REDUCING DIMENSIONALITY OF HYPERSPECTRAL DATA WITH DIFFUSION MAPS AND CLUSTERING WITH K-MEANS AND FUZZY ART

Louis du Plessis[*], Rui Xu[†], Steven Damelin[‡], Michael Sears[§] and Donald C. Wunsch II[¶]

Abstract

It is very difficult to analyze large amounts of hyperspectral data. Here we present a method based on reducing the dimensionality of the data and clustering the result in moving toward classification of the data. Dimensionality reduction is done with diffusion maps, which interpret the eigenfunctions of Markov matrices as a system of coordinates on the original dataset in order to obtain an efficient representation of data geometric descriptions. Clustering is done using k-means and a neural network clustering theory, fuzzy ART. The process is done on a subset of core data from AngloGold Ashanti, and compared to results obtained by AngloGold Ashanti's proprietary method. Experimental results show that the proposed methods are promising in addressing the complicated hyperspectral data and identifying the minerals in core samples.

1 Introduction

New spectral imaging techniques are making hyperspectral imaging more accessible. Spectral imaging refers to the process of sampling an image at several different frequencies [12]. Digital colour photography is a form of spectral imaging. The picture is sampled at three different frequencies, one in the blue range of the spectrum and the other two in red and green respectively. Every sample gives a matrix of intensity values, which could be plotted to give a gray-scale image. When all three matrices are blended together a colour image is produced.

Multispectral imaging refers to the process of sampling an image at more frequencies. Images are commonly sampled in the frequency range between 0.4 and 2.5 μm, since this is the range of the optical spectrum where the sun provides useful illumination [12]. In hyperspectral imaging the image is sampled at hundreds of frequencies, compared to only a few for multispectral imaging. Furthermore, the different frequencies in multispectral imaging are usually distributed in an irregular fashion, whereas the bands in hyperspectral imaging are regularly spaced [12].

Because of the regular spacing of narrow bands, a continuous spectrum can be drawn for every pixel in the image. Instead of ending up with a flat two-dimensional matrix of values, we obtain a "hypercube" of data, as shown in Fig. 1. This is where the problem in analyzing and storing hyperspectral data comes in. Having more than a hundred bands for every pixel means having

[*] School of Computational and Applied Mathematics, University of the Witwatersrand, South Africa, *e-mail:* `Laduplessis@gmail.com`

[†] Applied Computational Intelligence Laboratory, Department of Electrical & Computer Engineering, Missouri University of Science and Technology, MO 65409 USA, *e-mail:* `rxu@mst.edu`

[‡] The Unit for Advances in Mathematics and its Applications, Department of Mathematical Sciences, Georgia Southern University, GA 30460 USA, and the School of Computational and Applied Mathematics, University of the Witwatersrand, South Africa. *e-mail:* `damelin@georgiasouthern.edu`

[§] School of Computer Science, University of the Witwatersrand, South Africa, *e-mail:* `michael.sears@wits.ac.za`

[¶] Department of Electrical & Computer Engineering, Missouri University of Science & Technology, Rolla, MO 65409 USA. *e-mail:* `dwunsch@mst.edu`

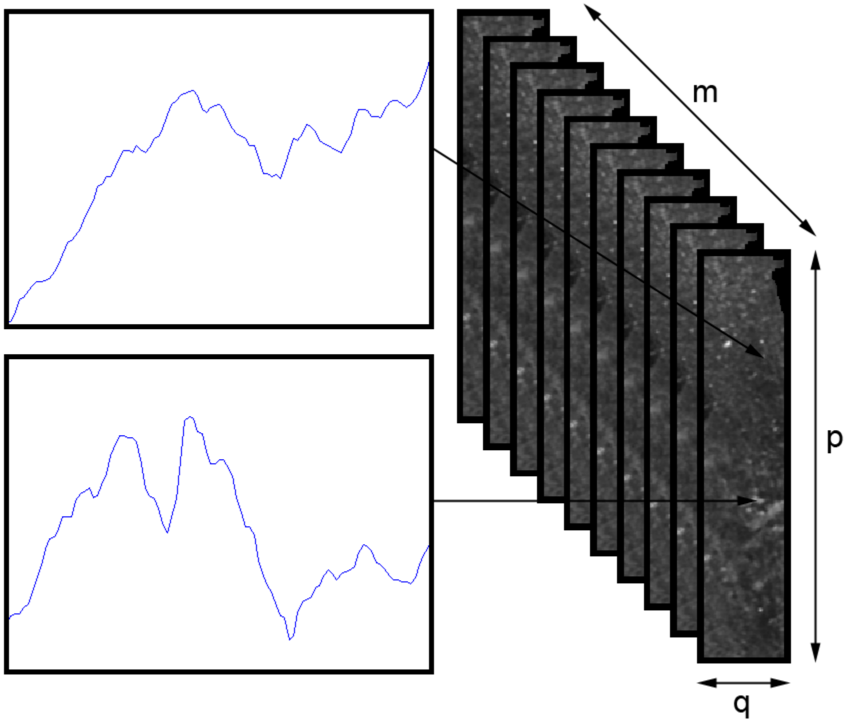


Figure 1: The hyperspectral data cube. For every one of the m frequencies sampled, there is an image of $p \times q$ of intensity values. Similarly, for every one of the pixels in the image, there is a complete spectrum of values. This cube image is generated with data from the HCI.

enormous amounts of data. If the hyperspectral imager scans in d bands, then every pixel of the hyperspectral image can be seen as a d dimensional vector.

Currently, hyperspectral imaging is mainly used in airborne surveillance techniques [6]. Some uses for hyperspectral imaging include crop assessment, environmental applications, and mineral exploitation [3]. The data this paper is concerned with was produced by the Hyperspectral Core Imager (HCI), which was developed by AngloGold Ashanti. The instrument is used to produce hyperspectral images of core samples; long cylindrical pieces of rock drilled in prospective mining sites. A decision on whether to mine or not at a specific site is dependent on the presence of certain minerals within the core samples. Analysis by traditional methods is painstaking and time-consuming, and is usually only complete after the decision on whether to mine or not has been taken.

This has prompted the development of the HCI. However, a reliable, automated method of classification or target detection remains elusive. In this paper we focus on the problem of classification. Due to the nature of the HCI the problem is significantly transformed from its usual form, as used in aerial and satellite imaging. Most hyperspectral systems have a very coarse spatial resolution. However, because of the close proximity of the core sample to the HCI, the spatial resolution of the data we are concerned with is several orders of magnitude higher.

