Anomaly Detection Using Causal Sliding Windows

Chein-I Chang, *Fellow, IEEE*, Yulei Wang, and Shih-Yu Chen, *Member, IEEE*

***Abstract*—Anomaly detection using sliding windows is not new but using causal sliding windows has not been explored in the past. The need of causality arises from real-time processing where the used sliding windows should not include future data samples that have not been visited, i.e., data samples come in after the currently being processed data sample. This paper develops an approach to anomaly detection using causal sliding windows, which has the capability of being implemented in real time. In doing so, three types of causal windows are defined: 1) causal sliding square matrix windows; 2) causal sliding rectangular matrix windows; and 3) causal sliding array windows. By virtue of causal sliding windows, a causal sample covariance/correlation matrix can be derived for causal anomaly detection. As for the causal sliding array windows, recursive update equations are also derived and thus speed up real-time processing.**

***Index Terms*—Causal anomaly detection, causal sliding array window, causal sliding rectangular matrix window, causal sliding square matrix window, K-RX detector (K-RXD), R-RX detector (R-RXD).**

I. Introduction

ANOMALY detection has received considerable interest in hyperspectral data exploitation [1] since a hyperspectral imager can uncover many subtle targets, which are not known *a priori* or cannot be visualized by inspection. It is particularly crucial when anomalies such as moving targets may appear in a short period and vanish thereafter, in which case, timely detection is necessary and real-time processing of anomaly detection becomes inevitable. Unfortunately, many anomaly detection algorithms reported in the literature are actually not real-time processing algorithms even though some of them claim to be. For example, the most widely used anomaly detector known as RX detector (RXD) developed by Reed and Yu in [2] along with its many variants [3] cannot be implemented in real time, due to its use of covariance matrix which requires entire data sample vectors to calculate the sample mean vector [4]–[8]. In addition, many local or adaptive anomaly detectors which make use of sliding windows to capture local statistics to improve anomaly detection are not real-time processing detectors either because their used sliding windows include future data sample vectors come after the currently being processed data sample vector [9]–[14]. All these types of anomaly detection algorithms violate a key element required for real-time processing, which is causality [15]. According to [16], a causal signal processing algorithm can only process data samples vector up to the data sample vector currently being processed. In other words, the data sample vectors used for data processing can be only those which have been visited and any future data sample vector that comes in after the current data sample vector should not be included in data processing. Recently, such issue in causal anomaly detection has been investigated for real-time processing [17], [18]. However, anomaly detection using sliding causal windows remains unresolved and has received little interest. This is mainly due to the fact that if a sliding window to be used for anomaly detection is relatively small, its processing time is negligible. In this case, it can be processed in near real time, but it is still not a real-time processing algorithm because the used window centered at the current data sample vector includes future data sample vectors which come after the center of the window. Another issue is the size of the used sliding window. If it is small and can be implemented in near real time, the resulting performance may not be desirable. If it is too large, the resulting performance may be better but it cannot be implemented in real time since the processing time may exceed time constraints. To resolve this issue, this paper develops an approach to anomaly detection using causal sliding windows which can be implemented in a causal manner where a causal sample covariance/correlation matrix can be defined by data sample vectors embraced in a causal sliding window. Three types of causal sliding windows are defined: 1) causal sliding square matrix windows; 2) causal sliding rectangular windows; and 3) causal sliding array windows. While a causal sliding square and rectangular matrix window requires bookkeeping to keep track of data sample vectors, a causal sliding array window works like a queue. As a result, recursive equations can be derived for the causal sliding array window so that anomaly detection using a causal sliding array window can be updated recursively by only including the new incoming data sample vector for data processing without reprocessing the entire previously visited data sample vectors over and over again. Accordingly, this capability provides feasibility of real-time processing.

Various anomalies may exhibit different local properties in terms of size, spectral signature, and spectral correlation with their surrounding sample vectors. The development of adaptive anomaly detection is designed to capture their local spectral

Manuscript received September 30, 2014; revised February 03, 2015; accepted March 23, 2015. Date of publication May 15, 2015; date of current version August 11, 2015. The work of S.-Y. Chen was supported by MOST 103-2218-E-224-005. *(Corresponding author: Shih-Yu Chen.)*

C.-I Chang is with the Remote Sensing Signal and Image Processing Laboratory, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, MD 21250 USA, and with the Department of Electrical Engineering, National Chung Hsing University, Taichung 402, Taiwan, and also with the Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan, R.O.C. (e-mail: cchang@umbc.edu).

Y. Wang is with the Information and Communication Engineering College, Harbin Engineering University, Harbin 150001, China (e-mail: wangyulei.wyl@163.com).

S.-Y. Chen is with the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan (e-mail: sychen@yuntech.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2015.2422996

statistics. In order to adapt local statistics of anomalies, many reports have modified K-RXD by replacing the covariance matrix $\mathbf{K}$ with a local covariance matrix calculated by data sample vectors embraced by so-called local or sliding window specified by W. By virtue of the window W, anomalies can be detected by spectral statistics of the data sample vectors in the window W which varies sample by sample [19]. This type of anomaly detection is generally referred to as adaptive anomaly detection and performs K-RXD by varying local spectral statistics characterized by data sample vectors in a window W. However, it seems that using a sliding window to capture local statistics in the sense of causality has never been investigated in the literature. Since causality is a prerequisite to real-time processing, no anomaly detector using sliding windows can be considered as a real-time anomaly detector. This paper addresses this issue and further develops three different types of causal sliding windows, called causal sliding square matrix window, causal sliding rectangular matrix window and causal sliding array window, all of which can be implemented in real-time processing.

II. Commonly Used Anomaly Detection

The most widely used anomaly detector is probably the one developed by Reed and Yu in [1] referred to as K-RXD, where $\mathbf{K}$ is the global sample covariance matrix. Since then, many various K-RXD-type anomaly detectors have been proposed including some using sliding window to make anomaly detection adaptive [13], [15], [18]–[20].

Assume that $\{\mathbf{r}_i\}_{i=1}^N$, where N is the total number of entire data sample vectors in the data; and $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iL})^T$ is the ith data sample vector, where L is the total number of spectral bands.

A. RX Detector

The K-RXD, denoted by $\delta^{\text{K-RXD}}(\mathbf{r})$, is specified by

$$\delta^{\text{K-RXD}}(\mathbf{r}) = (\mathbf{r} - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{r} - \boldsymbol{\mu}) \tag{1}$$

where $\boldsymbol{\mu}$ is the global sample mean vector and $\mathbf{K}$ is the global sample data covariance matrix formed by $\mathbf{K} = (1/N) \sum_{i=1}^N (\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T$. The form of $\delta^{\text{K-RXD}}(\mathbf{r})$ in (1) is actually the well-known Mahalanobis distance. Since $\mathbf{K}$ is a nonnegative definite matrix, it can be expressed as $\mathbf{K} = \mathbf{K}^{1/2}\mathbf{K}^{1/2}$. Using $\mathbf{K}^{-1/2}$ as a transformation matrix, (1) can be simply reduced to

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

with $\tilde{\mathbf{r}} = \mathbf{K}^{-1/2}(\mathbf{r} - \boldsymbol{\mu})$. Equation (2) shows that K-RXD actually calculates the square of the vector length of $\tilde{\mathbf{r}}$, $||\tilde{\mathbf{r}}||^2$ 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, the use of $\mathbf{K}$ and $\boldsymbol{\mu}$ to remove the first two-order statistics is called data sphering. Since the image background can be generally described by the second-order statistics, K-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 (2), so as to achieve background suppression.

B. R-RXD

Another anomaly detector derived from K-RXD is denoted by $\delta^{\text{R-RXD}}(\mathbf{r})$ and is specified by

$$\delta^{\text{R-RXD}}(\mathbf{r}) = \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} \tag{3}$$

where $\mathbf{R}$ is the global sample data autocorrelation matrix formed by $\mathbf{R} = (1/N)\sum_{i=1}^N \mathbf{r}_i\mathbf{r}_i^T$. The form of K-RXD in (1) may not explain quite well about how anomaly detector performs by using its detected anomaly intensity and contrast. However, by virtue of (3), the concept of anomaly intensity and contrast can be well-explained by R-RXD. First of all, the form of (3) can be decomposed into two components: 1) $\mathbf{R}^{-1}\mathbf{r}$; and 2) $\mathbf{r}^T$. The first component, $\mathbf{R}^{-1}\mathbf{r}$ carried out by R-RXD, actually performs background suppression via the use of the 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 performs a matched filter to detect anomaly intensity via matching the background-compressed data sample $\mathbf{R}^{-1}\mathbf{r}$ using its own signature $\mathbf{r}$ as a matched signal source vector. 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 mapper (SAM) [1] by calculating the angle between $\mathbf{R}^{-1}\mathbf{r}$ and $\mathbf{r}^T$. Therefore, from a practical point of view there should have an absolute value in (3) to avoid negative values caused by the angle. It should be noted that the negative value of R-RXD does not imply negative magnitude. It simply says that $\mathbf{R}^{-1}\mathbf{r}$ and $\mathbf{r}^T$ are in opposite directions.

