ENHANCED DATA REDUCTION, SEGMENTATION, AND SPATIAL MULTIPLEXING METHODS FOR HYPERSPECTRAL IMAGING

LEANNA N. ERGIN

Bachelor of Forensic Chemistry
Ohio University
June 2006

Submitted in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY IN BIOANALYTICAL CHEMISTRY at the CLEVELAND STATE UNIVERSITY
July 19th, 2017
We hereby approve this dissertation for

Leanna N. Ergin

Candidate for the Doctor of Philosophy in Clinical-Bioanalytical Chemistry degree for

the Department of Chemistry and CLEVELAND STATE UNIVERSITY

College of Graduate Studies

Dissertation Committee Chairperson, Dr. John F. Turner II

Dissertation Committee Member, Dr. David W. Ball

Dissertation Committee Member, Dr. Petru S. Fodor

Dissertation Committee Member, Dr. Xue-Long Sun

Dissertation Committee Member, Dr. Yan Xu

Dissertation Committee Member, Dr. Aimin Zhou

Date of Defense: July 19th, 2017
Dedicated to

my husband, Can Ergin.
ACKNOWLEDGEMENT

I would like to thank my advisor, Dr. John F. Turner, II for his support and giving me the opportunity to work in the Advanced Chemical Imaging Facility at Cleveland State University. His guidance, enthusiasm, and intellect enabled me to learn more than I thought I would be able to learn.

I would like to express my gratitude to my committee members for their support and suggestions on my research. Over the years, Dr. David W. Ball, Dr. Yan Xu, Dr. Xue-Long Sun, Dr. Aimin Zhou, and Dr. Petru S. Fodor all had helpful suggestions and feedback about my research. A special thanks is deserved by Dr. Anne O’Connor for her support in my teaching and for her friendship.

The support of members of my research group was also invaluable. Venkat Bobba, Rajesh Morampudi, and Ajaykumar Zalavadia were a pleasure to work with and provided me with knowledge about a lot of the instrumentation in the lab. I wish everyone in the group success in the careers and in their lives.

My husband, Can Ergin, gave me unending support during my graduate school process. He stood by my side throughout the process and enabled me to succeed. I will be eternally grateful.

Lastly, thank you to the Cleveland State University Doctoral Dissertation Research Award Program for granting me funds to carry out my research.
A hyperspectral image is a dataset consisting of both spectra and spatial information. It can be thought of either as a full spectrum taken at many pixel locations on a sample or many images of the same sample, each at a different wavelength. In recent decades, hyperspectral imaging has become a routine analytical method due to rapid advances in instrumentation and technique. Advances such as the speed of data acquisition, improved signal-to-noise-ratio, improved spatial resolution, and miniaturization of the instrumentation have all occurred, making chemical imaging methods more robust and more widely used.

The work presented here deals with three issues in the field of hyperspectral imaging: unassisted data processing that is chemically meaningful and allows for subsequent chemometric analyses, visualization of the data that utilizes the full colorspace of modern red, green, blue (RGB) displays, and data collection with improved signal-to-noise ratios and comparably short acquisition times.
Hyperspectral image data processing is a fundamental challenge in the field. There is a need for reliable processing techniques that can operate on the large amount of data in a hyperspectral image dataset. Because of the large quantity of data, currently-used methods for data processing are problematic because of how time-consuming and calculation-intensive they are or because of increased error that is observed in the less-intensive methods. The work presented here includes a user-unassisted method for rapidly generating chemical-based image contrast from hyperspectral image data. Our method, reduction of spectral images (ROSI), is an effective hyperspectral image processing method. A full theoretical description of the method is given along with performance metrics. The description has been generalized to work with any number of wavelength dimensions and spectra. A concise protocol is put forth that will enable other researchers to utilize this method by following a short, simple list of steps.

ROSI can also be used as a data reduction method, as it achieves a threshold information density in the spectral dimension for all image pixels. ROSI results are suitable for subsequent data analysis enabling ROSI to be performed alone or as a preprocessing data reduction step.

This research also improves upon a spatially-multiplexed Raman imaging system based on the digital micromirror device (DMD). The system provides signal-to-noise ratio enhancement while maintaining laser powers below the damage threshold of the sample and comparably short acquisition times. In the work presented here, the spatial resolution of the DMD imager has been improved such that features with a width of 2.19μm could be resolved, whereas the previous limit was 7.81μm.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>xvi</td>
</tr>
<tr>
<td>CHAPTER I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Significance</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Hyperspectral Imaging</td>
<td>6</td>
</tr>
<tr>
<td>1.2.1 Spectroscopy</td>
<td>6</td>
</tr>
<tr>
<td>1.2.2 Hyperspectral Images</td>
<td>6</td>
</tr>
<tr>
<td>1.2.3 Imaging Strategies</td>
<td>7</td>
</tr>
<tr>
<td>1.2.3.1 Point mapping</td>
<td>7</td>
</tr>
<tr>
<td>1.2.3.2 Line Scanning</td>
<td>9</td>
</tr>
<tr>
<td>1.2.3.3 Wide-Field Imaging</td>
<td>9</td>
</tr>
<tr>
<td>1.2.3.4 Hadamard Multiplexing</td>
<td>9</td>
</tr>
<tr>
<td>1.3 Hyperspectral Image Processing Methods</td>
<td>11</td>
</tr>
<tr>
<td>1.3.1 Artificial Neural Network</td>
<td>11</td>
</tr>
<tr>
<td>1.3.2 Machine Learning and Deep Learning Methods</td>
<td>12</td>
</tr>
<tr>
<td>1.3.2.1 Pattern Recognition</td>
<td>13</td>
</tr>
<tr>
<td>1.3.3 Factor Analysis (Data Reduction) Methods</td>
<td>14</td>
</tr>
<tr>
<td>1.3.3.1 Spectral Angle Methods and Statistical Correlation Methods</td>
<td>14</td>
</tr>
<tr>
<td>1.3.3.2 Distance Methods</td>
<td>15</td>
</tr>
</tbody>
</table>
1.3.3.3 Principal Component Analysis .............................. 16
1.3.4 Regression Methods .............................................. 16
  1.3.4.1 K-Means ......................................................... 16
  1.3.4.2 Fuzzy C-Means .................................................. 17
  1.3.4.3 Multiple Linear Regression .................................. 17
  1.3.4.4 Partial Least Squares ......................................... 18
  1.3.4.5 Principal Component Regression ............................. 18
1.3.5 End-Member Classification ..................................... 18
  1.3.5.1 Multivariate Curve Resolution .............................. 19
  1.3.5.2 Orthogonal Subspace Projection ............................. 19
  1.3.5.3 Classical Least Squares and Non-Negative Least Squares 20
1.3.6 Clustering Methods .............................................. 20
  1.3.6.1 Affinity Propagation .......................................... 20
  1.3.6.2 Virtual Dimensionality ........................................ 21
  1.3.6.3 Linear Discriminant Analysis and Soft Independent Modeling of Class Analogies ........................................... 21
1.3.7 Data Compression Methods .................................... 22
  1.3.7.1 Discrete Cosine Transform ................................... 22
  1.3.7.2 Burrows-Wheeler Transform ................................ 22
  1.3.7.3 Lempel-Ziv 1977 and Lempel-Ziv 1978 ................... 23
REFERENCES ................................................................. 24

CHAPTER II. REDUCTION OF SPECTRAL IMAGES FOR HYPERSPECTRAL IMAGE ANALYSIS: A VISUALIZATION METHOD ................. 31
2.1 Significance

2.2 Methodology of Reduction of Spectral Images

2.2.1 User Input of Desired Reliability Threshold

2.2.2 Casting of the Spectra into n-Dimensional Space

2.2.3 Conversion of the Data to Direction Cosine Space

2.2.4 Mean Centering of the Dataset

2.2.5 Selection of Reference Vectors Using Principal Component Analysis

2.2.6 Determination of Cosine Correlation Scores for All Spectral Vectors with Respect to the PCA Reference Vectors

2.2.7 Deciding Which Eigenvectors to Retain

2.2.8 Categorization to Determine Order of Principal Components

2.2.9 Visualization of the Scores Using a Colormap

2.3 Methodology of Principal Component Analysis

2.3.1 The Scale Variant Problem in PCA

2.4 Methodology of Classical Least Squares

2.5 Results of ROSI as a Hyperspectral Imaging Method

2.5.1 Concrete Raman Dataset

2.5.2 Piano Dataset

2.5.3 Five Component Filled Molar Fraction Space Dataset

2.6 Conclusion

REFERENCES

CHAPTER III. UNASSISTED REDUCTION AND SEGMENTATION OF LARGE HYPERSONTRICAL IMAGE DATASETS
3.1 Significance

3.2 Methodology of ROSI as a Data Reduction Method

3.2.1 Casting of the Spectra into n-Dimensional Space and Recording Amplitudes

3.2.2 Conversion of the Data to Direction Cosine Space

3.2.3 Mean Center the Dataset and Record the Amplitudes

3.2.4 Selection of Reference Vectors Using Principal Component Analysis

3.2.5 Determination of Cosine Correlation Scores for All Spectral Vectors with Respect to the PCA Reference Vectors

3.2.6 Reliability Threshold and Deciding Which Eigenvectors to Retain

3.2.7 Data Compression

3.2.8 Recalculating the Original Vectors

3.3 Methodology of PCA as a Data Reduction Method

3.4 Results of ROSI as a Data Reduction Method

3.4.1 Four Component Raman Point Mapping Dataset

3.4.2 Four Component Raman Filled Molar Fraction Space Dataset

3.5 Conclusion

REFERENCES

CHAPTER IV. IMPROVEMENT OF A SPATIALLY-MULTIPLEXED RAMAN HYPERSPECTRAL IMAGING SYSTEM

4.1 Significance

4.2 Multiplexing

4.2.1 Cyclic S-Matrix Multiplexing
4.3 Digital Micromirror Device................................................................. 129
4.4 Multiplexed Hadamard Transform Raman Hyperspectral Imaging System...... 131
  4.4.1 The Need for a Manhattan to Diamond Conversion Technique............. 134
  4.4.2 Original Manhattan to Diamond Conversion Technique: The 3x6 Super-Pixel Approach................................................................. 135
  4.4.3 Manhattan to Diamond Conversion Technique with Improved Resolution: The Coordinate Conversion Approach......................... 137
4.5 Results................................................................................................. 141
4.6 Conclusion......................................................................................... 145
REFERENCES............................................................................................ 147
APPENDIX I. REDUCTION OF SPECTRAL IMAGES (ROSI) MATLAB CODE.. 149
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Tables</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Variables Needed for Data Recalculation</td>
<td>98</td>
</tr>
<tr>
<td>2. Bytes of Variables Needed for Data Recalculation</td>
<td>105</td>
</tr>
<tr>
<td>3. Percentage of Original Data Size at Various Reliability Thresholds and Signal-to-Noise Ratios (SNR)</td>
<td>111</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The Concept of a Hyperspectral Image Data Cube</td>
<td>3</td>
</tr>
<tr>
<td>2. Imaging Strategies</td>
<td>8</td>
</tr>
<tr>
<td>3. Casting of the Spectra into n-Dimensional Space and Conversion of the Data to Direction Cosine Space</td>
<td>39</td>
</tr>
<tr>
<td>4. Mean Centering the Data</td>
<td>43</td>
</tr>
<tr>
<td>5. Selection of Reference Vectors using PCA and Determination of Cosine Correlation Scores with Respect to PCA Vectors</td>
<td>47</td>
</tr>
<tr>
<td>6. Colormapping Procedure</td>
<td>53</td>
</tr>
<tr>
<td>7. The Scale Variant Problem in PCA</td>
<td>57</td>
</tr>
<tr>
<td>8. Brightfield Image and Selected Spectra of the Concrete Dataset</td>
<td>60</td>
</tr>
<tr>
<td>9. ROSI Images of the Concrete Dataset</td>
<td>62</td>
</tr>
<tr>
<td>10. All ROSI Images of Concrete Dataset until Desired Reliability Threshold is Reached</td>
<td>63</td>
</tr>
<tr>
<td>11. Resultant Images of Concrete using Other Methods</td>
<td>64</td>
</tr>
<tr>
<td>12. Five Components of Piano Image and Tetrahedral Datasets</td>
<td>66</td>
</tr>
<tr>
<td>13. The Molar Fraction Masks for the Five Spectra in the Piano Dataset</td>
<td>67</td>
</tr>
<tr>
<td>14. ROSI Images of the Piano Dataset</td>
<td>69</td>
</tr>
<tr>
<td>15. Resultant Images of the Piano Dataset using Other Methods</td>
<td>70</td>
</tr>
<tr>
<td>16. ROSI Resultant Image of the Five Component Filled Molar Space Dataset</td>
<td>73</td>
</tr>
<tr>
<td>17. Resultant Image of Other Methods for the Five Component Filled Molar Space Dataset</td>
<td>74</td>
</tr>
</tbody>
</table>
18. Comparison of the Spectra which were assigned the Same Color in the ROSI Resultant Image .......................................................................................................................... 76

19. Comparison of the Spectra which were assigned the Same Color in the PCA Resultant Image ..................................................................................................................... 77

20. Flow Chart for ROSI Methodology .............................................................................................................................................................................................. 89

21. Brightfield Image of the Four Component Raman Point Mapping Dataset and the Known Chemistries by Pixel Location ......................................................................... 103

22. Visualization of ROSI Result ............................................................................................................................................................................................ 107

23. Visualization of ROSI Result until the Reliability Threshold of 0.954 is Reached ........................................................................................................................................ 108

24. Visualization of PCA Result ............................................................................................................................................................................................ 109

25. Percentage of Original Data Size at Various Reliability Thresholds and Signal-to-Noise Ratios ..................................................................................................................................... 112

26. Visualization of ROSI Results of the Four Component Point-Mapping Dataset .................................................................................................................. 114

27. A Matrix Configuration Showing a 7x7 Circulant S-Matrix ..................................................................................................................................................... 126

28. DMD Mirrors ........................................................................................................................................................................................................... 130

29. Diamond Pixel Projection and Manhattan Pixel Projection ........................................................................................................................................ 132

30. Spatially-Multiplexed Raman Hyperspectral Imager Instrument Setup ......................................................................................................................... 133

31. Nine Super-Pixels as they appear in Manhattan and Diamond Pixel Projection Using the 3x6 Super-Pixel Conversion Technique ........................................................................... 136

32. The Manhattan to Diamond Pixel Projection Process ............................................................................................................................................ 140

33. A Brightfield Image of the USAF 1951 Resolution Target ............................................................................................................................................. 142
34. A 4095 Pixel Image of the USAF 1951 Resolution Target Using the 3x6 Super-Pixel Approach and the Brightfield image of the Resolution Target Cropped to Match the Hyperspectral Image........................................................................................................ 143

35. A 4095 Pixel Image of the USAF 1951 Resolution Target Using the Coordinate Conversion Approach and the Brightfield image of the Resolution Target Cropped to Match the Hyperspectral Image........................................................................................................ 144
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>AP</td>
<td>Affinity Propagation</td>
</tr>
<tr>
<td>BWT</td>
<td>Burrows-Wheeler Transform</td>
</tr>
<tr>
<td>CCA</td>
<td>Cosine Correlation Analysis</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge Coupled Device</td>
</tr>
<tr>
<td>CLS</td>
<td>Classical Least Squares</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DMD</td>
<td>Digital Micromirror Device</td>
</tr>
<tr>
<td>FCM</td>
<td>Fuzzy C-Means</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>LZ77</td>
<td>Lempel-Ziv 1977</td>
</tr>
<tr>
<td>LZ78</td>
<td>Lempel-Ziv 1978</td>
</tr>
<tr>
<td>MCR</td>
<td>Multivariate Curve Resolution</td>
</tr>
<tr>
<td>MLR</td>
<td>Multiple Linear Regression</td>
</tr>
<tr>
<td>NNLS</td>
<td>Non-Negative Least Squares</td>
</tr>
<tr>
<td>OLS</td>
<td>Ordinary Least Squares</td>
</tr>
<tr>
<td>OSP</td>
<td>Orthogonal Subspace Projection</td>
</tr>
<tr>
<td>PC</td>
<td>Principal Component</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Components Analysis</td>
</tr>
<tr>
<td>PCR</td>
<td>Principal Component Regression</td>
</tr>
<tr>
<td>PLS</td>
<td>Partial Least Squares</td>
</tr>
<tr>
<td>PR</td>
<td>Pattern Recognition</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green, Blue</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>ROSI</td>
<td>Reduction of Spectral Images</td>
</tr>
<tr>
<td>SAM</td>
<td>Spectral Angle Mapping</td>
</tr>
<tr>
<td>SIM</td>
<td>Spectral Identity Mapping</td>
</tr>
<tr>
<td>SIMCA</td>
<td>Soft Independent Modeling of Class Analogies</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

In 1666, Sir Isaac Newton studied the rainbow effect and coined the term ‘spectrum.’ He showed that sunlight is composed of a spectrum of colors, which each refract at different angles when passed through a prism. He also showed that the colors of light could be combined back into white light by passing the rainbow through another prism.\(^1\) This was an early experiment in spectroscopy. Since that time, there have been numerous advances in the field of spectroscopy that have enabled it to become both a qualitative and quantitative tool for studying the chemistries of substances based on the interaction of light and matter. A spectrum is a plot of the intensities of electromagnetic radiation that are absorbed, emitted, or scattered by a sample versus a quantity related to photon energy, such as wavelength or frequency.\(^2\) Differences in the absorption, emission, and scattering of light that interacts with matter can be measured and analyzed to draw conclusions about the chemical composition of the sample.

A hyperspectral image is a dataset consisting of spectra taken at different locations on a sample. It can also be thought of as many images of the same sample, each
at a different wavelength. For each pixel location in the field of view, a full spectrum is recorded. They are three dimensional data cubes, with two spatial dimensions and one wavelength dimension. Figure 1 shows a depiction of a hyperspectral image data cube in which a spectrum has been acquired at each pixel location.

In recent decades, hyperspectral imaging has become a routine analytical method due to rapid advances in instrumentation and technique. Advances such as the speed of data acquisition, improved signal-to-noise-ratio, improved spatial resolution, and miniaturization of the instrumentation have all occurred, making chemical imaging methods more robust and more widely used.

The development of effective hyperspectral image processing methods is a fundamental challenge in the field. There is a need for reliable processing techniques that can operate on the large amount of data in a hyperspectral image dataset. Because of the large quantity of data, currently-used methods for data processing are time-consuming, calculation-intensive, or impart either qualitative or quantitative errors.

Conventional reduction techniques like those based on principal components analysis (PCA) sacrifice the contributions of minority pixel populations while retaining those representing a greater portion of the overall variability. The effect is that some regions in the reconstructed images achieve a higher degree of recovery than other locations, making it difficult to assess the meaning or relevance of the minority pixels, even when this information would reveal important sample defects or spectral inhomogeneities.

In this research, we have developed a method called reduction of spectral images
Figure 1. The Concept of a Hyperspectral Image Data Cube. A shows the spectra of two pixel locations, while B shows the concept of the hyperspectral image data cube, with two spatial dimensions, $x$ and $y$, and one wavelength dimension, $\lambda$. 
(ROSI), which is effective in differentiating spectral shapes at different locations in the hyperspectral image data. The method removes the degeneracies caused by angular symmetry, can be done rapidly, requires no training datasets or a priori information about the data, and takes advantage of the full colorspace that modern RGB displays can handle when displaying the resultant image.

1.1 Significance

Any type of instrumentation which can gather a spectrum can be equipped to gather hyperspectral images by collecting a spectrum at each pixel location on a sample. Datasets may contain Raman scattering, infrared adsorption, fluorescence emission, near infrared absorption and reflectance, ultraviolet-visible absorption and reflectance, or any other type of spectra. This means that hyperspectral imaging is used on samples which are vastly different from one another, such ancient paintings, pharmaceuticals, cancer detection, and remote sensing of Earth.