Mixing is a predominant concern in most hyperspectral imaging systems. The term refers to the fact that any one pixel in an image will contain spectral reflectance characteristics from several different materials. The resulting spectrum of the pixel is a mixture of the pure spectra of each of the materials present in the pixel. Usually a linear mixing is assumed, although this is almost never the case. Although one can never obtain an image that consists only of pure pixels [12], regardless of how fine the resolution is, this issue is of much less concern for HCI data. Because the spatial resolution is high compared to other imaging systems, and the core samples consist of minerals in fairly homogeneous groups, we are not concerned with unmixing pixels in this paper. A related problem deals with the fact that the transition between minerals is often gradual, as opposed to a sudden, hard boundary between groups of different minerals.

Here, we address high dimensional hyperspectral data using diffusion maps, which consider the eigenfunctions of Markov matrices as a system of coordinates on the original dataset in order to obtain efficient representation of data geometric descriptions [7, 14, 13]. The major difference between diffusion maps and methods like principal components analysis (PCA) is that in diffusion maps, a kernel is chosen before the procedure. This kernel is chosen by our prior definition of the geometry of the data [7]. In PCA, all correlations between values are considered, while only high correlation values are considered in diffusion maps. Diffusion maps have already been applied in the analyses of protein data [11], gene expression data [22], video sequences [15], and so on, and have achieved attractive performances.

The assumption is that every core sample contains only a few different kinds of minerals so that there is a lot of redundant data. It should therefore be sufficient to have only a few key values per pixel to identify different materials.

The reduced data are then clustered with k-means and a neural network cluster theory, Fuzzy ART (FA) [5], to generate clusters of the potential minerals. The standard built-in k-means routine in Matlab is used for the k-means clusterings. FA is based on Adaptive Resonance Theory (ART) [4, 9], which was inspired by neural modeling research and was developed as a solution to the plasticity-stability dilemma: how adaptable (plastic) should a learning system be so that it does not suffer from catastrophic forgetting of previously-learned rules (stability)? ART can learn arbitrary input patterns in a stable, fast, and self-organizing way, thus overcoming the effect of learning instability that plagues many other competitive networks.

The results are compared to a clustering obtained on the same data by AngloGold Ashanti's proprietary method, which makes use of an endmember extraction algorithm and self-organizing maps [18]. The proprietary method also requires the need to specify the amount of clusters. Since it is impossible to know the amount of mineral clusters in the core, this is undesirable. Although k-means also requires the specification of the amount of clusters, fuzzy ART does not. Experimental results on a subset of hyperspectral data show that the proposed methods are promising in addressing the complicated hyperspectral data and identifying the minerals in core samples. The methods also produce results similar to those obtained by the proprietary method. Initial investigations into reducing the dimensionality of HCI data and then clustering with fuzzy ART were made in [23].

The remainder of this paper is organized as follows. Sections 2 and 3 briefly introduce diffusion maps and fuzzy ART. Section 4 gives a more detailed overview of the Hyperspectral Core Imager (HCI). The experimental results are given in section 5. In addition to real hyperspectral data, the method is also tested on a small artificial sample, to verify its feasibility. The results, as well as related and future work are discussed in section 6 and section 7 concludes.

2 Diffusion Maps

Given a data set $\mathbf{X} = \mathbf{x}_i, i = 1, \ldots, N$ on an m-dimensional data space, a finite graph with N nodes corresponding to N data points can be constructed on X as follows. Every two nodes in the graph are connected by an edge weighted through a non-negative, symmetric, and positive definite kernel $w : \mathbf{X} \times \mathbf{X} \rightarrow (0, \infty)$. Typically, a Gaussian kernel is defined as

$$w(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \tag{1}$$

where σ is the kernel width parameter. The kernel reflects the degree of similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^m$. The resulting symmetric semi-positive definite matrix $\mathbf{W} = \{w(\mathbf{x}_i, \mathbf{x}_j)\}_{N \times N}$ is called the affinity matrix.

Let

$$d(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \mathbf{X}} w(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

be the degree of $\mathbf{x}_i$; the Markov or transition matrix $\mathbf{P}$ is then constructed by calculating each entry as

$$p(\mathbf{x}_i, \mathbf{x}_j) = \frac{w(\mathbf{x}_i, \mathbf{x}_j)}{d(\mathbf{x}_i)} \tag{3}$$

From the definition of the weight function, $p(\mathbf{x}_i, \mathbf{x}_j)$ can be interpreted as the transition probability from $\mathbf{x}_i$ to $\mathbf{x}_j$ in one time step. From the definition of the Gaussian kernel it can be seen that the transition probability will be high for similar elements. This idea can be further extended by considering $p^t(\mathbf{x}_i, \mathbf{x}_j)$ in the t^{th} power $\mathbf{P}^t$ of $\mathbf{P}$ as the probability of transition from $\mathbf{x}_i$ to $\mathbf{x}_j$ in t time steps [7]. Hence, the parameter t defines the granularity of the analysis. With the increase of the value of t, local geometric information of data is also integrated. The change in size of t makes it possible to control the generation of more specific or broader clusters.

Because of the symmetry property of the kernel function, for each $t \geq 1$, we may obtain a sequence of N eigenvalues of $\mathbf{P}, 1 = \lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_N$, with the corresponding eigenvectors $\{\mathbf{\Phi}_j, j = 1, \ldots, N\}$, satisfying,

$$\mathbf{P}^t \Phi_j = \lambda_j^t \Phi_j \tag{4}$$

Using the eigenvectors as a new set of coordinates on the data set, the mapping from the original data space to an L-dimensional ($L < m$) Euclidean space $\mathbb{R}^L$ can be defined as

$$\Psi_t : \mathbf{x}_i \rightarrow \left[\lambda_1^t \Phi_1(\mathbf{x}_i), \ldots \lambda_L^t \Phi_L(\mathbf{x}_i)\right]^{\mathrm{T}} \tag{5}$$

Correspondingly, the diffusion distance between a pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$,

$$D_t(\mathbf{x}_i, \mathbf{x}_j) = \left\| p^t(\mathbf{x}_i, \cdot) - p^t(\mathbf{x}_j, \cdot) \right\|_{1/\phi_0} \tag{6}$$

where ϕ_0 is the unique stationary distribution

$$\phi_0(\mathbf{x}) = \frac{d(\mathbf{x})}{\sum_{\mathbf{x}_i \in \mathbf{X}} d(\mathbf{x}_i)} \qquad \mathbf{x} \in \mathbb{R}^m \tag{7}$$

is approximated with the Euclidean distance in $\mathbb{R}^L$, written as

$$D_t(\mathbf{x}_i, \mathbf{x}_j) = \|\Psi_t(\mathbf{x}_i) - \Psi_t(\mathbf{x}_j)\| \tag{8}$$

It can be seen that the more paths that connect two points in the graph, the smaller the diffusion distance is.