However, it should be noted that there is a significant difference between R-RXD and K-RXD because the former detects data sample vector itself; but the latter actually detects the data sample variation from the global sample mean, i.e., data sample vectors which have large gradients. So, technically speaking, K-RXD does not detect intensities of anomalies themselves but rather their gradient intensities. As a result, K-RXD works as a distance measure like Mahalanobis distance, whereas R-RXD which can be considered as a matched filter [1], [3].

III. Design of Causal Sliding Windows

In this section, we design three types of causal sliding windows: 1) causal sliding square matrix window; 2) causal sliding rectangular matrix window; and 3) causal sliding array window, all of which can be used by an anomaly detector to adjust $\mathbf{K}$ or $\mathbf{R}$ dynamically to capture changes in sample by sample in the background so as to achieve sample varying background suppression as opposed to sample invariant-based anomaly detectors reported in the literature.

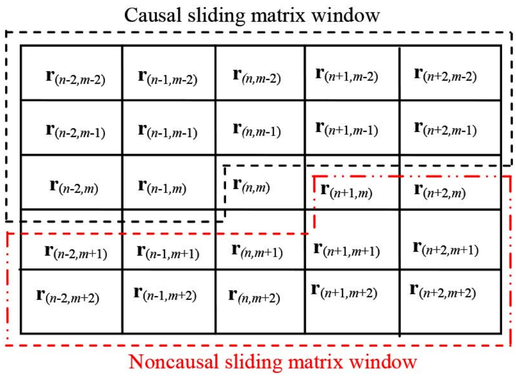

Fig. 1. Causal and noncausal windows of a window with window size $w_a = 25$ and width $a = 2$.

A. Causal Sliding Square Matrix Windows

First of all, consider a standard window commonly used in image processing which is a square window [21]. Assume that the square window is specified by $w^2 = (2a+1) \times (2a+1)$ with $w = 2a+1$ and $a \geq 0$ where $w^2 = w \times w$ is defined as "*window size*" and $a = (w-1)/2$ is considered as "*window width.*" It is worth noting that the window is generally odd because it is centered at a given data sample vector $\mathbf{r}_n$, which is currently being processed. So, when $a = 0$, the square matrix window is reduced to its center $\mathbf{r}_n$ and no sample correlation surrounding $\mathbf{r}_n$ is considered. Basically, the data sample vectors in a square matrix window can be equally split into two halves of data sample vectors, each of which has an equal number of data sample vectors $(w^2-1)/2 = 2(a^2+a)$. The first half is called causal data sample vectors, which precede the current data sample vector $\mathbf{r}_n$ and the other half is noncausal data sample vectors, which appear after the current data sample vector $\mathbf{r}_n$.

Let $\{\mathbf{r}_i\}_{i=1}^{n}$ be a set of all data sample vectors up to the currently being processed data sample vector $\mathbf{r}_n$. A sliding causal square matrix window W is then defined by its window size and width specified by $w_a = (2a+1) \times (2a+1)$ and $a = (\sqrt{w_a} - 1)/2$, respectively, as a window, which includes all the $(w_a - 1)/2 = 2(a^2+a)$ causal data sample vectors in the square window W that appear before the $\mathbf{r}_n$ and have been visited, whereas a noncausal matrix window includes only those $(w_a-1)/2 = 2(a^2+a)$ noncausal data sample vectors which are future data sample vectors yet to be visited within the square window W. With this definition $w^2 = w_a$.

Fig. 1 illustrates its concept by specifying the window W with size of $w_a = 5 \times 5$ and $a = (\sqrt{w_a} - 1)/2 = (5-1)/2 = 2$, where the pixel currently being processed is specified by its 2-D spatial location, $\mathbf{r}_{(n,m)}$ with (n,m) indicating its spatial location for a better illustrative purpose. In this case, the causal square matrix window comprises of all the causal 12 data sample vectors $\left\{\mathbf{r}_{(n-i,m-j)}\right\}_{i=0,j=0}^{2,2} - \mathbf{r}_{(n,m)}$ and the noncausal square matrix window highlighted by RED is also made up of all the 12 noncausal data sample vectors yet to be processed in W $\left\{\mathbf{r}_{(n+i,m+j)}\right\}_{i=0,j=0}^{2,2} - \mathbf{r}_{(n,m)}$.

So, when the sliding square matrix window in Fig. 1 moves its center to the next data sample vector $\mathbf{r}_{n+1}$, the causal matrix window also moves and the data sample vectors included in this moved causal matrix window are shown in Fig. 2, where the two sliding causal matrix windows at $\mathbf{r}_n$ and $\mathbf{r}_{n+1}$ are specified by dotted and dashed lines, respectively.

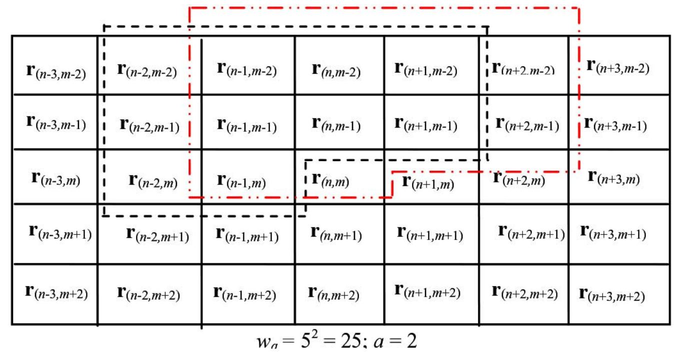

Fig. 2. Causal square matrix windows at $\mathrm{r}_{(n,m)}$ and $\mathrm{r}_{(n+1,m)}$.

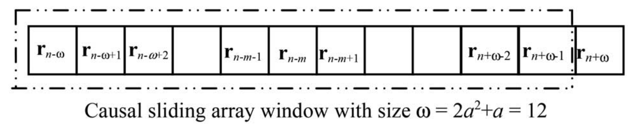

Fig. 3. A causal sliding array window with size specified by $\omega = 2a^2 + a = 12$.

B. Causal Sliding Array Windows

As we can see from Fig. 2, all data sample vectors excluded from the causal matrix window are not removed in sequence. In this case, let $\mathbf{r}_{n-m} = \mathbf{r}_{(n,m)}$. For example, in Fig. 2, the $\mathbf{r}_{(n-2,m-2)}$, $\mathbf{r}_{(n-2,m-1)}$ and $\mathbf{r}_{(n-2,m)}$ in the causal matrix window centered at $\mathbf{r}_{n-m}$ are removed from the causal matrix window centered at $\mathbf{r}_{n-m+1}$, while $\mathbf{r}_{(n+2,m-2)}$ and $\mathbf{r}_{(n+2,m-1)}$ which are not included in the causal matrix window centered at $\mathbf{r}_{n-m}$ are now added to the causal matrix window centered at $\mathbf{r}_{n-m+1}$. Obviously, it requires bookkeeping to keep track of which data sample vector should be removed and which data sample vectors should be added as a causal matrix window moves on. To resolve this issue, we can stretch out the causal matrix window in Fig. 1 as a linear array shown in Fig. 3 by letting $\mathbf{r}_{n-m} = \mathbf{r}_{(n,m)}$, $\mathbf{r}_{n-\omega} = \mathbf{r}_{(n-2,m-2)}$, $\mathbf{r}_{n-\omega+1} = \mathbf{r}_{(n-1,m-2)}$, etc., in which case, we can define the array window size as $\omega = 2(a^2+a) = 12$.

