Real-Time Causal Processing of Anomaly Detection for Hyperspectral Imagery

SHIH-YU CHEN, Student Member, IEEE
University of Maryland, Baltimore County

YULEI WANG
Harbin Engineering University
China
University of Maryland, Baltimore County

CHAO-CHENG WU, Member, IEEE
National Taipei University of Technology
Taiwan

CHUNHONG LIU
China Agriculture University
Beijing

CHEIN-I CHANG, Fellow, IEEE
University of Maryland, Baltimore County
Providence University
Taiwan
National Chung Hsing University
Taiwan

Anomaly detection generally requires real-time processing to find targets on a timely basis. However, for an algorithm to be implemented in real time, the used data samples can be only those up to the data sample being visited; no future data samples should be involved in the data processing. Such a property is generally called causality, which has unfortunately received little interest thus far in real-time hyperspectral data processing. This paper develops causal processing to perform anomaly detection that can be also implemented in real time. The ability of real-time causal processing is derived from the concept of innovations used to derive a Kalman filter via a recursive causal update equation. Specifically, two commonly used anomaly detectors, sample covariance matrix (K)–based Reed-Xiaoli detector (RXD), called K-RXD, and sample correlation matrix (R)–based RXD, called R-RXD, are derived for their real-time causal processing versions. To substantiate their utility in applications of anomaly detection, real image data sets are conducted for experiments.

I. INTRODUCTION

Because of availability of very high spectral resolution, a hyperspectral imaging sensor is now capable of uncovering many subtle signal sources that cannot be known by prior knowledge or be visually inspected by image analysts [1]. Many such signal sources generally appear as anomalies in the data. As a result, anomaly detection has received considerable interest in hyperspectral imaging in recent years [1–15]. While a cut-and-dried definition of anomaly may not be possible [9], a consensus is that an anomaly should be a target whose presence cannot be known before data processing but that can be characterized by several unique features: (1) unexpected presence, (2) low probability of occurrence, (3) relatively small sample population, and (4) most importantly, a signature spectrally distinct from spectral signatures of its surrounding data samples. Targets with these properties include endmembers defined as pure signatures to specify spectral classes, special species in agriculture and ecology, rare mineral in geology, toxic wastes in environmental monitoring, oil spills in water pollution, drug and smuggler trafficking in law enforcement, man-made objects in battlefields, unusual terrorism activities in intelligence gathering, and tumors in medical imaging. To effectively detect such targets, an algorithm developed by Reed and Yu [2], called the Reed-Xiaoli detector (RXD), has been widely used for this purpose. Since then, many RXD-like anomaly detectors have been proposed [1, 3, 9–13]. Of particular interest are anomaly detectors that modify RXD by replacing the global sample covariance matrix $\mathbf{K}$ with the global sample correlation matrix $\mathbf{R}$. The resulting RXD using matrix $\mathbf{R}$ is called R-RXD, while the RXD using matrix $\mathbf{K}$ is denoted as K-RXD for distinction. R-RXD was further used as a base to develop a causal version of R-RXD (CR-RXD) in [1, 9, 12], which implements R-RXD using a so-called causal sample correlation matrix $\mathbf{R}(n)$ formed by only data sample vectors $\{\mathbf{r}_i\}_{i=1}^{n-1}$ up to the data sample vector $\mathbf{r}_n$ being processed and $\mathbf{r}_n$. This CR-RXD is then used to derive a causal version of K-RXD (CK-RXD). One of most important applications for anomaly detection is detection of moving unknown

Manuscript received January 31, 2013; revised June 3, 2013; released for publication September 6, 2013.

DOI. No. 10.1109/TAES.2014.130065.

Refereeing of this contribution was handled by H. Kwon.

The work of Y. Wang is supported by the National Natural Science Foundation of China (Grants No. 61077079 and No. 60802059), the Ph.D. Programs Foundation of Ministry of Education of China (Grant No. 20102304110013), the Key Program of Heilongjiang Natural Science Foundation (Grant No. ZD201216), and Fundamental Research Funds for the Central Universities (Grant No. HEUCF1208). The work of C. C. Wu is supported by National Science Council in Taiwan (Grant No. 102-2221-E-027 -034). The work of C. Liu was supported by Youth National Natural Science Foundation of China (Grant No. 61201415).

Authors' addresses: S. Y. Chen, Y. Wang, C.-I Chang, Remote Sensing Signal and Image Processing Laboratory, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250; C. C. Wu, Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, Republic of China; Corresponding author: C. Liu, Information and Communication Engineering College, China Agriculture University, Beijing, China, E-mail: (sophia31314209@qq.com).

targets in real time to locate these targets before they disappear, or are compromised by background, or are dominated by other signal sources. Both CR-RXD and CK-RXD pave a way for real-time causal versions of anomaly detectors to be designed later in this paper.

Although many real-time processing algorithms have been proposed for target detection and classification in the literature [16–20], none of reports in [17–20] are technically true real-time processing algorithms; rather, they are fast computational algorithms. Theoretically, a true real-time processing algorithm must produce its output at the same time as an input comes in. In reality, this is impossible, because there is always a time delay caused by data processing. With this interpretation, all claimed real-time processing algorithms can be only called near real time with an assumption that the data processing time is negligible, such as constrained linear discriminant analysis developed in [17–19] or parallel processing algorithms, such as anomaly detection using a multivariate normal mixture model and a graphics processing unit [20]. Nevertheless, from a practical point of view, such a time delay is determined by a specific application. For example, in surveillance and reconnaissance applications, finding moving targets such as missiles is imminent and the responding time must be instantaneous. In this case, little time delay should be allowed. As another example, for applications in fire damage management and assessment, the time to respond can be minutes or hours, in which case the allowable time delay can be longer. So, as long as an algorithm can meet a required time constraint, it can be considered a real-time processing algorithm.

In anomaly detection, real-time causal processing is particularly crucial. First, it saves tremendous cost and payload in data storage and archiving in data communication, specifically in satellite communication. Second, it meets the constraint of available limited bandwidth. Third, it achieves data compression while the data are being processed. Fourth, it can detect anomalies, such as moving targets, which may not stay long enough and have duration of their presence that is very short. In many cases, they may show up suddenly and instantly and then vanish quickly afterward. Finally, and most importantly, real-time processing allows users to see progressive background suppression, which cannot be accomplished by traditional one-shot operation anomaly detectors. This is crucial in anomaly detection because no prior knowledge is available for anomaly detection and progressive background suppression provides an opportunity for seeing how anomalies are detected in real time as the detection process moves along. It is also particularly critical when weak anomalies are detected and may be overwhelmed by subsequently detected strong anomalies. Therefore, for an algorithm to be able to detect these targets on a timely basis, the process must be in real time. In the meantime, the data that can be used for real-time data processing should be only those that already have been visited and processed. Accordingly, an anomaly detection process should be carried out causally. However, because of the nature of anomaly detection, an anomaly generally has distinct spectral properties from those in its surrounding neighborhood. To capture such distinct spectral characteristics, an anomaly detector generally requires intersample statistics, such as sample correlation, and covariance statistics, such as using a sliding window centered on the data sample being processed. This requirement makes anomaly detection inapplicable to causal processing because it must calculate sample covariance or correlation statistics from the entire set of data sample vectors or the samples within a used window, which cannot be calculated causally. For example, K-RXD uses the global sample covariance matrix $\mathbf{K}$, which needs to be calculated from the global sample mean of all data sample vectors. This cannot be done without full access to the entire data set, and it must be done before anomaly detection. Therefore, from an algorithmic implementation point of view, K-RXD is neither a causal processing algorithm nor a real-time processing algorithm. To resolve this issue, CR-RXD, proposed in [9, 12], suggested the use of the sample correlation matrix $\mathbf{R}$ to replace the sample covariance matrix in K-RXD so that CR-RXD can be implemented via a QR decomposition in real time [16]. However, neither this approach nor [17–20] addressed the issue of causality in real-time implementation. Interestingly, on one end, a causal processing does not have to be real time. On the other end, real-time processing must be causal. Unfortunately, this causal issue has never been addressed in the many reported real-time processing algorithms. This paper takes a different approach to design and development of “causal processing” for real-time implementation via the concept of innovation information discussed in [21, footnote pp. 78], which is defined as the new information of the current input sample that cannot be predicted from the past. It was originally proposed by Kailath [22] to develop a Kalman filter and has been shown to be a promising and effective means of updating data in a causal and real time manner. Two commonly used anomaly detectors, K-RXD and CR-RXD, are selected for investigation, because many existing anomaly detectors are variants of one of them.

The idea of the proposed causal processing arises in updating needed information only through the data sample vector being processed and the information generated by processing previous data sample vectors. Although it is similar to Kalman filtering, there are several notable differences between these two. First, because anomaly detection is performed on a single data sample basis, there is no counterpart of a state equation used by a Kalman filter corresponding to anomaly detection. Second, the measurement equation used by an anomaly detector is quite different from that used by a Kalman filter, where a noise term involved in a Kalman filter is not present in an anomaly detector. Third, an anomaly detector usually requires inversion of a sample correlation and covariance matrix. To implement anomaly detection in real time, the matrix inversion must be updated sample by sample. Such an update is not found in the measurement equation in a

Kalman filter, which is updated from the state equation. So, direct application of real-time Kalman filtering to anomaly detection is not feasible because of the lack of a state equation that can be derived for an anomaly detector. To resolve this issue, an alternative approach is to use Woodbury's identity [23] to derive causal innovation information update equations for CK-RXD and CR-RXD in the same way used to derive a Kalman filter. With this approach, only the data sample vector being processed and the processed information obtained by previous data sample vectors can be used to generate the innovation information and then update data processing sample by sample.

II. ANOMALY DETECTION

One of most widely used anomaly detectors is K-RXD, the algorithm developed by Reed and Yu in [2]. K-RXD uses the global sample covariance matrix $\mathbf{K}$ to account for spectral statistics among data sample vectors. Since its development, many various K-RXD–like anomaly detectors have been proposed. Among them, CR-RXD, developed in [9], is of particular interest. In what follows, we describe these two anomaly detectors. Assume that $\{\mathbf{r}_i\}_{i=1}^N$, where N is the total number of entire data sample vectors in the data and $\mathbf{r}_i = (r_{i1}, r_{i2}, \cdots, r_{iL})^T$ is the ith data sample vector, where L is the total number of spectral bands.

A. K-RXD

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 sample mean and $\mathbf{K}$ is the sample data covariance matrix. The form of $\delta^{\text{K-RXD}}(\mathbf{r})$ in (1) is the well-known Mahalanobis distance. However, from a detection point of view, the use of $\mathbf{K}^{-1}$ can be interpreted as a whitening process to suppress image background.

B. CR-RXD

Let $\{\mathbf{r}_i\}_{i=1}^N$ be a set of data sample vectors to be processed. CR-RXD, denoted by $\delta^{\text{CR-RXD}}(\mathbf{r})$, is specified by

$$\delta^{\text{CR-RXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \mathbf{R}(n)^{-1} \mathbf{r}_n \tag{2}$$

where $\mathbf{r}_n$ is the nth data sample vector being processed and $\mathbf{R}(n)$ is the sample data autocorrelation matrix formed by $\mathbf{R}(n) = (1/n) \sum_{i=1}^n \mathbf{r}_i \mathbf{r}_i^T$. Here, $\mathbf{r}_i = (r_{i1}, r_{i2}, \cdots, r_{iL})^T$ is the ith data sample vector, where L is the total number of spectral bands. Because of (2), CK-RXD in (1) can be reexpressed as

$$\delta^{\text{CK-RXD}}(\mathbf{r}_n) = (\mathbf{r}_n - \boldsymbol{\mu}(n))^T \mathbf{K}(n)^{-1} (\mathbf{r}_n - \boldsymbol{\mu}(n)) \tag{3}$$

where $\boldsymbol{\mu}(n) = (1/n) \sum_{i=1}^n \mathbf{r}_i$ is the causal sample mean averaged over all data sample vectors $\{\mathbf{r}_i\}_{i=1}^n$ and $\mathbf{K}(n) = (1/n) \sum_{i=1}^n (\mathbf{r}_i - \boldsymbol{\mu}(n)) (\mathbf{r}_i - \boldsymbol{\mu}(n))^T$ is the causal covariance matrix formed by the data sample vectors $\{\mathbf{r}_i\}_{i=1}^n$. In light of (3), K-RXD can be considered a special case of CK-RXD, but the two detectors are identical only when both detectors reach the last data sample vector $\mathbf{r}_N$. That is, K-RXD in (1) can be reexpressed as $\delta^{\text{K-RXD}}(\mathbf{r}_n) = (\mathbf{r}_n - \boldsymbol{\mu}(N))^T \mathbf{K}(N)^{-1} (\mathbf{r}_n - \boldsymbol{\mu}(N))$, where $\boldsymbol{\mu}(N) = (1/N) \sum_{i=1}^N \mathbf{r}_i$ is the global sample mean averaged over all data sample vectors $\{\mathbf{r}_i\}_{i=1}^N$ and $\mathbf{K}(N) = (1/N) \sum_{i=1}^N (\mathbf{r}_i - \boldsymbol{\mu}(N)) (\mathbf{r}_i - \boldsymbol{\mu}(N))^T$ is the global covariance matrix formed by all data sample vectors $\{\mathbf{r}_i\}_{i=1}^N$.

If the $\mathbf{r}_n^T$ in (2) is replaced by $\left(\mathbf{d}/\mathbf{d}^T \mathbf{R}^{-1}(n) \mathbf{d}\right)^T$, where the $\mathbf{d}$ is the desired signature to be detected, (2) becomes a causal version of a well-known subpixel detector, called constrained energy minimization (CEM) in [1, 24].

III. CAUSAL PROCESSING OF ANOMALY DETECTION

Once CR-RXD and CK-RXD are specified by (2) and (3), a follow-up task is to implement these causal anomaly detectors sample by sample so as to achieve real-time processing. The following Woodbury's matrix identity [23]

$$\left[\mathbf{A} + \mathbf{u}\mathbf{v}^T\right]^{-1} = \mathbf{A}^{-1} - \frac{\left[\mathbf{A}^{-1}\mathbf{u}\right]\left[\mathbf{v}^T\mathbf{A}^{-1}\right]}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \tag{4}$$

is helpful.