The kernel width parameter σ represents the rate at which the similarity between two points decays. There is no good theory to guide the choice of σ. Several heuristics have been proposed, and they boil down to trading off sparseness of the kernel matrix (small σ) with adequate characterization of the true affinity of two points. One of the main reasons for using spectral clustering methods is that, with sparse kernel matrices, long range affinities are accommodated through the chaining of many local interactions as opposed to standard Euclidean distance methods - e.g. correlation - that impute global influence into each pair-wise affinity metric, making long range interactions dominate local interactions.

It is apparent from the previous introductions that the most costly part of the diffusion map is the construction of the affinity matrix, as we square the amount of data. However, this matrix is symmetric, and all the diagonal entries are equal to one. This means that only $\left(N^2 - N\right)/2$ entries of the matrix need to be calculated. Calculating this matrix could be very easily parallelized however, since any two entries are completely independent of each other. The transition matrix can be obtained from $\mathbf{W}$ by dividing every row element-wise with $d(\mathbf{x}_i)$. This could also be done in parallel. Experimental results show that the resulting matrix is sparse, and since we only need to find the first few eigenvectors and eigenvalues, this does not pose much of a problem.

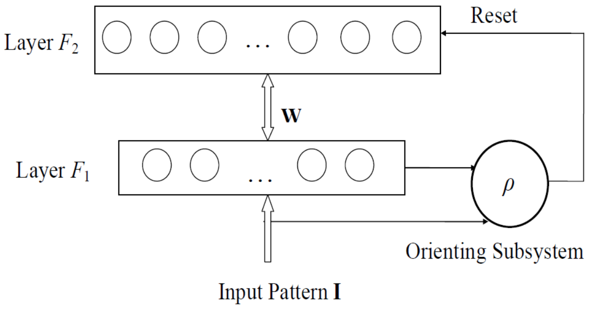


Figure 2: Topological structure of Fuzzy ART. Layers F_1 and F_2 are connected via adaptive weights $\mathbf{W}$. The orienting subsystem is controlled by the vigilance parameter ρ.

3 Fuzzy ART

Fuzzy ART (FA) incorporates fuzzy set theory into ART and extends the ART family by allowing stable recognition of clusters in response to both binary and real-valued input patterns with either fast or slow learning [5]. The basic FA architecture consists of two-layer nodes or neurons, the feature representation field F_1, and the category representation field F_2, as illustrated in Fig. 2. The neurons in layer F_1 are activated by the input pattern, while the prototypes of the formed clusters are stored in layer F_2. The neurons in layer F_2 that are already being used as representations of input patterns are said to be committed. Correspondingly, the uncommitted neuron encodes no input patterns. The two layers are connected via adaptive weights $\mathbf{w}_j$, emanating from node j in layer F_2. After an input pattern is presented, the neurons (including a certain number of committed neurons and one uncommitted neuron) in layer F_2 compete by calculating the category choice function

$$T_j = \frac{|\mathbf{x} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \tag{9}$$

where $\wedge$ is the fuzzy AND operator defined by

$$(\mathbf{x} \wedge \mathbf{y})_i = \min(x_i, y_i) \tag{10}$$

and $\alpha > 0$ is the choice parameter to break the tie when more than one prototype vector is a fuzzy subset of the input pattern, based on the winner-take-all rule,

$$T_j = \max_j (T_j) \tag{11}$$

The winning neuron J then becomes activated, and an expectation is reflected in layer F_1 and compared with the input pattern. The orienting subsystem with the pre-specified vigilance parameter ρ $(0 \leq \rho \leq 1)$ determines whether the expectation and the input pattern are closely matched. If the match meets the vigilance criterion,

$$\rho \leq \frac{|\mathbf{x} \wedge \mathbf{w}_j|}{|\mathbf{x}|} \tag{12}$$

weight adaptation occurs, where learning starts and the weights are updated using the following learning rule,

$$\mathbf{w}_j(\text{new}) = \beta(\mathbf{x} \wedge \mathbf{w}_j(\text{old})) + (1-\beta)\mathbf{w}_j(\text{old}) \tag{13}$$

where $\beta \in [0, 1]$ is the learning rate parameter. This procedure is called resonance, which suggests the name of ART. On the other hand, if the vigilance criterion is not met, a reset signal is sent back to layer F_2 to shut off the current winning neuron, which will remain disabled for the entire duration of the presentation of this input pattern, and a new competition is performed among the rest of the neurons. This new expectation is then projected into layer F_1, and this process repeats until the vigilance criterion is met. In the case that an uncommitted neuron is selected for coding, a new uncommitted neuron is created to represent a potential new cluster.

FA displays many attractive characteristics that are also inherent and general in the ART family. FA is capable of both on-line (incremental) and off-line (batch) learning. The computational cost of FA is $O(N \log N)$ or $O(N)$ for one-pass variant [17], and it can cope with large amounts of multidimensional data, maintaining efficiency. In comparison, the commonly used standard agglomerative hierarchical clustering algorithms run at least $O(N^2)$ [21]. FA dynamically generates clusters without the requirement of specifying the number of clusters in advance as in the classical k-means algorithm. Another important feature of FA is the capability of detecting atypical patters or outliers during its learning. The detection of such patterns is accomplished via the employment of a match-based criterion that decides to which degree a particular pattern matches the characteristics of an already formed category in layer F_2. Finally, FA is far simpler to implement, for example, than kernel-based clustering or clustering algorithms based on mixture densities. More discussions on the properties of FA in terms of prototype, access, reset, and the number of learning epochs required for weight stabilization can also be found in [10].

4 The Hyperspectral Core Imager

The HCI has three spectrometers with overlapping ranges. The first spectrometer has a range from 0.44μm to 0.98μm. The second and third spectrometers have ranges from 0.79μm to 1.61μm and 1.27μm to 2.63μm respectively. In total a hyperspectral image produced by the HCI contains 640 bands. The spectral resolution varies between 0.003μm and 0.006μm. Due to noise effects at the ends of the ranges of the spectrometers, these bands are discarded. Furthermore, the images produced by the three spectrometers are not completely co-registered, which makes it difficult to use the results from all three spectrometers in one experiment. In order to get a workable hyperspectral image the number of bands is usually reduced to 201. The bands used are predominantly from the short wave infrared part of the spectrum, since these bands provide the most features for identifying minerals.