Using Fig. 3, we define a causal sliding array window corresponding to Fig. 1 as a linear array with array window size given by $\omega = 2a^2 + a = 12$ sliding along with the currently being processed nth data sample vector $\mathbf{r}_n$ according to data processing line by line as a linear array, which embraces $2a^2 + a = 12$ pixels $\{\mathbf{r}_i\}_{i=n-\omega}^{n-1}$ preceding the processed data sample vector $\mathbf{r}_n$. In other words, the causal sliding array window of width ω defined in Fig. 3 is formed by a linear array which consists of ω data sample vectors preceding the current processed data sample $\mathbf{r}_n$. It is no longer a square window of size w^2 shown in Fig. 1. It should be also noted that the current data sample vector $\mathbf{r}_n$ is not included in the causal sliding array window. So, when a causal sliding array window moves along with the data sample vectors, the linear array simply performs like a queue,

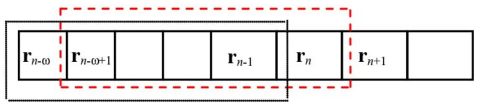

Fig. 4. Causal sliding array window at r_n with width specified by ω.

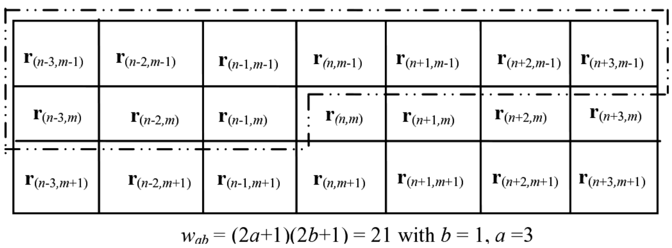

Fig. 5. Causal rectangular matrix window at $\mathrm{r}_{(n,m)}$ with window size of $w_{ab} = 3 \times 7$.

first in and first out. Fig. 4 shows the causal sliding array window at $\mathbf{r}_n$ depicted by dotted lines and the causal sliding array window at $\mathbf{r}_{n+1}$ depicted by dashed lines, where the farthest data sample vector $\mathbf{r}_{n-\omega}$ from $\mathbf{r}_n$ in the causal sliding array window at $\mathbf{r}_n$ is removed from the causal sliding array window at $\mathbf{r}_n$, while the most recent data sample vector $\mathbf{r}_n$ is then added to the causal sliding array window at $\mathbf{r}_{n+1}$.

The difference between the causal sliding square matrix window shown in Fig. 2 and the causal sliding array window shown in Fig. 4 is that when a new data sample vector is due to being processed $\mathbf{r}_{(n+1,m)}$ in Fig. 2 and $\mathbf{r}_{n+1}$ in Fig. 4, the data sample vectors to be removed, $\mathbf{r}_{(n-2,m-2)}$, $\mathbf{r}_{(n-2,m-1)}$ and $\mathbf{r}_{(n-2,m)}$ and added, $\mathbf{r}_{(n+2,m-2)}$, $\mathbf{r}_{(n+2,m-1)}$ in Fig. 2 are not consecutive, while the data sample vector to be removed from the array $\mathbf{r}_{n-\omega}$ and the data sample vector to be added, $\mathbf{r}_n$ in Fig. 4 are successive. As a consequence, from practical implementation using causal sliding array window is much simpler than using causal sliding square matrix window even though the latter is the common practice in image processing [21].

C. Causal Sliding Rectangular Matrix Window

Interestingly, both the causal sliding square matrix window defined in Section III-A and causal sliding array window defined in Section III-B can be interpreted as special cases of a more general form which makes use of a causal sliding rectangular window W specified by its length a, and width b, and size $w_{ab} = (2b+1) \times (2a+1)$. Its idea can be illustrated in Fig. 5 using Fig. 1 as an example.

As a result, when $b = a$, the window size w_{ab} becomes w_a and the causal sliding rectangular matrix window is reduced to a causal sliding square matrix window. When $b = 0$, then the window size w_{ab} becomes ω and the causal sliding rectangular matrix window is reduced to a causal sliding array window as shown in Fig. 6, where $\mathbf{r}_n$ is the currently being processed data sample vector and ω is the array window size.

IV. Causal Anomaly Detection

Using causal sliding windows defined in Section III, we can now define a causal anomaly detector which makes use of causal windows to capture background varying with sample vectors to perform adaptive anomaly detection. Since K-RXD and R-RXD described in Section II are of major interests, these two anomaly detectors will be used to derive causal anomaly detectors as follows.

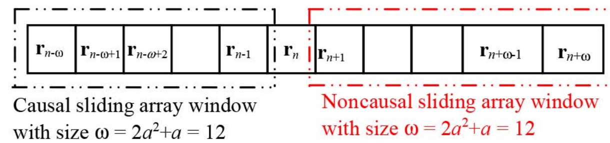

Fig. 6. Causal and noncausal sliding array window with size $\omega = 2a^2 + a = 12$.

A causal R-RXD (CR-RXD) using a causal sliding window W, denoted by $\delta^{\text{CR-RXD}}(\mathbf{r})$, can be derived from (3) and specified by

$$\delta_{\mathrm{W}}^{\text{CR-RXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \tilde{\mathbf{R}}^{-1}(n)\mathbf{r}_n \tag{4}$$

where $\mathbf{r}_n$ is the nth data sample vector currently being processed and $\tilde{\mathbf{R}}(n)$ is called "*causal*" sample correlation matrix formed by data sample vectors in a causal sliding window W if it is defined by $\tilde{\mathbf{R}}(n) = (1/n_{\mathrm{W}}) \sum_{\mathbf{r}_i \in \mathrm{W}} \mathbf{r}_i \mathbf{r}_i^T$, where n_W is the total number of data sample vectors in W.

In analogy with (4), a causal version of the K-RXD in (1) can be re-expressed as

$$\delta_{\mathrm{W}}^{\text{CK-RXD}}(\mathbf{r}_n) = (\mathbf{r}_n - \tilde{\boldsymbol{\mu}}(n))^T \tilde{\mathbf{K}}(n)^{-1} (\mathbf{r}_n - \tilde{\boldsymbol{\mu}}(n)) \tag{5}$$

where $\tilde{\boldsymbol{\mu}}(n) = (1/n_{\mathrm{W}}) \sum_{\mathbf{r}_i \in \mathrm{W}} \mathbf{r}_i$ is the "*causal*" sample mean averaged over all data sample vectors, $\{\mathbf{r}_i\}_{i=1}^{n-1}$ and $\tilde{\mathbf{K}}(n) = (1/n_{\mathrm{W}}) \sum_{\mathbf{r}_i \in \mathrm{W}} (\mathbf{r}_i - \tilde{\boldsymbol{\mu}}(n)) (\mathbf{r}_i - \tilde{\boldsymbol{\mu}}(n))^T$ is the "*causal*" covariance matrix formed by all the data sample vectors in a causal sliding window W.

Most recently, causal anomaly detection without using causal sliding windows was investigated in [18], where two causal anomaly detectors: 1) causal R-RXD (CR-RXD) and 2) causal K-RXD (CK-RXD) were developed. However, the causal sample correlation used in both CR-RXD and CK-RXD is specified by all data sample vectors $\{\mathbf{r}_i\}_{i=1}^{n-1}$ that have been visited and processed before the current data sample vector $\mathbf{r}_n$. It is interesting to note that these two causal anomaly detectors can be actually considered as special cases of our proposed causal anomaly detectors $\delta_{\mathrm{W}}^{\text{CR-RXD}}(\mathbf{r}_n)$ in (4) and $\delta_{\mathrm{W}}^{\text{KR-RXD}}(\mathbf{r}_n)$ in (5) using a causal sliding array window W, which grows and keeps adding new data sample vectors, i.e., $\mathrm{W} = \cup_{i=1}^{n-1} \mathbf{r}_i$. Accordingly, this paper can be considered as adaptive version of causal anomaly detection in [18] with using a causal sliding window W to capture local spectral statistics among data sample vectors in W.

V. Recursive Causal Anomaly Detection

Theoretically, (4) can be implemented in real time. However, the causal sample correlation $\tilde{\mathbf{R}}(n)$ in (4) varies with data sample vectors to be processed and must be recalculated each time as long as a new data sample vector is fed in. This processing time generally goes beyond time constraints required for real-time implementation. In order to resolve this issue, this section derives a recursive causal information update equation which only needs to update causal anomaly detection by including innovations information provided by the new data sample vector and its correlation with processed information obtained from previous data sample vectors.

