

Ben-Gurion University of the Negev

Faculty of Engineering Science

School of Electrical and Computer Engineering

Dept. of Electrical and Computer Engineering

361.2.6291 - Selected Topics in Image Processing

Course Project

**Anomaly Detection in Hyperspectral Images   
Using Subspace Clustering**

**Students** (name & I.D):

Shaked Cohen, I.D 304920929

Ofir Refael Cohen, I.D 203118591

**Supervisors**: Prof. Rotman Stanley

**Submitting date**: 02/06/2020

**Contents**

[Introduction 3](#_Toc41738085)

[Background 4](#_Toc41738086)

[Clustering methods 4](#_Toc41738087)

[K-Means Clustering 4](#_Toc41738088)

[K-Subspaces Clustering 7](#_Toc41738089)

[Anomaly detection 10](#_Toc41738090)

[RX Algorithm 10](#_Toc41738091)

[Methodology 11](#_Toc41738092)

[Algorithm implementation and initialization 11](#_Toc41738093)

[RX scoring methods 13](#_Toc41738094)

[Data Generation 14](#_Toc41738095)

[Segmented images 15](#_Toc41738096)

[RX scores 16](#_Toc41738097)

[Data Exploration 16](#_Toc41738098)

[Findings 17](#_Toc41738099)

[Conclusions From Findings 30](#_Toc41738100)

[Further work to be done 31](#_Toc41738101)

[Conclusions 32](#_Toc41738102)

[Bibliography 33](#_Toc41738103)

# Introduction

Clustering is a method of unsupervised learning and is a common technique for statistical data analysis usedin many fields such as:   
Pattern Recognition, Computer Vision, Bioinformatics, Data Compression,  
ML and many others.   
Clustering is also a useful tool for **image processing** – classifying each pixel of the image into a specific group (cluster) creating a form of image segmentation with unknown properties.

There are many clustering algorithms in data analysis.   
One of the most popular is the **K-Means** algorithm we will discuss later.   
It is well-known in its simplicity, fast convergence and many other advantages.   
However, K-Means is not very effective with high-dimensional data.   
That’s why we also experimenting with the **K-Subspaces** algorithm.  
  
The K-Subspaces is a generalization of K-Means algorithm to higher dimensions –   
we consider it might be useful for processing high-dimensional data such as **Hyperspectral image**.

In our project we are using both algorithms as a pre-processing tool for anomaly detection. We apply the **RX algorithm** (a commonly used anomaly detection algorithm that we will discuss later) in various **NEW** approaches on the segmented data which we get from the **K-Means and K-Subspaces** algorithms and by comparing the results we try to find some insights.

# Background

## Clustering methods

### K-Means Clustering

The K-Means clustering method based on the notion that the data points are distributed around a collection of cluster centers called centroids. By knowing the value of , the K-Means is an iterative algorithm that estimates the clusters centers, using 2 stages:

1. Minimizing the squared distances from the data points to the centers   
   (L2 norm) - thus assigning point to each center and creating the cluster :
2. Calculating the mean of each cluster as the new centers:

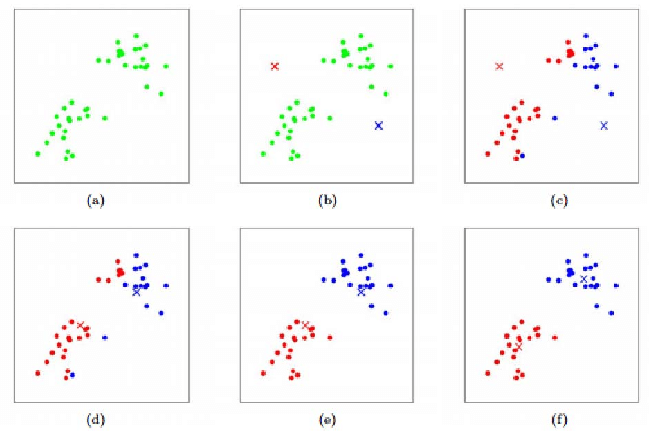


Figure 1: Visualization of K-Means steps

The algorithm is widely common especially because of its simple implementation and fast convergence.

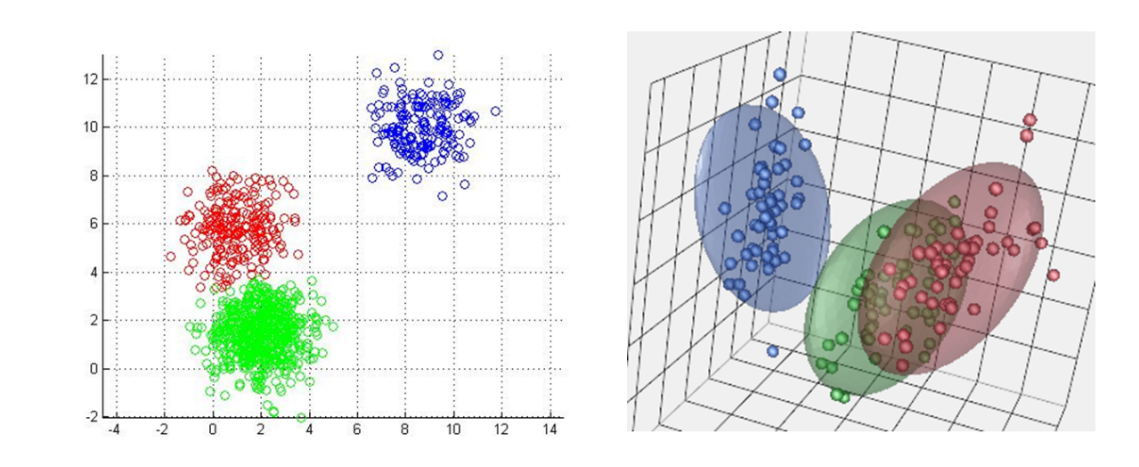
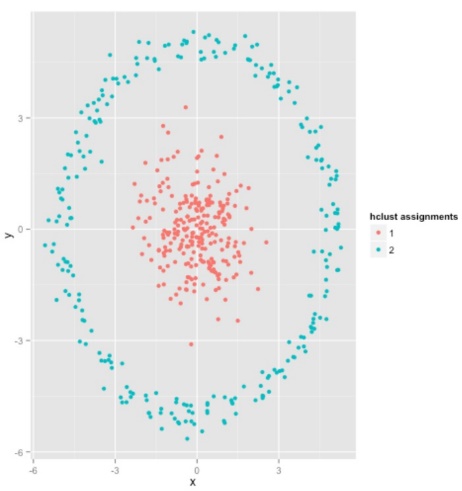
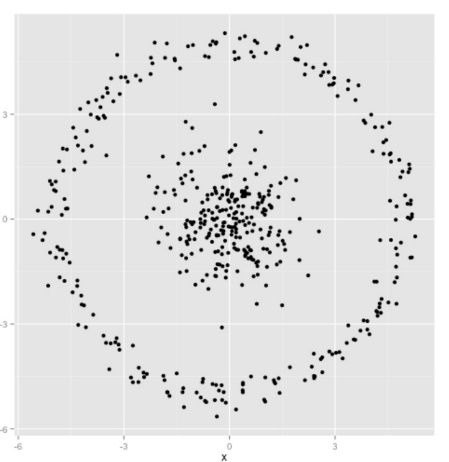
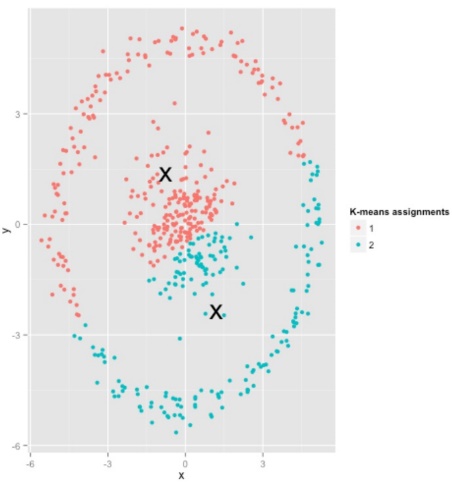
However, one of its disadvantages is that the minimizing distance method is based  
on the Eculidian distance (L2 norm).   
As a result, it creates **only** spherical (ball-shaped) data clusters.

