DCT and Haar wavelets.

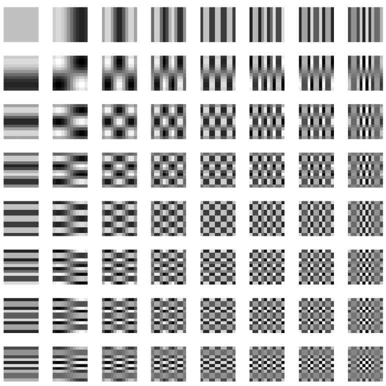

Figure 6: DCT basis

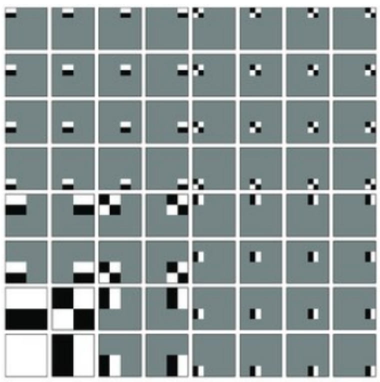

Figure 7: Haar basis

For each mask of Ψ, we get a different sum. That is the forward transformation to calculate the $a_i = \sum_{r,c} \text{mask}_{i,r,c} \times \text{image}_{r,c}$. The 2D transformation can be expressed as $A = DXD^\top$, where A is the transformed coefficients, D is the transformation matrix,and X is the raw image. The term $D_i X D_i^\top$ is the same as $\sum_{r,c} \text{mask}_{i,r,c} \times \text{image}_{r,c}$. By $X = D^{-1} A D^{-1^\top}$, we can find the relationship between forward calculation of A from X and the back calculation of X from A. Reshaping this image into a vector, and we can get the final Ψ.

3. **Impacts of brightness**
The measured intensity obeys Poisson distribution, where the SNR in this case is the square root of the ground truth. Theoretically, the brighter the environment is, the better SNR our measurement will have. Thus, the image's brightness is also an important parameter in this virtual measurement.

This parameter can be expressed as how many photons come from the image into our virtual sensor. In our first step, we assume our sensor can only collect 10^6 photons per second. Then we can see the number of photons per mask per pixel is a function about the exposure time and number of measurements. These two variables are thus our most interested ones for investigation.

