

Article

# Global and Local Real-Time Anomaly Detectors for Hyperspectral Remote Sensing Imagery

Chunhui Zhao <sup>1,\*</sup>, Yulei Wang <sup>1,\*</sup>, Bin Qi <sup>2</sup> and Jia Wang <sup>1</sup>

<sup>1</sup> College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China; E-Mail: wangyulei@hrbeu.edu.cn

<sup>2</sup> College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China; E-Mail: qibinwinter@gmail.com

\* Authors to whom correspondence should be addressed;  
E-Mails: zhaochunhui@hrbeu.edu.cn (C.Z.); wangyulei@hrbeu.edu.cn (Y.W.).

Academic Editors: Gonzalo Pajares Martinsanz and Prasad S. Thenkabail

Received: 23 October 2014 / Accepted: 18 March 2015 / Published: 1 April 2015

---

**Abstract:** Anomaly detection has received considerable interest for hyperspectral data exploitation due to its high spectral resolution. A well-known algorithm for hyperspectral anomaly detection is the RX detector. A number of variations have been studied since then, including global and local versions for different type of anomalies. Aiming at a real-time requirement for practical applications, this paper extends the concept of global and local anomaly detectors to be real-time detectors. The algorithms exploit the fact that a true real-time detector must produce its output in a causal manner and at the same time as an input comes in. Taking advantage of the Woodbury matrix identity, the global and local real-time detectors can be implemented and processed pixel-by-pixel in real time. Both synthetic and real hyperspectral imagery are conducted to demonstrate their performance.

**Keywords:** hyperspectral remote sensing; anomaly detection; real-time; Woodbury matrix identity; sliding local window

---

## 1. Introduction

Hyperspectral imagery (HSI) processing methodology has generally become increasingly important due to its very high spectral resolution and capability of uncovering many subtle materials that cannot

be known by traditional remote sensing monitors. Anomaly detection (AD) is very important for practical use since, in most applications, no prior knowledge of the target spectrum or background information is available before the process of detection—for example, special species in agriculture and ecology, rare minerals in geology, toxic wastes in environmental monitoring, oil spills in water pollution, drug trafficking and smuggling in law enforcement, man-made objects in battlefields, unusual terrorism activities in intelligence gatherings, tumors in medical imaging, *etc.* [1,2]. Since no prior knowledge is assumed, the detection is typically based on a statistical characterization of the multivariate data. Those pixels with significantly different spectral signatures from their neighboring background pixels or the global background pixels are detected as anomalies.

Many types of anomaly detectors have been proposed in the literature [3–16]. They fall into two categories, global and local methodologies. In global AD algorithms, small rare targets with significantly different spectral signatures from global spectral statistics are defined as anomalies. The most popular global AD algorithm, developed by Reed and Xiaoli Yu, refers to an RX detector [4], which is considered the benchmark AD approach for multi- and hyperspectral imagery. Basically, an RX detector is a constant false alarm rate (CFAR) detector derived by the generalized likelihood ratio test (GLRT). It actually performs a Mahalanobis Distance calculation between the pixel being processed and the background information. In [6], Kwon presents a nonlinear model of RX detector, called Kernel RX. The Kernel RX detector achieves a better performance by mapping the original input space to a higher-dimensional space via a non-linear function, especially when the original signatures are mixed in a non-linear model, which is always the case in reality. Thanks to the “kernel-trick” [17], the Kernel RX detector could be implemented in the higher-dimensional space by means of kernel functions defined on pairs of dot products of input data, without identifying the non-linear mapping function. Chang presents many other global AD algorithms, including Low Probability Detection (LPD) [18], Constrained Energy Minimization (CEM) [19,20], *etc.* Unlike the RX detector, the LPD algorithm uses unity vector with ones in all the components as its match signal, while RX uses the pixel being processed. CEM, on the other hand, is a target detection algorithm that could detect small and sub-pixel targets of interest in the hyperspectral imagery. The target signature could be obtained in advance by endmember extraction algorithms or other pre-processing techniques.

Local AD algorithms, on the other hand, are of importance due to the weakness of global AD algorithms when the targets are extremely small or only abnormal in a local background but buried in the global background. The local AD algorithms are designed to detect pixels whose spectral signatures are conspicuous or inconspicuous in relation to surrounding pixels. Variants of local AD algorithms consist of applying locally the same concept of a sliding window centered by the pixel under test, of which the most popular is the local RX detector derived from the benchmark RX detector [12–14]. There are also many other local AD algorithms in the literature; for example, Kwon *et al.* recently developed the so-called dual window-based eigenspace separation transform (DWEST) [15]. An alternative approach, called nested spatial window-based target detection (NSWTD), uses a set of nested spatial windows to detect anomalies [21,22].

Despite the fact that many global and local AD algorithms have been proposed in the literature, most of them are time-consuming and difficult for hardware implementation due to the large amount of computing power needed for matrix and matrix inversion. However, in practical applications, there is an increasing interest in real-time processing for hyperspectral imagery. Several attributes may enhance the

need for real-time anomaly detection for hyperspectral imagery. One may be that finding some targets, especially moving targets, quickly is essential in decision-making. The other reason may be that, with the development of modern remote sensing technology, hyperspectral imagery can offer more detailed information; at the same time, it also has challenges of data storage, transmission, and system responding time due to the huge amount of hyperspectral data. With a real-time processing algorithm, we could reap the benefits of reducing data storage, which is important in data processing and getting pre-AD results in the course of data transmission. Despite the fact that there are many so-called real-time anomaly detection algorithms in the literature [23–27], to the authors' best knowledge, almost none of them are actual real-time procedures but simply fast algorithms. In order to address the real-time issue in hyperspectral anomaly detection, this paper presents pixel-by-pixel real-time processing algorithms for both global and local AD in hyperspectral imagery. Both synthetic and real hyperspectral imagery experiments are designed to show the real-time procedure and the difference between global and local real-time detectors.

This paper is organized as follows. Section 2 describes the methodologies of commonly used global and local anomaly detectors. Section 3 presents the real-time anomaly detection algorithms for both global and local forms. In Section 4 synthetic data and real hyperspectral imagery are conducted using all the AD algorithms above, including non-real-time and real-time versions, to prove the efficiency of the proposed methods. Finally, some conclusions are drawn in Section 5.

## 2. Commonly Used Global and Local Anomaly Detectors

The most widely used anomaly detector is probably the one developed by Reed and Xiaoli Yu in [4], referred to as K-RX detector (K-RXD), where  $\mathbf{K}$  is the global sample covariance matrix. Since then many versions of RX-type anomaly detectors have been proposed including a correlation-matrix-based RX detector and some local detectors using a sliding window to make anomaly detection adaptive [6,8,12–14].

### 2.1. Global RX Detector

Reed and Yu [4] developed, through a GLRT, a so-called RX anomaly detector for HSI data, assuming that the spectrum of the received signals (spectral pixels) and the covariance of the background clutter are unknown. It attempts to locate anything that looks different from the entire background clutter. Let each input spectral signal consisting of  $L$  spectral bands be denoted by  $\mathbf{x}_i = [x_{1i}, x_{2i}, \dots, x_{Li}]$ . The two competing hypotheses that the RX algorithm needs to distinguish are given by:

$$\begin{cases} \mathbf{H}_0 : \mathbf{x} = \mathbf{n} & \text{(target absent)} \\ \mathbf{H}_1 : \mathbf{x} = a\mathbf{s} + \mathbf{n} & \text{(target present)} \end{cases} \quad (1)$$

where  $a = 0$  under  $\mathbf{H}_0$  and  $a > 0$  under  $\mathbf{H}_1$ , respectively.  $\mathbf{n}$  is a vector that represents the background clutter noise process, and  $\mathbf{s}$  is the spectral signature of the signal (target) given by  $\mathbf{s} = [s_1, s_1, \dots, s_L]^T$ . The target signature and background covariance are assumed to be unknown. The model assumes that the data arise from two normal probability density functions with the same covariance matrix but different means. Under  $\mathbf{H}_0$  (background clutter), the data is modeled as  $N(\mathbf{0}, \mathbf{C}_b)$ , and under  $\mathbf{H}_1$  the data is modeled as  $N(\mathbf{s}, \mathbf{C}_b)$ .

Assuming a single pixel target as the observation test vector  $\mathbf{r}$ , the RX detector (RXD), referred to as  $\delta^{\text{RXD}}(\mathbf{r})$ , is specified by:

$$\delta^{\text{RXD}}(\mathbf{r}) = (\mathbf{r} - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{r} - \boldsymbol{\mu}) \begin{matrix} > \\ \leq \end{matrix} \eta \quad (2)$$

where  $\boldsymbol{\mu}$  is the estimated background sample mean,  $\mathbf{K}$  is the background covariance matrix estimated from the background pixels, and  $\eta$  is the threshold. It should be pointed out that the model assumes the data arise from two normal probability density functions with the same covariance matrix but different means.