The method that is presented, the reduction of spectral images (ROSI), is a fully-automated unassisted technique that proves image segmentation and data classification based on the subtle changes in spectral shape. It is generalized to work with any number of wavelength dimensions and image pixel resolutions. ROSI is generalized to operate on any type of hyperspectral image, regardless of type of spectroscopy or type of sample. Many conventional data processing methods are not equipped for use with the large amount of data in a hyperspectral image and others provide analyses in either the spatial or spectral dimension in a way that does not fully utilize the information content of the data.
The fact that ROSI does not require \textit{a priori} information (training data) for the hyperspectral image is a great benefit to ROSI over many other methods because the manufacturing of a training dataset can be complex, or even impossible, for many sample types. To manufacture a training dataset, a user must have knowledge of the chemistries present in the sample. The user must gather a pure component spectrum of each of the chemical contributions in the sample. This may involve a library search for the needed spectra, but if the appropriate spectra are not found, samples of each chemical component must be obtained so that spectra can be gathered. A training dataset typically consists of both pure component spectra and a series of weighted averages of the pure component spectra to simulate mixtures. Each and every spectrum in the hyperspectral image is compared to the spectra in the training dataset. There are many different mathematical methods for comparing spectra, and as such, there are many different methods for post-processing hyperspectral images, many of which are described in this chapter.

If the hyperspectral image contains unexpected or unknown chemistries, a user would not know to include the pure component spectra of those chemistries in the training dataset, causing some of the data in the hyperspectral image to be overlooked. For example, in the quality control of pharmaceuticals, a contaminant might be overlooked because a pure component spectrum of the contaminant was not included in the training dataset. In a biological tissue sample, a user would need to select pure component spectra of every possible chemical component of the tissue sample. Such a task is beyond our current abilities. ROSI can successfully identify differences in chemistry based on subtle differences in the shape of the spectra in the dataset. Thus, ROSI is successful even when the dataset is complex or has unknown constituents.
1.2 Hyperspectral Imaging

1.2.1 Spectroscopy

A spectrum is a plot of the intensity of electromagnetic radiation that is emitted, absorbed, or scattered by a sample versus the wavelength or frequency of the radiation. The ways in which electromagnetic radiation interacts with matter yields spectra that can impart both qualitative and quantitative chemical information. The study of this interaction is called spectroscopy. The elementary particle which is quantified in spectroscopy is the photon, which also shares properties with waves. The energy of a photon can be related to its wavelength and frequency through the use of Plank’s constant, $h$ \((6.626 \times 10^{-34} \text{ J} \cdot \text{s})\), through

$$E = h \nu = \frac{hc}{\lambda} \quad (1)$$

where $E$ is energy in Joules (J), $\nu$ is frequency in $\text{s}^{-1}$, $c$ is the speed of light in a vacuum ($\sim 2.99 \times 10^8 \text{ m/s}$), and $\lambda$ is the wavelength (m). Rotational, vibrational, and electronic transition energies are probed using optical spectroscopic technologies. Hence, spectroscopy is a particularly important field in analytical chemistry.

1.2.2 Hyperspectral Images

Raman imaging approaches were first described in 1975 by Delhaye and Dhamelincourt.\(^{15}\) Although the charge coupled device (CCD) detector had been developed some five years earlier,\(^{16,17}\) it was not yet widely available, causing early use of the methodology to be limited. Advances in technology have now made hyperspectral imaging a routine analytical method. The focus of this research has been on the Raman
imaging modality, a type of vibrational spectroscopy in which the wavelength shift of inelastically scattered light corresponds to vibrational/rotational transitions in the molecule. Raman spectroscopy is chemically specific and requires little or no sample preparation. Consequently, its use in bioanalysis and material science continues to grow rapidly. Nevertheless, the technique is optically inefficient and would benefit from improved image collection strategies and enhanced multivariate analysis methods that do not rely on \textit{a priori} data (training data).

\subsection*{1.2.3 Imaging Strategies}

Various imaging strategies exist to gather the three dimensions of data in hyperspectral imaging. Of those that exist, the most commonly used for chemical imaging include spatial scanning approaches, spectral scanning approaches, and multiplexing. Spatial scanning approaches gather full spectra at once and build up the dataset pixel by pixel by rastering to new locations. Spectral scanning approaches gather a full image at one wavelength in one acquisition step and build up the dataset wavelength frame by wavelength frame. Multiplexing methods gather combined signals, which are mathematically separated to determine the spatial and spectral data. \textbf{Figure 2} shows a depiction of these four imaging strategies.

\subsubsection*{1.2.3.1 Point mapping}

Point mapping is a spatial scanning imaging strategy in which the detector, usually a spectrograph equipped with a linear array detector, is capable of collecting one spectrum at a time. The sample is moved in an $x,y$ raster pattern and a spectrum is
Figure 2. Imaging Strategies. Point mapping (A), line scanning (B), wide-field (C), and Spatial Multiplexing (D) are various imaging strategies for hyperspectral imaging.
collected at each \(x,y\) position (pixel location). Point mapping produces hyperspectral images and is simple to perform, but exhibits comparatively long sample acquisition times, as the sample collection time is mostly dependent on the number of pixels in the image.\(^{18,19}\)

### 1.2.3.2 Line Scanning

Line scanning, also called push-broom scanning, is another spatial scanning approach which was first demonstrated for Raman in 1990, by which time CCD detectors had become more widely available.\(^{19,20}\) It requires a linear detector array. Because a line of full spectra can be collected at once, the raster pattern needs only move in one spatial direction. At each position in the raster pattern, a full line of spectra is collected. Line scanning offers a dramatic reduction in sampling time compared to point mapping.

### 1.2.3.3 Wide-Field Imaging

In wide-field imaging, the intensities of all spatial points are gathered at once using an array detector, such as a CCD, for one wavelength band. The wavelength band is selected by use of a tunable wavelength filter that is capable of preserving image information in the light field upon which it acts. Wide-field imaging can provide high spatial resolution and shorter acquisition times for large pixel densities relative to the number of wavelengths.

### 1.2.3.4 Hadamard Multiplexing

While other multiplexing methods exist, Hadamard multiplexing is most commonly used for spatial multiplexing. In Hadamard spatial multiplexing, pixel
dimensions are arranged into groups and a combined spectrum is gathered for each group.
A Hadamard encoding scheme is used to determine how the combinations are assigned
and how to mathematically separate them into their individual wavelengths or pixels.
Several advantages exist over conventional techniques. The main advantage, known as
Fellgett’s advantage, is improved signal-to-noise ratios, brought about through the fact
that the signal is comparatively higher than the noise when grouped. The Jacquinot
advantage is the higher throughput of light that is allowed in the absence of slits or
apertures needed to achieve resolution. Connes advantage refers to the higher precision
that is achieved when no or few moving parts are used. Due to hardware design and
limitations, not all multiplexing schemes realize all three advantages.
1.3 Hyperspectral Image Processing Methods

The large amount of data in hyperspectral images requires processing methods to extract or find important characteristics in the data. An obstacle that researchers face is determining effective and efficient methods to accomplish this goal. A second goal is to reduce the storage space required to save the hyperspectral image, which requires reversible data compression methods. There are many unique hyperspectral image processing and compression methods. A few of the important ones are described here, albeit briefly, to help reveal the current state of the field.

1.3.1 Artificial Neural Network

The artificial neural network (ANN) method of processing hyperspectral image data was inspired by the way in which the central nervous system of animals operates. It consists of a series of discrete bundles of software code called “neurons” that perform calculations based on the recognition and classification of the data in the network. The neurons can be arranged in layers, with “pathways,” or orders of neuron operation, between them. Each neuron is weighted, or given a score for the importance of those calculations, such that the output of using this method gives the desired result. How correct or incorrect any given weight is depended upon the task at hand. A benefit of using ANNs is that they have the ability to process large amounts of data rapidly, as is needed in hyperspectral image analysis. However, ANNs need to be trained. The speed advantage is only for their application after training. Training can take a very long time and depends on the complexity of the system being studied.
There are numerous types of ANN architectures in existence, each having been developed for a specific task. It is important to choose the appropriate type of ANN architecture for the task at hand, thus the operator must be well-versed in the types of artificial neural networks. Another disadvantage of using an ANN is that while the input and the results are clearly seen, the process by which those results are achieved is a black box, in that the process is not always apparent.

1.3.2 Machine Learning and Deep Learning Methods

Machine learning, as its name suggests, is the manufacturing of computational systems that can learn from the data that they process. It is closely related to artificial neural networking in that training data with an anticipated outcome is operated upon within the neural network. The weighing values on each discrete software package are calculated such that the results given on training data come as close as possible to the anticipated or desired results. In general, the performance of tasks of a machine learning system improves through experience. The goal is to enable computers to perform as humans can perform with as little human interaction as possible. There are many categories of machine learning including supervised, unsupervised, semi-supervised, transduction, and reinforcement learning.

Deep learning is a technique for implementing machine learning. Deep learning is when a large ANN has been trained on thousands or even millions of datasets, the weighing values assigned to each neuron have been calculated so precisely that the anticipated result is almost always given. One example of its implementation is today’s
image recognition software: Software such as Deepdream, OpenCV, AForge.Vision, and Amazon Rekognition were trained mostly using the vast amount of data on the internet.

The reduction of spectral images, (ROSI), which will be discussed in detail in the following chapters, is an example of a software bundle which could be used as a neuron in an artificial neural network. The following methods in this chapter are other examples of “neurons” used in ANNs for the processing of hyperspectral image datasets.

1.3.2.1 Pattern Recognition

Pattern recognition (PR) is not one specific method, but rather, a general term that describes methods that recognize patterns in the spectra of hyperspectral image data. As such, this section describes a generalized example of PR. When PR is used to process hyperspectral images, the first step is for the operator to choose or obtain training spectra for each class or chemical composition of interest. Because PR works poorly with a very large number of wavelengths, both in the reliability of results and the computing power required to process the data, a dimensionality-reducing method that retains the discriminating features in the data is chosen and applied first. After this, the spectra are classified based on similarity to the training spectra, often by a method such as Gaussian maximum likelihood classification. Threshold values can be set to reject spectra that are not similar to any of the training spectra, so that they are not misclassified. Finally, maps are generated based on which training spectra is most likely matching the data spectra at each pixel location.24 The disadvantages of pattern recognition for use in hyperspectral image processing is that a priori information about the data is needed and that a skilled
operator needs to choose the training spectra and oversee data reduction, leaving room for error.

1.3.3 Factor Analysis (Data Reduction) Methods

Factor analysis methods are used to describe variability among measurements and characterize the data according to one or more factors called scores. A number of factor analysis methods exist which are very useful for qualitative analyses.

1.3.3.1 Spectral Angle Methods and Statistical Correlation Methods

A commonly used type of factor analysis is based on correlation. Correlation methods measure the similarity between each member of the spectral set and a known reference spectrum. Spectral angle mapping (SAM), cosine correlation analysis (CCA), and spectral identity mapping (SIM) are widely used correlation methods that provide rapid chemical-based image contrast in hyperspectral image data. Because correlation methods measure the similarity between each member of the spectral set and a reference spectrum, they can be employed to identify unknown samples against a database of known spectra. For a large number of spectra, such as those in hyperspectral image analyses, obtaining the same correlation score for two distinctly different spectra is probable and results in the same correlation score for different spectra.

In CCA, SIM, and SAM, the spectra are cast into \( n \)-dimensional wavelength space and scores are assigned to each spectrum based on the angle (in SAM) or cosine of the angle (in CCA and SIM) between the spectral vector and a reference spectrum.\(^{25-27}\) The problem with SAM and CCA is that any set of spectral vectors that share the same angle to the reference spectrum will yield identical scores, even if those spectral vectors are
dissimilar. A set of concentric hypercones with the reference vector lying along their axes of symmetry exist. Any vectors which lie on the same hypercone (have the same correlation to the reference vector) are indistinguishable. SIM is an effort to overcome the symmetry problems of CCA and SAM. SIM perturbs the data by point-wise multiplying each band in the spectrum with a factor taken from a perturbation vector, such as a sine wave or a cosine wave. The challenge with SIM is that no mathematical proof of symmetry removal has been found, although symmetry is removed.

1.3.3.2 Distance Methods

Euclidean distance and Mahalanobis distance are both statistical tools that yield the distance between two points. In the case of hyperspectral image analysis, those two points are the heads of vectors once they are cast into $n$-dimensional wavelength space. Euclidean distance is the ordinary distance used in day-to-day life as given by the Pythagorean Theorem. Euclidean distance is scale-variant, meaning that the results would change if all of the data were multiplied by a common constant. Mahalanobis distance is the relative distance from one data point to a reference point and it is, therefore, scale-invariant, meaning that the results would not change if all of the data were multiplied by a common constant. Mahalanobis distance is used to gauge probability of belonging to a spectrum of a group. Both of these methods would have similar symmetry problems to the spectral angle and correlation methods, in that some spectra are indistinguishable because they are the same distance to another vector, even if the distance is measured in different directions.
1.3.3.3 Principal Component Analysis

Principal component analysis (PCA) can be used to extract pseudo-pure component spectra from mixed spectra in a hyperspectral image. It is a method that reduces the dimensionality of the data. It recasts the data points into a subspace that is orthogonal to the greatest amount of variance. The axes in this orthogonal space are called principal components. A series of principal components are iteratively computed. The first principal component accounts for the most variation; the subsequent principal components account for progressively less variation, with the latter principal components accounting predominantly for noise. The latter (noisy) principal components are discarded. The problem with using this approach when processing a hyperspectral image is that it is both time-consuming in its iterative approach and largely distinguishes only predominant features in the spectra, ignoring some features that are important but that contribute negligibly to variability.

1.3.4 Regression Methods

Regression methods are used to estimate the variation in the data and to describe relationships between the dependent and independent variables. There are many types of regression operations. The set of regression methods is useful for quantitative analyses.

1.3.4.1 K-Means

K-means is a partial clustering method that usually uses Euclidean distance as a similarity measurement. Some variations of the algorithm use Mahalanobis distance or other similarity functions in place of Euclidean distance. K-means first chooses spectra that are cluster centers and then assigns each spectrum to the cluster that has the nearest cluster center using an algorithm. The method is usually an assisted method that relies on
trained individuals to identify the initial cluster centers. The cluster centers, once chosen, are repositioned automatically. All that is initially needed are the number of centers and their approximate positions. The disadvantage of using K-means is that the size of the clusters and the data variation within the clusters is difficult to control.  

1.3.4.2 Fuzzy C-Means

Fuzzy methods are a class of clustering algorithms in which the boundaries between classes are not strict, but are said to be “fuzzy.” Specifically, when fuzzy C-means (FCM) is used to process hyperspectral image data, each spectrum is given a set of scores that reflect how well the spectrum agrees with each cluster. A spectrum with a low agreement score to a particular cluster belongs to that cluster to a lesser degree than a spectrum that is closer to the cluster center.\textsuperscript{29,30} FCM has the advantage of being an unsupervised method, but has the same disadvantage as K-Means: The size of the clusters greatly depends on the distance or similarity measurement being used and the initial choice of the number of clusters. Vastly different results can be achieved with only slight variation in the initial choices, thus it is difficult to optimize the method.

1.3.4.3 Multiple Linear Regression

Multiple linear regression (MLR) seeks to explain the variation in the data by describing the data using fewer explanatory variables. If there are two datasets, x and y, MLR is used to predict the most probable distribution of dataset y, given a specific distribution of x. If the variation in y can be explained by the value of x, then x can be used to describe y. Rather than using all of the values in the dataset y to describe the data, only the value of x is needed. The problem with linear regression methods lies in the key assumptions that are made: For example, it is assumed in MLR that the variance
in the dependent variables is constant, regardless of the value of the independent variable. It is also assumed that the relationship between the dependent and independent variables is a linear relationship. This is sometimes mitigated by taking logarithms or derivatives of the data, but utilizing this pathway requires the user to determine how to make the relationship between variables linear.²¹

1.3.4.4 Partial Least Squares

Partial least squares (PLS) is a multiple linear regression-based method used to compare data matrices. PLS is used to predict the variable(s) of one matrix based on observing the variables of another matrix.²⁹ The major drawback of PLS is that a calibration step is necessary to find the mathematical relationship between the two matrices. This means that quantitative information needs to be available for this comparison.³²

1.3.4.5 Principal Component Regression

Principal component regression (PCR) is based on principal component analysis. After PCA recasts the vectors in a subspace that is orthogonal to the greatest amount of variance and discards those dimensions that account for lesser amounts of data variance, PCR is used to describe the variation in the data according to the non-discarded principal component axes. Like PCA, PCR discards some of the variation in the data and requires a large amount of computation time.³³

1.3.5 End-Member Classification

In end-member classification methods, the pure chemical components are referred to as end-members. These methods are used to estimate the pure components that
contribute to the entire set of spectral data as well as their abundances for each spectrum in the dataset. It’s assumed that each spectrum is a linear combination of the end-member spectra. These methods could also be referred to as spectral unmixing methods.\textsuperscript{34} As a whole, end-member classification methods have the disadvantage of not taking into account non-linear combinations of pure component spectra that are needed due to interferences in the spectra of mixtures or differences in the levels of reflectance or absorbance of the pure components.

1.3.5.1 Multivariate Curve Resolution

When applied to hyperspectral image processing, multivariate curve resolution (MCR) seeks to determine the pure chemical components that make up the spectra at each pixel location. Beer’s law states that the absorbance of light is proportional to the pathlength that the light travels through the sample times the concentration. It utilizes Beer’s Law and expands the equation to incorporate matrices. However, the first step of MCR is to guess the identity of a component or components. In MCR, matrices containing the collected spectra are compared with matrices containing the to-be-determined concentrations.\textsuperscript{35} The initial guess plays a large role in determining the quality of the results, which is the downside of the method.

1.3.5.2 Orthogonal Subspace Projection

Orthogonal subspace projection (OSP) operates by projecting each spectral vector into a space that is orthogonal to undesired spectra. It is a method that is well-suited for suppressing interferences, when those interferences are chosen as the undesired spectra. Once the interferences have been diminished, the optimized spectral vector is cast against the signal of interest. It results in a single image for each component of interest.\textsuperscript{36} The
disadvantage to this method is that \textit{a priori} information about the abundance of spectral signature data is needed in order to select undesired spectra. In most real world situations, this information is difficult to obtain.\textsuperscript{37}

\subsection*{1.3.5.3 Classical Least Squares Methods and Non-Negative Least Squares}

Least squares methods are regression methods that aim to minimize the sum of the squared error between the observed responses and the responses predicted by a linear determination. Classical least squares (CLS) is also known as ordinary least squares (OLS). In OLS, it is assumed that concentrations are a linear combination of pure component spectra. Thus each spectrum is a weighted sum of the pure component spectra. Those weights or abundances are the values that CLS seeks to estimate. Non-negative least squares (NNLS) has one main difference from CLS: It is assumed that no value in the spectra (pure component spectra or mixed spectra) can be negative.\textsuperscript{34,38}

Least squares regressions are important for quantitative analyses, but require \textit{a priori} training data.

\subsection*{1.3.6 Clustering Methods}

Clustering methods group the spectra of hyperspectral image data in groups (clusters) based on some similarity measurement or related criteria. Ideally, the resultant clusters contain spectra that exhibit the greatest chemical similarities to each other.

\subsubsection*{1.3.6.1 Affinity Propagation}

Affinity propagation (AP) is a clustering method in which similarity between pairs of data points is initially input. From that input, representative spectral “exemplars” are chosen such that a broad range of the data is represented in the exemplar spectra. An
iterative process is performed that minimizes the Euclidean distance between spectra and exemplars until cluster centers and data clusters emerge. One deficiency of AP is that it is not well-suited for datasets with a large spectral variety, such as when multiple subclasses of spectra are present. The method uses one exemplar to define each class. When subclasses are present that need to be distinguished, AP shows poor performance.

1.3.6.2 Virtual Dimensionality

Virtual dimensionality was introduced in 2004 as a method to estimate the number of distinct spectra that characterize a hyperspectral image. It is not a stand-alone method, but is used to categorize the spectra prior to principal component regression. Based on computer models of the method, the effectiveness of virtual dimensionality decreases with non-identically distributed noise, as would be present in most real datasets.