A. Real-Time CR-RXD

Because 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 reexpressed as $\mathbf{R}^{-1}(n) = [((n-1)/n) \mathbf{R}(n-1) + (1/n) \mathbf{r}_n \mathbf{r}_n^T]^{-1}$. Because of (4), we can derive a real-time causal version of R-RXD (RT-CR-RXD), $\delta^{\text{RT-CR-RXD}}(\mathbf{r}_n)$, as follows: The innovation information can be obtained by a causal innovation information update equation by dictating the difference between the pixel $\mathbf{r}_n$ being processed and the processed information obtained by previous $n-1$ data sample vectors $\{\mathbf{r}_i\}_{i=1}^{n-1}$, which is $\mathbf{R}^{-1}(n-1)$. This is the information contained in $\mathbf{r}_n$ but that cannot be obtained and predicted from previously visited data sample vectors $\{\mathbf{r}_i\}_{i=1}^{n-1}$. Now we rewrite $\mathbf{R}^{-1}(n) = \left[((n-1)/n) \mathbf{R}(n-1) + (1/n) \mathbf{r}_n \mathbf{r}_n^T\right]^{-1}$ and use (4) by letting $\mathbf{A} = ((n-1)/n) \mathbf{R}(n-1)$ and $\mathbf{u} = \mathbf{v} = \left(1/\sqrt{n}\right) \mathbf{r}_n$. Then, $\mathbf{R}^{-1}(n)$ can be expressed as

$$\mathbf{R}^{-1}(n) = [(1-1/n)\mathbf{R}(n-1)]^{-1} - \frac{\left\{[(1-1/n)\mathbf{R}(n-1)]^{-1}\left(1/\sqrt{n}\right)\mathbf{r}_n\right\}\left\{\left(1/\sqrt{n}\right)\mathbf{r}_n^T[(1-1/n)\mathbf{R}(n-1)]^{-1}\right\}}{1 + \left(1/\sqrt{n}\right)\mathbf{r}_n^T[(1-1/n)\mathbf{R}(n-1)]^{-1}\left(1/\sqrt{n}\right)\mathbf{r}_n} \tag{5}$$

Using (2) in conjunction with (5), $\delta^{\text{RT-CR-RXD}}(\mathbf{r}_n)$ can be derived as follows, where the first term is the result of combining the information $\mathbf{R}^{-1}(n-1)$ obtained by previous data samples $\{\mathbf{r}_i\}_{i=1}^{n-1}$ and current data sample $\mathbf{r}_n$ and the second term is new information, considered innovation information, obtained by correlating $\mathbf{R}^{-1}(n-1)$ and $\mathbf{r}_n$, which cannot be predicted from $\mathbf{R}^{-1}(n-1)$ or from $\mathbf{r}_n$ alone:

$$\delta^{\text{RT-CR-RXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \mathbf{R}^{-1}(n)\mathbf{r}_n$$
$$= \mathbf{r}_n^T \left[(1-1/n)\,\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n - \mathbf{r}_n^T \frac{\left\{\left[(1-1/n)\,\mathbf{R}(n-1)\right]^{-1}\left(1/\sqrt{n}\right)\mathbf{r}_n\right\}\left\{\left(1/\sqrt{n}\right)\mathbf{r}_n^T\left[((1-1/n)\,\mathbf{R}(n-1)\right]^{-1}\right\}}{1+\left(1/\sqrt{n}\right)\mathbf{r}_n^T\left[(1-1/n)\,\mathbf{R}(n-1)\right]^{-1}\left(1/\sqrt{n}\right)\mathbf{r}_n}\mathbf{r}_n$$
$$= (1-1/n)^{-1}\,\mathbf{r}_n^T\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n - \frac{\left((1-1/n)\sqrt{n}\right)^{-2}\mathbf{r}_n^T\left\{\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\left[\mathbf{R}(n-1)\right]^{-1}\right\}\mathbf{r}_n}{1+((1-1/n)n)^{-1}\,\mathbf{r}_n^T\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n} \tag{6}$$

If we further let $\mathbf{A} = ((n-1)/n)\,\mathbf{R}(n-1) \equiv \tilde{\mathbf{R}}(n)$ and $\mathbf{u} = \mathbf{v} = \left(1/\sqrt{n}\right)\mathbf{r}_n \equiv \tilde{\mathbf{r}}_n$ in (4), where $\tilde{\mathbf{R}}^{-1}(n-1) = ((1-1/n)\,\mathbf{R}(n-1))^{-1}$ and $\tilde{\mathbf{r}}_n = \left(1/\sqrt{n}\right)\mathbf{r}_n$, Fig. 1 depicts a flow chart of implementing (5), where $\tilde{\mathbf{R}}^{-1}(n-1) = (1-1/n)^{-1}\,\mathbf{R}^{-1}(n-1)$, $\tilde{\mathbf{r}}_n = \left(1/\sqrt{n}\right)\mathbf{r}_n$, and D is one time unit delay. In particular, it shows that how $\tilde{\mathbf{R}}(n)$ can be updated recursively by previously calculated $\tilde{\mathbf{R}}(n-1)$ and the data sample vector $\tilde{\mathbf{r}}_n$ being processed via (5).

According to (6), there are three types of information involved with calculation. One type is new information provided by new incoming data sample vector $\mathbf{r}_n$. Another type is processed information generated by processing all previous data sample vectors to obtain $\mathbf{R}^{-1}(n-1)$. A third type of information is called innovation information and is obtained by the correlation information between the pixel $\mathbf{r}_n$ being processed and the processed information $\mathbf{R}^{-1}(n-1)$, which is a part of the information of $\mathbf{r}_n$ but cannot be predicted from previous data samples $\{\mathbf{r}_i\}_{i=1}^{n-1}$. This information is then used to generate required information: $\mathbf{r}_n^T\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n$ and $\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n$ or $\mathbf{r}_n^T\left[\mathbf{R}(n-1)\right]^{-1}$. The innovation information is vital, because it is the only piece of information that correlates new and processed information and can be used to calculate (6). Thus, (6) is called the causal innovation information update equation. Also, in light of (6), we only need to calculate the innovation information of $\mathbf{r}_n^T\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n$ and $\left[\mathbf{R}(n-1)\right]^{-1}\mathbf{r}_n$ to update $\mathbf{R}^{-1}(n)$ recursively—without reprocessing previously visited data sample vectors $\{\mathbf{r}_i\}_{i=1}^{n-1}$. As a result, one immediate benefit resulting from (6) is that data sample vectors can be processed in real time. This recursive equation is similar to two recursive equations implemented by a Kalman filter [22].

B. Real-Time CK-RXD

Unlike CR-RXD, which uses the sample autocorrelation matrix without calculating the sample mean, finding a real-time causal version of K-RXD (RT-CK-RXD) is not trivial. It requires the causal sample mean before calculating the causal sample covariance matrix. From the derivations given in the appendix for causal processing of K-RXD, RT-CK-RXD derived in (15) implements a causal innovation information update equation similar to (6) for RT-CR-RXD. It can be carried out sample by sample as follows:

$$\mathbf{r}_n - \boldsymbol{\mu}(n) = \mathbf{r}_n - ((n-1)/n)\,(1/(n-1))\sum_{i=1}^{n-1}\mathbf{r}_i - (1/n)\,\mathbf{r}_n = \mathbf{r}_n - (1-1/n)\,\boldsymbol{\mu}(n-1) - (1/n)\,\mathbf{r}_n \tag{7}$$

$$\delta^{\text{RT-CK-RXD}}(\mathbf{r}_n) = (\mathbf{r}_n - \boldsymbol{\mu}(n))^T\,\mathbf{K}^{-1}(n)\,(\mathbf{r}_n - \boldsymbol{\mu}(n))$$
$$= (1-1/n)^{-1}\,(\mathbf{r}_n - \boldsymbol{\mu}(n))^T\left[\mathbf{K}(n-1)\right]^{-1}(\mathbf{r}_n - \boldsymbol{\mu}(n))^T$$
$$-\frac{(1-1/n)^{-2}\left(\sqrt{n-1}/n\right)^2(\mathbf{r}_n-\boldsymbol{\mu}(n))^T\left[\{\mathbf{K}(n-1)\}^{-1}(\mathbf{r}_n-\boldsymbol{\mu}(n-1))\right]\left[(\mathbf{r}_n-\boldsymbol{\mu}(n-1)^T\{\mathbf{K}(n-1)\}^{-1}\right](\mathbf{r}_n-\boldsymbol{\mu}(n))}{1+(1-1/n)^{-1}\left(\sqrt{n-1}/n\right)^2\left[(\mathbf{r}_n-\boldsymbol{\mu}(n-1)^T\{\mathbf{K}(n-1)\}^{-1}(\mathbf{r}_n-\boldsymbol{\mu}(n-1)\right]} \tag{8}$$

From (7) and (8), the $\mathbf{K}^{-1}(n)$ used in RT-CK-RXD, $\delta^{\text{RT-CR-RXD}}(\mathbf{r}_n)$, can be easily updated by $\mathbf{K}^{-1}(n-1)$, $\boldsymbol{\mu}(n-1)$, as well as by the current input data sample vector $\mathbf{r}_n$, with no inverse required to calculate once the initial calculation of $\mathbf{K}^{-1}(1)$ is done. In addition, there are no similar derivations to (7) and (8) derived in [19], because all classifiers in [19] used the sample correlation matrix $\mathbf{R}$, not the sample covariance matrix $\mathbf{K}$.

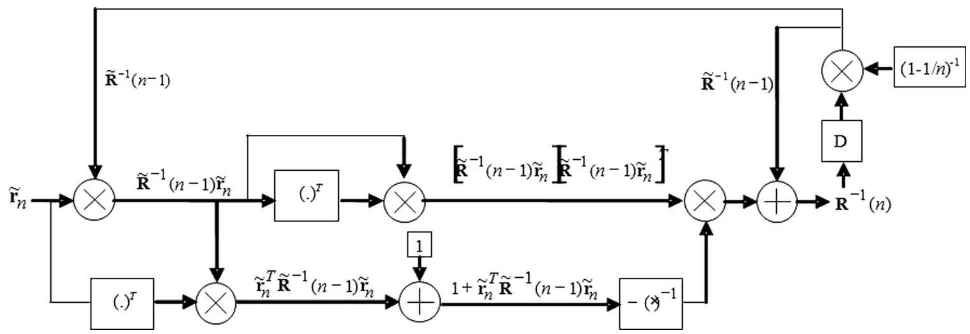


Fig. 1. Flow chart of implementing recursive equation specified by (6) to find $\mathbf{R}^{-1}(n)$.

Even though Woodbury's identity is not new and has been widely used in the literature, the use of this identity to drive a Kalman-like recursive equation is new. According to [25, pp. 25–26], there are three ways to acquire remotely sensed images: band sequential (BSQ), band interleaved by line, and band interleaved by sample (BIS), also called band interleaved by pixel (BIP). In this paper, the use of Woodbury's identity is suitable for hyperspectral imagery acquired by the BIS (BIP) format. Recently, another identity in [26, (12.25)] was used to derive real-time progressive band processing of anomaly detection for hyperspectral imagery that was acquired by the BSQ format in [25].

IV. COMPUTATIONAL COMPLEXITY

This section provides a detailed analysis on the computational complexity of calculating (6) or (8). Because the computational complexity of (8) is the same as that of (6), only (6) is discussed.

First, to avoid a singularity problem, the initial condition of calculating $\mathbf{R}^{-1}(n)$ in (6) should collect a sufficient number of data sample vectors to ensure the sample correlation and covariance matrix is of full rank. In this case, the causal processing must begin with the Lth data sample vector $\mathbf{r}_L$, where L is the total number of spectral bands. Also, according to (6), the $\mathbf{R}^{-1}(n)$ in RT-CR-RXD, $\delta^{\text{RT-CR-RXD}}(\mathbf{r}_n)$, can be easily updated by the previously processed information $\mathbf{R}^{-1}(n-1)$ and the innovation information provided by $\mathbf{r}_n$, without calculating the matrix inverse, once $\mathbf{R}^{-1}(L)$ is initially calculated. In other words, we only need to calculate a matrix inverse once, which is $\mathbf{R}^{-1}(L)$. However, in causal processing, $\mathbf{R}^{-1}(n)$ must be recalculated to include each incoming data sample vector $\mathbf{r}_n$.

In causal processing, there are three major operations involved for each data sample vector. One is to calculate the outer product (OP) of the incoming data sample vector $\mathbf{r}_n$, which is $\mathbf{r}_n\mathbf{r}_n^T$, to update $\mathbf{R}^{-1}(n-1)$ to $\mathbf{R}^{-1}(n)$, which requires L^2 multiplications. Another is to calculate the inverse of $\mathbf{R}^{-1}(n)$ with size L by L. The third operation is to calculate $\mathbf{r}_n^T\,[\mathbf{R}(n)]^{-1}\,\mathbf{r}_n$ which requires $(L+1)$ inner products (IPs), each of which requires L multiplications. As a result, the computational complexity of finding $\mathbf{R}^{-1}(n)$ is the same as that required for $\mathbf{R}^{-1}(L)$ and is constant for all n.

However, real-time causal processing using (6) requires three operations of calculating $\mathbf{x}^T\mathbf{A}\mathbf{y}$ (i.e., two calculations of $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$ and one calculation of $\mathbf{r}_n^T\left\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}\mathbf{r}_n$), each of which requires $L+1$ IPs, which result in $L(L+1)$ multiplications, and one OP $\left\{[\mathbf{R}(n-1)]^{-1}\mathbf{r}_n\right\}\left\{[\mathbf{R}(n-1)]^{-1}\mathbf{r}_n\right\}^T$, which requires L^2 multiplications. More specifically, at the nth data sample vector $\mathbf{r}_n$ with $n \geq L+1$, the calculation of $\mathbf{R}^{-1}(n)$ requires calculations of L-dimensional IPs, i.e., $[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$ and $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$, which result in L^2 multiplications, and one OP $\mathbf{O}(n-1) = \left\{[\mathbf{R}(n-1)]^{-1}\mathbf{r}_n\right\}\left\{[\mathbf{R}(n-1)]^{-1}\mathbf{r}_n\right\}^T$, which carries out L^2 multiplications, plus another L-dimensional IPs to calculate $\mathbf{r}_n^T\mathbf{O}(n-1)\mathbf{r}_n$, which amounts to L multiplications. As a result, a total number of two L-dimensional scalar IPs and one matrix OP are required and give rise to L^2+L multiplications, which is the same as the one derived in [19]. But the entire complexity to carry out a real-time causal anomaly detector in (6) is actually $3(L^2+L)+L^2$ multiplications plus the complexity of calculating the initial condition $\mathbf{R}^{-1}(L)/\mathbf{K}^{-1}(L)$.

According to the preceding analysis, the computational complexity of causal and real-time anomaly detectors is determined by the computer processing time (CPT) required by calculating three elements: an inverse of an $L \times L$ matrix, an IP of two L-dimensional vectors, and an OP of two L-dimensional vectors. Most importantly, in both causal processing and real causal processing, the computational complexity is independent of the data sample vectors to be processed and is linearly increased with the number of data sample vectors. A detailed study on computational complexity is provided in Table I, which tabulates the computational complexity of CR-RXD and CK-RXD and of RT-CR-RXD and RT-CK-RXD in terms of required number of multiplications $\times$ used for the calculation, where the constant c is included to account for multiplications used to balance the normalization constants n and $n-1$ in calculating $\mathbf{R}(n)/\mathbf{K}(n)$. For example, for CR-RXD and CK-RXD, $c=2$ to account for two multiplications used to scale $(1/n)\mathbf{r}_n\mathbf{r}_n^T$ and balance

TABLE I
Comparative Analysis on Computational Complexity Between CR-RXD and CK-RXD and Between RT-CR-RXD and RT-CK-RXD

	CR-RXD/CK-RXD	RT-CR-RXD/RT-CK-RXD
Initial condition	N/A	$\mathbf{R}^{-1}(L)/\mathbf{K}^{-1}(L)$
Input $\mathbf{r}_n$	$\mathbf{R}^{-1}(n)/\mathbf{K}^{-1}(n)$ for each input sample $\mathbf{r}_n$	N/A
No. IPs/sample	$L+1$	$3(L+1)$
No. OPs/sample	L (one for $(\mathbf{r}_n(\mathbf{r}_n)^T)$	L (one for $(\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\}\,\{\mathbf{r}\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\}^T)$
No. $\times$/sample	$[(L+1)L+L^2=2L^2+L+c]/2$	$[3(L+1)L+L^2=4L^2+3L+c]/2$
No. $\times$/data set	$(N-L)(2L^2+L+c)/2$	$(N-L)(4L^2+3L+c)/2$
Complexity	$O(N)$	$O(N)$

N/A = not applicable.

$[(n-1)/n]\mathbf{R}^{-1}(n)/\mathbf{K}^{-1}(n)$. In addition, in the second column under a causal anomaly detector, an additional number of L^2 multiplications is required to calculate $\mathbf{r}_n\mathbf{r}_n^T$ and thus update the sample correlation and covariance matrix $\mathbf{R}(n)/\mathbf{K}(n)$ plus the processing time of inverting $\mathbf{R}(n)/\mathbf{K}(n)$ of a $L \times L$ matrix size as n varies. Neither is needed if a real-time causal anomaly detector is used via (6) or (8). Finally, the number of multiplications in Table I can be reduced roughly by half because of the symmetry of the matrices.