Figure 2: Visualization of K-Means ball-shaped clusters

Clusters can be modeled in different shapes, forms and sizes.  
This flexiblity is necessary especially when the objective is involving visual aspects such as **hyperspectral images**.  
   
Therefore, using K-Means limits us to ball-shaped clusters and prone to errors.

Here are some examples:

**

Original Data K-Means Clusters Expected Clusters

Figure 3: K-Means Failure example 1

As we can see from the expected clusters, both of the clusters have the same center. The inner circle cluster overlap with outer circle cluster.  
The K-Means found 2 different centers with a clear seperation between each cluster. Therefore, we can conclude that K-Means fails when there is an overlap between clusters.

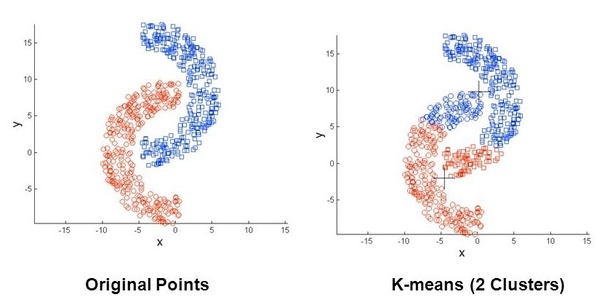


Figure 4: K-Means Failure example 2

We can see that K-Means has difficulties dealing with non-convex/non-round shape clusters.

### K-Subspaces Clustering

The K-Subspaces clustering is a generalization of the K-Means algorithm.

This algorithm extends the spherical data clustering from K-Means to many other kinds of clustering such as line-shaped clusters, plane-shaped clusters etc.  
The extended clusters are classified to subspaces while the algorithm steps are overall quite similar to the K-Means steps, in order to maintain it simple to implement and with fast convergence.

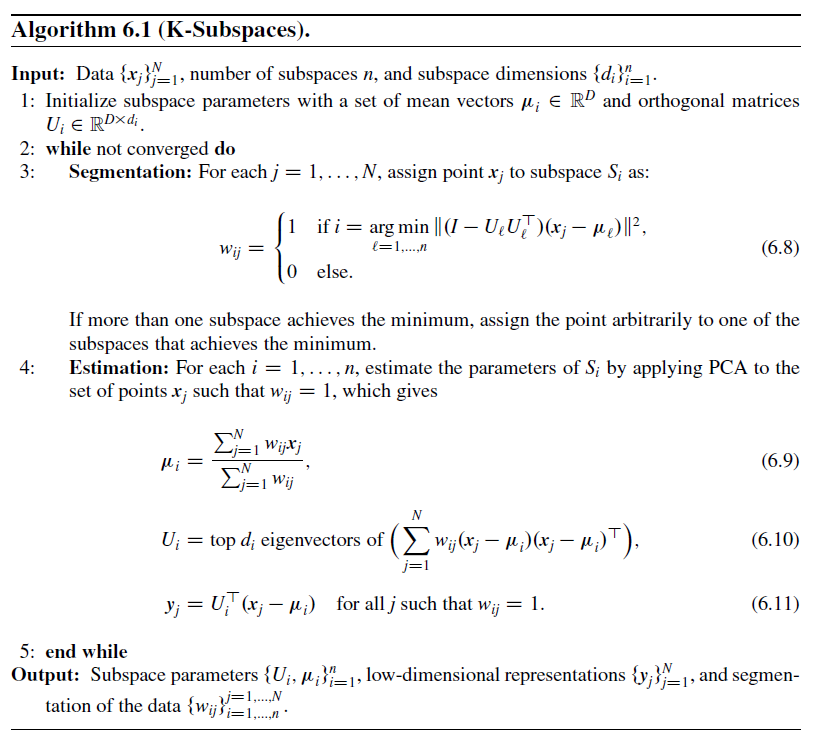
The data points in this case are distributed to multiple affine subspaces of any dimensions instead of multiple cluster centers.  
By knowing the number of subspaces and their dimensions , The data points assigned to their closest subspace of defined as - where is an orthonormal basis for the subspace, is a low-dimensional representation of point in sub space and is the affine parameter.

The K-Subspaces algorithm is 2 stage algorithm that estimates the points , the subspace bases , and the segmentation of the data :

1. **Segmentation:** is done by assigning the data point to the closest subspace using the L2 norm.
2. **Estimation:**  calculating the new subspace parameters using PCA

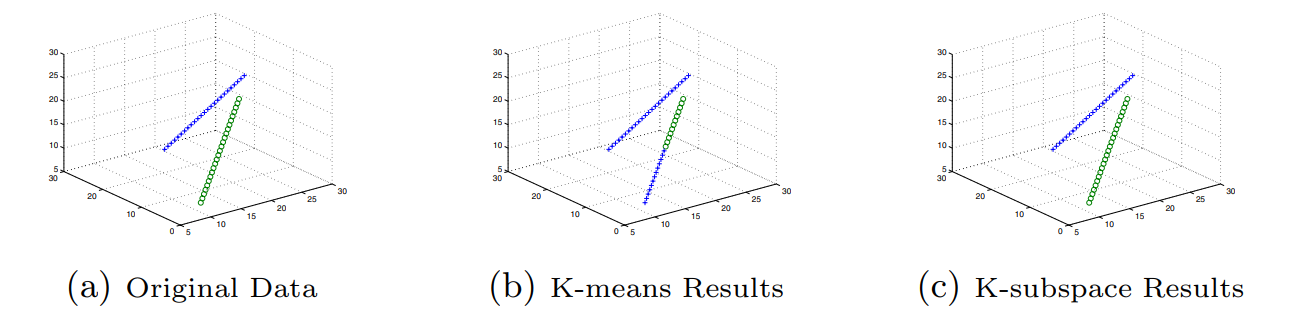
We notice that for we get the K-Means algorithm.

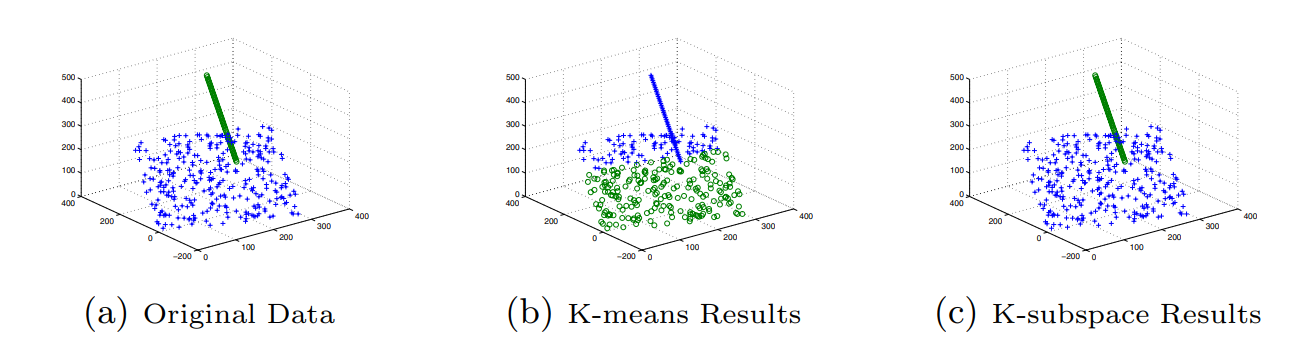
The algorithm:

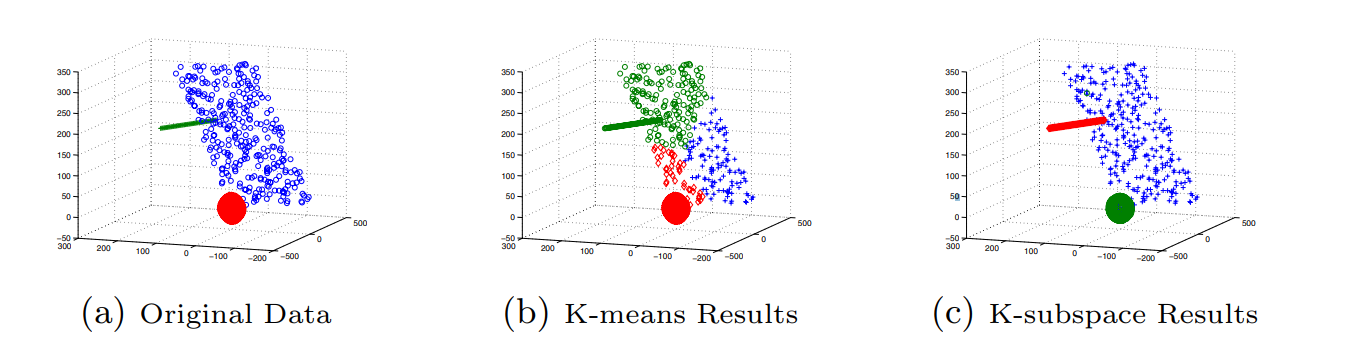


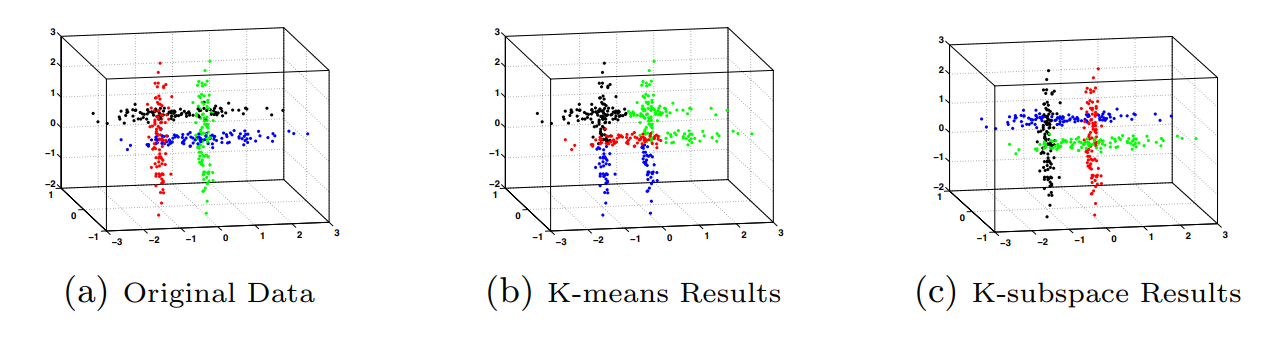
*Figure 5: K-Subspaces algorithm*

Now with the K-Subspaces algorithm, we can overcome some of the difficulties of the K-Means we mentioned before and achieve the flexibility for extended clustering   
as we needed for Hyperspectral Imaging.

Here are some examples:

*Figure 6: Two lines in 3D space*

*Figure 7: Line-Plane in 3D space*

*Figure 8: Line-Plane-Ball in 3D space*

*Figure 9: Four clusters in 3D space*

## Anomaly detection

### RX Algorithm

The RX (Reed-Xiaoli) algorithm is based on the Mahalanobis distance and is given by:

is a given pixel, is the estimation of , is the covariance matrix of X and is a threshold for detection.

The term is the estimation error of the pixel, if that error is big it means the pixel was poorly estimated, hence it implies an anomaly.

This algorithm assumes no prior knowledge of a target. Therefore, it is very useful when the target is unknown.

The inverse covariance matrix is used to produce weights and help with the noise in the image.

Basically, The RX algorithm scores each pixel from the image. Higher scores are given to more anomalous pixels.

# Methodology

The dataset used for this project is the self-test image from the RIT target detection dataset[[1]](#footnote-1). It includes a hyperspectral image of Cooke City Montana. The image is of size 800 x 200 x 126 (126 spectral bands, each of size 800x200).



Figure 10: Cooke City, MT

## Algorithm implementation and initialization

The first step was to implement the K-Subspaces algorithm in MATLAB and find a good initialization for the K-Subspaces algorithm.

Initialization of unsupervised ML algorithms is a common problem, and since our algorithm works in high dimensions a bad Initialization can be a big problem:

1. It may take too long to converge if at all – this also depends on the way you update subspaces to avoid convergence to empty subspaces. [[2]](#footnote-2)
2. The algorithm converges into a local minimum of the objective function thus there as a variety of final outpost from the algorithm.

We found out that using the K-Means algorithm in order to get initial parameters for the K-Subspaces algorithm gets us good segmentation results and is a good solution for the initialization problem.

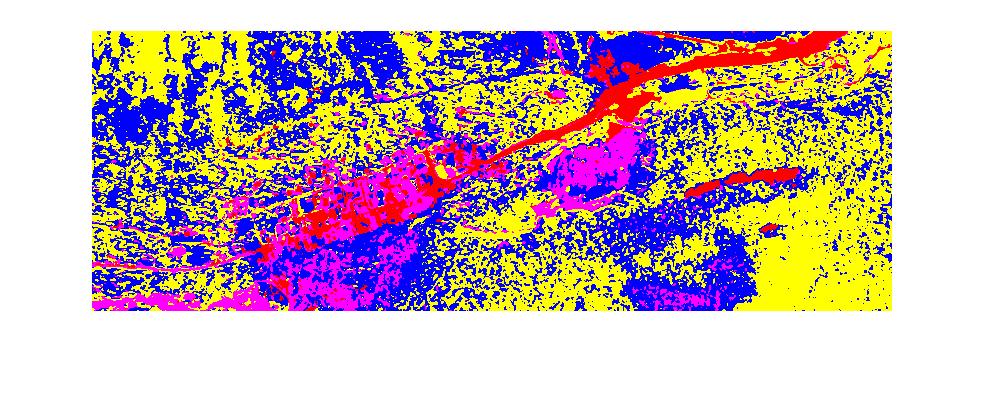


Figure 11: K-Subspaces segmentation with random initialization – K= 6, SubSpaceDim = 9

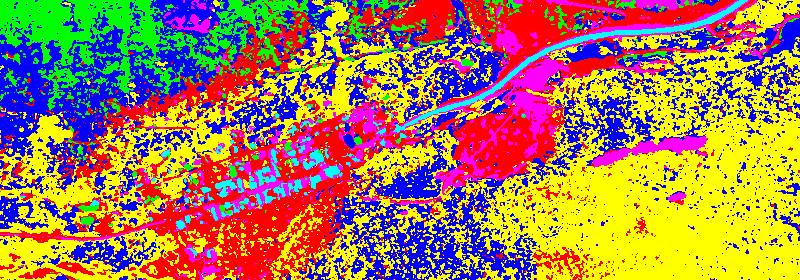


Figure 12: K-Subspaces segmentation with k means initialization – K= 6, SubSpaceDim = 9

As we can see the initialization has a big impact over the final segmentation, with random initialization we got this time only 4 segments instead of 6!!

This is because as we discussed there are local minimums with empty clusters.

**Form this point onwards we run the K-Subspaces algorithm with K-Means as initialization.**

## RX scoring methods

After having segmentation results of both algorithms, we realized that the normal approach of the RX algorithm no longer applies. This is since the normal approach assumes the mean of the pixel is approximated using the pixel neighbors, but the cluster “pixel neighbors” are unknown or does not exist.

