Compressive sampling methods applied to 2D IR spectroscopy

Jonathan James Humston

University of Iowa
COMPRESSIVE SAMPLING METHODS
APPLIED TO 2D IR SPECTROSCOPY

by

Jonathan James Humston

A thesis submitted in partial fulfillment
of the requirements for the Doctor of Philosophy
degree in Chemistry in the
Graduate College of
The University of Iowa

December 2017

Thesis Supervisor:  Associate Professor Christopher M. Cheatum
This is to certify that the Ph.D. thesis of

Jonathan James Humston

has been approved by the Examining Committee for
the thesis requirement for the Doctor of Philosophy degree
in Chemistry at the December 2017 graduation.

Thesis Committee:  

Christopher M. Cheatum, Thesis Supervisor

____________________________________________
Mathews Jacob

____________________________________________
Mark Arnold

____________________________________________
Alexei Tivanski

____________________________________________
Sarah Larsen
To my children, my reason for it all.
A man goes to knowledge as he goes to war, wide-awake, with fear, with respect, and with absolute assurance. Going to knowledge or going to war in any other manner is a mistake, and whoever makes it will live to regret his steps.

Carlos Castaneda

*The Teachings of Don Juan*
ACKNOWLEDGEMENTS

It is hard to imagine my life changing more in a six-year period than it did while I was a graduate student. Since starting my graduate work, I met my wife, got married, got a puppy, became an Ironman triathlete, bought a house, and became the father of two amazing boys. You could say I became an adult. Less tangible, but more to the point here, I learned what it means to be a scientist, and how to be one.

I must thank my wife, Tina, for her unfailing support through these years. I don’t know where I would be right now if not for her. I think she truly believes I can do anything, and she is happy and willing to let me try. For her understanding of my commitments to work, for the inspiration I get from seeing her commitments to her own work, for the way she is raising our children and the life we have built together, for more than I can say, I thank her.

My parents are a blessing to me and I am grateful to them for all their help, from watching the boys, walking my dog, bringing over lunch, to a million other things. Always with open minds and open arms, they are there whenever I need them. I thank them.

I thank all my siblings for their support and interest in my success. And specifically, to the three who achieved their chemistry PhDs before me, they blazed the trail that brought me to this point.

And finally, I thank my advisor Chris Cheatum. He is truly a mentor to me in every sense of the word. It has been an honor and a joy to work with him for six years.
ABSTRACT

Two-dimensional infrared spectroscopy (2D IR) is a powerful tool to investigate molecular structures and dynamics on femtosecond to picosecond time scales and is applied to diverse systems. Current technologies allow for the acquisition of a single 2D IR spectrum in a few hundreds of milliseconds using a pulse shaper and an array detector, but demanding applications require spectra for many waiting times and involve considerable signal averaging, resulting in data acquisition times that can be many days of laboratory measurement time.

Compressive sampling is an emerging signal processing technique to reduce data acquisition time in diverse fields by requiring only a fraction of the traditional number of measurements while yielding much of the same information as the fully-sampled data.

Here we combine cutting-edge 2D IR methodology with a novel compressive sampling reconstruction algorithm to reduce the data acquisition time of 2D IR spectroscopy without distorting lineshapes. We introduce the Generic Iteratively Reweighted Annihilating Filter (GIRAF) algorithm re-engineered to the specific problem of 2D IR reconstruction and show its effectiveness applied to various systems, including those with low signal, with multiple peaks, and with differing amounts of frequency shifting.

Additionally, we lay the groundwork for 2D IR microscopic imaging using compressive sampling in the spatial image domain. The first instance of a single-pixel camera in the infrared is introduced.
I perform experiments with lasers, and what it is that I do is called spectroscopy. We use the interaction of light and matter to learn something about the matter. If I can measure certain properties of light, such as intensities and colors, before it passes through a sample, and then measure them again after the sample, the difference in these two measurements will be due to the sample.

Specific chemical bonds absorb light of specific frequencies in specific ways. Molecules are made up of bonded atoms, so every molecule has a sort of fingerprint of its bonds absorbing light. Like taking a snapshot of what is there, we can use spectroscopy to determine the structure of a molecule.

Our research goes further by interacting multiple laser pulses with a sample with precise time delays between them. Like a strobe light, this shows us not only what is there but also how it is changing.

One area we work in is looking at the motions of an enzyme as it helps to catalyze biological reactions. The experiments are difficult and the amounts of data we collect are massive. My research has focused on improving the methods for collecting data while collecting it more efficiently. Further, we lay the groundwork for the development of a microscope to generate images from the data we collect. This is analogous to magnetic resonance imaging, only we will be able to answer new and different questions by looking at processes on much faster timescales.
TABLE OF CONTENTS

LIST OF TABLES .................................................................................................................. viii
LIST OF FIGURES ............................................................................................................... ix

1. MOTIVATION AND OVERVIEW ............................................................................... 1

2. GROUP TESTING AND COMPRESSIVE SAMPLING .......................................... 4
   Motivation and Introduction ............................................................................................ 4
   Nonadaptive Group Testing .......................................................................................... 8
   Sparsity ......................................................................................................................... 11
   Data Compression ......................................................................................................... 13
   Compressive Sampling ................................................................................................. 16

3. COMPRESSIVE SAMPLING IMAGING ................................................................. 23
   Motivation and Introduction ......................................................................................... 23
   Approaches to Imaging ................................................................................................. 27
   The Digital Micromirror Device ................................................................................ 30
   The Visible Single-Pixel Camera ................................................................................ 32
   The Single-Pixel Microscope .................................................................................... 37
   The Infrared Single-Pixel Camera ................................................................................ 41
   2D IR Compressive Sampling Microscopy ................................................................... 43

4. COMPRESSIVELY SAMPLED 2D IR THAT PRESERVES LINESHAPE INFORMATI ON .......................................................... 47
   Motivation and Introduction ......................................................................................... 47
   Experimental Methods .................................................................................................. 49
   Results and Discussion ................................................................................................. 53
   Conclusions ................................................................................................................... 60

5. THE GIRAF RECONSTRUCTION ALGORITHM .............................................. 63
   Motivation and Introduction ......................................................................................... 63
   Algorithm Advancements ............................................................................................. 65
   The Effects of Frequency Shifting ................................................................................ 70
   The Effects of Multiple Peaks ....................................................................................... 81

6. CONCLUSIONS AND FUTURE WORK ............................................................. 89

A. APPENDIX: DATA COLLECTION CODES ......................................................... 92
   The 2D IR Data Collection Code in LabVIEW ............................................................ 92
   Operating the GIRAF Reconstruction Algorithm ...................................................... 93
   Reconstruction on the Cluster .................................................................................... 95
   Determining the Optimal Value of the Parameter lambda0.......................................... 98
   Centerline Slope Analysis on the Cluster ................................................................... 101

REFERENCES ............................................................................................................... 103
LIST OF TABLES

TABLE 4.1 CLS PARAMETERS WITH COMPRESSIVE SAMPLING ...................... 59
TABLE 5.1 SIGNAL, NOISE, AND SNR OF GIRAF FROM NOisy SIMULATED
DATA ....................................................................................................................... 80
TABLE 5.2 DIAGONAL PEAK DECAY FIT PARAMETERS .................................. 84
TABLE 5.3 RELATIVE AMPLITUDE DECAY FIT PARAMETERS ....................... 85
# LIST OF FIGURES

**FIGURE 2.1** A SPARSE IMAGE .................................................................................................................. 5
**FIGURE 2.2** RELATIVE TESTING COST OF GROUP TESTING ..................................................... 7
**FIGURE 2.3** NONADAPTIVE RESULTS OF THE 12-COIN PROBLEM ............................................... 9
**FIGURE 2.4** SPARSE IMAGE IN THE FOURIER DOMAIN ................................................................. 11
**FIGURE 2.5** 2D IR DATA IN TIME AND FREQUENCY DOMAINS ........................................... 12
**FIGURE 2.6** SQUARE WAVE RECONSTRUCTED FROM COMPRESSED DATA ..................... 14
**FIGURE 2.7** INTUITIVE EXAMPLE OF CS RECOVERY ALGORITHM ........................................... 17
**FIGURE 2.8** FULLY SAMPLED PHANTOM IMAGE ........................................................................... 18
**FIGURE 2.9** DMD SAMPLING PATTERN AND MEASUREMENTS ........................................ 20
**FIGURE 2.10** COMPRESSIVE SAMPLING RECONSTRUCTED IMAGES OF PHANTOM .............. 21
**FIGURE 3.1** DATACUBE ....................................................................................................................... 25
**FIGURE 3.2** SIMPLE 2X2 IMAGE ....................................................................................................... 27
**FIGURE 3.3** DIGITAL MICROMIRROR DEVICE .................................................................................... 30
**FIGURE 3.4** VERSATILITY OF SAMPLING PATTERNS ON THE DMD. ........................................ 32
**FIGURE 3.5** SINGLE-PIXEL CAMERA TABLETOP SETUP ........................................................... 33
**FIGURE 3.6** SCREENSHOT OF SPC DATA COLLECTION CODE ................................................... 34
**FIGURE 3.7** SPC IMAGE OF WRENCHES ......................................................................................... 35
**FIGURE 3.8** FILTERING SPC DATA ..................................................................................................... 36
**FIGURE 3.9** SCHEMATIC SPC MICROSCOPE TABLETOP ............................................................. 37
**FIGURE 3.10** CAMERA IMAGE OF OFF MIRRORS .......................................................................... 38
**FIGURE 3.11** SPC MICROSCOPE IMAGE ......................................................................................... 40
**FIGURE 3.12** SPC INFRARED MICROSCOPE IMAGE ................................................................. 42
**FIGURE 3.13** SCHEMATIC OF 2D IR APPARATUS WITH COMPRESSIVE SAMPLING IMAGING ......................................................................................................................... 45
**FIGURE 4.1** 2D IR SAMPLING MASKS ................................................................................................. 51
**FIGURE 4.2** 2D IR SPECTRA COMPRESSIVELY SAMPLED ............................................................ 54
**FIGURE 4.3** 2D GAUSSIAN FIT PARAMETERS ..................................................................................... 55
1. MOTIVATION AND OVERVIEW

Two-dimensional infrared (2D IR) spectroscopy is a powerful tool to investigate molecular structures and dynamics on femtosecond to picosecond timescales. 2D IR is applied to diverse systems by a growing number of researchers around the world, yet it remains a specialized technique. 2D IR spectrometers are becoming more available to the greater scientific community as the practical barriers to 2D IR research are being reduced by the development of commercially available parts.

Even with the necessary instrumentation, a barrier that remains to widespread use of 2D IR to answer interesting problems is it requires a great amount of data to be collected. Although current technologies allow for the acquisition of a single 2D IR spectrum in a few hundreds of milliseconds using a pulse shaper and an array detector, demanding applications require spectra for many waiting times with considerable signal averaging resulting in data acquisition times that may be many days or weeks of laboratory measurement time. Long data acquisition times are a longstanding problem in multidimensional spectroscopies.

We introduce compressive sampling techniques to 2D IR to accelerate data acquisition times. The emerging signal processing field of compressive sampling has shown the information content in a signal can be recovered from a fraction of the traditional number of measurements. By reducing the time for collection of a 2D IR data
set, without changing the apparatus and while accurately reproducing line shape information, we hope to enable challenging new applications of 2D IR and make 2D IR more accessible. Compressively undersampling will result in a loss of signal strength, however we find that it also results in a proportionately greater reduction of noise and show that the signal-to-noise ratio of a 2D IR spectrum actually increases when compressively sampled.

Compressive sampling calls for a new way of thinking about data and information. The following chapter conceptually introduces group testing and data compression leading to an understanding of compressive sampling. Then, we outline the steps taken to apply compressive sampling to microscopy with the single-pixel camera, with the goal to realize 2D IR spectroscopic imaging. Spectral features that overlap and interfere in linear FTIR imaging can be effectively separated and spread across two dimensions with 2D IR leading to an imaging modality that, if realized, would open new applications in chemical imaging. We aim to build a 2D IR microscope using compressive sampling to acquire the spatially heterogeneous information.

Chapter 4 presents the first demonstration of pseudorandom compressive sampling techniques applied to 2D IR. We use the standard Total Variation (TV) reconstruction algorithm to reduce the time for collection of a 2D IR data set of a particularly demanding application from 8 to 2 days while preserving line shape information. This requires no changes to the apparatus and only changes the programming of data collection.

Finally, we introduce the compressive sampling reconstruction algorithm called the Generic Iteratively Reweighted Annihilating Filter (GIRAF) re-engineered with our
colleagues, Mathews Jacob and coworkers, to specifically solve the recovery of 2D IR spectra. We demonstrate the improved ability of GIRAF reconstruction to recover 2D IR spectra, test the difference between compressively sampling 2D IR with and without frequency shifting, and show that GIRAF successfully and accurately reconstructs spectra with multiple peaks. By combining frequency shifting and compressive sampling we show that 2D IR spectra may be acquired from as few as ten measurements, a greater than $16\times$ reduction from current practices.
Motivation and Introduction

Group testing is a good conceptual introduction to compressive sampling. Briefly, group testing is the procedure of simultaneously testing a large group or collection of elements in order to locate a few elements that have certain properties. Individual testing, in contrast to group testing, is the procedure of administering a test to every individual element. In searching for a needle in a haystack, would you examine every straw in the stack to see if it is a needle, or examine a pitchfork-full?

Like finding Waldo at a busy market, consider acquiring a sparse image in which we know that one pixel out of N contains all the intensity and all other pixels have zero intensity. The task is to find the high pixel using the smallest number of measurements. The individual testing approach to test one pixel at a time requires N measurements to guarantee and answer. Group testing is a powerful approach to collect such thinly dispersed signals and devising an appropriate data collection scheme, called a pooling design, is an important first step. To acquire the image, seen in Figure 2.1A, a group testing measurement scheme is shown in Figure 2.1B in which white indicates the pixel is tested and black indicates it is not. Each measurement examines half the pixels, and subsequent measurements narrow down the location of the high pixel. The first test
indicates which half the pixel is in, the second test indicates which quadrant, and so on.

This simple pooling design will guarantee an answer in $\log_2 N$. For example, a $128 \times 128$ ($N = 16384$) pixel image will guarantee an answer in 14 measurements.

Group testing can be used across domains and contexts. We find an interesting and informative example of group testing in a 1943 paper entitled "The Detection of Defective Members of Large Populations\(^2\) detailing the testing of army recruits for syphilis. The United States Public Health Service and the Selective Service System were tasked with finding a relatively small number of syphilitic men from a large population. The obvious approach is to test each man individually by first drawing a blood sample and then subjecting each sample to a laboratory analysis revealing the presence or not of the syphilitic antigen. A population of $N$ members requires $N$ chemical analyses.

The identifying test for syphilis is very sensitive and will return positive results even if the sample is greatly diluted. This fact, combined with the knowledge that the prevalence of syphilis is low, led to the idea of group testing. Blood samples drawn for each man individually were pooled into groups, and the pooled samples were then subjected to the test.

Say the pooled group size is five, for example. If none of the five samples contain the syphilitic antigen then the test on the group will be negative. If the test result is
positive, then the blood of at least one of the members of the group contains the syphilitic antigen and the individual samples must be retested. If the prevalence rate of the defected members is known, one can determine mathematically the best group size to minimize the number of tests.