A. Derivations of Recursive Equations

Assume that the width of a causal sliding array window is specified by ω and the data sample vector to be processed is $\mathbf{r}_n$. To emphasize the width of ω and the processed data sample vector $\mathbf{r}_n$, we re-write $\tilde{\mathbf{R}}(n)$ in (4) as $\tilde{\mathbf{R}}_\omega(n)$. Then $\tilde{\mathbf{R}}_\omega(n+1)$ can be further expressed as

$$\tilde{\mathbf{R}}_\omega(n+1) = \left[\left(\tilde{\mathbf{R}}_\omega(n) - \mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T\right) + \mathbf{r}_n\mathbf{r}_n^T\right]. \quad (6)$$

Now, in order to calculate the inverse of $\tilde{\mathbf{R}}_\omega(n+1)$, i.e., $\tilde{\mathbf{R}}_\omega^{-1}(n+1)$, we repeatedly make use of the following Woodbury matrix identity [22] twice

$$\left[\mathbf{A} + \mathbf{u}\mathbf{v}^T\right]^{-1} = \mathbf{A}^{-1} - \frac{\left[\mathbf{A}^{-1}\mathbf{u}\right]\left[\mathbf{v}^T\mathbf{A}^{-1}\right]}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \quad (7)$$

to first bring out $\mathbf{r}_n\mathbf{r}_n^T$ with $\mathbf{A} = \left(\tilde{\mathbf{R}}_\omega(n) - \mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T\right)$ and $\mathbf{u} = \mathbf{v} = \mathbf{r}_n$; then bring out $-\mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T$ by letting $\mathbf{A} = \tilde{\mathbf{R}}_\omega(n)$ and $\mathbf{u} = -\mathbf{v} = \mathbf{r}_{n-\omega}$ as follows [(8), shown at the bottom of the page], where $\left(\tilde{\mathbf{R}}_\omega(n) - \mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T\right)^{-1}$ can be further updated recursively by

$$\begin{aligned}\left(\tilde{\mathbf{R}}_\omega(n) - \mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T\right)^{-1} &= \tilde{\mathbf{R}}_\omega^{-1}(n) - \frac{\left[\tilde{\mathbf{R}}_\omega^{-1}(n)\mathbf{r}_{n-\omega}\right]\left[-\mathbf{r}_{n-\omega}^T\tilde{\mathbf{R}}_\omega^{-1}(n)\right]}{1 - \mathbf{r}_{n-\omega}^T\tilde{\mathbf{R}}_\omega^{-1}(n)\mathbf{r}_{n-\omega}} \\ &= \tilde{\mathbf{R}}_\omega^{-1}(n) + \frac{\left[\tilde{\mathbf{R}}_\omega^{-1}(n)\mathbf{r}_{n-\omega}\right]\left[\mathbf{r}_{n-\omega}^T\tilde{\mathbf{R}}_\omega^{-1}(n)\right]}{1 - \mathbf{r}_{n-\omega}^T\tilde{\mathbf{R}}_\omega^{-1}(n)\mathbf{r}_{n-\omega}}. \quad (9)\end{aligned}$$

By virtue of (8) and (9), $\tilde{\mathbf{R}}_\omega(n+1)$ can be updated recursively by $\tilde{\mathbf{R}}_\omega(n)$ via deleting the information $\mathbf{r}_{n-\omega}$ and adding the new information $\mathbf{r}_n$.

B. Computational Complexity

The advantage of using the causal sliding array windows over causal sliding matrix windows is the use of recursive equations (8) and (9), where deriving similar recursive equations for using causal sliding matrix windows is feasible but is much more complicated as described in the beginning paragraph of Section III-B. In particular, it must repeatedly implement Woodbury's identity as many times as it brings out excluded as well as included data sample vectors. In addition, this number is also determined by the size of the used causal window. So, it is practically not worthwhile. By contrast, the use of causal sliding array window requires only two implementations of Woodbury's identity regardless of its width as shown in (8) and (9). Such a significant benefit arises from the recursive nature in (8) and (9).

According to (8), it only requires calculations of three quantities:

1) an $L \times 1$ vector calculated by $\boldsymbol{\varphi} = (\tilde{\mathbf{R}}_\omega(n) - \mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T)^{-1}\mathbf{r}_n$;
2) an $L \times L$ matrix calculated by an outer product of $\boldsymbol{\varphi}$: $\boldsymbol{\varphi}\boldsymbol{\varphi}^T$;
3) a scalar calculated by an inner product: $\mathbf{r}_n^T\boldsymbol{\varphi}$.

where $\left(\tilde{\mathbf{R}}_\omega(n) - \mathbf{r}_{n-\omega}\mathbf{r}_{n-\omega}^T\right)^{-1}$ can be calculated by (9) and also requires another three similar quantities:

1) an $L \times 1$ vector calculated by $\boldsymbol{\psi} = \tilde{\mathbf{R}}_\omega^{-1}(n)\mathbf{r}_{n-\omega}$;
2) an $L \times L$ matrix calculated by an outer product of $\boldsymbol{\psi}$: $\boldsymbol{\psi}\boldsymbol{\psi}^T$;
3) a scalar calculated by an inner product: $\mathbf{r}_n^T\boldsymbol{\psi}$.

So, the computational complexity of processing a single data sample vector using a causal sliding window W specified by its window size ω requires calculations of two $L \times 1$ vectors required by (1) and (a), two outer products of an $L \times 1$ vector by (2) and (b), and two inner products of two $L \times 1$ vectors by (3) and (c). In addition to that, it only needs to calculate its initial condition, the inverse of $\tilde{\mathbf{R}}_\omega(n_0)$ once. It should be also noted that $\tilde{\mathbf{R}}_\omega^{-1}(n)$ is updated by (8), where its initial condition n_0 must guarantee that $\tilde{\mathbf{R}}_\omega(n_0)$ is of full rank to avoid singularity. In other words, the size of the used causal sliding window W, ω must at least equal to or greater than the total number of spectral bands.

Three comments are worthwhile.

1) The causal sliding window should not include the current data sample vector $\mathbf{r}_{(n,m)}$ or $\mathbf{r}_n$, because it will cause $\mathbf{r}_{(n,m)}$ or $\mathbf{r}_n$ to be suppressed in the background [15].
2) The sliding causal sliding array window defined in Fig. 3 can be made sample-variant. More specifically, the width ω can be made a function of the data sample vector $\mathbf{r}_n$ to be processed, denoted by $\omega(\mathbf{r}_n)$. For example, if $\omega(\mathbf{r}_n) = n-1$, then the anomaly detection using sliding windows with width $n-1$ is reduced to causal anomaly detection developed in [18].
3) Like $\tilde{\mathbf{R}}_\omega(n)$ the causal sample covariance matrix, $\tilde{\mathbf{K}}(n)$ in (5) can be also obtained by recursive update equations similar to (8) and (9) but their derivations are more complicated than (8)–(9) (see the derivations of CK-RXD in [18]) and are not included here.

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

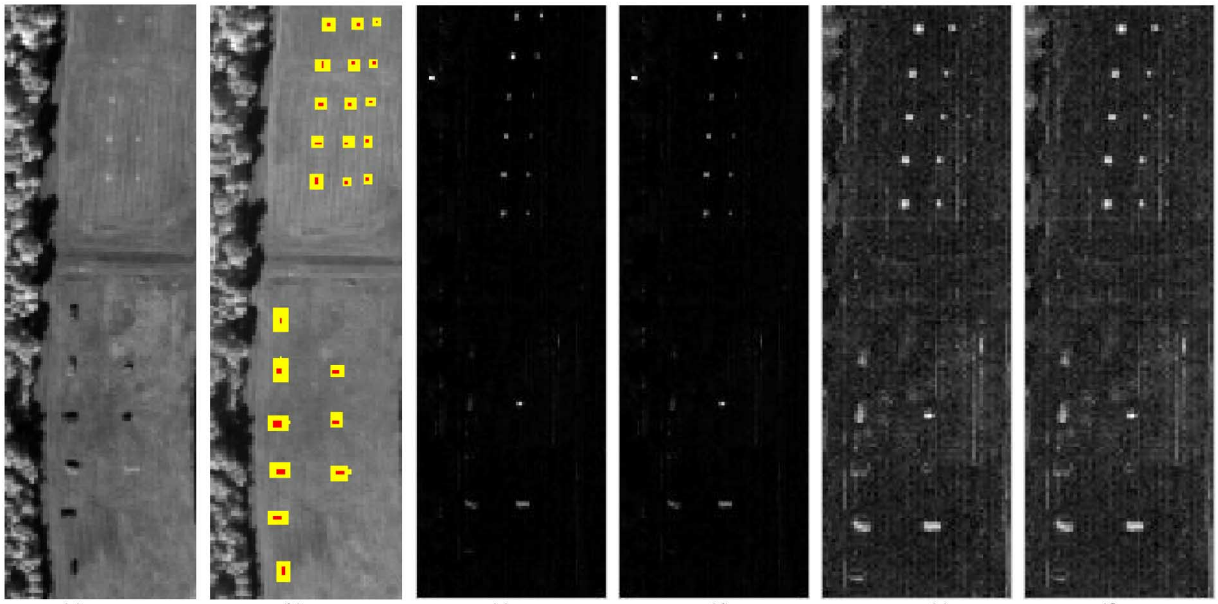

Fig. 7. (a) HYDICE panel-vehicle scene and (b) its ground truth; abundance fractional maps by two commonly used global anomaly detectors: (c) K-RXD; (d) R-RXD; (e) K-RXD in db; (f) R-RXD in db.