Therefore, we created the following **NEW** approaches for applying the RX algorithm on the segmented data:

1. **The direct cluster approach**: applying RX algorithm on each cluster independently using the cluster mean and covariance.
2. **The indirect cluster approach:** applying RX algorithm on each cluster independently using the original mean (neighbors mean) of the image and the local clusters covariance.

We also realized that normalization of the RX scores in different “times” is significant. After we have the scores of the whole image VS after we have the scores for each cluster.

Therefore, we created the following approaches for normalizing the RX algorithm scores:

1. **Cluster Normalization:** normalizing the RX scores according to each cluster separately, this allows for a bigger dynamic range.
2. **Whole data Normalization**: normalizing the RX scores according to the whole of the data, this allows for getting stronger anomalies only.

## Data Generation

For the data comparison we use the K-Means algorithm.

### Segmented images

The first step was to run the K-Means and then the K-Subspaces for different K and subspace dimensions, and create segmented images for visual comparison and evaluation:



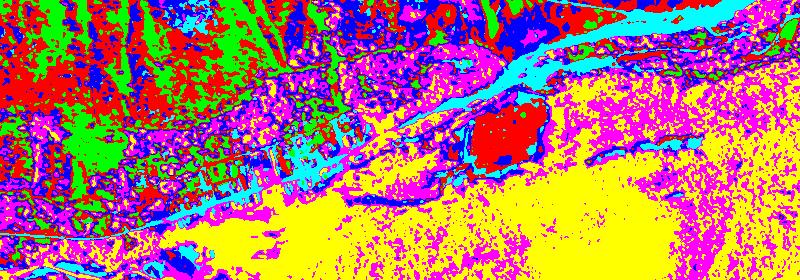
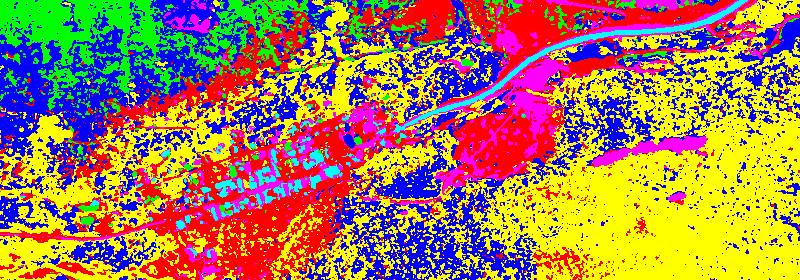
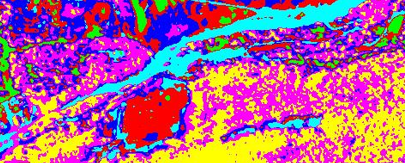
Figure 13: original image in RGB

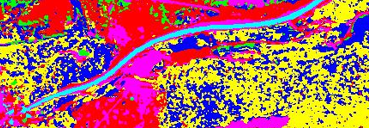
Figure 14: K-Means segmentation – K= 6, SubSpaceDim = 9

Figure 15: K-Subspaces segmentation – K= 6, SubSpaceDim = 9

From the results, we can clearly notice that the K-Subspaces yields **smoother** segmentation results than K-Means.

Here are some examples:

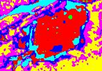
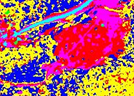
* The road curves are sharper and less noisy in the K-Subspaces segmentation.



*K-Means segmentation K-Subspaces segmentation Original Image*

Figure 16: Segmentation comparison

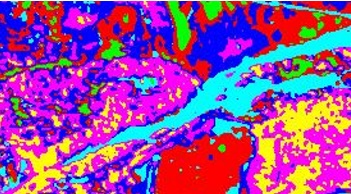
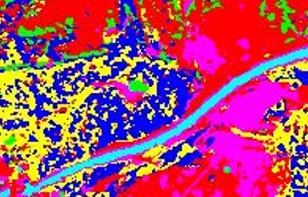
* The K-Subspaces “recognized” the line and spots as seen from the images below (marked as pink) while K-Means did not. In addition, the white spot is smoother and less noisy in the K-Subspaces segmentation (marked as green):



K-Means segmentation K-Subspaces segmentation Original Image

Figure 17: Segmentation comparison 2

* The K-Subspaces “recognized” the white building as seen from the images below (marked as green) while K-Means did not:



K-Means segmentation K-Subspaces segmentation Original Image / Google Earth

Figure 18: Segmentation comparison 3

### RX scores

The second step was to run the RX algorithm in order to generate data for comparison and evaluation.

The following algorithms were run and saved for both K-Means and K-Subspaces:

1. The directcluster approach with Cluster Normalization
2. The directcluster approach with Whole data Normalization
3. The indirect cluster approach with Cluster Normalization
4. The indirect cluster approach with Whole data Normalization

## Data Exploration

In order to compare between the 2 algorithms, different graphs and visualizations were made.

Visualizations used:

* Scatter plots
* Spatial distributions
* Segmented RGB images (as we have already shown figures 14,15)

# Findings

For our findings we will show the results we have found using K=6 and subspace dimension=9. We have run the algorithm and plotted the results for k=2:15 and subspace dimension=2:15.

The reason for this is to allow us to properly analyze these results.   
We are working on a proper metric that will allow us to compare results from different K and subspaces dimensions with meaningful insight.

For each of the following approaches we will first look at the scatter plot of the RX scores of the K-Means Vs the K-Subspaces.   
Using this information, we will see the spatial distribution of the meaningful anomalies on the image.

The segmented RGB images of each algorithm is the same in all the different approaches (we will refer you back to figure 14, and 15).

## The directcluster approach with Cluster Normalization

Remembering the RX algorithm:

In this approach since in the cluster there are no “Natural” neighbors for each pixel, “m” represents the mean of the pixels in the same cluster. Moreover, we use the inverse covariance matrix generated by the cluster. Thus, for each cluster separately we apply the RX algorithm in order to get “cluster scores”:

Where X is the hyperspectral image, S is the cluster (associated with the subspace), is the cluster mean, and is the inverse covariance matrix of the cluster.

After getting the different scores from each cluster we normalize the scores according to each cluster:

Let’s represent the cluster scores as then the normalized scores are: and the scores of the whole image is:

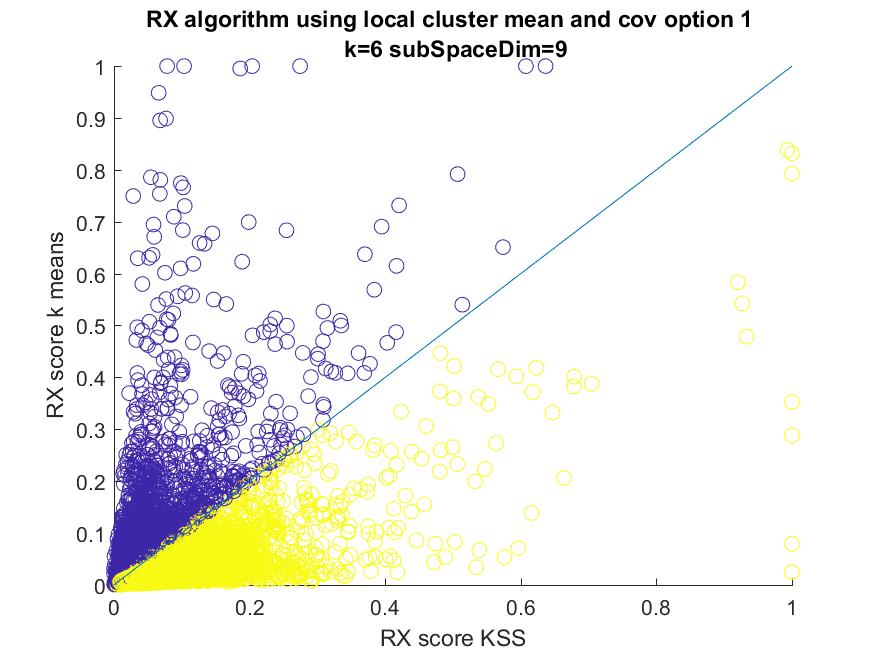
Using this we get the following RX scores, of the K-Means Vs the K-Subspaces:

Figure 19: Direct cluster approach with cluster normalization RX scores

As we can clearly see there is a vast difference in the anomalies detected using the different segmentations, this means that some anomalies can only be found in one algorithm and not the other.