1.3.6.3 Linear Discriminant Analysis and Soft Independent Modeling of Class Analogies

Linear discriminant analysis (LDA) finds a between-to-within group variance ratio and finds a coordinate space that maximizes that ratio to best separate the clusters or classes. Soft independent modeling of class analogies (SIMCA) computes PCA results for each class of results in LDA. Unknown spectra are cast into each PCA space, and if the spectra falls within the limits of that class, the spectra is said to belong to that class. The drawback is that LDA needs user input to select a spectral region for processing, thus, the reliability of the results is heavily dependent on the selection.
1.3.7 Data Compression Methods

Data compression methods are algorithms to reduce the size of the dataset. Often, the data is compressed while in storage and re-expanded for data processing. Few, if any of these methods generate maps showing the spatial location of various chemical compositions.

1.3.7.1 Discrete Cosine Transform

Discrete cosine transform (DCT) is a lossy data compression method. It is used to express each spectrum as the sum of cosine functions at different frequencies and amplitudes.\textsuperscript{42} DCT has advantages over other data compression methods in that it uses a low amount of computer memory and processing capability. To create image maps based on DCT results, most algorithms combine a spatial region with 8x8 or more spectra into a single block and define the combined spectra as a function of the cosine functions.\textsuperscript{43,44} This combination of spatially-similar spectra could overlook small regions that are chemically different, such as a small inclusion in a mineral or a contaminant in a biological sample.

1.3.7.2 Burrows-Wheeler Transform

The Burrows-Wheeler transform (BWT) operates by permutating the wavelengths in the spectra so that the new order has as many strings of repeating values as possible. BWT is valuable because it reduces the need for redundancy in the data: It is easier to compress a string with many repeating values. It is necessary to save an index of the original order of the wavelengths to reverse the BWT compression. This index would have the same number of values as there are wavelengths in a given spectra.\textsuperscript{45} If BWT is to be used to compress data in hyperspectral images, the spectra need to be very similar
in order for BWT to be effective. If each spectrum has its own index of orders, the size of the data after compression could be the same size as the original data. Ideally, each spectrum in the hyperspectral image should be permuted in the same order so that only one set of index values is needed. BWT works well when the spectra are all similar, but decreases in effectiveness as the spectra become more and more unique.

1.3.7.3 Lempel-Ziv 1977 and Lempel-Ziv 1978

Lempel-Ziv 1977 (LZ77) and Lempel-Ziv 1978 (LZ78) are compression algorithms that achieve compression by replacing strings of data with references to one copy of the string. In LZ77, the strings are matched to the first appearance of that particular string in the data, and the references are to that first appearance location. In LZ78, the strings are referenced to locations in a dictionary of strings. Both methods are considered to be lossless. Like BWT, LZ77 and LZ78 are effective in compressing the data when many spectra are similar, but decrease in effectiveness as the spectra become more and more unique.
REFERENCES


31 Y.L. Grossman, S.L Ustin, S. Jacquemoud, E.W. Sanderson, G. Schmuck, and J. Verdebout. “Critique of Stepwise Multiple Linear Regression for the Extraction of


CHAPTER II

REDUCTION OF SPECTRAL IMAGES FOR HYPERSPECTRAL IMAGE ANALYSIS: A VISUALIZATION METHOD

In recent decades, hyperspectral imaging has become a routine analytical method due to rapid advances in instrumentation and technique. Advances such as the speed of data acquisition,1-3 improved signal-to-noise-ratio,4 improved spatial resolution,5,6 and miniaturization of the instrumentation7 have all occurred, making chemical imaging methods more robust and more widely used.

Hyperspectral images contain spatial and spectral information for the sample. The most common techniques for hyperspectral image data acquisition include point-mapping, line-scanning, and wide-field spectral imaging.8 Any type of instrumentation which can gather a spectrum can be equipped to gather hyperspectral images by collecting a spectrum at each pixel location on a sample. Datasets may contain Raman scattering, infrared adsorption, fluorescence emission, near infrared absorption and reflectance, ultraviolet-visible absorption and reflectance, or any other type of spectra. This means that hyperspectral imaging is used on samples which are vastly different from one another, such ancient paintings,9 pharmaceuticals,10 cancer detection,11 and remote
Due to the large amount of data that is typically present in a hyperspectral image, there is a need for a reliable data processing method that can rapidly generate chemical-based image contrast from the data. Many samples lack *a priori* information or have unanticipated chemistries, thus limiting the usefulness of methods which require training data, such as classical least squares or various end-member classification methods. Principal component analysis (PCA) is a widely used method that does not require training data. PCA reduces the dimensionality of the data by recasting the data points into a subspace that is orthogonal to the greatest amount of variance. A series of principal component axes are iteratively computed with the first principal component accounting for the most variation, and subsequent principal components accounting for progressively less variation. Later (noisy) principal components are discarded. The problem with using this approach when processing a hyperspectral image is that it ignores some features that are important, but that contribute only to negligible variability.

### 2.1 Significance

In the work presented here, a method called reduction of spectral images (ROSI) is introduced. ROSI is a fully-automated, unassisted data processing technique that provides image segmentation and data classification based on chemical composition, without the need for complex training data sets. A novel data visualization strategy that takes advantage of the full colorspace of modern RGB displays for visualizing the image result and provides chemical composition-based image contrast for samples with large numbers of chemically important constituents is also presented. ROSI focuses on
processing hyperspectral image data that differentiates the spectra based on subtle changes in spectra shape, removes the degeneracies caused by spectral angle symmetry, and is generalized to work with any number of wavelength dimensions and image pixel resolutions.

The aim of ROSI is to achieve a threshold information density in the spectral dimension for all image pixels. The result effectively segments the image in a manner that provides rapid image contrast based on differences in chemistries. The result is comparable to traditionally classified images, but does so without training data. In addition, ROSI results are suitable for subsequent data analysis enabling ROSI to be performed alone or as a preprocessing data reduction step.

A second aim of ROSI is to eliminate the need for trained personnel to make decisions on how a dataset is processed. As the size of datasets increase, the demand for an automated method also increases. Otherwise, the amount of human oversight needed in to perform the analyses would also continue to increase. Automated analyses are also necessary for developing smart technologies that are usable by untrained persons.

This chapter discusses the use of ROSI as a hyperspectral image visualization method, whereas the next chapter discusses the use of ROSI as a data reduction method. The methodology will be discussed in both chapters, as there are differences between them, the most notable of which is the use of direction cosine space for image visualization and the use of squared direction cosine space for data reduction.
2.2 Methodology of Reduction of Spectral Images

In ROSI, each spectrum is thought of as a vector in $n$-dimensional space, where $n$ is the number of wavelengths. Each coordinate axis in the $n$-dimensional space represents one wavelength. For each spectrum, the position along each coordinate axis is the intensity at that wavelength. Each spectral vector has its head at the location dictated by the spectrum and its tail at the origin. The direction in which the spectral vectors point correlates to chemical composition and the length correlates to spectral intensity.

Because ROSI is a method to map the chemistries at each pixel location of a hyperspectral image, ROSI is concerned with the direction (chemical composition) of each spectral vector. As such, each spectral vector is converted into direction cosine space, as this retains the directions of the spectral vectors, but removes the effect of the lengths (spectral intensities). At this point, the heads of the spectral vectors lie on the surface of a hypersphere in $n$-dimensional space.

This is the point at which ROSI as a visualization method and ROSI as a data reduction method differ. In the data reduction method, each spectral vector is converted into direction cosine space and then squared. Rather than the heads of the spectral vector lying on the surface of a hypersphere, they lie on the surface of an $n$-dimensional plane. In other words, when the direction cosine values are squared, the dimensionality of the dataset is reduced by one, as there is now a dimension in which the value of each spectral vector is zero.

In either method, the dataset is then mean centered. In the visualization method, the heads of the spectral vectors still lie on the surface of the hypersphere, but the origin
is not necessarily at the center of the hypersphere, but closer to areas of the hypersphere with higher data density. When considering the angles between spectral vectors, moving the origin in this fashion exaggerates the differences in high data density areas of the dataset. This enables improved differentiation of those high data density regions. It does, at the same time, diminish the differentiation on the opposing side of the data cloud, but that region would have fewer data points in need of differentiation.

The set of principal component axes are found for the dataset, although principal component analysis scores are not computed. The first principal component axis points in the direction of most data variability. Subsequent principal component axes are all orthogonal to one another and pointing in directions of decreasing data variability. These axes are useful in that much of the data variability is defined in the first few dimensions.

A set of scores is determined that describes each spectral vector. The cosine correlation score is a measure of the similarity of two vectors. It is defined as the cosine of the angle between those vectors. Returning cosine correlation scores yields a statistically-relevant and intuitive score based on the similarity of the vector to the reference vector. Spectral vectors of spectra that have the same shape as the reference vector will have cosine correlation scores equal to 1 ($\theta = 0^\circ$), and spectra that are entirely dissimilar will yield cosine correlation scores equal to 0 ($\theta = 90^\circ$).\textsuperscript{15} Cosine correlation scores of the spectral vectors in the dataset are calculated against each reference vector. This yields a set of scores that describe the level of similarity of the spectral shapes of each vector in the dataset to the reference vector.

The square root of the sum of the squared ROSI scores describing each spectral
vector is equal to one, due to the fact that they are cosines of angles between a spectral vector and each of a set of orthogonal axes. If this calculation is performed on a subset of scores, the result can be from 0 to 1. Thus, if only some scores are retained for each spectral vector, this calculation is a measure of how well each spectral vector is described. These values are referred to as ‘reliability scores.’ The user inputs a threshold reliability score to which all spectral vector ROSI score sets are compared. For this comparison, each ROSI score set is sorted into descending order of magnitude (absolute values are taken). The minimum number of scores which can meet or exceed the desired reliability threshold are retained for each spectral vector. The scores and their index locations (which PC axes are needed for each spectral vector) are retained.

Principal component axes are inherently ordered in the directions of most to least data variability, but the goal of ROSI is to generate an image where differences in chemistry are most apparent. PCA assumes that the directions with the most variability are the most important. The principal component axes are reordered in ROSI so that chemistries that are a minority of the population can still be represented in the resultant image. Each spectral vector is assigned to a category based on the list of principal component axes needed to achieve the reliability threshold. Principal component axes are sorted by the number of categories that require that axis, such that the first axis is the principal component that is needed to define the largest number of groups.

In the ROSI algorithm, the direction of each spectral vector in a hyperspectral image is reduced to cosine correlation scores. The purpose of displaying the results in a colormap is to show differences in color based on spectral shape. Spectral shape corresponds to chemical composition. If each set of scores is used as an intensity value, a
map can be generated based on the location of each spectra in the original hyperspectral image. A resultant ROSI image is obtained by color-mapping each of three score set results to one of the color channels in a red, green, blue (RGB) image. The first ROSI image is generated from the first, second, and third score sets. Subsequent ROSI images may be generated from the subsequent score sets if desired, but they will, in general, show decreasing amounts of information.

Squaring versus not squaring the direction cosine space is a topic that would need to be studied further to develop a full explanation. However, in the results of each method, it is apparent that not squaring is beneficial to the visualization method and squaring is beneficial to the data reduction method.

In the ROSI visualization method, the resultant images show more of the subtlety of the dataset. It is believed that not squaring the values amplifies the variability. After putting the spectra into direction cosine space, the overall length of the spectral vector is equal to one, and therefore, each value that comprises the spectral vector is below one. When values between zero and one are squared, they become more concentrated towards zero and less concentrated towards one. This results in more detail being visible in a resultant image when direction cosine is used.

On the other hand, when ROSI is used as a data reduction method, like in the next chapter, the amount of data that needs to be retained to reach a threshold level of data recovery is lower when the direction cosine values are squared. This is likely for the opposite reason: Squaring the values likely diminishes the variability. In data reduction, the higher order subtly that amplifies the visualization result is not desirable.
2.2.1 User Input of Desired Reliability Threshold

Because ROSI resultant scores are cosines of the angles between a spectral vector and a set of orthogonal coordinate axes, the root sum square of a set of ROSI scores for a given spectrum is equal to one. The term ‘reliability threshold,’ denoted R, refers to the root sum square of a subset of scores for a given spectrum and can range from 0 to 1, depending on which scores are selected. As the reliability threshold approaches one, the spectrum is more reliably described by the set of scores. The user inputs a desired reliability threshold value, from 0 to 1, which is used as a threshold to determine which scores are retained for each spectral vector. For each spectral vector, scores are retained in order of greatest to least magnitude until the desired reliability threshold is met. (The sets of principal components needed to describe the dataset is later used to reorder the principal component axes.)

2.2.2 Casting of the Spectra into n-Dimensional Space

Each spectrum is cast into $n$-dimensional space, where $n$ is the number of wavelengths. Each axis in the $n$-dimensional space corresponds to a wavelength. Each spectrum is represented by a vector, called a spectral vector, whose tail resides at the origin and whose head resides at the point in space that was defined by the spectrum intensities at each wavelength in the basis space. Figure 3A shows a representation of two spectral vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$, plotted in $n$-dimensional space.

The data points in this figure, and the subsequent ROSI process figures, were generated in Matlab using four clusters of data, each with various positioning and standard deviations. The data points in each of the four categories were treated
Figure 3. A Casting of the Spectra into n-Dimensional Space and B Conversion of the Data to Direction Cosine Space.
identically throughout the ROSI process, but they are shown in different colors so that the way in which ROSI treats each cluster is apparent.

The set of spectral vectors is given by $V$ where the $i^{th}$ column, $V_{*,i}$, is the $i^{th}$ spectral vector, $v_{i}$, in wavelength order.

Spectral vector set $V$ is given by

$$V = \begin{bmatrix}
v_{1,1} & v_{1,2} & \cdots & v_{1,m} \\
v_{2,1} & v_{2,2} & \cdots & v_{2,m} \\
\vdots & \vdots & \ddots & \vdots \\
v_{n,1} & v_{n,2} & \cdots & v_{n,m}
\end{bmatrix}$$

(2)

2.2.3 Conversion of the Data to Direction Cosine Space

Each spectral vector is converted to direction cosine coordinates. The set of direction cosines for a vector contains the cosines of the angles between each coordinate axis and the vector. This effectively ignores the amplitude of each spectrum in subsequent steps as the length of each direction cosine-corrected spectral vector is equal to one due to the following equality: $\sum_{i=1}^{n} \cos^2 \alpha_i = 1$. This is an advantage because the goal of ROSI is to determine the chemistries at each pixel location, and in direction cosine space, the contributions of intensity are negated. Figure 3B shows the conversion of data points into direction cosine space.

An identity matrix of order $n$ is utilized to calculate the direction cosine spectra. Each column in the identity matrix, $I_{*,i}$, denoted $I_j$, is used as a wavelength axis when calculating direction cosine scores. The identity matrix for $n$-dimensional direction cosine space, $I_n$, is given by
The direction cosine is the cosine of the angle between two vectors. In notation, \( \langle \cos(\mathbf{A}\mathbf{B}) \rangle \) refers to the cosine of the angle between vectors \( \mathbf{A} \) and \( \mathbf{B} \). The set of direction cosine-corrected spectral vectors is given by \( \mathbf{V}' \), while each value within the matrix is denoted \( \gamma_{j,i} \) and given by

\[
\gamma_{j,i} = \cos \theta_{j,i} = \frac{\mathbf{I}_j \cdot \mathbf{V}_i}{|\mathbf{I}_j| |\mathbf{V}_i|} = \langle \cos(\mathbf{I}_j\mathbf{V}_i) \rangle
\]  

(4)

\[
\mathbf{V}' = \begin{bmatrix}
\gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,m} \\
\gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,m} \\
\vdots & \vdots & \ddots & \vdots \\
\gamma_{n,1} & \gamma_{n,2} & \cdots & \gamma_{n,m}
\end{bmatrix}
\]  

(5)

2.2.4 Mean Centering of the Dataset

Because PCA operates on mean-centered data, it is important to mean center the dataset so that it can be directly compared with the principal component reference vectors. The mean center vector is found by averaging the \( \mathbf{V}' \) matrix across each row, such that the result is a matrix containing one column with \( n \) values. The mean center is given by

\[
\overline{\mathbf{v}'} = \begin{bmatrix}
\overline{\gamma_{1,*}} \\
\overline{\gamma_{2,*}} \\
\vdots \\
\overline{\gamma_{n,*}}
\end{bmatrix}
\]  

(6)

Mean centering for each spectral vector in the dataset is given by

\[
\mathbf{v}_{i,*}' = \mathbf{v}_{i,*} - \overline{\mathbf{v}}'
\]  

(7)
The full dataset matrix is given by

\[
\mathbf{V}'' = \begin{bmatrix}
\mathbf{v}_{1,1}'' & \mathbf{v}_{1,2}'' & \ldots & \mathbf{v}_{1,n}'' \\
\mathbf{v}_{2,1}'' & \mathbf{v}_{2,2}'' & \ldots & \mathbf{v}_{2,n}'' \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{v}_{m,1}'' & \mathbf{v}_{m,2}'' & \ldots & \mathbf{v}_{m,n}''
\end{bmatrix}
\]  

(8)

Figure 4 shows the dataset being mean centered.

2.2.5 Selection of Reference Vectors Using Principal Component Analysis

Reference spectra are needed to optimize the cosine correlation of each spectral vector. The reference vectors must contain the same number of intensity values as there are wavelengths in the hyperspectral image dataset. While artificially-generated spectra, each having \(n\) randomly-selected positive intensity values, is mathematically suitable to distinguish the spectral vectors, having a reference vector that lies within the data cloud enhances the visual contrast of the resultant colormap. The reference vectors which are used are the principal components of the mean centered, directional cosine dataset, \(\mathbf{V}''\).

PCA can be used to extract pseudo-pure component spectra from mixed spectra in a hyperspectral image. It is a method that reduces the dimensionality of the data. PCA recasts the data points into a subspace that is orthogonal to the greatest amount of variance. The axes in this orthogonal space are called principal components. A series of principal components are iteratively computed. The first principal component accounts for the most data variation; the subsequent principal components account for progressively less variation, with the latter principal components accounting predominantly for noise.\(^{16}\) The problem with using PCA alone when processing a hyperspectral image is that it is both time-consuming in its iterative approach and largely
Figure 4. Mean Centering the Data
distinguishes only predominant features in the spectra, ignoring some features that are important but that contribute only to negligible variability. However, when using principal components as reference vectors in a correlation method, information about how different each spectral vector is from the principal component is also preserved.

The original set of basis space axes are defined by the identity matrix (equation 3). Principal component analysis is used to generate a new set of basis space axes for the dataset, and those axes (vectors) are all orthogonal to each other. PCA assumes that the directions with the most variability are the most important, and as such, the axes are ordered from most to least data variability. The first step in performing PCA is to generate a covariance matrix. Because the variance is the squared standard deviation, the values are denoted with $\sigma^2$. The calculation for the $ij^{th}$ value in the covariance matrix is given by substituting the dot product of $\mathbf{v}_i''$ and $\mathbf{v}_j''$ into

$$\sigma^2_{v_i''v_j''} = \frac{1}{n-1} \mathbf{v}_i''\mathbf{v}_j''^T$$

where the superscript T denotes a transposition of the data. The full covariance matrix is given by

$$\mathbf{S} = \frac{1}{n-1} \mathbf{V}'' \mathbf{V}''^T$$

$\mathbf{S}$ describes the relationships between all possible pairs of measurements in the dataset. $\mathbf{S}$ is a square matrix of order $n$. The diagonal values are the variances of that measurement, while the off-diagonal values are the covariances between those measurements.

Eigendecomposition is performed on the covariance matrix in order to represent
the matrix in terms of its eigenvalues and eigenvectors. In our approach, eigendecomposition operates on the data cast into $n$-dimensional mean-centered direction cosine space, instead of $n$-dimensional mean-centered wavelength space. Eigenvectors are vectors that when multiplied by the dataset may be scaled by a factor, $u$, but do not change direction. Thus, eigendecomposition satisfies the linear equation\(^{17}\) given by

$$S_{VV'} p_i = u p_i$$  \hspace{1cm} (11)

where $p_i$ are the eigenvectors and $u$ are the eigenvalues corresponding to $p_i$. The system of equations has non-trivial solutions\(^{17}\) if and only if

$$S_{VV'} - u I_n = 0$$  \hspace{1cm} (12)

Eigenvectors are found by substituting the eigenvalues back into equation 10. The eigenvectors are used as principal component axes (the new basis space axes), while the eigenvalues are used to sort the axes. PCA prioritizes variables with high variances, thus the eigenvectors (PCs) are sorted based on descending order of the eigenvalues (variances).