As noted in Table I, a real-time causal anomaly detector using the causal innovation information update equation specified by (6) or (8) only requires L-dimensional scalar IPs and OPs plus the calculation of the initial condition of matrix inverse $\mathbf{R}^{-1}(L)$.

A similar form to (6) was independently derived for the BIP format in [19]. Several significant differences need to be mentioned. First, Woodbury's identity in (4) to calculate $\mathbf{R}^{-1}(n)$ is different from that used in [19]. Second, (6) is specifically derived from a particular anomaly detector, while the one in [19] was derived for the sample correlation matrix $\mathbf{R}$ only as part of the operation of implementing classifiers considered in [19].The same derivation of (6) can be applied to their classifiers. Third, there is a lack of computational complexity analysis in [19]. In addition, the computations in our analysis are based on the number of IPs of two vectors and matrix OPs of a single vector instead of multiplications, where each L-dimensional IP performs L^2 multiplications and each matrix output of an L-dimensional vector also requires L^2 multiplications.

V. SYNTHETIC IMAGE EXPERIMENTS

The goal of using synthetic images for experiments is to conduct a detailed comparative analysis between causal processing with or without real-time processing on detection performance and the CPT, where the ground truth can provide accurate assessment. These synthetic images were previously designed in [27] and have been widely used in many research efforts, such as [28, 29]. The image data used to design synthetic images is a real Cuprite image scene shown in Fig. 2(a), which is available at the U.S. Geological Survey website [30]. It is a 224-band image with a size of 350 × 350 pixels and was

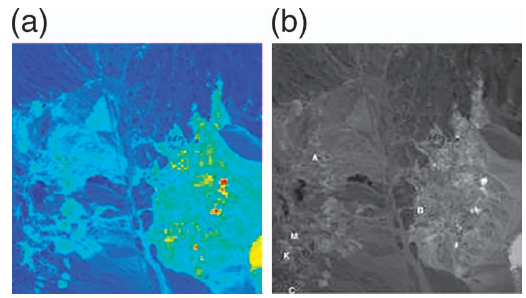


Fig. 2. (a) Cuprite AVIRIS image scene and (b) spatial positions of five pure pixels corresponding to minerals A, B, C, K, and M.

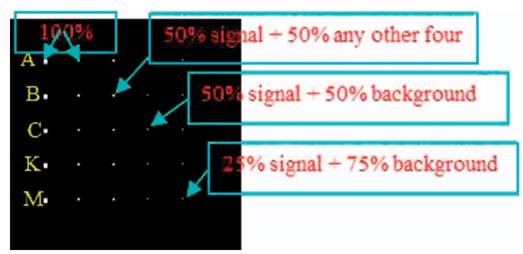


Fig. 3. Set of 25 panels simulated by A, B, C, K, and M.

collected over the Cuprite mining site, Nevada, in 1997. A total of 189 bands were used for experiments, where bands 1–3, 105–115, and 150–170 were removed before the analysis because of water absorption and a low signal-to-noise ratio (SNR) in those bands. The ground truth available for this region provides the pixel locations of five minerals shown in Fig. 2(b): alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M).

The synthetic image designed here is one of the scenarios presented in [27–29], where the five mineral spectral signatures—A, B, C, K, and M, marked by circles in Fig. 2(b)—were used to simulate the 25 panels shown in Fig. 3, with 5 panels in each row simulated by the same mineral signature and 5 panels in each column having the same size. The 25 panels consist of five 4 × 4 pure-pixel panels for each row in the first column, five 2 × 2 pure-pixel panels for each row in the second column, five 2 × 2 mixed-pixel panels for each row in the third column, and five 1 × 1 subpixel panels for each row in both the fourth column and the fifth column, where the mixed and subpanel pixels were simulated according to the legends in Fig. 2.

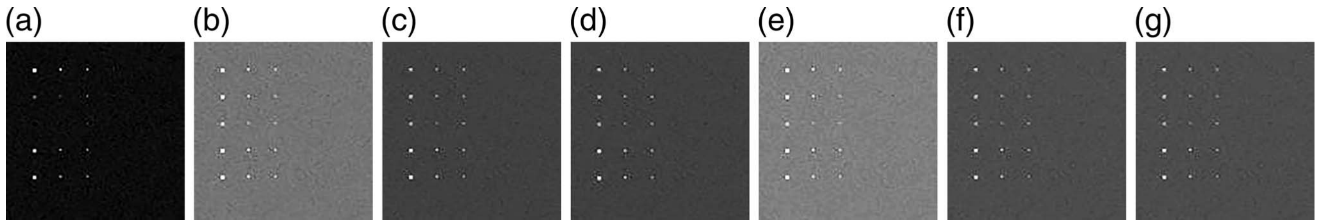


Fig. 4. Detection results for TI with detected abundance fractions in decibels: (a) fiftieth band of scenario TI, (b) K-RXD, (c) CK-RXD, (d) RT-CK-RXD, (e) R-RXD, (f) CR-RXD, and (g) RT-CR-RXD.

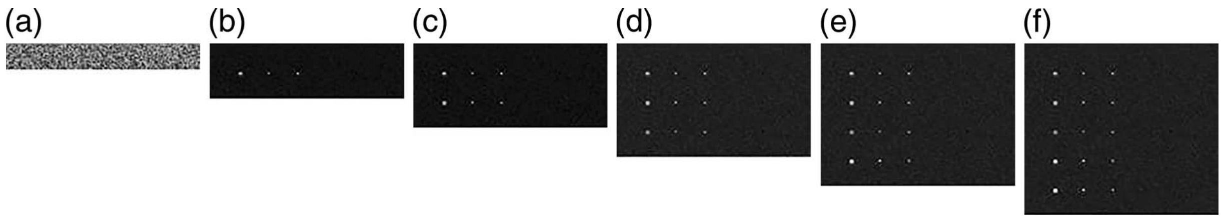


Fig. 5. CK-RXD detection results for TI with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

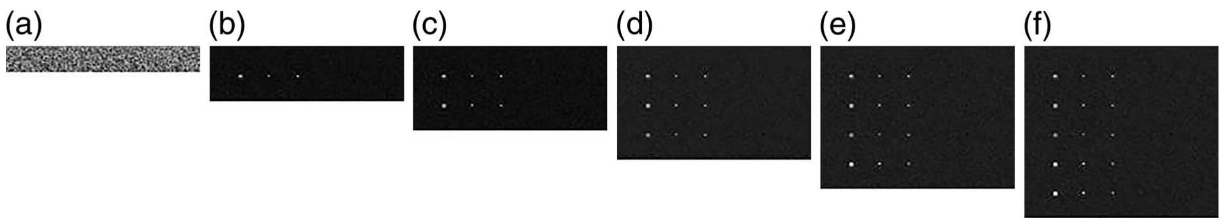


Fig. 6. RT-CK-RXD detection results for TI with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

So, 100 pure pixels (80 in the first column and 20 in the second column), called endmember pixels, were simulated in the data by the five endmembers: A, B, C, K, and M.

These 25 panels were then inserted in a synthetic image with a size of 200×200 pixels in a way that the background pixels were removed to accommodate the inserted target pixels. The background was simulated by the sample mean of the real image scene in Fig. 3(a) corrupted by a Gaussian noise to achieve the SNR of 20:1 defined in [31].

Once targets are simulated as explained earlier, an additive Gaussian noise was added to achieve a certain SNR. Once target pixels and background are simulated, two types of target insertion, called target implantation (TI) and target embeddedness (TE), can be designed to simulate experiments for various applications. Two types of six anomaly detectors—correlation matrix **R**–based anomaly detectors, R-RXD, CR-RXD, and RT-CR-RXD, and covariance matrix **K**–based anomaly detectors, K-RXD, CK-RXD, and RT-CK-RXD—are evaluated for detection performance and the CPT.

A. Target Implantation

The first type of target insertion is TI, which inserts the preceding 130 panel pixels into the image by replacing their corresponding background pixels. So, the resulting synthetic image has clean panel pixels implanted in a noisy background with an additive Gaussian noise of SNR = 20:1 for this scenario, as shown in Fig. 4(a). Figs. 4(b)–4(g) show results of traditional R-RXD and K-RXD, along with CR-RXD and CK-RXD, as well as with RT-CR-RXD and RT-CK-RXD, in terms of detected abundance fractions, where the value of x is represented in decibels (i.e., $20 \log_{10} x$) to enhance visual assessment. As we can see, all versions of anomaly detectors performed comparably except for different degrees of background suppression resulting from the use of global and causal correlation matrices.

Because CR-RXD and RT-CR-RXD use the same updating autocorrelation matrix $\mathbf{R}(n)$, both produce the same detection performance with only a difference in the CPT resulting from whether the causal innovation information update (5) is implemented. This is also true for CK-RXD and RT-CK-RXD. Figs. 5–8 show detection results of CK-RXD, RT-CK-RXD, CR-RXD, and RT-CR-RXD, respectively.

B. Target Embeddedness

The second type of target insertion is TE, which is the same as the TI described earlier except for the way the panel pixels were inserted. The background pixels were not removed to accommodate the inserted panel pixels, as

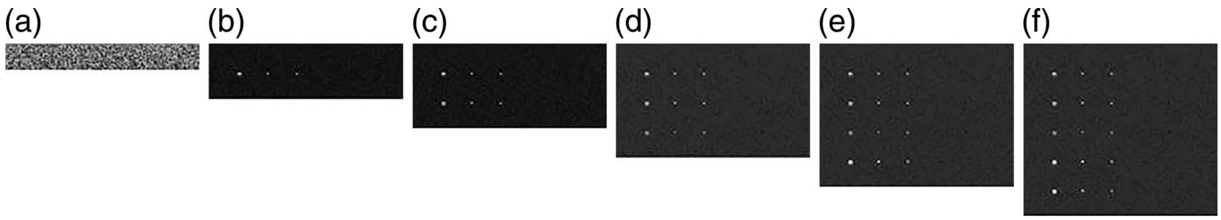


Fig. 7. CR-RXD detection results for TI with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

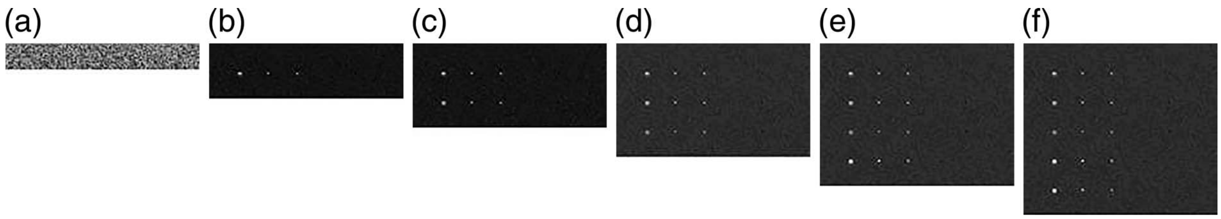


Fig. 8. RT-CR-RXD detection results for TI with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

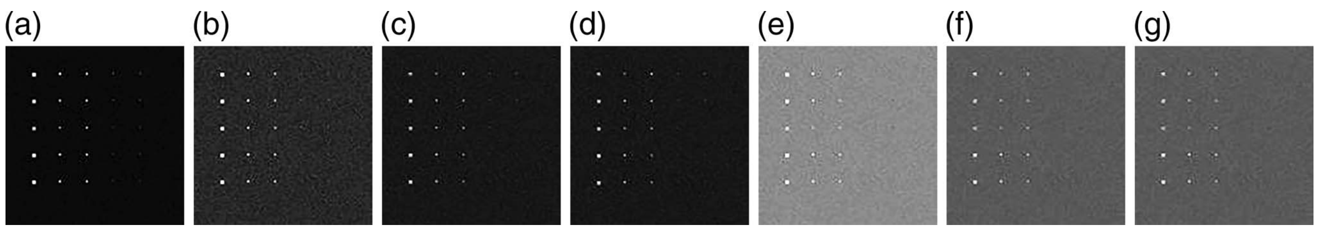


Fig. 9. Detection results for TE with detected abundance fractions in decibels: (a) fiftieth band of scenario TE, (b) K-RXD, (c) CK-RXD, (d) RT-CK-RXD, (e) R-RXD, (f) CR-RXD, and (g) RT-CR-RXD.

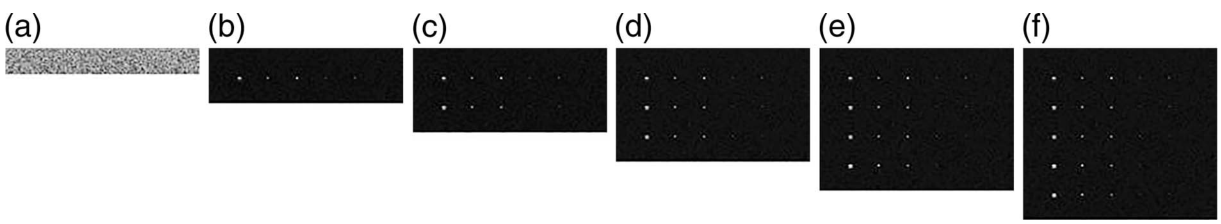


Fig. 10. CK-RXD detection results for TE with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

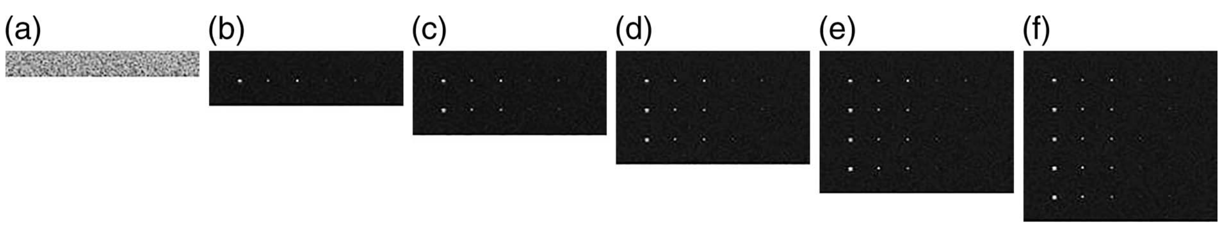


Fig. 11. RT-CK-RXD detection results for TE with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

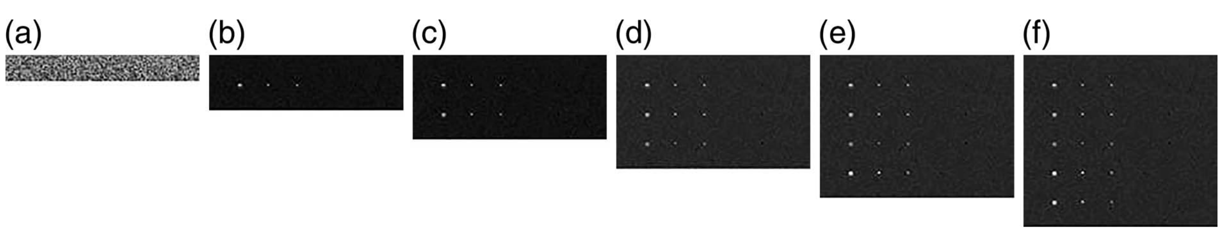


Fig. 12. CR-RXD detection results for TE with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

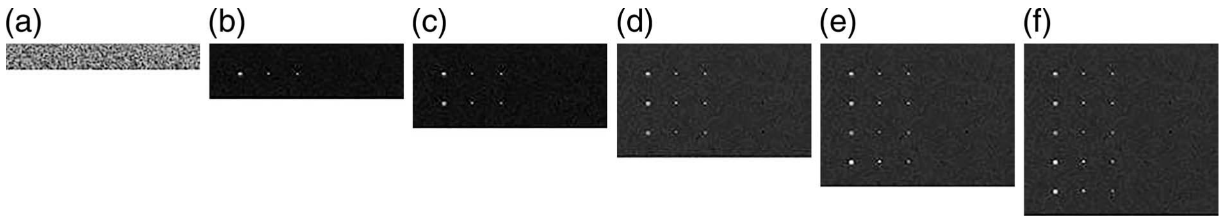


Fig. 13. RT-CR-RXD detection results for TE with detected abundance fractions in decibels: (a) no panels detected, (b) row 1 panels detected, (c) row 2 panels detected, (d) row 3 panels detected, (e) row 4 panels detected, and (f) row 5 panels detected.