If the prevalence rate of syphilis among a given group is 1%, the optimum group size is 11 members and the number of tests required to find the defected members is reduced by a factor of five from the population size. Even at a prevalence rate of 10%, groups of size 4 results in 40% fewer measurements needed. The lowest rate of syphilis at the time was found in white men from Wisconsin, with a prevalence of 0.05%. At this rate, group testing with a group size of 15 reduces the number of tests by a factor greater than 7.

Figure 2.2A shows the expected relative testing cost of different group sizes at different rates of prevalence. The relative testing cost is the ratio of the number of tests required by group testing to the number of tests required by individual testing. A lower relative testing cost is better. Also shown in Figure 2.2B-C, are the optimum group size and optimum relative testing cost, respectively, at different rates of prevalence. Group testing becomes more efficient as the relative number of men with syphilis to the total number of men becomes smaller. As prevalence rates go up the benefit of group testing goes down until it is no better than individual testing. In general, group testing will be more economical if the sought-after members of a population are sparse, and if testing a group is no more difficult than testing individuals.
If large groups and multiple rounds of pooling are allowed and practical, group testing is even more efficient. Say an initial large group of 100 pooled samples has a positive result; the group can be split into subgroups for a second round of pooled testing. This subgroup can then be pooled into smaller groups for the next round, and so on.

It is worth noting that the total population size, N, cancels out of all calculations and the optimum group size only depends on the prevalence rate.

The top panel (A) shows the expected relative testing cost of different group sizes at different rates of prevalence. The expected relative cost is the ratio of the number of tests required by group testing to the number of tests required by individual testing. The prevalence rates shown vary from 0.5% to 10%. Also shown are the calculated optimum group size (B) and the best relative testing cost (C) at the various rates of prevalence.
Nonadaptive Group Testing

The preceding examples demonstrate two types of group testing schemes. The case of detection of syphilis is an example of an adaptive procedure, where the next step in the procedure depends on the results of the prior step. The case of the sparse image is an example of a nonadaptive procedure, where the testing scheme does not depend on the results of any step, but is consistent and pre-determined. To show the power of nonadaptive group testing, we find another interesting and fun example in the famous 12-coin problem. The problem is as follows:

You have 12 coins of the same size. 11 of these coins are the same weight, one is different; you do not know whether this counterfeit coin is heavier or lighter than the others. You have a balance on which any group of coins can be weighed against any other group. How can you discover by means of three weighings which one is the counterfeit coin and whether it is heavier or lighter than the other coins?³

An adaptive procedure solution can be found by a complicated series of steps that start by weighing four coins against another four coins, and continues by adapting the next steps to the results of each previous weighing. You are free to figure this out on your own time, and I encourage you to do so.

The nonadaptive solution is elegant and simple to perform. The following is one possible measuring scheme. Start by labeling the coins 1 to 12. For the first weighing compare coins 1, 2, 3, and 4 on the left side of the balance against coins 5, 6, 7, and 8 on the right. The balance can lean one of three ways: to the left, to the right, or balanced. Record the result. Next, weigh 1, 4, 8, and 9 against 2, 3, 11, and 12 and record the result.
And finally, weigh 3, 7, 9, and 12 against 1, 2, 5, and 10 and record the result. Performing these three weighing will exactly reveal the counterfeit coin every time. In fact, this combination is only one of many possible combinations.

Figure 2.3 shows all the possible outcomes if the counterfeit coin was light. A plus sign indicates that the left side went up, and minus sign indicates that the left side went down, and 0 indicates that the scale was balanced. For all the heavy coin possibilities, all plusses and minuses would switch places. For example, if the three weighings returned plus-minus-zero, the distinct solution is coin 8 is heavy.

![Figure 2.3 Nonadaptive Results of the 12-Coin Problem](image)

The results of three weighings lead to distinct solutions to the 12-coin problem. Here are the results one gets if a coin is light. A plus sign indicates that the left side went up, a minus sign indicates that the left side went down, and a zero indicates that the scale was balanced. For a heavy counterfeit coin, all plusses and minuses switch places.

A comparison of the adaptive and nonadaptive approaches reveals the advantage of the nonadaptive approach. With adaptive group testing, each move depends on the results of the previous move and the complexity occurs during the data collection. With nonadaptive group testing, the same moves are made every time, regardless of the results of the previous moves and the complexity occurs behind the scenes, in this case to devise the pooling design before the data collection. The data collection is not interrupted in the nonadaptive procedure.
The nonadaptive solution fits into a clean mathematical framework, which we introduce here. We write a vector, \( x \) to represent the unknown image for which we will solve. In this case, the elements of the column vector represent the coins, 1 through 12. If the coin is light it will be represented as a +1, if it is heavy it will be –1, and all others will be zero. For example, the vector \( x \) for a heavy coin 8 is

\[
x = \begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
-1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\]

We write a row vector, \( \phi \), representing the weighing measurement. The element is +1 is the coin is on the left side of the scale, –1 if the coin is on the right side of the scale, and 0 if the coin is not a part of the measurement. The three measurements described above are represented as

\[
\Phi = \begin{pmatrix}
\phi_1 \\
\phi_2 \\
\phi_3
\end{pmatrix} = \begin{pmatrix}
+1 & +1 & +1 & +1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
+1 & -1 & -1 & +1 & 0 & 0 & 0 & 1 & +1 & 0 & -1 \\
-1 & -1 & +1 & 0 & -1 & 0 & +1 & 0 & +1 & -1 & 0
\end{pmatrix}
\]

The result of a measurement is the inner product of the sampling mask, meaning the row representing the coins measured, and the column vector representing the true image. Putting it all together, from the above measurements \( \Phi \) and truth \( x \) we obtain the weighings, \( W \),

\[
\Phi x = W = \begin{pmatrix}
+1 \\
-1 \\
0
\end{pmatrix}
\]
Because $\Phi$ and $W$ are known, and we know that only one element of $x$ is nonzero, $x$ can be solved.

**Sparsity**

It is important to note the situations when group testing is helpful. Group testing allows for the location of a single high pixel in an image in $\log_2 N$ measurements. It allows for the identification of syphilitic antigen with up to 7 times fewer measurements. It allows us to find the one counterfeit coin out of 12 with only three measurements. In general terms, we determined the information in a signal vector with less data collection than expected. The key similarity between these examples is that we have *a priori* knowledge that the signal vector is sparse. Technically, sparsity of a vector or matrix is a measure of how many zero elements it contains. More broadly defined, sparsity is the measure of

![Figure 2.4 Sparse Image in the Fourier Domain](image)

An image of N total pixels (A) that does not look sparse, with low bands surrounded by bands of high intensity. Half the pixels contain intensity. The image domain (B) is represented by taking each row of (A) and adding it to the end of the previous row. The square wave seen in (B) is Fourier transformed to represent the image in the Fourier domain and (C) plots the coefficients in the sine function basis set. The image is sparse in the Fourier domain.
information contained in a vector or matrix. So a sparse vector is one in which only a few elements of the signal vector are non-zero and most elements are small or zero. In the example cases we have seen, the vector is sparse in the direct representation described. However, in general an image or signal will be sparse in some transform domain besides the pixel domain. For example, the image in Figure 2.4A is not sparse in the pixel domain. Of the N total pixels, half contain high intensity and half contain none. Figure 2.4B represents the intensities in the direct pixel (image) domain as a function of pixel by counting across each row, and adding a row to the previous one. The result is that the image appears as a periodic square wave with a wavelength equal to half the length of one side, $\sqrt{N}/2$. We can perform an integral transform on this vector. The Fourier transformation converts the information into a series of sine functions of increasing frequencies and diminishing amplitudes. The coefficients are plotted in Figure 2.4C and we see that the amplitude of the coefficients diminishes and most of the information is contained in a small number of elements. The image is sparse in the Fourier domain.

Finally, as an example using real 2D IR data collected in our lab, consider the signal in Figure 2.5. This is a slice taken from a two-dimensional data array, showing the
signal as it was collected, as a function of a time delay between two pulses. The data is not sparse because there are very few elements with value zero and all the points are needed to represent the information. After a Fourier transform, Figure 2.5B is acquired, representing the signal in the frequency domain, where a single, large peak rises above the noise, which is mostly zero. This vector is sparse in the frequency domain.

The key to sparsity is that if a signal is not sparse in the measurement domain, it will likely be sparse in some other basis set. The examples presented here use common transformations with physical interpretation, however the transformations may be purely mathematical and need not have physical meaning. Once transformed, the signal can then be store and passed around in the sparse basis set, reducing the amount of data required to contain the information.

Data Compression

In the middle of the second decade of the 21st century we are facing a data deluge. More digital photos are being taken and shared on the internet than ever before. More files are being saved to the cloud and streamed around the world than ever before. More websites are being loaded and visited than ever before. A critical technology that makes all this possible is data compression. Data and information are not the same, and the aim of data compression is to have the same amount of information contained in less data.

Traditional data compression follows five basic steps. The first step is to collect the full $N$ element signal, $x$, and next is to transform the signal into some sparse domain that can be described by
\[ x = \sum_{n=1}^{\infty} \theta_n \psi_n \]

where \( \theta_n \) are coefficients of the \( N \times 1 \) basis vectors \( \psi_n \). Next, one computes the complete set of coefficients, where \( \theta_n = \langle \psi_n | x \rangle \). Next, the coefficients above a determined threshold are retained and all others below the threshold are discarded. Only the values and locations of these large coefficients are enough to encode, store and transmit the information. The final step is to inverse transform back to the desired domain for viewing. The signal is said to be sparse in the transform domain if \( K \ll N \), where \( K \) is the number of coefficients kept. This process underlies the standard JPEG image compression that allows you to store thousands of photos on your computer and post them online.

Consider Figure 2.4 and Figure 2.5 again to see the power of compression. In the case that the image in Figure 2.4A is 128×128 pixels, 16,384 pixels are required to

![Figure 2.6 Square Wave Reconstructed from Compressed Data](image)

Image in Figure 2.4 (blue) reconstructed from 100 (red) and 10 (yellow) coefficients in the Fourier domain. The x-axis is one wavelength of the periodic square wave.
create the image. Figure 2.4C shows the equivalent information in a different representation. Using only the coefficients from the Fourier domain, one can perform an inverse Fourier transform and reconstruct the original image. Selecting only the largest 100 coefficients nearly perfectly reconstructs the image, as seen by the red signal in Figure 2.6. Using only the first ten coefficients, the inverse Fourier transform returns the yellow signal in Figure 2.6. The reconstruction is not perfect but the threshold for compression can be selected as needed. In this case, the information in the image can be stored and transmitted as only ten numbers, or 0.06\% of the total number of pixels.

Figure 2.5B shows the frequency domain slice of a 2D IR spectrum. Though the full slice is 512 points in length, the peak only spans about 20 data points and be stored simply as these 20 points, essentially zooming in on the peak and discarding the rest. Further, the peak shape is approximated by a Gaussian profile and in that domain, the same information can be given in only three numbers: the amplitude, center frequency, and full-width half max. In this case, the information in the signal can be stored and transmitted by 0.6\% of the total number of data points.

Real images are not as simple as the examples presented here, though a sparsifying transform can still be found that maps the signal vector of image data to a sparse vector. We know that complex images can still be compressed because we compress them billions of times daily when dealing with digital photos and videos. Real images are known to be sparse in the discrete cosine transform and wavelet transform, which are used by JPEG and JPEG-2000, respectively. JPEG-2000 image compression works by breaking an image into smaller chunks with the intensity and color of one pixel similar to the intensity and color of the pixels around it. The algorithm then transforms
these chunks into the wavelet domain and compresses, or breaks them into smaller chunks and does so. The result is that a digital image can be compressed by orders of magnitude from its original file size.

Compressive Sampling

We know from data compression that any signal can be represented in a sparse domain, and we know from group testing that with clever pooling designs we can acquire a sparse signal with less data collection than expected. Combining these two ideas, the mathematician and statistician Emanuel Candés, a pioneer of compressive sampling, asked the provocative question in 2006:5

“Because most signals are compressible, why spend so much effort acquiring all the data when we know most of it will be discarded? Wouldn’t it be possible to acquire the data in already compressed form so that one does not need to throw away anything?”

We answer in the affirmative, and data collected this way is called compressive sampling.6-11 As we’ve seen in this chapter, if a signal is sparse and compressible, we ought to be able to devise a nonadaptive data collection method to acquire the signal in fewer measurements than traditionally expected. However, this results in a strictly underdetermined problem, and the signal cannot be exactly reconstructed. Using the assumption of sparsity, one can devise an algorithm to determine the simplest, sparsest solution that matches the data collected.

Lustig et al.12 present an example of an intuitive procedure to recover a signal from random undersampling. This is a simplified case to demonstrate the feasibility of compressive sampling. Consider the sparse frequency signal of Figure 2.7A, 256 samples
long, undersampled in time by a factor of eight (Figure 2.7B). Straightforward reconstruction from equispaced undersampling results in coherent aliasing (Figure 2.7C) and it is not possible to determine the original signal from the aliased replicas.

Reconstruction from pseudorandom undersampling results in incoherent aliasing caused by energy leakage from elements in the original signal (Figure 2.7D). A plausible recovery procedure is shown in Figure 2.7E-H. The strong components above a threshold are detected (e) and recovered (f). The interference of these components is calculated (g) and subtracted (h) from the initial reconstruction, lowering the total interference allowing the recovery of weaker components. Figure modified from Reference 12.

An example of a real reconstruction algorithm is based on minimizing the total variation of a signal or image. One conception of sparsity is that a signal is smooth, or that the intensity at one pixel is like the intensities of the pixels around it. Smoothness can be quantified by the total variation, which is the integral of the absolute gradient of
the signal. One can fill in the spaces around known pixels while keeping the total variation minimized to within a parameter (often called $\lambda$) that enforces the level of sparsity in the solution. Total variation regularization is often used in image processing and denoising. The yellow signal in Figure 2.6 shows exact reconstruction of a square wave from only ten elements after compression of the full sample. The reconstruction is improved by including total variation denoising, which is effective at smoothing away noise detail while preserving edge detail.

To demonstrate the effectiveness of compressive sampling and total variation image reconstruction (TV), we present a standard phantom image created in MATLAB to test reconstruction algorithms. The head phantom in Figure 2.8 is a 64×64 = 4096 pixel image in gray scale intensity. Designed to simulate medical images it contains one
large ellipse to represent a brain and several smaller ellipses to represent features of the brain.

We take a series of completely random linear combinations of pixels. In other words, we group test the image by measuring the sum of intensities from a random grouping of half of the pixels. As we saw in the Lustig et al. example and will see again, random undersampling leads to the highest quality reconstruction. Figure 2.9A shows one random sampling pattern, where half of the pixels are measured (white) and half of the pixels are not measured (black). We call the measured and non-measured pixels ON and OFF, respectively. The pixels turned ON are used for reconstruction and the pixels turned OFF are discarded and not used in the reconstruction. Thus, the image is compressively sampled.

The measurement of a single value is the sum of the intensities of the pixels turned ON, which we have seen mathematically as the multiplication of the sampling pattern vector and the image vector. Figure 2.9B shows the result of 1024 measurements of different sampling patterns. A series of measurements have only small variation in total intensity from one measurement to the next, and looks rather like noise.

