Ben-Gurion University of the Negev

Faculty of Engineering Science

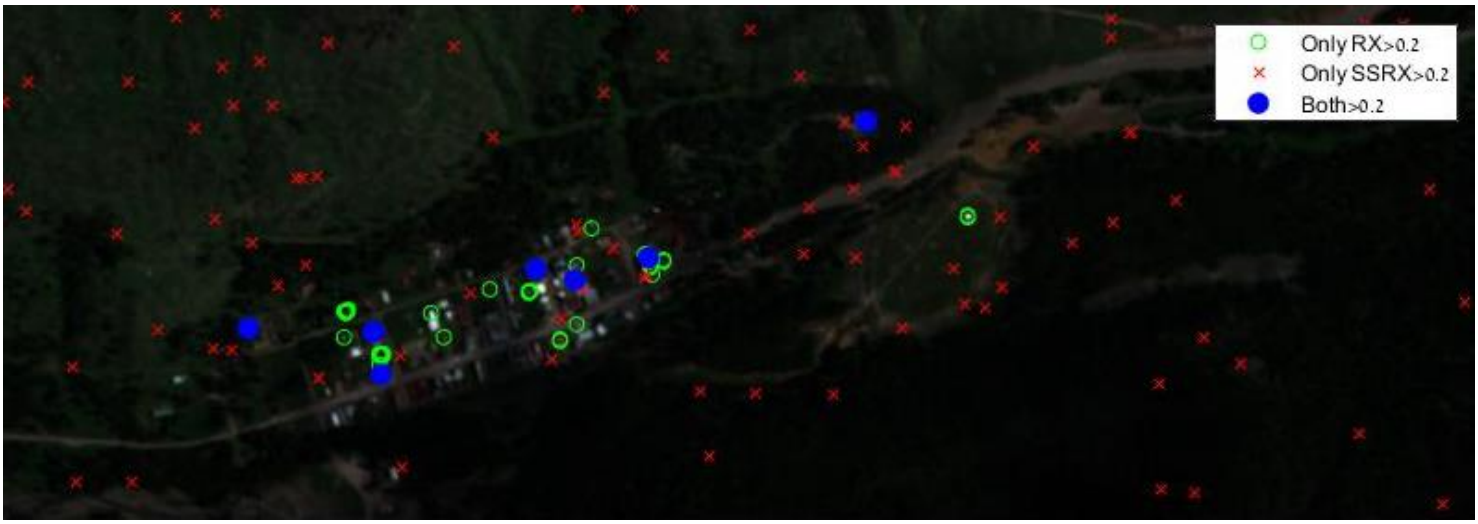
School of Electrical and Computer Engineering

Dept. of Electrical and Computer Engineering

Selected Topics in Image Processing

Course Project

Comparing RX and SSRX Algorithms



Students (name & ID):

Serebro Doron, I.D 204728919

Supervisors: Prof. Rotman Stanley

Submitting date: 01/06/2020

Contents

Introduction

The goal of this project is to compare two anomaly detection algorithms for hyperspectral images – the RX algorithm and its subspace projection variation known as the SSRX algorithm. While these two algorithms are mathematically and logically similar, the different results achieved when applying them on real data are interesting to research.

In order to compare RX and SSRX, a series of experiments were preformed, that introduced different metrics for evaluating the differences and commonalities between the 2 algorithms. Notable metrics are top X anomalies, spatial distribution of anomalies, shared/unique anomalies, and others.

For the innovation part of this project I chose to compare RX and SSRX in the context of change detection rather than anomaly detection. Since the Chrono-chrome algorithm relies heavily on the RX algorithm, this comparison was almost called for. This new perspective provided meaningful insight that proved to be quite redeeming for the SSRX algorithm.

An important caveat for this project is the that all observations were made based on only 3 images. This makes results and conclusions less likely to be generalized for common use cases of RX and SSRX. That being said, the results still shed some light on the benefits of each algorithm.

Background

Anomaly Detection

Anomaly detection refers to a set of techniques and algorithms that aim to identify rare items, events or observations which raise suspicions by differing significantly from the majority of the data. In hyperspectral imagery (HSI), these algorithms prove especially useful and are commonly used.

A common model for anomaly detection in HSI relies in the distinction between two possible hypotheses in the image:

$$H_0: \quad X = Bb + n$$
$$H_1: \quad X = Tt + n$$

The background hypothesis H_0 and the target hypothesis H_1.

- X – a random variable whose instances x are spectral signatures of different measured pixels in the image.
- T – a matrix with independent columns that span the subspace of all possible anomalous signatures.
- t – a column vector of unknown amplitudes. Together with T, the expression Tt represents the anomalies within the image.
- B – a matrix with independent columns that span the subspace of all possible background signatures.
- b – a column vector of amplitudes. Together with B, the expression Bb models the background.
- n – common gaussian noise vector. $n \sim N(0, \phi_n)$

According to this model anomalies are rare in the data, therefore, H_0 is almost always true and our goal is to locate the few pixels in which H_1 is true. Different anomaly detection algorithms are based on different assumptions regarding the subspaces spanned by T and B.

The 2 algorithms discussed in this project are RX and SSRX.

RX Algorithm

The RX algorithm is based on the Mahalanobis distance[1] and is given by:

$$RX: (x-m)^T \phi_X^{-1} (x-m) > \eta$$

where x is a given pixel, m is the local mean of x, ϕ_X is the empiric covariance matrix of X and η is a threshold for detection.