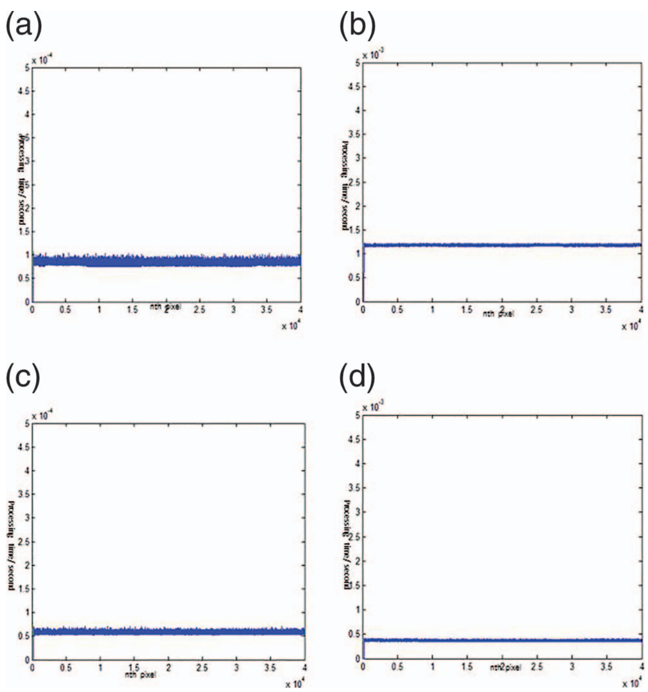


Fig. 14. CPT of CR-RXD, RT-CR-RXD, CK-RXD, and RT-CK-RXD required for TI: (a) $\mathbf{r}_n\mathbf{r}_n^T$, (b) $\mathbf{R}^{-1}(n)$, (c) $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$, and (d) $\left\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}$.

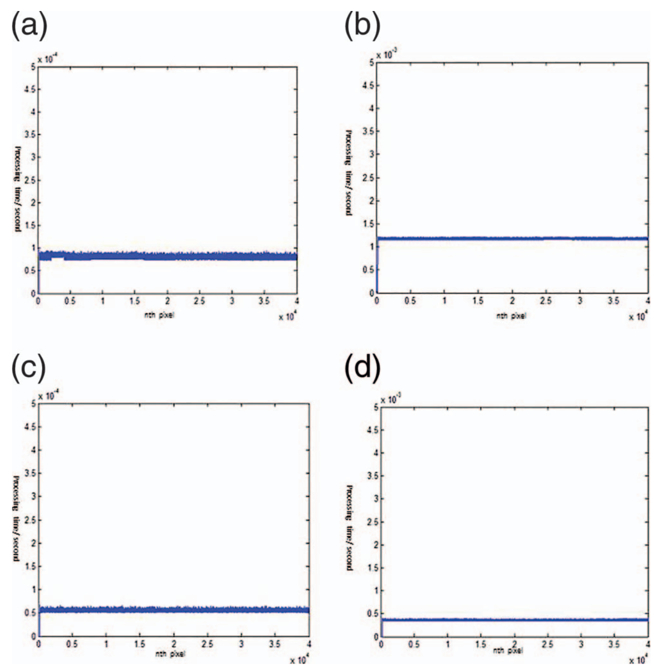


Fig. 15. CPT of CR-RXD, RT-CR-RXD, CK-RXD, and RT-CK-RXD required for TE: (a) $\mathbf{r}_n\mathbf{r}_n^T$, (b) $\mathbf{R}^{-1}(n)$, (c) $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$, and (d) $\left\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}$.

was done in TI; rather, they were superimposed with the inserted panel pixels. In this case, the resulting synthetic image, shown in Fig. 9(a), has clean panel pixels embedded in a noisy background with the same additive Gaussian noise as TI. The same experiments conducted for TI in Section V.A were repeated for scenario TE. Figs. 9(b)–9(g) show the detection results produced by K-RXD and R-RXD, along with CR-RXD and CK-RXD and with RT-CR-RXD and RT-CK-RXD, where their performances are close but have various degrees of background suppression because of the use of global and causal correlation matrices. Figs. 10–13 show detection results of CK-RXD, RT-CK-RXD, CR-RXD, and RT-CR-RXD, respectively.

C. Computational Complexity

Although causal and real-time causal anomaly detectors produce the same detection results, their required CPTs are different. The computer environments used for experiments were 64-bit operating systems with Intel i5-2500, a central processing unit (CPU) of 3.3 GHz, and 8 GB of random access memory (RAM). Figs. 14(a)–14(d) plot the CPT of calculating $\mathbf{R}^{-1}(n)$, $\mathbf{r}_n\mathbf{r}_n^T$, $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$, and $\left\{[\mathbf{R}(n-1)]^{-1}\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}$ required for per pixel vector by running four anomaly detectors, CR-RXD, RT-CR-RXD, K-RXD, and RT-CK-RXD, on scenario TI, and Figs. 15(a)–15(d) plot the same results for scenario TE. In both cases, the x-axis and y-axis represent the order of pixels being processed, i.e., the nth pixel vector, with n varying from 189 to 4000, and the CPT required to process nth pixel vector, respectively. Comparing Figs. 14(a), 14(c) and 15(a), 15(c) to Figs. 14(b), 14(d) and 15(b), 15(d), respectively, the computational complexity requiring computing a matrix inverse in 10^{-3} s is one order higher than computing an IP in 10^{-4} s.

As shown in Figs. 14 and 15, the processing time of running CR-RXD, CK-RXD, RT-CR-RXD, and RT-CK-RXD on both the TI and the TE scenarios is nearly

TABLE II
CPT for CR-RXD and CK-RXD and for RT-CR-RXD and RT-CK-RXD

CPT	CR-RXD/CK-RXD		RT-CR-RXD/RT-CK-RXD	
Initial condition	$\mathrm{CPT}(\mathbf{R}(L)) = L\mathrm{CPT}(\mathrm{OP}(L))$		$\mathrm{CPT}(\mathbf{R}(L) = L\mathrm{CPT}(\mathrm{OP}(L)) + \mathrm{CPT}(\mathrm{MI}(L))(\mathbf{R}^{-1}(L)/\mathbf{K}^{-1}(L))$	
	TI	TE	TI	TE
	0.01375 s	0.02111 s	0.0150 s	0.02882 s
CPT/pixel $n > L$	CPT $(\mathbf{r}_n(\mathbf{r}_n)^T)$		$\mathrm{CPT}(\{[\mathbf{R}(n-1)]^{-1}\, \mathbf{r}_n\}\,\{\mathbf{r}_n^T\, [\mathbf{R}(n-1)]^{-1}\}) + 3\mathrm{CPT}(\mathbf{x}^T\mathbf{Ay})$	
	TI	TE	TI	TE
	0.0001075	0.00009723	0.0004898 s	0.0004650 s
	$\mathrm{CPT}(\mathbf{R}^{-1}(n)/\mathbf{K}^{-1}(n))$			
	TI	TE		
	0.001222 s	0.001215 s		
Total CPT	$\sum_{n=L}^{N} \mathrm{CPT}(\mathrm{OP}(n)) + \mathrm{CPT}(\mathrm{MI}(n))$		$(N-L)\{[3(L+1)\mathrm{CPT}(\mathrm{IP}(L))\} + \mathrm{CPT}(\mathrm{OP}(L))]\}$	
	TI	TE	TI	TE
	54.0114 s	53.3096 s	20.8528 s	19.8661 s

constant for each pixel vector. So, if an anomaly detector is run on the entire image data, the CPT must be calculated by processing all image pixel vectors. In this case, the CPT required for CR-RXD or CK-RXD is $\mathrm{CPT}(\mathrm{OP}(L)) + \sum_{n=189}^{40000} [\mathrm{CPT}(\mathrm{MI}(n)) + (n+1)\mathrm{CPT}(\mathrm{IP}(L))]$, where $\mathrm{CPT}(\mathrm{MI}(n))$ and $\mathrm{CPT}(\mathrm{IP}(L))$ are the CPTs required to process finding matrix inverse $\mathbf{R}^{-1}(n)/\mathbf{K}^{-1}(n)$ and IP of two L-dimensional vectors, respectively, and $\mathrm{CPT}(\mathrm{OP}(L))$ is used to calculate the matrix OP $\mathbf{r}_n\mathbf{r}_n^T$ of the new nth L-dimensional input vector $\mathbf{r}_n$. However, the CPT required for RT-CR-RXD or RT-CK-RXD is $(40000 - 189)$ $\{(L+1)\mathrm{CPT}(\mathrm{IP}(L)) + \mathrm{CPT}(\mathrm{OP}(L)\}$ plus the CPT of processing the initial condition $\mathrm{CPT}(\mathrm{MI}(L))$, where $\mathrm{CPT}(\mathrm{OP}(L)$ is the CPT required to calculate the matrix OP of two L-dimensional vectors. The results of various CPTs in seconds required by running CR-RXD, CK-RXD, RT-CR-RXD, and RT-CK-RXD on complete images of scenarios TI and TE are tabulated in Table II.

One final comment on the performance in Figs. 4 and 9: If we plot the areas under their receiver operating characteristic (ROC) curve [21], they are all nearly close to 1. However, their detection maps are quite different in terms of background suppression by visual inspection. This is mainly because of the use of different sample correlation and covariance matrices implemented by various anomaly detectors. This phenomenon is particularly visible in the hyperspectral digital imagery collection experiments (HYDICEs) conducted in Section VI.

VI. REAL IMAGE EXPERIMENTS

Two real hyperspectral image scenes were specifically selected for experiments to conduct a performance evaluation of anomaly detection.

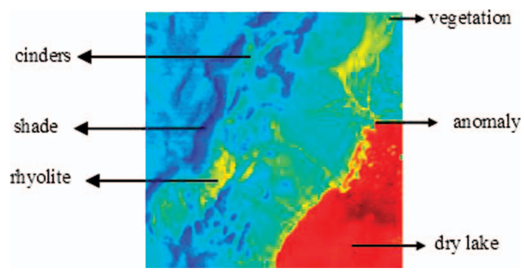


Fig. 16. AVIRIS LCVF subscene.

A. Airborne Visible Infrared Imaging Spectrometer Data

An airborne visible infrared imaging spectrometer (AVIRIS) image data set is used for the experiments shown in Fig. 16, using the Lunar Crater Volcanic Field (LCVF) located in Northern Nye County, Nevada. Atmospheric water bands and low SNR bands have been removed from the data, reducing the image cube from 224 to 158 bands. The image in Fig. 16 has a 10-nm spectral resolution and a 20-m spatial resolution. There are five target of interest: the radiance spectra of red oxidized basaltic cinders, rhyolite, playa (dry lake), vegetation, and shade. This scene is of particular interest because there is a 2-pixel-wide anomaly located at the left top edge of the crater.

Fig. 17 shows the final detection maps in decibels produced by six anomaly detectors: K-RXD, R-RXD, CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD. All were able to detect the 2-pixel-wide anomaly. Figs. 18–21 show progressive real-time causal processing of CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD, respectively, in six progressive stages; the detected abundance fraction maps are displayed in decibels for better visual assessment. Interestingly, as the detection process

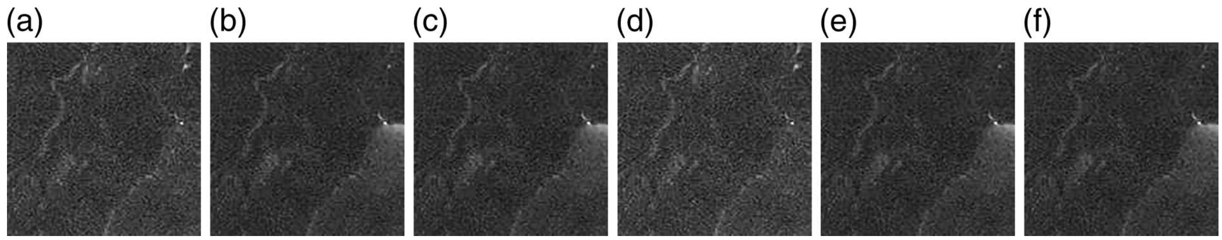


Fig. 17. Detection maps of LCVF with detected abundance fractions in decibels: (a) K-RXD, (b) CK-RXD, (c) RT-CK-RXD, (d) R-RXD, (e) CR-RXD, and (f) RT-CR-RXD.

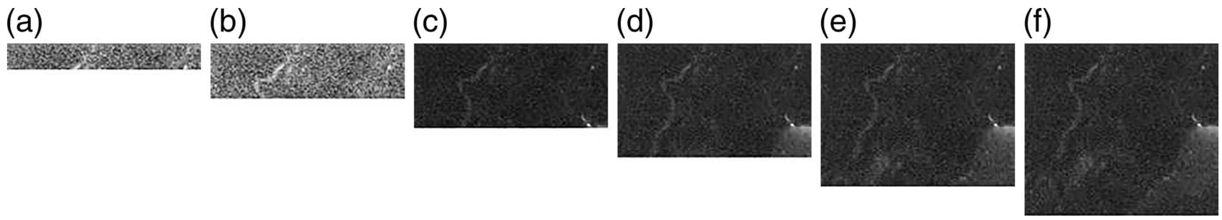


Fig. 18. CK-RXD detection results with detected abundance fractions in decibels: (a) vegetation, (b) cinders, (c) playa and anomaly detected, (d) shade, (e) rhyolite, and (f).

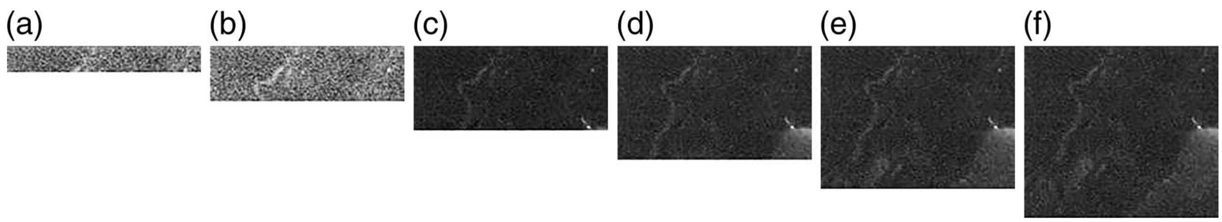


Fig. 19. CR-RXD detection results with detected abundance fractions in decibels: (a) vegetation, (b) cinders, (c) playa and anomaly detected, (d) shade, (e) rhyolite, and (f).

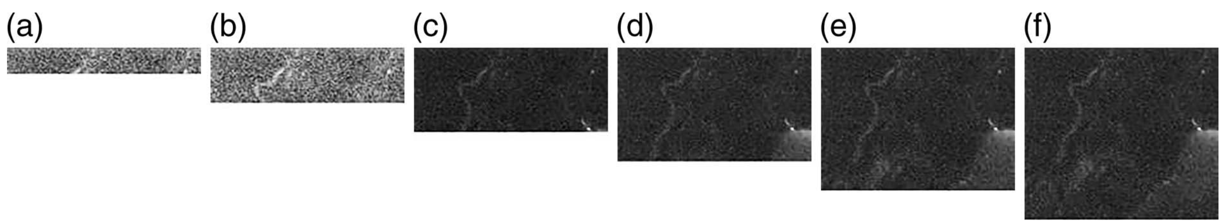


Fig. 20. RT-CK-RXD detection results with detected abundance fractions in decibels: (a) vegetation, (b) cinders, (c) playa and anomaly detected, (d) shade, (e) rhyolite, and (f).