The TV algorithm takes as inputs the series of sampling mask patterns and the corresponding measurements. The algorithm finds the solution to the image vector that reproduces the measured data with the greatest fidelity while enforcing sparsity on the image by minimizing the sum of the total variation of the solution. Further details of the TV algorithm are provided in the next chapter.
To fully sample an image, the number of measurements is equal to the number of pixels and the problem is a large linear algebra problem that is fully determined. Any number of measurements less than the number of pixels is compressive sampling and a reconstruction algorithm must be used to solve the image. The number of measurements relative to the fully sampled case was referred to as testing cost earlier in this chapter, and we also refer to it as compression factor or level of compression. Figure 2.10 shows the phantom reconstructed from different levels of compression. At $8 \times$ compression only 512 different sampling patterns are measured and the reconstructed image is of poor quality, however the main ellipse is clearly present and the two black features of the phantom are roughly observed. As a coarse scan, this level of compression may be sufficient to determine the next course of action. For example, one could repeat measurements with a smaller pixel size on a particular region of interest. Reconstruction from $4 \times$ compression (1024 measurements) the white outline of the phantom is sharper.
and the shapes of the two features are more defined, however the image quality is modest. At $2\times$ compression, 2048 measurements, all features of the original phantom image are reconstructed with high fidelity.

Figure 2.10 Compressive Sampling Reconstructed Images of Phantom

The phantom reconstructed by 2X compression (2048 measurements), 4X (1024) and 8X (512).

We observe that the more measurements one collects the more faithful the reconstruction is to the original. More measurements mean a more complex reconstruction calculation, but the nonadaptive nature of group testing and compressive sampling means that data can be collected quickly and the complicated reconstruction happens by computer algorithm after the data collection.

More measurements also take more laboratory time to acquire, and there is a tradeoff between acquisition time and reconstruction quality. Still an exciting advantage of compressive sampling is that one can accelerate the acquisition of signals and images. This can be especially useful in medical imaging, such as MRI or CT scans, where a patient may be exposed to harmful radiation or may be scared and kept still in an uncomfortable chamber during the extent of the data collection.
In the next chapter, we will focus on the implementation of compressive sampling imaging in our laboratory with an apparatus called the single-pixel camera.
3. COMPRESSIVE SAMPLING IMAGING

Motivation and Introduction

We all know the saying that a picture is worth a thousand words. An image has the power to enlighten a person, to transform a society. The ubiquity of digital cameras and digital images today gives us access to the depths of the oceans and to the peak of the highest mountains, to the edge of the solar system and to the cells in our bodies. Humans are visual creatures. Capturing an image captures the mind like almost nothing else can.

Imaging is the art and science of converting a spatially varying intensity into a picture. A picture element, called a pixel, holds some measured value of intensity to represent the light at some real point in space. This spatially dependent intensity is converted to a color-scale to produce an image.

The development of microscopy revolutionized the way we interact with the world. By allowing smaller pixel size, microscopy reveals smaller parts of the world; parts that we never even knew existed. As technology continues to advance, our instruments extend our senses beyond the microscopic. With detectors that go past the visible range, we can now 'see' the world illuminated by light we cannot see with our eyes.

For example, thermal imaging uses infrared-sensitive detectors to detect infrared light. All bodies give off infrared radiation, which is longer wavelength and lower energy
than visible. Infrared cameras detect infrared radiation as a function of space and produce an image based on intensity of infrared. Thus, we have night vision and we can see behind walls.

Infrared is also powerful in that it excites unseen molecular bonds to vibrate but not break. Bonds vibrate at specific frequencies, giving off radiation at corresponding energies. Detecting the intensity of infrared radiation as a function of frequency is called spectroscopy, and the resulting data is called a spectrum. From this we can determine the chemical bonds that are present and thus the chemical structure of a molecule. By detecting radiation as a function of space (imaging) and frequency (spectroscopy) the inherent chemical composition of a sample provides the contrast to visualize its structure as an image. Spectroscopic imaging collects a spectrum at each pixel and uses some spectral feature, such as the amplitude of a certain peak, to reduce the data to a single value for each pixel, thus an image.

The earliest infrared spectroscopic imaging experiments were performed in the 1950's. Then, as now, there was great interest in imaging human cells and tissues with low-energy, noninvasive infrared light. By the 1980's, chemical imaging was being applied to living systems. Since then, combining the advantages of optical microscopy with the selectivity of vibrational spectroscopy, chemical imaging has been applied to thousands of applications for many decades.

Conventional optical imaging produces a two-dimensional data set, represented as $I = f(x, y)$, with intensity as a function of spatial dimensions, $x$ and $y$. Spectroscopic imaging produces a three-dimensional datacube (Figure 3.1) represented as $I = f(x, y, \omega)$, which also includes the frequency axis, $\omega$. To generate an image from the
datacube, select an xy-plane at the frequency of the peak of interest. The amplitude of the peak is proportional to the presence of the bond that corresponds to that peak. Thus, mapping this plane onto an image will show where a particular molecule is and is not. One can produce different images from the same data by selecting different spectral features.

Infrared spectroscopic imaging is a powerful tool in pathology for the identification of diseased cells.\textsuperscript{18-23} Because the inherent chemical composition provides the imaging contrast, typical stains or dyes of optical imaging, which suffer from nonspecific binding, are not needed. However, a tissue is composed of many different molecules with similar bond structures making it difficult to separate overlapping spectral features. Thus, the main weakness of infrared spectroscopic imaging is that it is difficult
to separate the various molecular contributions to the spectrum, which can lead to subjective interpretation and diagnostic uncertainty.

Two-dimensional infrared (2D IR) spectroscopy is an extension of infrared spectroscopy that reveals more information and details of the chemical structure of a sample. 2D IR uses multiple laser pulses to interact with a sample at precise time delays. The first set of pulses act like one-dimensional spectroscopy in that they provide information about system. The next pulses provide information about how the system evolved in the time between. Peaks that appear along the diagonal of the 2-D spectrum probe the environment of chemical bonds, as infrared spectroscopy does. Cross-peaks that appear off the diagonal reveal deeper information about vibrational couplings, time dependence, environmental dynamics and structural kinetics. As infrared spectroscopy is like a spotlight that reveals chemical structure, 2D IR is like a strobe light that reveals structural changes in time.

2D IR is effective in separating overlapping spectral components of complex samples by spreading the features across the second dimension. Because there is added structural sensitivity in a 2D IR spectrum, it provides new modalities of contrast for imaging. The development of 2D IR microscopy will present important new applications in biomedical imaging.

A 2D IR spectrum is a function of two frequencies at a given time delay, \( T \). Therefore, 2D IR imaging produces a five-dimensional hypercube of data represented as \( I = f(x, y, \omega_1, \omega_3, T) \). In this chapter we will develop the preliminary apparatus and results and discuss the proposal to construct a 2D IR microscope.
Approaches to Imaging

There are two basic approaches to collecting an image. The first is to collect the intensity of every pixel all at once with an array. This is what a digital camera does and is what most people think of when they think of taking a picture. One acquires an image all at once. The second approach is to collect the intensity on only one pixel at a time. One moves the sample across the desired field of view and collects the individual measurements. This is called raster scanning and only requires a single-element detector. One constructs an image by compiling the individual measurements.

Because only one pixel is used instead of an array, the number of measurements will be directly proportional to the number of pixels. Thus, a raster-scanned image will take longer to acquire by a factor of $N$, where $N$ is the number of pixels. Though it takes longer, there are some advantages. The user has control over the field of view and the number of pixels. Furthermore, a single-element detector has financial advantages. In the infrared specifically, they are much cheaper and more sensitive than arrays. Focal plane array (FPA) detectors in the infrared come with a small number of pixels and are currently prohibitively expensive at around $100,000 USD.

Figure 3.2 Simple 2x2 Image

![Simple 2x2 Image](image)

Simple 2x2 pixel image with the intensity of each pixel labeled within. We will use this simple image to the group testing approach to imaging.
A less common approach is group testing, which is a combination of imaging and raster scanning. Extending our discussion of group testing in the introductory chapter, group testing imaging acquires the intensity of multiple pixels onto a single-element detector. We will walk through an example of a group testing approach with a simple 2×2 pixel image. Practical methods of isolating and selecting individual pixels will be discussed in the following section with the introduction to the digital micromirror device (DMD).

The basic approach is to collect the sum of intensities of different sets of pixels and from these measurements calculate the intensity at each pixel. We represent the problem mathematically. A column vector, \( x \), is written to represent the image, which is the collection of intensities at each pixel,

\[
\begin{pmatrix}
I_A \\
I_B \\
I_C \\
I_D
\end{pmatrix}
\]

A row vector, \( \phi \), represents the measurement, called a sampling mask,

\[
\phi_i = (\phi_A \quad \phi_B \quad \phi_C \quad \phi_D)
\]

where \( \phi \) elements are equal to one if the pixel is measured, and zero otherwise.

One way to group test this image is to use a single-pixel element detector to make the following four measurements:

\[
M_1 = I_A + I_B \\
M_2 = I_A + I_C \\
M_3 = I_A + I_D \\
M_4 = I_C + I_D
\]
Each measurement results in one value, acquired from the detector, that is the sum of the intensities of the pixels selected. The measurement is the inner product of the sampling mask and the true unknown image. For example, the first measurement is

\[ M_1 = \langle \phi_1, x \rangle = (1 \ 1 \ 0 \ 0) \begin{pmatrix} I_A \\ I_B \\ I_C \\ I_D \end{pmatrix} = I_A + I_B \]

The full problem representing all sampling masks and measurements will form one linear algebra matrix problem. Using the measurements from this example, we get

\[ \Phi x = M \]

\[
\begin{pmatrix}
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
I_A \\
I_B \\
I_C \\
I_D
\end{pmatrix}
= 
\begin{pmatrix}
M_1 \\
M_2 \\
M_3 \\
M_4
\end{pmatrix}
= 
\begin{pmatrix}
I_A + I_B \\
I_A + I_C \\
I_A + I_D \\
I_C + I_D
\end{pmatrix}
\]

Because there are four unknowns and four equations, this is a perfectly determined problem and solving for \( x \) is straightforward.

\[ \Phi x = M \]

\[ \Phi^{-1} \Phi x = \Phi^{-1} M \]

\[ x = \Phi^{-1} M \]

The group testing approach detailed above can be extended to higher pixel densities with \( N \) pixels, where generally each measurement will be a different set of half the pixels investigated. Given the condition that the image is sparse and compressible in some domain, it is possible to reconstruct an image from a number of measurements much less than the number of pixels, though this is strictly an underdetermined problem.
The Digital Micromirror Device

We group test by isolating and selecting individual pixels of an image using a Digital Micromirror Device (DMD). A DMD is a reflecting spatial light modulator, meaning it can modulate the intensity of light as a function of space. Much like an overhead projector is a spatial light modulator in transmission mode, a DMD is a spatial light modulator in reflection mode. The DMD is composed of hundreds of thousands of 13 μm mirrors (1024 x 768) that can be independently rotated to control the direction of reflection of incident light at ±12° off the surface (Figure 3.3).

We place the DMD at the focal plane of the imaging device, with each mirror corresponding to a pixel. The DMD mirrors are individually controlled by computer and programmable. We call one direction of reflection ON and focus the reflection into a single beam that is directed toward and focused onto a single-pixel detector, which

Figure 3.3 Digital Micromirror Device

Example of a sampling pattern on the surface of a DMD
measures the sum of intensities of all mirrors in the ON direction. The opposite direction, OFF, directs light away from detector for other purposes or to a beam dump. The DMD encodes spatial information into the reflected beam.

Significant advantages of a DMD are that the field of view and the effective pixel density are completely scalable by programming and without any physical experimental changes. There is always going to be a balance between spatial resolution, field of view and the time to measure and reconstruct an image. The time to collect an image is proportional to the number of pixels, and the cost of reconstruction goes up with higher pixel densities. If a user desires higher spatial resolution, the cost will be field of view or image acquisition time. If a user desires a larger field of view, the cost will be spatial resolution or image acquisition time. A DMD is versatile and customizable to the needs of the experiment.

The size of each effective pixel can be changed to control the field of view of imaging by binning groups of mirrors together to act effectively as one pixel. For example, consider different ways that an image with pixel density 128×128 can occupy the 768×1024 mirrors on the surface of the DMD. With each mirror acting independently as a pixel with size 13 \( \mu m \), only the equivalent number of mirrors on the DMD are used. By binning 2×2 sets of mirrors to move together and act as one effective pixel with a pixel size of 26 \( \mu m \), four times more mirrors are used on the DMD and an area four times larger can be imaged without changing the number of pixels or anything else about the apparatus. Figure 3.4A shows the DMD usable surface of two different 128×128 pixel images with different effective pixel sizes corresponding to different fields of view. In combination with pixel size, for a given field of view the user controls the effective pixel
density. Figure 3.4B shows three images that have the same field of view but differing pixel size, pixel density, and spatial resolutions.

In addition to controlling the pixel size and number of pixels, the position of the sampling pattern on the DMD is programmable. In most cases, the imaging surface of the DMD will not occupy the entire surface. By moving the sampling pattern around the plane of the DMD, different locations within the sample can be imaged without any moving parts.

One can imagine the following scheme to acquire a high spatial resolution image in a small field of view. Start with a low pixel density and large pixel size, covering most of the DMD surface, to quickly acquire a rough image. Proceed by zooming in on the desired image and optimizing the effective pixel size and pixel density.

The Visible Single-Pixel Camera

To illustrate the concept of using the DMD to compressively sample an image, we demonstrate the single-pixel camera with visible light. Based on the work of the Baraniuk
group from Rice University\textsuperscript{26}, we built a homemade single-pixel camera (SPC). The SPC is straightforward to build. We focus light from a 780 nm wavelength collimated laser diode (Thorlabs CPS192) into a sample and imaged onto the DMD, which encodes the spatial information into a beam that is directed to a single-element detector (Figure 3.5). We use LabVIEW to control the DMD and data acquisition. Figure 3.6 shows a screenshot of the data collection code, including user control inputs for scan parameters (number of scans to average, the number of masks to measure, the number of samples to run per mask) and DMD parameters (the effective pixel density, the effective pixel size, the location of the mask pattern on the DMD). The current scan is updated in real time and the average is updated after each scan. To separate the signal from noise and other light around the detector, we chop the laser beam and the signal is amplified in a lock-in amplifier (Stanford Research SR810) before being sent to the data acquisition card (National Instruments DAQ) and the computer.
A proof of concept experiment does not require a sophisticated sample. The first image acquired using the SPC was of three wrenches of various sizes attached to a lens mount. A conventional image of the target object is shown in Figure 3.7A. With spatial information encoded into a single beam with the accumulated intensity of the pixels aimed at the detector, we take measurements on a single element detector and to reconstruct an image. We show the reconstructed images at various levels of compression at two different pixel densities. The field of view, that is, the total number of mirrors used on the DMD, is constant for the reconstructed images shown. This is clear due to the images all covering the same area of the sample. Figure 3.7B and 3.7C are collected at a pixel density of $64 \times 64$ pixels where each pixel is a $2 \times 2$ group of mirrors. Figure 3.7D is
collected with a pixel density of 128×128 where each pixel is one mirror. Figure 3.7C and 3.7D are approximately the same compression factor.

Figure 3.7B is the reconstructed image from acquiring 1024 measurements with random sampling patterns, a 4× level of compression, or 25% of the number of pixels. The thick line in the middle is clearly present, though not in focus, and the smaller lines on either side are faint. Smaller pixel density and fewer measurements makes for a good first run to verify the position of the sample quickly.