VI. Real Image Experiments

To demonstrate that anomaly detection using causal sliding windows works in real hyperspectral imagery, a size of 200×74 pixels HYDICE image scene shown in Fig. 7(a) along with its ground truth provided in Fig. 7(b) where the center and boundary pixels of objects are highlighted by red and yellow, respectively, is used for the experiment. It was acquired by 210 spectral bands with a spectral coverage from 0.4 to $2.5\ \mu\mathrm{m}$ where the spatial resolution is 1.56 m and spectral resolution is 10 nm. Low signal/high noise bands: bands 1–3 and bands 202–210; and water vapor absorption bands: bands 101–112 and bands 137–153 were removed. So, a total of 169 bands were used in experiments.

There are several advantages of using this HYDICE image scene in Fig. 7(a). First, the ground truth provides precise spatial locations of all man-made target pixels which allow us to evaluate performance of anomaly detection pixel by pixel. Second, the provided ground truth enables us to perform receiver operating characteristic (ROC) analysis for anomaly detection via ROC curves of detection rate versus false alarm rate. Third, the scenes has various sizes of objects that can be used to evaluate ability of an anomaly detector in detecting anomalies with different sizes, an issue that has not been really addressed in many reports. Finally and most importantly, the natural background and known targets make visual assessment more easily to see various degrees of background be suppressed by an anomaly detector.

In order to verify the effectiveness of local causal anomaly detectors, only causal sliding array window was implemented as the reasons discussed at the end of Section V. Two recursive causal anomaly detectors described in Section V were implemented using various sizes of causal sliding array windows as shown in Fig. 7(c)–(f) in db for background assessment where db is defined by $20\log_{10}x$ according to signal processing. Apparently, the global anomaly detectors had very good performance especially for the panels of the upper part shown in Fig. 7(e) and (f), where the subpanel pixels in the third column were actually detected. However, they cannot be implemented in real time due to the calculation of global covariance matrix or correlation matrix, which is implemented by the global sample spectral correlation formed by the entire image data.

Anomaly detection using causal sliding array windows is implemented in a real-time and causal manner. This causal anomaly detector is different from the commonly used dual local detectors with inner window and outer window centered by the pixel being processed. Due to the need of real-time process, the local window is designed causally which only uses pixels in a causal sliding array window of a fixed size up to the data sample being processed. It should be noted that the width of the causal sliding array window must be greater or equal to the total band number to avoid a singularity problem in the inversion of the correlation matrix. In order to see how anomaly detection using causal sliding array windows of various widths from 200 up 900 with step size of 100 pixels, Fig. 8(a)–(h) shows the detection abundance fractional maps with their detected abundance fractions shown in db scale in Fig. 9. Figs. 8 and 9 show the gray scale detection map in original scale and dB scale for the two detectors. It seems that db scale gives a better visual inspection.

According to our experiments, the detection result was poor using causal sliding array window $\mathrm{width} = 200$, whereas the performance began to improve as the causal sliding array window width increases. When the causal sliding array window width becomes very large, their detection performances were similar by visual inspection as shown in Fig. 8(e)–(h), with the causal sliding array window width greater than or equal to 600.

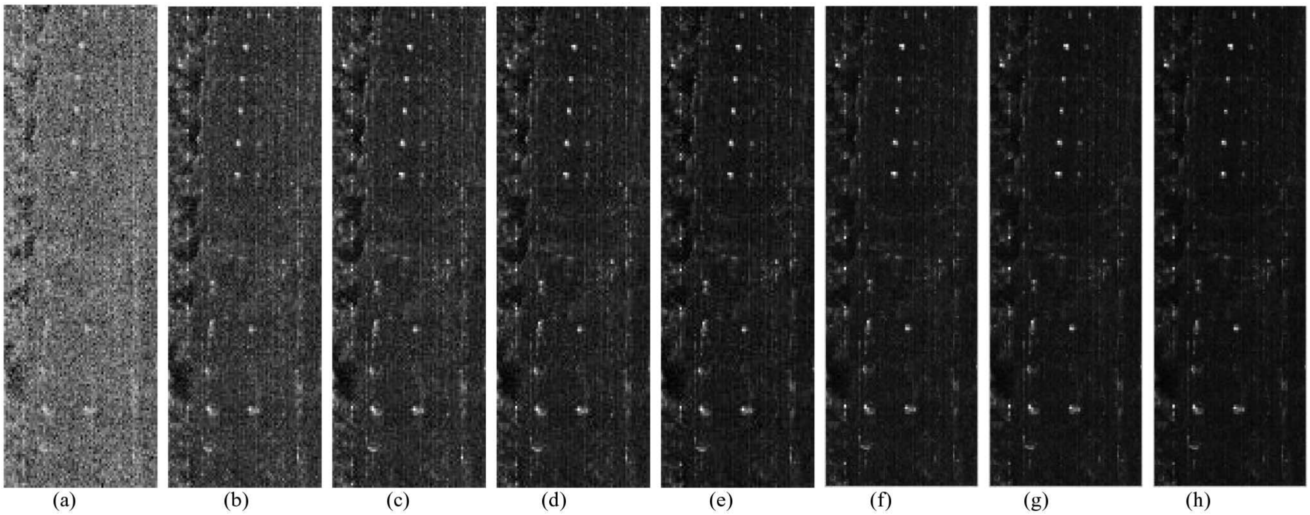

Fig. 8. Detection abundance fractional maps by causal anomaly detectors with different causal sliding array window width where (a) $\mathrm{width} = 200$ pixels; (b) width 300; (c) $\mathrm{width} = 400$; (d) $\mathrm{width} = 500$; (e) $\mathrm{width} = 600$; (f) $\mathrm{width} = 700$; (g) $\mathrm{width} = 800$; and (h) $\mathrm{width} = 900$.

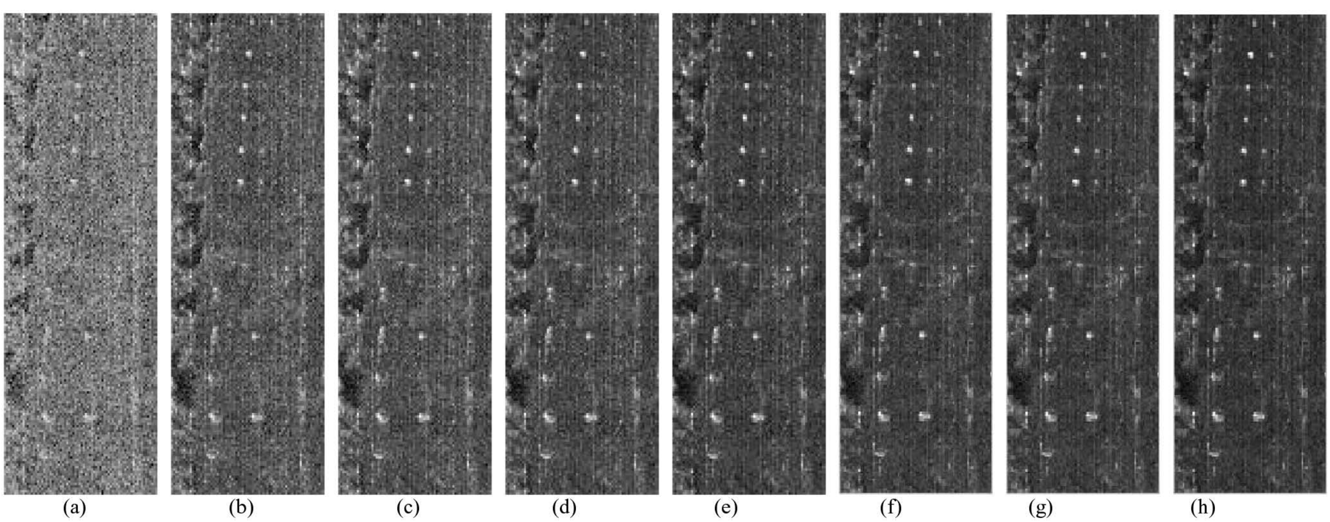

Fig. 9. Detection abundance fractional maps in db scale shown in Fig. 8.