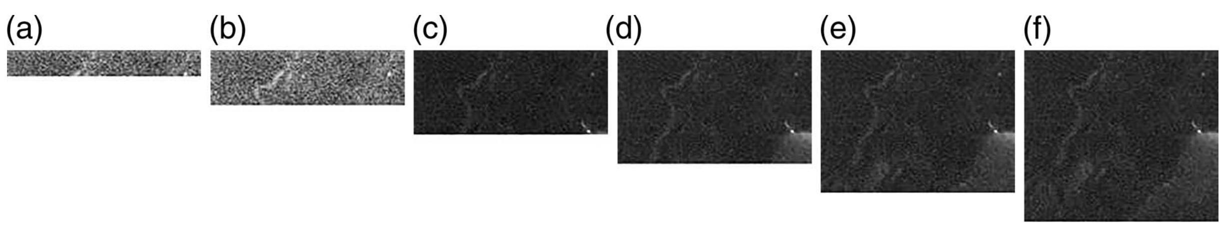


Fig. 21. RT-CR-RXD detection results with detected abundance fractions in decibels: (a) vegetation, (b) cinders, (c) playa and anomaly detected, (d) shade, (e) rhyolite, and (f).

progresses, different levels of background suppression could be also witnessed. This was particularly evident when the background was significantly suppressed once the process detected the anomaly. This was because the detected abundance fraction of the anomaly was so strong that the previously detected background information was overwhelmed by the anomaly. This is a good example to use in demonstrating the issue of background suppression in anomaly detection, which is discussed further in Section VI.C.

To further evaluate computational complexity, Fig. 22 plots the averaged CPT (10^{-4} s for Figs. 22(a), 22(c) and

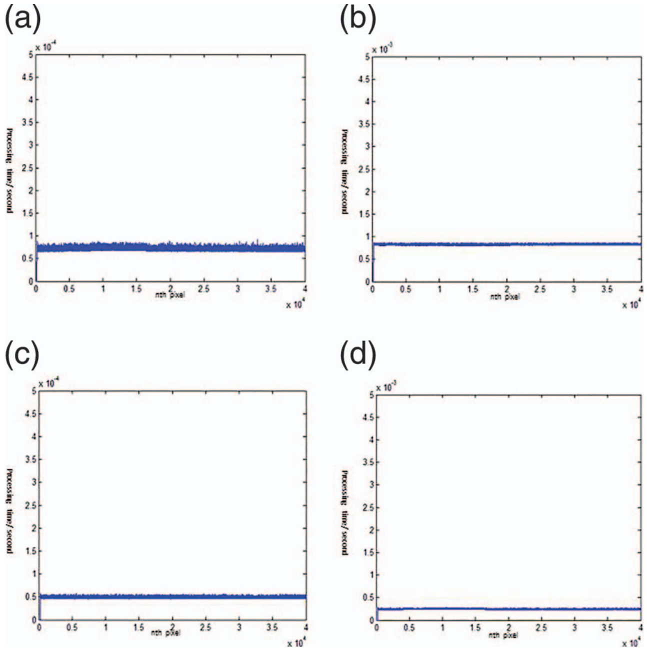


Fig. 22. CPT of CR-RXD, RT-CR-RXD, CK-RXD, and RT-CK-RXD required for LCVF: (a) $\mathbf{r}_n\mathbf{r}_n^T$, (b) $\mathbf{R}^{-1}(n)$, (c) $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$, and (d) $\left\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}$.

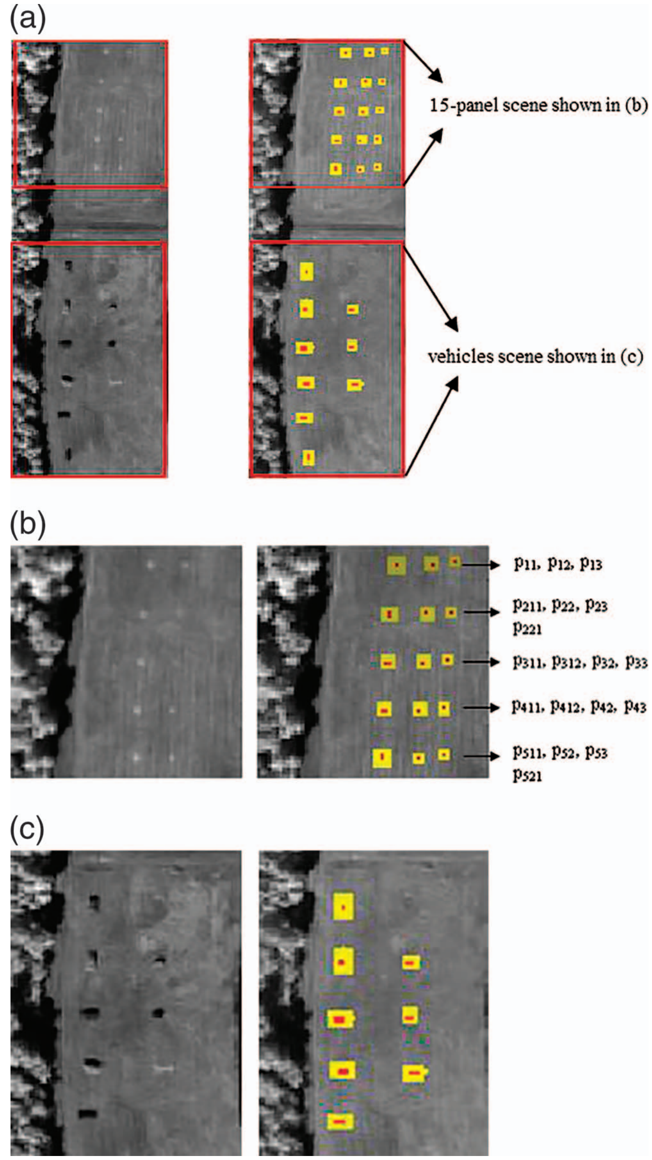


Fig. 23. HYDICE panels + vehicles scene. (a) HYDICE scene with ground truth map of spatial locations of 15 panels, five vehicles, and three objects. (b) Scene that contains 15 panels, with detailed ground truth map of spatial locations of 15 panels. (c) Vehicles + objects scene with ground truth map of five vehicles and three objects.

10^{-3} s for Figs. 22(b), 22(d)) of running CR-RXD, RT-CR-RXD, CK-RXD, and RT-CK-RXD on LCVF five times to compute various individual operations: $\mathbf{r}_n\mathbf{r}_n^T$, $\mathbf{R}^{-1}(n)$, $\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n$, and $\left\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}$. In Fig. 22, the x-axis and y-axis represent the order of pixels being processed, i.e., the nth pixel vector in the data and the CPT required to process the nth pixel vector, respectively. The figure clearly shows that the CPT resulting from using the recursive update equations specified by (5) and (7) is nearly constant for each image pixel vector. This implies that the CPT increases linearly with the number of data sample vectors processed. Similar to Figs. 14 and 15, it also required one order higher to invert a matrix (in 10^{-3} s) than to compute IP (in 10^{-4} s).

B. HYDICE Data

The HYDICE image scene shown in Fig. 23(a) has size of 200 × 74-pixel vectors, along with its ground truth provided in Fig. 23(b), where the center and boundary pixels of objects are highlighted by red and yellow, respectively.

The upper part in Fig. 23(b) contains fabric panels with sizes of 3, 2, and 1 m^2 from the first column to the third column. Because the spatial resolution of the data is 1.56 m^2, the panels in the third column are considered subpixel anomalies. The lower part in Fig. 23(c) contains different vehicles with sizes of 4m × 8m (the first four vehicles in the first column) and 6m × 3m (the bottom vehicle in the first column) and three objects in the second column (the first two have a size of 2 pixels and the bottom one has a size of 3 pixels). In this particular scene, there are three types of man-made targets with different sizes: small targets (panels of 3, 2, and 1 m^2), large targets (vehicles of 4m × 8m and 6m × 3m), and three objects of 2 and 3 pixels to be used to validate and test anomaly detection performance.

There are several advantages of using the HYDICE image scene in Fig. 23(a). One is that the ground truth provides precise spatial locations of all man-made target pixels, which allows us to evaluate real-time processing performance of anomaly detection pixel by pixel, a task that has not been explored in the past. Second, the provided ground truth enables us to perform ROC analysis for anomaly detection via ROC curves of detection rate versus false-alarm rate. Third, the scenes have various sizes of objects that can be used to evaluate the ability of

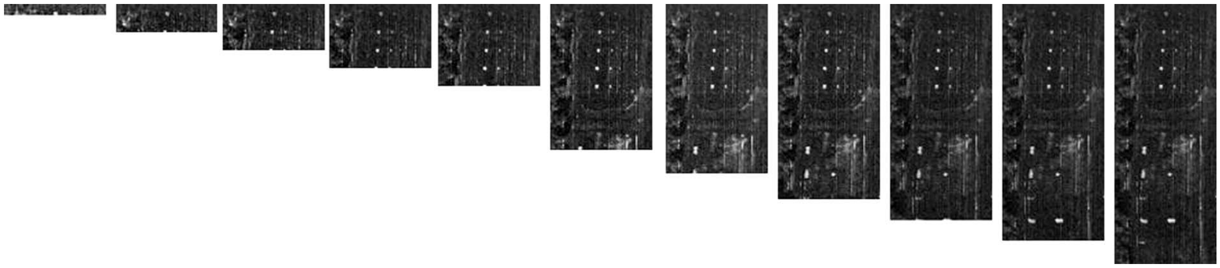


Fig. 24. CK-RXD with detected abundance fractions in decibels.

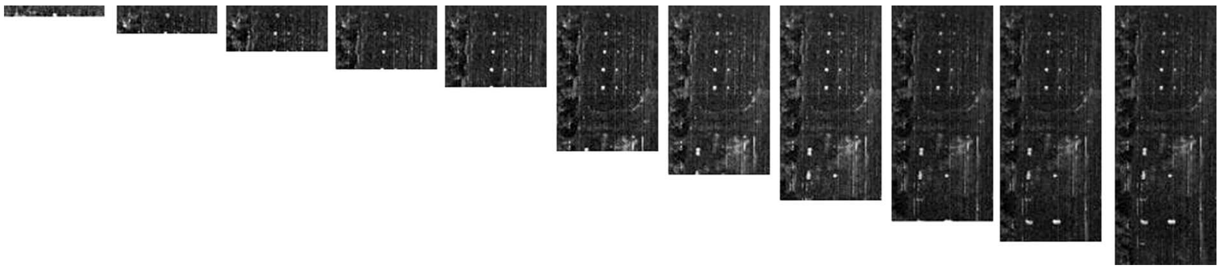


Fig. 25. CR-RXD with detected abundance fractions in decibels.

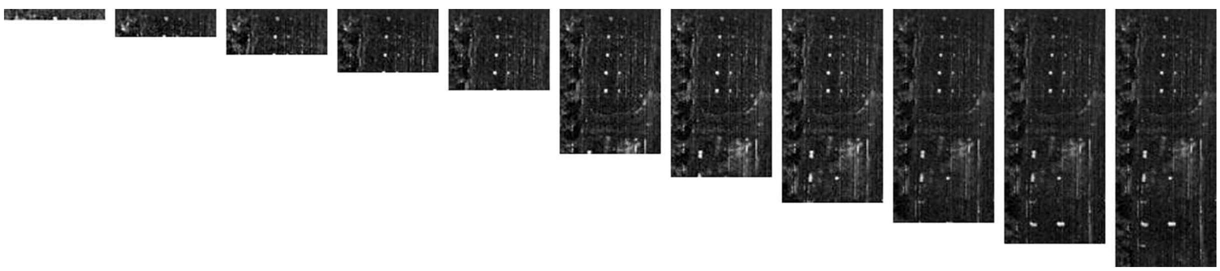


Fig. 26. RT-CK-RXD with detected abundance fractions in decibels.

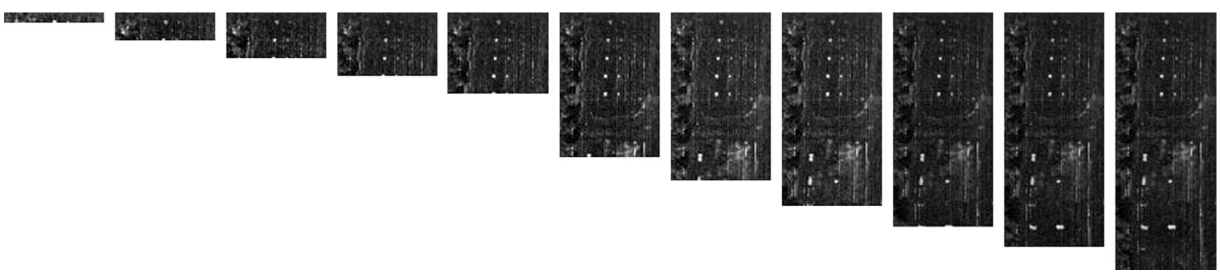


Fig. 27. RT-CR-RXD with detected abundance fractions in decibels.

an anomaly detector in detecting anomalies with different sizes, an issue that has not been addressed in many reports. Fourth, this scene can be processed by operating the same anomaly detector on the three image sizes shown in Figs. 23(a)–23(c) (i.e., a 15-panel scene of 64×64-pixel vectors, marked by an upper rectangle; a vehicles $+$ objects scene of 100×64-pixel vectors, marked by a lower rectangle; and the entire scene containing 15 panels and vehicles $+$ objects) to evaluate the effectiveness of its performance. Finally, and most importantly, the clean natural background and targets make visual assessment easier, revealing various degrees of background being suppressed by an anomaly detector.

1) *Real-Time Causal Processing*: To see how causal and real-time causal anomaly detection perform, Figs. 24–27 show the real-time causal processing of CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD, respectively, on Fig. 23(a). Their detected abundance fractions are in decibels, and each pass shows a real-time detection map of different targets.

Because Figs. 23(b) and 23(c) are part of the scene in Fig. 23(a), the results of real-time processing of these two subscenes are not included here. Nevertheless, their detection results are discussed in detail in the following two subsections. To avoid the singularity problem of calculating the inverse of the sample correlation and covariance matrix used by anomaly detectors, an anomaly detector does not begin to operate until it collects a sufficient number of initial data sample vectors, which is the total number of spectral bands of the image to be processed. K-RXD and R-RXD are not included in the experiments because they are neither causal nor real time. By visually inspecting the results in Figs. 24–27, sample CR-RXD and RT-CR-RXD seemed to perform slightly better that than their counterparts CK-RXD and RT-CK-RXD in terms of panel pixel detection. Interestingly, the conclusion is reversed if detection of vehicles is of major interest. This observation is confirmed by the following ROC analysis. Because the CPT for this scene is similar to Fig. 22, its plots are not included here.

2) *Detection Performance and Three-Dimensional ROC Analysis*: Using the ground truth provided by Fig. 23, we can perform quantitative study via ROC analysis. In doing so, an idea similar to that proposed in [1, 29, 32] can be derived by converting real values to hard decisions as follows.