Moreover, in this method we get a high dynamic range of the scores  
(there are no big gaps between high scored anomalies and low scored ones).

By plotting, only the distinct anomalies that we find (using the scatter plot to determine ) on the image we get the following:

In this image we see the distinct anomalies that appear only in one of the algorithms:

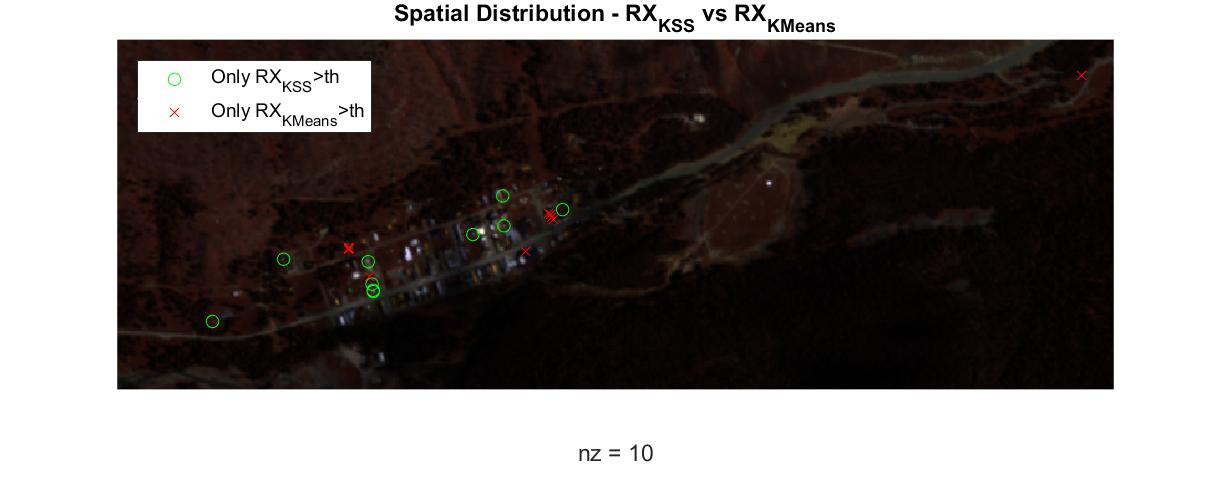


Figure 20: Direct cluster approach with cluster normalization,

distinct anomalies plotted on image separating each algorithm

As we can see there are 10 anomalies plotted on the image, all of them are in urban area of the image and they find different houses in the image.

In this image we see the distinct anomalies that appear in both of the algorithms:

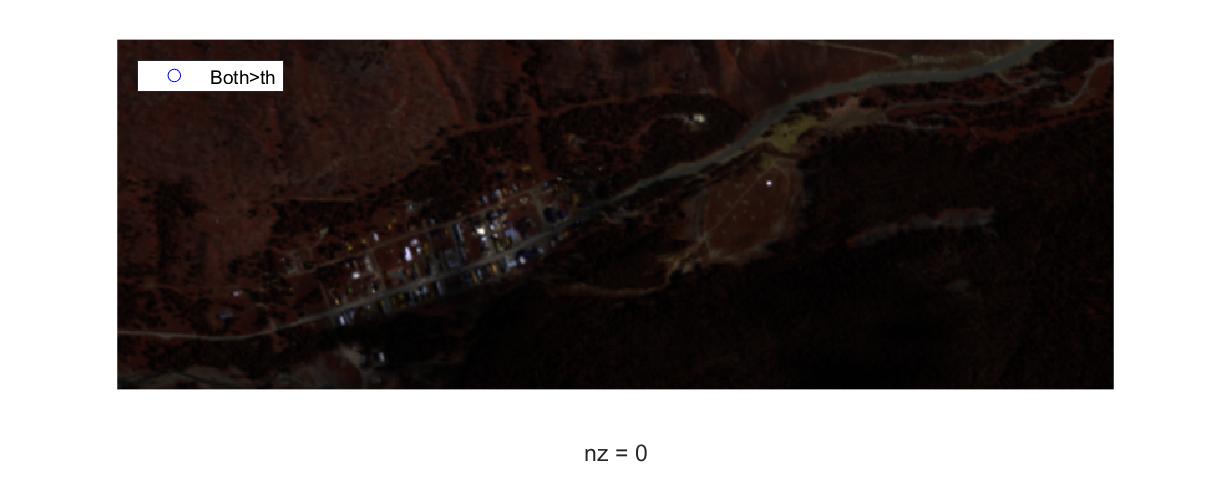


Figure 21: Direct cluster approach with cluster normalization,

distinct anomalies plotted on image that appear for both algorithms

As we can see there are none.  
In this approach the distinction between the different algorithms is very noticeable.

## The DirectCluster Approach with Whole Data Normalization

As we described earlier this is still the direct cluster approach meaning the scores are given in this form:

Where X is the hyperspectral image, S is the cluster (associated with the subspace), is the cluster mean, and is the inverse covariance matrix of the cluster.

The difference is in the normalization:

After getting the different scores from each cluster we normalize the scores according to the whole image:

Let’s represent the cluster scores as then the image scores are: and the normalized scores are: .

Using this we get the following RX scores, of the K-Means Vs the K-Subspaces:

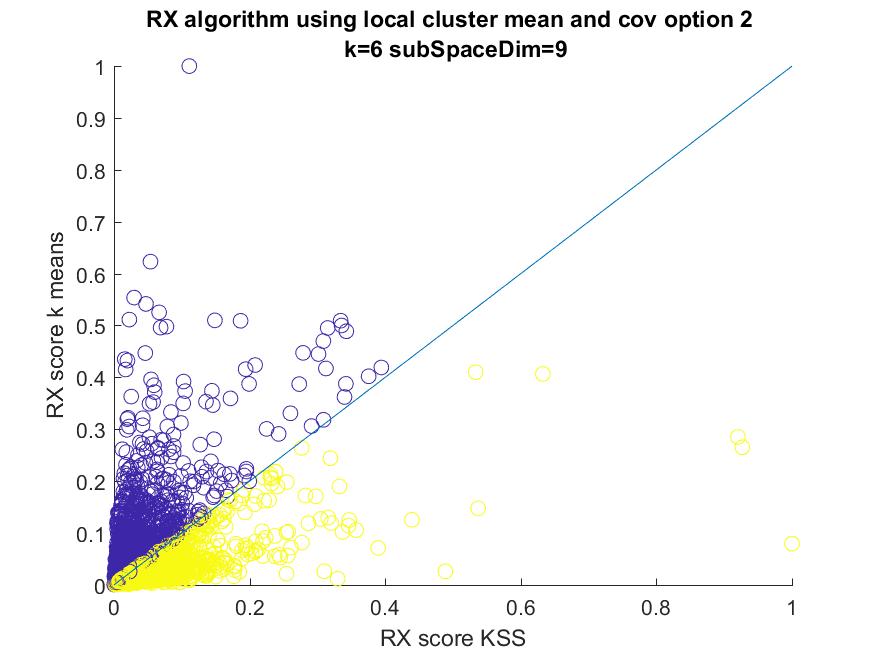


Figure 22: Direct cluster approach with whole image normalization RX scores

As we can see the dynamic range of the scores is smaller thus the gap between the high scored anomalies to the low scored ones is bigger than in the cluster normalization approach.

But still as was in the cluster normalization approach we get a vast difference in the anomalies detected using the different segmentations.

