Selecting Library Endmembers for Spectral Unmixing Using k-means Clustering

BLAKE GARWOOD

I. INTRODUCTION

Spectral unmixing allows us to find out what type of materials are in a pixel by comparing its spectral signature to known pixels' signatures. This is mainly useful for hyperspectral images because of the vast number of spectral bands we can compare.

Spectral unmixing uses a library of known spectral signals L, which it compares against a selected unknown signal. The algorithm returns an "abundance vector," which consists of the predicted amount of each library element in the unknown pixel.

In this paper, we will try to see how to find the best signals to represent each element in our signal library. Library endmembers are often selected by hand. We will explore the use of k-means clustering to select our signals.

II. HAND-SELECTION OF ENDMEMBERS

Hand-selection of endmembers is problematic for many reasons. It is hard to find a pure pixel that contains only one material. Because of the large spatial area of pixels from satellite images, a given pixel may contain materials/objects that are unknown to the user. Therefore, the pixel may not be representative of the class that the user is gathering data for.

For example, Figure 1 shows the spectral signatures of four different selections from the same parking lot in an image. Pixels 1, 2, and 4 seem to have very similar spectral signals, but the third pixel is noticeably different. This could be caused by something like a puddle, dirt patch, or shopping cart in the pixel that we cannot see.

To attempt to mitigate this issue, we will begin by taking the average vector of multiple selections. This will help to filter out the noise from unseen objects in our selection, and help to produce a spectral signature that is more representative of its intended class of material/object.

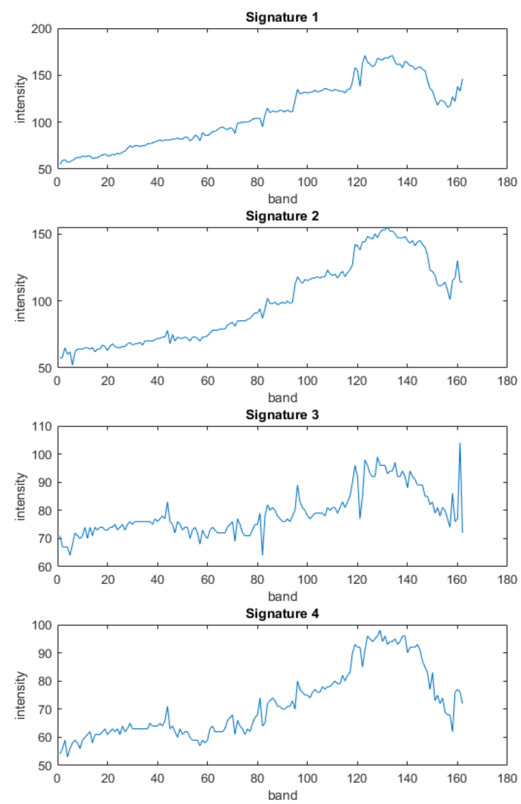

Figure 1: Spectral signatures of four different pixels from the same parking lot in the same image.

II. AVERAGING MULTIPLE SELECTIONS

To find the average vector of multiple spectral signatures, we will simply create a new vector using the pointwise average of all selections at each wavelength.

Implementing this averaging process in our library endmember selection process decreases the effects of picking a bad pixel (shown in Figure 2). Figure 3 shows the results of spectral unmixing using a single selection library and an averaged selection library.

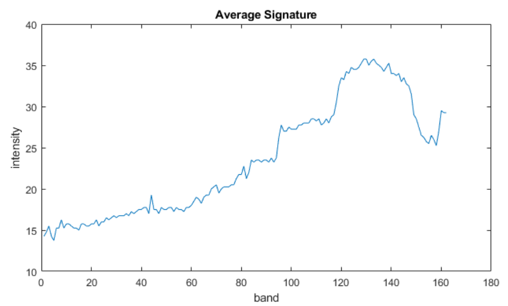

Figure 2: Average vector of spectral signals from Figure 1.

		Abundance Values			
	Library	Road	Trees	Grass	Roof
Road	Single	0.983	0	0.008	0
	Average	2.585	0.021	0	0.084
Tree	Single	0	0.914	0	0
	Average	0	7.706	0	0
Grass	Single	0.074	0.040	0.831	0.009
	Average	0	0.866	3.158	0
Roof	Single	0.021	0	0	0.981
	Average	0	0	0	4.183

Figure 3: Spectral unmixing results with single library selections vs. averaged library selections.