Because the spectrometers are only a few centimeters above the core sample the HCI has a much better spatial resolution than most hyperspectral imaging systems. The spatial resolution on the HCI is around half a millimeter per pixel, compared to more than a meter for most aerial and satellite imaging systems. The close proximity of the sensors to the core sample also means that the predominant sources of noise are different. There are practically no atmospheric or adjacency effects (see [12]), however noise is caused by the movement of the core on the imaging tray. This causes blurring on the edges of the image. These parts of the image are usually masked. Lastly, the viewing angle of the spectrometers always stays constant.

The HCI produces about 250 000 pixels for every meter of core. The instrument scans about 5 meters of core per hour, and with core running into thousands of meters it is obvious that the amount of data produced is very difficult to handle. AngloGold Ashanti has developed a proprietary method for classifying HCI data. The method uses an endmember extraction algorithm and then clusters the resulting endmembers with a self-organizing map. The resulting clusters are then classified individually. The method generally performs well, however it does produce occasional mismatches. It also has difficulty in finding the boundary between minerals, since the clustering method only produces hard clusters. This often results in the cluster boundaries being somewhat arbitrary.

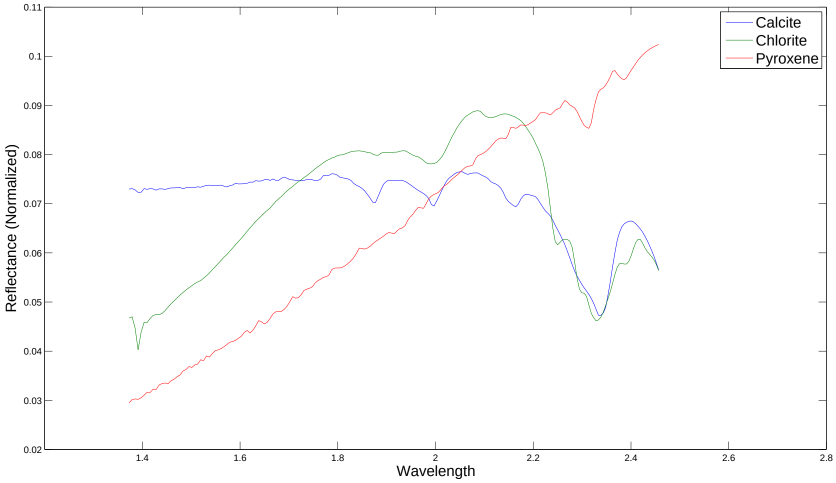


Figure 3: The spectra of the three minerals used to construct the artificial dataset. The mineral spectra were obtained from `http://speclab.cr.usgs.gov/`.

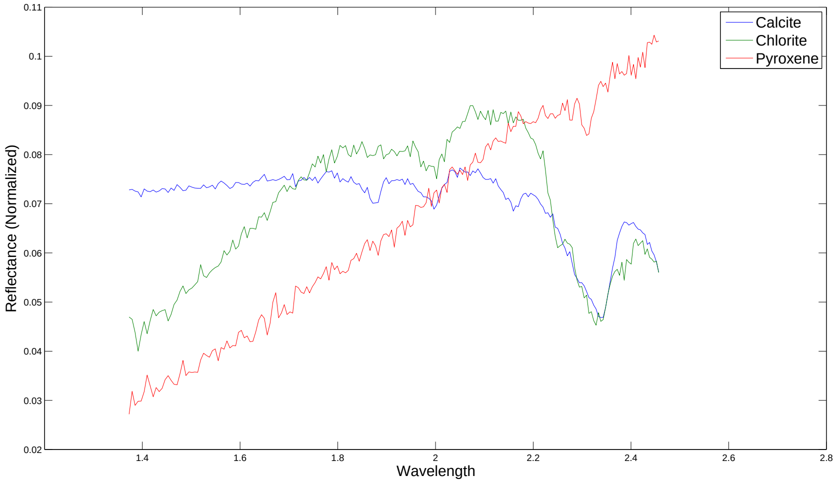


Figure 4: Example spectra of each mineral from the noisy artificial dataset.

5 Experimental Results

5.1 Artificial Data

In order to test the feasibility of the method it was first tested on an artificial dataset. The dataset was constructed using the spectra of three different minerals over the spectral range of the HCI, with the same number of bands. The spectra of the minerals used are given in Fig. 3.

The first artificial dataset contains all three minerals in equal proportions. In this case the transition matrix, $\mathbf{P}$ used for the diffusion map yielded exactly three non-zero eigenvalues, each equal to one. This proves that the diffusion map can correctly identify redundancies in non-noisy data. A diffusion map was then used to map the data to a three dimensional and a two dimensional space. Displaying the three dimensional representation of the data as an RGB image yielded the correct image, with the three different bands clearly showing. In both three and two dimensional space k-means, as well as fuzzy ART were able to cluster the data into the correct clusters. For fuzzy ART ρ was set to 0.85 and α to 0.1.

In the second artificial dataset the proportions of the minerals in the dataset were changed. The second dataset contains approximately twice as much Calcite as Pyroxene, and approximately twice as much Pyroxene as Chlorite. The results were identical to the first dataset.

Random Gaussian noise was added to the first dataset to better simulate real data. An example of three noisy spectra is given in Fig. 4. In this dataset the three largest eigenvalues of $\mathbf{P}$ were again one each, but, there were also further non-zero eigenvalues. However, the next largest eigenvalue was only 0.0037, proving once again that the diffusion map was able to identify and remove the redundancies within the data. While k-means had no difficulty clustering the dataset into three clusters, the vigilance parameter, ρ, had to be increased to 0.96 before fuzzy ART would divide the data into three clusters. In both cases the clusters were correct.

5.2 HCI Data

In this section the results of applying a diffusion map to real hyperspectral data and using k-means and fuzzy ART to cluster the lower-dimensional data are given. The process was first done on a section of roughly 5 cm of core. A high-resolution colour image of this section is given in Fig. 5. The data has 201 bands, ranging from 1.3729μm to 2.4567μm with an average spectral resolution of 0.0054μm. The wavelengths are also monotonically increasing. Each spectrum was normalized so the Euclidean mean of the spectrum is 1. The eigenvalues of the transition matrix, $\mathbf{P}$, of this section are plotted against their magnitude in Fig. 6. It can be seen that the magnitude of the eigenvalue decay exponentially to a point, after which the decay is linear.

The true dimension of the data was estimated as the point where the decay switches from exponential to linear. The assumption is made that the remaining eigenvalues and eigenvectors would only describe noise if they were included in the diffusion map. Hence, the dimensionality of this dataset was reduced to 10 using a diffusion map. The first three components of the diffusion map, as well as an RGB composite formed from them are given in Fig. 7. Even when only three components are used, clusters of minerals can be clearly seen in the image.