The form of  $\delta^{\text{RXD}}(\mathbf{r})$  in Equation (2) is actually the well-known Mahalanobis distance. Since  $\mathbf{K}$  is a non-negative definite matrix it can be expressed as  $\mathbf{K} = \mathbf{K}^{1/2} \mathbf{K}^{1/2}$ . Using  $\mathbf{K}^{1/2}$  as a transform matrix, Equation (2) can be simply reduced to:

$$\begin{aligned} \delta^{\text{RXD}}(\mathbf{r}) &= (\mathbf{r} - \boldsymbol{\mu})^T \mathbf{K}^{-1/2} \mathbf{K}^{-1/2} (\mathbf{r} - \boldsymbol{\mu}) \\ &= (\mathbf{K}^{-1/2} \mathbf{r} - \mathbf{K}^{-1/2} \boldsymbol{\mu})^T (\mathbf{K}^{-1/2} \mathbf{r} - \mathbf{K}^{-1/2} \boldsymbol{\mu}) \\ &= \tilde{\mathbf{r}}^T \tilde{\mathbf{r}} = \|\tilde{\mathbf{r}}\|^2 \end{aligned} \quad (3)$$

with  $\tilde{\mathbf{r}} = \mathbf{K}^{-1/2} (\mathbf{r} - \boldsymbol{\mu})$  Equation (3) shows that RXD actually calculates  $\|\tilde{\mathbf{r}}\|^2$ , the square of the vector length of  $\tilde{\mathbf{r}}$ , which represents the gray-level intensity of  $\tilde{\mathbf{r}}$ . So, from a detection point of view,  $\mathbf{K}^{-1/2}$  can be interpreted as a whitening matrix. However, from a signal processing point of view, using  $\mathbf{K}$  and  $\boldsymbol{\mu}$  to remove the first two order statistics is called data sphering. Since the image background can be generally described by the 2nd order statistics, RXD performs anomaly detection by finding the higher intensities of sphered data sample vectors. In other words, anomaly detection is enhanced by anomaly contrast resulting from removing image background via data sphering in Equation (3) so as to achieve background suppression.

Chang replaces global covariance in Equation (2) with a sample correlation matrix, referred to as R-RX detector (R-RXD) [18], specified as follows:

$$\delta^{\text{R-RXD}}(\mathbf{r}) = \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} \quad (4)$$

where  $\mathbf{R}$  is the global sample data autocorrelation matrix formed by  $\mathbf{R} = (1/N) \sum_{i=1}^N r_i r_i^T$ .

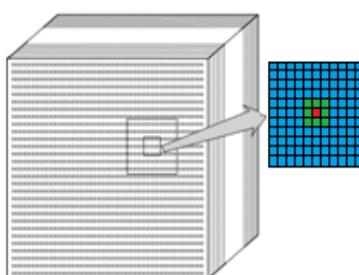
The form of RXD in Equation (1) may not explain quite how an anomaly detector performs by using its detected intensity and contrast. However, by virtue of Equation (4), the concept of intensity and contrast in anomaly detection algorithms can be explained well by R-RXD. First of all, the form of Equation (4) can be decomposed into two components,  $\mathbf{R}^{-1} \mathbf{r}$  and  $\mathbf{r}^T$ . The first component,  $\mathbf{R}^{-1} \mathbf{r}$ , carried out by R-RXD, actually performs background suppression via the use of global sample correlation matrix to increase the contrast of anomalies against the entire image background. This is followed by the second component,  $\mathbf{r}^T$ , which uses a matched filter to detect the intensity of anomalies via matching the background-compressed data sample  $\mathbf{R}^{-1} \mathbf{r}$ , using its own signature  $\mathbf{r}$  as a matched signal source vector. Since such a matched filter takes an inner product of the incoming signal source vector  $\mathbf{r}$  with the matched signal source vector  $\mathbf{R}^{-1} \mathbf{r}$ , it actually performs the spectral angle mapping (SAM) by calculating the angle between  $\mathbf{R}^{-1} \mathbf{r}$  and  $\mathbf{r}^T$ .

Following the same argument, Equation (1) can also be interpreted in a similar way, except that the data sample  $\mathbf{r}$  is now replaced by  $\mathbf{r} - \boldsymbol{\mu}$  and the global correlation matrix  $\mathbf{R}$  is replaced by the covariance matrix  $\mathbf{K}$ . However, it should be noted that this is a significant difference between R-RXD and original covariance-based RXD because the former detects the data sample vector itself while the latter actually detects the data sample variation from the global sample mean, *i.e.*, data sample vectors that have large

gradients. So, technically speaking, the original covariance-based RXD does not detect intensities of anomalies themselves but rather their gradient intensities. As a result, RXD works as a distance measure like Mahalanobis distance, while R-RXD can be considered as a matched filter.

### 2.2. Local RX Detector

Local anomaly detection is very important since the global RX anomaly detector fails to work when the anomalies are relatively small or only distinct from the local surroundings but buried in the global background. The most widely used local anomaly detection algorithm, derived from the commonly used RXD, is the local-RX detector (LRXD), which makes use of a local dual window to get the local background statistics sliding through the whole data set. The form of the dual window is shown in Figure 1. Sliding dual window for LRXD, which is the commonly used local sliding window for hyperspectral anomaly detection.



**Figure 1.** The sliding dual window for LRXD is the most commonly used for hyperspectral anomaly detection.

The LRXD is specified as follows:

$$\delta^{LRXD}(\mathbf{r}) = (\mathbf{r} - \boldsymbol{\mu}_{local})^T \mathbf{K}_{local}^{-1} (\mathbf{r} - \boldsymbol{\mu}_{local}) \tag{5}$$

where  $\boldsymbol{\mu}_{local}$  is the local background mean and  $\mathbf{K}_{local}$  is the local background covariance matrix.

This local anomaly detection returns a higher detection rate due to the fact that the local background model can be tightly fitted. However, this is accompanied by a proper window size, which is unknown, and the problem is there is no universal way to completely avoid over-fitting or under-fitting. Another problem is that this process is time-consuming. The LRXD detector is computationally more expensive than the original RXD, caused by a repeated calculation of a covariance/correlation matrix and its inverse in every local window as opposed to the global RXD, which performs only once for the entire data set.

### 3. Real-Time Anomaly Detectors

None of the global and local RX detectors mentioned above are actually real-time detectors since they need to use either the entire data vectors or the data vectors in the local window, which also include future pixels. Once the causal version specified by Equation (4) is proposed, a follow-up task is to implement it pixel-by-pixel in a real-time way so as to achieve real-time processing. The following Woodbury matrix identity [28,29] is very helpful:

$$\left[ \mathbf{A} + \mathbf{u}\mathbf{v}^T \right]^{-1} = \mathbf{A}^{-1} - \frac{\left[ \mathbf{A}^{-1}\mathbf{u} \right] \left[ \mathbf{v}^T \mathbf{A}^{-1} \right]}{1 + \mathbf{v}^T \mathbf{A}^{-1} \mathbf{u}} \tag{6}$$

### 3.1. Global Real-Time Detector

In order to implement the real-time RX detector, a causal version of the RX detector (CRXD) is firstly introduced here. Let  $\{\mathbf{r}_i\}_{i=1}^N$  ( $N$  is the total number of pixels of the entire data set) be a set of data sample vectors to be processed. The causal RX detector (RXD) [30], denoted by  $\delta^{\text{CRXD}}(\mathbf{r}_n)$ , is specified by:

$$\delta^{\text{CRXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \mathbf{R}(n)^{-1} \mathbf{r}_n \quad (7)$$

where  $\mathbf{r}_n$  is the  $n^{\text{th}}$  data sample vector currently being processed,  $\mathbf{R}(n)$  is the sample autocorrelation matrix formed by  $\mathbf{R}(n) = (1/n) \sum_{i=1}^n \mathbf{r}_i \mathbf{r}_i^T$ , and  $\mathbf{r}_i = (\mathbf{r}_{i1}, \mathbf{r}_{i2}, \dots, \mathbf{r}_{iL})^T$  is the  $i^{\text{th}}$  data sample vector (where  $L$  is the total number of spectral bands).

Since the causal sample autocorrelation matrix is defined by  $\mathbf{R}(n) = (1/n) \sum_{i=1}^n \mathbf{r}_i \mathbf{r}_i^T$ ,  $\mathbf{R}^{-1}(n)$  can be re-expressed as follows:

$$\begin{aligned} \mathbf{R}^{-1}(n) &= \left[ \frac{1}{n} \sum_{i=1}^n \mathbf{r}_i \mathbf{r}_i^T \right]^{-1} = n \cdot \left[ \sum_{i=1}^n \mathbf{r}_i \mathbf{r}_i^T \right]^{-1} \\ &= n \cdot \left[ (n-1) \frac{1}{n-1} \sum_{i=1}^{n-1} \mathbf{r}_i \mathbf{r}_i^T + \mathbf{r}_n \mathbf{r}_n^T \right]^{-1} \\ &= n \cdot \left[ (n-1) \mathbf{R}(n-1) + \mathbf{r}_n \mathbf{r}_n^T \right]^{-1} \end{aligned} \quad (8)$$