Assume that $\delta^{\mathrm{AD}}(\mathbf{r})$ 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

$$\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r}) = \frac{\hat{\delta}^{\mathrm{AD}}(\mathbf{r}) - \min_{\mathbf{r}} \hat{\delta}^{\mathrm{AD}}(\mathbf{r})}{\max_{\mathbf{r}} \hat{\delta}^{\mathrm{AD}}(\mathbf{r}) - \min_{\mathbf{r}} \hat{\delta}^{\mathrm{AD}}(\mathbf{r})}. \tag{9}$$

More specifically, $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ in (9) can be regarded as a probability vector, which calculates the likelihood of the data sample vector $\mathbf{r}$ being detected as an anomaly according to its detected abundance fraction $\delta^{\mathrm{AD}}(\mathbf{r})$. Because of (9), we can develop an abundance percentage anomaly converter (APAC) with $a\%$ as a threshold criterion ($a\%$APAC), $\chi_{a\%\mathrm{PAC}}(\mathbf{r})$, which is similar to one proposed in [1, 29, 32], as follows:

$$\chi_{a\%\mathrm{PAC}}(\mathbf{r}) = \begin{cases} 1; \text{if } \hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r}) \geq \tau = \frac{a}{100} \\ 0; \text{otherwise} \end{cases}. \tag{10}$$

If $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ in (10) exceeds $\tau = a\%/100$, then the $\mathbf{r}$ is detected as an anomaly. So, a 1 produced by (10) indicates that pixel $\mathbf{r}$ is detected as an anomaly; otherwise, it is considered a background pixel.

In the context of (10), we consider the Neyman-Pearson detection theory for a binary hypothesis testing problem to perform signal detection [21], where $\hat{\delta}^{\mathrm{AD}}_{\mathrm{normalized}}(\mathbf{r})$ in (9) can be used as a Neyman-Pearson detector to perform the ROC analysis as a performance evaluation tool. For example, for a particular threshold τ, a detection probability or power P_D and a false-alarm probability P_F can be calculated. By varying the threshold $\tau = a\%/100$ in (10), we can produce an ROC curve of P_D versus P_F and further calculate the area under the ROC curve for quantitative performance analysis. Interestingly, the threshold τ is absent from the traditional ROC curve. But according to (10), the values of P_D and P_F are calculated through τ. To address this issue, a three-dimensional (3D) ROC analysis was recently developed in [29, 32], where 3D ROC curves can be generated by considering P_D, P_F, and τ as three parameters, each of which represents one dimension. In other words, a 3D ROC curve is a 3D curve of (P_D, P_F, τ) from which three two-dimensional (2D) ROC curves can be also generated: the 2D ROC curve of (P_D, P_F), which is the traditional ROC curve discussed in [21], and two new 2D ROC curves, the 2D ROC curve of (P_D, τ) and the 2D ROC curve of (P_F, τ). Fig. 28 plots the 3D ROC curves, along with their corresponding three 2D ROC curves produced by the six anomaly detection algorithms K-RXD, R-RXD, CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD, for three image scenes in Figs. 23(a)–23(c): entire image scene, 15-panel scene, and vehicles + objects scene.

To perform quantitative analysis, we further calculated the area under the curve, denoted by A_z, for each of 2D ROC curves produced in Figs. 28(b)–28(d) by six anomaly detection algorithms: K-RXD, R-RXD, CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD. Their results are tabulated in Tables III–V, where the best results are shaded. The results of two global anomaly detectors, K-RXD and R-RXD, are included for comparison. For 2D ROC curves of (P_D, P_F) and (P_D, τ), the higher the value of A_z, the better the detector. Conversely, for 2D ROC curves of (P_F, τ), the lower the value of A_z, the better the detector.

Based on Tables III–V, the best performance of anomaly detection varies with image size even if the same targets are present in the three image scenes in Figs. 23(a)–23(c). For example, the same 15 panels are present in Figs. 23(a), 23(b), but the best anomaly detectors differed in terms of A_z calculated for 2D ROC curves of (P_D, P_F) and (P_D, τ) in Tables III and IV, i.e., K-RXD for the entire image and R-RXD for the 15-panel scene. However, for the same five vehicles and three objects in Figs. 23(a), 23(c), the best anomaly detector was RT-CK-RXD or CK-RXD for the vehicles scene in Table III, according to the values of A_z calculated for 2D ROC curves of (P_D, P_F) and (P_D, τ). Interestingly, for all three image scenes, the best anomaly detector to produce the smallest A_z of (P_F, τ) was RT-CK-RXD or RT-CR-RXD. This indicates that a smaller A_z of (P_F, τ) implies less background suppression. Furthermore, a higher A_z of (P_D, P_F) does not necessarily imply a higher A_z of (P_F, τ), as shown in Tables IV and V. Unfortunately, such pieces of information are not provided by traditional 2D ROC analysis, A_z of (P_D, P_F). These experiments demonstrate the utility of 3D ROC analysis via three 2D ROC curves generated from a 3D ROC curve. That is, anomaly detection performance can be analyzed through interrelationships among P_D, P_F, and the threshold τ via three 2D ROC curves plotted based on three pairs: (P_D, P_F), (P_D, τ), and (P_F, τ).

3) *Background Suppression*: In general, the performance of anomaly detection is evaluated based on its detection rates or ROC analysis as demonstrated in

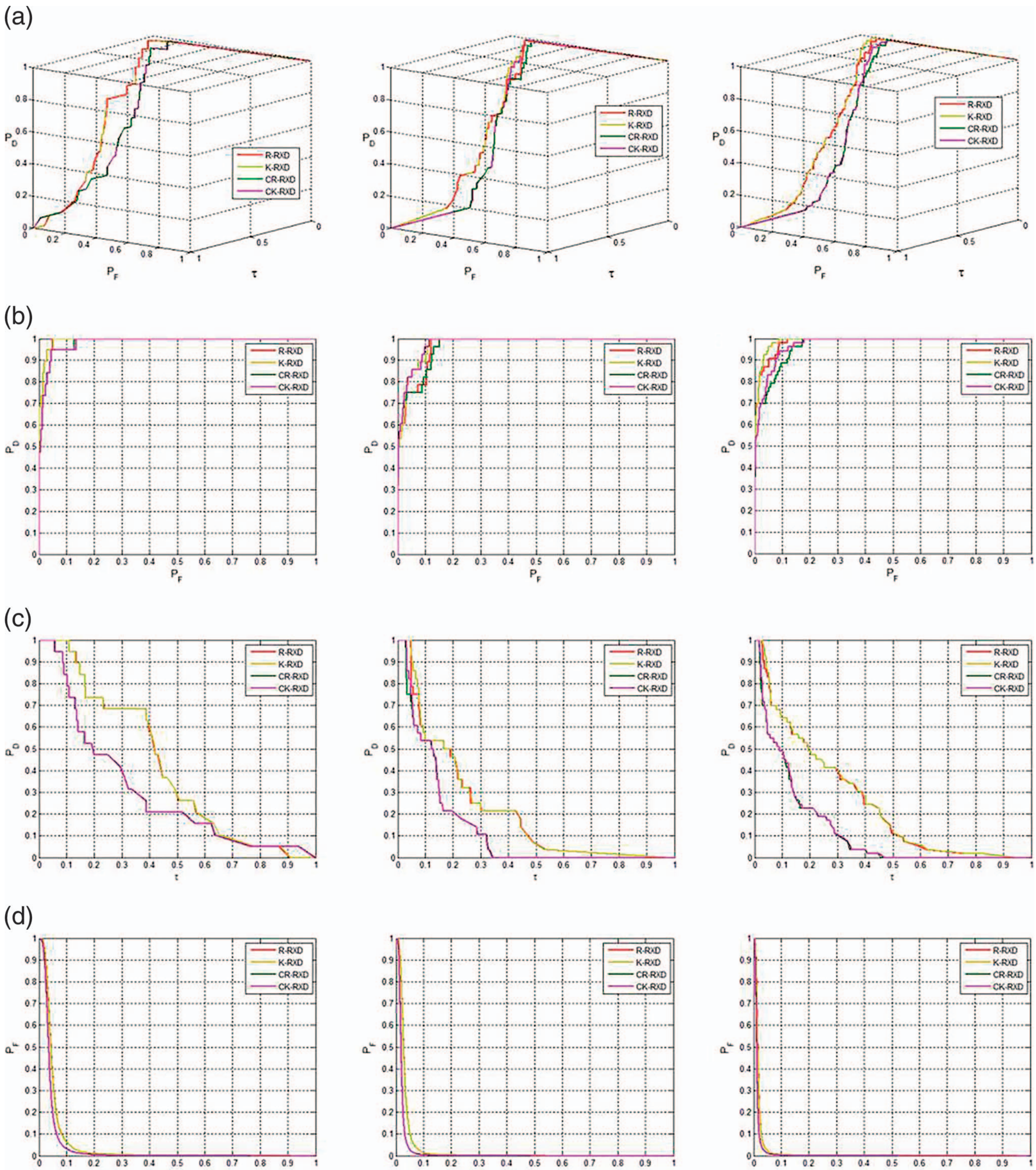


Fig. 28. 3D ROC curves and three corresponding 2D ROC curves for 15-panel (left), vehicles (center), and entire (right) scenes. (a) 3D ROC curves of (P_D, P_F, τ). (b) 2D ROC curves of (P_D, P_F). (c) 2D ROC curves of (P_D, τ). (d) 2D ROC curves of (P_F, τ).

Section VI.B.2. However, because anomaly detection is carried out without prior knowledge or ground truth, there is no way of using ROC analysis to conduct performance evaluation. It must rely on visual inspection, which becomes the only means of evaluating anomaly detection performance. In this case, background suppression has an impact on visual inspection and is crucial for anomaly detection. This was already demonstrated in Figs. 18–21 for the LCVF scene, where the 2-pixel-wide anomaly dominated the entire detection process. In other words, if we consider background as a null hypothesis H_0 versus targets as an alternative hypothesis H_1 in a binary

TABLE III
Values of Areas Under Three 2D ROC Curves (A_z) Produced by Six Algorithms (Entire Panels + Vehicles Scene)

Algorithm	K-RXD	CK-RXD	RT-CK-RXD	R-RXD	CR-RXD	RT-CR-RXD
A_z of (P_D, P_F)	**0.9886**	0.9818	0.9819	0.9840	0.9747	0.9747
A_z of (P_D, τ)	**0.2368**	0.1372	0.1372	0.2349	0.1356	0.1356
A_z of (P_F, τ)	0.0193	**0.0144**	**0.0144**	0.0199	0.0145	0.0145

Note: Best results are in bold.

TABLE IV
Values of Areas Under Three 2D ROC Curves (A_z) Produced by Six Algorithms (15-Panel Scene)

Algorithm	K-RXD	CK-RXD	RT-CK-RXD	R-RXD	CR-RXD	RT-CR-RXD
A_z of (P_D, P_F)	0.9898	0.9680	0.9683	**0.99**	0.9691	0.9691
A_z of (P_D, τ)	0.3329	0.2590	0.2590	**0.3342**	0.2596	0.2596
A_z of (P_F, τ)	0.0428	**0.0372**	**0.0372**	0.0433	0.0377	0.0377

Note: Best results are bold.

TABLE V
Values of Areas Under Three 2D ROC Curves (A_z) Produced by Six Algorithms (Vehicles Scene)

Algorithm	K-RXD	CK-RXD	RT-CK-RXD	R-RXD	CR-RXD	RT-CR-RXD
A_z of (P_D, P_F)	0.9751	**0.9776**	**0.9776**	0.9669	0.9662	0.9662
A_z of (P_D, τ)	**0.2172**	0.1307	0.1307	0.2150	0.1294	0.1294
A_z of (P_F, τ)	0.0332	**0.0221**	**0.0221**	0.0333	0.0222	0.0222

Note: Best results are bold.

hypothesis testing problem, 3D ROC analysis dictates the behavior of a detector in terms of detection rate P_D and false-alarm rate P_F versus threshold τ. That is, a better target detection produces a higher A_z of (P_D, τ), as well as a higher A_z of (P_F, τ) as false-alarm probability, and thus results in better a background suppression, which indicates poor background detection according to binary hypothesis testing formulation. Unfortunately, to the best of our knowledge, the issue in background suppression has not been explored or investigated. This HYDICE image data offers an excellent opportunity to look into this issue and further demonstrates that an anomaly detector with a high detection rate may generate a higher false-alarm rate, which in turn may have more background suppression. But does it imply that better background suppression gives rise to better anomaly detection? To illustrate this phenomenon, Figs. 29(c)–29(f) show detected abundance fraction maps of three scenes generated by completing the real-time processing of CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD. We include the detected abundance fraction maps produced by the global anomaly detectors, K-RXD and R-RXD, in Figs. 29(a), 29(b), respectively, for comparison. By examining the abundance fractions detected by the six anomaly detectors, there is no appreciable visual difference among all the results. However, the original detected abundance fractional values in decibels (i.e., $20(\log_{10} x)$, with x being the original detected abundance fraction), as shown in Fig. 30, provides better visual inspection and assessment than does Fig. 29.

From Fig. 30, it seems that all six anomaly detectors performed comparably in detection of targets but that the global anomaly detectors, K-RXD and R-RXD, had better background suppression than their real-time and causal counterparts in terms of suppressing grass surrounding panels, vehicles, and objects. This makes sense. Because a global anomaly detector uses the global spectral correlation provided by the sample correlation and covariance matrix of the entire image data, it performs better background suppression than any local anomaly detector, as expected. However, on many occasions, when there is no prior knowledge is available, background information may help image analysts perform better data analysis because background generally provides crucial information surrounding anomalies. If background suppression is overdone, we may not have clues about anomalies. For example, in Fig. 29, anomalies were detected with clean background suppression; we have no idea what these anomalies are and simply know their spatial locations. But if we look into Fig. 30, the background has a tree line along the left edge, panels were placed on grass, and vehicles were parked in a dirt field. This is particularly true for medical imaging, where background detection is interpreted as tissue anatomical structures, which help doctors greatly in their diagnoses.

C. Computational Complexity

This section calculated the computing time in seconds required by running CK-RXD, CR-RXD, RT-CK-RXD, and RT-CR-RXD on LCVF and three HYDICE scenes by

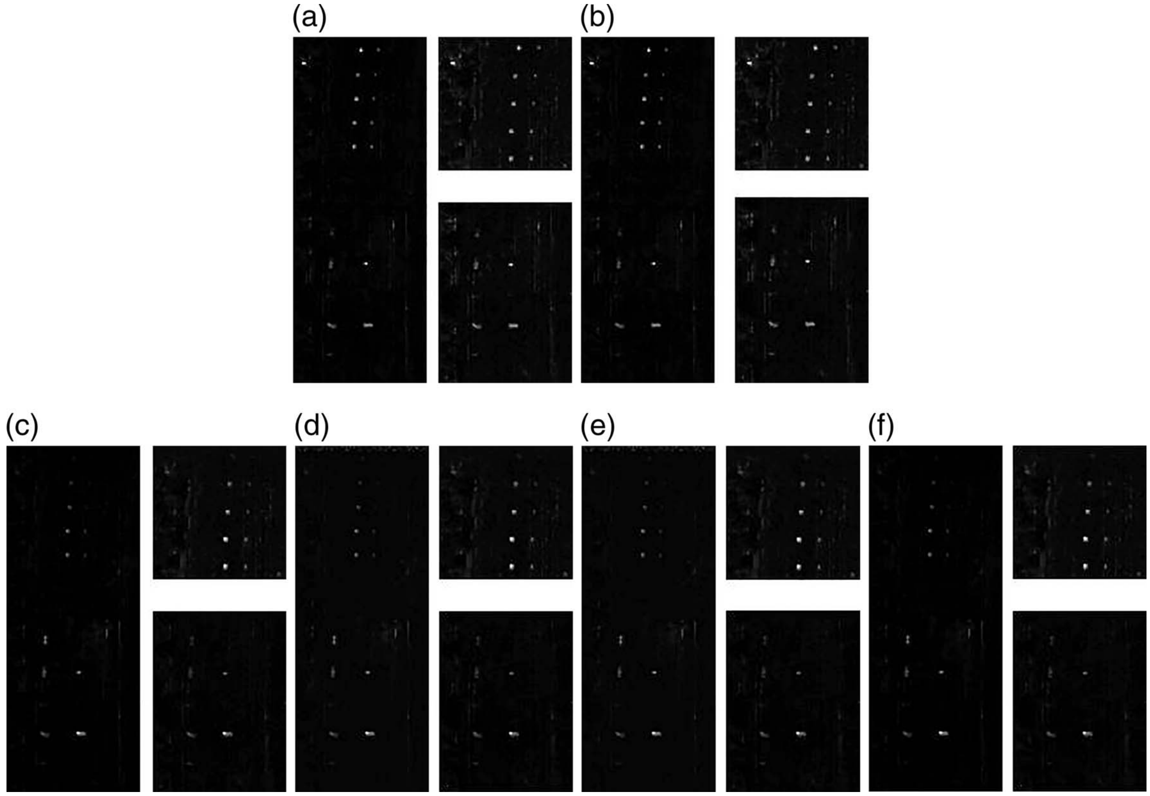


Fig. 29. Detection maps with detected abundance fractions: (a) K-RXD, (b) R-RXD, (c) CK-RXD, (d) CR-RXD, (e) RT-CK-RXD, and (f) RT-CR-RXD.