5 Results

1. **Reconstruction quality VS compression rate**

   (a) **Haar basis reconstruction**

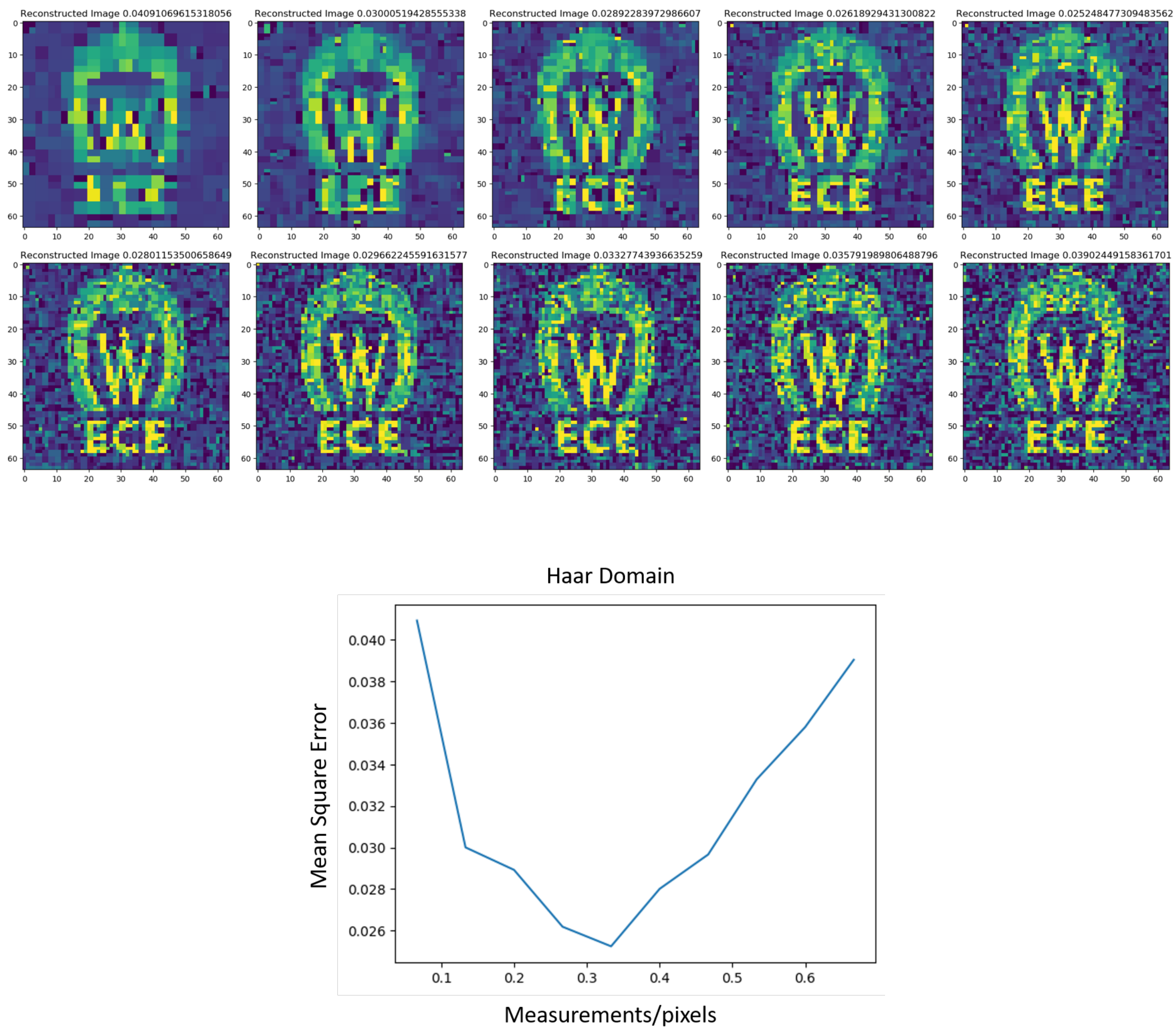

Figure 8: Reconstruction in Haar Wavelets domain exposed for 100 seconds

   (b) **DCT basis reconstruction**

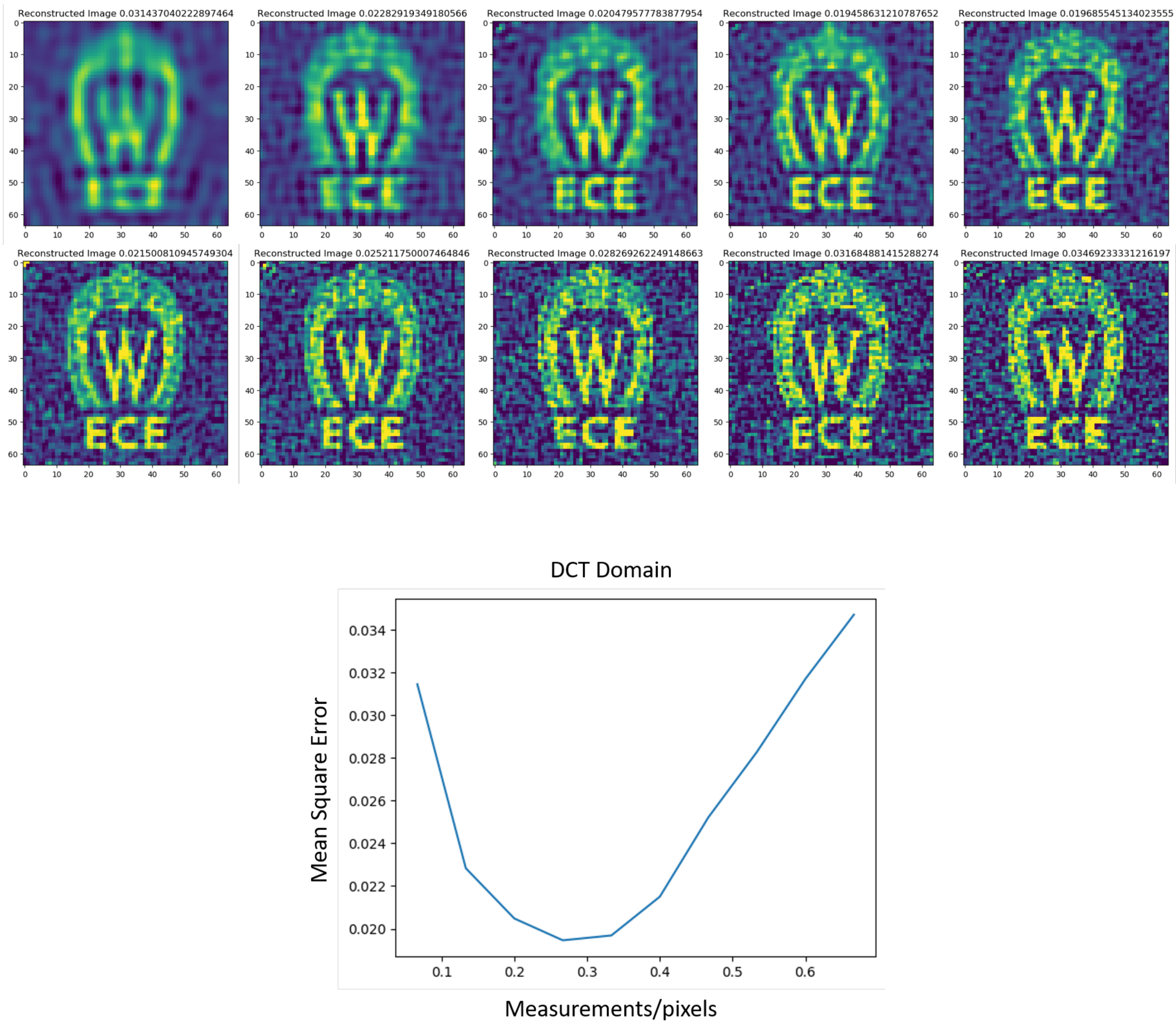

Figure 9: Reconstruction in DCT Wavelets domain exposed for 100 seconds

(c) **Objective 1 conclusion**
The exposure time is 100 seconds for both Haar and DCT condition. Here we can see 0.3 is the quasi optimal for compression. And also, DCT domain seems slightly better than Haar domain. We can keep this compression rate for the next projects.

2. **The optimal measurement time**
When extend the exposure time, we see the optimal moves to the right. The reason is that each frame or each mask can be exposed for 3 times of the original exposure time. As a result, each $\hat{y}$ can have $\sqrt{3}$ times of SNR. Because both increasing the exposure time and increasing the number of masks can improve the reconstructed image, the new optimal will also requires more masks.

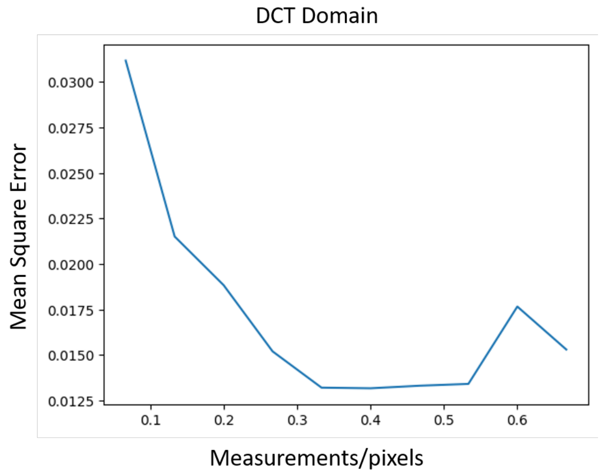

Figure 10: Reconstruction in DCT Wavelets domain exposed for 300 seconds

3. **The optimal scan strategy**