Now, by virtue of Equation (6) we can get a global real-time causal R-RXD (GRTCR-RXD), and the inverse of the correlation matrix can be obtained by a causal innovations information update equation where the  $\mathbf{R}^{-1}(n)$  can be easily updated by  $\mathbf{R}^{-1}(n-1)$  as well as the current input data sample vector  $\mathbf{r}_n$ . If we follow Equation (8) and use Equation (6) by letting  $\mathbf{A} = (n-1)\mathbf{R}(n-1)$  and  $\mathbf{u} = \mathbf{v} = \mathbf{r}_n$ , then we will have Equation (9):

$$\begin{aligned} \mathbf{R}^{-1}(n) &= n \cdot \left[ (n-1) \mathbf{R}(n-1) + \mathbf{r}_n \mathbf{r}_n^T \right]^{-1} \\ &= n \cdot \left\{ \frac{1}{n-1} \mathbf{R}(n-1)^{-1} - \frac{\left[ \frac{1}{(n-1)} \mathbf{R}(n-1)^{-1} \mathbf{r}_n \right] \left[ \frac{1}{(n-1)} \mathbf{r}_n^T \mathbf{R}(n-1)^{-1} \right]}{1 + \frac{1}{(n-1)} \mathbf{r}_n^T \mathbf{R}(n-1)^{-1} \mathbf{r}_n} \right\} \\ &= \frac{n}{n-1} \mathbf{R}(n-1)^{-1} - \frac{n \mathbf{R}(n-1)^{-1} \mathbf{r}_n \mathbf{r}_n^T \mathbf{R}(n-1)^{-1}}{(n-1)^2 + (n-1) \mathbf{r}_n^T \mathbf{R}(n-1)^{-1} \mathbf{r}_n} \end{aligned} \quad (9)$$

Then  $\mathbf{R}^{-1}(n)$  can be expressed as Equation (9) and the final GRTCRXD is specified in Equation (10), where  $\tilde{\delta} = \mathbf{r}_n^T \mathbf{R}(n-1)^{-1} \mathbf{r}_n$  and  $\mathbf{M} = \mathbf{r}_n^T \mathbf{R}(n-1)^{-1}$ :

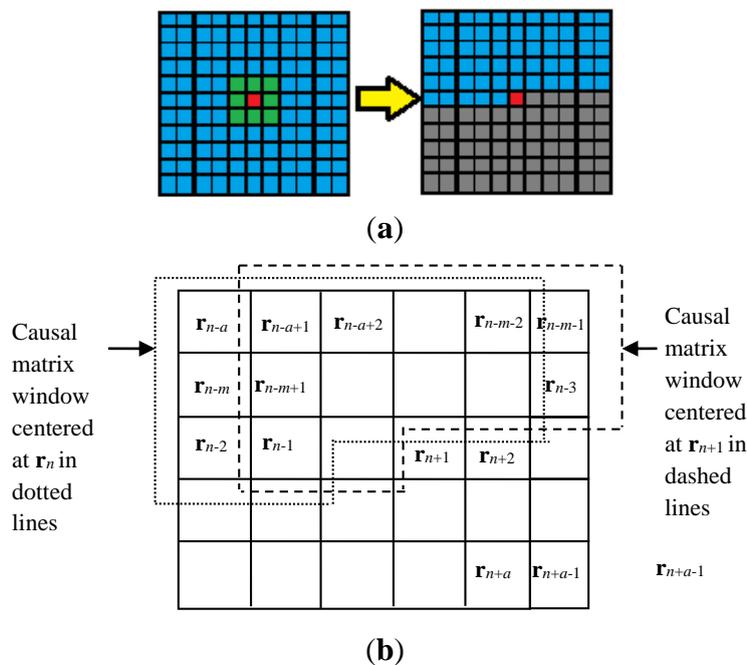
$$\delta^{\text{GRTCRXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \mathbf{R}^{-1}(n) \mathbf{r}_n = \frac{n}{n-1} \tilde{\delta} - \frac{n \cdot \tilde{\delta} \mathbf{M}}{(n-1)^2 + (n-1) \tilde{\delta}} \quad (10)$$

### 3.2. Local Real-Time Detector

The local anomaly detector is of particular importance, as we have explained in Section 2. However, there are still some weaknesses, of which the most severe is that all the local AD algorithms are time-consuming and computationally more expensive than the original global RX, caused by a repeated calculation of a covariance/correlation matrix and its inversion in every local window. Most real-time detectors are implemented in a global view; there is no real-time anomaly detection algorithm in the form of a local view. Local real-time detection is more difficult to implement than the global version. One of the reasons

may be that the sliding window in the commonly used local AD algorithms still needs future pixels after the pixel being processed, as shown in Figure 1. If we want to let the local anomaly detector to work in to a truly real-time manner, we must design a causal sliding window in which background pixels only include the pixels before, that is to say, no further pixels should be included in the local causal window. It is much more difficult to get the inversion of a correlation matrix in a recursive manner since more than one pixel in the local window will change each time. The correlation matrix  $\mathbf{R}(n)$  is no longer an easy addition function of  $\mathbf{R}(n - 1)$  but must include the new information  $\mathbf{r}_n$  we derived in Equation (9) for the global real-time detector.

In order to solve the first problem, a Causal Matrix (CM) local window is firstly presented, as shown in Figure 2.



**Figure 2.** (a) Causal matrix local window derived from the traditional local window; (b) Causal matrix local windows at  $\mathbf{r}_n$  and  $\mathbf{r}_{n+1}$ .

This CM local window comes from the view of the traditional local window and the concept of “causal” characteristics, *i.e.*, that the detector could only use those pixels prior to the pixel currently being processed; no further information could be included. It is a little bit different from the traditional local window, which is formed by an inner window and an outer window centered by the pixel to be processed. The pixels in blue are background information in Figure 2a, removing all the further pixels in gray color. Figure 2b shows how the causal matrix window slides and how information changes caused by in-and-out pixels. When the window slides from  $\mathbf{r}_n$  (dotted line window) to  $\mathbf{r}_{n+1}$  (dashed line window), all the far left column pixels in the dotted line window are going out and all the far right column pixels in the dashed line window need to be added.

The local real-time detector based on CM window is specified as follows:

$$\delta^{\text{LRTC MRXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \tilde{\mathbf{R}}(n)^{-1} \mathbf{r}_n \tag{11}$$

Technically speaking, Equation (11) can be implemented in real time. However, with a complex, real-time AD algorithm, using the CM window is no longer as easy as the global real-time detector, in which only one recursive equation is able to update  $\mathbf{R}^{-1}(n)$  for detection. As we can see from Figure 2b, all data sample vectors excluded from the causal matrix window are not removed in sequence. For example, in Figure 2b the  $\mathbf{r}_{n-a}$ ,  $\mathbf{r}_{n-m}$ , and  $\mathbf{r}_{n-2}$  in the causal matrix window centered at  $\mathbf{r}_n$  are removed from the causal matrix window centered at  $\mathbf{r}_{n+1}$ , while  $\mathbf{r}_{n-m-1}$  and  $\mathbf{r}_{n-3}$ , which are not included in the causal matrix window centered at  $\mathbf{r}_n$ , are now added to the causal matrix window centered at  $\mathbf{r}_{n+1}$ . Obviously, it requires a bookmark to keep track of which data sample vectors should be removed and which data sample vectors should be added as a causal matrix window moves on. Otherwise, the CM window may need to follow the same method as LRXD—loading and storing all pixels in the local window, then calculating  $\mathbf{R}(n)$  and  $\mathbf{R}^{-1}(n)$  repeatedly while the window slides. This processing does not meet the real-time requirement since it is time-consuming, even though the processing procedure has already been causally identified.

In order to resolve this issue, we can stretch out the causal matrix window in Figure 2b as a linear array shown in Figure 3. By virtue of Figure 3a, we define a causal array window of width  $w$  sliding along with a processed data sample vector  $\mathbf{r}_n$  as a linear array  $\{\mathbf{r}_i\}_{i=n-w}^{n-1}$ , shown in Figure 3b, named a causal array (CA) local window.

Thus, a CA window moves along with the process of the image data, it simply performs like a queue, first in and first out. In fact, the CA local window is no longer the traditional matrix window of a square window, as shown in Figure 1, but a linear array window consisting of  $w$  data sample vectors preceding the current processed data sample  $\mathbf{r}_n$ . The In order to make it clear how the CA local window works, Figure 3c shows the CA window at  $\mathbf{r}_n$  (depicted by dotted lines) and at  $\mathbf{r}_{n+1}$  (depicted by dashed lines), where the farthest data sample vector  $\mathbf{r}_{n-w}$  from  $\mathbf{r}_n$  in the causal array window at  $\mathbf{r}_n$  is removed from the causal array window at  $\mathbf{r}_{n+1}$ , while the most recent data sample vector  $\mathbf{r}_n$  is then added to the causal array window at  $\mathbf{r}_{n+1}$ .