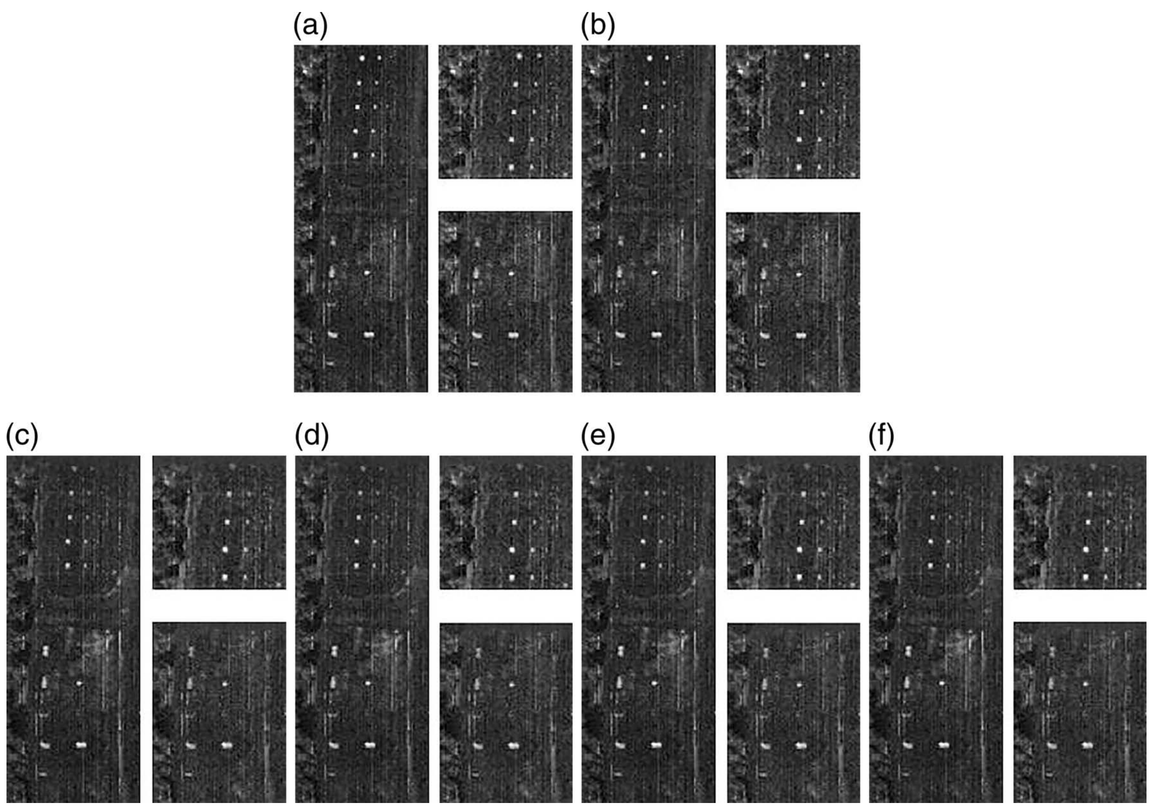


Fig. 30. Detection maps with detected abundance fractions in decibels: (a) K-RXD, (b) R-RXD, (c) CK-RXD, (d) CR-RXD, (e) RT-CK-RXD, and (f) RT-CR-RXD.

TABLE VI
CPT in Seconds for CR-RXD/CK-RXD and RT-CR-RXD/RT-CK-RXD

CPT	CR-RXD/CK-RXD				RT-CR-RXD/RT-CK-RXD			
Initial condition	$\mathrm{CPT}(\mathbf{R}(L)) = L\mathrm{CPT}(\mathrm{OP}(L))$				$\mathrm{CPT}(\mathbf{R}(L) = L\mathrm{CPT}(\mathrm{OP}(L)) + \mathrm{CPT}(\mathrm{MI}(L))\,(\mathbf{R}^{-1}(L)/\mathbf{K}^{-1}(L))$			
	LCVF	Vehicles + panels	Panels	Vehicles	LCVF	Vehicles + panels	Panels	Vehicles
	0.03075	0.009483	0.004943	0.01503	0.01965	0.003772	0.01040	0.002998
CPT/pixel $n > L$	$\mathrm{CPT}\ (\mathbf{r}_n(\mathbf{r}_n)^T)$				$\mathrm{CPT}(\left\{[\mathbf{R}(n-1)]^{-1}\,\mathbf{r}_n\right\}\left\{\mathbf{r}_n^T\,[\mathbf{R}(n-1)]^{-1}\right\}) + 3\mathrm{CPT}(\mathbf{x}^T\mathbf{Ay})$			
	LCVF	Vehicles + panels	Panels	Vehicles	LCVF	Vehicles + panels	Panels	Vehicles
	9.224e−05	6.227e−05	5.4498e−05	6.0257e−05	3.398e−04	3.1712e−04	3.1981e−04	3.2198e−04
	$\mathrm{CPT}(\mathbf{R}^{-1}(n)/\mathbf{K}^{-1}(n))$							
	LCVF	Vehicles + panels	Panels	Vehicles				
	8.737e−04	9.9041e−04	9.6491e−04	9.9305e−04				
Total CPT	$\sum_{n=L}^{N} \mathrm{CPT}(\mathrm{OP}(n)) + \mathrm{CPT}(\mathrm{MI}(n))$				$(N-L)\{[3(L+1)\mathrm{CPT}(\mathrm{IP}(L))\} + \mathrm{CPT}(\mathrm{OP}(L))]\}$			
	LCVF	Vehicles + panels	Panels	Vehicles	LCVF	Vehicles + panels	Panels	Vehicles
	39.426	15.1476	4.2796	7.9846	14.754	4.4624	1.3203	2.3856

TABLE VII
CPT in Seconds for K-RXD and R-RXD

K-RXD $\mathbf{K}^{-1}$				R-RXD $\mathbf{R}^{-1}$			
LCVF	Vehicles + panels	Panels	Vehicles	LCVF	Vehicles + panels	Panels	Vehicles
0.108553	0.036878	0.015634	0.018485	0.115556	0.035656	0.013076	0.026824
$(\mathbf{r}_n - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{r}_n - \boldsymbol{\mu})$				$(\mathbf{r}_n)^T \mathbf{R}^{-1} (\mathbf{r}_n)$			
LCVF	Vehicles + panels	Panels	Vehicles	LCVF	Vehicles + panels	Panels	Vehicles
0.190389	0.069578	0.024734	0.050299	0.120606	0.067772	0.020679	0.027323

averaging five runs, where the computer environments used for experiments were a 64-bit operating system with Intel i5-2500, a CPU of 3.3 GHz, and 8 GB of RAM. Their results are tabulated in Table VI. In Table VI, the time required by causal RXD to compute initial conditions for a vehicles scene was greater than that for an entire scene, which includes a vehicles scene as a subscene. This is also true for real-time causal RXD, where the computing time for a panels scene is greater than a panels + vehicles scene. This is because all the causal and real-time casual anomaly detectors do not start processing data until sufficient data sample vectors are collected to compute initial conditions, such as the total number of spectral bands, to avoid an ill-rank issue arising in calculating a sample correlation and covariance matrix. In this case, all the algorithms use the same set of 169 image pixels to calculate their initial conditions, and their algorithmic structures determine the computational complexity.

As we can see from Table VI, a real-time causal anomaly detector generally runs two or three times faster than its causal counterpart while retaining the same performance. Table VII tabulates the computing time required by two global anomaly detectors, K-RXD and R-RXD, to process the entire image data. Comparing Table VII to Table VI may lead to a brief that the computational complexity required by a real-time causal anomaly detector is exceedingly high. It is not if we consider that a real-time causal anomaly detector must update and recalculate its sample correlation and covariance matrix every time a new input data sample vector comes in. In addition, its computational complexity is linearly increased with the number of data sample vectors required to be processed, as shown in Fig. 22, where the processing time for each pixel is nearly constant. So, because a global anomaly detector only needs to calculate the sample correlation and covariance matrix once for all data sample vectors, the computing time of a real-time causal anomaly detector is supposed to be the computing time required by a global anomaly detector multiplied by its total number of data sample vectors being processed to some extent. However, the computing time documented in Table VI is significantly less than that. This tremendous saving was mainly because a real-time causal anomaly detector uses a recursive causal update equation specified by either (6) or (8), which uses only innovation information provided by the data sample vector to update the equation without reprocessing already-visited data sample vectors.

VII. CONCLUSIONS

One of most important applications in hyperspectral data exploitation is anomaly detection. However, to see how effectively an anomaly detector can perform, real-time processing is more practical in real-world applications, specifically detection of moving or instantaneous targets. Most significantly, real-time processing anomaly detection provides an unparalleled advantage that commonly used anomaly detectors cannot offer: progressive changes in different levels of background suppression for visual assessment and evaluation. Unfortunately, a true real-time causal processing algorithm generally does not exist if real-time processing is interpreted as having input and output data simultaneously. However, from a practical point of view, as long as an algorithm can process data in a negligible amount of time, satisfying constraints imposed by specific applications, it can be viewed as a real processing algorithm. With this interpretation, many supposedly real processing algorithms are actually fast computational algorithms, as determined by exploring various data organizations, parallel structures, field programmable gate array architectures, etc. Nevertheless, there is a missing element in such real-time processing algorithms, which is causality, an important prerequisite to real-time processing. In other words, a real-time processing algorithm must be also a causal algorithm, because a real-time processing algorithm does not have access to future inputs during the course of data processing. This is particularly applied to many window-based anomaly detectors, which are not causal and thus are not real-time processing detectors. This paper is believed to be the first work devoted to exploring this concept in anomaly detection. Specifically, it derives a causal innovation information update equation for implementing real-time causal anomaly detection. To investigate the computational complexity issue, a comprehensive comparative analysis on the CPT of running causal and real causal RXD-based anomaly detectors is conducted in theory and experiments. Finally, the real image experiments conducted in Section VI bring up an interesting and intriguing issue in progressive background suppression, which has a large impact on anomaly detection. In global anomaly detection, little has been done in background suppression. However, as demonstrated in anomaly detection of LCVF in Fig. 22, real-time processing offers a significant advantage to see time-varying changes in background information as time moves along, where various levels of background suppression produce different rates of false alarms and thus have tremendous effect on visual assessment. This issue is worth pursuing, although it is beyond the scope of this paper. Here, we have made an effort to investigate the impact of background suppression on anomaly detection. We summarize several contributions made in this paper as follows:

1) Causality has been introduced into real-time processing. To the best of our knowledge, no real-time processing algorithms investigate this issue in the remote sensing community, which is crucial and a prerequisite to real-time processing: No anomaly detector can be implemented in real time because of its use of sample correlation and windows, which require future data samples before the data sample can be processed. Our RT-CR-RXD and RT-CK-RXD are such real-time causal anomaly detectors, which have never been developed in the literature.

2) One of most important contributions resulting from this paper is real-time progressive analysis of anomaly detection, which allows users to see how an anomaly detector performs various degrees of background suppression. This paper is believed to be the first work to investigate this issue, because background suppression provides users with better understanding of what detected anomalies are. Specifically, some weak anomalies detected earlier may be overwhelmed by strong anomalies detected later. This phenomenon is important in anomaly detection but cannot be observed using commonly used anomaly detectors that perform a one-shot operation to show final detected anomalies.

3) We use 3D ROC analysis to analyze detection performance, specifically, progressive performance of background suppression. This is also believed to be the first work in this area reported in anomaly detection literature.

APPENDIX

This appendix provides detailed derivations for causal innovation information update equations for RT-CK-RXD. Following the same treatment derived for RT-CR-RXD in Section III.A, we can derive RT-CK-RXD as follows.

Let $\boldsymbol{\mu}(n) = (1/n)\sum_{i=1}^{n} \mathbf{r}_i$ and $\mathbf{K}(n) = (1/n)\sum_{i=1}^{n} (\mathbf{r}_i - \boldsymbol{\mu}(n))(\mathbf{r}_i - \boldsymbol{\mu}(n))^T$. We can derive these as follows:

$$\begin{aligned}\boldsymbol{\mu}(n) &= (1/n)\sum_{i=1}^{n} \mathbf{r}_i = (1/n)\sum_{i=1}^{n-1} \mathbf{r}_i + (1/n)\mathbf{r}_n = ((n-1)/n)\,(1/(n-1))\sum_{i=1}^{n-1} \mathbf{r}_i + (1/n)\mathbf{r}_n \\ &= (1-1/n)\,\boldsymbol{\mu}(n-1) + (1/n)\mathbf{r}_n.\end{aligned} \tag{11}$$

$$\boldsymbol{\mu}(n)\boldsymbol{\mu}^T(n) = ((n-1)/n)^2\left[\boldsymbol{\mu}(n-1)\boldsymbol{\mu}^T(n-1)\right] + \left((n-1)/n^2\right)\left[\boldsymbol{\mu}(n-1)\mathbf{r}_n^T + \mathbf{r}_n\boldsymbol{\mu}^T(n-1)\right] + (1/n^2)\mathbf{r}_n\mathbf{r}_n^T \tag{12}$$

$$\begin{aligned}\mathbf{K}(n) &= \mathbf{R}(n) - \boldsymbol{\mu}(n)\boldsymbol{\mu}^T(n) = (1/n)\sum_{i=1}^{n} \mathbf{r}_i\mathbf{r}_i^T - \boldsymbol{\mu}(n)\boldsymbol{\mu}^T(n) = (1/n)\sum_{i=1}^{n-1} \mathbf{r}_i\mathbf{r}_i^T + (1/n)\mathbf{r}_n\mathbf{r}_n^T - \boldsymbol{\mu}(n)\boldsymbol{\mu}^T(n) \\ &= ((n-1)/n)\mathbf{R}(n-1) + (1/n)\mathbf{r}_n\mathbf{r}_n^T - ((n-1)/n)^2\left[\boldsymbol{\mu}(n-1)\boldsymbol{\mu}^T(n-1)\right] - ((n-1)/n^2)\left[\boldsymbol{\mu}(n-1)\mathbf{r}_n^T + \mathbf{r}_n\boldsymbol{\mu}^T(n-1)\right] - (1/n^2)\mathbf{r}_n\mathbf{r}_n^T \\ &= ((n-1)/n)\left\{\mathbf{R}(n-1) - ((n-1)/n)\left[\boldsymbol{\mu}(n-1)\boldsymbol{\mu}^T(n-1)\right]\right\} - \left((n-1)/n^2\right)\left[\boldsymbol{\mu}(n-1)\mathbf{r}_n^T + \mathbf{r}_n\boldsymbol{\mu}^T(n-1)\right] + ((n-1)/n^2)\mathbf{r}_n\mathbf{r}_n^T \\ &= ((n-1)/n)\left\{\mathbf{K}(n-1) + (1/n)\left[\boldsymbol{\mu}(n-1)\boldsymbol{\mu}^T(n-1)\right]\right\} - \left((n-1)/n^2\right)\left[\boldsymbol{\mu}(n-1)\mathbf{r}_n^T + \mathbf{r}_n\boldsymbol{\mu}^T(n-1)\right] + ((n-1)/n^2)\mathbf{r}_n\mathbf{r}_n^T \\ &= ((n-1)/n)\,\mathbf{K}(n-1) + \left((n-1)/n^2\right)\left[\boldsymbol{\mu}(n-1)\boldsymbol{\mu}(n-1)^T - \boldsymbol{\mu}(n-1)\mathbf{r}_n^T - \mathbf{r}_n\boldsymbol{\mu}^T(n-1) + \mathbf{r}_n\mathbf{r}_n^T\right] \\ &= (1-1/n)\,\mathbf{K}(n-1) + \left((n-1)/n^2\right)\left[(\mathbf{r}_n - \boldsymbol{\mu}(n-1))\,(\mathbf{r}_n - \boldsymbol{\mu}(n-1))^T\right]\end{aligned} \tag{13}$$