   Here we compare Raster Scan, Basis Scan and Compressed Sensing under Poisson noise. Raster scan means the sensor loops through all the pixels and doesn't need any matrix operation to reconstruct the image. In this strategy, each pixel obeys an independent Poisson distribution. Basis Scan and Compressed Sensing measures half of the image at one time, so the sum of this part will obey an independent Poisson distribution. BS uses a full rank Φ and reconstructs the image by matrix inverse, while CS only use a few rows in this Φ and reconstructs the image in frequency domain.

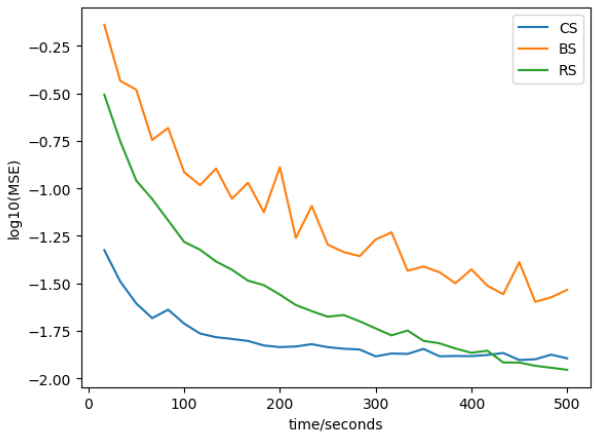

Figure 11: Different scan strategies under Poisson noise

6 Discussion

From the figures above, we can see even we only had quasi optimal about the compression rate, CS still gave the best reconstruction. In this optimization, two factors are fundamental. The first one is the Poisson noise, which tries to ruin our image reconstruction by bring much uncertainty. The other one is the number of measurements. During a certain time span, fewer measurements leads to longer exposure time per mask. Consequently, we can still achieve a better SNR. The first step of determining the optimal is not in the best approach, but is just a compromise due to the gigantic cost in this computation by JuMP. Now, just ignore if 0.3 is the true optimal in this optimization, and focus on why the last figure has this tendency.