This algorithm assumes no prior knowledge of T, meaning that anomalies are uniformly distributed across the image. Given this assumption, the generalized likelihood ratio test (GLRT) becomes proportional to the RX expression[2].

An intuition for this algorithm can be based on inner products and noise whitening:

The term $(x-m)$ can be seen as an estimation error of a given pixel based on its neighbors. An anomalous pixel will be poorly estimated in this method (since its quite different than its neighbors), therefore, it will have a large value of $(x-m)$. Recalling that the maximal inner product achieved for a given vector is with itself, we can understand why the more anomalous a pixel is, the higher RX score it will receive.

As for the inverted covariance matrix, it is added to the algorithm in order to whiten the noise in the image. This way, the noise in the data can match our assumed model of white gaussian noise.

Observing the RX algorithm in the hyperspace, yields an ellipsoid decision surface, where every observation outside of the ellipsoid is considered anomalous as seen in figure 1 below:

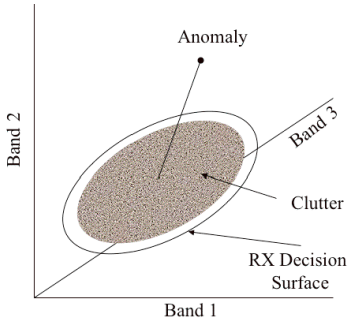


Figure 1 - RX descision surface

[1] Mahalanobis distance represents how far an observation is from a given distribution.
[2] See Bibliography article #1 for the full derivation.

SSRX Algorithm

While the RX is an optimal detector for uniformly distributed anomalies, it has been observed that in real cases anomalies are not uniformly distributed. The SSRX algorithm aims to adapt the RX algorithm to overcome this issue.

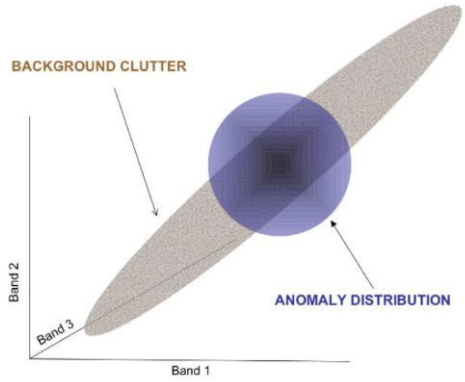


Figure 2 a more realistic model for anomaly distribution

This is done in the following way:

1. Perform principle component analysis (PCA) on the image. Project the image onto a lower dimension by omitting a subset of eigenvectors from V that correspond to the largest eigenvalues.

$$\phi_x = V^t \Lambda V = [V_1 \cdots V_N]^T \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_N \end{bmatrix} [V_1 \cdots V_N], \quad \lambda_1 > \lambda_2 > \cdots > \lambda_N$$

$$\widetilde{V_q} = \left[V_1 \cdots V_{N-q}\right]^T, \quad q = number\ of\ deleted\ PCs$$

$$\tilde{X} = \widetilde{V_q}^T \phi_X^{-1} (X - \mu_X)$$

2. Calculate RX scores for the new image

$$SSRX: (\tilde{x} - m)^T \phi_{\tilde{x}}^{-1} (\tilde{x} - m) > \eta$$

The motivation for deleting the high variance PCs stems from real data observations. In outdoor HSI, it was observed that the first few PCs have a variance that is several orders of magnitude larger than that of all the next PCs combined. These observations make sense once we recall that H_0 is almost true. It means that the largest variance PC in HSI almost always contribute mostly to the background, and therefore, eliminating them would help us filter out noise and be more sensitive to anomalies in the remaining subspace (spanned by the rest of the PCs).

Change Detection

Change detection is a unique problem of anomaly detection given 2 images where the goal is to locate areas of anomalous change. This is useful for comparing images of the same area that were taken over time (days, months, or years).

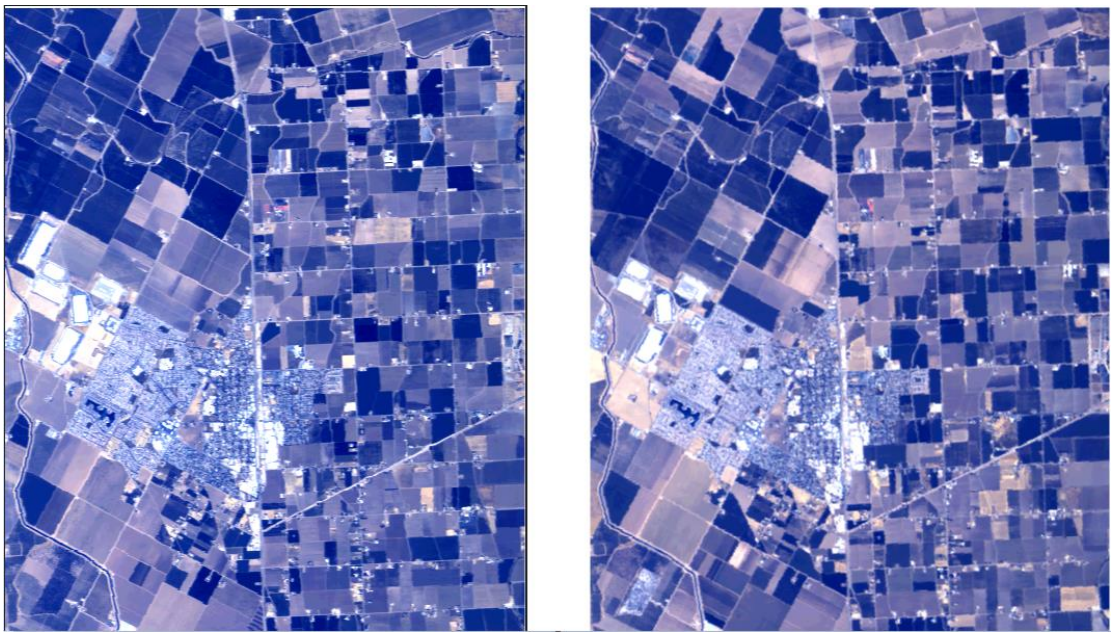


Figure 3 Bay area images.