By plotting, only the distinct anomalies that we find (using the scatter plot to determine ) on the image we get the following:

In this image we see the distinct anomalies that appear only in one of the algorithms:

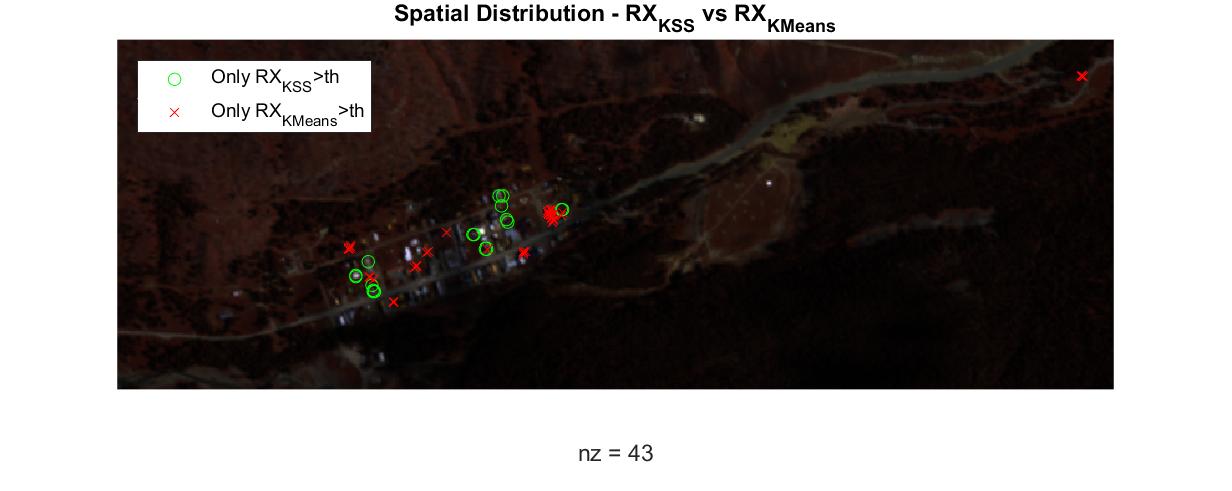


Figure 23: Direct cluster approach with whole image normalization,

distinct anomalies plotted on image separating each algorithm

As we can see there are 43 anomalies plotted on the image, most of them are in urban area of the image (only 1 is not) and they find different houses in the image.

In this image we see the distinct anomalies that appear in both of the algorithms:

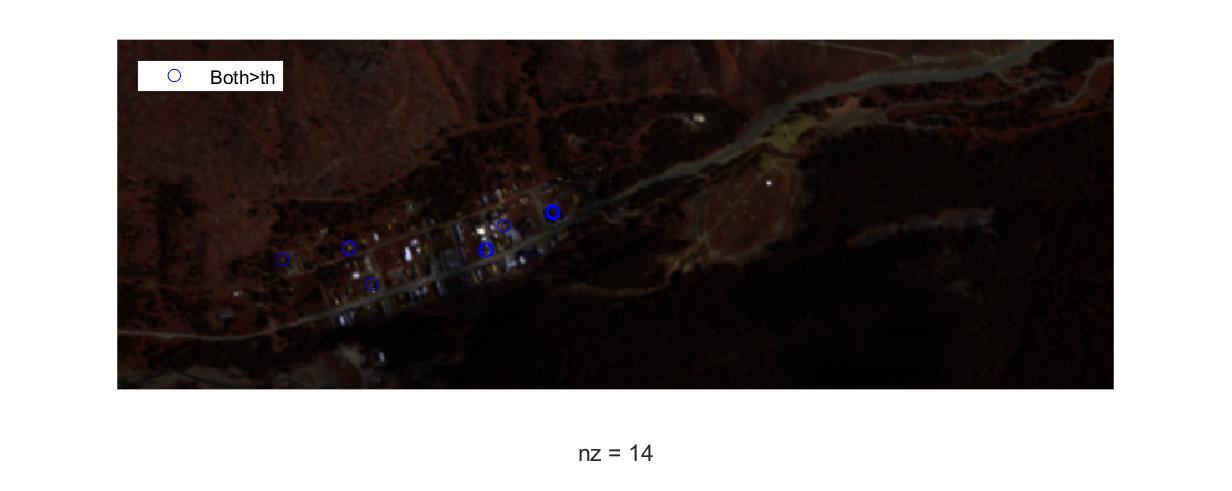


Figure 24: Direct cluster approach with whole image normalization,

distinct anomalies plotted on image that appear for both algorithms

As we can see there are 14 all of them are in urban area of the image.  
In this approach the distinction between the different algorithms is noticeable yet there is more common ground between the algorithms.

The anomalies found in both are particularly interesting for us, because they may indicate strong anomaly properties (since the 2 algorithms vary so much).

## The Indirect Cluster Approach with Cluster Normalization

Remembering the RX algorithm:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

In this approach we us the immediate neighbors of each pixel in the original image to estimate its value:

where are the immediate neighbors of :

Moreover, we use the inverse covariance matrix generated by the cluster.  
Thus, for each cluster separately we apply the RX algorithm in order to get   
“cluster scores”:

Where is the hyperspectral image, is the cluster (associated with the subspace), is the neighbors mean, and is the inverse covariance matrix of the cluster.

After getting the different scores from each cluster we us cluster normalization as discussed before.

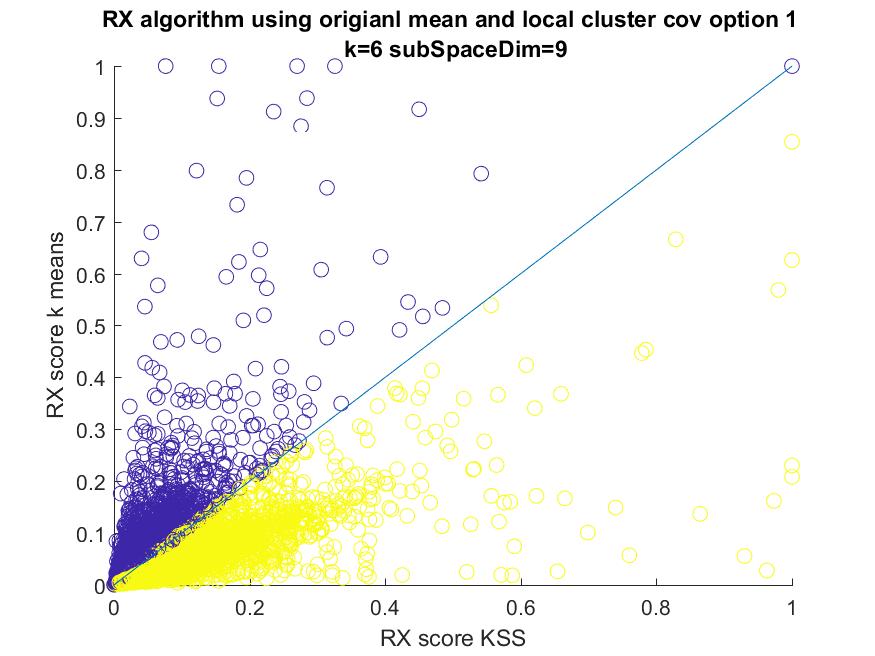
Using this we get the following RX scores, of the K-Means Vs the K-Subspaces:

Figure 25: Indirect cluster approach with cluster normalization RX scores

As we can clearly see there is a vast difference in the anomalies detected using the different segmentations, this means that some anomalies can only be found in one algorithm and not the other.

Moreover, in this method we get a high dynamic range of the scores  
(there are no big gaps between high scored anomalies and low scored ones).