In order to resolve the processing time issue, a recursive causal information update equation is derived using the same methodology as the global real-time AD algorithm, the Woodbury matrix identity. However, there is a small difference, in that the local real-time AD algorithm needs to apply the Woodbury matrix identity twice in order to get the update equation. The detector formula is the same as LRTCMRXD, marked as LRTCARXD:

$$\delta^{\text{LRTCARXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \tilde{\mathbf{R}}(n)^{-1} \mathbf{r}_n \tag{12}$$

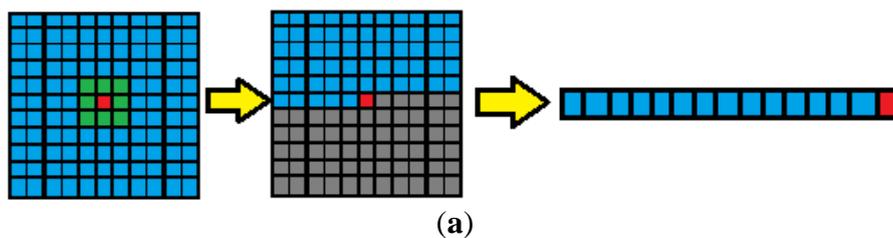
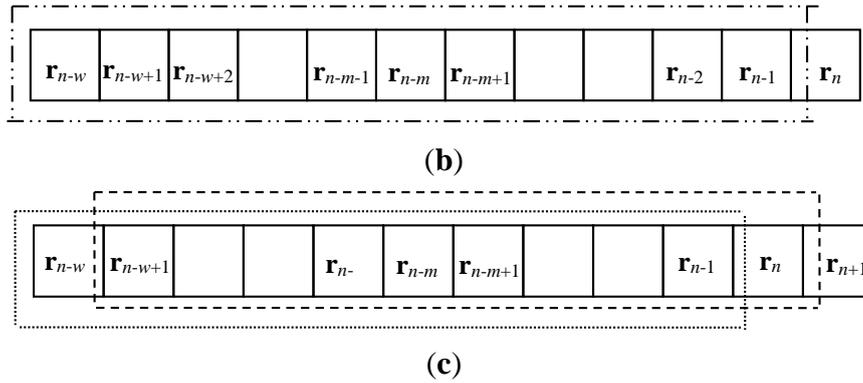


Figure 3. Cont.



**Figure 3.** (a) Three different local windows; (b) causal array window of width  $w$ ; (c) causal array local windows at  $\mathbf{r}_n$  and  $\mathbf{r}_{n+1}$ .

Assume that the width of a sliding causal array window is  $w$  and the data sample vector to be processed is  $\mathbf{r}_n$ . To emphasize the width of  $w$  and the processed data sample vector  $\mathbf{r}_n$ , we rewrite  $\tilde{\mathbf{R}}(n)$  in Equation (12) as  $\tilde{\mathbf{R}}_w(n)$ . Then according to Figure 3c,  $\tilde{\mathbf{R}}_w(n + 1)$  can be further expressed as:

$$\begin{aligned} \tilde{\mathbf{R}}_w(n + 1) &= \frac{1}{w} \sum_{i \in \text{current window}} \mathbf{r}_i \mathbf{r}_i^T = \frac{1}{w} \left[ \sum_{i \in \text{previous window}} \mathbf{r}_i \mathbf{r}_i^T - \mathbf{r}_{n-w} \mathbf{r}_{n-w}^T + \mathbf{r}_n \mathbf{r}_n^T \right] \\ &= \tilde{\mathbf{R}}_w(n) - \frac{1}{w} \mathbf{r}_{n-w} \mathbf{r}_{n-w}^T + \frac{1}{w} \mathbf{r}_n \mathbf{r}_n^T \end{aligned} \tag{13}$$

Assuming the first two parts, as a whole, equate to

$$\tilde{\tilde{\mathbf{R}}}_w(n) = \tilde{\mathbf{R}}_w(n) - \frac{1}{w} \mathbf{r}_{n-w} \mathbf{r}_{n-w}^T \tag{14}$$

Then, according to Equation (13) and (14), we would have

$$\tilde{\mathbf{R}}_w(n + 1) = \tilde{\tilde{\mathbf{R}}}_w(n) + \frac{1}{w} \mathbf{r}_n \mathbf{r}_n^T \tag{15}$$

Then we apply the first use of the Woodbury identity in Equation (6) by letting  $\mathbf{A} = \tilde{\tilde{\mathbf{R}}}_w(n)$  and  $\mathbf{u} = \mathbf{v} = \sqrt{1/w} \cdot \mathbf{r}_n$ . Thus  $\tilde{\mathbf{R}}_w^{-1}(n + 1)$  can be expressed as Equation (16):

$$\begin{aligned} \tilde{\mathbf{R}}_w^{-1}(n + 1) &= \left[ \tilde{\tilde{\mathbf{R}}}_w(n) + \frac{1}{w} \mathbf{r}_n \mathbf{r}_n^T \right]^{-1} \\ &= \tilde{\tilde{\mathbf{R}}}_w^{-1}(n) - \frac{(1/w) \tilde{\tilde{\mathbf{R}}}_w^{-1}(n) \mathbf{r}_n \mathbf{r}_n^T \tilde{\tilde{\mathbf{R}}}_w^{-1}(n)}{1 + (1/w) \mathbf{r}_n^T \tilde{\tilde{\mathbf{R}}}_w^{-1}(n) \mathbf{r}_n} \end{aligned} \tag{16}$$

Now, in order to calculate the inverse of  $\tilde{\mathbf{R}}_w(n + 1)$ , i.e.,  $\tilde{\mathbf{R}}_w^{-1}(n + 1)$ , we need to get the innovation information  $\tilde{\tilde{\mathbf{R}}}_w^{-1}(n)$  first, where the second use of the Woodbury matrix identity is applied. The Woodbury matrix identity used here could be specified as:

$$\left[ \mathbf{A} - \mathbf{u} \mathbf{v}^T \right]^{-1} = \mathbf{A}^{-1} + \frac{\left[ \mathbf{A}^{-1} \mathbf{u} \right] \left[ \mathbf{v}^T \mathbf{A}^{-1} \right]}{1 - \mathbf{v}^T \mathbf{A}^{-1} \mathbf{u}} \tag{17}$$

Referring to Equations (14) and (17), we could get  $\tilde{\mathbf{R}}_w^{-1}(n)$  by letting  $\mathbf{A} = \tilde{\mathbf{R}}_w(n)$  and  $\mathbf{u} = \mathbf{v} = \sqrt{1/w} \cdot \mathbf{r}_{n-w}$ , shown as follows:

$$\begin{aligned} \tilde{\mathbf{R}}_w^{-1}(n) &= \left[ \tilde{\mathbf{R}}_w(n) - \frac{1}{w} \mathbf{r}_{n-w} \mathbf{r}_{n-w}^T \right]^{-1} \\ &= \tilde{\mathbf{R}}_w^{-1}(n) + \frac{(1/w) \tilde{\mathbf{R}}_w^{-1}(n) \mathbf{r}_{n-w} \mathbf{r}_{n-w}^T \tilde{\mathbf{R}}_w^{-1}(n)}{1 - (1/w) \mathbf{r}_{n-w}^T \tilde{\mathbf{R}}_w^{-1}(n) \mathbf{r}_{n-w}} \end{aligned} \tag{18}$$

By virtue of Equation (16) and (18),  $\tilde{\mathbf{R}}_w(n + 1)$  can be updated recursively by  $\tilde{\mathbf{R}}_w(n)$  via deleting the information  $\mathbf{r}_{n-w}$  and adding the new information  $\mathbf{r}_n$ , so as to realize the real-time procedure of the local anomaly detection in Equation (12).

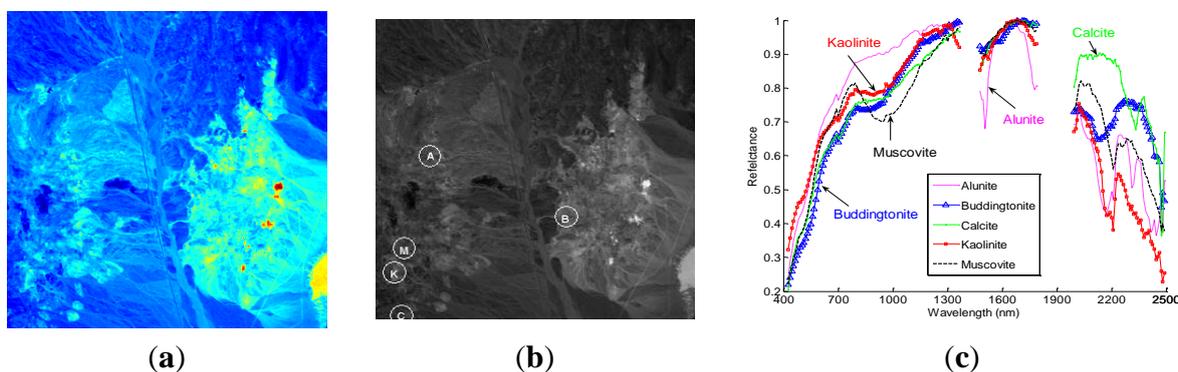
### 4. Experiments

In this section, three sets of hyperspectral data, including two synthetic images and one real image, are used for global and local real-time implementations.

#### 4.1. Data Set Descriptions

In order to verify the performance of proposed real-time detectors, both synthetic and real hyperspectral datasets are used to evaluate our methods.

A synthetic hyperspectral data is custom designed with real ground truth for analysis. The synthetic images are simulated by a real Cuprite image scene with 189 bands, as shown in Figure 4a, which is available at the USGS website [31]. The ground truth available for this region provides the pixel locations of five minerals: alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M), shown in Figure 4b. The spectral features of the five pure minerals are shown in Figure 4c.