Left: Taken in 2013. *Right: Taken in 2015*

Chrono-chrome Algorithm

This change detection algorithms utilizes the RX algorithm in order to detect changes between 2 images.

1. Estimate Y from X and the cross-covariance matrix ϕ_{XY}:

$$\tilde{Y} = \phi_{XY}\phi_X^{-1}X$$

(This is a least squares estimator of Y given X)

2. Calculate estimation error

$$\varepsilon = \tilde{Y} - Y$$

3. Run the RX algorithm on the estimation error

$$CC: \quad (\varepsilon - m)^T \phi_\varepsilon^{-1}(\varepsilon - m) > \eta$$

The intuition for this algorithm is as so: In pixels that did not change significantly between the 2 images, the estimation error should be small and therefore their score will be small. In contrast, changed pixels would cause a large estimation error, thus causing an increased CC score.

Methodology

The dataset used for this project is the self-test image from the RIT target detection dataset[3]. 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 4 Cooke City, MT

Data Generation

The first step was to run the RX and SSRX algorithm in order to generate data for comparison and evaluation. The following algorithms were run and saved:

1. Classic RX
2. SSRX with varying number of high-variance PCs dropped:

$$q\epsilon\{2,5,10,20,50,75,100,120\}$$

3. SSRX with varying number of low-variance PCs dropped:

$$q\epsilon\{5,20,50,75,100\}$$

4. Classic Chrono-Chrome
5. Subspace Chrono-Chrome with varying number of high-variance PCs dropped:

$$q\epsilon\{50,100,150,200\}$$

Note! Chronochrome (CC) and Subspace Chronochrome (SSCC) are for the innovation part of this report (see details in the innovation section).

Data Exploration

In order to compare between the 2 algorithms, different graphs and visualizations were made. Due to a lack of ground truth or a large HSI dataset, this method of comparison was the best available. Visualizations used:

- Spatial distributions
- Scatter plots
- Histograms
- More…

[3] http://dirsapps.cis.rit.edu/blindtest/

Findings

1. Top 10 anomalies remain the same for RX and SSRX
   Plotting the 10 pixels with the highest SSRX values yields the following image:

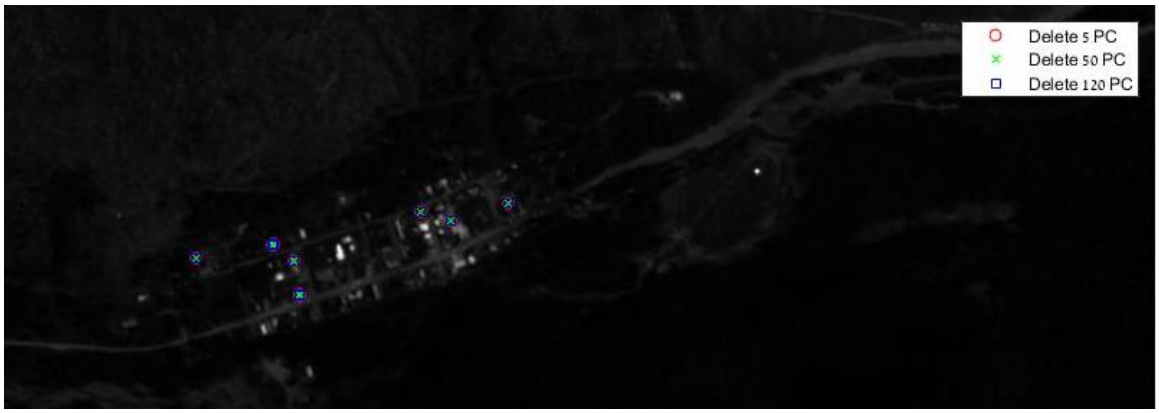


Figure 5 Top 10 SSRX anomalies

All anomalies overlap, meaning that dropping more PCs does not completely change the results of the algorithm. Plotting these results along with the top 10 RX anomalies yields a similar image:

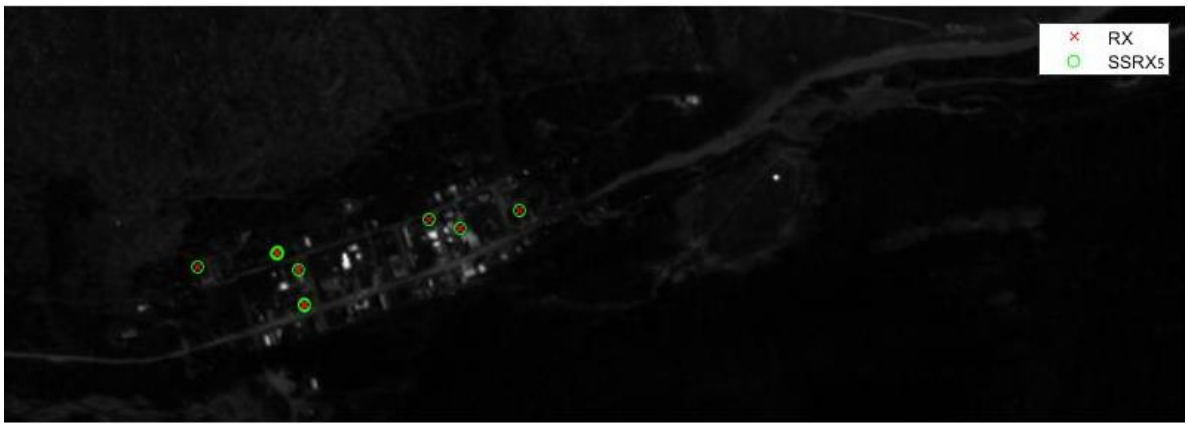


Figure 6 Top 10 RX vs SSRX

This is a great sanity check that proves the SSRX is able to successfully detect the 10 most anomalous pixels in the image.

2. Visualizing unique detections
   while top anomalies remain the same, there is still a noticeable difference between the 2 algorithms. This can be clearly illustrated by choosing a threshold and plotting all anomalies together.

   For this illustration, I chose the 3 sigma threshold, a very popular threshold in anomaly detection:

$$\eta = \mu_{RX} + 3\sigma_{RX}$$

   This means that every pixel who's RX score is larger than the mean score by more than 3 standard deviations is considered an anomaly (the same applied to SSRX).

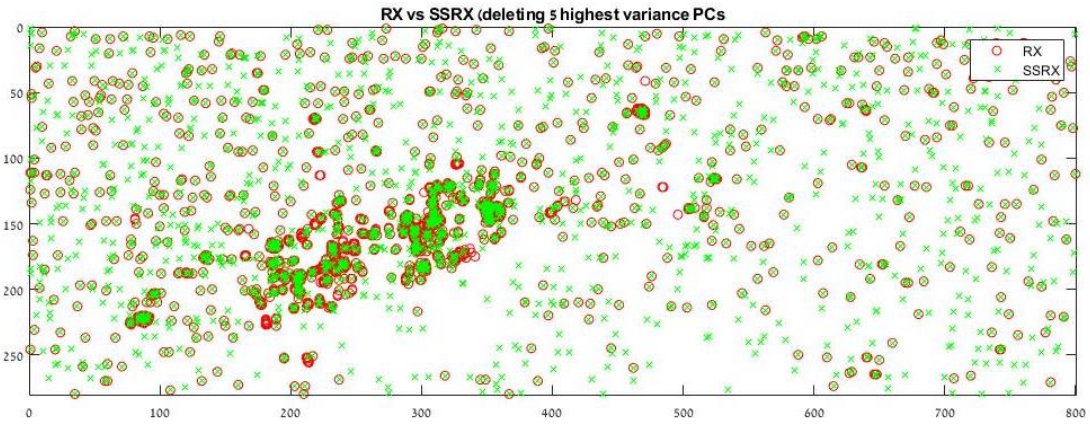


Figure 7 RX vs SSRX: 3 Sigma anomalies

This image shows that the RX is superior when using the 3-sigma threshold. In this case SSRX detects all anomalies detected by RX but adds many more detections which seem like noise.

In order to better see this difference, it is best to look anomalies unique to RX and anomalies unique to SSRX separately:

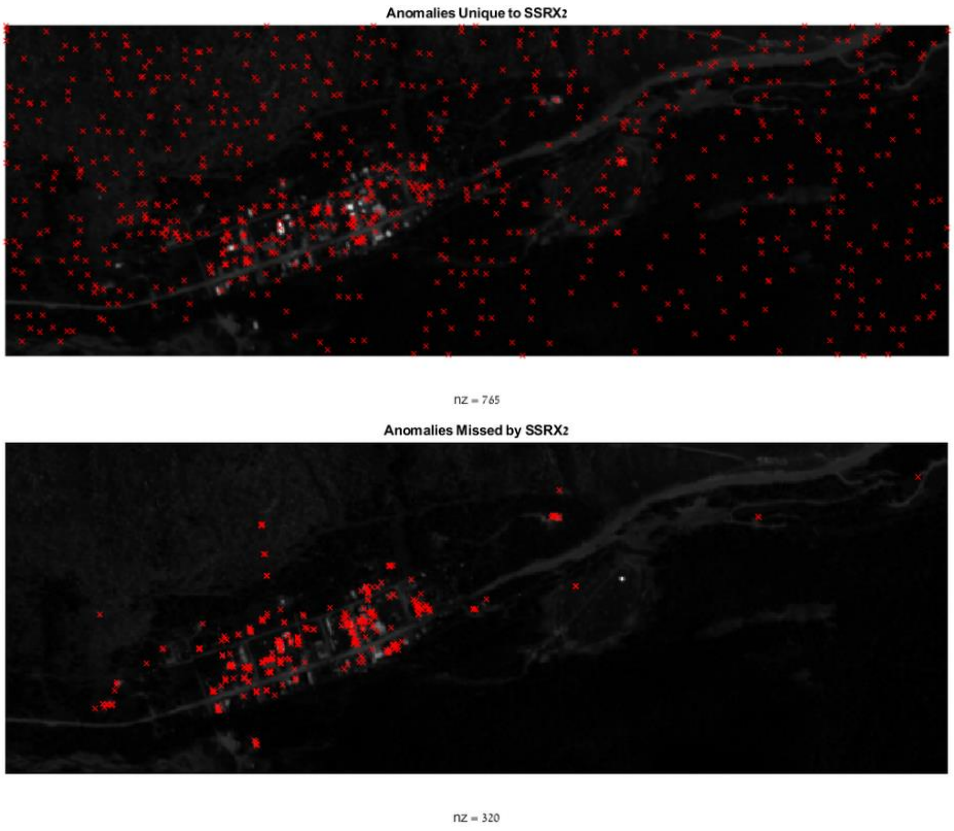


Figure 8 Unique anomalies

The above images are obtained by comparing RX with SSRX2 (dropping only 2 PCs). It is clear to see that SSRX adds a lot of anomalies across the entire image while RX anomalies are centered around the city.

3. The more PCs dropped, the less corelated the results of the 2 algorithms:
   Figure 9 is made up of 6 scatter plots of SSRX scores as a function of RX scores. Starting with the top left, we can see clear linear correlation between RX and SSRX with minimum number of PCs dropped. Moving to the bottom right, we can see that correlation decrease, until the point where after dropping 100PCs, RX and SSRX provide vastly different scores.

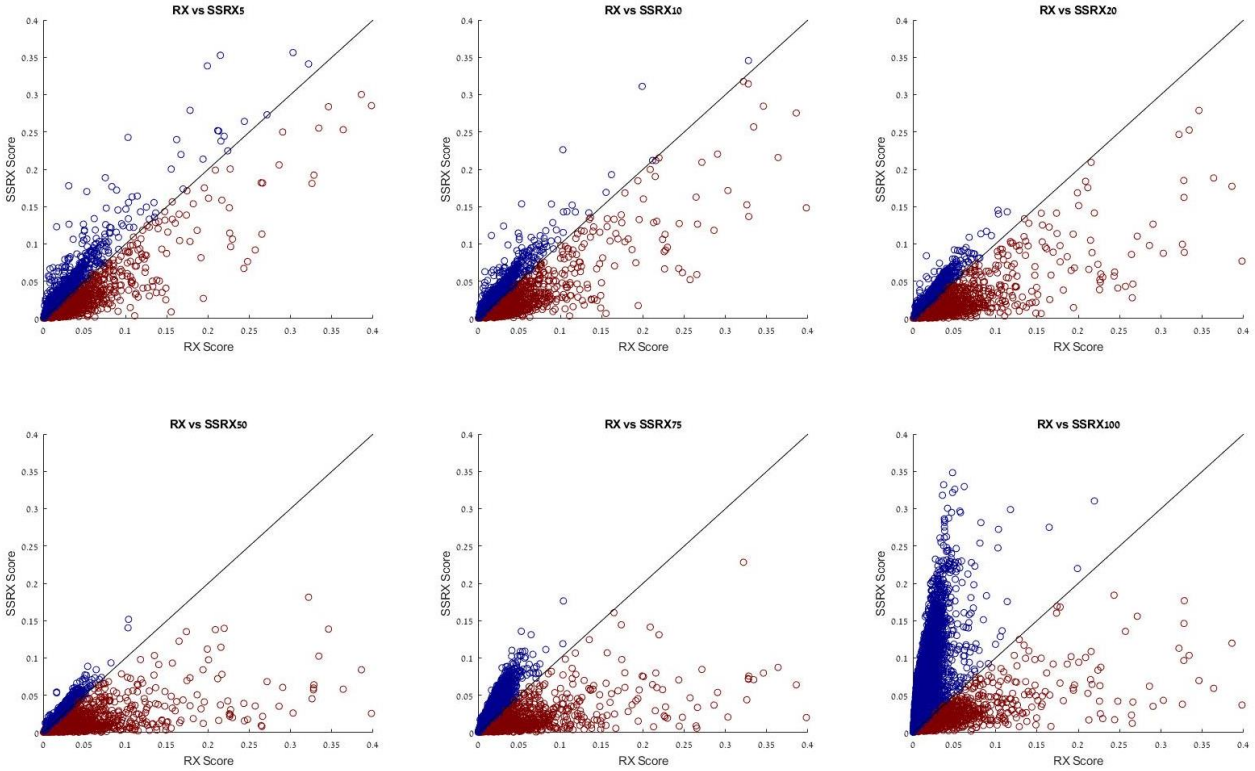


Figure 9 Scatter plots of RX vs SSRX

Note – Dropping low-variance PCs hardly affects correlation, no matter how many. This proves that RX is heavily influenced by the first few PCs.

4. Deep Dive into SSRX10 and SSRX100
   In order to better understand the scatter plots, they were partitioned into quadrants by a threshold on each axis:

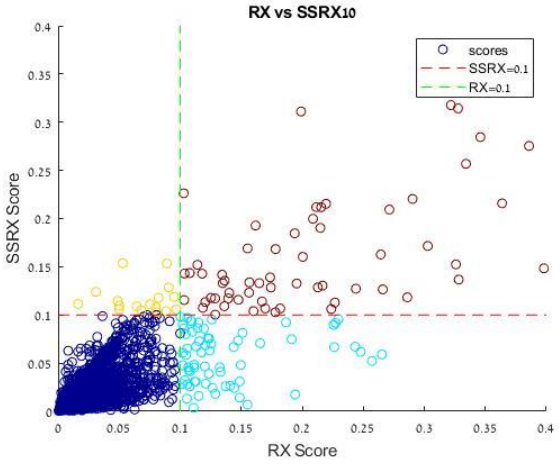


Figure 10 Partitioned scatter plot

The partitioned scatter plot can be better explained. The dark blue pixels are below both thresholds and are therefore noise. The red pixels are anomalies that were detected by both algorithms. The remaining pixels, yellow and light blue, are anomalies detected by only the SSRX or only the RX accordingly.