1. **CS is still the most robust in darkness**
   When looking at the short exposure time region, we see CS gave much better results than BS and RS. Our reason is, when the number of photons are tiny, Poisson noise is significant. CS, owing

to its property of compression, can collect much more photons for each measurement. Though we can not measure every pixel even once during this process, the closer numerical values for clusters of pixels by CS can still help us figure out what is in this image.

2. **Skellam distribution**
It is super surprising to see BS, which has excellent performance compared with RS under Gaussian noise, losing the game under Poisson noise. Gaussian noise adds values obeying normal distribution with certain variance to the measured values, so BS which collects photons from half of the image has greater SNR than RS which only measures 1 pixel for each measurement. But Poisson noise is different, as the sum of variables obeying Poisson distribution still obeys Poisson distribution. Maybe this sum has greater SNR, each pixel in this sum, however, doesn't has any extra benefit. In the other word, summing more pixels up in each measurement cannot increase the SNR for each pixel. As a result, the upper limit of BS is RS in front of Poisson noise. But what is the reason rendering BS even worse?

In BS, each measurement obeys an independent Poisson distribution, and a pixel can be involved in multiple measurements. This factor requires us to do not only addition but also subtraction among all the measurement vectors, the rows in Φ.

$$\textbf{RS:} \quad \hat{y} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x \tag{14}$$

$$\hat{y}_i \sim \mathcal{P}(x_i), \quad \forall i \in \{1, 2\} \tag{15}$$

$$\textbf{BS:} \quad \hat{y}_i = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} x \tag{16}$$

$$\hat{y}_1 \sim \mathcal{P}(x_1), \quad \hat{y}_2 \sim \mathcal{P}(x_1 + x_2) \tag{17}$$

In RS, we don't need do any thing else, and just let $x_i = \hat{y}_i$. But in BS, we have $x_1 \sim \mathcal{P}(x_1)$ and $x_1 + x_2 \sim \mathcal{P}(x_1 + x_2)$. So the x_2 we actually calculated is $x_2 \sim (\mathcal{P}(x_1 + x_2) - \mathcal{P}(x_1))$. In this formula we notice x_2 obeys the subtraction between two Poisson distributions, which is exactly a Skellam distribution.

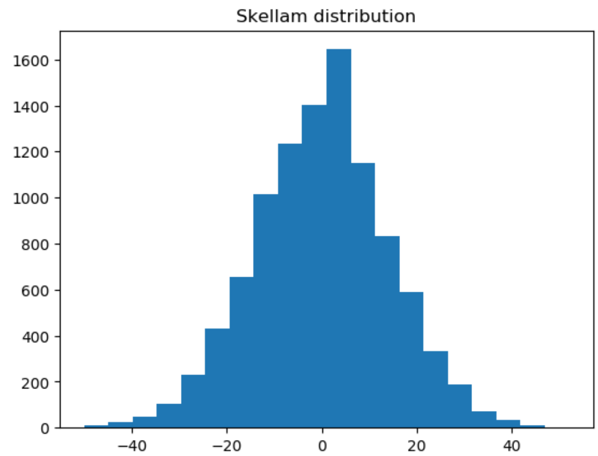

Figure 12: Numerically simulated Skellam distribution

As a result, x_2 has a mean of x_2, but a variance of $2x_1 + x_2$.

$$\textbf{SNR}_{x_2} = \frac{x_2}{\sqrt{2x_1 + x_2}} \leq \sqrt{x_2} \tag{18}$$

That answers why BS has worse performance under Poisson noise.

3. **Shortcomings**
At last we would like to recognize some shortcomings in this project.

(a) The first shortcoming is, the reconstruction optimal for CS is not accurate. We selected "ratio", standing for the time of measurements over the number of pixels, which could be any positive values between 0 and 1. However, because the number of measurements is an integer, thus it is exactly a integer convex optimization. But due to the high calculation expense in this project, we abandoned our original design and switch to using ten integers to get the rough optimal measurement times.

(b) The model. We expected to set our objective as

$$\underset{a}{\textbf{minimize}} \quad \|\Phi\Psi x - \hat{y}\|_2^2 + \lambda\|a\|_1$$

Also due to the high calculation costs, we modified our model to

$$\underset{a, \quad \text{s.t.} \quad \|\Phi\Psi x - \hat{y}\|_2^2 \leq \epsilon}{\textbf{minimize}} \|a\|_1$$

Part of the objective is set as a constraint, but we cannot guarantee the optimal to be right at the boundary of it.