**Figure 4.** (a) Cuprite AVIRIS image scene; (b) spatial positions of A, B, C, K, and M; (c) spectra of the five signatures extracted from the scene in (b).

The synthetic images are designed especially in terms of local detectors, compensating for the weakness of global AD algorithms when the targets are extremely small or only abnormal in a local background but buried in the global background. The five mineral spectral signatures—A, B, C, K, M, marked by circles in Figure 4b—are used to simulate 25 panels with different sizes in Figure 5a. Figure 5b gives the 100th band of the synthetic hyperspectral image.

Another type of hyperspectral image data to be used for experiments is a real AVIRIS image scene, shown in Figure 6, which is the Lunar Crater Volcanic Field (LCVF) with the parameters shown in Table 1.

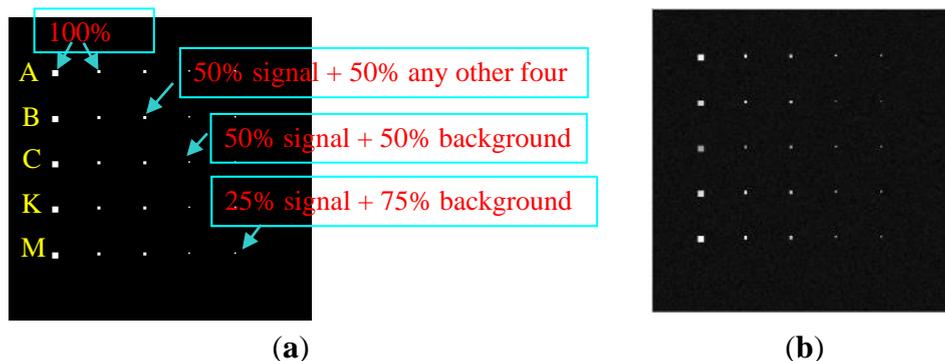


Figure 5. (a) A set of 25 simulated panels; (b) 100th band of synthetic hyperspectral image.

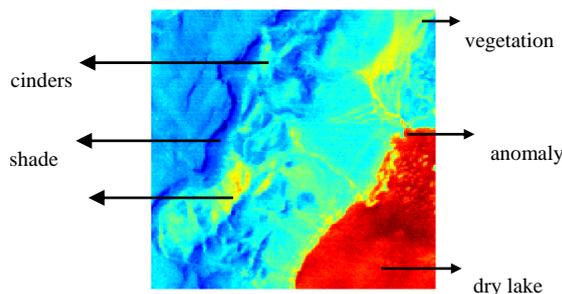


Figure 6. AVIRIS LCVF subscene.

Table 1. Parameters of AVIRIS LCVF.

Sensor	AVIRIS
Wavelength	400–1800 nm
Bands	158
Spectral resolution	10 nm
Spatial resolution	20 m
Image size	200 × 200
Gray range	0–10,000
Location	Northern Nye County, NV

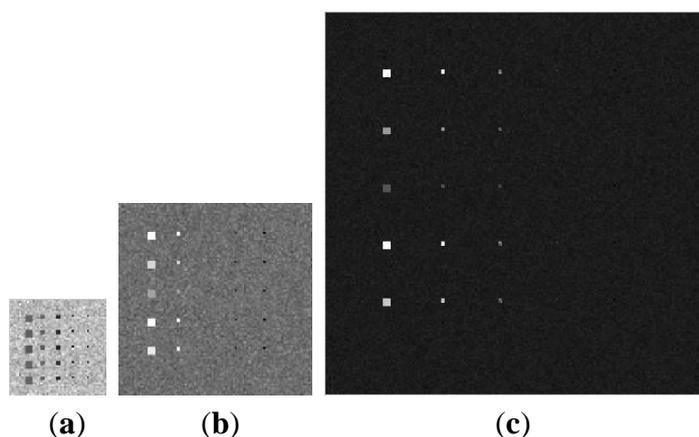
4.2. Global Real-Time Processing Experiments

Several issues arising in anomaly detection have been discussed. One question of particular interest is “what is an anomaly?” In other words, what types of targets are considered as anomalies? A first attempt to address this issue was made in [32], which concluded that an anomaly is closely related to its size relative to the image to be processed.

In this section, before conducting the global real-time processing experiments, the following simple example provides a clue to how controversial this issue is. Figure 7a–c shows the same set of target panels given above with image sizes of 50 × 50 pixel vectors, 100 × 100 pixel vectors and 200 × 200 pixel vectors, respectively. All of the targets are clearly detected in Figure 7a with an image size of 50 × 50; however, the 2 × 2 mixed-pixel panels in the third column are missing in the 100 × 100 image in Figure 7b and all

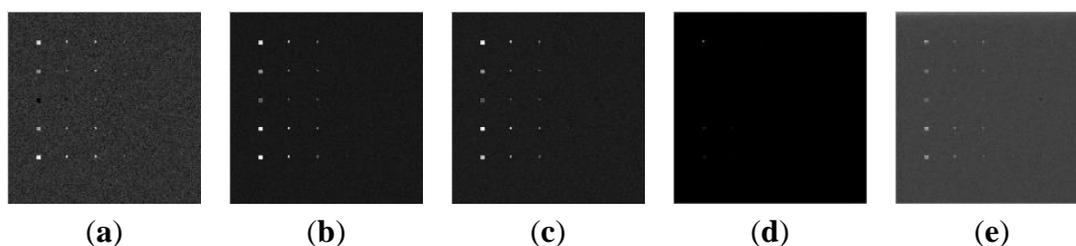
the  $1 \times 1$  sub-pixel panels are missing from the  $200 \times 200$  image in Figure 7c. The experimental results give further evidence that an anomaly is closely related in size to the image to be processed.

Several experiments are then implemented on both synthetic and real hyperspectral images to verify the effectiveness of the global real-time processing. The 100th band of the synthetic data is shown in Figure 8a. We firstly implement experiments to illustrate two commonly used K-RXD and R-RXD using global sample covariance/correlation matrix K/R and the GRTCRXD presented in Section 3.1. Figure 8b–d shows the grayscale results of global K-RXD, RRXD and GRTCRXD, respectively. By visual inspection of Figure 8b,c, there is no appreciable difference. Three-dimensional plots are used to verify the detailed differences between K-RXD and R-RXD. As we can see from the 3D plots, both versions (K-RXD and R-RXD) of global anomaly detectors performed comparably, except that there were different degrees of background suppression resulting from the use of covariance and correlation matrices.

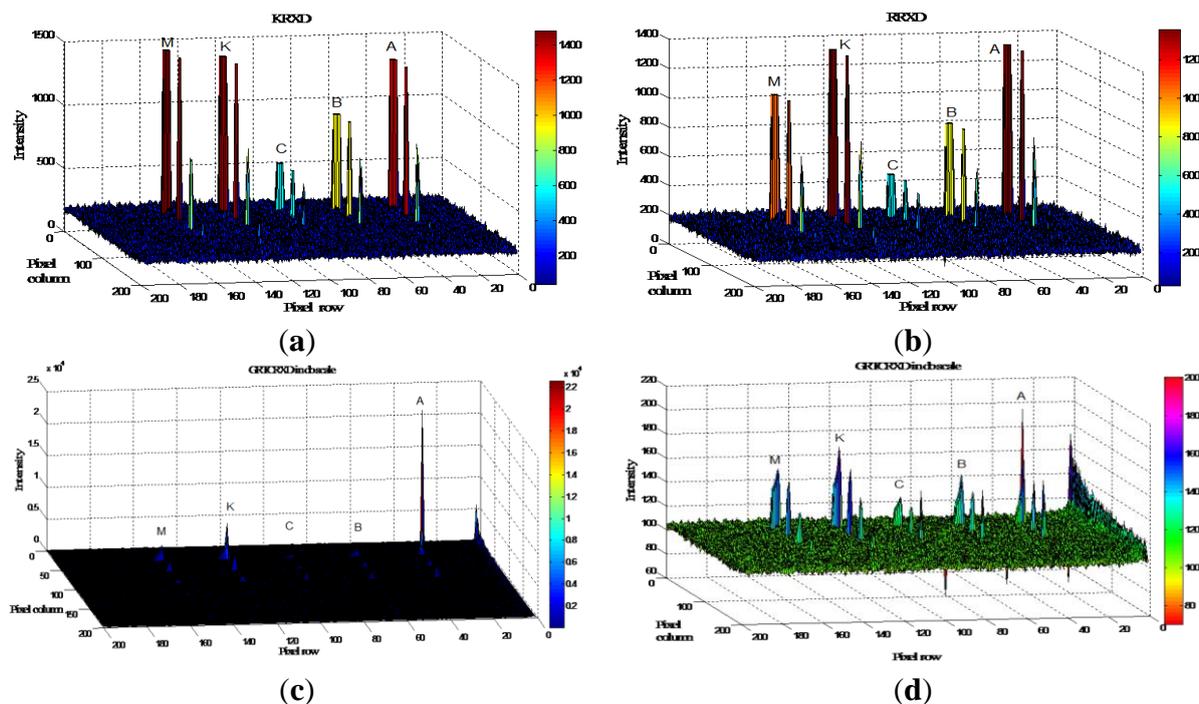


**Figure 7.** Detection results of the same set of target panels with different image sizes: (a) image size of  $50 \times 50$ ; (b) image size of  $100 \times 100$ ; (c) image size of  $200 \times 200$ .