Fig. 9(a)–(h) also shows detection maps in db of Fig. 8(a)–(h) for comparison.

In order to further quantitatively measure detection performance, a three-dimensional (3-D) ROC analysis is performed using the ground truth provided by Fig. 7(b). In doing so, an idea similar to that proposed in [23] and [24] can be derived by converting real values to hard decisions as follows.

Assume that $\delta^{AD}(\mathbf{r})$ is the detected abundance fraction obtained by operating an anomaly detector on a data sample vector $\mathbf{r}$. We then define a normalized detected abundance fraction $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ by

$$\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r}) = \frac{\hat{\delta}^{\mathrm{AD}}(\mathbf{r}) - \min_{\mathbf{r}}\hat{\delta}^{\mathrm{AD}}(\mathbf{r})}{\max_{\mathbf{r}}\hat{\delta}^{\mathrm{AD}}(\mathbf{r}) - \min_{\mathbf{r}}\hat{\delta}^{\mathrm{AD}}(\mathbf{r})}. \tag{10}$$

More specifically, $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ in (10) can be regarded as a probability vector which calculates the likelihood of the data sample vector $\mathbf{r}$ to be detected as anomaly according to its detected abundance fraction, $\delta^{AD}(\mathbf{r})$. By virtue of (10) we can develop an abundance percentage anomaly converter (ACV) with $a\%$ as a thresholding criterion, referred to as $a\%\mathrm{ACV}$, $\chi_{a\%\mathrm{ACM}}(\mathbf{r})$ similar to one proposed in [1], [25] as follows:

$$\chi_{a\%\mathrm{ACM}}(\mathbf{r}) = \begin{cases} 1, & \text{if } \hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r}) \geq \tau = \frac{a}{100} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

If $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ in (11) exceeds $\tau = a\%/100$, then the $\mathbf{r}$ will be detected as an anomaly. So, a "1" produced by (11) indicates that the pixel $\mathbf{r}$ is detected as an anomaly; otherwise, it is considered as a background pixel.

In context of (11), we consider the Neyman Pearson detection theory for a binary hypothesis testing problem to perform signal detection [16], where $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ in (10) can be used as a Neyman Pearson detector to perform the ROC analysis as a performance evaluation tool. For example, for a particular threshold, a detection probability/power, $\mathrm{P_D}$ and a false

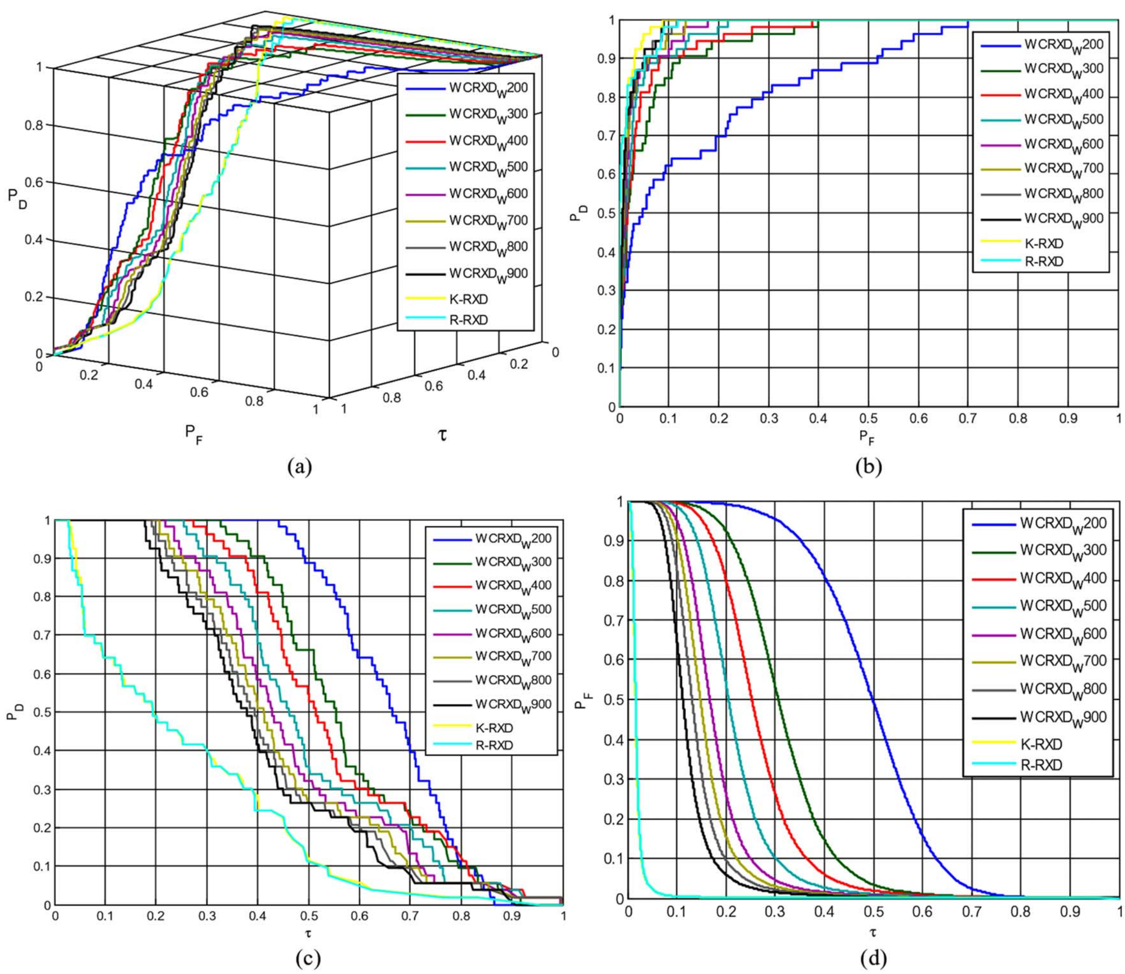

Fig. 10. 3-D ROC curve and its three corresponding 2-D ROC curves for HYDICE panel-vehicle scene. (a) 3-D ROC curve of $(\mathrm{P_D}, \mathrm{P_F}, \tau)$. (b) 2-D ROC curve of $(\mathrm{P_D}, \mathrm{P_F})$. (c) 2-D ROC curve of $(\mathrm{P_D}, \tau)$. (d) 2-D ROC curve of $(\mathrm{P_F}, \tau)$.

alarm probability, $\mathrm{P_F}$ can be calculated. By varying the threshold $\tau = a\%/100$ in (11), we can produce an ROC curve of $\mathrm{P_D}$ versus $\mathrm{P_F}$ and further calculate the area under the ROC curve for quantitative performance analysis. Interestingly, the threshold is absent in the traditional ROC curve. But according to (11) the values of $\mathrm{P_D}$ and $\mathrm{P_F}$ are actually calculated through the threshold τ. In order to address this issue, a 3-D ROC analysis was recently developed in [23], where a 3-D ROC curves can be generated by considering $\mathrm{P_D}$, $\mathrm{P_F}$, and τ as three parameters, each of which represents one dimension. In other words, a 3-D ROC curve is a three dimensional curve of $(\mathrm{P_D}, \mathrm{P_F}, \tau)$ from which three two-dimensional (2-D) ROC curves can be also generated, i.e., 2-D ROC of $(\mathrm{P_D}, \mathrm{P_F})$ which is the traditional ROC curve discussed in [16] along with two other new 2-D ROC curves, 2-D ROC curve of $(\mathrm{P_D}, \tau)$, and 2-D ROC curve of $(\mathrm{P_F},\tau)$.

There are advantages of using 3-D ROC analysis. First of all, it allows users to evaluate $\mathrm{P_D}$ versus τ independent of $\mathrm{P_F}$. Similarly, users can also use the 2-D ROC curve of $(\mathrm{P_F}, \tau)$ without referring to $\mathrm{P_D}$. Consequently, by varying the value of τ we are able to observe progressive changes in $\mathrm{P_D}$ and $\mathrm{P_F}$ individually, which the traditional 2-D ROC curve of $(\mathrm{P_D}, \mathrm{P_F})$ cannot offer. Second, in the traditional 2-D ROC curve of $(\mathrm{P_D}, \mathrm{P_F})$ $\mathrm{P_D}$ is expressed as a function $\mathrm{P_F}$. So, there is no direct information of $\mathrm{P_D}$ specified by the threshold τ. The 2-D ROC curve of $(\mathrm{P_D}, \tau)$ can profile progressive detection power as the threshold τ changes, Finally, the 2-D ROC curve of $(\mathrm{P_F}, \tau)$ actually provides crucial information of progressive background suppression as the threshold τ varies when it comes to interpretation of anomaly detection by visual inspection with no availability of ground truth. This issue was investigated in [26] and will be demonstrated in the following experiments.