(c) As we see in the first and the second results, the longer exposure time will push the optimal compression rate to the larger values. But when plotting the log(MSE) VS exposure time, the compression rate is set by a hard constraint of 0.3. This may answer why the RS exceeds CS when the exposure time is long enough. Another possible reason is also the Skellam distribution.

(d) The selection of the masks Φ may not be good enough. In our projects, we preferred to "turn multiple adjacent chips on" at the same time. This means we may lose some information about the details in this raw image. So this process is similar to inverse Haar Wavelets transformation that we start with reconstruct a low-resolution image and improve the resolution by more measurements. In the ideal case, Φ is a noiselets basis while Ψ is Haar Wavelets basis. But JuMP may not be able to deal with imaginary parts, so this design is still imperfect.

(e) We still have no idea which one between CS and BS will have better performance when exposed long enough, even though they both suffer from Skellam distribution. So far, I still have no idea how to estimate the error in CS caused by Poisson noise and the compression, but I would like to post my expectation here. If we still use the compression rate of 0.3, BS will finally defeat CS, because Poisson noise becomes less and less significant as the time goes on. However, if in every step we find the optimal compression rate for CS, it will always have better performance until it converges with BS at the infinity. When it comes to the comparison between CS and RS, I would say RS will win this game is exposed for enough time, because summing thousands of pixels together won't improve the SNR under Poisson noise, and will finally suffers from Skellam distribution. But it could be another story if we take both Gaussian and Poisson noises into consideration.

7 Conclusions

This project is a pure simulated work. In this project, we investigated the performances between Raster Scan, Basis Scan and Compressed Sensing under Poisson noise. The results shows that Compressed Sensing is still a robust sensing strategy. Nevertheless, Basis scan, which used to perform well under Gaussian noise, suffers from Skellam distribution from matrix operation and performs worse than Raster Scan.

As both Compressed Sensing and Basis Scan are affected by Poisson noise a lot, the further work may be orientated to optimizing the system with both Gaussian and Poisson noise. We planned to find a new method instead of JuMP optimizer, such as Stochastic Gradient Descent, to find the optimal faster with less computation costs, so that we can focus on improving the shortcomings we posted above. Afterwards, we would like to apply our discovery into the field of Medical Fluorescence Imaging to provide a new tool in live-capture of tumors during surgeries.

Acknowledgement

Thanks to Dr. Line Roald for consistent patience in this class even when the whole campus under the impacts of COVID-19 and her understanding in our situation where one team member dropped this class

without telling us. This inspired me to donate a bunch of efforts onto this project to try to make it a perfect ending of this class. The knowledge from her class brings me new ideas in my research and also sharpens my skills in Julia programming.
As Julia's inventors said in 2012:
Even though we recognize that we are inexcusably greedy, we still want to have it all. About two and a half years ago, we set out to create the language of our greed. It's not complete, but it's time for an initial 1 release — the language we've created is called Julia. It already delivers on 90% *of our ungracious demands, and now it needs the ungracious demands of others to shape it further. So, if you are also a greedy, unreasonable, demanding programmer, we want you to give it a try.*
Because of these words, I generated a crazy idea: using Julia to solve a problem nobody knows if feasible. We are satisfied with our assumption, our efforts, and our results, but I know this job stills need further improvements. The reason is simple: I am greedy, unreasonable and demanding.

Yizhou Lu

Codes

https://github.com/Lorenzo-lu/CS-524-Optimization-Project-Julia-

Authors

Yizhou Lu (Lorenzo)(ylu289@wisc.edu). PhD student in Electrical Engineering. He received a B.S (not Basis Scan!) degree in Geochemistry at Nanjing University in China in 2017 and a M.S degree in Materials Science and Engineering in the University of Wisconsin-Madison in 2019. His research interests involves designing a computational camera for fluorescence imaging.
Akshay Singh Bhadoria(abhadoria@wisc.edu(opens in new tab)). I am a M.S. student in Industrial and Systems Engineering. I received B.S. degree in Industrial and Production Engineering in India in 2019. My interests are Supply Chain Analytics and Quality Engineering.

References

[1] E. J. Candès M. B. Wakin. **An introduction to compressive sampling**. Signal Processing Magazine. 2008.

[2] Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N.Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk. **Single-Pixel Imaging via Compressive Sampling**. Signal Processing Magazine. 2008.