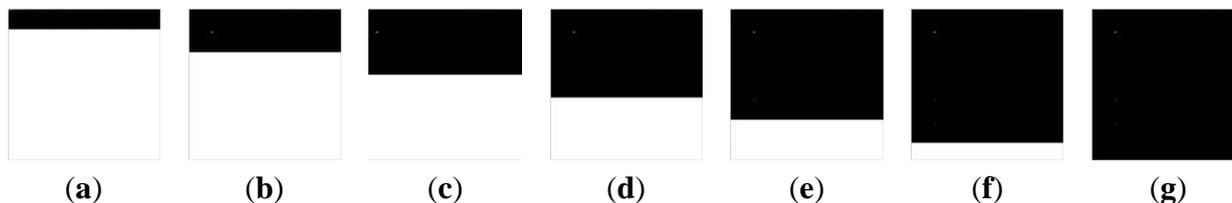
In Figure 8d, which shows the grayscale results of GRTCRXD, only the first anomaly pixel came out when the results are shown in grayscale. When we convert into db space, the other anomalies came out. That is because the intensity of the first anomaly pixel is very high. In Figure 9, four different 3D results are given for comparison. In 3D plots, the intensity of anomalies are shown clearly, especially in Figure 9c,d, where the 3D plots verified our hypothesis that other anomalies are embedded in the background because of the high intensity of the first anomaly. Figures 10 and 11 are progressive procedure of GRTCRXD in either gray scale or db scale. Background suppression changes with the processing.



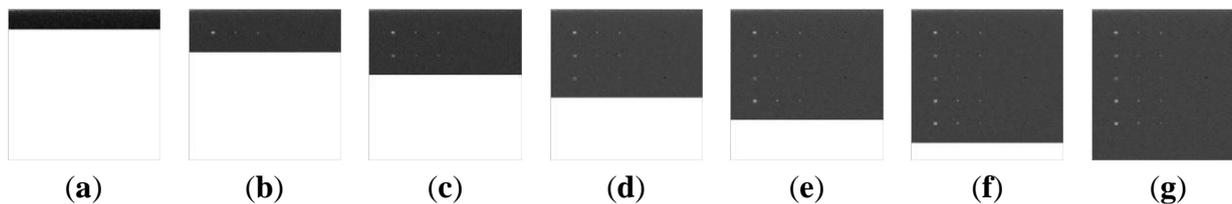
**Figure 8.** Detection results using global RX detector and its real-time version: (a) 100th band of original data; (b) grayscale result of K-RXD; (c) grayscale result of R-RXD; (d) grayscale result of GRTCRXD; (e) db scale result of GRTCRXD.



**Figure 9.** (a) 3D plot of global K-RXD; (b) 3D plot of global R-RXD; (c) 3D plot of original GRTCRXD; (d) 3D plot of GRTCRXD in db scale.

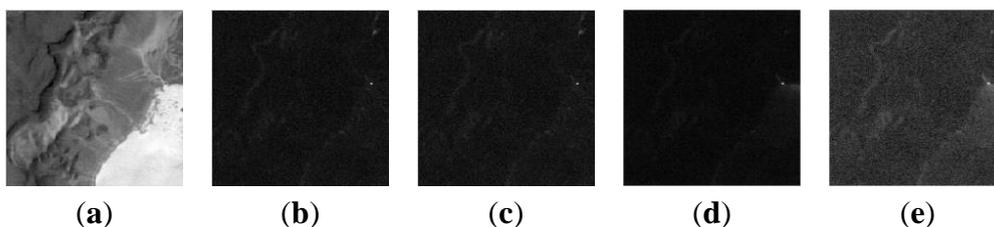


**Figure 10.** Grayscale detection results of GRTCRXD for TI: (a) no panels detected; (b) row 1 panels detected; (c) row 2 panels detected; (d) row 3 panels detected; (e) row 4 panels detected; (f) row 5 panels detected; (g) final detection result.

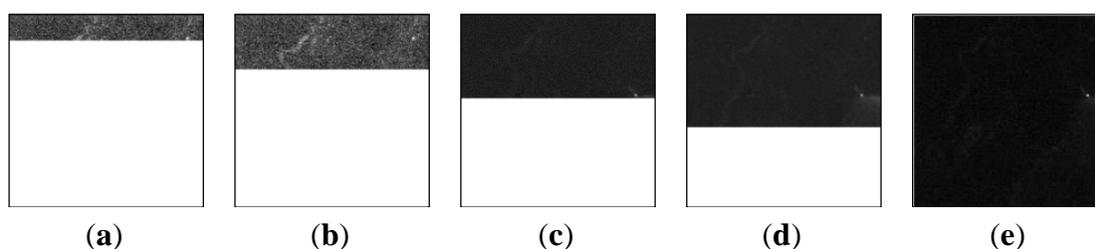


**Figure 11** Progressive procedures of GRTCRXD in either grayscale or db scale. Background suppression changes with the processing.

The same experiments are conducted on a real hyperspectral LCVF image. Figure 12 gives the detection results for a real LCVF hyperspectral image using K-RXD and R-RXD, together with global real-time causal RXD in both original grayscale and db scale. The progressive procedure of GRTCRXD for LCVF data is given in Figure 13, which will clearly show the background suppression issue along with the processing procedure.



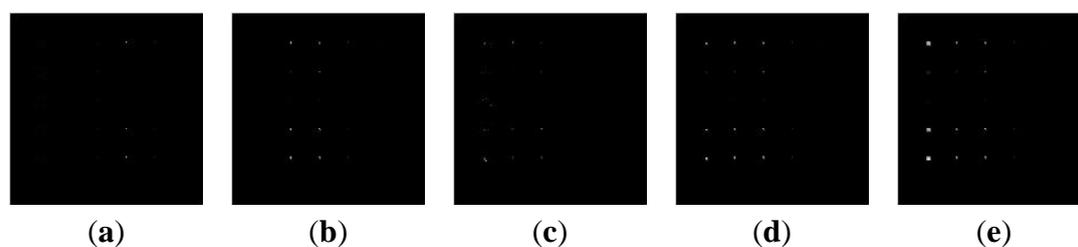
**Figure 12.** Detection results of real hyperspectral LCVF image using global RX detector and its real-time version: (a) 100th band of LCVF image; (b) grayscale result of K-RXD; (c) grayscale result of R-RXD; (d) grayscale result of GRPCRXD; (e) db scale result of GRPCRXD.



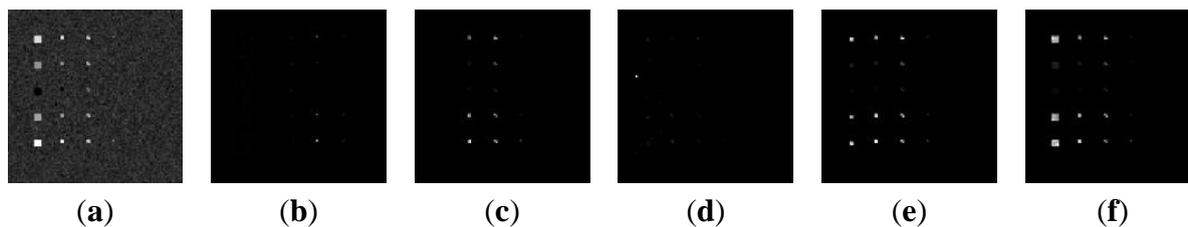
**Figure 13.** GRPCRXD detection results for LCVF: (a) vegetation appears; (b) vegetation and cinder detected; (c) anomaly just appears; (d) anomaly detected; (e) final detection result.

### 4.3. Local Real-Time Processing Experiments

Another question of interest derives from local anomaly detection: “what is the performance impact of the local window size on anomaly detection?” As in the previous section, before conducting the local real-time processing experiments, several experiments using TI images were conducted to explore this question. Figure 14 shows the LRXD results using different local window size with image size. When the inner window size equals 1, only the pixels in the 4th column could be detected, as shown in Figure 14a. When the inner window size increases to 3, the targets in the 2nd and 3rd columns could also be detected, as shown in Figure 14b. However, when the window size equals 5 or 7, the 1st column targets appeared on the grayscale results, as shown in Figure 14c–e. Comparing Figure 14a–e, it is clearly proved that the inner window size depends on the size of anomaly targets and the outer window size is much larger than the inner window size. The detection results get better from Figure 14a through Figure 14e. Figure 15 repeats the same experiments with image size. Comparing the corresponding sub-figures of Figures 14 and 15, we could know that the image size will also affect the detection results.

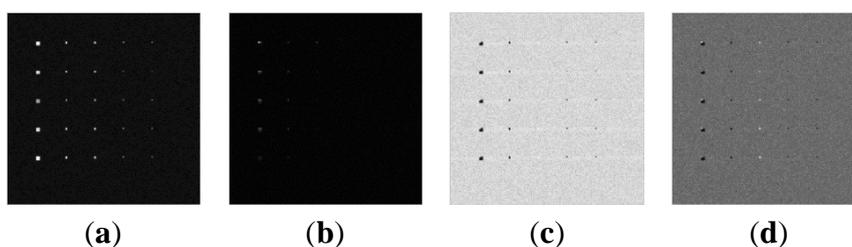


**Figure 14.** LRXD results using different local window size with image size  $200 \times 200$ : (a) window size 1/15 (inner window size 1 and outer window size 15); (b) window size 3/15; (c) window size 5/15; (d) window size 5/17; (e) window size 7/17.