Let us visualize them spatially:

Figure 11 RX vs SSRX10 Anomalies for th=0.1

Both algorithms detect similar anomalies inside the city, but it seems like RX is slightly better at detecting such urban anomalies while SSRX is slightly better at locating anomalies withing the vegetation.

This behavior is more clearly visible when plotting the same graphs for SSRX100:

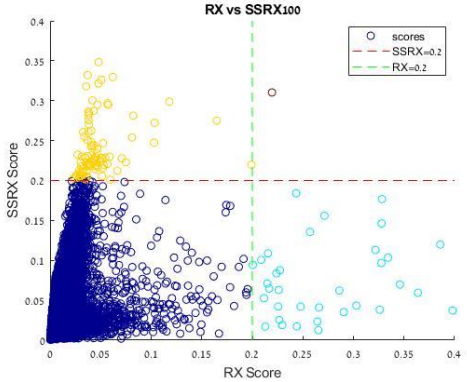


Figure 12 SSRX100 scatter plot

This time the threshold was increased to 0.2 in order to keep the number of plotted anomalies roughly the same:

Figure 13 RX vs SSRX100 Anomalies for th=0.2

Unlike the previous graph, this time most anomalies are not common to the 2 algorithms, yet trend remains: RX finds most of the urban anomalies while SSRX anomalies are mostly around vegetation.

Conclusions

All of these findings indicate that deleting some PCs allows for the detection of meaningful anomalies that would have been otherwise discarded by regular RX. That being said, the price of these new anomalies comes at the cost of more potential noise and losing good anomalies detected by RX. In order to overcome this, it might be beneficial to use both algorithms in future hyperspectral anomaly detectors.

Innovation – Subspace Chronochrome

This part of the project is meant to expand upon the ideas introduced earlier, specifically by moving from the general anomaly detection domain to that of anomalous change detection.

Since the Chronochrome algorithm is essentially RX algorithm where the input is a linear estimation error, it is possible to utilize SSRX and RX interchangeably to compare the two algorithms in the context of change detection.

Dataset

The data used for this task is a hyperspectral change detection dataset[4] containing two images of the California bay area. The images were taken date back to 2013 and 2015 and were taken with the AVIRIS sensor. The images are of size 600x500x224 and depict the surroundings of the city of Patterson, CA (see figure 3 for RGB images).

Methodology

The main idea was to follow the same methodology used for RX and SSRX, but instead use Chronochrome (CC) and Subspace Chronochrome (SSCC). This includes running CC and multiple variations of SSCC by dropping an increasing number of PCs. In this case, the number of dropped PCs was 50,100,150 and 200.

Findings

When looking at scatter plots, results match those of RX and SSRX. Dropping more PCs reduces correlation between the 2 algorithms and creates.

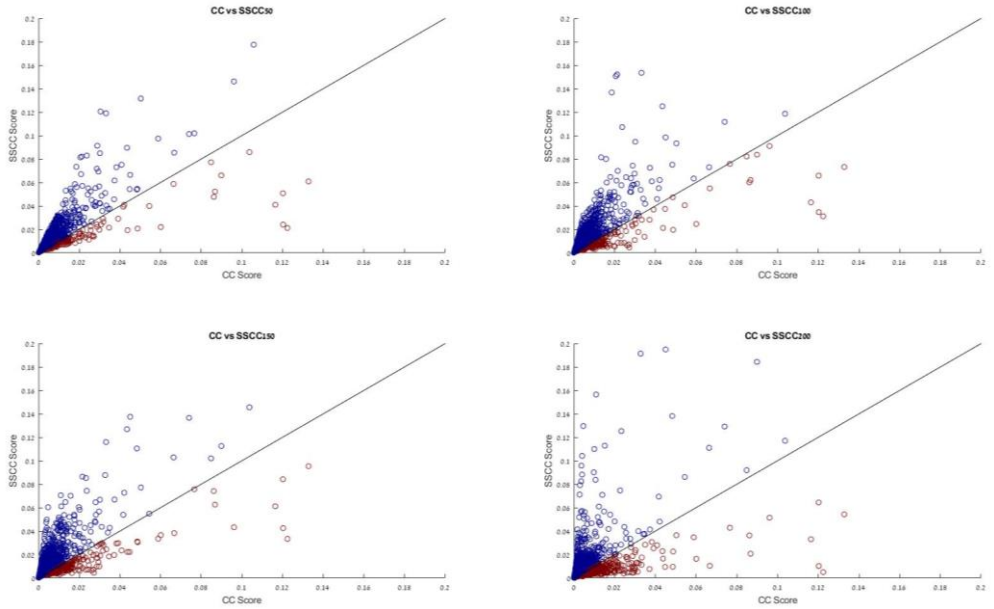


Figure14 CC and SSCC scatter plots

[4] Can be found here: https://citius.usc.es/investigacion/datasets/hyperspectral-change-detection-dataset