As we make more measurements the reconstructed image improves. Figure 3.7C is acquired with the same settings and scan parameters as Figure 3.7B but with 2000 random measurements (> 2× compression, or < 50% pixels). The edges of the lines are more pronounced and the contrast of line to background improves. The data acquired to reconstruct Figure 3.7B is a subset of the data for Figure 3.7C. Compressive sampling data collection is progressive; if the quality of a reconstructed image is not adequate, one can acquire more scans with different sampling masks and compile the data to enhance the image.
Figure 3.7 D

We reconstruct the image in Figure 3.7D with approximately the same level of compression as Figure 3.7C, however the pixel size is smaller and the number of pixels is larger. Thus, resolution of the image is improved due to the smaller pixel size and we observe more defined line edges and greater contrast between image and background. The tradeoff, however, is that the data collection time is directly proportional to the number of measurements.

There can be unpredictable long-term slow fluctuations in the intensity of the source, which hurts the reconstruction. It is common to filter out high frequency components of data, as that is usually noise on top of the signal. In our case, the high frequency component is the signal data and we need to filter out the low frequency background. We built a simple filtering scheme in MATLAB into the reconstruction code to convert the data into a spectrum, filter out all frequencies below a threshold, and convert back into time domain before the compressive sampling reconstruction. Figure
3.8 shows an example of a raw data set (A) and the reconstructed image (B) the same data filtered (C), along with the corresponding reconstruction of the image from the filtered data (D).

The Single-Pixel Microscope

The SPC is straightforwardly converted to a microscope with the addition of a microscope finite conjugate objective (Nikon Achromatic 40X, NA=0.65). The objective is placed in space between the sample and the DMD. The sample is on a three-dimensional micrometer stage and we position it to be the working distance from the

Figure 3.9 Schematic SPC Microscope Tabletop

![Schematic SPC Microscope Tabletop](image)

Schematic illustration of the SPC microscope tabletop setup.
front of the objective (0.6 mm) and the image is magnified a finite distance away (160 mm), where we place the DMD. The exact distances are necessary to obtain a focused image on the DMD. The reflection off the DMD is captured on a 100 mm focal length curved mirror and focused into the single-element detector. Figure 3.9 provides a schematic diagram of the tabletop setup.

We use the USAF Resolution Test Target as our sample, a standard test pattern used to determine the resolution of imaging apparatus. Consisting of sets of lines of precise varying sizes, resolution is determined by the smallest set of lines that can be discerned. Figure 3.10 shows a camera image of the smallest group of lines on our test target.

The sample, objective, and DMD are on micrometer stages. The correct distances must be precise and can only be established by collecting compressively sampled data at
various positions and reconstructing images. By systematically adjusting the positions of the relevant pieces the required positions are set.

The reflection off the DMD in the OFF direction contains information and there are ways in which that information can be used. The OFF reflection includes the negative image from the negative sampling pattern as well as all the unused mirrors surrounding the pattern. Once the focusing positions of the sample, objective, and DMD have been established, a CCD camera is set up to capture the reflection in the OFF direction (Figure 3.10). The dark square in the middle of the Figure 3.10 shows the area of the DMD in use for compressive sampling imaging and the light around it shows the full beam. Since the micromirrors surrounding the sampling pattern are all turned in the same direction, they collectively act as a mirror we observe a clear image of a much larger section of the sample. Now the sample can be removed, replaced and focused on the camera and it will be in focus for the compressive sampling apparatus.

There are two methods of changing the section of the sample to be imaged. One is to keep the sample fixed and move the sampling pattern around the DMD. However, this will slightly change where the reflected beam comes off the DMD and may affect the pointing of the focused beam into the detector. The second and better method is to keep the DMD settings fixed and reposition the sample around a two-dimensional plane with the stages.

Figure 3.11 shows images acquired with the single-pixel microscope at different levels of compression. We image the four smallest sets of lines on the USAF test target (Group 7 Elements 5 and 6), which are 2.46 µm (upper left sets) and 2.19 µm (lower right sets) wide and separated by the same distance. We collect the images at 128×
128 = 16384 pixel density with a pixel size of 13 μm (1 mirror). At the 40× magnification of the objective the smallest lines will appear as 88 μm lines in the image at the DMD and will span approximately 7 pixels. Figure 3.11A was obtained with 8192 measurements, a compression factor of 2×. Figure 3.11B and 3.11C were obtained from 4096 and 1638 measurements, compression factors of 4× and 10×, respectively. The axes are labeled with pixel number. Acquiring a number of measurements equal to 10× less than the number of pixels leads to an image with clearly visible lines, though the intensity if not evenly reconstructed across the image and the quality remains quite modest. At 4× compression, all 12 lines are separable, although they are blurry. Figure 3.11A shows a clear image of the smallest lines available on the test target. The lines are reconstructed with even intensity and the background is uniform. Quantitative analysis of an image is important to support any claim about results. Figure 3.11D shows the line
profiles of the images above each plot. The line profile starts at the upper left corner and ends at the lower right corner of each image and crosses the six lines.

This figure demonstrates a wide-field image collected using visible light and compressive sampling with resolution down to a few microns.

The Infrared Single-Pixel Camera

To realize two-dimensional infrared spectroscopic imaging, one must to be able to compressively sample in the infrared. The differences between the visible single-pixel camera and the infrared single-pixel camera are not trivial. Moving along the light path, there are four major differences to address: the light source, the objective, the DMD window, and the detector.

We generate the infrared light from an amplified Ti:Sapphire laser system (Spitfire Pro) that outputs pulses at a 1 kHz repetition rate that are tuned to the desired mid-IR frequency using a home-built optical parametric amplification and difference frequency generation setup. We use mid-IR pulses ~120 fs in duration and centered at 2050 cm^-1 at a repetition rate of 1 kHz.

The refractive objective used in visible microscopes is not an infrared compatible optic. Reflective objectives are wavelength independent and are useful across all ranges of frequencies, however Serrano et. al show that refractive objectives are preferable for high resolution imaging with coherent radiation. A reflective objective, such as a Schwarzchild objective, obscures the light near the center of the beam which effectively puts a high-pass filter on the resulting image. We use an aspherical ZnSe lens (Edmund Optics) with a 12.5 mm focal length and a 0.7 numerical aperture.
The standard DMD has a glass protective window over the mirrors, which is opaque to infrared radiation. To operate in the infrared, the glass is replaced with a CaF$_2$ window. Due to the small size of the micromirrors, this is nontrivial.

We use a mercury-cadmium-telluride (MCT) photodetector for infrared radiation detection. Even with a modest pixel density (32×32), MCT array detectors can be prohibitively expensive. The fact that the SPC apparatus uses a single-element detector becomes a significant advantage in the infrared.

When the pulse shaper is operated with a uniform waveform of high signal, the entire beam is diffracted and the pulse shaper is essentially a zero-dispersion stretcher, meaning the output beam is the same as the input beam. There is inherent inefficiency
due to the acousto-optic modulator, but it is effective due to the fact that the system is already in place. We redirect the output of the pulse shaper into the SPC apparatus. We focus the beam to a spot just past the sample, and place the ZnSe ashpere lens after the focal point. We place the DMD where the diverging beam comes into a focal plane and we apply a random sampling pattern to the DMD to encode the spatial information of focused image into the reflecting beam. The image in Figure 3.12 is imaged with 5 \( \mu m \) laser pulses and shows the IR beam profile with a hole in the middle of it caused by a 40 \( \mu m \) polystyrene bead. This is the first image of a singe-pixel microscope in the mid-IR of which we are aware.

Spectroscopic compressive sampling imaging in the infrared can be easily realized from this setup by using the pulse shaper to generate two pulses and collect data as a function of the time delay between the two pulses.

2D IR Compressive Sampling Microscopy

To realize a 2D IR spectroscopic microscope we need to measure a signal as a function of two frequencies and two spatial positions to acquire all the spectral and image data. We need to collect the equivalency of a complete 2D IR spectrum at each spatial position. However, there is a fundamental conflict between how 2D IR is currently collected and how images are collected with a microscope. A conventional 2D IR spectrometer (Figure 3.13A) spreads the signal off a grating into space as a function of frequency in the spectrometer for detection with an array detector. A convention wide-field microscope spreads the signal out in space as a function of space for detection with a focal plane array detector. In combining the spectral properties of 2D IR with the spatial
properties of imaging we cannot simultaneously map both frequency and space onto the same space for array detection.

Two approaches to solving this problem have been investigated by other groups. Tokmakoff and coworkers\textsuperscript{28} realized a rudimentary form of 2D IR imaging by focusing the beam to a spot of approximately 5 $\mu m$ diameter and raster scan the position of the sample across the beam. They measure the 2D IR spectrum at each position and put these points together to form an image. This is time consuming (over 6 hours for a 16$\times$16 pixel image) and will not be practical over a wide field of view with necessary spatial resolution. Zanni and coworkers\textsuperscript{29} realized spatially resolved 2D IR via wide-field microscopy using a focal plane array detector and collecting the signal fully in the time domain. Focal plane array detectors in the infrared are extremely expensive ($>$\$100,000), have large pixel sizes on the order of 40 $\mu m$, and suffer from slow readout speeds and pixel-to-pixel variation.

Compressive sampling microscopy is the solution to this challenge. Incorporating a focusing lens and aspheric objective around the sample, we will image the sample onto the DMD as we do in the single-pixel apparatus. The light reflected form the mirror will have the image information encoded and will be propagated down the table into our spectrometer and we will measure the 2D IR spectrum just as we do now (Figure 3.13B). We will collect a 2D IR spectrum for each mirror pattern, scan DMD sampling patterns, and reconstruct 2D IR spectra at each pixel.
Our preliminary results suggest that we can use established compressive sampling methods to simultaneously collect the spectral and image information necessary to reconstruct an image from 2D IR spectroscopy. Incorporating an aspheric lens as an objective into our current experimental setup and using a digital micromirror device to encode spatial information, we can construct a first-of-its-kind 2D IR spectroscopic microscope.

Not only is the fundamental technical challenge solved by compressive sampling, we also reduce the massive amounts of data required by 2D IR spectroscopic imaging. Recall that 2D IR imaging produces a five-dimensional hypercube of data represented as $I = f(x, y, \omega_1, \omega_3, T)$ and with compressive sampling we do not need to acquire every spatial point directly. As we’ve seen, compressive sampling is not exclusive to imaging.
In addition to the undersampling of the spatial dimension, we can also undersample one of the spectral dimensions, thus reducing the number of values we have to collect to fill in the hypercube of data by 10× or more.

Without considering imaging, just collecting the data for a 2D IR experiment is data intensive and may take days for particular samples of interest to our group. In the following chapters, we discuss the details of how we collect 2D IR data and how we use compressively sampling to accelerate data acquisition.
4. COMPRESSIVELY SAMPLED 2D IR THAT PRESERVES LINESHAPE INFORMATION

Motivation and Introduction

Two-dimensional infrared (2D IR) spectroscopy is a powerful tool to investigate molecular structures and dynamics on femtosecond to picosecond time scales and is applied to diverse systems. Current technologies allow for the acquisition of a single 2D IR spectrum in a few tens of milliseconds using a pulse shaper and an array detector, but demanding applications require spectra for many waiting times and involve considerable signal averaging resulting in data acquisition times that can be many days or weeks of laboratory measurement time. Using compressive sampling, we show that we can reduce the time for collection of a 2D IR data set in a particularly demanding application from 8 days to 2 days, a factor of $4 \times$, without changing the apparatus, and while accurately reproducing the lineshape information that is most relevant to this application. This result is a potent example of the potential of compressive sampling to enable challenging new applications of 2D IR.

Long data acquisition times are a longstanding problem in many experimental methods and can limit the number of studies that are practicable, especially those involving time evolving or unstable samples. Over the last decade, compressive sampling
has reduced data acquisition time in diverse fields such as geophysics,\textsuperscript{31} medical imaging,\textsuperscript{11} computational biology,\textsuperscript{32} and astronomy.\textsuperscript{33} Compressive sampling requires only a fraction of the traditional number of measurements while yielding much of the same information as the fully-sampled data. Here we introduce an implementation of compressive sampling to reduce the data acquisition time of two-dimensional infrared (2D IR) spectroscopy without distorting spectral peak lineshapes.

Natural signals often have an underlying structure that causes the signal or its coefficients in an appropriate transform domain (e.g. Fourier coefficients, finite differences) to be sparse, meaning that most coefficients have small values and there are few large coefficients. In traditional data compression, the fully sampled signal is transformed into a fixed basis and the significant coefficients are stored or transmitted. The compressed data can then be decompressed for viewing. This idea underlies methods such as JPEG compression of images. A massive data acquisition followed by the removal of most of the data, however, is inefficient. Compressive sampling aims to bypass the first step and to directly acquire the compressed signal with no \textit{a priori} knowledge of the signal being measured.

2D IR spectroscopy is a multidimensional spectroscopic method used in the investigation of molecular structure and dynamics on femtosecond to picosecond time scales, and has been applied to diverse systems from dilute solutions to solids to membranes.\textsuperscript{34} There are different approaches to building a 2D IR spectrometer, and different experimental designs can lead to big differences in data acquisition times.\textsuperscript{35} Current technologies allow for the acquisition of a single 2D IR spectrum in a few tens of milliseconds using a pulse shaper and an array detector. For some applications, acquiring
one spectrum is enough to answer the questions being asked, and improving the data acquisition time may not be important. Other applications, however, require spectra for many waiting times or may involve considerable signal averaging, and further reducing the time to acquire one spectrum can significantly reduce the overall measurement time.

Previously, Dunbar et al. applied a form of compressive sampling to 2D IR. By scanning small, evenly spaced time points over a relatively short window, they determined peak positions and relative peak amplitudes with a reduction in acquisition time of a factor of 16. Unfortunately, their approach requires that the approximate frequencies, or at least the frequency splitting, must be known a priori to know where to place the sampling window. This method is further limited by the fact that it lacks the ability to reproduce lineshape information.

Experimental Methods

We present 2D IR data collected in our lab measuring the antisymmetric stretch of the azide anion bound to formate dehydrogenase in a ternary complex with NAD$^+$ to determine the frequency-frequency correlation function (FFCF). The full data set required 8-10 days of data acquisition because of the high density of waiting times and the extensive averaging necessary for the measurement. For each 2D IR spectrum, we scan $\tau_1$ from 0 to 4 ps, taking 24 fs steps in the rotating frame and use a four-pulse phase cycle to isolate the signal. We measure waiting times from 0 to 5 ps, taking 50 fs steps. To increase signal to noise, each waiting time is the average of approximately 300 scans. Finally, we average the centerline slope (CLS) values from 20-25 replicate measurements for each waiting time. This data collection requires more than 500 million laser shots, and
with a 1 kHz repetition rate laser would take just under 6 days without missing a shot. In reality, it takes longer because it is necessary to occasionally re-optimize the signal over the course of such a long data collection period.

Compressive sampling favors non-uniform undersampling such that it produces incoherent aliasing artifacts. We implement compressive sampling in the \( \tau_1 \) time delay, meaning we do not take evenly spaced time steps in \( \tau_1 \), but rather a random selection of time points from a normal distribution centered at \( \tau_1 = 0 \) fs. The generation of the spectrum from these undersampled data is an underdetermined problem. We use an algorithm known as total variation (TV) regularized recovery, which is widely used in image processing to recover the data. TV seeks to find a solution that matches the measured data, while minimizing the sum of absolute variation of the solution.