Fig. 10 plots 3-D ROC curves along with their corresponding three 2-D ROC curves produced by the global, K-RXD, R-RXD anomaly detector, and causal local detectors using different causal sliding array window widths in Fig. 8 for the HYDICE panel-vehicle scene in Fig. 7(a). For a further quantitative analysis, the area under curve (AUC) are calculated, denoted by $\mathrm{A_z}$, for each of 2-D ROC curves produced in Fig. 10(b)–(d) by global and local anomaly detectors, and their results are tabulated in Table I, where the best results of causal local detectors are highlighted and the results of global anomaly detector K-RXD and R-RXD is also included for comparison. For 2-D ROC curves of $(\mathrm{P_D}, \mathrm{P_F})$ and $(\mathrm{P_D}, \tau)$, the higher the value of $\mathrm{A_z}$, the better is the detector. Conversely, for 2-D ROC curves of $(\mathrm{P_F}, \tau)$, the lower the value of $\mathrm{A_z}$, the better is the detector.

Based on the results in Figs. 10 and 11 and Table I, as the causal sliding array window width goes up, a higher $\mathrm{A_z}(\mathrm{P_D}, \mathrm{P_F})$, which is the traditional 2-D ROC analysis, is

TABLE I
VALUES OF THREE AREAS UNDER 2-D ROC CURVES A_z PRODUCED BY GLOBAL ANOMALY DETECTOR AND LOCAL CAUSAL ANOMALY DETECTOR WITH DIFFERENT SLIDING CAUSAL SLIDING ARRAY WINDOW WIDTH

Algorithm	K-RXD	R-RXD	CW = 200	CW = 300	CW = 400	CW = 500	CW = 600	CW = 700	CW = 800	CW = 900
Az (P_D,P_F)	0.990	0.985	0.850	0.948	0.959	0.969	0.975	0.978	0.981	0.983
Az o (P_D,•)	0.255	0.252	0.662	0.574	0.553	0.508	0.475	0.451	0.433	0.414
Az (P_F,•)	0.019	0.019	0.493	0.315	0.265	0.218	0.179	0.159	0.142	0.123

CW, Causal sliding array window width.

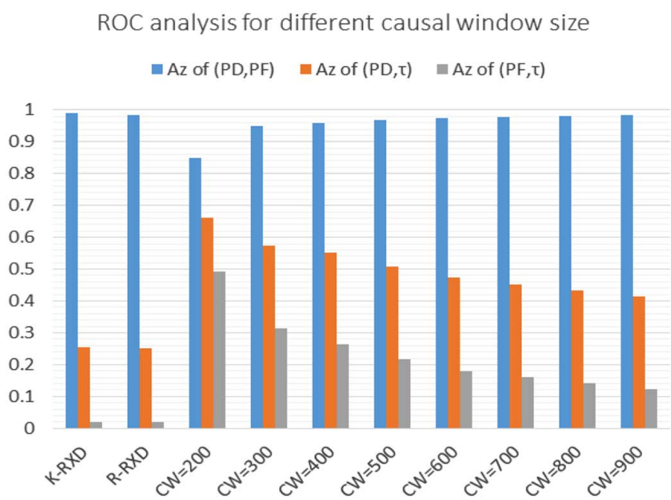

Fig. 11. Values of three areas under 2-D ROC curves using global detector and different sliding causal sliding array window widths.

obtained. This indicates a better detection power. Furthermore, a larger causal sliding array window width will also have a smaller $A_z(P_F, \tau)$, which indicates a better background suppression. However, a higher $A_z(P_D, P_F)$ does not necessarily imply a higher $A_z(P_D, \tau)$ as shown in Table I. Unfortunately, such two pieces of information are not provided by the traditional 2-D ROC analysis $A_z(P_D, P_F)$.

For a better representation of Table I and a better interpretation of Fig. 10, Fig. 11 further plots the results in Table I as histograms, where several conclusions can be made as follows.

1) The results showed that K-RXD and R-RXD performed nearly the same. There was no visible difference between these two.
2) From Fig. 11, the area under the 2-D ROC curve of detection power versus the threshold τ specified by (9), i.e., $A_Z(P_D, \tau)$ calculated from causal anomaly detectors with causal sliding array windows is always greater than that obtained by K-RXD and R-RXD. However, this was also traded for a higher value of $A_Z(P_F, \tau)$ as also shown in Fig. 11. By contrast, both K-RXD and R-RXD produced lowest values of $A_Z(P_D, \tau)$ and $A_Z(P_F, \tau)$.
3) According to Fig. 11, as causal sliding array window width W increased the value of $A_Z(P_D, P_F)$ also increased. On the other hand, as the causal sliding array window W increased the values of both $A_Z(P_D, \tau)$ and $A_Z(P_F, \tau)$ decreased. So, as W became very large and was close to the global window size, all the three values of $A_Z(P_D, P_F)$, $A_Z(P_D, \tau)$, and $A_Z(P_F, \tau)$ would converge to their corresponding values of K-RXD and R-RXD. This indicates that detection maps produced by a causal anomaly detector using various causal sliding array windows provide progressive anomaly detection maps of K-RXD and R-RXD as the causal sliding array window width W is progressively increased. Such progressive anomaly maps have been shown to be very valuable for visual inspection as they also provide progressive background suppression [26]. As an alternative interpretation, the progressive anomaly detection maps can be viewed as stage-by-stage slow motions of a detection map produced by a global anomaly detector. For example, detection maps in Fig. 8(a)–(h) and Fig. 9(a)–(h) can be considered as slow motions of the detection maps of Fig. 7(d) and (f) as the causal sliding array window size ω is slowly changing its size from 200 to 900.

Finally, we would like to point out that the experiments using the same HYDCE scene in Fig. 7(a) and (b) were conducted in detail in [18] for real-time causal anomaly detectors, CR-RXD and CK-RXD, without using causal sliding windows. It will be great beneficial if this paper is studied with [18] as a companion paper.

VII. CONCLUSION

While anomaly detection has been studied extensively in the literature, causal anomaly detection seems to receive little interest. For an anomaly detector to be implemented in real time, causality is a required process and must be included as a prerequisite to any anomaly detection real-time process [18]. This is particularly true for those adaptive or local anomaly detectors using sliding windows which are actually not causal. So, theoretically speaking they are not real-time anomaly detectors. This paper addresses this issue and further designs three types of causal sliding windows: 1) causal sliding square matrix window; 2) causal sliding rectangular matrix window; and 3) causal sliding array window. In order to implement causal anomaly detectors in real-time, recursive causal anomaly detectors are also developed for this purpose. As a result of real-time causal anomaly detection, progressive detection maps can be produced for visual assessment. In addition, causal anomaly detection also provides progressive background suppression that can be further used for image interpretation. In particular, there may be some weak anomalies detected earlier but later overwhelmed by subsequently detected strong anomalies. Under such circumstances, these weak anomalies will not be shown in the final detection maps but rather be captured in a certain stage during progressive anomaly detection; a fact was also demonstrated in [26].

Acknowledgment

The authors would like to thank anonymous reviewers for their very constructive and insightful comments, which lead to a new section (Section IV-C), and also new improvements in the presentation.

References

[1] C.-I Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Norwell, MA, USA: Kluwer/Plenum, 2003.

[2] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.

[3] C.-I Chang and S.-S. Chiang, "Anomaly detection and classification for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 2, pp. 1314–1325, Feb. 2002.

[4] C. M. Stellman *et al.*, "Real-time hyperspectral detection and cuing," *Opt. Eng.*, vol. 39, no. 7, pp. 1928–1935, Jul. 2000.

[5] Q. Du and H. Ren, "Real-time constrained linear discriminant analysis to target detection and classification in hyperspectral imagery," *Pattern Recognit.*, vol. 36, pp. 1–12, 2003.