Figure 3 shows the results of unmixing pixels of each class (road, tree, etc.) using the “lsqnonneg” command in Matlab. The results with two different libraries are shown: A library that is a single pixel selection and a library that contains an average of multiple selections.

Furthermore, this process allows the user to select endmembers they deem to be of the same class that are slightly different. This allows the user to create more generalized spectral signals for their desired classes of materials.

This was done in the averaged library from Figure 3 for the “road” classification. Pixels were selected from the parking lot, the highway, as well as neighborhood roads. These types of pavement are often slightly different, and averaging their spectral signals allows us to classify them all as the same category.

Figure 4: Band #123 of the hyperspectral image used in tests. This image has 162 total bands after removal of the noisy water absorption bands.

III. USING K-MEANS CLUSTERING

We will now attempt to implement an unsupervised version of this selection process by using k-means clustering to find classes of endmembers. If k-means clustering will be able to select similar classes for us, we can further generalize the signals of our endmembers.

The signals we put in our library will simply be the average of all the pixels in each cluster. The user will have to determine the number of clusters they want, and then look at the clusters to figure out which material they represent.

One trick that helped the k-means algorithm select more relevant points was increasing the number of clusters to at least one more than the desired library size. This reduced the amount of outliers the clustering algorithm attempts to include in each of the classes.

IV. RESULTS

To begin, the spectral signatures that our k-means clustering algorithm averaged out to look very similar to the averaged selection endmembers. This is a good sign that we will have a successful unmixing. Three libraries are shown in Figure 5. The multiple hand-selection averaged library, the library that was produced using our k-means clustering algorithm, and also a library that averages the pixels in the ground-truth image similarly to our clustering algorithm.

To determine whether k-means clustering was successful in finding endmembers that accurately represented the true classes, we calculated the percent error for each band of each of the four library endmembers, and averaged them.

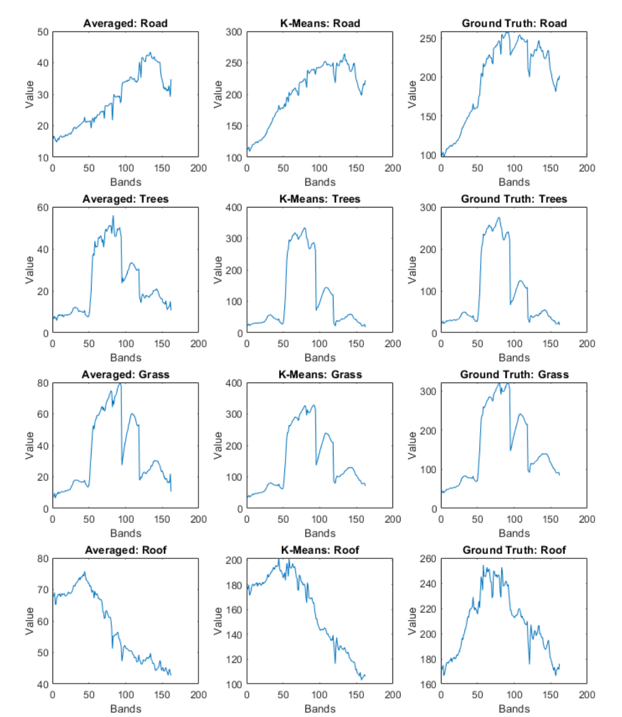

Figure 5: Side-by-side: multiple selection averaged library, k-means clustering library, and averaged ground truth library.

Overall, we had a percent error of 23%. This is not terrible but it isn't great. Upon closer inspection, we noticed that the algorithm is not performing well for the road and roof. Removing these classes from our libraries, we get a percent error of about 6.8%. This is very good.

V. CONCLUSION

There are some benefits and drawbacks of using k-means clustering over averaging multiple user selections. These effects come from the different perspectives of the computer and the user.

This algorithm can be performed with little help from a user. All the user must do is determine which endmember each cluster represents. The user can then make sense of the abundance vector that is produced in unmixing.

One drawback is the performance. As we can see from our results section, some of the endmembers in our k-means clustering library were vastly different from the averaged hand-selection library. This is more noticeable with some materials than others, but is still a major issue with the reliability of this algorithm.

Also, the classes are not based on the user's idea of material, but rather the algorithm's evaluation of which pixels are most similar. This works well for the most part. However, the signal the algorithm generates may be biased away from the true mean of the material that the user wants. This is because a cluster may include relatively similar materials that are not the same. Adjusting the number of clusters and selecting the most representative clusters for the class of material may help, but that requires more supervision from a user.