Specifically, the regularization penalty is chosen as \( l_1 \)-norm of the finite differences of the signal (denoted by \( \| \psi x \|_1 \)). Note that \( l_1 \)-norm minimization promotes solutions with few large coefficients over many small coefficients. The above regularization penalty is minimized subject to the constraint that \( \| \mathcal{F}_s x - b \|_2 < \epsilon \), where \( b \) are the compressively sampled time domain measurements themselves, and \( \mathcal{F}_s \) is a one-dimensional inverse Fourier transform along the reconstructed spectral dimension that contains the sampling pattern information needed to select the delay times that are actually measured. The fidelity factor, \( \epsilon \), constrains the reconstructed spectrum to the measured data within a certain experimental noise level. The above optimization problem is solved using an algorithm called as the Alternating Direction Method of Multipliers (ADMM).
Notably, what we refer to as fully sampled is already undersampled using frequency shifting in the rotating frame using the pulse shaper. The oscillation at 2050 cm\(^{-1}\) has a period of 16 fs and must be sampled with time intervals of less than 8 fs according to the Nyquist sampling theorem. However, using time-proportional phase incrementing we shift the observed frequency of the emitted field to a lower frequency so that the signal is fully measured with larger steps. We take 24 fs steps resulting in 167 measurements, or a 3x rate of undersampling, before compressive sampling. The term compression factor as we will use it here denotes the number of \(\tau_1\) measurements relative to this number of measurements.

To assess the effects of different levels of compression, we collect the fully sampled data once and use a pseudorandom mask to filter the fully sampled data set to obtain the compressively sampled data allowing us to avoid noise variations for different measurements. Figure 4.1 illustrates the sampling patterns used at different compression factors. Each dot represents a measurement at one \(\tau_1\) value. The blue row shows all 167 evenly spaced steps of the fully sampled data. The other rows show the exact samples selected from the full set for compression factors of 2\(\times\) (84 points, red), 4\(\times\) (42 points, green), 6\(\times\) (28 points, yellow), and 8\(\times\) (14 points, purple), and 10\(\times\) (13 points, cyan).
yellow), 6× (28 points, purple), 8× (21 points, green), and 10× (17 points, light blue).

The choice of delay time values for the masks is made based on a random Gaussian
distribution of times centered at 0 fs, which gives greater weight to shorter times, when
the signal is stronger. The preferred sampling patterns for exponentially decaying time-
domain data sample the stronger signal portion more heavily and have a reduced
sampling density as one moves out toward longer times. This is similar to Poisson-gap
sampling, which produces non-uniform random sampling patterns with smaller gaps at
the beginning and end of the measurement and has been shown to be consistently more
accurate in reconstruction than uniformly random or other non-uniform sampling
patterns. Additionally, it is more robust, meaning that any one random sampling pattern
of this scheme will not be better or worse than any other random sampling pattern. Due to
apodization, gaps at the end of the sampling pattern are much less impactful than those at
the beginning. The distribution has also been shown to result in better signal-to-noise in
reconstruction than random sampling. The only deviation from the random distribution
is that we sample the first four \( \tau_1 \) points to capture one full oscillation.

We apodize and zero pad the full data set before filtering by the sampling mask.
We apply the same apodization window across the full range (0 – 4 ps, 167 pts) for each
compression factor. Thus, though the last measured time point is different from one mask
to another, the apodization times for different compression factors are consistent. We
zero pad the fully sampled data to 1024 total points, adding 857 zeros after \( \tau_4 = 4 \) ps.
The \( \tau_1 \) time at which zero padding begins and the number of zeros added are also the
same for each compression factor.
We analyze the lineshape of the 0-1 transition in the 2D IR spectra by calculating the CLS\textsuperscript{44-45} and fitting the CLS decay to a model equation. We also fit the peak to a 2-D Gaussian function to determine the peak positions, widths and amplitudes.

Results and Discussion

We use 23 replicates of the 2D IR spectrum at a waiting time $T = 2$ ps to assess the effects of compressive sampling on the signal-to-noise ratio (S/N). Figure 4.2(A-F) shows representative spectra at $T = 2$ ps for fully sampled, 2×, 4×, 6×, 8×, and 10× compression factors. For better comparison of the S/N of the spectra, we have normalized each spectrum to the maximum of the 0-1 transition and we display 11 equally spaced contour levels from -1 to 1. We calculate the S/N by calculating the RMS noise of a spectral area enclosed between $\omega_1 = 2004-2015$ cm\textsuperscript{-1} and $\omega_3 = 2070-2085$ cm\textsuperscript{-1} and we use the difference in amplitudes of the oppositely signed peaks as the measure of the signal. Figure 4.2(G) shows the average S/N as a function of the compression factor, with error bars indicating the 95% confidence interval. Figure 4.2(H) shows that the amplitude of the signal decreases as compression increases (with 95% error bars indicating the distribution from 23 replicates), while the noise decreases in proportion to the signal, leaving the S/N consistent. The relatively large distribution of the calculated S/N comes from data collected over the course of 8 days. As already mentioned, there are signal fluctuations that vary with laboratory measurement time over the course of such a long data collection period. Notably, both the mean S/N and the distribution are unaffected by compression. The underlying data spans the entire data collection period, and it is likely that when
compressively sampled over a shorter period of laboratory time, the S/N may actually improve somewhat with increasing compression.

Figure 4.3 shows the parameters from the two-dimensional Gaussian fits of the 0-1 transition at $T = 2$ ps. We fit the lineshape using the functional form:
$S = A \exp \left\{ \frac{-1}{2(1 - \text{cor}^2)} \left[ \frac{(x-x_0)^2}{\sigma_x} + \frac{(y-y_0)^2}{\sigma_y} - \frac{2\text{cor}(x-x_0)(y-y_0)}{\sigma_x \sigma_y} \right] \right\}$

where $A$ is the amplitude, $x_0$ and $y_0$ are the center frequencies in $\omega_1$ and $\omega_3$, $\sigma_x$ and $\sigma_y$ are variances of the peak in $\omega_1$ and $\omega_3$, and $\text{cor}$ is the 2D Gaussian correlation coefficient, which is proportional to the value of the FFCF. The amplitude (not shown) follows the same trend as the signal (Figure 4.2(H)). Figure 4.3(A) shows that the correlation coefficient decreases only slightly as compression increases, until a significant drop occurs at compressions of 8× and higher.

Figure 4.3(B) shows the observed center frequency in both the pump (blue) and probe (red) dimensions as a function of compression. We observe that the center
frequency and width of the peak on the probe axis remain unchanged, and the peak center on the pump axis does not change from the fully sampled peaks until a shift of 1.5 cm\(^{-1}\) at 8\(\times\).

The peak width in \(\omega_1\) gradually and steadily increases with compression factor (Figure 4.3(C)), effectively stretching the peak. This effect is similar to that of shortening the apodization window, which we have reported on previously.\(^{44}\) Though the apodization window covers the same time range for every compression factor, at greater compression there are fewer points contained within the measurement vector leading to poorer spectral resolution in the pump dimension.

Figure 4.4(A) shows the average CLS as a function of waiting time along with the fit to the CLS decay, at each level of compression. For a given waiting time, increasing compression depresses the CLS. The most important measure, however, is not the absolute CLS at a particular waiting time but the CLS decay as a function of waiting time, which is proportional to the normalized FFCF.

As is clear from Figure 4.4(A), the behavior naturally falls into two groups, with fully sampled, 2\(\times\), and 4\(\times\) in one group, and the higher compressions in another. In the first group, the overall shapes of the CLS decays are qualitatively similar, clearly capturing the oscillations that are present, but with a small shift in the offset of the CLS with increasing compression. For the second group in Figure 4.4(A), the CLS decay at higher compression factors do not clearly show the same oscillations in the CLS and the uncertainties in the individual CLS values become larger than the variations in the CLS value as a function of waiting time. Oscillations do appear in the 6\(\times\) and 8\(\times\) results, but it is not clear that they are the same oscillations as the first group.
For a more quantitative analysis of these results, we fit the CLS decay to the following functional form,

\[ CLS(T) = A_0 e^{-T/\tau_0} + \sum_{i=1,2} A_i e^{-T/\tau_i} \times \cos(2\pi \omega_i T + \phi_i) \]  

(2)

where we use a sum of exponential functions that includes damped cosines to model the oscillatory features in the decay. We have described the origin of the oscillatory features elsewhere,\textsuperscript{37} and here we focus on the ability of compressive sampling to resolve these oscillations and return the correct frequencies, time constants, and relative amplitude parameters. Figure 4.4(B-I) shows the fit parameters with 95% confidence intervals.

Increasing compression results in increases in the errors on the parameters of both the high-frequency (blue) and low-frequency (red) oscillatory terms with the most substantial effect occurring between 4× and 6× compression. For both the low-frequency oscillatory

---

**Figure 4.4 CLS and Fit Parameters**

The CLS as a function of waiting time (A) for fully sampled (blue), 2X (red), 4X (yellow), 6X (purple), 8X (green), and 10X (teal). Each point is the average CLS from the analysis of 20-25 replicates, and lines are fits to Equation 2. Panels (B-I) show the fit parameters of the high frequency (blue), low frequency (red), and non-oscillating (black) terms of Equation 2 at each compression factor.
term and the non-oscillating, slow exponential decay term there is a large decrease in amplitude between 4× and 6× compression. The amplitude of the oscillating terms have only small variations up to 4×, though the non-oscillating term has a clear downward trend. These changes show that the effects of increased compression on the CLS decay are not simply a multiplicative effect on the relative amplitude. Rather, there is a differential impact on each of the three terms that contribute to the overall decay. In addition, we have fit to a predetermined model that includes two oscillations and a slow exponential decay. At the higher order compressions, however, this model is not justified based on the quality of the CLS data, and, therefore, this model yields parameters with large uncertainties. We have used this fit function because we know a priori what the functional form of the CLS decay is already from the fully-sampled measurements. When actually using compressive sampling, however, we will not know the correct model a priori and must use statistical methods to determine the minimum appropriate model. At 6×, a statistical F-test comparing the full model to a fit with only one oscillatory term revealed that the more complicated model is statistically justified ($F = 3.7 > F_{crit} = 2.7$). At 8× and 10× compressions, however, the weighted sum of squared residuals for the model fit is barely better than that for a linear fit indicating that by the highest levels of compression the lineshape is sufficiently distorted that the correct parameter values cannot be determined from the data. Table 1 summarizes the CLS fit parameters for the fully sampled, 2×, and 4× data.
Table 4.1 CLS Parameters with Compressive Sampling

<table>
<thead>
<tr>
<th></th>
<th>$A_0$</th>
<th>$\tau_0$ (ps)</th>
<th>$A_1$</th>
<th>$\tau_1$ (ps)</th>
<th>$\omega_1$ (cm$^{-1}$)</th>
<th>$A_2$</th>
<th>$\tau_2$ (ps)</th>
<th>$\omega_2$ (cm$^{-1}$)</th>
<th>Square root of Relative Amplitudes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully sampled</td>
<td>0.26±0.003</td>
<td>80±25</td>
<td>0.050±0.02</td>
<td>64±0.2</td>
<td>240±1.6</td>
<td>0.069±0.01</td>
<td>1.08±0.17</td>
<td>9.9±0.46</td>
<td>0.83±0.02</td>
</tr>
<tr>
<td>2×</td>
<td>0.24±0.004</td>
<td>92±47</td>
<td>0.056±0.03</td>
<td>62±0.3</td>
<td>239±2.0</td>
<td>0.065±0.01</td>
<td>1.12±0.24</td>
<td>10.3±0.63</td>
<td>0.81±0.08</td>
</tr>
<tr>
<td>4×</td>
<td>0.21±0.007</td>
<td>72±52</td>
<td>0.048±0.03</td>
<td>63±0.4</td>
<td>233±3.5</td>
<td>0.067±0.01</td>
<td>1.08±0.34</td>
<td>10.3±0.99</td>
<td>0.80±0.08</td>
</tr>
</tbody>
</table>

The magnitude of the CLS varies with compression factor in a similar way that the CLS depends on the apodization. The exact value of the CLS is not critical if the time constants, frequency components, and relative amplitudes remain unaffected. Without correct absolute amplitudes, it is possible to extract the full FFCF, including homogeneous contribution and the absolute amplitudes of inhomogeneous frequency fluctuations, from the compressively sampled CLS decay by fitting to the FTIR spectrum. The amplitude of each term of the full FFCF is proportional to product of the FWHM of the FTIR spectrum and the square root of the relative amplitudes of the CLS, which are shown in the last three columns of Table 3.1. The square root of the relative amplitudes at 2× and 4× compression are not all within the 95% confidence limit of the fully sampled measurement. However, all of the compressively sampled values are within 10% of the fully sampled values. At 4× compression, the time constants, frequency components, and relative amplitudes are remarkably close given that this is a first attempt at compressive sampling and we use an unrefined reconstruction algorithm. This is a reasonable compromise in a case where the measurement may not be feasible without compressive sampling.
The appropriate compression factor at which to sample a new system will be unknown *a priori*. However, there are practical ways to determine the highest compression factor that still provides the required information. For example, one can fully measure only a few waiting times and compare those data to those obtained from compressively sampling. This will not add a significant amount of time to the measurement, preserving the benefits of compressive sampling, and will provide a check to ensure adequate results. Additionally, one advantage of compressive sampling is that it is future-proof, meaning more data can be collected and added to the set. Furthermore, if a more robust and accurate algorithm specific to this application is developed, the same data can be used to reconstruct spectra.

Conclusions

Based on our results, it is clear that the acceptable level of compression depends on the required information. At 10× compression, which only measures 17 time points in \( \tau_1 \), the reconstructed peaks have comparable peak positions and S/N to the fully sampled data, but the lineshapes are significantly distorted. For some applications, this result would be entirely adequate. For many applications, however, the CLS and lineshape information are essential. Based on qualitative and quantitative inspection of the CLS decay curves and CLS decay fit parameters, most importantly the frequencies and time constants of the oscillations, compressive sampling can accelerate data acquisition of 2D IR spectra by a factor of at least 4× while still adequately reproducing lineshapes, which is still a significant acceleration. With 4× compression, the 8 days required to collect the FDH data presented here can be reduced to 2 days, which is a substantial improvement.
An area of continued study is to test the difference between compressively sampling 2D IR with and without frequency shifting. In the future, it may be more practical to talk about the number of measurements, which corresponds directly with laboratory time, rather than compression factor. For example, the same 42 time delay measurements which correspond to 4× compression in the current frequency shifted scheme is 12× compression if sampled in the true carrier frequency scheme. Additionally, when probing higher or lower vibrational frequencies, the Nyquist sampling rate scales proportionally and the same compression factor corresponds to different numbers of measurements. There ultimately will be a lower limit to the number of measurements required. For example, because fewer points are needed to fully sample lower vibrational frequencies to reproduce the relevant lineshape features, higher compression factors will not necessarily be obtained at lower vibrational frequencies.