Where things get more interesting is when we look at the spatial distribution of the different anomalies – here SSCC truly shines. For example, when setting $th = 0.05$ and comparing CC (blue) to SSCC50 (yellow) seems like CC detects only 5 changes, all of which were shared with SSCC (red):

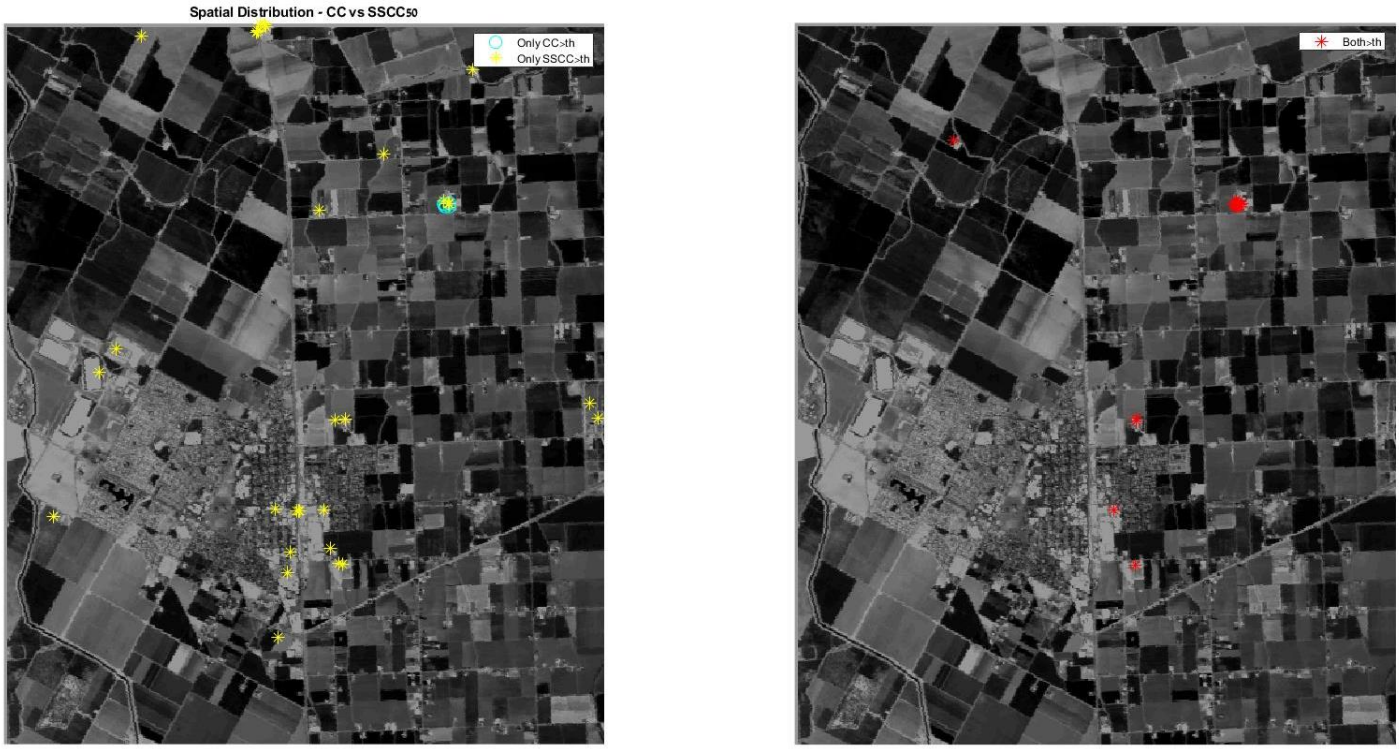


Figure 15 CC vs SSCC50, th=0.05

Unlike SSRX, which proved to contribute a significant amount of noise, seems like the SSCC algorithm is able to add valuable information without adding too much noise.

Then again, all good things come to an end – when comparing CC to SSCC200 with the same threshold, we get a different result:

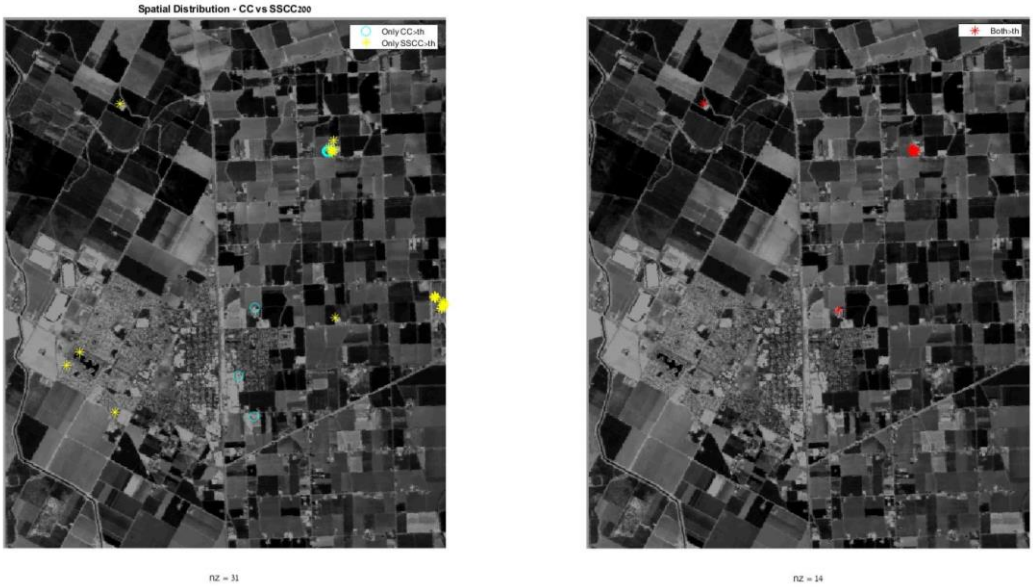


Figure 16 CC vs SSCC200, th-0.05

When we drop too many PCs, SSCC is still able to detect some key chages and some new ones that are not detect by CC , but the SSCC now misses 3 changes that were previously detected by both algorithms. This is in line with the findings regarding SSRX – dropping a few PCs is usually beneficial, but you have to be careful not to drop too many, otherwise you risk either too many false alarms or an increasing amount of false negatives.

Another interesting finding is the fact that dropping PCs allowed for a better variance of change detection scores. If we look at the scatter plot in figure 17 below, we can clearly see that for the same threshold value SSCC detects more changed pixels than CC.

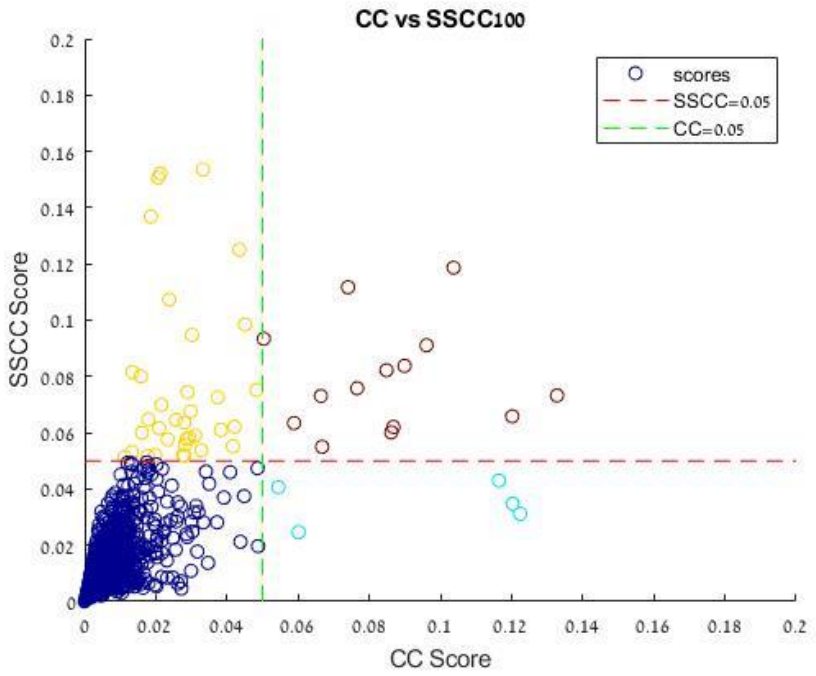


Figure 17 CC vs SSCC100 scatter plot, th=0.05

This property, along with the bigger spatial distribution indicate that projecting our estimation error to a smaller sub-space can greatly contribute to our ability to detect anomalous change. This method of evaluation allows us to visualize the precision-recall trade-off where CC tend to be more precise, SSCC tend to have better recall.

Moreover, these types of visualizations can provide meaningful insight when trying to determine exactly how many PCs should be dropped when implementing SSCC.

Conclusions

This project has shown that while RX and SSRX can both detect the most anomalous pixels in an image, each detect a different subset of less anomalous pixels. RX anomalies are usually much more spatially dense in smaller regions of the image, and SSRX anomalies are almost uniformly spread across the image. Given some prior knowledge of the distribution of expected anomalies, one might use this knowledge to prioritize one algorithm over another.

This project has also compared between RX and SSRX under the scope of change detection and presented consistent results. That being said, in change detection the spatial denseness of the RX anomalies was exaggerated further, making SSRX anomalies significantly more appealing. This is almost counter-intuitive since in change detection it is expected for anomalous change to be more uniformly distributed, thus making the RX detector ideal, but as experiments show, the bigger PCs are still mostly noise related and dropping them is beneficial. Another argument for the benefits of SSRX in change detection can be made claiming that estimation error of background pixels (due to change in reflectance or misregistration) is increasing the variance of the already big background PCs. Thus, deleting these PCs is now even more important in order for our algorithms to be accurate and robust.

Lastly, this project provided a set of experiments and visualization that allow for careful inspection of algorithms parameters such as the number of dropped PCs. These tools allow us to make more informed decisions for choosing one algorithm over another, and fine-tuning our parameters to maximize our needs.

Bibliography

1. Alan P. Schaum and Alan D. Stocker "Joint hyperspectral subspace detection derived from a Bayesian likelihood ratio test", Proc. SPIE 4725, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII, (2 August 2002)

2. Alan P. Schaum and Alan Stocker "Hyperspectral change detection and supervised matched filtering based on covariance equalization", Proc. SPIE 5425, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery X, (12 August 2004)

3. A. P. Schaum "Hyperspectral anomaly detection beyond RX", Proc. SPIE 6565, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII, 656502 (14 May 2007)

4. Sofia Aizenshtein, Ido Abergel, Moshe Mailler, Gili Segal, and Stanley R. Rotman "Non-negative matrix factorization for hyperspectral anomaly detection", Proc. SPIE 11392, Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXVI, 1139210 (24 April 2020)