The set of principal components is given by $P$ where the $i^{th}$ column, $P_{*,i}$, is the $i^{th}$ principal component, $p_i$.

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\
p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix}$$  \hspace{1cm} (13)

Figure 5A shows the transformation of the data to the new principal component basis space.
2.2.6 Determination of Cosine Correlation Scores for All Spectral Vectors with Respect to the PCA Reference Vectors

The cosine correlation score is a measure of the similarity of two vectors. It is defined as the cosine of the angle between those vectors. Cosine correlation scores of the vectors in the dataset are calculated against each reference vector. This yields a set of scores that describe the level of similarity of the spectral shapes of each vector in the dataset to the reference vector.

The cosine correlation score between each reference vector, \( p_i \), and a mean-centered spectral vector, \( v_i'' \), is given by \( \gamma_{j,i}' \), while the full set of cosine correlation scores is given by \( \Gamma \). Each row in \( \Gamma \) contain the cosine correlation scores for one spectral vector.

\[
\gamma_{j,i}' = \cos \theta_{j,i} = \frac{p_j \cdot v_i''}{|p_j||v_i''|} = \langle \cos(p_jv_i'') \rangle \quad (14)
\]

\[
\Gamma = \begin{bmatrix}
\gamma_{1,1}' & \gamma_{1,2}' & \ldots & \gamma_{1,m}' \\
\gamma_{2,1}' & \gamma_{2,2}' & \ldots & \gamma_{2,m}' \\
\vdots & \vdots & \ddots & \vdots \\
\gamma_{n,1}' & \gamma_{n,2}' & \ldots & \gamma_{n,m}'
\end{bmatrix} \quad (15)
\]

Figure 5B shows the angles, \( \theta_{j,i} \), between the ROSI score vectors, \( v_i'' \), and the reference vectors, \( p_j \), on the left and depicts the relative magnitudes of the cosine correlation scores, \( \Gamma_{j,i} \), on the right.
Figure 5. A Selection of Reference Vectors using PCA and B Determination of Cosine Correlation Scores with Respect to PCA Vectors. The angles between each spectral vector and each principal component axis are labeled $\theta_{j,i}$ on the left and the relative magnitudes of the cosines of the angles $\theta_{j,i}$, labeled $\Gamma_{j,i}$ on the right.
2.2.7 Deciding Which Eigenvectors to Retain

When using ROSI as a data reduction method, the scores retained for each spectral vector can be used to determine how well each spectral vector is described for a given set of scores. The set of cosine scores for a spectral vector can be thought of as a set of coordinates that plot the reliability vector, a normalized spectral vector in mean-centered principal component space. If all of the scores are used, the length of that vector is equal to one. If fewer scores are used, the length can be from 0 to 1, but the closer it is to 1, the better the reliability of the spectral vector. If cosine correlation scores for all \( n \) principal component vectors had been retained, the length of the vector from the mean center to the position described by the scores would be equal to 1. Likewise, if fewer scores are used, but the spectral vector is still fully described with the chosen principal component directions, the reliability vector length would still be equal to 1. If a portion of the spectral vector is not described in the principal component directions being used, it will have a shorter length between 0 and 1. This value is called the reliability score. It is used as a threshold value to determine when each spectral vector has been adequately defined.

For each ROSI score vector, \( \Gamma_i' \), the absolute values of the scores are sorted from maximum to minimum to generate vector \( G_i \). The index order locations of the sorted values in \( \Gamma_i' \) is recorded in matrix \( S_i \). Thus, vector \( G_i \) contains the same data as vector \( \Gamma_i' \), but reordered based on \( S_i \).

\[
\Gamma_i' = \begin{bmatrix} \Gamma_{1,i}' \\ \vdots \\ \Gamma_{n,i}' \end{bmatrix} \tag{16}
\]
The term ‘reliability score,’ denoted $R$, refers to the root sum square of a subset of scores for a given spectrum. Because the resultant scores of ROSI are cosines of the angles between a spectral vector and a set of orthogonal coordinate axes, the root sum square of an entire set of ROSI scores for a given spectrum is equal to one. When the reliability score is found on a subset of scores, the value is between 0 and 1. As the reliability score approaches one, the spectrum is more reliability described by the set of scores. The following equation can be used to determine when the set of scores has reached the desired reliability threshold, $R$. The reliability score, $r_i$, for each vector in $G_i$ is given by

$$r_i = \sqrt{\sum_{k=1}^{m} G_{k,i}^2}$$

where $R$ is the desired reliability threshold and $m$ is the minimum number of scores that can satisfy $r_i \geq R$.

A binary matrix, $Q$, is used to record which scores are needed for each spectral vector. The position in the matrix has a value of one if the score is retained, and a value of zero if it is ignored.
Each spectral vector is assigned to a category based on the list of principal component axes needed to achieve the reliability threshold. Principal component axes are sorted by the number of categories that require that axis, such that the first axis is the principal component that is needed to define the largest number of groups.

Each unique column in $Q$ defines a category. The matrix of categories, $C$, is a matrix in which the unique columns of $Q$ are retained. Thus, in the case of replicate columns, only the first instance of that column in $Q$ is retained.

\[ C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,k} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,k} \end{bmatrix} \]  

(21)

where $k$ is the $k^{th}$ column in the $C$ matrix and the number of unique columns in matrix $Q$.

Each spectral vector is assigned to the category, $C_i$, that matches the list of principal components needed to achieve the reliability threshold. In other words, it belongs to the category where the column in $C$ that defines the category is the same as the column in $Q$ that describes the spectral vector.

The principal component axes are reordered based on the number of occurrences in the list of categories. To find the prevalence of each axis, sum across each row in $C$ such that $C'$ contains one column of length $n$, where $n$ is the number of principal
component axes. The prevalence of each principal component axis in the ROSI score set, $C'$, is given by

$$C' = \sum_{i=1}^{k} C_{j,i} \tag{22}$$

where $j$ is the $j^{th}$ row and $i$ is the $i^{th}$ column in matrix $C$.

The principal component axes and scores are reordered based on the values in $C'$. The principal component axes, $p^*$, are sorted such that their prevalence values, $C'$, are in descending order. The values in $C'$ are sorted in descending order in $C''$. The index locations of how $C'$ was reordered are recorded in $S'$.

$$C'' = \begin{bmatrix} |C'_{1,i}|_{max} \\ \vdots \\ |C'_{n,i}|_{min} \end{bmatrix} \tag{23}$$

$$S' = \begin{bmatrix} s_{1,i} \\ \vdots \\ s_{n,i} \end{bmatrix} \tag{24}$$

Each column in $\Gamma'$ is sorted based on the index order recorded in $S'$ to give the final ROSI scores, $\Gamma''$.

$$\Gamma'' = \begin{bmatrix} \Gamma'_{s'_{1,1}} & \Gamma'_{s'_{1,2}} & \ldots & \Gamma'_{s'_{1,n}} \\ \Gamma'_{s'_{2,1}} & \Gamma'_{s'_{2,2}} & \ldots & \Gamma'_{s'_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma'_{s'_{m,1}} & \Gamma'_{s'_{m,2}} & \ldots & \Gamma'_{s'_{m,n}} \end{bmatrix} \tag{25}$$

### 2.2.9 Visualization of the Scores Using a Colormap

In the ROSI algorithm presented here, the direction of each spectral vector in a hyperspectral image is reduced to three cosine correlation scores. The purpose of displaying the results in a colormap is to show differences in color based on spectral
shape. Spectral shape corresponds to chemical composition. If each set of scores is used as an intensity value, a map can be generated based on the location of each spectra in the original hyperspectral image. A resultant ROSI image is obtained by color-mapping each of the grayscale score set results to one of the color channels in an RGB image.

The first three sets of ROSI scores (the first three rows of $\mathbf{\Gamma}''$) are used to generate the resultant image. These three rows are referred to as $\Gamma_\alpha$, $\Gamma_\beta$, and $\Gamma_\gamma$, respectively. The $\Gamma_\alpha$ scores are used as red intensities in the resultant image. Likewise, $\Gamma_\beta$ and $\Gamma_\gamma$ are used as green and blue intensities, respectively. The RGB colorspace of modern computer displays can display 256 shades of the three colors and all the combinations thereof. The minimum and maximum score in each perturbation can be scaled from 0 to 255 to maximize the colorspace. For that reason, each of the three sets of scores are scaled from 0 to 255. The three colormapped images are combined to generate a composite image. The resultant image is intuitively understood in that similar colors are assigned to similar chemical compositions and dissimilar colors are assigned to dissimilar chemical compositions.

Figure 6 depicts the colormapping procedure.
Figure 6. Colormapping Procedure
2.3 Methodology of Principal Component Analysis

For the purposes of comparison, principal component analysis (PCA) is also discussed. To perform PCA, the first step is to mean center the dataset. The mean centering for each spectral vector in the dataset is given by

$$\mathbf{w}_{i,*} = \mathbf{v}_{i,*} - \bar{\mathbf{v}}$$

(26)

For PCA, the covariance matrix is generated using the mean centered data. The calculation for the $ij^{th}$ value in the covariance matrix is given by substituting the dot product of $\mathbf{w}_i$ and $\mathbf{w}_j$ into

$$\sigma^2_{w_iw_j} = \frac{1}{n-1} \mathbf{w}_i \mathbf{w}_j^T$$

(27)

where the superscript $T$ denotes a transposition of the data. The full covariance matrix is given by

$$\mathbf{S}_w \equiv \frac{1}{n-1} \mathbf{W} \mathbf{W}^T$$

(28)

$\mathbf{S}_w$ is a square matrix of order $n$ that describes the relationships between all possible pairs of measurements in the dataset. The variances of each measurement are the diagonal values, while the covariances between measurements are the off-diagonal values. Eigenvectors and eigenvalues are found for the covariance matrix. Eigenvectors are vectors that do not change direction when multiplied by the dataset, although they may be scaled by a factor, $\alpha$. Eigendecomposition satisfies the linear equation given by

$$\mathbf{S}_w \mathbf{q}_l = \alpha \mathbf{q}_l$$

(29)
where $\mathbf{q}_i$ are the eigenvectors and $\alpha$ are the eigenvalues corresponding to $\mathbf{q}_i$. For the system of equations to have non-trivial solutions, the following equality must be true:

$$
\mathbf{S}_w - a\mathbf{I}_n = \mathbf{0}
$$

(30)

Eigenvectors are found by substituting the eigenvalues back into Equation 28. In PCA, the eigenvectors are used as the new basis space axes, while the eigenvalues are used to sort the axes in order of decreasing variances.

The set of principal components is given by $\mathbf{Q}$ where the $i$th column, $\mathbf{Q}_{*,i}$, is the $i$th principal component, $\mathbf{Q}_i$.

$$
\mathbf{Q} = \begin{bmatrix}
q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\
q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
q_{n,1} & q_{n,2} & \cdots & q_{n,n}
\end{bmatrix}
$$

(31)

The PCA scores for each mean centered spectral vector are the portions of the length of the spectral vector in the direction of each principal component axis. In other words, the principal component axes are used as coordinate axes in Cartesian space and the set of scores for each mean centered spectral vector is the position of the head of the vector in that space. Scores are calculated between each vector in the dataset and each principal component axis. Each score is given by $\varepsilon_{j,i}$.

$$
\varepsilon_{j,i} = \mathbf{q}_j \cdot \mathbf{w}_i
$$

(32)

The resultant image is generated by scaling the first three sets of scores from 0 to 255 and using those values as red, green, and blue intensities, respectively, at each pixel location.
2.3.1 The Scale Variant Problem in PCA

The problem with PCA is that it is scale variant. In other words, scores are assigned based on a combination of spectral vector chemistries and intensities. Spectra with the same chemistries are represented by spectral vectors pointing in the same direction. The lengths of those vectors correspond to the intensity of the spectra. In PCA, it is possible to have spectral vectors pointing in different directions that are assigned the same scores in the first few dimension due to a coincidence in the lengths of the vectors that cause the projections along an axis or several axes to be the same. A depiction of this is shown in Figure 7A. These same spectral vectors in ROSI, however, are assigned scores based on cosines of the angles between the spectral vectors and the axes, causing the same spectral vectors to have different scores, as shown in Figure 7B.

Another example of the scale variant problem in PCA is for spectral vectors that point in the same direction but have different lengths. (These would represent spectra with the same chemistries, but different intensities.) The resultant scores in PCA would differ for these spectral vectors, as shown in Figure 7C. In ROSI, their scores would be the same, as shown in Figure 7D.

In both cases, a small subset of the sets of scores are used to generate an image. It is quite possible that the scores needed to overcome the scale variant problem in PCA will exist in later sets of scores that are dismissed. The color assignment for each pixel in the PCA resultant image is due to a combination of the chemistry and the intensity for the corresponding spectral vector. If the goal is to see a map of which chemistries are at which pixel locations, the PCA resultant image in unreliable and ROSI is a better choice.
Figure 7. The Scale Variant Problem in PCA. A and B show the scores of spectral vectors with different chemistries, \( V_1 \) and \( V_2 \), in PCA and ROSI, respectively. These vectors are pointing in different direction, and therefore have different chemistries. Ideally, they should have different scores. For these particular intensities, the scores are the same in PCA. C and D show two spectral vectors with the same chemistries, but different intensities, \( V_1 \) and \( V_2 \), in PCA and ROSI, respectively. In PCA, these vectors are assigned different scores, even though the chemistries are the same. In PCA, spectral vectors are assigned scores due to a combination of the chemistries and the intensities of the spectra, whereas ROSI is specific to chemistries.
2.4 Methodology of Classical Least Squares

Classical Least Squares (CLS) will also be used as a comparison for the results. CLS is a regression method that aims to minimize the sum of the squared error between the observed responses and the responses predicted by a linear determination. Thus each spectrum is a weighted sum of the pure component spectra. Those weights or abundances are the values that CLS seeks to estimate.\textsuperscript{18,19}

CLS uses the linear least squares approach for assigning scores to each spectrum. The user inputs \( n \) pure component spectra to which each spectrum in the dataset is compared. CLS assumes a linear relationship between all of the possible combinations of pure component spectra. Scores are assigned such that the sum of the squared differences between each dataset spectral vector and each pure component spectral vector are minimized. Each dataset spectral vector is assigned \( n \) scores which describe the similarity of that vector to each of the \( n \) pure components. Different visualization strategies exist for CLS. Each pixel can be assigned one of \( n \) colors based on the pure component to which it is most similar. Those \( n \) colors can be displayed in a resultant image as is, or they can be scaled by the scores to show bright to dark variations among each color. This shows how well each pixel has been described by the pure component category to which it was described.

The drawback of CLS is that a trained user must have knowledge about the sample and must select appropriate pure component spectra. If most of the sample is understood, but a minority of pixels represent an unexpected chemistry, it can easily be
missed. This method is simple to use and reliable, but it is only as good as the knowledge that the user has about each sample. This severely limits the usefulness of CLS.

2.5 Results of ROSI as a Hyperspectral Imaging Method

ROSI results are given for three different hyperspectral image datasets. The first dataset is a Raman image collected from a sample of polished concrete. This sample was chosen because of its inherent complexity and rough morphology. The second and third hyperspectral images were manufactured to pose a challenge to the ROSI method. The second dataset was designed to study the limitation of the ROSI resultant image, in that a recognizable image of piano keys was subtly built into the chemistries at various pixel locations. In the third dataset, every pixel in the dataset is unique. It was designed to study the limitation of the overlap of color assignments in the ROSI resultant image.

2.5.1 Concrete Raman Dataset

Experimental

The concrete dataset was collected as a real-world example of a sample in which limited information was available about the chemical components of the dataset. It is a Raman hyperspectral image, containing 17,423 pixels (arranged 131 x 133) at 502 wavelengths, ranging from ~200 cm\(^{-1}\) to ~1800 cm\(^{-1}\). It was collected from a 10μm x 10μm square on the surface of the sample. All analyses were performed on the image data using routines written in Matlab R2013a (The Mathworks, Inc.).

Results

A brightfield image of the concrete sample is shown in Figure 8A, and selected spectra are shown in Figure 8B. ROSI was run with a reliability threshold of 0.954. The
Figure 8. Brightfield Image and Selected Spectra of the Concrete Dataset. A shows the brightfield image of the concrete sample and B shows the mean spectra of three selected regions.
ROSI resultant image of the concrete sample can be seen in Figure 9A. The reliability thresholds were calculated for each pixel location based on the three scores which were used to generate the resultant colormap. A reliability threshold map is shown in Figure 9B. In this image, the root sum square of each of the three scores used for each pixel was calculated. Darker pixels represent a lower reliability threshold and brighter pixels represent a higher reliability threshold.

While the primary ROSI resultant image (Figure 9A) displays as much of the data reliability as ROSI and modern computer displays allow in one image, more images are sometimes necessary to visualize each pixel to the desired reliability threshold. After the primary image is generated, the reliability threshold is calculated for each pixel (Figure 9B). The reliability threshold for each pixel are compared to the threshold reliability threshold that was input by the user. Any pixels which have not achieved the reliability threshold are present in the following image, with the three subsequent ROSI scores sets mapped to red, green, and blue, respectively. Pixels which have achieved the desired reliability threshold are black in all subsequent images. A series of images is created in this fashion until every pixel has achieved the threshold reliability threshold, as shown in Figure 10.

Common methods for processing hyperspectral image datasets include principal component analysis (PCA) and classical least squares (CLS). PCA, like ROSI, does not utilize training data and does not require a trained user. The resultant image for PCA is shown in Figure 11A. CLS, unlike ROSI, requires a trained user to select pure component spectra to which each spectrum in the dataset is compared. The resultant image for CLS is shown in Figure 11B.
Figure 9. ROSI Images of the Concrete Dataset. A shows the ROSI resultant image and B shows the reliability of the ROSI resultant image.
Figure 10. All ROSI Images of Concrete Dataset until Desired Reliability Threshold is Reached.
Figure 11. Resultant Images of Concrete using Other Methods. **A** shows the principal component analysis (PCA) resultant image of the concrete dataset and **B** shows the classical least squares (CLS) resultant image.
2.5.2 Piano Dataset

Experimental

The piano hyperspectral image dataset was manufactured to analyze the limitations of ROSI. Five Raman spectra were chosen (afwillite, cristobalite, quartz, langbeinite, and dolomite) and are shown in Figure 12. These spectra were ordered such that when they are in equal proportions and principal component axes are found, the first three axes correlate best with components A, B, and C. Each spectrum (pixel location) is a linear combination of the five components. Figure 13A shows the high contrast version of the masks of the five components, where white is a molar fraction of one and black is a molar fraction of zero. The square patterns in the top portion of the image, in addition to the boarder around the piano image area, contain all of the possible combinations of the five components.

Figure 13B shows the actual molar fraction masks used to generate the linear combinations of the five components at each pixel location. The molar fractions in the images of the piano keys in components D and E were reduced so that these chemistries would be in the minority. An additional requirement was added such that the five molar fractions of each pixel summed to one. All dataset design and analyses were performed using routines written in Matlab R2013a (The Mathworks, Inc.).
Figure 12. Five Components of Piano Image and Tetrahedral Datasets: Afwillite, Cristobalite, Quartz, Langbeinite, and Dolomite, Respectively.
Figure 13. The Molar Fraction Masks for the Five Spectra in the Piano Dataset. A shows the high contrast molar fraction masks and B shows the actual molar fraction masks used for the five spectra. The chemical components for which each mask was used is above each image.
Results

ROSI was run on the piano dataset with a reliability threshold of 0.954. The resultant images are shown in Figures 14 and 15.

Figure 14A shows the ROSI resultant image. The reliability thresholds were calculated for each pixel location based on the three scores which were used to generate the resultant colormap. A reliability threshold map is shown in Figure 14B. In this image, the root sum square of each of the three scores used for each pixel was calculated. Darker pixels represent a lower reliability threshold and brighter pixels represent a higher reliability threshold.