It is a likely that even greater compression than shown here is possible. We have used a common reconstruction algorithm from image processing with essentially no optimization of the reconstruction parameters. In addition, it is possible to compressively sample in both \( \tau_1 \) and the waiting time, \( T \). Two-dimensional reconstruction methods could leverage the redundancy of information in the 2D IR spectra at different waiting times. Using different masks at different waiting times and compressively sampling in both time delays, it is possible that we could realize an order of magnitude or more increase in the compression factor without distorting the lineshapes significantly. Such possibilities could eventually enable us to collect the full FDH sample presented here in a matter of a few hours rather than days. Pulse shaping will be a key advantage in
experimentally implementing compressive sampling methods. Taking uneven step sizes quickly is difficult with translation stages, but is trivial with a pulse shaper.
5. THE GIRAF RECONSTRUCTION ALGORITHM

Motivation and Introduction

In the previously presented work, the compressive sampling reconstruction algorithm based on the generic image processing algorithm known as Total Variation was sufficient as 2D IR compressive sampling proof of concept. This method and others assume no \textit{a priori} knowledge about the underlying structure of the signal other than its sparsity in the Fourier domain\textsuperscript{46} or its piece-wise smoothness.\textsuperscript{47} This makes the TV reconstruction useful in wide application, but not ideal in the specific application to 2D IR. In fact, we do know the form of the response function of 2D IR signal in the time domain, and we exploit this information to engineer a more efficient reconstruction algorithm specific to 2D IR with superior recovery.

Our collaborators engineered the Generic Iteratively Reweighted Annihilating Filter (GIRAF)\textsuperscript{48-49} reconstruction algorithm to solve the reconstruction in 2D IR compressive sampling, and the details of the algorithm are given in our recently published work.\textsuperscript{50}

While the vibrational spectra of 2D IR may contain few peaks in a large matrix of mostly zero, the peaks themselves are often broad and require several spikes to be represented in the Fourier domain. Thus, the Fourier representation is non sparse and the
assumption of sparsity leads to challenging signal recovery. The Fourier basis is represented in the time domain as undamped exponentials. We observe that a broad peak in frequency can be efficiently approximated as linear combination of a few damped exponentials in the time domain, with different frequencies and damping coefficients. Damped exponentials is a richer representation than the Fourier basis, which is composed of undamped exponentials and is essentially a subspace. Using a linear combination of Lorentzians to approximate smooth functions is well studied. In fact, a Voigt lineshape is well approximated by three or four Lorentzians.51

Briefly, the algorithm uses the property of damped exponentials that they can be annihilated by a filter that has been parameterized by the frequencies and damping coefficients.52 The annihilation property results in a low-rank constraint on the Toeplitz matrix constructed from the signal samples. The low-rank constraint on the Toeplitz matrix leads to modeling the signal samples with as few damped exponentials as possible within a threshold of error. The aim of the algorithm is to minimize the cost of the image recovery according to the equation

$$c(\tilde{x}) = \|Ax - \tilde{b}\|^2 + \lambda \cdot f(\tilde{x})$$

where $\tilde{x}$ is the fully reconstructed 2D IR spectrum we for which we are solving, $A$ is an operator which filters $\tilde{x}$ for the samples measured, $\tilde{b}$ is the measured data, $\lambda$ is a Lagrange multiplier and $f(\tilde{x})$ is a constraint imposed on the problem, which in our case is the Schatten $p$ norm, which is a proxy for the rank, which is proportional to sparsity. The first term is called the data fidelity term because it matches the result with the measured data, and the second term is called the regularization term because it imposes the model onto the solution. The algorithm does not simply put $\tilde{b}$ into $\tilde{x}$ and fill in the missing points, but
rather recovers a whole new matrix based on the signal samples and within the constraints of the model.

We collect 2D IR in time – frequency mode. We measure the probe frequency directly in a spectrometer by spectrally dispersing the signal onto a linear array detector and we collect the pump frequency dimension interferometrically by scanning the time delay between the first two pulses and performing a Fourier transform to frequency. The full probe axis is acquired on every laser shot as a function of pump time delay. GIRAF reconstructs the signal in time – time mode. Therefore, we truncate the probe frequency axis to remove noise from the edges of the array and perform an inverse Fourier transform to the fully time-dependent 2D IR signal.

We have shown that non-uniform undersampling provides lower errors than uniform undersampling with GIRAF, and we collect data as a function of pseudorandom time delay points as we did for the TV method.

Algorithm Advancements

As we have discussed before, compressive sampling is progressive. Once a more advanced algorithm is engineered, one can use the same data to recover better reconstructions. To determine the qualitative and quantitative performance of the GIRAF algorithm, we use the same FDH data from the previous study, apply the exact same sampling masks, and compare the reconstructed spectra from the GIRAF algorithm to those from the TV algorithm.

Again, we use 23 replicates of the 2D IR spectrum at a waiting time \( T = 2 \) ps to assess the effects of GIRAF on the signal-to-noise ratio (S/N). Similar to Figure 4.2,
Figure 5.1 GIRAF Reconstruction with SNR

Representative 2D IR spectra, reconstructed with the GIRAF algorithm, of azide bound to FDH with NAD+ at a waiting time of $T = 2$ ps from full sampled to 2X, 4X, 6X, 8X, and 10X compressed, normalized and with equal contour lines. Panels (G) and (H) show the average signal-to-noise ratio from 23 replicates at the experimental compression levels and the average signal (blue) and noise (red), respectively.

Figure 5.1(A-F) shows representative spectra at $T = 2$ ps for fully sampled, 2X, 4X, 6X, 8X, and 10X compression factors. We have normalized each spectrum to the maximum of the 0-1 transition and we display 11 equally spaced contour levels from -1 to 1. We observe qualitatively that the signal relative to the noise does not degrade at higher levels of compression. At a compression level of 10X we see almost no noise in the spectrum.
Figure 5.1G shows the average signal-to-noise ratio (SNR) as a function of compression factor, which we obtain by calculating the RMS noise of a spectral area enclosed between \( \omega_1 = 2004-2015 \text{ cm}^{-1} \) and \( \omega_3 = 2070-2085 \text{ cm}^{-1} \) and we use the difference in amplitudes of the oppositely signed peaks as the measure of the signal. SNR is calculated for each spectrum and then the mean is calculated. Remarkably there is a linear increase in the SNR as compression factor increases. At the highest compression level of 10\( \times \) (17 measurements), the SNR is a factor of almost 6 greater than the fully sampled value.

Figure 5.1H shows that the amplitude of the signal decreases slightly as compression increases while the noise decreases a relatively greater amount, leading to increasing SNR. The scale of Figure 4.2H and Figure 5.1H are the same, and we observe that the signal is more stable and the noise is decreased with the GIRAF reconstruction than with TV. We see a decrease in the noise when GIRAF is used because the algorithm does not model the noise but reconstructs the spectrum \( \tilde{x} \) as a linear combination of few damped exponentials based on the measured data and within the constraints of the problem.

The metric of importance is the centerline slope (CLS) calculated from the 2D IR spectrum, which leads to the frequency-frequency correlation function. The CLS is a value sensitive to the lineshape of the diagonal peak. Figure 5.2 shows the average CLS decay for each compression level as a function waiting time based on TV reconstruction (A) and GIRAF reconstruction (B). Each point is the average of 20-25 trials and we show 95% confidence bars. Results based on the GIRAF reconstruction shows the behavior of the CLS decay still falls naturally into two groups, with the fully sampled, 2\( \times \), and 4\( \times \) on top, and 6\( \times \) and 8\( \times \) below (10\( \times \) not shown), though the distinction is not as clear as the TV reconstruction. The biggest improvement of GIRAF over TV is in the amplitude of
the CLS decay, which is a measure of the elongation of the peak along the diagonal. At compression levels of $2\times$ and $4\times$ the amplitudes across all waiting times track consistently on top of the fully sampled values. The CLS decay at $6\times$ with GIRAF is comparable to the CLS decay at $4\times$ with TV.

We fit the CLS decay to a function composed of three exponential decay terms with oscillating parts on two of the terms (Equation 2) as discussed in the previous chapter. The time constants and oscillation frequencies from the fit are consistent with the TV results. All three amplitude values from the fit are improved with GIRAF at every level of compression.
The purpose for calculating the CLS is to obtain the full frequency-frequency correlation function, in which the amplitude of each term is directly proportional to the square root of the relative amplitudes of the CLS terms. With GIRAF these values are within the error of the fully sampled values up to $4\times$ compression, except for a slight error of A2. Figure 5.3 shows these results with the black point and black lines corresponding to the fully sampled value and its limits of 95% confidence, blue points show the values from the GIRAF reconstruction, and red points show the values from the TV reconstruction ($6\times$ and $8\times$ compression for TV are not shown because they are out of the range of the axes).

At comparable levels of compression, GIRAF performs as good or better than TV at reconstructing signal, SNR, and lineshapes. Notably, we observe an increase in SNR from the fully sampled results at every level of compression. This implies that we may be able to accelerate data acquisition beyond simply compressively sampling time delays. With higher SNR we can reduce the number of scans we average to acquire a 2D IR.
spectrum, decreasing the number of laser shots required to collect our datasets by another factor of 2× or greater.

The Effects of Frequency Shifting

Previously, we have discussed the benefits of compressive sampling in terms of a data acquisition compression factor, such as 4× compression. That is, how many measurements relative to the fully sampled and frequency-shifted experiment are required. However, frequency shifting is itself a form of undersampling that should be considered. It is more practical to talk about the total number of time delay measurements required, which corresponds directly to the experiment time. Thus, some combination of frequency shifting and compressive sampling results in the fewest total measurements.

Briefly, we use the pulse shaper to frequency shift into the rotating frame and shift the observed carrier frequency of the emitted field to a lower frequency so that the signal is fully measured with larger steps. This is done by incrementing the phase of the pump pulses proportional to the time delay.34 The only possible signal is found in the relatively narrow bandwidth of the laser pulse so aliasing is corrected. The shape of the envelope function is maintained.

As analogy, consider a car going around a circle track at a rate of one lap per minute from the frame of reference of an observer standing at the starting line. The car will pass the observer once a minute. Now another observer is in a car also traveling around the track at a slower speed than the first car, say one lap per two minutes. That is, this viewer is in a rotating frame of reference around the track. The second observer will
see the first car pull away at a rate that is the difference between the two speeds. The first car will be traveling at a slower observed rate of one lap in two minutes.

We use simulated 2D IR data to demonstrate the qualitative and quantitative effects of frequency shifting. We simulate the 2050 cm\(^{-1}\) azide peak using a three-level system Kubo lineshape model.\(^{31}\) The correlation time is set at 2.5 ps with peaks centered at 2050, 250, and 50 cm\(^{-1}\), corresponding to the same peak with frequency shifts of 0, 1800, and 2000 cm\(^{-1}\), respectively. All simulated data parameters have been generated to match with experimental data. Experimentally we use a frequency shift of 1800 cm\(^{-1}\) and 24 fs size steps covering a time delay range from 0 – 4 ps with a sampling rate of 5.5 samples per period of oscillation. The data is zero padded to a total of 1024 points, a total time range of 24.5 ps. The data collection range and total time range are consistent in all three simulated datasets to maintain consistent frequency spacing. We select the step sizes in the simulated datasets to obtain the same sampling rate for each simulated signal.

Figure 5.4 FID Without and With Frequency Shifting

The free-induction decay at the probe frequency of the peak. Panel A shows the fully sampled 2050 cm\(^{-1}\) signal, with a 16.2 fs period sampled at 3 fs steps resulting in 1333 measurements. Panel B shows the FID for the signal frequency shifted by 2000 cm\(^{-1}\) to a 50 cm\(^{-1}\) signal, which has a period of 667 fs and we sample at 121 fs steps for a total of 4 ps resulting in 34 measurements.
Figure 5.4 shows the effects of frequency shifting on the simulated time data. The free-induction decay (FID) at a waiting time of 2 ps at a given $\omega_{probe}$ value is shown with no frequency shifting (B) for a 2050 cm$^{-1}$ signal. With a 16.2 fs period this signal is sampled at 3 fs size steps for a total of 1333 measurements. The FID is shown for a high level of frequency shifting (A) where the signal is shifted by 2000 cm$^{-1}$, resulting in a 50 cm$^{-1}$ observed signal. With a period of 666.7 fs this signal is sampled at 121 fs size steps.

Figure 5.5 2D IR Spectra with Different Frequency Shifting

Figure 5.5 2D IR Spectra with Different Frequency Shifting

![2D IR Spectra](image)

2D IR spectra of a 2050 cm$^{-1}$ signal simulated with zero frequency shift (A), 1800 cm$^{-1}$ frequency shift (B), and 2000 cm$^{-1}$ frequency shift (C). These spectra are acquired from 1333, 167, and 34 samples, respectively. The highest amount of frequency shifting is 40X undersampled relative to the fully sampled case.

resulting in 34 measurements. The control case applies an 1800 cm$^{-1}$ shift to the signal with 24 fs size steps. We apply the same effective sampling rate, 5.5 samples/period, in each case. We use a waiting time of 0 ps in all simulations.

One drawback of frequency shifting is observed in Figure 5.4. Because we take larger steps, the maximum and minimum peaks of the oscillation are not consistently
measured and the total amplitude range is greatly diminished, resulting in smaller signal intensity.

Figure 5.6 GIRAF Reconstruction of Frequency Shifted Data

![Figure 5.6 GIRAF Reconstruction of Frequency Shifted Data](image)

Spectra reconstructed with the GIRAF algorithm from 10 measurements with zero frequency shift (A), 1800 cm\(^{-1}\) frequency shift (B), and 2000 cm\(^{-1}\) frequency shift (C), corresponding to compression levels of 133X, 17X, and 3.4X, respectively. Panel D shows the fully sampled spectrum with no frequency shifting and no compressive sampling acquired from 1333 samples.

We test the effects of frequency shifting on compressive sampling reconstructions by compressively sampling the same simulated signal with zero, 1800, and 2000 cm\(^{-1}\) frequency shifts. Figure 5.5 shows the fully sampled spectra calculated with the experimental details as described above. Though the absolute amplitude of the spectrum decreases with higher frequency shifting, comparable signal-to-noise ratio is observable in the figure, displayed with 11 even spaced contours. All three of these fully sampled experiments give very similar spectra and validate the use of frequency shifting. The three spectra shown are acquired from 1333, 167, and 34 time delay measurements (A-C).

The sequence of reconstructing spectra that have been acquired with both frequency shifting and compressive sampling starts with using GIRAF to reconstruct the
spectrum and then simply adding the frequency shift value to the spectral axis. Frequency shifting is not a GIRAF input parameter. The GIRAF algorithm reconstructs a spectrum for a given set of data, and the fewer the missing data points the better the reconstruction.

An important aspect of combining frequency shifting and compressive sampling is the accuracy of construction of a spectrum from frequency shifting alone, without compressive sampling. A given number of measurements will be a smaller compression factor relative to a higher amount of frequency shifting than to a smaller one, and we have shown that reconstructions improve as compression factor decreases. More frequency shifting allows for better reconstructions at fewer measurements up to the point that frequency shifting alone leads to accurate spectrum construction. Indeed, we observe that the highest frequency shift tested leads to the most accurate reconstruction at few measurements.