**Figure 15.** LRXD results using different local window size with image size  $100 \times 100$ : (a) window size 1/15 (inner/outer window); (b) size 3/15; (c) size 5/15; (d) size 5/17; (e) size 7/17.

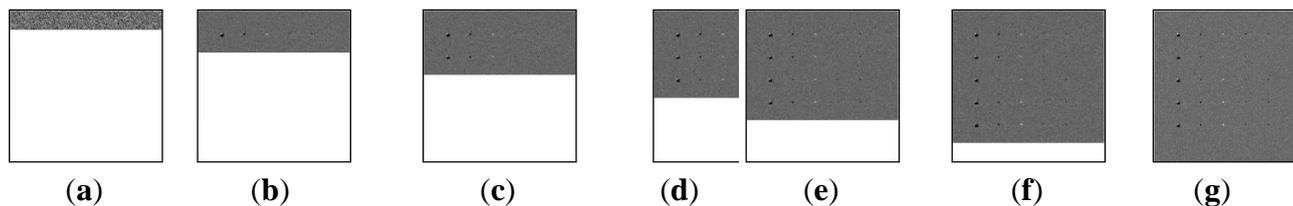
The following experiments are conducted to verify the effectiveness of the local real-time detector using synthetic HSI data. Local AD concept is very important, as we said at the beginning, especially when global AD algorithms fail to detect targets that are extremely small or only abnormal in a local background but buried in the global background. Figure 16 is a very good explanation for this pattern. Figure 16a is generated using the same way Figure 5a does, but changes the order of A, B, C, K, M to K, A, M, B, C. Since material C is the nearest matching with the background signature (shown in Table 2), running with global real-time processing in Figure 16b, anomalies with C signature in the 5th row is hardly seen in the scene. However, when local real-time procedure is conducted to the same image (with causal array window size  $15 \times 15$ ), all the anomalies could be well defined except for the  $2 \times 2$ -mixed pixel panels in the third column, much better than the global real-time detector. Figure 16d is the same detector with causal array window size  $21 \times 21$ , in which the  $2 \times 2$ -mixed pixel panels in the third column could be detected. Figure 17(a–g) are steps of local real-time causal array RXD detection results.



**Figure 16.** Local real-time anomaly detection experiments: (a) synthetic data scene; (b) global real-time causal RXD anomaly detection results; (c) local real-time causal array RXD anomaly detection results with array window width  $15 \times 15$ , (d) local real-time causal array RXD anomaly detection results with array window width  $21 \times 21$ .

**Table 2.** Comparative analysis of spectral similarity between background and five target signatures.

Measures	Alunite (A)	Buddingtonite (B)	Calcite (C)	Kaolinite (K)	Muscovite (M)
SAM	0.1821	0.0656	0.0441	0.1923	0.0924
SID	0.0405	0.0046	0.0025	0.0522	0.0103



**Figure 17.** Gray scale detection results of local real-time causal array RXD: (a) no panels detected; (b) row 1 panels detected; (c) row 2 panels detected; (d) row 3 panels detected; (e) row 4 panels detected; (f) row 5 panels detected; (g) final detection result.

#### 4.4. Computational Analysis

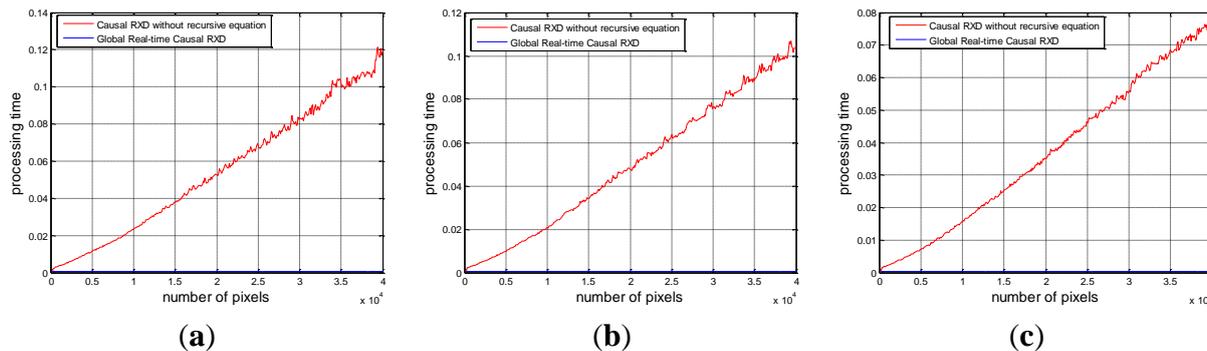
Computational complexity is also an important topic for HSI anomaly detection algorithms. The computational complexity of RX-based algorithms stems from the inversion of the covariance matrix that is required for the background estimation, which in general is of order  $O(L^3)$ , where  $L$  is the number of bands according to Table 2. For global anomaly detection algorithms, due to the requirement of real-time processing, the global background information (covariance/correlation matrix and its inversion) is re-calculated for each pixel. For local anomaly detection algorithms, on the other hand, this inversion is also required at every pixel due to the local nature of background estimation in a sliding local window at each spatial location in the hypercube. In this case, without using our new methodology, the total processing is of order  $O(N \cdot L^3)$ , where  $N$  is the total number of spatial pixels in the HSI scene. This results in a high computational cost, which could be reduced by using the Woodbury matrix identity in our real-time detectors.

The computer environments used for experiments are 64-bit operating systems with Intel(R) Core (TM) i7-4770K, 3.5 GHz CPU, and 16 GB memory (RAM). All the experiments were conducted five times and averaged to remove computer error. The average computing times (CPT) for both synthetic and real hyperspectral images are given in Table 3 using all detectors including global detectors (K-RXD, R-RXD, CRXD, GRTCRXD) and local detectors (LRXD, LRTCARXD). From Table 3 we could see that, using our new recursive equations, both global and local processing times were much reduced, especially the GRTCRXD, which was reduced by 100-fold compared with CRXD without recursive equation.

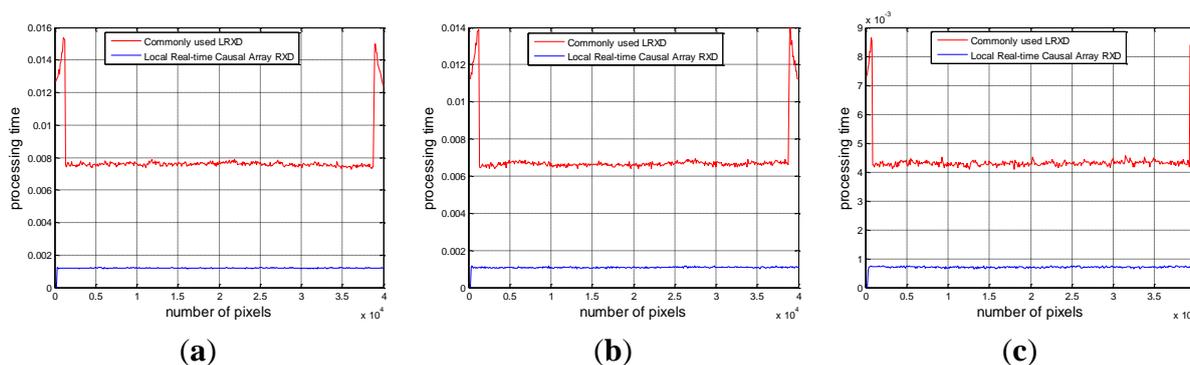
**Table 3.** Total processing time for TI, TE, and LCVF.

	<b>K-RXD</b> (seconds)	<b>R-RXD</b> (seconds)	<b>CRXD</b> (seconds)	<b>GRTCRXD</b> (seconds)	<b>LRXD</b> (seconds)	<b>LRTCARXD</b> (seconds)
Scenario TI	1.7067	1.1825	2178.97	24.1223	332.8399	47.8953
Scenario TE	1.4789	1.1045	1984.92	22.4953	294.2294	43.1142
AVIRIS LCVF	1.2293	0.7925	1471.51	14.9151	183.2476	28.1530

In order to further evaluate computational complexity, Figures 18 and 19 give the averaged CPT for each pixel using both global and local detectors, where the x-axis and y-axis represent the order of pixels being processed and the CPT required for the current pixel, respectively.



**Figure 18.** CPT (Computing time) for global detectors. (a) TI image; (b) TE image; (c) LCVF image.



**Figure 19.** CPT (Computing time) for local detectors. (a) TI image; (b) TE image; (c) LCVF image.

For the global detectors in Figure 18, the data clearly show that the CPT that results from using the recursive updating equations specified by Equation (9) is much reduced, while the CPT from the CRXD algorithm linearly increased with the number of data sample vectors processed. Similar to Figure 19 for the local detectors, they also clearly show that the real-time version local detector has much reduced CPT compared with the commonly-used LRXD provided in Figure 1. It should be noted that the commonly-used LRXD CPT of the first and last few pixels plotted in Figure 19 is a little high, resulting from the boundary extension by local window principle in Figure 1.