By plotting, only the distinct anomalies that we find (using the scatter plot to determine ) on the image we get the following:

In this image we see the distinct anomalies that appear only in one of the algorithms:

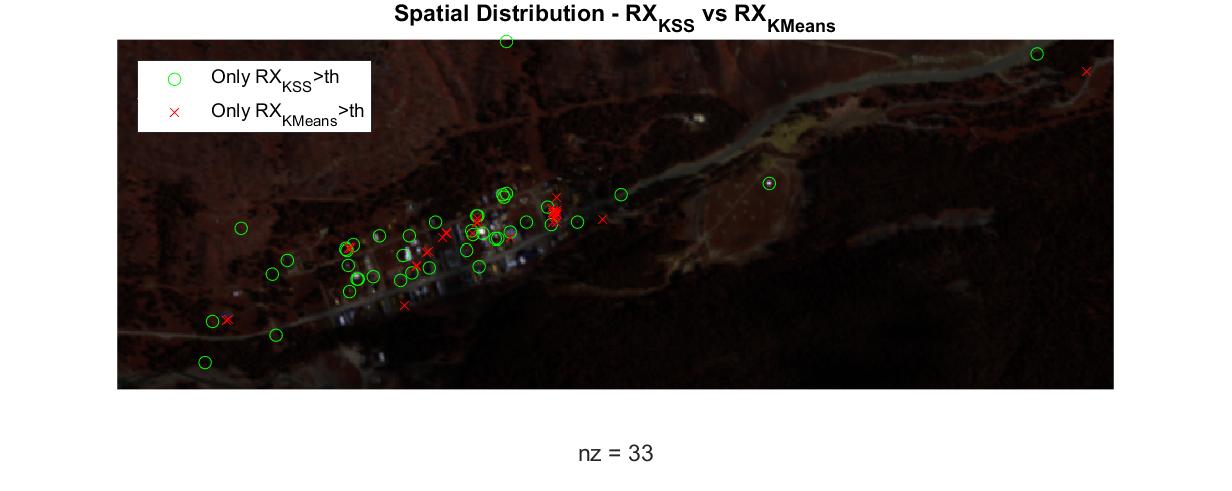


Figure 26: Indirect cluster approach with cluster normalization,

distinct anomalies plotted on image separating each algorithm

As we can see there are 33 anomalies plotted on the image, most of them are in urban area of the image and they find different houses in the image, but unlike the direct approach there are more anomalies (33 Vs 10) and more of the anomalies are scattered outside the urban area. This is somewhat expected since it is more likely to resemble the normal RX without clustering, therefore we predicted that there will be more common ground for the algorithms (as we will see in the next image), and that more anomalies that are based on surrounding rather than group association will pop up.

In this image we see the distinct anomalies that appear in both of the algorithms:

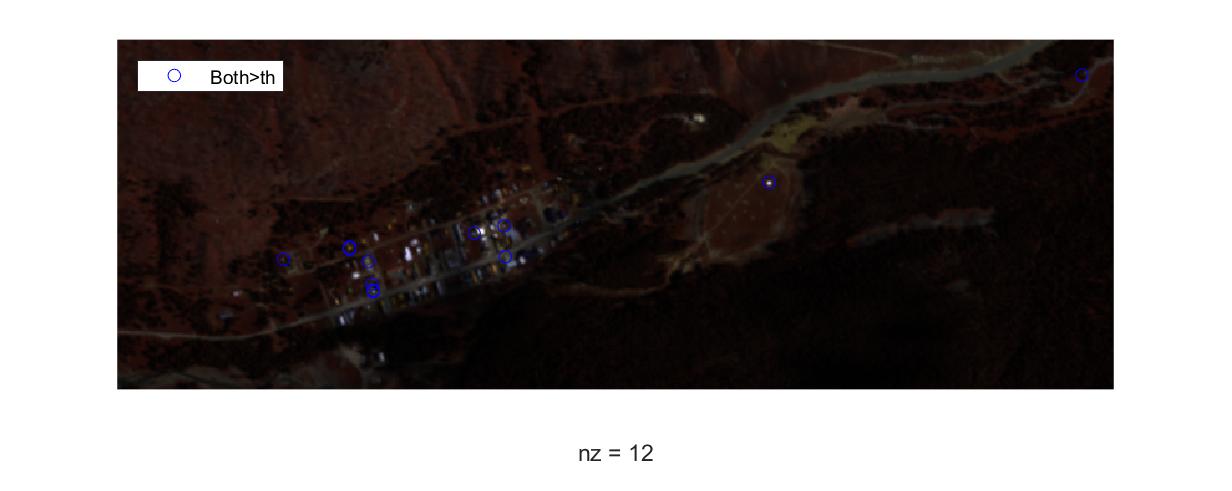


Figure 27: Indirect cluster approach with cluster normalization,

distinct anomalies plotted on image that appear for both algorithms

As predicted, there is more common ground: there are 12 anomalies   
(Vs 0 in the direct approach).  
In this approach the distinction between the different algorithms is noticeable yet there is more common ground between the algorithms.

There are 2 anomalies that are found outside the urban area.   
At least 1 of them is very noticeable since it is very different from its surrounding (the bright white dot in the middle of the brown field).

## The Indirect Cluster Approach with Whole Data Normalization

As we described earlier this is still the indirect cluster approach meaning the scores are given in this form:

Where is the hyperspectral image, is the cluster (associated with the subspace), is the neighbors mean, and is the inverse covariance matrix of the cluster.

The difference is in the normalization:

After getting the different scores from each cluster we us cluster normalization as discussed before.

Using this we get the following RX scores, of the K-Means Vs the K-Subspaces:

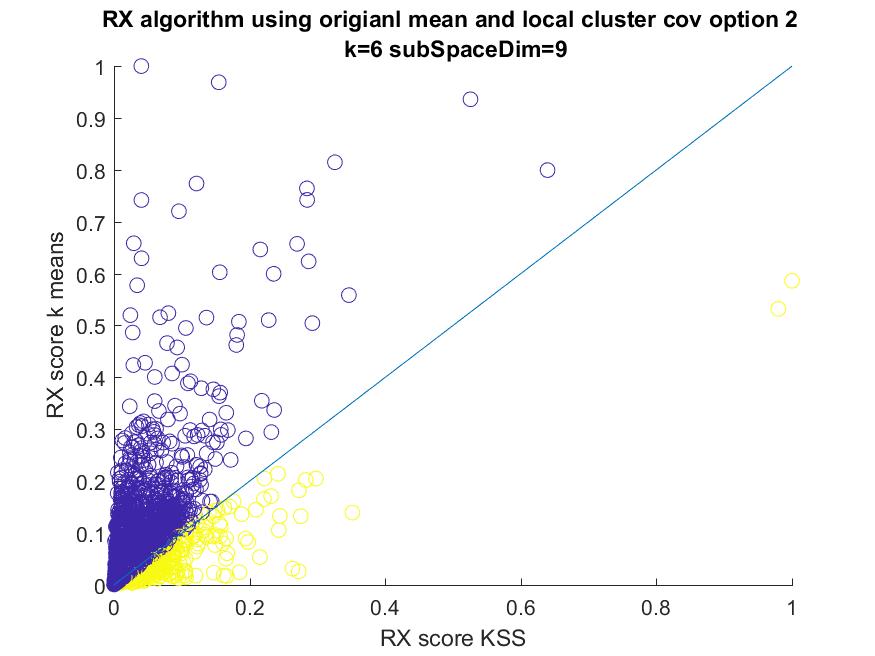


Figure 28: Indirect cluster approach with whole data normalization RX scores

As we can see the dynamic range of the scores is smaller thus the gap between the high scored anomalies to the low scored ones is bigger than in the cluster normalization approach.

But still as was in the cluster normalization approach we get a vast difference in the anomalies detected using the different segmentations.