For comparison the principal components of the data were found. The first three principal components as well as an RGB composite are given in Fig. 8. It is interesting to note that the first principal component agrees quite closely with the first component of the diffusion map, the second principal component to the third of the diffusion map, and the third principal component to the second of the diffusion map. In order to make the RGB composites look similar, the order of the second and third principal components was reversed in the composite. Note that clusters of minerals are much clearer in the composite from the diffusion map. By comparison the clusters on the composite from the principal components appear smudged and not clearly defined. The clustering obtained by the proprietary method used by AngloGold Ashanti is given in Fig. 9. It can be seen that both composite images exhibit similar looking clusters.

Both the high dimensional image (201 dimensions) as well as the low dimensional representation (10 dimensions) were clustered using k-means and fuzzy ART. In both cases the k-means clustering

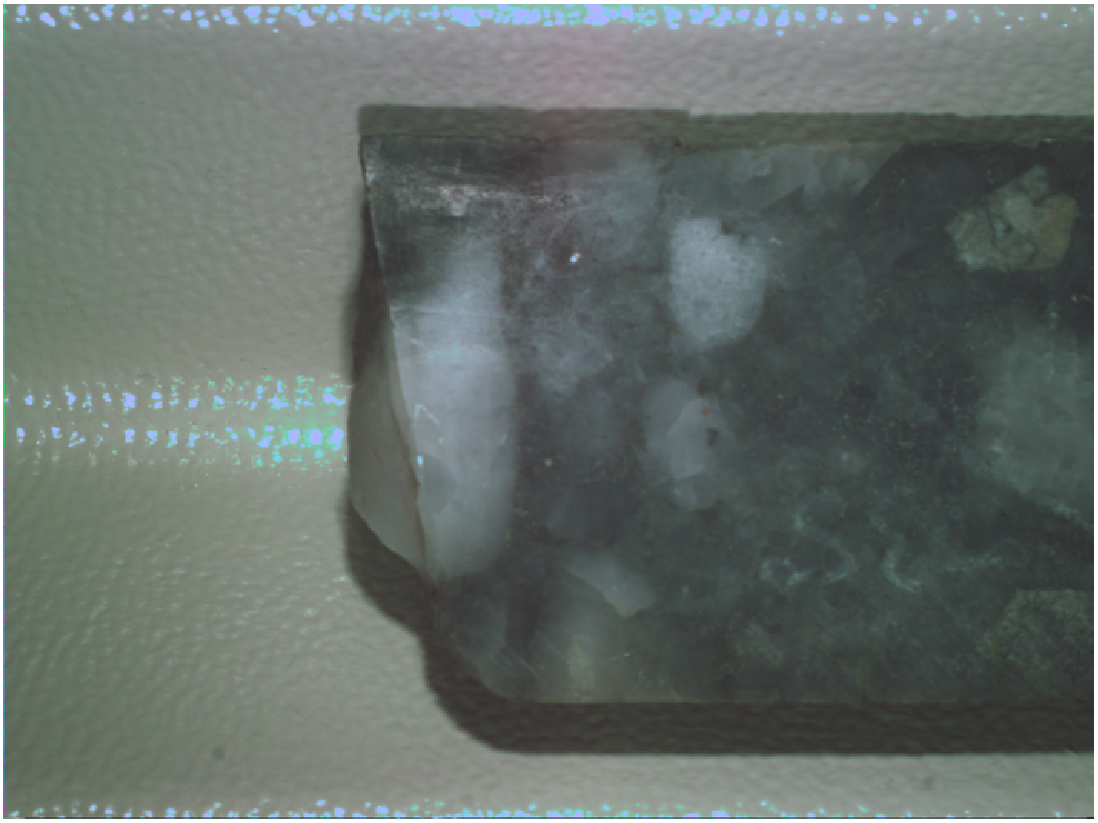


Figure 5: High resolution colour image of the data used.

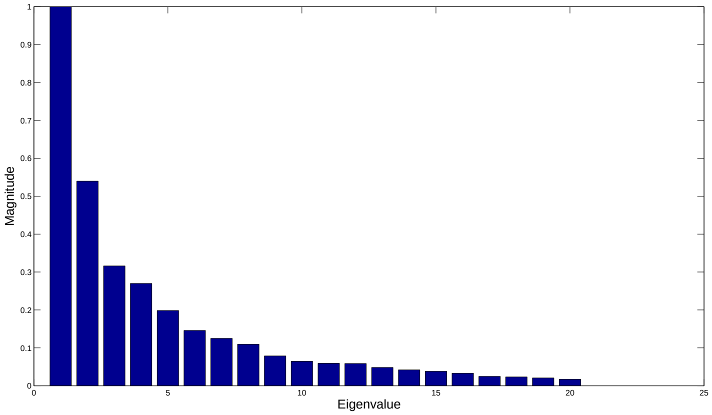


Figure 6: The magnitudes of the 20 largest eigenvalues of the transition matrix **P** of the data.

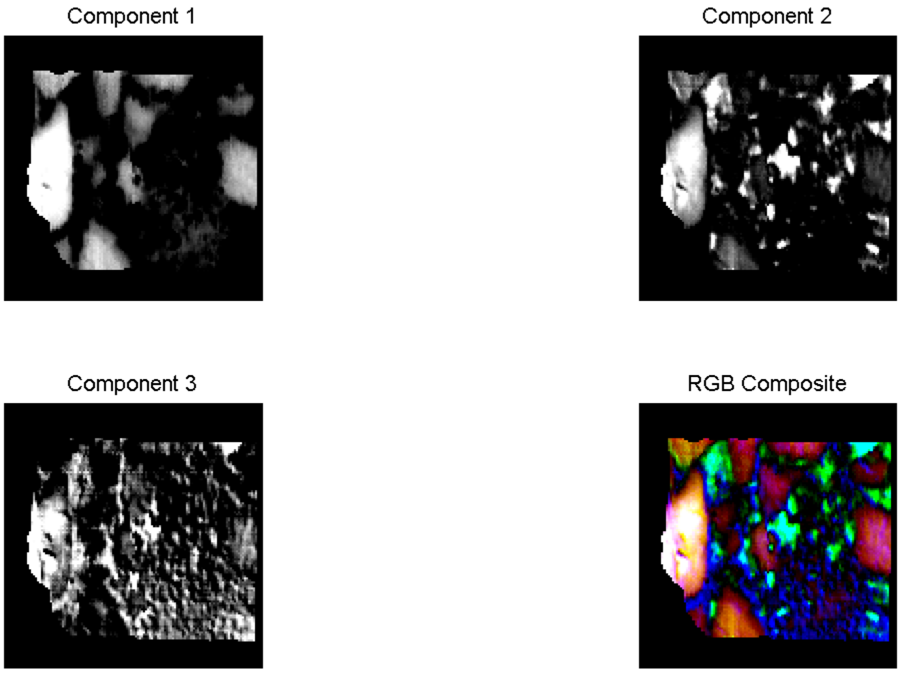


Figure 7: The first three components of the data after the dimension has been reduced by a diffusion map and the resulting RGB composite image.