Figure 14C shows all of the ROSI resultant images until the reliability threshold of 0.954 is reached. Pixel locations in which the reliability threshold was reached in the first image and omitted (black) in the subsequent image.

Figure 15 shows image results of other commonly-used hyperspectral imaging processing methods. Figure 15A shows the resultant image using PCA, whereas Figure 15B shows the resultant image using CLS.
Figure 14. ROSI Images of the Piano Dataset. A shows the ROSI resultant image, B shows the reliability of the ROSI resultant image, and C shows all of the ROSI resultant images until the reliability threshold of 0.954 is reached.
Figure 15. Resultant Images of the Piano Dataset using Other Methods. A shows the principal component analysis (PCA) resultant image of the piano dataset and B shows the classical least squares (CLS) resultant image.
2.5.3 Five Component Filled Molar Fraction Space Dataset

Experimental

The five component filled molar fraction space dataset was also manufactured to analyze the limitations of ROSI. Five Raman spectra were chosen (afwillite, cristobalite, quartz, langbeinite, and dolomite) and are shown in Figure 12, as they were the same spectra used in the piano dataset. Each spectrum (pixel location) is a linear combination of the five components.

All molar fractions were an increment of 1/30 from 0 to 1. Additionally, the five molar fractions for each pixel have a sum of 1. A total of 46,376 molar fraction combinations were found that fit these criteria. Thus, the hyperspectral image dataset contains 46,376 unique spectra, each at a different pixel location. All dataset design and analyses were performed using routines written in Matlab R2013a (The Mathworks, Inc.).

Results

ROSI was run with a reliability threshold of 0.954. The resultant images, shown in Figures 16 and 17, are the standard set of resultant images, as described previously. These images appear complex, but the overall appearance is not what is important. The value of these images is in the number of colors displayed and how the colors are assigned to pixels representing similar and dissimilar spectra. Ideally, in a resultant image for a dataset containing spectra that are all unique, the number of unique colors should be the same as the number of spectra (pixels). Additionally, the color should be a function of how similar the spectra are to one another. In other words, similar spectra should be assigned similar colors and dissimilar spectra should be assigned dissimilar
colors. **Figure 16A** shows the ROSI image result of the five component dataset. The image contains 42,331 unique colors for the 46,376 unique spectra; thus, some unique spectra are assigned the same color. However, the PCA resultant image (**Figure 17A**) contains 31,106 unique colors for the same 46,376 unique spectra. In this regard, ROSI has outperformed PCA. **Figure 17B** shows the CLS resultant image. As CLS assigns each spectrum to the most similar pure component spectrum, the image has only five unique colors, as there are five pure component spectra. **Figure 16B and C** show the reliability of the ROSI resultant image and all of the ROSI images needed to reach a reliability threshold of 0.954.

The groups of spectra that were assigned the same color were compared for both ROSI and PCA. These pairs are the failure points in each method. The cosine of the angle between two spectral vectors is used as a measure of similarity between the two spectra, in that the cosine of the angle between identical spectra is equal to one and between dissimilar spectra is zero. All pairs of spectra which were assigned to the same colors in the primary ROSI resultant image and in the PCA image are compared in this fashion. Because similar spectra should be assigned similar colors, the ideal outcome would be that the spectra which were assigned the same color are similar to one another.

**Figures 18 and 19** contain plots of cosine correlation scores between pairs of spectra which were assigned the same color in ROSI and PCA, respectively, versus the reliability scores for those spectra. Each pair of spectra is represented by two points which are vertically aligned, in that they have the same cosine correlation score, but they each have their own reliability score. Reliability scores for PCA were the square root of the sum squared of the scores used in the colormap divided by the overall length of the
Figure 16. ROSI Resultant Image of the Five Component Filled Molar Space Dataset. 
A shows the ROSI resultant image run at a reliability threshold of 0.954. 
B shows the ROSI reliability of the resultant image in A. 
C shows all of the ROSI images that are required for each pixel to reach the reliability threshold of 0.954.
Figure 17. Resultant Image of Other Methods for the Five Component Filled Molar Space Dataset. A shows the resultant image when the dataset is processed using PCA. This image displays 31,106 unique colors. B shows the CLS resultant image.
score vector, such that if all of the scores were used, the value would be equal to one, like in ROSI.

**Figure 18A** contains a plot of cosine of the angle between spectra versus reliability thresholds for sets spectra in the primary ROSI image which were assigned the same color. **Figure 18B** contains a subset of that same plot, but containing only sets of spectra that were assigned the same color and are similar (the cosine of the angle between the spectra is ≥0.99). It can be seen that the vast majority of spectra that were assigned the same color are similar to one another and spectra which are dissimilar have poor reliability thresholds. **Figure 19A** contains a plot of cosine of the angle between spectra versus reliability thresholds for sets of spectra which were assigned the same color in PCA. **Figure 19B** shows the subset of spectra which were assigned the same color and are similar. Note that while the cosine of the angle between spectra is close to one, the spectra tend to be more dissimilar than in ROSI.
Figure 18. Comparison of the Spectra which were assigned the Same Color in the ROSI Resultant Image. A shows a plot containing all spectra, where B shows a subsection of that plot containing spectra that are similar to one another.
Figure 19. Comparison of the Spectra which were assigned the Same Color in the PCA Resultant Image. A shows a plot containing all spectra, where B shows a subsection of that plot containing spectra that are similar to one another.
2.6 Conclusion

A fully automated, user unassisted data processing technique that generates chemically-relevant image contrast from hyperspectral image data, without the need for training data sets has been designed. ROSI achieves a threshold information density in the spectra dimension for all image pixels and provides rapid image contrast using a novel data visualization strategy that takes advantage of the full colorspace of modern RGB displays. In addition, ROSI results are suitable for subsequent data analysis enabling ROSI to be performed alone or as a preprocessing data reduction step.

A concise protocol was put forth that will enable other researchers to utilize this method by following a short list of steps. The description has been generalized to work with any number of wavelength dimensions and spectra. In addition, ROSI image results from hyperspectral data on both model data and real samples were presented.
REFERENCES


CHAPTER III
UNASSISTED REDUCTION AND SEGMENTATION OF LARGE
HYPERSPECTRAL IMAGE DATASETS

For heterogeneous samples, hyperspectral imaging is used to map variations in chemical composition. The data consists of a set of spectra in which individual spectra represent unique pixel locations in the rendered image. In recent years, there have been advances in the speed of data acquisition\(^1\)-\(^3\) brought about by fast scanning tunable wavelength filters (acousto-optic tunable filters\(^4\),\(^5\) and liquid crystal tunable filters\(^6\)), tunable light sources, and more sensitive array detectors. Advances have also lead to improved signal-to-noise-ratios,\(^7\) improved spatial resolution,\(^8\),\(^9\) larger image formats with higher pixel densities, and miniaturization of the instrumentation,\(^10\) making chemical imaging methods more robust, more nimble, and more attractive to research across many disciplines.\(^11\)-\(^16\) All of this has culminated in larger and increasingly complex datasets, leading to an ever-increasing need for reliable data reduction, segmentation, and classification methods.

The information density in hyperspectral data is not uniform across the spectral
and spatial dimensions, and the overall information sparsity is often high. While pixel-to-pixel spectral differences underpin the sought-after image contrast upon which compositional information is based, high sparsity generates unnecessarily long acquisition and data processing times. When the objective of hyperspectral imaging is to enable differentiation between chemical domains, which are distinguishable by band positions and the relative differences in band intensities, the overall amplitudes of the spectra are not needed, even for mixtures. Amplitudes are necessary, however, for establishing either the absolute or relative concentrations of the chemical constituents. Consequently, the shape of a spectrum yield qualitative chemical information while the amplitude can be used to quantify the amounts of each chemical constituent. The aim of effective data reduction methods for chemical imaging applications is to reduce the data sparsity while retaining relevant information for either qualitative or quantitative analyses, or both.

To date, reduction techniques based on principal components analysis (PCA) have been widely used.\textsuperscript{17-22} The conventional use of PCA in hyperspectral imaging is to exploit the covariance between spectra to establish a linearly independent set of wavelength combinations (the principal component basis space) that accounts for most of the data variability. The data compression afforded by PCA is obtained by sacrificing the information-bearing contributions of minority pixel populations while retaining the spectral information for pixel populations that represent greater portions of the overall variability. The effect is that some pixel regions in the PCA-compressed image data retain useful information while other regions are poorly rendered. This makes it difficult to assess the chemical characteristics or relevance of the minority pixel regions, even
when this information would reveal important sample defects or spectral
inhomogeneities. Casting all spectra into a single basis space of reduced dimensionality,
even when it retains most of the data variability, has the undesired effect of obscuring the
subtle sample characteristics that hyperspectral imaging is designed to reveal.

Data reduction is always lossy, but highly effective data reduction methods reduce
data sparsity in a way that retains only that information relevant for the analysis being
performed. Data sparsity can be due to noise, acquisition errors, data redundancies, and
the inclusion of unnecessary data for which no meaningful variability can be attributed.
Random fluctuations in hyperspectral data arising from shot noise or thermal noise are
often unavoidable and lead to increased sparsity. Determinate errors, such as errors in the
calibration process, can be mitigated, while other error sources may not be known.23,24
Redundancies in the data can occur when different pixel locations have similar
chemistries, thereby producing similar spectra. It is likely that not all wavelengths in the
spectral region being interrogated are important for elucidating chemical differences in
the sample. Hence, data sparsity is not necessarily the same for every application. By
extension, the spectra acquired during hyperspectral imaging can be thought of as a set of
independent experiments performed on different samples, each one representing a
different location on the imaged surface. Hence, the sparsity should not be assumed to be
same for all spectra in the dataset, yet the conventional use of PCA applies the same
reduced set of principal components to each spectrum. This is analogous to having
assumed that the data sparsity resides in the same wavelength bands for all spectra in the
dataset.
3.1 Significance

The objective of this work is to provide data reduction for hyperspectral imaging under the restraint that the spectral information retained for each image pixel meets or exceeds a minimum threshold value called the reliability threshold. To meet this objective, a fully-automated and user unassisted data reduction method called reduction of spectral images (ROSI) has been developed. By preserving spectral shape differences, even for minority constituents and small pixel populations, the reduced dataset generated by ROSI provides reliable image segmentation across the entire field of view. The ROSI data format is suitable for immediate visualization and interpretation, or for subsequent chemometric processing. Enough information is retained to generate estimates of the original spectra by decompressing the ROSI data. A set of scores between 0 and 1, called the reliability scores, identify how well each regenerated spectrum compares to its original spectrum. The ROSI procedure is useful for many applications beyond elucidating chemical information from hyperspectral image data. It can be used as a reduction or segmentation method for any collection of ordered sets in which relative differences between the sets are important. It also has considerable potential in the development of neural networks and, because of its effectiveness, can reduce the complexity of the hidden layer designs used in machine learning applications.

A full theoretical description of ROSI is presented here along with illustrations that highlight key data transformations and the reduction steps. In addition, ROSI is applied to two types of Raman hyperspectral image datasets to test its efficacy for qualitative analyses. The first dataset contains actual Raman point-mapped data acquired from a four component system of minerals. The sample was manually constructed and
serves as a model system for which the chemical composition at each pixel location in the hyperspectral image data is known. Data reduction is performed for this data using both ROSI and PCA and a comparison of their performance is discussed. The second type of data is artificially constructed by generating linear combinations of the spectra from the pure components used in the first dataset. The collective set of spectral mixtures uniformly and densely span the concentration space, and each resulting spectrum appears only once in the hyperspectral data. These conditions are selected to test how well ROSI provides data reduction when there are no minority or majority pixel populations, and when there is no clustering of the data that could indicate compositional demarcation. Because ROSI provides spectral data reduction alone and does not compress the spatial information, it is unnecessary to assign these spectra to pixel locations within an image. As is the treatment of all hyperspectral data by ROSI, the data is acted on as an ordered set of spectra. When image data is used, rendering of the final image result is accomplished by reshaping the ordered set of ROSI results according to the pixel-dimensions of the image. To evaluate the performance of ROSI for hyperspectral data exhibiting different signal-to-noise (S/N) levels, noise was added to the artificial data. Results are shown here for 11 S/N levels between 4 and 100. For each of these, ROSI was repeated for 7 different reliability thresholds between 0.5 and 1.0.

3.2 Methodology of ROSI as a Data Reduction Method

In ROSI, restrained data reduction that preserves qualitative chemical information from hyperspectral image data is sought. Because qualitative information is encoded by spectral shape, characterized by band positions and relative band intensities, the overall amplitudes of the spectra are not needed and this information is removed to eliminate its
potential effects on data reduction. By comparison, the conventional use of PCA embeds amplitude information into the compressed data, thereby making it difficult to distinguish the qualitative characteristics of the data from the quantitative ones. The separation of spectral shape from amplitude has often been attempted by normalizing each spectrum to span the intensity values between 0 and 1. The actual effect, however, does not remove the amplitude information, it simply sets the amplitudes to 1. Another commonly used technique has been to convert each spectrum into a vector of length 1. This is accomplished by dividing each intensity value in a spectrum by the square root of the sum of squared intensities comprising the spectrum, which is the same as dividing a vector by its Euclidean length to produce a unit vector. In this case, the spectral vectors are all unit vectors, but the resulting amplitudes can be dissimilar for spectra of different shapes. Separation and elimination of the amplitudes while preserving spectral shape information is achievable using the concept of spectral angle, which is the first data transformation that occurs in ROSI.

One way to envision the hyperspectral data is as a collection of \( m \) spectra. For performing ROSI, it is convenient to think of each spectrum as a vector, \( \mathbf{v} \), in an \( n \)-dimensional orthogonal basis space, where \( n \) is the number of wavelength bands in common with each spectrum. The coordinate axes each represent one wavelength band and the position along each axis is the intensity for that band.

In ROSI, each spectrum is replaced with set of numbers, the amplitude of the original spectrum and a set of scores that correspond to the statistical correlation of the spectrum to basis spectra. If the set of basis spectra are known, the original spectra can be recovered from the correlation scores after rescaling to achieve the initial amplitude.
Because the same set of basis spectra is used to recast all of the spectral data, they do not need to be included more than once in the compressed data format. Consequently, the degree of data reduction is dependent mostly on the number of spectra. As the number of spectra increases, so does the degree of data compression. It is important to note that the data is not simply recast according to a new basis space in a way somewhat analogous to PCA.

Because the goal of ROSI is to preserve the chemistries at each pixel location of a hyperspectral image, the direction (chemical composition) of each spectral vector is important. As such, each spectral vector is converted into squared direction cosine space to remove the effect of the lengths (spectral intensities).

The dataset is mean centered, and the set of principal component axes are found for the dataset, although principal component analysis scores are not computed. These axes are useful in that much of the data variability is defined in the first few dimensions. A set of cosine correlation scores is determined that describe each spectral vector. A cosine correlation score is the cosine of the angle between two vectors. This yields statistically-relevant and intuitive scores based on the similarity of those two vectors, as scores range from 0 to 1 for dissimilar and similar vectors, respectively. The scores are useful in determining how well each spectral vector is described from a subset of the scores and they are the retained values in the data compression method.

A flow chart overviewing the steps in ROSI can be seen in Figure 20.
1. **Hyperspectral Data (Raman)**
   
   After performing background subtraction and cosmic removal, the amplitude for each spectral vector is recorded. The hyperspectral image has 2 spatial dimensions (x and y) and 1 wavelength dimension (λ).

2. **Cast Data into Squared Direction Cosine Space**
   
   The squares of the cosines of the angles between each spectral vector and each coordinate axis are found.

3. **Mean Center the Data and Record Amplitudes**
   
   The mean of all squared direction cosine spectral vectors is found and subtracted from each vector. The amplitudes of each vector is recorded at this time.

4. **Cast Data into PCA Space**
   
   Principal component axes are found and used as coordinate axes for the dataset.

5. **Perform CCA**
   
   Cosine correlation analysis is performed by finding the cosines of the angles between each dataset vector and each principal component axis.

6. **Retain Scores to Meet Reliability Threshold**
   
   For each dataset vector, retain the minimum number of scores needed to reach the reliability threshold. Scores of the largest amplitudes are prioritized so that the fewest number of scores possible are retained.

---

**Figure 20.** Flow Chart for ROSI Methodology
3.2.1 Casting of the Spectra into $n$-Dimensional Space and Recording Amplitudes

Each spectrum can be thought of as a vector in $n$-dimensional space, where $n$ is the number of wavelengths. The position of head of the vector is defined by the spectral intensities along each wavelength axis, while the tail of the vector resides at the origin.

The set of spectral vectors is given by $\mathbf{V}$ where the $i^{\text{th}}$ column, $\mathbf{V}_{*,i}$, is the $i^{\text{th}}$ spectral vector, $\mathbf{v}_i$, in wavelength order. Spectral vector set $\mathbf{V}$ is given by

$$
\mathbf{V} = 
\begin{bmatrix}
\mathbf{v}_{1,1} & \mathbf{v}_{1,2} & \cdots & \mathbf{v}_{1,m} \\
\mathbf{v}_{2,1} & \mathbf{v}_{2,2} & \cdots & \mathbf{v}_{2,m} \\
\vdots & \ddots & \vdots \\
\mathbf{v}_{n,1} & \mathbf{v}_{n,2} & \cdots & \mathbf{v}_{n,m}
\end{bmatrix}.
$$

(33)

The amplitude of each spectral vector corresponds to quantitative information at a particular image pixel and the direction of each vector corresponds to chemical composition. The amplitudes of each spectral vector should also be recorded, as they are needed to recalculate the dataset after data reduction. The amplitudes are not explicitly needed for the qualitative aspect of ROSI, but the amplitude of each spectral vector is recorded because that information is used in recalculating the original spectral vector when utilizing the data reduction methodology of ROSI.

The amplitude, $a_i$, is given by

$$
a_i = \sqrt{\sum_{j=1}^{n} V_{j,i}^2}.
$$

(34)

where $j$ is the $j^{\text{th}}$ wavelength band or frame number.
3.2.2 Conversion of the Data to Squared Direction Cosine Space

Per step 2 in Figure 20, each spectral vector is converted to squared direction cosine coordinates. Direction cosine coordinates for a vector contain the cosines of the angles between each coordinate axis and the vector. Because the goal of ROSI is to preserve chemical identities, the benefit of direction cosine space is that amplitude (quantitative information) of each spectral vector is ignored, as the following equality is true:

$$\sum_{i=1}^{n} \cos^2 \alpha_i = 1$$  \hspace{1cm} (35)$$

Each column in an identity matrix of order \(n\) is used as a wavelength axis when calculating the squared direction cosine spectra. The identity matrix for \(n\)-dimensional squared direction cosine space, \(I_n\), is given by

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}_{n \times n}$$ \hspace{1cm} (36)$$

and each column, \(I_j\), is denoted \(I_j\). In notation, \(\langle \cos^2(AB) \rangle\) refers to the square of the cosine of the angle between vectors \(A\) and \(B\). The set of squared direction cosine-corrected spectral vectors is given by \(V'\), and each value within the matrix is denoted \(\gamma_{j,i}\) and given by

$$\gamma_{j,i} = \cos^2 \theta_{j,i} = \left( \frac{I_j \cdot V_i}{|I_j||V_i|} \right)^2 = \langle \cos^2(I_j V_i) \rangle$$ \hspace{1cm} (37)$$

$$V' = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \ldots & \gamma_{1,m} \\ \gamma_{2,1} & \gamma_{2,2} & \ldots & \gamma_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n,1} & \gamma_{n,2} & \ldots & \gamma_{n,m} \end{bmatrix}$$ \hspace{1cm} (38)$$
### 3.2.3 Mean Center the Dataset and Record the Amplitudes

The dataset is mean centered so that when a new set of PCA-based coordinate axes are found, the origin is within the data cloud. The mean center vector is found by averaging the $\mathbf{V}'$ matrix across each row, such that the result is a matrix containing one column with $n$ values. The mean center is given by

$$\overline{\mathbf{v}'} = \begin{bmatrix} \overline{y}_{1,*} \\ \overline{y}_{2,*} \\ \vdots \\ \overline{y}_{n,*} \end{bmatrix}.$$  \hfill (39)

Mean centering for each spectral vector in the dataset is given by

$$\mathbf{v}_{i,*}' = \mathbf{v}_{i,*}' - \overline{\mathbf{v}'}.$$  \hfill (40)

The full dataset matrix is given by

$$\mathbf{V}'' = \begin{bmatrix} v_{1,1}'' & v_{1,2}'' & \cdots & v_{1,n}'' \\ v_{2,1}'' & v_{2,2}'' & \cdots & v_{2,n}'' \\ \vdots & \vdots & \ddots & \vdots \\ v_{m,1}'' & v_{m,2}'' & \cdots & v_{m,n}'' \end{bmatrix}.$$  \hfill (41)

At this point, the amplitude for each mean-centered, squared direction cosine spectral vector needs to be recorded. The amplitude, $a_{mc,i}$, is given by

$$a_{mc,i} = \sqrt{\sum_{j=1}^{n} \mathbf{v}_{j,i}'''^{2}}.$$  \hfill (42)

where $j$ is the $j^{th}$ wavelength band or frame number.