We compressively sample these three cases with various numbers of measurements. Figure 5.6 shows the results of compressively sampling with only ten measurements. At 0, 1800, and 2000 cm\(^{-1}\) frequency shifting this is 133\(\times\), 17\(\times\) and 3.4\(\times\) compression, respectively. With no frequency shifting, GIRAF reconstruction obtains correct peaks, but also reconstructs aliased peaks all along the pump frequency, as it is unable to pinpoint the correct frequency from such a high level of compression. At a frequency shift of 1800 cm\(^{-1}\) only the correct peaks are obtained, but with distorted lineshapes. A frequency shift of 2000 cm\(^{-1}\) gives the best reconstruction at this given number of measurements.
We perform a quantitative analysis of the peak shape by fitting the diagonal peak to a two-dimensional Gaussian function. Figure 5.7 shows the full-width at half the maximum (peak width) along the compressed dimension (A), the correlation value of the 2D Gaussian expression (B), and the center frequency of the peak (C). Fit parameters of spectra reconstructed from 40 measurements and fewer are shown. The true values are considered the fit of the zero-frequency shifted, fully sampled spectrum obtained from Figure 5.7 2D Gaussian Fit Parameters for GIRAF with Different Frequency Shifts

2D Gaussian fit parameters of reconstructed spectra of various levels of frequency shifting and numbers of measurements. The FWHM of the peak along the compressed dimension (A), the 2D Gaussian correlation coefficient (B), and the peak position (C) are shown with zero frequency shift (black), 1800 frequency shift (red), and 2000 frequency shift (blue) and the value from the fully sampled spectrum as the line. The highest frequency shift has the best results at the lowest number of measurements. Panel D shows the amplitude.
1333 measurements. All three levels of frequency shifting give similar values down to 20 measurements. At 10 measurements, the higher frequency shift is more accurate for all parameters.

There is appears to be a lower limit to the number of measurements required to reconstruct accurate spectra. With only 5 measurements, the reconstruction is very poor quality even though it is only $6.8 \times$ compression with a 2000 cm$^{-1}$ frequency shift. The current sampling scheme always measures the first four time delays, which means that when five measurements are made only one more point is measured after the first four. A future direction is experimenting with different sampling schemes that don’t always include the first four points, or always include the first five or six. It may be beneficial to acquire at least the first full oscillation.

Figure 5.7D shows the amplitude for each level of frequency shifting at each number of measurements, from the fewest to fully-sampled, shown as a log scale. As expected, we see that both frequency shifting and compressive sampling cause a decrease in signal. However, also as expected, the noise decreases and the signal-to-noise ratio is consistent.

We further analyze the lineshape of the reconstructed peak by calculating the centerline slope (CLS). The importance of the CLS and the method of calculation are described in a previous section. Figure 5.8 shows the calculated CLS from various numbers of measurements from each of the frequency shift values. Each plot shows the CLS calculated from the full range of number of measurements, requiring a log scale on the x-axis for all but the highest frequency shift. We see that the same CLS is calculated for each fully sampled frequency shift value, thus frequency shift up to at least this high
does not affect the CLS, preserving the lineshape. At comparable compression factors, we calculate more accurate CLS values at the lower frequency shifted data. However, as a factor of absolute number of measurements, the higher the frequency shift the more accurate the CLS.

From this analysis, frequency shifting allows for better reconstruction at fewer measurements, up to the point that frequency shifting alone returns accurate spectra. In fact, optimizing frequency shifting alone may be better than previous attempts at
compressive sampling. Only increasing the frequency shift from 1800 cm\(^{-1}\) to 2000 cm\(^{-1}\) is equivalent to 5\(\times\) compression relative to our current method. In combination, optimizing frequency shifting and compressive sampling will lead to the fewest required measurements and the greatest data acquisition acceleration. We have shown a 50\(\times\) level of compression from the minimum of two samples per oscillation required by the Nyquist sampling theorem. In practice, more than two samples per oscillation are often acquired and we have shown a factor of 133\(\times\) acceleration from a common sampling rate.

We also studied the robustness of GIRAF in the presence of artificially added Gaussian noise to the time domain data before performing reconstruction. Three different signal-to-noise (SNR) levels of 20, 34, and 40 dB are tested at the ten-measurement level with a 2000 cm\(^{-1}\) frequency shift. SNR is calculated in dB as

\[
SNR_{dB} = 20 \times \log \left( \frac{\text{norm}(\text{true time data})}{\text{norm}(\text{noise})} \right)
\]

where \text{norm} indicates the \(\ell^2\)norm of the data matrix and the generated random noise matrix. The ratio inside the log argument takes values of 10, 50 and 100 for 20, 34, and 40 dB, respectively.

Figure 5.9A-D shows the 2D IR peaks reconstructed with GIRAF from 10 measurements of time data and 2000 cm\(^{-1}\) shift frequency. The title inside each spectrum indicates the SNR levels of the input time data, not of the reconstructed spectrum. The spectrum generated from the time data with the most noise also has the most noise. However, even at the lowest SNR the overall peak shape is maintained, as seen in Figure 5.9E showing the normalized diagonal 0 \(\rightarrow\) 1 transition peak profile along \(\omega_{pump}\). In all cases using compressive sampling, GIRAF produces wings at the edge of the peak.
because it is unable to perfectly isolate the prominent frequencies from such few measurements.

We calculate SNR for spectra by finding the signal as the half the difference between the oppositely signed peaks and the noise as the root-mean-square of a spectral area away from the peaks. In all cases of adding noise to the time domain data the signal
intensity of the reconstruction was unaffected compared to the case without added noise. The spectral noise was most affected for the lowest SNR time dataset, with a reconstructed noise an order of magnitude greater than the other cases. Calculated values are seen in Table 4.1.

Table 5.1 Signal, Noise, and SNR of GIRAF from Noisy Simulated Data

<table>
<thead>
<tr>
<th>SNR of Simulated Time Data</th>
<th>GIRAF Spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Signal</td>
</tr>
<tr>
<td>20 dB</td>
<td>178.40</td>
</tr>
<tr>
<td>30 dB</td>
<td>175.66</td>
</tr>
<tr>
<td>40 dB</td>
<td>177.93</td>
</tr>
<tr>
<td>No Noise Added</td>
<td>177.14</td>
</tr>
</tbody>
</table>

The peak shape and peak properties are quantitatively analyzed by fitting to a 2D Gaussian. Figure 5.10 shows the fit parameters for the three test cases (with 95% confidence interval) and the values from the GIRAF reconstruction with no noise input.

Figure 5.10 2D Gaussian Fit Parameters from GIRAF from Noisy Simulated Data

![Figure 5.10](image_url)

2D Gaussian fits of the spectra peaks reconstructed from 10 samples of 2000 cm⁻¹ simulated data as a function of the SNR of the time domain data. The red line indicates the true value; blue line is the GIRAF reconstructed value from no noise added time data. The added noise has negligible effect in all but the case with the most noise added.

(blue line) and the fully sampled no frequency shifted true value. The peak width and center frequency of the ω_pump axis are given because this is the compressively sampled dimension. The amplitude is found as the signal in Table 4.1. In the test cases with time
domain noise of 34 and 40 dB the results are easily within error bar of the control for all parameters. The wings on the edge of the peak from the 20 dB data elongate the peak width and lower the correlation, however these values are still very close to the control. GIRAF is able to determine the same peak position in all cases.

These results indicate that frequency shifting and compressively sampling with GIRAF reconstruction is robust to time domain noise and can accurately produce spectra from as little as ten time points. Using a four-pulse phase cycle to eliminate scattering in experimental data collection, in some cases we may be able to acquire a full 2D IR spectrum in 40 laser shots.

The Effects of Multiple Peaks

In many interesting experiments, the system under study contains multiple peaks and the experimenter is interested in the features of multiple peaks, such as the relative amplitude of those peaks, or the rate of peak growth or decay. A system with more peaks is less sparse than a system with only one. The aim of this study is to demonstrate that one can use GIRAF to effectively reconstruct the spectrum of a system with multiple peaks.

The system under study, rhodium dicarbonyl (RDC) in dichloromethane solvent, has two peaks resulting from the symmetric and antisymmetric stretches of the carbonyl, as well as cross peaks resulting from coupling between the two vibrational modes. RDC is a strong infrared chromophore with high signal intensity. Therefore, RDC is a good model to test the GIRAF algorithm by controlling for signal and isolating the dependent variable as the fact that there are multiple peaks.
We measure the 2D IR spectrum of RDC in DCM with an 1800 cm$^{-1}$ frequency shift, over a waiting time spanning 80 picoseconds. We compressively sampled data with 9, 17, 21, 28 tau time measurements. These values correspond with 20×, 10×, 8×, 6× undersampling relative to the frequency shifted data collection. The result from the data collected with 167 measurements is not compressively sampled and constructed from the algorithm and is considered the true data. Compressive sampling and frequency shifting in combination result in 148×, 78×, 64×, 48× and 8× undersampling from the true fully sampled data.

Figure 5.11 shows reconstructed spectra at each level of compression few a waiting times. Descending each column shows the reconstruction for a given waiting time for a decreasing number of measurements. Each column shows the change in the
Figure 5.12 RDC Peak Amplitudes

The absolute peak amplitude of each of the four peaks for each level of compression as a function of waiting time. Panels A-D are positioned where the peaks are observed in the 2D IR spectrum, such that C and B are the diagonal peaks and A and D are cross peaks. The oscillations in the first 2 ps of waiting time are shown in the insets. The peak amplitudes are shown for 167 (blue), 28 (red), 21 (yellow), 17 (purple), and 9 (green) measurements.

spectrum for a different number of measurements at a given waiting time. At zero waiting time, the diagonal peaks are qualitatively similar, even at only 9 tau times measured. The cross peaks are not as well resolved above the noise at fewer measurements. At longer waiting times the cross peaks grow in to be comparable intensity as the diagonal peaks and the reconstruction gets worse. The peaks spread along $\omega_1$, the pump axis, at fewer measurements. The pump dimension is the compressively sampled dimension, and the GIRAF algorithm is unable to resolve the two peaks. At 28 measurements, spectra at all waiting times reconstruct all four peaks, and at as few as 21 measurements diagonal peaks are qualitatively reconstructed. A more quantitative study follows.
Figure 5.12 shows the absolute amplitudes of all four peaks as a function of waiting time. The diagonal peaks decay exponentially, and the cross-peaks grow in to about 20 picoseconds and then decay. Insets show just the first 2 picoseconds. Diagonal peak decay can be approximated by a bi-exponential decay fit with fast and slow time constants. The fit parameters of the peak at \((\omega_1, \omega_2) = (2083,2083)\) are shown in Table 4.2, with 95% confidence interval. We see the overall lower intensity of the peak at higher compression in the diminishing amplitude of the slow decay at higher compression.

GIRAF underestimates the absolute peak intensity for all peaks, which we expect based on previous studies. When compressively sampling with fewer measurements, the full amplitude of the free induction decay is not captured, and the resulting reconstructed spectrum has lower intensity. However, the change we see is small relative to the peak intensity, and from Table 1 we see that GIRAF reconstruction obtains accurate peak amplitude and rate of peak change with only 17 measurements.

In the fully sampled spectrum, the peak intensities oscillate in the first picosecond of waiting time, as shown in the insets of Figure 5.12. With as few as 17 tau time measurements we obtain the oscillations, though with a smaller amplitude of oscillation.

Often when studying a system with multiple peaks, we are interested in the relative amplitudes of the peaks. We want to determine population transfers from one

<table>
<thead>
<tr>
<th>(n) points</th>
<th>A1</th>
<th>(\tau_1) (ps)</th>
<th>A2</th>
<th>(\tau_2) (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>0.47 ± 0.05</td>
<td>2.0 ± 0.4</td>
<td>0.64 ± 0.05</td>
<td>48 ± 7</td>
</tr>
<tr>
<td>28</td>
<td>0.47 ± 0.06</td>
<td>1.9 ± 0.5</td>
<td>0.57 ± 0.06</td>
<td>43 ± 7</td>
</tr>
<tr>
<td>21</td>
<td>0.46 ± 0.07</td>
<td>2.5 ± 0.7</td>
<td>0.53 ± 0.07</td>
<td>44 ± 9</td>
</tr>
<tr>
<td>17</td>
<td>0.41 ± 0.09</td>
<td>2.4 ± 0.8</td>
<td>0.50 ± 0.10</td>
<td>38 ± 9</td>
</tr>
<tr>
<td>9</td>
<td>0.54 ± 0.16</td>
<td>0.8 ± 0.3</td>
<td>0.21 ± 0.09</td>
<td>40 ± 22</td>
</tr>
</tbody>
</table>
mode to another or how coupling strength, for examples. Figure 5.13 shows the relative peak intensities between the diagonal peaks and the cross peaks. The y-axis shows the proportion of the total peak intensity that is contained in the two diagonal peaks. Relative peak intensity increases when fewer measurements are used; 167 (blue), 28 (red), 21 (yellow), and 17 (purple).

All levels of compressive sampling produced higher S/N

| Table 5.3 Relative Amplitude Decay Fit Parameters |
|----------------|----------------|
| 167 measurements | 0.76 ± 0.03 | 120 ± 24 |
| 28 measurements   | 0.83 ± 0.03 | 101 ± 18 |
| 21 measurements   | 0.87 ± 0.03 | 98 ± 16  |
| 17 measurements   | 0.87 ± 0.02 | 102 ± 15 |

Figure 5.13 Relative Peak Amplitudes

The relative amplitude of the diagonal peaks for different number of measurements as a function of waiting time. The relative amplitude is given as the proportion of the total peak intensity that is contained in the two diagonal peaks. Relative peak intensity increases when fewer measurements are used; 167 (blue), 28 (red), 21 (yellow), and 17 (purple).
than the fully sampled measurement, but among compressively sampled experiments S/N 
decreased with fewer measurements Though the signal intensity decreases as 
compression increases, the noise is consistently reduced by nearly a factor of 3 for all 
levels of compression compared to the fully sampled. Figure 5.14 shows the S/N for each 
level of compression at a waiting time of 1ps, which is representative of all waiting times. 
For ease of visualizing, the values are shown as a function of compression factor rather 
than number of measurements. We calculate the S/N by calculating the RMS noise of a 
spectral area enclosed between $\omega_1 = 2208 - 2478$ cm$^{-1}$ and the entire $\omega_3$ axis (1880 - 
2246 cm$^{-1}$), and we use the difference in amplitude of the oppositely signed peaks of the 
higher-frequency diagonal peak as the measure of the signal. The optimum S/N is 
obtained at 6× (28 measurements) and 10× (17 measurements). We see a dip in the 
signal at 8× (21 measurements). This could be an artifact from the random nature of the 
sampling, though random sampling has been shown to be consistently more accurate in 
reconstructions than other nonuniform sampling patterns. A more likely explanation is 
that this is real compressively sampled data, and there was an inaccuracy in the data 
collection that would have been eliminated by averaging.

We also examine the peak center positions along the compressed axis, $\omega_{pump}$, as 
a function of the number of measurements. We do not show the peak position along the 
$\omega_{probe}$ axis because it is not compressed and sees no change. There is error in 
reproducing the center frequency of all four peaks, however there is a trend for the two 
low-frequency peaks and a different trend for the two high-frequency peaks. 
Interestingly, the low-frequency peaks exhibit a smaller error in peak position at the 
fewest measurements than at the most compressively sampled measurements. 28
measurements red shifts the low peaks by 1 cm\(^{-1}\) while 17 measurements blue shifts by almost 2 cm\(^{-1}\). The high peaks are significantly red shifted at the fewest measurements. Overall, the peak positions are consistent as a function of waiting time.