By plotting, only the distinct anomalies that we find (using the scatter plot to determine ) on the image we get the following:

In this image we see the distinct anomalies that appear only in one of the algorithms:

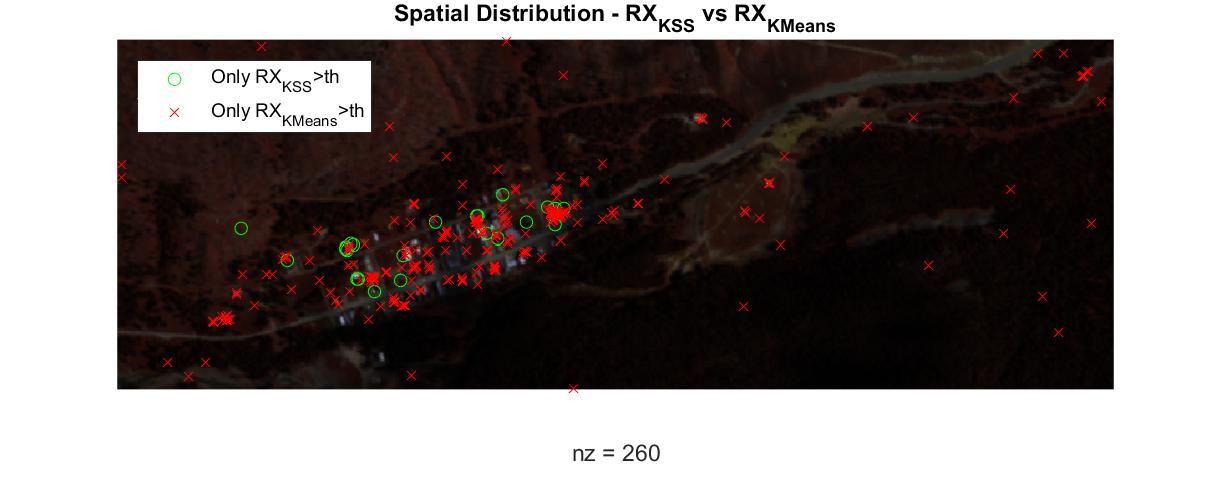


Figure 29: Direct cluster approach with whole image normalization,

distinct anomalies plotted on image separating each algorithm

As we can see there are 260 anomalies plotted on the image, most of them are in urban area of the image and they find different houses in the image.

In this case it is very clear that the K-Means algorithm find more anomalies than the K-Subspaces algorithm. This might mean that there are more false alarms using the   
K-Means algorithm.

In this image we see the distinct anomalies that appear in both of the algorithms:

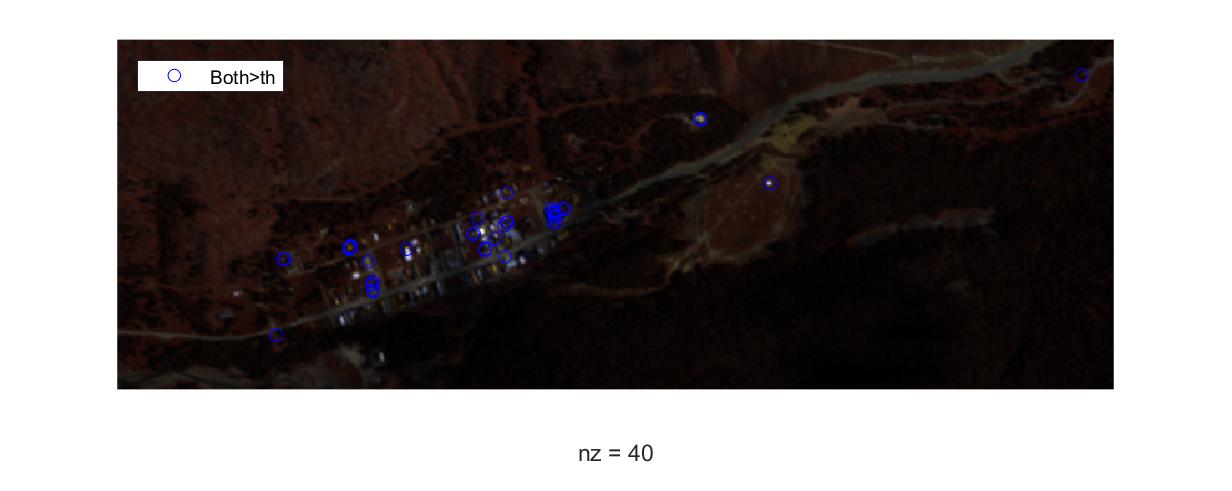


Figure 30: direct cluster approach with whole image normalization,

distinct anomalies plotted on image that appear for both algorithms

As we can see there are 40 most of them are in urban area of the image.  
In this approach the distinction between the different algorithms is noticeable yet there is more common ground between the algorithms.

Unlike the cluster normalization we find more common ground and more anomalies outside the urban area.

## Conclusions From Findings

All these findings indicate that there is a whole new range of anomalies that we can detect using K-Subspaces that we cannot detect otherwise (in comparison to   
K-Means). The nature of this anomalies is yet to be determined.

Furthermore, we notice that some anomalies keep appearing throughout the different tests for both algorithms, therefore it indicates a strong potential for this points to be desired anomalies. It taunts us to consider a different method of scoring for anomalies – maybe a weighted sum of the RX scores from different methods and K’s.

It might be beneficial to use both algorithms in future hyperspectral anomaly detectors in order to find desired anomalies.

It might be beneficial to use the K-Subspaces algorithm to find ingroup anomalies as compared to cross group anomalies.

# Further work to be done:

During our work on K-Subspaces algorithm we encounter many obstacles that opened us to new venues:

1. Dynamic subspaces algorithms – the K-Subspaces algorithm has 2 main problems:
   1. It assumes we know the subspace dimensions
   2. Proper initialization is required to get good results

For now, it seems we have a good initialization, but we want to look for more complex algorithms that may help find the subspace dimensions as well and explore new options for initializations (different initializations can yield different results since global optimum convergence is not assured).

1. Finding a metric for evaluating “performance” of our algorithm according to K parameter and the subspace dimensions.
2. Creating a new anomaly score based on a weighted sum of different anomaly scores- as we explained we created different RX scores for the same parameters.
3. Consider soft segmentation process (instead of we use   
    what is sometimes called in optimization “relaxation”).

# Conclusions

This project has shown that we can utilize unsupervised machine learning as a powerful tool for pre-processing of hyperspectral images, creating segmented data that can be used in various ways to detect anomalies.

It may allow us to detect different kinds of anomalies, further work and understanding of our various scoring methods need to be done in order to utilize it better.

It will be interesting to check this tool in the use of other target detection algorithms such as the Match Filter, ACE, Change detection etc.

# Bibliography

1. René Vidal, Yi Ma and S. Shankar Sastry “Generalized Principal Component Analysis”.
2. "K-Subspace Clustering -"Dingding Wang1, Chris Ding2, and Tao Li1 1   
   School of Computer Science, Florida International Univ., Miami, FL 33199, USA 2 CSE Department, University of Texas, Arlington, Arlington, TX 76019, USA
3. "Subspace Clustering using Ensembles of K-Subspaces" -John Lipor1 , David Hong2∗ ,   
   Yan Shuo Tan3 , and Laura Balzano2 1Department of Electrical & Computer Engineering, Portland State University 2Department of Electrical Engineering & Computer Science, University of Michigan 3Foundations of Data Analysis Institute, University of California, Berkeley

(23 Sep 2018).

1. “A Novel Detection Paradigm and its Comparison to Statistical and Kernel-Based

Anomaly Detection Algorithms for Hyperspectral Imagery”-Colin C. Olson and Timothy Doster

U.S. Naval Research Laboratory 4555 Overlook Ave., SW, Washington, DC

1. <http://dirsapps.cis.rit.edu/blindtest/> [↑](#footnote-ref-1)
2. In order to try and avoid converging to empty subspaces, if a sub space has less data points associated with it then its dimension, we randomize a new subspace to replace it for the following iteration. [↑](#footnote-ref-2)