### 3.2.4 Selection of Reference Vectors Using Principal Component Analysis

A new set of coordinate axes are found which are intended to optimize the cosine correlation of each spectral vector. While artificially-generated spectra each having $n$
randomly-selected positive intensity values is mathematically suitable to distinguish the spectral vectors, having a reference vector that lies within the data cloud enhances the contrast between relevant and sparse data. These new coordinate axes are the principal components of the mean centered, squared direction cosine dataset, $\mathbf{V}''$.

When processing a hyperspectral image with PCA alone, predominate features in the spectra are largely distinguished, but some features which are important but contribute only negligible variability are ignored. In ROSI, the principal components act as reference vectors in a correlation method. This means that information about how different each spectral vector is from the principal component is also retained.

In PCA, a series of principal components are iteratively computed. The first principal component points in the direction of the most data variability, and each subsequent principal component is orthogonal and points in directions of progressively less variability. Later principal components account predominately for noise. In ROSI, the data is recast with principal components as the coordinate axes.

To find principal components, the first step is to generate a covariance matrix. Because the variance is the squared standard deviation, the values are denoted with $\sigma^2$. The calculation for the $ij^{th}$ value in the covariance matrix is given by substituting the dot product of $\mathbf{v}_i''$ and $\mathbf{v}_j''$ into

$$\sigma^2\mathbf{v}_i''\mathbf{v}_j'' = \frac{1}{n-1}\mathbf{v}_i''\mathbf{v}_j''^T$$

where the superscript $T$ denotes a transposition of the data. The full covariance matrix is given by
where $S_{V''}$ is a square matrix of order $n$ that describes the relationships between all possible pairs of measurements in the dataset. The variances of each measurement are the diagonal values, while the covariances between measurements are the off-diagonal values. Eigenvectors and eigenvalues are found for the covariance matrix. Eigenvectors are vectors that do not change direction when multiplied by the dataset, although they may be scaled by a factor, $u$. Eigendecomposition satisfies the linear equation given by

$$S_{V''}p_i = up_i$$

where $p_i$ are the eigenvectors and $u$ are the eigenvalues corresponding to $p_i$. For the system of equations to have non-trivial solutions, the following equality must be true:

$$S_{V''} - uI_n = 0$$

Eigenvalues are found by substituting the eigenvalues back into equation 44. In PCA, the eigenvectors are used as the new basis space axes, while the eigenvalues are used to sort the axes in order of decreasing variances.

The set of principal components is given by $P$ where the $i^{th}$ column, $P_{*,i}$, is the $i^{th}$ principal component, $p_i$.

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix}$$
3.2.5 Determination of Cosine Correlation Scores for All Spectral Vectors with Respect to the PCA Reference Vectors

The cosine correlation score is defined as the cosine of the angle between two vectors. It is used as a measure of the similarity of the two vectors. Cosine correlation scores are calculated between each vector in the dataset and each basis space vector. The cosine correlation score between each reference vector, \( \mathbf{p}_i \), and a mean-centered spectral vector, \( \mathbf{v}_i'' \) is given by \( \gamma_{j,i}' \), while the full set of cosine correlation scores is given by \( \mathbf{\Gamma} \).

Each row in \( \mathbf{\Gamma} \) contains the cosine correlation scores for one spectral vector.

\[
\gamma_{j,i}' = \cos \theta_{j,i} = \frac{\mathbf{p}_j \cdot \mathbf{v}_i''}{|\mathbf{p}_j||\mathbf{v}_i''|} = \langle \cos(\mathbf{p}_j \mathbf{v}_i'') \rangle
\]

\[
\mathbf{\Gamma} = \begin{bmatrix}
\gamma_{1,1}' & \gamma_{1,2}' & \cdots & \gamma_{1,m}' \\
\gamma_{2,1}' & \gamma_{2,2}' & \cdots & \gamma_{2,m}' \\
\vdots & \vdots & \ddots & \vdots \\
\gamma_{n,1}' & \gamma_{n,2}' & \cdots & \gamma_{n,m}'
\end{bmatrix}
\]

3.2.6 Reliability Threshold and Deciding Which Eigenvectors to Retain

A set of cosine correlation scores between one vector and each coordinate axis in a basis space has a root sum square equal to one. Thus, for the values in every column of \( \mathbf{\Gamma} \), \( \sum_{i=1}^{n} \cos^2 \gamma_i' = 1 \). If a subset of the scores are used, rather than all of the scores in a column, the result could range from 0 to 1. In ROSI, this fact is exploited and used as a means of determining how well a spectral vector is described by a subset of values. The term ‘reliability score,’ denoted R, is the root sum square of a subset of scores for one spectral vector. As the reliability score approaches one, the spectral vector is more reliability described. The user inputs a desired reliability threshold, from 0 to 1, which is
used to determine which scores to retain for each spectral vector. For each spectral vector, the minimum number of scores are retained to meet the reliability threshold. Scores for each spectral vector are retained in order of decreasing magnitude, as this yields the fewest scores needed for data reduction while meeting the threshold goal.

For each ROSI score vector, $\Gamma'_i$, the absolute values of the scores are sorted from maximum to minimum to generate vector $G_i$. The index order locations of the sorted values in $\Gamma'_i$ is recorded in matrix $S_i$. Thus, vector $G_i$ contains the same data as vector $\Gamma'_i$, but reordered based on $S_i$.

\[
\Gamma'_i = \begin{bmatrix}
\Gamma'_{1,i} \\
\vdots \\
\Gamma'_{n,i}
\end{bmatrix}
\]  
(50)

\[
G_i = \begin{bmatrix}
|\Gamma^*,i|_{max} \\
\vdots \\
|\Gamma^*,i|_{min}
\end{bmatrix}
\]  
(51)

\[
S_i = \begin{bmatrix}
s_{1,i} \\
\vdots \\
s_{n,i}
\end{bmatrix}
\]  
(52)

The following equation can be used to determine when the set of scores has reached the desired reliability threshold, $R$. The reliability score, $r_i$, for each vector in $G_i$ is given by

\[
r_i = \sqrt{\sum_{k=1}^{m} G^2_{k,i}}
\]  
(53)

where $R$ is the desired reliability threshold and $m$ is the minimum number of scores that can satisfy $r_i \geq R$. 

96
A binary matrix, $Q$, is used to record which scores are needed for each spectral vector. In computer programming, the size of a binary matrix is one bit per value recorded. In Matlab specifically, which was used here, the size of a binary matrix is one byte per value recorded. The size of a double precision integer (to 16 decimal places) is 8 bytes per value. It is often more economical for only the needed values to be recorded, alongside a binary matrix that is used in data recovery to determine where within the dataset the scores belong. Thus, the data is reduced farther if only the scores that are needed are double precision integers.

The position in the matrix has a value of one if the score is retained, and a value of zero if it is ignored.

\[
Q = \begin{bmatrix}
q_{1,1} & q_{1,2} & \ldots & q_{1,m} \\
q_{2,1} & q_{2,2} & \ldots & q_{2,m} \\
\vdots & \ddots & \vdots \\
q_{n,1} & q_{n,2} & \ldots & q_{n,m}
\end{bmatrix}
\]  

(54)

3.2.7 Data Compression

Table 1 lists the variables that need to be retained in order to recalculate the dataset after data compression. The number of data points in each variable, as well as the number of bytes per value are also listed. Note that for the matrices $\Gamma'$ and $P$, only partial matrices are retained. Those partial matrices contain only the scores which are needed to achieve the desired reliability threshold. The binary matrix, $Q$, is used to record which scores are needed for each spectral vector, and as such, is used to position the retained values within $\Gamma'$ and $P$. 

97
Table 1. Variables Needed for Data Recalculation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Number of Data Points</th>
<th>Bytes per Data Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>$y$</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>$n$</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>$a$</td>
<td>$x \cdot y$</td>
<td>8</td>
</tr>
<tr>
<td>$a_{mc}$</td>
<td>$x \cdot y$</td>
<td>8</td>
</tr>
<tr>
<td>$\overline{v}'$</td>
<td>$n$</td>
<td>8</td>
</tr>
<tr>
<td>$Q$</td>
<td>$x \cdot y \cdot n$</td>
<td>1</td>
</tr>
<tr>
<td>$\Gamma'$</td>
<td>(number of scores needed, $s$)</td>
<td>8</td>
</tr>
<tr>
<td>$P$</td>
<td>(number PCs needed, $t \cdot n$)</td>
<td>8</td>
</tr>
</tbody>
</table>
3.2.8 Recalculating the Original Vectors

Because only portions of $\Gamma'$ and $P$ are recorded, the binary matrix $Q$ is used to shape the lists of values and place the scores in the appropriate locations within recalculated $\Gamma'$ and $P$ matrices. Any values which were zeros in $Q$ (scores that were not needed/not retained) can be filled in with zeros in $\Gamma'$ and $P$. The variables $x$, $y$, and $n$ hold the number of dimensions in the original hyperspectral image dataset, with $x$ and $y$ being the spatial dimensions, and $n$ being the number of spectral dimensions. These values are sometimes needed to reshape the recalculated spectra to the original hyperspectral image dimensionality.

The original spectral vectors can be recalculated based on the reference vectors, $P_l$, the ROSI scores, $\Gamma'_l$, the original amplitudes, $a$, the mean-centered amplitudes, $a_{mc}$, and the mean-centering correction, $v'$. The formula for recalculating a spectral vector is given by

\[
V_k = \sqrt{\frac{\sum_{i=1}^{n} \Gamma'^T_l p_i |u|}{\sum_{i=1}^{n} \Gamma'^T_l p_i}} + \sqrt{\frac{\sum_{i=1}^{n} \Gamma'^T p_i |u|}{\sum_{i=1}^{n} \Gamma'^T p_i}}.
\]  

The equation to determine the total number of bytes which need to be retained in the data reduction method is given by

\[
\text{bytes retained} = 24 + 16xy + 8n + xyn + 8s + 8tn.
\]
3.3 Methodology of PCA as a Data Reduction Method

For the purposes of comparison, principal component analysis (PCA) as a data reduction method is also discussed. To perform PCA, the first step is to mean center the dataset. The mean centering for each spectral vector in the dataset is given by

\[ \mathbf{w}_{t,*} = \mathbf{v}_{t,*} - \overline{\mathbf{v}} \quad . \quad (57) \]

For PCA, the covariance matrix is generated using the mean centered data. The calculation for the \( ij^{th} \) value in the covariance matrix is given by substituting the dot product of \( \mathbf{w}_i \) and \( \mathbf{w}_j \) into

\[ \sigma^2_{w_i,w_j} = \frac{1}{n-1} \mathbf{w}_i \mathbf{w}_j^T \quad (58) \]

where the superscript \( T \) denotes a transposition of the data. The full covariance matrix is given by

\[ \mathbf{S}_w \equiv \frac{1}{n-1} \mathbf{W} \mathbf{W}^T \quad . \quad (59) \]

\( \mathbf{S}_w \) is a square matrix of order \( n \) that describes the relationships between all possible pairs of measurements in the dataset. The variances of each measurement are the diagonal values, while the covariances between measurements are the off-diagonal values. Eigenvectors and eigenvalues are found for the covariance matrix. Eigenvectors are vectors that do not change direction when multiplied by the dataset, although they may be scaled by a factor, \( a \). Eigendecomposition satisfies the linear equation given by

\[ \mathbf{S}_w \mathbf{q}_i = a\mathbf{q}_i \quad (60) \]

where \( \mathbf{q}_i \) are the eigenvectors and \( a \) are the eigenvalues corresponding to \( \mathbf{q}_i \). For the system of equations to have non-trivial solutions, the following equality must be true:
$S_w - aI_n = 0$  \hspace{1cm} (61)

Eigenvectors are found by substituting the eigenvalues back into equation 59. In PCA, the eigenvectors are used as the new basis space axes, while the eigenvalues are used to sort the axes in order of decreasing variances.

The set of principal components is given by $Q$ where the $i^{th}$ column, $Q_{*,i}$, is the $i^{th}$ principal component, $Q_i$.

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n,1} & q_{n,2} & \cdots & q_{n,n} \end{bmatrix} \quad (62)$$

The PCA scores for each mean centered spectral vector are the portions of the length of the spectral vector in the direction of each principal component axis. In other words, the principal component axes are used as coordinate axes in Cartesian space and the set of scores for each mean centered spectral vector is the position of the head of the vector in that space. Scores are calculated between each vector in the dataset and each principal component axis. Each score is given by $\epsilon_{j,i}$.

$$\epsilon_{j,i} = q_j \cdot w_i \quad (63)$$

### 3.4 Results of ROSI as a Data Reduction Method

#### 3.4.1 Four Component Raman Point Mapping Dataset

**Experimental**

A Raman point mapping dataset was collected of a four component sample consisting of quartz, sulfur, lead chromate, and potassium dichromate. The sample was illuminated by a 785nm laser having an intensity of 1.5 watts (Process Instruments),
which was focused on the sample using the 10x infinity corrected objective (Olympus, 0.30 NA). The sample was placed on a motorized stage controlled by stepper motors (MTS25-Z-8, Thor labs). Raman scattered light passed through holographic notch filters (Super Notch Plus, Kaiser Optical), but these filters blocked the laser light. Raman scattered light reached a fiber optic cable (CeramOptec 4708), and was transmitted to and characterized by a monochrometer (Chromex, 500is/sm) equipped with a liquid nitrogen-cooled charge coupled detector (CCD, Roper; now Princeton Instruments, EEV 100x1340B). A spectrum was acquired for each position of the motorized stage. The motorized stage followed a raster pattern at increments of 0.05mm until each region of the sample was covered. A total of 3366 pixels were collected (66 x 51), each at 1340 wavelengths ranging from about 200 to 1250 cm\(^{-1}\). The Raman image was collected on a 2.5mm by 3.25mm square region of the sample. All analyses of the image data were performed using routines written in Matlab R2013a (The Mathworks, Inc.).

This sample was chosen such that the chemistries at each pixel location are known. ROSI was performed without the use of this knowledge, but that data was available as a comparison point for the ROSI resultant image. **Figure 21A** shows the brightfield image of the sample, while **Figures 21B and C** show a map of the chemistries in the sample and representative spectra for each of the four components, respectively. The map was known because the sample was arranged with known components. Although the chemical information at each pixel location was known, that information was not used to assist ROSI.

Principal component analysis (PCA) was also performed on the dataset for comparison. PCA is performed by finding the principal component axes on the mean
Figure 21. Brightfield Image of the Four Component Raman Point Mapping Dataset and the Known Chemistries by Pixel Location. A shows the brightfield image, B shows a map of known chemistries by pixel location, and C shows the representative spectrum of each component.
centered data. The axes are ordered from most to least data variability and the subsequent scores are the projected lengths of the spectral vectors along each axis.

**Results**

The size of the original dataset was 36,083,520 bytes (3366 pixels x 1340 wavelengths x 8 bytes/value). As the number of scores which need to be retained (components of $\Gamma'$) and the number of principal components which need to be retained (components of $P$) vary based on the dataset and the desired reliability threshold, **Table 2** shows data compression metrics at various reliability thresholds.
Table 2. Bytes of Variables Needed for Data Recalculation

<table>
<thead>
<tr>
<th>Reliability Threshold</th>
<th>Bytes</th>
<th>Percent Original Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50000</td>
<td>4,762,768</td>
<td>13.20</td>
</tr>
<tr>
<td>0.68269</td>
<td>4,810,400</td>
<td>13.33</td>
</tr>
<tr>
<td>0.80000</td>
<td>4,838,408</td>
<td>13.41</td>
</tr>
<tr>
<td>0.90000</td>
<td>4,881,480</td>
<td>13.53</td>
</tr>
<tr>
<td>0.95450</td>
<td>5,018,328</td>
<td>13.90</td>
</tr>
<tr>
<td>0.99730</td>
<td>5,503,840</td>
<td>15.25</td>
</tr>
<tr>
<td>0.99900</td>
<td>5,881,696</td>
<td>16.30</td>
</tr>
<tr>
<td>0.99990</td>
<td>17,914,528</td>
<td>49.65</td>
</tr>
<tr>
<td>0.99994</td>
<td>20,230,832</td>
<td>56.07</td>
</tr>
<tr>
<td>0.99999</td>
<td>26,947,912</td>
<td>74.68</td>
</tr>
<tr>
<td>1.00000</td>
<td>55,021,544</td>
<td>152.48</td>
</tr>
</tbody>
</table>
Additionally, the dataset was visualized by scaling each of the first three sets of scores from 0 to 255 and using them as red, green, and blue intensities, respectively, at each pixel location, as seen in Figure 22A. This visualization, however, did not reach the reliability threshold for every pixel location. Figure 22B shows the reliability of each pixel in the image, with a reliability of 1 mapped to white, and a reliability of 0 mapped to black. As the initial visualization did not reach the desired reliability threshold of 0.954 for each pixel location, Figure 23 shows multiple ROSI images, each with three subsequent sets of scores mapped to red, green, and blue. Once a pixel has reached the reliability threshold, it appears black in subsequent images. Once all pixels have reached the reliability threshold, no further images are displayed. The 12 images represent 36 sets of scores, or up to 33 values per spectrum. This is much less than the original number of 1340 values per spectrum.

Principal component analysis was also used to visualize the dataset for comparison, results of which can be seen in Figure 24.
**Figure 22.** Visualization of ROSI Result. In **A**, the first three score sets were scaled from 0 to 255 and used as red, green, and blue intensities, respectively, at each pixel location. **B** shows the reliability of the visualization of ROSI result. The reliability was calculated for each pixel using only the first three sets of scores and mapped from a reliability of 1, or fully described, at white, to a reliability of 0, or poorly-described, at black. The mean of the reliability image is 0.9975±0.0595.
Figure 23. Visualization of ROSI Result until the Reliability Threshold of 0.954 is Reached. These images were each made from mapping three sets of scores to red, green, and blue, respectively. Each subsequent images uses the three subsequent score sets, as listed above each image. Once a pixel has reached a reliability of 0.954, it appears black in all subsequent images.
Figure 24. Visualization of PCA Result. A was made from mapping the first three sets of scores to red, green, and blue, respectively. B shows the reliability for each pixel in (A), which was calculated as the square root of sum squared of the first three scores divided by the square root of the sum squared for all scores. The mean of the reliability image is 0.9934±0.0189.
3.4.2 Four Component Raman Filled Molar Fraction Space Dataset

**Experimental**

The four component Raman filled molar fraction space dataset was manufactured to analyze the data compression metrics of ROSI for various reliability thresholds and signal-to-noise ratios (SNR). Four representative Raman spectra were used (quartz, sulfur, lead chromate, and potassium dichromate) and are shown in Figure 21C, as the representative spectra from the point mapping dataset were used. Each spectrum (pixel location) in the manufactured dataset is a linear combination of the four components.

All molar fractions were an increment of 1/30 from 0 to 1. Additionally, the four molar fractions for each pixel have a sum of 1. A total of 5,456 molar fraction combinations are found that fit these criteria. Thus, the hyperspectral image dataset contains 5,456 unique spectra, each at a different pixel location. This dataset was replicated 11 times, and to each replicate, random noise was added to achieve desired signal-to-noise ratios. The 11 datasets were each run at various reliability thresholds. All dataset design and analyses were performed using routines written in Matlab R2013a (The Mathworks, Inc.).