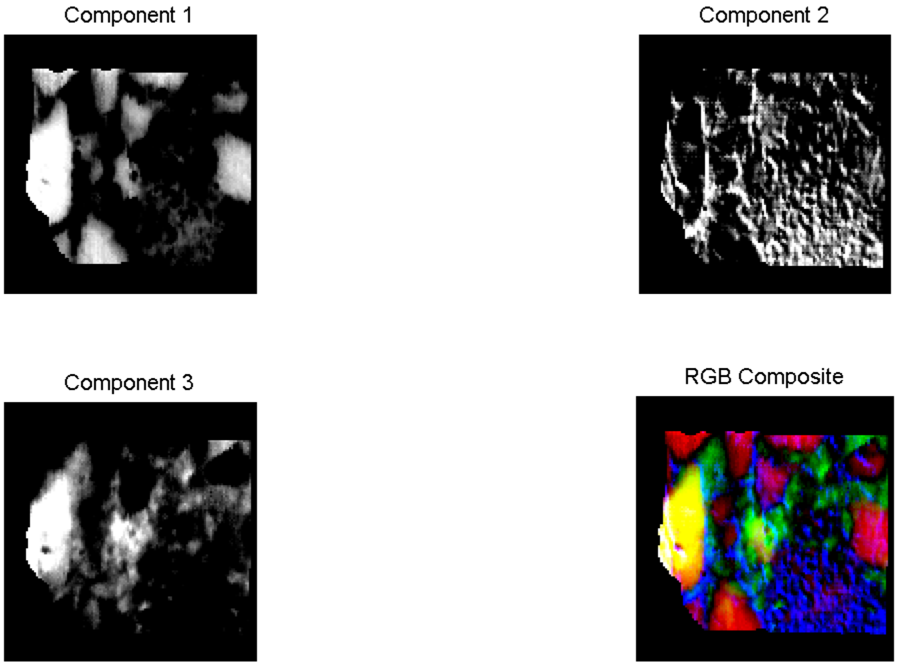


Figure 8: The first three principal components of the data and the resulting RGB composite image. In the RGB composite, component 2 was used for the blue band and component 3 for the green band to obtain an RGB composite image similar to the one in Fig. 7.

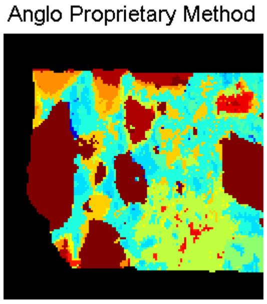


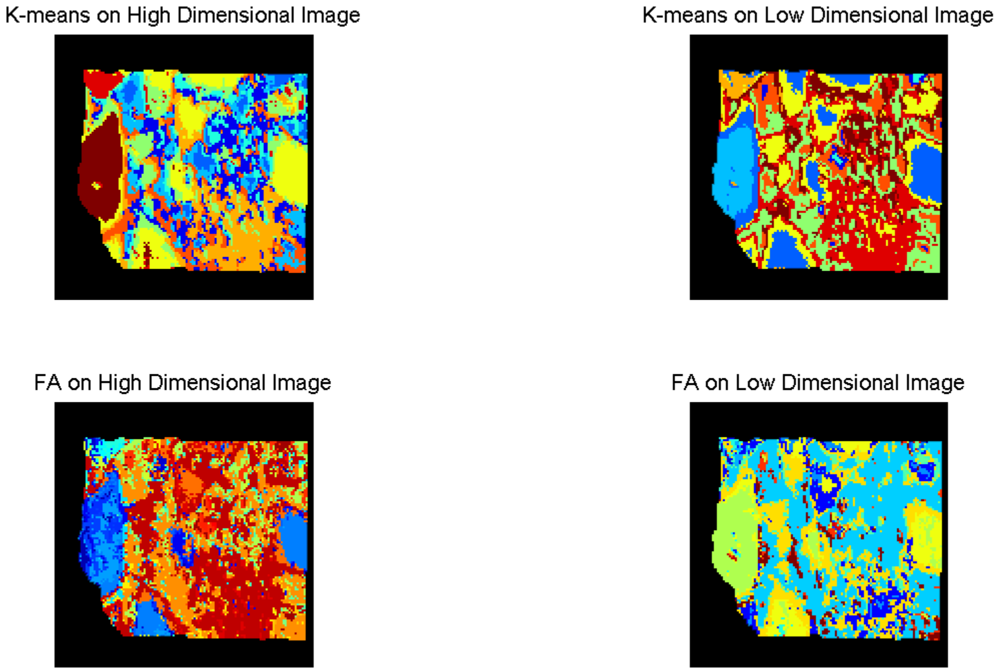
Figure 9: Clustering on the data produced by the proprietary method of AngloGold Ashanti.

was done with 10 cluster centers. On the high dimensional image FA produced 33 clusters, with ρ at 0.8 and α at 0.1. In order to get a clustering similar to the clustering produced by Anglo's proprietary method on the low dimensional representation of the data, it was found necessary to set ρ to 0.95 and α to 1. In this case FA produced 34 clusters.

Fig. 10 shows the clusterings produced and table 1 the size distribution of the clusters. The proprietary method from Anglo has 18 clusters, of which 11 have more than 100 elements. The clustering produced by k-means on the low-dimensional data looks similar to the one produced on the high dimensional data, however an analysis of the cluster sizes shows that it has more large clusters. Of particular interest with the low dimensional k-means clustering is the rim formed around some clusters (e.g. the yellow rim around the blue clusters). This is indicative of one mineral transitioning toward another mineral. Although the low dimensional clustering by FA produced more clusters than the high dimensional clustering, it only has 10 clusters of more than 100 elements, compared to the 22 of the high dimensional clustering. It is obvious that most of the smaller clusters produced by FA are outliers. In all four cases the images bear a lot of similarity to the clustering from Anglo's proprietary method (Fig. 9). Note that the colour of clusters are arbitrary, and clusters with colours close to each other are not necessarily similar. It should also be kept in mind that the clustering from Anglo has probably been cleaned up to get rid of outliers.