### 5. Conclusions

This paper presents both global and local real-time algorithms for anomaly detection in hyperspectral imagery. RX detector is selected as the benchmark theory in this paper. Taking advantage of the Woodbury matrix identity, the proposed global and local detectors can be implemented and processed pixel-by-pixel in a real-time manner. The contributions of this work are summarized as follows:

1. The proposed real-time algorithm has two main features: (1) it is a causal procedure, in the sense that the data samples used for data processing should be only those up to the data sample vector currently being processed; (2) the processing time of the algorithm is negligible, and the proposed method could meet the efficiency requirement by updating the inverse of the correlation matrix without repeated recalculation. Taking advantage of these two features, we gain the benefit of

saving data storage since it only needs to store two types of information, about the previous moment and pixel the currently being processed; this is also much easier in terms of hardware implementation because it uses an update equation instead of matrix inversion calculation.

2. As demonstrated in anomaly detection results for both synthetic and real hyperspectral images, real-time processing offers the significant advantage of seeing time-varying changes in background information. As time moves along, various levels of background suppression produce false alarms, thus producing a tremendous effect on visual assessment. This issue is worth pursuing and beyond the scope of this paper. It would be interesting to investigate this issue since background suppression provides users with a better understanding of what the detected anomalies really are. Specifically, some weak anomalies detected earlier may be overwhelmed by strong anomalies detected later, as the experimental results show. However, this phenomenon is very important in anomaly detection but cannot be observed using commonly used anomaly detectors, which perform one-shot operation to show the final detected anomalies.
3. Despite the fact that local anomaly detection for hyperspectral imagery is not new and has been widely discussed in the literature, the concept of a causal matrix local window (CMLW) and a causal array local window (CALW) for real-time implementation is new.
4. The Woodbury matrix identity has been known for a long time but is newly used to derive a recursive equation in this paper, avoiding computing reversion of covariance/correlation matrix. This will greatly shorten the processing time. Computational complexity analysis is very important for anomaly detection because some targets, especially moving targets, provoke immediate decision-making because they may show up suddenly and then disappear quickly afterwards. As a result, for this kind of detection, the processing time must be very short. This paper gives a comparative discussion of the computing time of different algorithms, showing the advantages of time-saving through the Woodbury matrix identity theory.
5. The proposed real-time detection methods have the additional benefit of being portable. Other detection algorithms, such as CEM, TCIMF, *etc.*, could also be conducted as real-time detectors using the same methodology.

## Acknowledgments

The authors would like to thank Professor Chein-I Chang for his help and guidance, and also for the study within his lab. This study was partially supported by the National Natural Science Foundation of China (No. 61405041, No. 61275010), the China Postdoctoral Science Foundation (Grant No. 2014M551221), the Heilongjiang Postdoctoral Science Found (Grant No. LBH-Z13057), the Key Program of Heilongjiang Natural Science Foundation (No. ZD201216), Program Excellent Academic Leaders of Harbin (No. RC2013XK009003), and the Fundamental Research Funds for the Central Universities (No. HEUCF1408).

## Author Contributions

All authors assisted in the analyzing and editing of the paper. Yulei Wang is the main author who processed the data, developed the methodology, analyzed the results and wrote the manuscript. All authors reviewed and approved the final manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Chang, C.-I. *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*; Kluwer Academic/Plenum Publishers: New York, NY, USA, 2003.
2. Chang, C.-I. *Hyperspectral Data Processing: Algorithm Design and Analysis*; Wiley: Hoboken, NJ, USA, 2013.
3. Rosario, D.S. Algorithm Development for Hyperspectral Anomaly Detection. PhD Thesis, University of Maryland, College Park, MD, USA, 2008.
4. Reed, I.S.; Yu, X. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Trans. Signal. Process.* **1990**, *38*, 1760–1770.
5. Plaza, A.; Martinez, P.; Plaza, J. A new method for target detection in hyperspectral imagery based on extended morphological profiles. In Proceedings of the 2003 IEEE International Conference on Geoscience and Remote Sensing Symposium, 21–25 July, 2003, Toulouse, France; pp. 3772–3774.
6. Kwon, H.; Nasrabadi, N.M. Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 388–397.
7. Foy, B.R.; Theiler, J.; Fraser, A. M. Decision boundaries in two dimensions for target detection in hyperspectral imagery. *Opt. Express.* **2014**, *22*, 608–617.
8. Guo, Q.; Zhang, B.; Ran, Q.; Gao, L.; Li, J.; Plaza, A. Weighted-RXD and linear filter-based RXD: Improving background statistics estimation for anomaly detection in hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2351–2366.
9. Kwon, H.; Der, S.Z.; Nasrabadi, N.M. Projection-based adaptive anomaly detection for hyperspectral imagery. In Proceedings of the 2003 IEEE International Conference on Image Processing, 14–17 September, 2003, Barcelona, Spain; pp. 1001–1004.
10. Ranney, K.I.; Soumekh, M. Hyperspectral anomaly detection within the signal subspace. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 312–316.
11. Ma, L.; Crawford, M.M.; Tian, J. Anomaly detection for hyperspectral images using local tangent space alignment. In Proceedings of the 2010 IEEE International Conference on Geoscience and Remote Sensing Symposium, 25–30 July, 2010, Honolulu, HI, USA; pp. 824–827.
12. Taitano, Y.P.; Geier, B.A.; Bauer, K.W., Jr. A locally adaptable iterative RX detector. *Eurasip. J. Adv. Sign. Process.* **2010**, doi: 10.1155/2010/341908.
13. Matteoli, S.; Diani, M.; Corsini, G. A kurtosis-based test to efficiently detect targets placed in close proximity by means of local covariance-based hyperspectral anomaly detectors. In Proceedings of the 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 6–9 June, 2011, Lisbon, Portugal.
14. Molero, J.M.; Garzon, E.M.; Garcia, I.; Plaza, A. Analysis and optimizations of global and local versions of the RX algorithm for anomaly detection in hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 801–814.

15. Kwon, H.; Der, S.Z.; Nasrabadi, N.M. Adaptive anomaly detection using subspace separation for hyperspectral imagery. *Opt. Eng.* **2003**, *42*, 3342–3351.
16. Matteoli, S.; Veracini, T.; Diani, M.; Corsini, G. A locally adaptive background density estimator: An evolution for RX-based anomaly detectors. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 323–327.
17. Scholkopf; Smola, A.J. *Learning with Kernels*; MIT Press: Cambridge, MA, USA, 2002.
18. Chang, C.-I.; Chiang, S.-S. Anomaly detection and classification for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 1314–1325.
19. Ren, H.; Chang, C.-I. Target-constrained interference-minimized approach to subpixel target detection for hyperspectral images. *Opt. Eng.* **2000**, *39*, 3138–3145.
20. Wang, Y.; Schultz, R.; Chen, S.; Liu, C.; Chang, C.-I. Progressive constrained energy minimization for subpixel detection. *Proc. SPIE* **2013**, *8743*, 874321.
21. Liu, W.; Chang, C.-I. A nested Spatial window-based approach to target detection for hyperspectral imagery. In Proceedings of 2004 IEEE International Conference on Geoscience and Remote Sensing Symposium, 20–24 September, 2004, Anchorage, AK, USA; pp. 266–268.
22. Liu, W.; Chang, C.-I. Multiple-window anomaly detection for hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 644–658.
23. Stellman, C.M.; Hazel, G.G.; Bucholtz, F.; Michalowicz, J.V.; Stocker, A.; Scaaf, W. Real-time hyperspectral detection and cuing. *Opt. Eng.* **2009**, *17*, 17391–17411.
24. Zhao, C.; Wang, Y. Design and analysis of hyperspectral anomaly detection based on Kalman filter theory. *Chin. J. Electron.* **2013**, *22*, 849–854.
25. Molero, J.M.; Garzon, E.M.; Garcia, I.; Quintana-Orti, E.S.; Plaza, A. Efficient implementation of hyperspectral anomaly detection techniques on GPUs and multicore processors. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2256–2266.
26. Rossi, A.; Acito, N.; Diani, M.; Corsini, G. RX architectures for real-time anomaly detection in hyperspectral images. *J. Real-Time Image Process.* **2012**, doi: 10.1007/s11554-012-0292-3.
27. Ensafi, E.; Stocker, A D. An adaptive CFAR algorithm for real-time hyperspectral target detection. *Proc. SPIE* **2008**, doi:10.1117/12.782458.
28. Hager, W.W. Updating the inverse of a matrix. *SIAM Rev.* **1989**, *31*, 221–239.
29. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. *Numerical Recipes: The Art of Scientific Computing*, 3rd revised ed.; Cambridge University Press: Cambridge, UK, 2007.
30. Chen, S.; Wang, Y.; Wu, C.; Liu, C.; Chang, C.-I. “Real-time causal processing of anomaly detection for hyperspectral imagery. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 1511–1534.
31. USGS Website. Available online: <http://speclab.cr.usgs.gov/cuprite.html> (accessed on 27 July 2014).
32. Chang, C.-I.; Hsueh, M. Characterization of anomaly detection for hyperspectral imagery. *Sens. Rev.* **2006**, *26*, 137–146.