**Results**

The size of the original dataset was 58,488,320 bytes (5456 pixels x 1340 wavelengths x 8 bytes/value). Table 3 contains percentages of the original dataset size which need to be retained in order to recalculate the data after data reduction and Figure 25 shows a plot of the same information. Percentages are shown at various signal-to-noise ratios and various reliability thresholds.
Table 3. Percentage of Original Data Size at Various Reliability Thresholds and Signal-to-Noise Ratios

<table>
<thead>
<tr>
<th>SNR: Target (Actual)</th>
<th>Reliability Threshold:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5000</td>
</tr>
<tr>
<td>~4 (3.9)</td>
<td>35.17</td>
</tr>
<tr>
<td>~10 (10.0)</td>
<td>16.90</td>
</tr>
<tr>
<td>~20 (19.4)</td>
<td>13.83</td>
</tr>
<tr>
<td>~30 (30.2)</td>
<td>13.13</td>
</tr>
<tr>
<td>~40 (40.1)</td>
<td>13.04</td>
</tr>
<tr>
<td>~50 (50.2)</td>
<td>12.87</td>
</tr>
<tr>
<td>~60 (59.7)</td>
<td>12.83</td>
</tr>
<tr>
<td>~70 (69.9)</td>
<td>12.83</td>
</tr>
<tr>
<td>~80 (80.7)</td>
<td>12.82</td>
</tr>
<tr>
<td>~90 (90.8)</td>
<td>12.82</td>
</tr>
<tr>
<td>~100 (99.7)</td>
<td>12.82</td>
</tr>
</tbody>
</table>
Figure 25. Percentage of Original Data Size at Various Reliability Thresholds and Signal-to-Noise Ratios
Additionally, the dataset was visualized by scaling each of the first three sets of scores from 0 to 255 and using them as red, green, and blue intensities, respectively, at each pixel location, as seen in Figure 26A. This visualization, however, did not reach the reliability threshold for every pixel location. Figure 26B shows the reliability of each pixel in the image, with a reliability of 1 mapped to white, and a reliability of 0 mapped to black. As the initial visualization did not reach the desired reliability threshold of 0.954 for each pixel location, figure 26C shows multiple ROSI images, each with three subsequent sets of scores mapped to red, green, and blue. One a pixel has reach the reliability threshold, it appears black in subsequent images.
Figure 26. Visualization of ROSI Results of the Four Component Point-Mapping Dataset. A shows an image generated from the first three sets of scores in ROSI. Each score set was scaled from 0 to 255 and used as red, green, and blue intensities, respectively. B shows the reliability of the image in A. It was calculated from the first three scores for each pixel location and mapped from a reliability of 1, or fully described, at white, to a reliability of 0, or poorly-described, at black. The values have a mean of 0.977±0.060. As the reliability threshold of 0.954 was not reached for every pixel in the first image, C shows a series of images, each from three subsequent scores sets mapped to red, green, and blue with the three score set numbers below each image. Once a pixel has reached a reliability of 0.954, it appears black in all subsequent images.
3.5 Conclusion

Reduction of Spectral Images (ROSI) is used as a data reduction method. ROSI is fully-automated, user-unassisted, and is effective in classifying pixels within the dataset according to subtle differences in spectral shape. The results are in a suitable format for immediate interpretation or further chemometric processing. The chemical identities of spectra are preserved at a desired threshold so that most chemometric, signal processing, and image processing algorithms can be run on the data after reduction.

This work focused on data reduction of chemical domain types. In the future, the method could be improved upon by expanding the methodology to reduce data sparsity in spatial domains.
REFERENCES


CHAPTER IV
IMPROVEMENT OF A SPATIALLY-MULTIPLEXED RAMAN HYPERSPECTRAL IMAGING SYSTEM

In wide-field imaging, the intensities of all spatial points are gathered at once by an array detector, such as a charge coupled device (CCD), for one wavelength band. The sample is globally illuminated, but only a small fraction of that radiation reaches any single element of the array detector. For noise that is independent of the strength of the signal, the signal-to-noise ratio improves when radiation from a larger area reaches a detector element, as the signal increases alongside the surface area. If measurable elements, such as pixel locations or wavelength bands, are grouped and a combined measurement is collected, the signal-to-noise ratio is higher in that measurement than it would be for the measurement of a single element. It is possible to assign measurable elements to a series of groups, such that the contribution of each individual group member can be mathematically determined. This practice is known as multiplexing.

The main advantage of multiplexing, known as Fellgett’s advantage, is to increase the signal on the detector to increase the signal-to-noise ratio. This is brought about by the fact that the signal is comparatively higher than the noise when grouped. Another
advantage of multiplexing is the Jacquinot advantage, in which higher throughput of light is allowed in the absence of slits or apertures needed to achieve resolution. Connes advantage refers to the higher precision that is achieved when no or few moving parts are used. However, not all multiplexing schemes realize all three advantages due to hardware design and limitations.

Compact instrumentation that can collect Raman hyperspectral images has been developed. The spatially multiplexed Raman hyperspectral imager utilizes a component called the digital micro-mirror device (DMD). The DMD uses a series of mirrors that can be individually controlled to reflect the Raman scattered light in known patterns. A series of spectra are collected with different mirrors (pixels) turned on and off so that a multiplexing method can be used to generate a hyperspectral image. Because the signal is spatially multiplexed, there is significant improvement in the signal-to-noise ratio and the acquisition times are faster compared to traditional data collection methods.

In the work presented here, the spatial resolution of the imager has been improved. The mirrors on the DMD are in diamond pixel projection, whereas modern computer displays use the Manhattan pixel projection, thus the original software controlling the mirrors and displaying the results were in Manhattan pixel projection. The original pixel projection conversion technique grouped the mirrors into sets of 18 such that one pixel was made from a 3x6 array of micromirrors. Here, the mirrors have been ungrouped such that there are now 18 pixels in every area that used to have only one pixel. The spatial resolution improvement of the imager was improved such that features with a width of 2.19μm could be resolved, whereas the previous limit was 7.81μm.
4.1 Significance

The significance of the instrumentation is that it’s a proof of concept for a handheld Raman hyperspectral imaging device. Rugged, low-cost, handheld Raman devices that can collect one spectra at a time already exist.\textsuperscript{6,7} In combination with the DMD Hadamard Transform methodology, a handheld Raman hyperspectral imager is possible. The significance of the work presented here is that the spatial resolution of the resultant hyperspectral image has been improved.

4.2 Multiplexing

Multiplexing\textsuperscript{2} is a method by which multiple signals are combined so that they can be detected or transmitted by an expensive resource. Over the sampling area, the spectra from different pixel locations are combined. The detector records one spectrum from the combination of pixel locations. The pattern of pixels which are included in the combination is called a mask. A series of \( n \) suitable masks are applied to the sample and a combined spectrum is recorded for each. Upon measuring \( n \) suitably chosen combined spectra, \( n \) different individual spectra can be calculated.

The advantage of multiplexing most strongly seen in this multiplexing scheme is Fellgett’s advantage, where the signal-to-noise ratio is improved.\textsuperscript{2} All detectors output unwanted noise, but in multiplexing, the signal is stronger. In point mapping, line scanning, and wide-field imaging a certain amount of light is available to be measured for each pixel location or image frame. In multiplexing, a point detector is recording a combined spectrum for a significant portion of the sample, meaning much more light reaches the detector.
First described by Yates in 1935, a weighing design is a scheme by which objects are grouped for multiplexing. Yates described the weighing of four objects using a balance that imposes an error, $e$, on each measurement. Each measurement, $m_i$, is the combination of the weight, $w_i$, and the error, $e_i$. Thus, each measurement is given by

$$ m_i = w_i + e_i \quad . \quad (64) $$

Suppose the balance is a double pan balance and the following four measurements were made, where a positive $w_i$ value means the object was placed in the left pan and a negative $w_i$ value means the object was placed in the right pan:

$$ m_1 = w_1 + w_2 + w_3 + w_4 + e_1 \quad (65) $$
$$ m_2 = w_1 - w_2 + w_3 - w_4 + e_2 \quad (66) $$
$$ m_3 = w_1 + w_2 - w_3 - w_4 + e_3 \quad (67) $$
$$ m_4 = w_1 - w_2 - w_3 + w_4 + e_4 \quad (68) $$

Summing the four measurements gives

$$ \sum_{i=1}^{4} m_i = 4w_1 + \sum_{i=1}^{4} e_i \quad . \quad (69) $$

Solving for $w_1$ yields

$$ w_1 = \frac{1}{4} \left( \sum_{i=1}^{4} m_i - \sum_{i=1}^{4} e_i \right) \quad . \quad (70) $$

Because the variance is the squared deviation from an expected value, the variance of $ce$, where $c$ is a constant and $e$ is error, is $c^2$ times the variance of $e$. The variance of a sum of independent random variables is the sum of the individual variances. Thus, while the variance of the individual measurements is given by $\sigma^2$, the variance of the combined
measurement is $4\sigma^2/16$ or $\sigma^2/4$. This means that the proposed weighing design reduced the mean square error on the measurement of $w_1$ by a factor of four.

When multiplexing for the collection of a hyperspectral image, each measurement is a combined spectrum of a group of pixels, rather than the weight of an object, but the observed signal-to-noise ratio benefits hold true.

4.2.1 Cyclic S-Matrix Multiplexing

In Yates’ example, multiplexing was performed using a pan balance, in which the objects could be placed in the left pan, right pan, or not included. Thus, values of +1, -1, and 0 can be used in a pan balance weighing design. For our purposes, the unknown spectra can be either included or not included, but not easily subtracted. Therefore, a weighing design is needed that uses only values of +1 and 0.

A cyclic S-matrix, $S_n$, is said to be of order $n$ if the size of the matrix is $n \times n$. The first row of a cyclic S-matrix is a pseudo-random sequence which contains $(n-1)/2$ zeros and $(n+1)/2$ ones, where $n$ is the number of values. The matrix is cyclic because each subsequent row is obtained by shifting the previous row one place, with overflow filling in the gap that was created on the opposite side. A cyclic S-matrix of order $n$, where $n$ is the number of unknown spectra or pixels, was used as the weighing design. Each row in $S_n$ was used to determine which pixel locations were included in that measurement. A total of $n$ combined spectra were taken, one for each row in $S_n$.

When using an S-matrix, $S_n$, as the weighing design, where there are $n$ unknown spectra, the mean square error in each unknown is reduced by a factor of $(n + 1)^2/(4n)$,
or about $n/4$. This leads to the signal-to-noise increasing by a factor of $(n + 1)/(2\sqrt{n})$, or about $\sqrt{n}/2$.\textsuperscript{2,9}

An example cyclic S-matrix of order 7, $S_7$, is given by

$$S_7 = \begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0
\end{bmatrix}. \quad (71)$$

In this small example, the image would have seven pixels. Each row in the matrix is used to make a single mask, where a pixel is included if the value is one, and omitted if the value is zero. Figure 27 shows the $S_7$ matrix, followed by the seven masks made from this matrix. Each mask is reshaped to 3x3, with two dead pixels (in gray). Pixels which are included in the measurement have a value of one in the $S_7$ matrix and are shown in white in each mask.
Figure 27. A Matrix Configuration Showing a 7x7 Circulant S-Matrix. Each row in the matrix is reshaped into a 3x3 array with 2 dead pixels (in gray).
In the example of our seven pixel system, the seven measurements are as follows:

\[
\begin{align*}
  m_1 &= p_1 + p_2 + p_4 + p_7 + e_1 \quad (72) \\
  m_2 &= p_1 + p_3 + p_6 + p_7 + e_2 \quad (73) \\
  m_3 &= +p_2 + p_5 + p_6 + p_7 + e_3 \quad (74) \\
  m_4 &= p_1 + p_4 + p_5 + p_6 + e_4 \quad (75) \\
  m_5 &= +p_3 + p_4 + p_5 + p_7 + e_5 \quad (76) \\
  m_6 &= +p_2 + p_3 + p_4 + p_6 + e_6 \quad (77) \\
  m_7 &= p_1 + p_2 + p_3 + p_5 + e_7 \quad (78) \\
\end{align*}
\]

where \( m_i \) is the \( i^{th} \) measurement, with error \( e_i \), and \( p_j \) is the contribution of the spectrum at pixel \( j \).

For the purposes of this method, the number of pixels, \( n \), is confined to \( 2^k - 1 \) where \( k \) is a whole number. This constraint on the number of unknown values simplifies the process of solving for the contribution to the spectra from each individual pixel location. Estimating the spectral contribution from pixel \( j \) can be guided by the \( j^{th} \) row of the weighing design, \( S_n \).

For example, the first row in \( S_7 \) is \([1 1 0 1 0 0 1]\), meaning that the spectrum of pixel \( p_1 \) contributes to \( m_1, m_2, m_4 \), and \( m_7 \), but is omitted from \( m_3, m_5 \), and \( m_6 \). When estimating \( \hat{p}_1 \) from \( p_1 \), measurements \( m_1, m_2, m_4 \), and \( m_7 \) are summed, while measurements \( m_3, m_5 \), and \( m_6 \) are subtracted. The contributions of all of the spectra other than \( p_1 \) cancel out. In other words, when estimating \( \hat{p}_1 \), the index locations of 1s and 0s are found for the first row of \( S_7 \). Measurement numbers matching index locations of 1 are multiplied by +1, while the measurement numbers with index locations of 0 are multiplied by -1. All of the measurements are then summed. Thus, the seven measurements in order are multiplied by \([+1 +1 -1 +1 -1 -1 +1]\) and summed.
\[ m_1 + m_2 - m_3 + m_4 - m_5 - m_6 + m_7 \]

\[
\begin{array}{cccc}
p_1 & +p_2 & +p_4 & +p_7 + e_1 \\
+p_1 & +p_3 & +p_6 & +p_7 + e_2 \\
-p_2 & -p_5 & -p_6 & -p_7 - e_3 \\
= +p_1 & +p_4 & +p_5 & +p_6 + e_4 \\
-p_3 & -p_4 & -p_5 & -p_7 - e_5 \\
-p_2 & -p_3 & -p_4 & -p_6 - e_6 \\
+p_1 & +p_2 & +p_3 & +p_5 + e_7 \\
\end{array}
\]  

\[ = 4p_1 + \sum_{i=1}^{7} e_i \]  

Given that the number 1 appears in each row or column \((n+1)/2\) times, the sum needs to be divided by \((n+1)/2\) to solve for the individual pixel contribution.

\[
\hat{p}_1 = \frac{2}{7+1}(m_1 + m_2 - m_3 + m_4 - m_5 - m_6 + m_7) \quad (80)
\]

\[
\hat{p}_1 = \frac{2}{7+1}(4p_1 + \sum_{i=1}^{7} e_i) \quad (81)
\]

\[
\hat{p}_1 = p_1 + \frac{1}{4} \sum_{i=1}^{7} e_i \quad (82)
\]

Thus, when estimating the contribution of the spectrum at pixel \(j\), the \(j^{th}\) row in \(S_n\) is used. In general terms, the estimation of the contribution of the spectrum at pixel \(j\) is given by

\[
\hat{p}_j = \frac{2}{n+1} \sum_{i=1}^{n} (2S_{n,j,i} - 1)m_i \quad . \quad (83)
\]

where \(S_{n,j,i}\) is the value at the \(j^{th}\) row and \(i^{th}\) column of \(S_n\) and \((2S_{n,j,i} - 1)\) converts each 0 to -1 without changing the +1 values.
4.3 Digital Micromirror Device

In early multiplexed imaging systems, the masks were physical objects, often made from cutting slits in a brass plate. Cyclic weighing designs were often used because only a single mask with $2n-1$ elements was needed. A region of $n$ elements was used for each measurement and the brass plate was moved by the width of one element between each measurement.\textsuperscript{2,10,11} These early systems were often problematic in that the brass plate was prone to jamming, but recent developments in technology have overcome this limitation. In recent years, multiplexed imaging systems have become more common.

In the hyperspectral imaging system described here, the way in which the masks are imposed upon the sample is by use of a digital micromirror device (DMD). The DMD is an array of micro-electrical-mechanical mirrors. Each micromirror can be electronically controlled to exhibit one of two states: ‘on’ state or ‘off’ state. The states are accomplished by moving each mirror from $+12^\circ$ from parallel to $-12^\circ$ from parallel, as shown in Figure 28A. The mirrors can be individually controlled and switched in real time to display the series of masks generated through the Hadamard transform multiplexing weighing design.\textsuperscript{12}

The DMD utilized was the Digital Light Processor DLP3000 0.3 WVGA Series 220, which was developed by Texas Instruments. The DLP3000 is a semiconductor device with an array of aluminum micrometer-sized mirrors. Each micromirror is a square of 7.637μm by 7.637μm, as shown in Figure 28B.
Figure 28. DMD Mirrors
The mirrors on the DMD are arranged in a 684 x 608 diamond pixel projection (Figure 29A). In diamond pixel projection, the odd numbered pixels in each column are offset half a pixel in both the x and y dimensions compared to the even numbered pixels. This differs from modern computer displays which are in Manhattan pixel projection, as shown in Figure 29B. Thus, the masks to control the mirror configuration and the resultant image are displayed in Manhattan pixel projection. This distinction was the basis for the improvement of spatial resolution and will be discussed further in later sections.

### 4.4 Multiplexed Hadamard Transform Raman Hyperspectral Imaging System

Individual mirrors or a group of several mirrors can be considered one element or pixel in the weighing design, $S_n$. In other words, each pixel in the resultant image can be defined by one individual mirror or by a group of mirrors. A value of one in the cyclic matrix corresponds to a pixel being included in the measurement, and a value of zero corresponds to a pixel being disregarded for that measurement.

**Figure 30** shows a schematic of the multiplexed Hadamard transform Raman imaging system. The sample is globally illuminated by a 785nm laser. The light which has been scattered by the sample is comprised of both Raleigh scattered light and Raman scattered light. The light passes through an infinity corrected objective lens and is then filtered using a long-pass edge filter, which filters out the Raleigh scattered light and allows the Raman scattered light to pass. Radiation from the sample is focused on the DMD.
Figure 29. Diamond Pixel Projection A and Manhattan Pixel Projection B. The mirrors on the DMD utilize diamond pixel projection, while modern computer displays utilize Manhattan pixel projection.
Figure 30. Spatially-Multiplexed Raman Hyperspectral Imager Instrument Setup
The DMD is positioned such that mirrors in the ‘on’ state reflect light towards the detector and are measured, whereas mirrors in the ‘off’ state reflect light towards a beam block and are disregarded. A series of weighing designs using a cyclic S matrix are displayed on the DMD. The mirror configuration series is based upon the weighing design masks, as previously described. A combined spectrum is recorded for each mirror configuration. For a weighing design of order $n$, the detector collects $n$ measurements. The DMD micromirrors are controlled by sending bitmap files via a graphical user interface application from Texas Instruments.

Light reflected off of micromirrors in the ‘on’ state is homogenized and focused onto a fiber optic bundle. The light from the fiber is angularly dispersed using a spectrograph and focused onto a CCD detector. The intensity from each pixel column position on the detector corresponds to a wavelength. All analyses were performed using routines written in Matlab R2013a (The Mathwork, Inc.).

4.4.1 The Need for a Manhattan to Diamond Conversion Technique

As stated previously, the array of micromirrors on the DMD are in diamond pixel projection, whereas modern computer displays are in Manhattan pixel projection. The masks to control the mirrors are generated on a computer using a computer display with Manhattan pixel projection. The masks are then displayed on the DMD using diamond pixel projection. The resultant hyperspectral image is viewed using a computer display, once again in Manhattan pixel projection. This means that a conversion between the two pixel projections must take place.
4.4.2 Original Manhattan to Diamond Conversion Technique: The 3x6 Super-Pixel Approach

In the original conversion technique, each pixel was a super-pixel comprised of a group of 18 micromirrors. (The term super-pixel is used to describe a pixel made from multiple micromirrors.) In the Manhattan pixel projection, each pixel was a 3x6 group of mirrors, which appeared to be rectangular. With this super-pixel, when the $x$ and $y$ position of each Manhattan pixel was directly used as the $x$ and $y$ position of each diamond pixel, the super-pixel appeared roughly square. Figure 31 shows an array of nine super-pixels utilizing this conversion technique. Part A shows the Manhattan projection and part (B) shows the diamond projection.