[6] Q. Du and R. Nevovei, "Implementation of real-time constrained linear discriminant analysis to remote sensing image classification in hyperspectral imagery," *Pattern Recognit.*, vol. 38, no. 4, pp. 459–471, 2005.

[7] Q. Du and R. Nekovei, "Fast real-time onboard processing of hyperspectral imagery for detection and classification," *J. Real Time Image Process.*, vol. 4, no. 3, pp. 273–286, 2009.

[8] Y. Tarabalka, T. V. Haavardsholm, I. Kasen, and T. Skauli, "Real-time anomaly detection in hyperspectral images using multivariate normal mixture, pdels and GPU processing," *J. Real Time Image Process.*, vol. 4, no. 3, pp. 287–300, 2009.

[9] E. A. Ashton and A. Schaum, "Algorithms for the detection of subpixel targets in multispectral imagery," *Photogramm. Eng. Remote Sens.*, vol. 64, pp. 723–731, Jul. 1998.

[10] S.-S. Chiang, C.-I Chang, and I. W. Ginsberg, "Unsupervised subpixel target detection for hyperspectral images using projection pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1380–1391, Jul. 2001.

[11] D. W. Stein *et al.*, "Anomaly detection from hyperspectral imagery," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 58–69, Jan. 2002.

[12] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 29–43, Jan. 2002.

[13] H. Kwon, S. Z. Der, and N. M. Nasrabadi, "Adaptive anomaly detection using subspace separation for hyperspectral imagery," *Opt. Eng.*, vol. 42, no. 11, pp. 3342–3351, Nov. 2003.

[14] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 2, pp. 388–397, Feb. 2005.

[15] C.-I Chang and M. Hsueh, "Characterization of anomaly detection for hyperspectral imagery," *Sensor Rev.*, vol. 26, pp. 137–146, 2006.

[16] H. V. Poor, *An Introduction to Detection and Estimation*. Berlin, Germany: Springer-Verlag, 1994.

[17] C.-I Chang, H. Ren, and S. S. Chiang, "Real-time processing algorithms for target detection and classification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 4, pp. 760–768, Apr. 2001.

[18] S. Y. Chen, Y. Wang, C. C. Wu, C. Liu, and C.-I Chang, "Real time causal processing of anomaly detection in hyperspectral imagery," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1511–1534, Apr. 2014.

[19] Y. Wang, C. H. Zhao, and C.-I Chang, "Anomaly detection using sliding causal windows," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Quebec, Canada, Jul. 13–18, 2014, pp. 4600–4603.

[20] W. Liu and C.-I Chang, "Multiple window anomaly detection for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 664–658, Apr. 2013.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.

[22] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1980, p. 655.

[23] C.-I Chang, "Multiple-parameter receiver operating characteristic analysis for signal detection and classification," *IEEE Sensors J.*, vol. 10, no. 3, pp. 423–442, Mar. 2010.

[24] C.-I Chang, *Hyperspectral Data Processing: Algorithm Design and Analysis*. Hoboken, NJ, USA: Wiley, 2013, ch. 3.

[25] C.-I Chang and H. Ren, "An experiment-based quantitative and comparative analysis of hyperspectral target detection and image classification algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 2, pp. 1044–1063, Mar. 2000.

[26] Y. Wang, S. Y. Chen, C. Liu, and C.-I Chang, "Background suppression issues in anomaly detection for hyperspectral imagery," in *Proc. Satellite Data Compression Commun. Process. X (ST146) SPIE Int. Symp. SPIE Sens. Technol. Appl.*, Baltimore, MD, USA, May 2014, pp. 5–9.

Chein-I Chang (S'81–M'87–SM'92–F'10) received the B.S. degree in mathematics from Soochow University, Taipei, Taiwan, R.O.C, the M.S. degree in mathematics from the Institute of Mathematics, National Tsing Hua University, Hsinchu, Taiwan, R.O.C, the M.A. degree in mathematics from the State University of New York at Stony Brook, Stony Brook, NY, USA, the M.S. and M.S.E.E. degrees from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA.

Since 1987, he has been with the University of Maryland, Baltimore County (UMBC), and is currently a Professor with the Department of Computer Science and Electrical Engineering. Since 2014, he has been a Distinguished Chair Professor with the National Chung Hsing University, Taichung, Taiwan, R.O.C. From 1994 to 1995, he was a Visiting Research Specialist with the Institute of Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. Additionally, he was a Distinguished Lecturer Chair with the National Chung Hsing University, sponsored by the Ministry of Education, Taiwan, R.O.C., from 2005 to 2006. He was a Chair Professor with the Environmental Restoration and Disaster Reduction Research Center and the Department of Electrical Engineering, National Chung Hsing University, and has been a Chair Professor of Remote Sensing Technology with the National Chung Hsing University, since 2009. Currently, he is a Chair Professor with Providence University, Taichung, Taiwan, R.O.C., and an International Chair Professor of the National Taipei University of Technology, Taipei, Taiwan, R.O.C. From 2009 to 2010, he was a Distinguished Visiting Fellow/Fellow Professor sponsored by the National Science Council, Taiwan, R.O.C. He has five patents and several pending on hyperspectral image processing. He has authored two books: *Hyperspectral Imaging: Techniques for Spectral Detection and Classification* (Kluwer, 2003) and *Hyperspectral Data Processing: Signal Processing Algorithm Design and Analysis* (Wiley, 2013); edited two books: *Recent Advances in Hyperspectral Signal and Image Processing* (Transworld Research Network, India, 2006) and *Hyperspectral Data Exploitation: Theory and Applications* (Wiley, 2007); and co-edited with A. Plaza a book on *High Performance Computing in Remote Sensing* (CRC Press, 2007). He is currently working on a third book, *Real Time Hyperspectral Image Processing: Algorithm Architecture and Implementation* (Springer-Verlag, 2015). His research interests include multispectral/hyperspectral image processing, automatic target recognition, and medical imaging.

Dr. Chang is a Fellow of SPIE. He was a plenary speaker for *Society for Photo-optical Instrumentation Engineers (SPIE) Optics+Applications, Remote Sensing Symposium*, in 2009. In addition, he was also a keynote speaker at the *User Conference of Hyperspectral Imaging 2010*, December 30, 2010, Industrial Technology Research Institute, Hsinchu; *2009 Annual Meeting of the Radiological Society of the Republic of China*, 2009, Taichung; *2008 International Symposium on Spectral Sensing Research,* in 2008; and the *Conference on Computer Vision, Graphics, and Image Processing* 2003, Kimen and 2013, Nan-Tou, Taiwan. He was the Guest Editor of a special issue of the *Journal of High Speed Networks* on *Telemedicine and Applications* (April 2000) and Co-Guest Editor of another special issue of the same journal on *Broadband Multimedia Sensor Networks in Healthcare Applications* in April 2007. He is also the Coguest Editor of a special issue on High Performance Computing of Hyperspectral Imaging for the *International Journal of High Performance Computing Applications,* in December 2007 and of a special issue on *Signal Processing and System Design in Health Care Applications* for the *EURASIP Journal on Advances in Signal Processing,* in 2009. From 2001 to 2007, he was an Associate Editor in the area of hyperspectral signal processing for IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. Currently, he is an Associate Editor of *Artificial Intelligence Research* and also on the editorial boards of *Journal of High Speed Networks, Recent Patents on Mechanical Engineering, International Journal of Computational Sciences and Engineering, Journal of Robotics and Artificial Intelligence*, and *Open Remote Sensing Journal*. He received a National Research Council Senior Research Associateship Award from 2002 to 2003 sponsored by the U.S. Army Soldier and Biological Chemical Command, Edgewood Chemical and Biological Center, Aberdeen Proving Ground, MD, USA.

Yulei Wang is currently pursuing the Ph.D. degree at the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China. She is awarded by China Scholarship Council in 2011 as a Visiting Scholar to study in Remote Sensing Signal and Image Processing Laboratory, University of Maryland, Baltimore, MD, USA, from 2011 to 2012.

Her research interests include hyperspectral image processing and vital signs signal processing.

Shih-Yu Chen (M'15) received the B.S. degree in electrical engineering from Da-Yeh University, Changhua, Taiwan, R.O.C., in 2005, the M.S. degree in electrical engineering from National Chung Hsing University, Taichung, R.O.C., in 2010, and the Ph.D. degree in electrical engineering from the University of Maryland, (UMBC), Baltimore, MD, USA, in 2014.

Currently, he is an Assistant Professor with the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan, R.O.C. His research interests include multi/hyperspectral image processing, medical signal and image processing, and vital sign signal processing.