2D IR spectroscopy of rhodium dicarbonyl in dichloromethane is a demonstration of the possibility of compressive sampling and the GIRAF algorithm on a system with multiple peaks. Our analysis shows that with as few as 17 measured delay times we obtain accurate peak amplitudes, relative amplitudes, rate of peak change, and better S/N. This is 10× compression relative to the fully sampled frequency shifted data collection,
and a total of 78× fewer points acquired than fully sampled. More important to the researcher than compression level is the total number of laser shots and the time required to collect them. With 4-pulse phase cycling and our 1kHz repetition rate laser, a single spectrum was collected in 68 milliseconds.
6. CONCLUSIONS AND FUTURE WORK

We demonstrate that compressive sampling is a powerful technique to reduce the long data acquisition times of 2D IR, and the GIRAF algorithm accurately reconstructs peak positions and spectral lineshapes, even on a system, such as rhodium dicarbonyl, with multiple peaks. We also demonstrate that optimizing for the highest frequency shift first allows for the best reconstruction from the fewest number of measurements.

The formate dehydrogenase data presented in this thesis required the collection of approximately 500 million laser shots. With a 1 kHz repetition rate laser, this is just under six days. In addition to the laser shots, there is added laboratory time due to the fact that the system is not stable over the course of days. Pausing data collection to stabilize the system and restarting collection means we acquire the full data set in about 8-10 days of laboratory time.

There are many factors that contribute to reducing acquisition times with compressive sampling. The first is actually compressively sampling the tau time delay to collect fewer points. We show that an undersampling factor of $4 \times$ is adequate even for difficult systems. The second way compressive sampling reduces acquisition times is by increasing the SNR with the GIRAF algorithm. At a factor of $4 \times$ compression, the SNR was $3 \times$ larger than the fully sampled spectrum. Since SNR scales as the square root of the number of measurements, we can signal average $10 \times$ less than we do currently and
acquire similar SNR. These two factors alone mean that the entire data set that takes over a week to collect can now be acquired in 3.5 hours.

The laser system and signal is stable on the timescale of hours, and the additional time allotted for pausing for instability issues is no longer a concern. Additionally, with the data collected over a period of hours rather than days, the long-term drift of the laser and the changes in signal due to occasionally resetting the system are irrelevant. This means the 20 – 25 replicates currently collected could also be reduced.

Once exciting area of future studies is compressively sampling the waiting time. This requires re-engineering the GIRAF algorithm for higher dimensional reconstruction, a nontrivial task. A different set of random tau times will be programmed for each different randomly selected waiting time. Using the pulse shaper to move the two pump pulses in time relative to the probe pulse we can change the waiting time on every laser shot without moving a stage. Using similar sampling patterns on the waiting times, that is, biased toward early times, we will sample more time combinations with larger signal and thus the mean amount of signal averaging will be reduced. By compressively sampling waiting times by a similar factor as tau times, the full data sets shown here may be acquired in less than one hour.

There are currently 100 kHz repetition rate lasers on the market, with comparable pulse powers. Currently, the only barrier to one of these laser systems is price. Using a 100 kHz repetition rate laser the spectroscopist may well collect a week’s worth of data in less than one minute.

Another area of future work is using GIRAF as a denoising technique without the application of compressive sampling. We show that spectra reconstructed by GIRAF
showed a higher SNR than fully sampled spectra, because the algorithm does not model the noise. Applying the reconstruction algorithm to fully sampled data may be a way to further reduce the noise and increase the SNR.

2D IR is heavy in data yet rich in information. We conclude that compressive sampling is the next step forward in the evolution of two-dimensional infrared spectroscopy.
A. APPENDIX: DATA COLLECTION CODES

The 2D IR Data Collection Code in LabVIEW

The fully sampled 2D IR data collection code has been modified to implement compressive sampling as a toggle switch option. When compressive sampling is turned on, the user selects the desired level of compression in addition to the Start, End, and Interval scan parameters. The Generate Data Collection Masks button generates the pseudorandom sampling pattern of time delays, saves the mask in the user-defined folder location, and creates an array of pulse shapes for the selected times. The Load Data Collection Shapes button loads the generated masks into the arbitrary waveform generator buffer. The Collect 2D Data button starts data collection.

The compressive sampling 2D IR code has some practical differences, as well. The traditional data collection code performs all the data processing to generate the 2D spectrum, including phase cycling, apodization, zero padding and Fourier transformation, and presents the user with updated contour plots of the spectrum and running average. The compressive sampling code performs only the phase cycling processing and all other processing is off-loaded to the reconstruction code of the compressively sampled time-frequency data. As a result, LabVIEW does not calculate the spectrum and the user cannot observe a recently collected 2D IR spectrum within LabVIEW.
All other controls necessary to align, calibrate and collect data is consistent with the traditional code.

When collected data is saved, the output files are the tvsw data, sampling mask, and array calibration file. The first two files are necessary inputs to the reconstruction code, and the latter file is necessary to generate the axes of the reconstructed spectrum.

Next, we will walk through the operation of the GIRAF reconstruction algorithm.

Operating the GIRAF Reconstruction Algorithm

The GIRAF algorithm solves the two-dimensional reconstruction in the time domain. Data is collected with one dimension in time (tau time delays) and one dimension in frequency (off the detector), so an inverse Fourier transform is performed on the frequency dimension. However, there is actually no laser light hitting the pixels other than bandwidth of the beam in the middle, and in the phase cycling processing of the data in and the conversion from change in intensity to \( \Delta O D \), bands of noise are always obtained near the ends of the \( \omega_3 \) axis. The bands must be removed before the inverse Fourier transform. To determine the spectral bands to truncate one must plot the data and observe the best location for truncation. Within MATLAB, navigate to the file location and load the file. MATLAB reads a 2-D matrix as (rows x columns) and so once loaded the user must transpose the data to attain the correct axes orientation. Plot the data.

\[
\text{>> data} = \text{load('tvswA0.dat')}; \quad \text{%loads file, apostrophe means transpose, variable named data}
\]
\[
\text{>> figure; contour(data); } \text{%plot full data}
\]
Evaluate the contour plot and determine the appropriate indices to truncate. The resulting vector after truncation must be of even numbered length. Create the variables \( w3\text{Trunc\_low} \) and \( w3\text{Trunc\_high} \) to remove the bands and plot the truncated data to verify that the peak information is still present.

\[
> w3\text{Trunc\_low} = 250;
> w3\text{Trunc\_high} = 701;
> \text{figure; contour(data(w3\text{Trunc\_low:}w3\text{Truc\_high,:});}\%plot truncated data}
\]

Figure A.1 shows the figures resulting from the lines of code above for a sample of rhodium dicarbonyl compressively sampled from 21 points, and generates the two figures showing the time vs frequency data, and the truncated data.

**Figure A.1 Example of Truncated tvsw Data**

Matlab figures showing the necessity of truncation before taking the inverse Fourier transform along the probe axis. Shown is data compressively sampled from 21 random measurement of rhodium dicarbonyl. The figure on the left is the full data (1024 points along probe) and on the right is the truncated data (452 points along probe). Both contour plots are autoscaled to the data.

The values of the low and high truncation indices must be set in the GIRAF code before reconstruction.
Additional user-defined parameters are placed in the early lines of the code. These include the lambda0 value, undersampling factor, number of iterations, and filter size. The size of the zero-padded data, size of the final spectrum, and number of fully sampled tau times and regular step size must be set as well. Once set, these parameters are not often changed. The GIRAF code reads in the designated data file and corresponding sampling mask, and performs the reconstruction. After the code runs, the final reconstructed spectrum, the lambda0 value, alphanumeric value identifying the trial and waiting time, the undersampling factor, and the number of iterations are all saved in a .mat file in the current folder.

It is important to note that the calibration from pixel number on the camera to $\omega_3$ frequency axis is slightly nonlinear and must be linearized before the inverse Fourier transform. Two subcodes called CreateAxesAll.m and LinearGriddedData.m are supporting files that interpolate the data onto a linear grid and have been integrated into the code.

Reconstruction on the Cluster

When enough data files to reconstruct multiple spectra have been collected, the GIRAF reconstruction algorithm is best performed on the high-performance computing cluster operated by the University of Iowa, called Argon. Using this cluster, we take advantage of high throughput computing.

A cluster of compute nodes is a group of connected computers, called compute nodes, acting as a single system to solve large computational problems. This cluster of
nodes is used to solve problems that cannot be solved on a single processor (due to memory or CPU availability) or solves complex problems in a much shorter time.

The cluster can run both high performance computing (HPC) and high throughput computing (HTC) jobs. Basically, HPC enables a user to solve a single, large problem, and HTC enables a user to solve multiple smaller problems at the same time. Because the calculation of each spectrum is independent of all others, we use HTC on the cluster to reconstruct spectra from several data files, up to thousands, as fast as possible. We do this by submitting each calculation to a different node to run independently. If one calculation takes ten minutes, thousands of calculations only takes ten minutes, in the best-case scenario.

We automate the process of submitting thousands of jobs, with only one varying input, by creating copies of algorithms with a single variable substituted. Start with an initial MATLAB file, **GIRAF.m**, that has all the optimal parameters and file paths. Using the Linux utility sed (stream editor) copies of **GIRAF.m** are generated, each pointing to a different raw data file from which to reconstruct, and renamed with only an index number differentiation. A structure containing a list of all of the files is created, called **thetvswFiles**. The variable **currentFile** directs the code to load the file at a certain index within the structure **thetvswFiles**. The Linux utility sed is used to change the value of **currentFile** and rename and save the new file with suffix number. **currentFile** is initially set to unity, and the following shell script, named **changedata.sh**, changes the value on each iteration.

```bash
#!/bin/bash
for i in {1..10}
done
```
This script contains a for loop from 1 to 10, that for each iteration changes the variable `currentFile` in the file GIRAF.m from a value of 1 to a value of the iteration number, $i$, where $i$ refers to the iteration number. After this script is run by typing `sh changedata.sh` in the command line, the files GIRAF_job1.m through GIRAF_job10.m are generated. Each file will load and reconstruct a different data file.

We write a single shell script to submit the job to the cluster that references the index number within the script with the variable `SGE_TASK_ID`. Thus, with only one line of code, thousands of jobs can be submitted simultaneously and ran concurrently. A shell script is written to instruct the cluster to perform the desired calculations by opening and running the desired MATLAB file. The following script for submitting an array of jobs is named `GIRAF.sh`. A comment is indicated by `#` and instructions and code are indicated by `#$`. The job is named Recon, and all output files will be saved in the same folder as the reconstruction code. The script sends the job to the all.q queue. There are no limits on the number of jobs a user can run on the all.q, however a job may be kicked off if it is running on a dedicated node and a prioritized user asks to use it. A consequence of this is that some reconstructions may get killed and not finish. Having access to a dedicated group node would eliminate this problem, or devising a way to automatically restart any job that gets terminated. The script sets the number of slots the job will use, and loads the MATLAB module. The final section of the script sets the variable `SGE_TASK_ID` and calls the MATLAB file to run, with the index set in the job submission.
Determine the optimal value of the parameter lambda0

It is important to determine the optimal values of parameters in the GIRAF code. Large difference in the reconstructed spectrum can result from poor parameterization.
However, without the fully sampled spectrum to compare, a metric independent of the fully sampled result is needed to determine the optimal values. Of course, the point of using compressive sampling is to avoid fully sampling.

The code, in a basic concept, solves two terms corresponding to data consistency and regularization. The regularization term is related to how well the data fits the model. The parameter $\lambda_0$ controls the low-rankness of the solution, essentially putting different weights on the two terms. The correct parameterization of $\lambda_0$ is important to the outcome of the reconstruction.

Starting from a single compressively sampled data file, we perform the series of reconstructions by submitting an array of jobs only differing in their $\lambda_0$ values. This is done following essentially the same method described in the previous section, generating multiple files only different in the value of the variable $\lambda_0$ and submitting multiple jobs at once. The independent method to determine the optimal value of the parameter $\lambda_0$ compares the two terms, data consistency and regularization, at different values of $\lambda_0$. In a method we call the Two Terms Analysis, the two terms for each different $\lambda_0$ are plotted on a set of axes and the distance from the origin is calculated.

This is performed by running the MATLAB code called `FindOptimalLambda0.m`. The user directs the code to a folder containing the raw tvsw data from LabVIEW. The mask file must be in the same folder as the data. Next, the user locates the folder containing the reconstruction files, which have been reconstructed on the cluster and have been named to include the undersampling factor, waiting time, and $\lambda_0$ value in the name. One last dialog box asks the user for the undersampling factor, waiting time, and
the number of fully sampled tau time without compression. These inputs are all used to locate the correct files to evaluate. The code then automatically finds every file with a different $\lambda_0$ value according to the inputs and evaluates the Two Terms Analysis.

To determine the effectiveness of this method of optimization, we compare the series of reconstructed spectra testing different values of $\lambda_0$ to the fully sampled spectrum by root mean square error (RMSE). Figure A.2 shows an example of the $\lambda_0$ parameterization of azide in D$_2$O at a waiting time of $T = 0$ ps. The range of values of $\lambda_0$ that give the largest distance from the origin in the Two Terms Analysis correspond exactly with those that give smallest RMSE.

Figure A.2 Determining Optimal Lambda Value

For azide in D$_2$O at a waiting time of $T = 0$, the distance from the origin from the Two-Terms Analysis and the RMSE, comparing the reconstructed spectra to a fully sampled spectrum.

The plateau in the distance from the origin in the Two-Terms Analysis exactly matches the valley in the RMSE. We automate the process of finding the middle of the range in the code FindOptimalLambda0.m and give the user the opportunity to verify the analysis and automatically save the optimal value to the data folder.

In every case so far investigated, the optimal $\lambda_0$ value is consistent across
all waiting times for a given sample. It is important to perform the Lambda Test occasionally and anytime a new sample is used. However, GIRAF parameterization is independent of data collection, and one need not wait for it to collect data. Again, compressive sampling offers a great advantage in that all processing and computation occurs after data collection. All one needs are the time-frequency data and the sampling mask.

Once the optimal $\lambda_0$ value has been determined, the GIRAF reconstruction code should be set use that value.

Centerline Slope Analysis on the Cluster

At the end of the reconstruction code on the cluster, the generated spectrum is analyzed and the centerline slope is calculated and saved. Potentially thousands of spectra are reconstructed at once, and it is best to continue the analysis of the centerline slope where they are, on the cluster. Once all reconstructions are complete, the MATLAB code CLS_Analysis_Cluster.m performs the full analysis of the centerline slope. The code finds and loads the needed variables, the CLS value and the corresponding waiting time, from each reconstruction file. We order and reorganize the data by waiting time to inspect all CLS values for a given waiting time, then inspect and check for outliers. We remove any data point that is outside the 95% confidence level ($\pm 1.96 \times$ standard deviation) of all other values for a given waiting time. The number of points removed is also tracked. Then we calculate the average CLS value for each waiting time, along with the standard error of the mean (SEM) and the number of trials averaged.
The full CLS analysis includes fitting the CLS vs waiting time curve to a model equation. The functional form of the fit and the initial guesses for the parameters are controlled in the MATLAB code createCLSFit.m. Finally, we save the CLS Report, including the values of the fit parameters, including the 95% confidence interval, and the goodness of fit statistics, and a single organized matrix containing all CLS and waiting time data.
REFERENCES

melanoma cell line identification by FTIR imaging after formalin-fixation and paraffin-embedding. *Analyst 2013*, 138 (14), 4083-4091.


53. https://wiki.uiowa.edu/display/hpcdocs/Cluster+Systems+Documentation