The following equations are used for each pixel location to calculate the diamond pixel projection locations from the Manhattan pixel projection locations:

\[ x_D = x_M \]  \hspace{1cm} (84)
\[ y_D = y_M \]  \hspace{1cm} (85)

where $x_D$ and $y_D$ are the column number and row number of each micromirror in diamond projection, respectively, and $x_M$ and $y_M$ are the column number and row number of each micromirror in Manhattan projection, respectively.

This conversion technique was simple to use for two reasons: (1) a user looking at the mask on a computer display could easily imagine how the mask would appear on the DMD micromirror array and (2) the mathematical conversion between the row and column numbers for each pixel was very straightforward, in that the values are the same
Figure 31. Nine Super-Pixels as they appear in Manhattan (A) and Diamond (B) Pixel Projection Using the 3x6 Super-Pixel Conversion Technique.
for both projections. However, the grouping of mirrors into super-pixels significantly reduces the spatial resolution of the resultant hyperspectral image.

4.4.3 Manhattan to Diamond Conversion Technique with Improved Resolution:

The Coordinate Conversion Approach

A new technique to convert between Manhattan and diamond pixel projections was developed. It is accomplished by using each micromirror individually as one pixel. Using this technique greatly improves the spatial resolution capabilities of the multiplexed Hadamard transform Raman hyperspectral imaging system. However, it also necessitates a longer exposure time for each spectrum collected when compared to original conversion technique using the same laser power. This is due to the fact that the area of the sample is smaller, which means that less radiation is focused onto the DMD and less radiation reaches the detector. A longer exposure time compensates and allows the total amount of radiation that reaches the detector to be similar.

In the coordinate conversion approach, the desired layout of pixels is rotated by 45°, utilizing the rotation of the mirrors on the DMD and enabling the spatial relationship between pixels to remain intact. A user can generate masks for the DMD in Manhattan projection on a computer display, and then convert the masks for diamond projection without using super-pixels that reduce the spatial resolution. The number of pixels in a given area is 18 times higher in this approach than in the 3x6 super-pixel conversion technique. As the multiplexing method follows the $2^k-1$ rule, as described in section 4.2.1, and that those pixels can be arranged in a $\sqrt{2^k} \times \sqrt{2^k}$ square with 1 dead pixel
when \( k \) is an even number, this method assumes that the pixels are arranged in a square with the number of pixels along the length and width being equal.

Given desired pixel coordinates in Manhattan pixel projection, the row and column numbers of each pixel in diamond pixel projection are given by the following equations:

\[
x_D = \text{ceiling} \left( \frac{x_M - y_M + \sqrt{n + 1} + 1}{2} \right)
\]

\[
y_D = y_M + x_M - 1
\]

where \( x_D \) and \( y_D \) are the column number and row number in diamond projection, respectively, \( x_M \) and \( y_M \) are the column number and row number in Manhattan projection, respectively, \( n \) is the number of pixels or the number of masks, and \( \text{ceiling} \) means that the quantity in parenthesis is rounded up.

Matrix \( M \) contains \( x,y \) coordinates of pixel locations in the Manhattan pixel projection. \( M \) is given by

\[
M = \begin{bmatrix}
1,1 & 2,1 & 3,1 & 4,1 & 5,1 \\
1,2 & 2,2 & 3,2 & 4,2 & 5,2 \\
1,3 & 2,3 & 3,3 & 4,3 & 5,3 \\
1,4 & 2,4 & 3,4 & 4,4 & 5,4 \\
1,5 & 2,5 & 3,5 & 4,5
\end{bmatrix}
\]

When converted to diamond pixel projection, the \( x,y \) coordinates of the same spatial layout of pixels are given by
Figure 32 shows the Manhattan to diamond pixel projection conversion process using this approach. Part A shows the Manhattan projection of the pixel layout for the 24 pixels with coordinates from matrix M. These 24 pixels are in various colors, whereas ignored or background pixels are gray. This view of the pixel locations is used to view the layout of the masks in Manhattan pixel projection as they would appear on the DMD, as well as the resultant hyperspectral image.

Part B shows the Manhattan view of the same 24 pixels using the converted coordinates from matrix D. When a mask for use with diamond pixel projection is viewed on a computer with Manhattan pixel display, this is how it appears. Thus, when masks are made for use on the DMD, they have this type of layout on the computer display.

Part C shows the diamond pixel projection using pixel coordinates from matrix D. Notice that the 24 pixels of interest are in the same layout as in (A), but are rotated 45° counterclockwise.

$$D = \begin{bmatrix} 3,1 & 4,2 & 4,3 & 5,4 & 5,5 \\ 3,2 & 3,3 & 4,4 & 4,5 & 5,6 \\ 2,3 & 3,4 & 3,5 & 4,6 & 4,7 \\ 2,4 & 2,5 & 3,6 & 3,7 & 4,8 \\ 1,5 & 2,6 & 2,7 & 2,8 \end{bmatrix}. \quad (89)$$
Figure 32. The Manhattan to Diamond Pixel Projection Process. A shows the desired pixel layout of the 24 pixels of interest from matrix M in Manhattan projection. The pixels in the region of interest are shown in various colors, whereas the background region pixels are shown in gray. B shows the Manhattan projection of the pixels once they have been converted into matrix D for use in diamond projection. C shows the diamond projection of matrix D.
4.5 Results

Figure 33 shows a brightfield image of the 1951 U.S. Air Force Target. This target conforms to a standard which was set by the US Air Force in 1951 and is used to test the resolution of optical imaging systems, such as microscopes, cameras, and image scanners. The pattern consists of groups each with three bars, called Ronchi rulings. The line pairs in each group (one bright and one dark) have known widths. Group 6 element 1 has a width of 7.81μm. Group 7 elements 4, 5, and 6 have widths of 2.76μm, 2.46μm, and 2.19μm, respectively.

Hyperspectral images were taken of this target before and after the resolution improvement. Each dataset contained 4095 pixels (64x64 with one dead pixel). The blue and red squares on Figure 33 mark the regions of the target for which hyperspectral images were collected using both the 3x6 mirror super-pixel approach and the coordinate conversion approach, respectively. Figure 34 shows the spatial resolution of the super-pixel approach and Figure 35 shows the spatial resolution of the coordinate conversion approach. In both Figures 34 and 35, part A shows an image taken through the spatially-multiplexed Raman hyperspectral image, and part B shows a brightfield image of the resolution target that has been cropped to match.

As can be seen in the 3x6 super-pixel approach, the smallest line pair that can be resolved is group 6, element 1, which has a line pair width of 7.81μm. Through the use of the coordinate conversion technique, the resolution has improved such that group 7, element 6, with a line pair width of 2.19μm, is resolved. All analyses were performed on the image data using routines written in Matlab R2013a (The Mathworks, Inc.).
Figure 33. A Brightfield Image of the USAF 1951 Resolution Target. The blue square marks the region in which a 4095 pixel hyperspectral image was collected using the 3x6 super-pixel technique. The red square marks the region in which a 4095 pixel hyperspectral image was collected using the coordinate conversion technique.
Figure 34. A) A 4095 Pixel Image of the USAF 1951 Resolution Target Using the 3x6 Super-Pixel Approach. B) The Brightfield image of the Resolution Target Cropped to Match the Hyperspectral Image. Group 6, element 1 is the set of six bars which can be distinguished in A. A line pair width for this element is 7.81μm.
Figure 35. A) A 4095 Pixel Image of the USAF 1951 Resolution Target Using the Coordinate Conversion Approach. B) The Brightfield image of the Resolution Target Cropped to Match the Hyperspectral Image. Group 7, elements 4, 5, and 6 are resolved in A. The line pair width for group 7, element 6 is 2.19μm.
4.6 Conclusion

The spatial resolution of the multiplexed Hadamard transform Raman hyperspectral imaging system was improved such that the capability of resolving elements of size 2.19µm in a hyperspectral image now exists, where previously the smallest elements that could be resolved were 7.81µm.

One component of the system is the DMD, which consists of an array of individually-controllable micromirrors. During the collection of a hyperspectral image, the radiation from the sample is focused on the DMD. The DMD is positioned such that mirrors in the ‘on’ state reflect light towards the detector and are measured, whereas mirrors in the ‘off’ state reflect light towards a beam block and are disregarded.

The issue with the DMD is that the micromirrors are in diamond pixel projection, whereas modern computer displays are in Manhattan pixel projection, thus a conversion technique is needed. In the original technique, each pixel was defined by a 3x6 group of micromirrors. This was easy to use, in that the appearance of the pixel layouts on the DMD could be easily imagined from looking at the pixel layouts in Manhattan pixel projection; however, the spatial resolution suffered as a result.

A Manhattan to diamond pixel projection conversion technique was developed that uses one micromirror as a pixel, instead of 18. It is accomplished by rotating the region of interest by 45°. This enables the user to plan the pixel coordinates on a computer display in Manhattan pixel projection, and then convert the pixel coordinates for use on the DMD in diamond pixel projection.
This technique improved the spatial resolution of the resultant hyperspectral image in that the number of pixels per a given area is 18 times higher.
REFERENCES


12 DLP3000 DLP 0.3 WVGA Series 220 DMD, Dallas, Texas: Texas Instruments, 2012. Revised 2015.
APPENDIX I. REDUCTION OF SPECTRAL IMAGES (ROSI) MATLAB CODE

The following is source code for ROSI that was written for Matlab R2013a (The Mathworks, Inc.).

% rosi.m

mainmenu=menu('Choose an option:','Load a dataset','Perform ROSI','Generate Subsequent ROSI Images','Recalculate Original Spectra','Calculate Data Reduction Ratio','Exit');

while mainmenu<=4
    if mainmenu==1  % % Loads hyperspectral image, requests user input for reliability level, records vector amplitudes

        % % Loads 3-dimensionl dataset and assumes dimensions are x, y, and num_frames
        [filename,pathname]=uigetfile('C:\*.mat','Choose the .mat file in which the data is saved.');
        load(strcat(pathname,filename));
        [x y num_frames]=size(sidata);
        sidata=reshape(sidata,x*y,num_frames);  % reshapes sidata to sidata(total pixels,array)

        % Prompts user to input reliability threshold
prompt='What is the minimum reliability level (R)?
Must be 0 to 1 '

reliability_level=input(prompt);

% Prompt for visualization vs data reduction
vis_rec=menu('Choose an option:','ROSI for Visualization','ROSI for Data Reduction');

% % Records Vector Amplitudes
for j=1:1:size(sidata,1)
    ampl(j,1)=sqrt(sum(sidata(j,:).^2));
end

mainmenu=menu('Choose an option:','Load a dataset','Perform ROSI','Display All ROSI Images','Recalculate Original Spectra','Calculate Data Reduction Ratio','Exit');

elseif mainmenu==2  % % Performs ROSI
dircos=sidata./repmat(ampl,1,size(sidata,2));
% % directional cosine conversion

if vis_rec==1
%if performing ROSI for visualization
%do nothing
else
%if performing ROSI for data reduction
dircos=dircos.^2; %square the direction cosine space
end

mc=mean(dircos); %finds the mean center
dircos_mc=dircos-repmat(mc,x*y,1); %means centers the data

% Finds PCA axes
[col,row]=size(dircos_mc); %records size of dataset
dircos_mc=dircos_mc-repmat(mean(dircos_mc),col,1);
%means centers
covariance=(dircos_mc'*dircos_mc)./(row-1); %finds covariance matrix (eig function needs a square matrix)
[pc,v]=eig(covariance); %finds all pc vectors (eigenvectors) and variances (eigenvalues)
v=diag(v); %variance = diagonal of eigenvalue matrix
[a,indices]=sort(-1*v); %sorts variances in decreasing order;

v=v(indices); %sorted variances
pc=pc(:,indices);  %sorted pc vectors

% dircos_mc=dircos-repmat(mc,x*y,1);  %means centers the data
ampl_dircos_mc=sqrt(sum(dircos_mc'.^2))';  %records the amplitudes of the direction cosine, mean centered data

R=zeros(size(dircos_mc));
for wavelength=1:1:num_frames

    R(:,wavelength)=(sum((repmat(pc(:,wavelength)',size(dircos_mc,1),1).*dircos_mc)')')./(sqrt(sum(pc(:,wavelength)'.^2))*sqrt(sum(dircos_mc'.^2)))';

end

nR=(R-repmat(min(R),x*y,1))./repmat(max(R)-min(R),x*y,1);
R3d=reshape(nR,x,y,size(R,2));

% % Reorders PCs
% Category Reorder
R=R./repmat((sqrt(sum(R'.^2)))',1,size(R,2));  %normalizes set of scores
[Rsort, Rindex] = sort(-abs(R')); % gets index based on absolute intensity
Rindex = Rindex'; % transposes the matrix
Rsort = Rsort';

% Finds reliability thresholds
reliabilities = zeros(size(Rsort));
reliabilities(:, 1) = abs(Rsort(:, 1));
for frame = 2:1:size(Rsort, 2)
    reliabilities(:, frame) = sqrt(sum(Rsort(:, 1:frame)' .^ 2)');
end

% Locations of reliability thresholds needed for >= reliability_level
scores_needed = ones(x*y, num_frames);
[x_arl, y_arl] = find(reliabilities > reliability_level);
for pos = 1:1:size(x_arl, 1)
    scores_needed(x_arl(pos, 1), y_arl(pos, 1)) = 0;
end

scores_needed = [ones(x*y, 1) scores_needed(:, 1:end-1)];
% where scores_needed==1, the scores are needed to achieve reliability_level in order of Rsort

[Rindexsort,Rindexindex]=sort(Rindex');
Rindexsort=Rindexsort';
Rindexindex=Rindexindex';

for px=1:1:x*y

    index_needed_sort(px,:)=scores_needed(px,Rindexindex(px,:));

end

R_sn=R.*index_needed_sort; % contains the scores that are needed to recalculate dataset to reliability_level

bs_data=R_sn~=0; % bs_data==1 where score is needed, bs_data==0 where score in not needed
bs_list=unique(bs_data,'rows'); % lists unique basis spaces. size==(number of basis spaces) x (basis space)

if size(bs_list,1)==1
    [~,reord]=sort((bs_list),'descend');
else
    [~,reord]=sort(sum(bs_list),'descend');

end
R_sort=R(:,reord); %reorders the ROSI scores

% Scales scores from 0 to 1 to be used as RGB values in image
px=size(R,1);
nR=(R_sort-
repmat(min(R_sort),px,1))./repmat(max(R_sort)-
min(R_sort),px,1);
R3d_reordered=reshape(nR,x,y,size(R,2));

% Displays ROSI resultant image
figure;
imshow(R3d_reordered(:,:,1:3));
title('ROSI Reordered PCs');

mainmenu=menu('Choose an option:','Load a dataset','Perform ROSI','Display All ROSI Images','Recalculate Original Spectra','Calculate Data Reduction Ratio','Exit');

elseif mainmenu==3
% Displays subsequent images until reliability threshold is reached by every pixel

rel_accum=sqrt(sum(R_sort(:,1:3)'.^2))';  %reliability of the 1st ROSI image

im_count=1;  %image count

while sum(rel_accum<reliability_level)>0  %while some pixels have not yet reached reliability

    im_count=im_count+1;  %increment image count

    rel_pxs=(rel_accum<reliability_level);  %finds locations of pixels which have not yet met the reliability score

    figure;

    imshow(R3d_reordered(:,:,im_count*3-2:im_count*3).*reshape(repmat(rel_pxs,1,3),x,y,3)); %displays next image, with previously reliable pixels in black

    name=sprintf('rosi_image%d', im_count);  %generates filename

    imwrite(R3d_reordered(:,:,im_count*3-2:im_count*3).*reshape(repmat(rel_pxs,1,3),x,y,3),[pathname ,filename,name,'.tif']);  %writes images to file

    rel_accum=sqrt(sum(R_sort(:,1:im_count*3)'.^2))';  %updates reliability so far

end
mainmenu=menu('Choose an option:','Load a dataset','Perform ROSI','Display All ROSI Images','Recalculate Original Spectra','Calculate Data Reduction Ratio','Exit');

elseif mainmenu==4
    % Recalculate original spectra

    if vis_rec==1
        % if performing ROSI for visualization
        quad_a=sum(R_sn.^2*pc'.^2,2);
        quad_b=2*sum((R_sn*pc').*repmat(mc,x*y,1),2);
        quad_c=sum(mc.^2)-1;

        ampl_dircos_mc_add=(-quad_b+sqrt(quad_b.^2-4*quad_a*quad_c))./(2*quad_a);

        rd=(R_sn*pc').*repmat(ampl_dircos_mc_add,1,num_frames)./repmat(sqrt(sum(((R_sn.^2)*(pc.^2)')))',1,num_frames)+repmat(mc,x*y,1);
    else
        % if performing ROSI for data reduction
\[ rd = \sqrt{\text{repmat}(\text{ampl_dircos_mc}, 1, \text{num_frames}) \cdot (R_{\text{sn}} \cdot \text{pc}') + \text{repmat}(\text{mc}, x \cdot y, 1)}; \]
\[ rd = \text{real}(rd); \]
\[ \text{end} \]

\[ rd = (rd \cdot \text{repmat}(\text{ampl} ./ (\sqrt{\text{sum}(rd' \cdot \cdot 2)})'), 1, \text{num_frames}); \]

\[ \text{cca} = \text{sum}(rd' \cdot \text{sidata'}) ./ (\sqrt{\text{sum}(rd' \cdot \cdot 2)}) \cdot \sqrt{\text{sum}(\text{sidata'} \cdot \cdot 2)}); \]

\[ \text{figure;} \]
\[ \text{plot}(\text{cca}); \]
\[ \text{title('CCA between Original Spectra and Recalculated Spectra');} \]
\[ \text{xlabel('Pixel (Spectral) Number');} \]
\[ \text{ylabel('Cosine of Angle between Original and Recalculated Spectral Vectors');} \]

mainmenu = menu('Choose an option:', 'Load a dataset', 'Perform ROSI', 'Display All ROSI Images', 'Recalculate Original Spectra', 'Calculate Data Reduction Ratio', 'Exit');
elseif mainmenu==5

% Determines the size of data needed in data reduction
to recalculate spectra

% x
x_bytes=whos('x');
bytes(1)=x_bytes.bytes;

% y
y_bytes=whos('y');
bytes(2)=y_bytes.bytes;

% num_frames
num_frames_bytes=whos('num_frames');
bytes(3)=num_frames_bytes.bytes;

% ampl
ampl_bytes=whos('ampl');
bytes(4)=ampl_bytes.bytes;

% pc, but only the PCs that are needed
% number of PCs
needed=size(find(sum(index_needed_sort)>0),2);
bytes(5) = size(find(sum(index_needed_sort)>0),2)*num_frames*8;

% mc
mc_bytes = whos('mc');
bytes(6) = mc_bytes.bytes;

% logical index to record where scores that are needed belong
index_needed_sort = logical(index_needed_sort);
ins_bytes = whos('index_needed_sort');
bytes(7) = ins_bytes.bytes;

% R_sn (scores needed), but only the scores that are actually needed
bytes(8) = sum(sum(index_needed_sort))*8;

if vis_rec == 1
    % if performing ROSI for visualization
    % do nothing
else
    % if performing ROSI for data reduction
    % ampl_diricos_mc
adm_bytes=whos('ampl_dircos_mc');
bytes(9)=adm_bytes.bytes;
end

sidata_bytes=whos('sidata');
sidata_bytes=sidata_bytes.bytes;

rosi_data_compressed_bytes=sum(bytes)
original_dataset_bytes=x*y*num_frames*8
rosi_data_compression_ratio=sum(bytes)/sidata_bytes

mainmenu=menu('Choose an option:','Load a dataset','Perform ROSI','Display All ROSI Images','Recalculate Original Spectra','Calculate Data Reduction Ratio','Exit');

elseif mainmenu==6
    % Exits ROSI main menu and halts
end
end