The effect of varying ρ when clustering the low dimensional representation of the dataset is shown in Fig. 11. While too many clusters are formed when ρ is 0.95, when it is made lower, not enough clusters are formed. The results for $\rho = 0.9$ and $\rho = 0.85$ bear no correlation to the results from the proprietary method or the components of the diffusion map.

Because of memory limitations it was impossible to perform a diffusion map on more than 5 cm of core. In order to extend the method to larger pieces of core a section of a quarter of a meter was broken up into six overlapping sections. The dimension of each section was then individually reduced to 10 dimensions using a diffusion map. The eigenvalues of the transition matrix of all six sections exhibited similar decay characteristics. The lower-dimensional representation of each section was then individually clustered using k-means and FA. A mapping between the clusters of neighbouring sections was obtained as follows. First the clusters in the overlap are compared. Suppose cluster i of section A is considered. The majority cluster of the corresponding area in section B is mapped to i. This accounts for the majority of the mapping. The remaining unmapped clusters are mapped based on the representative cluster center in the original high dimensional representation. This is compared to the representative of all the clusters in the section it is being mapped to. If the Euclidean distance between two representatives is less than a threshold, a mapping is established. Using this procedure the clusterings given in Fig. 12 were obtained. For comparison the clustering from Anglo's proprietary method is also given. Both methods produce similar clusterings to the proprietary method. However, the mapping breaks down twice for the clustering from FA. This is because the clusters obtained from FA are highly variable. In order to obtain similar looking clusters on the different sections it was necessary to vary ρ between 0.8 and 0.95. Even so, the overlapping parts are not always similar. On this section there were 20 clusters

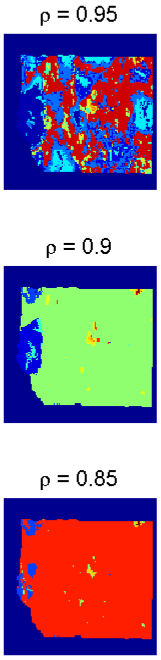

K-means on High Dimensional Image

K-means on Low Dimensional Image

FA on High Dimensional Image

FA on Low Dimensional Image

Figure 10: Clusterings of the data produced by k-means and FA. The high dimensional image has 201 dimensions and the low dimensional image 10.


$\rho = 0.95$

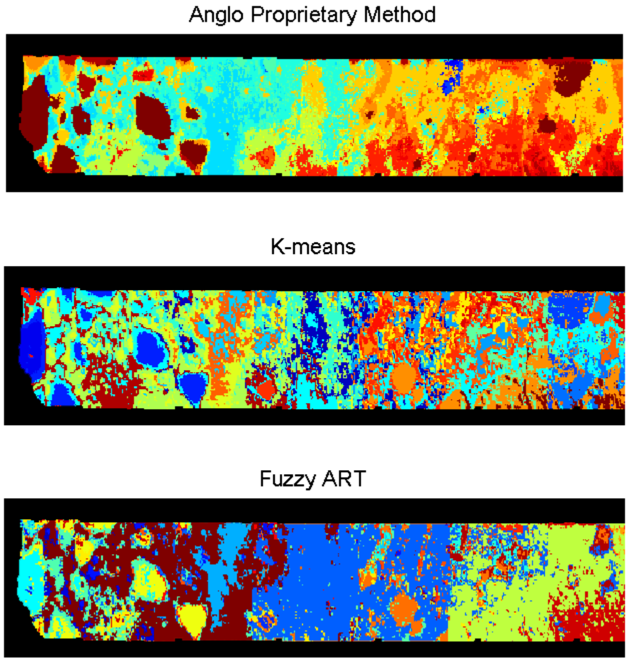
$\rho = 0.9$

$\rho = 0.85$

Figure 11: The effect of varying ρ when clustering with fuzzy ART.

Anglo Proprietary	K-means on High	K-means on Low	FA on High	FA on Low
Method	Dimensional Image	Dimensional Image	Dimensional Image	Dimensional Image
2867	1621	2349	3016	3704
2413	1510	1792	1800	2182
916	1343	1502	569	947
783	1247	1290	549	847
666	1074	1277	539	777
633	938	816	467	412
554	866	668	322	338
473	766	171	262	169
197	355	105	206	162
130	280	30	183	150
106			173	59
78			167	58
64			152	39
60			142	19
31			141	17
19			136	13
8			130	13
2			130	10
			123	9
			121	9
			119	9
			106	8
			95	7
			77	6
			61	6
			54	5
			47	5
			39	5
			30	5
			19	3
			16	2
			7	2
			2	2
				1

Table 1: The size distribution of the clusters in the five different clusterings.



Anglo Proprietary Method

K-means

Fuzzy ART

Figure 12: The matched clusterings obtained on a larger sample. The sample was divided into 6 overlapping section. The dimension of each section was then individually reduced and each section individually clustered. The clusterings were later matched.

in the proprietary clustering, 22 in the k-means clustering and 25 in the FA clustering. Although each section was clustered to 10 clusters using k-means, after matching different sections the number of clusters increased because not every cluster could be matched. Every section produced between 20 and 40 clusters when clustered individually with FA. To reduce the number of clusters, all the outliers were merged into one cluster, reducing the number clusters to between 8 and 15. Outliers were taken to be all clusters of less than 50 elements. This process helped to improve the mapping.

6 Discussion

The choice to use a diffusion map to reduce the dimensionality of the data was based on the fact that a diffusion map does not assume a linear representation of the data, as PCA does. Other nonlinear methods for dimensionality reduction include Isomap [20] and a Locally Linear Embedding (LLE) [19]. It is clear from the results that a diffusion map is a good method for reducing the dimensionality of hyperspectral data. However, the method remains computationally intensive, and it is difficult to apply it on large datasets.

As we discussed in Section 2, one way to speed up the process of the diffusion map is parallelization. However, a major bottleneck in the performance of parallel computing is communication. The speed of communication mediums is not currently comparable to that of processors [1]. In our particular case most of the overhead stems from having to distribute data to the different processors and then reassembling the results efficiently. We believe that one of the most efficient algorithms is to simply send the complete set of data points to every processor and use each processor to calculate $(N^2 - N)/2r$ weights, where r is the number of processors. A possible approach to implementing such an algorithm would be to use Google's MapReduce function, which is specifically designed to make it easy to implement concurrent programs that use large sets of data [8]. The data is mapped to many computers, and after processing, the results are reduced to a manageable scale. Speed-up can also be achieved in terms of graphics processing units (GPUs). Although GPUs are mostly familiar to us from computer games and 3-D graphics processors, they are increasingly becoming a complementary platform for general-purpose computation, as in computational intelligence, which is usually involved with highly expensive computation. The stream processing capability of GPUs, where the same instruction is performed on different data streams, makes them ideally suited to the computation of the transition matrix.

Additionally, an implementation of GPU in accelerating ART can produce a speed-up 22 times greater than the CPU implementation [16]. However, it is necessary to understand the limitations of the graphics processing hardware and to take these limitations into account in developing algorithms targeted at the GPU. Further investigation of the speed-up of the diffusion maps and FA, based on the above discussions and some possible pre-processing steps for noise reduction are important topics for further research.

The results from Fig. 12 show that the method used here to extend the process to larger samples is unstable. The method breaks down when the clusters produced in the overlapping regions between sections do not correspond. Although k-means seemed to produce similar clusters within the overlapping region, FA sometimes did not, which led to the two breaks in the composite image produced from FA clusterings. In particular, the method broke down when there was a significant difference between the variance of overlapping sections. This caused FA to produce different clusterings in the overlapping regions. Hence, a case where the method would break down is where a section with many different minerals borders a section with relatively few minerals.

Future work will have to focus on improving this aspect of the process. Most importantly, the clustering should be performed on the complete dataset in one step. There are three avenues of research we plan to pursue in realizing this goal. The first and most straightforward improvement would be to map larger sections of the core to the lower dimensional space. However, even with parallelization it would still be infeasible to obtain the diffusion map of a large dataset. A meter of core contains 250,000 pixels. This means that the transition matrix for a meter of core would take up more than 400 GB of memory. Although it is possible to calculate the eigenvectors and eigenvalues of the transition matrix without ever explicitly calculating this matrix, it is computationally much more intensive to do this. Parallelization may make this a realizable goal.

The second of our proposed improvements is a different take on the idea of splitting the core into manageable sections. Instead of clustering each section separately and then finding a mapping between the clustering, we propose to first find a mapping between the lower dimensional representations of the different sections and then to cluster the whole dataset. The third improvement would be to to use an out-of-sample extension similar to the ones mentioned in [2]. This would entail only performing the diffusion map on a subset of the data, either selected randomly, or through some coarse clustering method. The lower dimensional representation of the subset of the data is then probabilistically extended to the rest of the dataset. In this way an approximation of the whole dataset in the lower dimension is found. We are positive that each of the above proposed areas of research is realizable.

7 Conclusions

The occurrence of large amounts of hyperspectral data brings important challenges to storage and processing. Here, we investigate the performance of reducing the dimensionality with diffusion maps and clustering the lower dimensional data on real hyperspectral image data, from core samples provided by AngloGold Ashanti. We compare clusterings obtained from k-means and fuzzy ART on the lower dimensional representation of the data to the clustering produced by Anglo's proprietary method.

The experimental results are encouraging, with the clusters being similar to the ones produced by the proprietary method. Although the clusterings of the lower dimensional representation correlate well to the clusterings produced on the higher dimensional data, the lower dimensional data leads to more large clusters, which is more desirable. Currently the method can only be used on small sections of hyperspectral data. Future work will aim to extend this method so it can be applied to large datasets.

Acknowledgments

Dr. Xu and Dr. Wunsch gratefully acknowledge support from the National Science Foundation, the Missouri University of Science & Technology Intelligent Systems Center, and the M.K. Finley Missouri Endowment. Dr. Damelin acknowledges support from the School of Computational and Applied Mathematics at Wits, the Department of Mathematical Sciences at GSU, the Unit for Advances in Mathematics and its Applications at GSU and the National Science Foundation. We are grateful to Anglo Technical Division Geosciences Resource Group and AngloGold Ashanti for allowing us access to the HCI data.

References

[1] Gordon Bell. Bell's Law For the Birth and Death of Computer Classes: A theory of the computer's evolution. *Communications of the ACM*, 51(1):86–94, January 2008.

[2] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. *Advances in Neural Information Processing Systems*, 16:177–184, 2004.

[3] Jeffrey H. Bowles and David B. Gillis. An optical real-time adaptive spectral identification systems (ORASIS). In Chein-I Chang, editor, *Hyperspectral Data Exploitation: Theory and Applications*, pages 77–106. Wiley, 2007.

[4] G. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[5] G. Carpenter, S. Grossberg, and D. Rosen. FUZZY ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance. *Neural Networks*, 4:759–771, 1991.

[6] K. Cawse, S. Damelin, L. du Plessis, R. McIntyre, M. Mitchley, and M. Sears. An investigation of data compression techniques for hyperspectral core imager data. In *Proceedings of the Mathematics in Industry Study Group, South Africa – MISG2008*. To appear.

[7] Ronal R. Coifman and Stephane Lafon. Diffusion maps. *Journal of Applied and Computational Harmonic Analysis*, pages 5–30, April 2006.

[8] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):107–113, January 2008.

[9] S. Grossberg. Adaptive pattern recognition and universal encoding II: feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, 23:187–202, 1976.

[10] J. Huang, M. Georgiopoulos, and G. Heileman. Fuzzy ART properties. *Neural Networks*, 2:203–213, 1995.

[11] Yosi Keller, Stephane Lafon, and Michael Krauthammer. Protein cluster analysis via directed diffusion. *Bioinformatics preprint*, 2005.

[12] John P. Kerekes and John R. Schott. Hyperspectral imaging systems. In Chein-I Chang, editor, *Hyperspectral Data Exploitation: Theory and Applications*, pages 19–45. Wiley, 2007.

[13] S. Lafon, Y. Keller, and R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

[14] Stephane Lafon and Ann B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, September 2006.

[15] Y. Ma, S. Damelin, O. Masoud, and N. Papanikolopoulos. Activity recognition via classification constrained diffusion maps. In *International Symposium of Computer Vision*, pages 1–8, 2006.

[16] R. Meuth. GPUs surpass computers at repetitive calculations. *IEEE Potentials*, pages 12–15, 2007.

[17] S. Mulder and D. Wunsch II. Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks*, 16:827–832, 2003.

[18] B.D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[19] S. Roweis and L. Saul. Nonlinear dimension reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[20] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[21] R. Xu and D. Wunsch II. *Clustering.* IEEE/Wiley, 2008.

[22] Rui Xu, Steven Damelin, and Donald C. Wunsch II. Applications of diffusion maps in gene expression data-based cancer diagnosis analysis. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4613–4616, August 2007.

[23] Rui Xu, Louis du Plessis, Steven Damelin, Michael Sears, and Donald C. Wunsch II. Analysis of Hyperspectral Data with Diffusion Maps and Fuzzy ART. In *Proceedings of the IJCNN 2008*, 2009. To appear.