So, the same derivation for $\mathbf{R}^{-1}(n)$ in (6) can be applied to $\mathbf{K}(n)$ via Woodbury's matrix identity in (4) by setting $\mathbf{A} = (1-1/n)\,\mathbf{K}(n-1)$ and $\mathbf{u} = \mathbf{v} = \left[\left(\sqrt{n-1}/n\right)(\boldsymbol{\mu}(n-1) - \mathbf{r}_n)\right]$:

$$\mathbf{r}_n - \boldsymbol{\mu}(n) = \mathbf{r}_n - ((n-1)/n)\,(1/(n-1))\sum_{i=1}^{n-1} \mathbf{r}_i - (1/n)\mathbf{r}_n = \mathbf{r}_n - (1-1/n)\,\boldsymbol{\mu}(n-1) - (1/n)\mathbf{r}_n \tag{14}$$

$$
\begin{aligned}
&\delta^{\text{RT-CK-RXD}}(\mathbf{r}_n) = (\mathbf{r}_n - \boldsymbol{\mu}(n))^T\, \mathbf{K}^{-1}(n)\, (\mathbf{r}_n - \boldsymbol{\mu}(n)) \\
&= (\mathbf{r}_n - \boldsymbol{\mu}(n))^T \left\{(1-1/n)\, \mathbf{K}(n-1) + \left((n-1)/n^2\right) \left[(\mathbf{r}_n - \boldsymbol{\mu}(n-1))\,(\mathbf{r}_n - \boldsymbol{\mu}(n-1))^T\right]\right\}^{-1} (\mathbf{r}_n - \boldsymbol{\mu}(n)) \\
&= (\mathbf{r}_n - \boldsymbol{\mu}(n))^T\, [(1-1/n)\, \mathbf{K}(n-1)]^{-1}\, (\mathbf{r}_n - \boldsymbol{\mu}(n)) \\
&-(\mathbf{r}_n - \boldsymbol{\mu}(n))^T \frac{\left[\{(1-1/n)\mathbf{K}(n-1)\}^{-1}\left(\sqrt{n-1}/n\right)(\mathbf{r}_n - \boldsymbol{\mu}(n-1))\right]\left[\left(\sqrt{n-1}/n\right)(\mathbf{r}_n - \boldsymbol{\mu}(n-1))^T\{(1-1/n)\mathbf{K}(n-1)\}^{-1}\right]}{1+\left[(\sqrt{n-1}/n)(\mathbf{r}_n - \boldsymbol{\mu}(n-1))^T\{(1-1/n)\mathbf{K}(n-1)\}^{-1}(\sqrt{n-1}/n)(\mathbf{r}_n - \boldsymbol{\mu}(n-1))\right]}(\mathbf{r}_n - \boldsymbol{\mu}(n)) \\
&= (1-1/n)^{-1}\, (\mathbf{r}_n - \boldsymbol{\mu}(n))^T\, [\mathbf{K}(n-1)]^{-1}\, (\mathbf{r}_n - \boldsymbol{\mu}(n)) \\
&- \frac{(1-1/n)^{-2}\left(\sqrt{n-1}/n\right)^2 (\mathbf{r}_n - \boldsymbol{\mu}(n))^T \left[\{\mathbf{K}(n-1)\}^{-1}\,(\mathbf{r}_n - \boldsymbol{\mu}(n-1))\right]\left[(\mathbf{r}_n - \boldsymbol{\mu}(n-1))^T\, \{\mathbf{K}(n-1)\}^{-1}\right](\mathbf{r}_n - \boldsymbol{\mu}(n))}{1+(1-1/n)^{-1}\left(\sqrt{n-1}/n\right)^2\left[(\mathbf{r}_n - \boldsymbol{\mu}(n-1))^T\, \{\mathbf{K}(n-1)\}^{-1}\,(\mathbf{r}_n - \boldsymbol{\mu}(n-1))\right]} \qquad (15)
\end{aligned}
$$

REFERENCES

[1] Chang, C.-I
Hyperspectral Imaging: Techniques for Spectral Detection and Classification. The Netherlands: Kluwer Academic/Plenum, 2003.

[2] Reed, I. S., and Yu, X.
Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution.
IEEE Transactions on Acoustics, Speech and Signal Processing, **38** (1990), 1760–1770.

[3] Matteoli, S., Diani, M., and Corsini, G.
A tutorial overview of anomaly detection in hyperspectral images.
IEEE Aerospace and Electronic Systems Magazine, **25**, 7 (2010), 5–27.

[4] Ashton, E. A., and Schaum, A.
Algorithms for the detection of sub-pixel targets in multispectral imagery.
Photogrammetric Engineering and Remote Sensing, **64** (July 1998), 723–731.

[5] Stellman, C. M., Hazel, G. G., Bucholtz, F., Michalowicz, J. V., Stocker, A., and Scaaf, W.
Real-time hyperspectral detection and cuing.
Optical Engineering, **39**, 7 (July 2000), 1928–1935.

[6] Chiang, S.-S., Chang, C.-I, and Ginsberg, I. W.
Unsupervised subpixel target detection for hyperspectral images using projection pursuit.
IEEE Transactions on Geoscience and Remote Sensing, **39**, 7 (July 2001), 1380–1391.

[7] Stein, D. W., Beaven, S. G., Hoff, L. E., Winter, E. M., Schaum, A. P., and Stocker, A. D.
Anomaly detection from hyperspectral imagery.
IEEE Signal Processing Magazine, **19**, 1 (Jan. 2002), 58–69.

[8] Manolakis, D., and Shaw, G.
Detection algorithms for hyperspectral imaging applications.
IEEE Signal Processing Magazine, **19**, 1 (Jan. 2002), 29–43.

[9] Chang, C.-I, and Chiang, S.-S.
Anomaly detection and classification for hyperspectral imagery.
IEEE Transactions on Geoscience and Remote Sensing, **40**, 2 (Feb. 2002), 1314–1325.

[10] Kwon, H., Der, S. Z., and Nasrabadi, N. M.
Adaptive anomaly detection using subspace separation for hyperspectral imagery.
Optical Engineering, **42**, 11 (Nov. 2003), 3342–3351.

[11] Kwon, H., and Nasrabadi, N. M.
Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery.
IEEE Transactions on Geoscience and Remote Sensing, **43**, 2 (Feb. 2005), 388–397.

[12] Chang, C.-I, and Hsueh, M.
Characterization of anomaly detection for hyperspectral imagery.
Sensor Review, **26**, 2 (2006), 137–146.

[13] Liu, W., and Chang, C.-I
A nested spatial window-based approach to target detection for hyperspectral imagery.
In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Anchorage, AK, Sept. 20–24, 2004.

[14] Ren, H., and Chang, C.-I
Automatic spectral target recognition in hyperspectral imagery.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 4 (Oct. 2003), 1232–1249.

[15] Ren, H., Du, Q., Wang, J., Chang, C.-I, and Jensen, J.
Automatic target recognition hyperspectral imagery using high order statistics.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 4 (Oct. 2006), 1372–1385.

[16] Chang, C.-I, Ren, H., and Chiang, S. S.
Real-time processing algorithms for target detection and classification in hyperspectral imagery.
IEEE Transactions on Geoscience and Remote Sensing, **39**, 4 (Apr. 2001), 760–768.

[17] Du, Q., and Ren, H.
Real-time constrained linear discriminant analysis to target detection and classification in hyperspectral imagery.
Pattern Recognition, **36**, 1 (2003), 1–12.

[18] Du, Q., and Nevovei, R.
Implementation of real-time constrained linear discriminant analysis to remote sensing image classification in hyperspectral imagery.
Pattern Recognition, **38**, 4 (2005), 1–12.

[19] Du, Q., and Nekovei, R.
Fast real-time onboard processing of hyperspectral imagery for detection and classification.
Journal of Real-Time Image Processing, **4**, 3 (2009), 273–286.

[20] Tarabalka, Y., Haavardsholmm T. V., Kasen, I., and Skauli, T.
Real-time anomaly detection in hyperspectral images using multivariate normal mixture models and GPU processing.
Journal of Real-Time Image Processing, **4**, 3 (2009), 287–300.

[21] Poor, H. V.
An Introduction to Detection and Estimation. New York, NY: Springer-Verlag, 1994.

[22] Kailath, T.
An innovations approach to least squares estimation: I. Linear filtering in additive white noise.
IEEE Transactions on Automatic Control, **AC-13** (Dec. 1968), 646–655.

[23] Kailath, T.
Linear Systems. Upper Saddle Rive, NJ: Prentice Hall, 1980.

[24] Harsanyi, J. C.
Detection and Classification of Subpixel Spectral Signatures in Hyperspectral Image Sequences. Baltimore, MD: Department of Electrical Engineering, University of Maryland, Baltimore County, Aug. 1993.

[25] Schowengerdt, R. A.
Remote Sensing: Models and Methods for Image Processing (2nd ed.). Salt Lake City, Utah: Academic Press, 1997.

[26] Schultz, R., Chen, S. Y., Wang, Y., Liu, C., and Chang, C.-I.
Progressive band processing of anomaly detection.
In *Proceedings of the SPIE Conference on Satellite Data Compression, Communication and Processing IX* (OP 405), San Diego, CA, Aug. 25–29, 2013.

[27] Chang, Y.-C., Ren, H., Chang, C.-I, and Rand, B.
How to design synthetic images to validate and evaluate hyperspectral imaging algorithms.
In *Proceedings of the SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIV*, Orlando, FL, Mar. 16–20, 2008.

[28] Wu, C. C., Lo, C. S., and Chang, C.-I
Improved process for use of a simplex growing algorithm for endmember extraction.
IEEE Geoscience and Remote Sensing Letters, **6** (July 2009), 523–527.

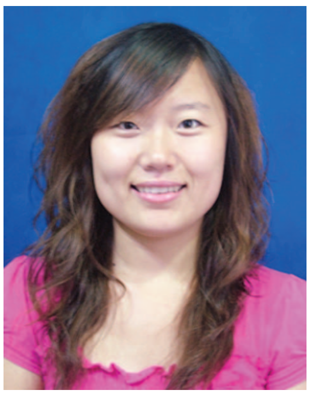
[29] Chang, C.-I
Hyperspectral Data Processing: Algorithm Design and Analysis. Hoboken, NJ: Wiley, 2013.

[30] U.S. Geological Survey Cuprite, Nevada research papers, http://speclab.cr.usgs.gov/cuprite.html, Feb. 1999.

[31] Harsanyi, J. C., and Chang, C.-I
Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach.
IEEE Transactions on Geoscience and Remote Sensing, **32** (July 1994), 779–785.

[32] Chang, C.-I
Multiple-parameter receiver operating characteristic analysis for signal detection and classification.
IEEE Sensors Journal, **10** (Mar. 2010), 423–442.

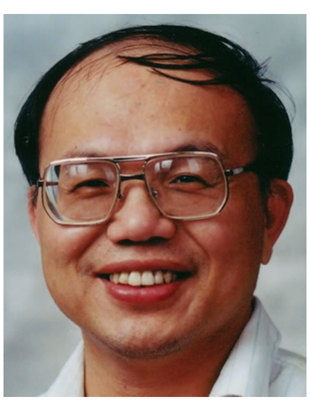
Shih-Yu Chen received a B.S. degree in electrical engineering from Da-Yeh University in 2005, an M.S.E.E. degree from National Chung Hsing University in 2010, and Ph.D. in E.E. from University of Maryland, Baltimore County in 2014. Dr. Chen's research interests include medical images, remote sensing images, and vital sign signal processing.



Yulei Wang is a Ph.D. student in the College of Information and Communication Engineering, Harbin Engineering University. She was awarded by China Scholarship Council in 2011 an opportunity, as a visiting Ph.D. Student, to study in the Remote Sensing Signal and Image Processing Laboratory, University of Maryland, Baltimore County. Her current research interests include hyperspectral image processing and vital signs signal processing.

Chao-Cheng Wu received a B.S. degree in electrical engineering from Tamkang University, Taipei, Taiwan, in 2002 and M.S. and Ph.D. degrees in electrical engineering from University of Maryland, Baltimore County, in 2006 and 2009, respectively. Dr. Wu is an assistant professor in the Department of Electrical Engineering, National Taipei University of Technology. His research interests include multispectral or hyperspectral image processing, medical signal and image processing, and pattern recognition.

Liu Chunhong is an associate professor of the Information and Electrical Engineering Department at China Agricultural University. She received her Ph.D. degree in signal and information processing from Harbin Engineering University in 2005. She has projects being supported by the China Postdoctoral Science Foundation and National Natural Science Fund of China. She has also conducted more than 10 projects in information and communication fields. Her research interests include target detection, transmission, and processing in hyperspectral images.



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

Dr. Chang has been with University of Maryland, Baltimore County, since 1987 and is a professor in the Department of Computer Science and Electrical Engineering. He was a visiting research specialist in the Institute of Information Engineering at the National Cheng Kung University, Tainan, Taiwan, from 1994 to 1995. In addition, he was a distinguished lecturer chair at the National Chung Hsing University, Taichung, Taiwan, sponsored by the Ministry of Education in Taiwan from 2005 to 2006. He was a chair professor of the Environmental Restoration and Disaster Reduction Research Center and the Department of Electrical Engineering of the National Chung Hsing University and has been the chair professor of remote sensing technology at the same institute since 2009. Currently, he is also chair professor of Providence University, Taichung, Taiwan, and international chair professor of National Taipei University of Technology, Taipei, Taiwan. He was a distinguished visiting fellow and fellow professor sponsored by the National Science Council in Taiwan from 2009 to 2010. He was a plenary speaker for the 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 at the Industrial Technology Research Institute, Hsinchu; the 2009 Annual Meeting of the Radiological Society of the Republic of China in Taichung; the 2008 International Symposium on Spectral Sensing Research; and the Conference on Computer Vision, Graphics, and Image Processing in 2003 in Kimen and in 2013 in Nan-Tou, Taiwan. He holds five patents and has several pending on hyperspectral image processing. He was the guest editor of a special issue of the *Journal of High Speed Networks* on telemedicine and applications (April 2000) and coguest editor of another special issue of the same journal on broadband multimedia sensor networks in healthcare applications (April 2007). He was 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. He has written 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 coedited with A. Plaza the book *High Performance Computing in Remote Sensing* (CRC Press, 2007). He is working on a third book, *Real Time Hyperspectral Image Processing: A Progressive Hyperspectral Imaging Perspective* (Springer-Verlag, 2015). His research interests include multispectral or hyperspectral image processing, automatic target recognition, and medical imaging.

Dr. Chang is a fellow of SPIE. He was an associate editor in the area of hyperspectral signal processing for *IEEE Transactions on Geoscience and Remote Sensing* in 2001–2007. He is an associate editor of *Artificial Intelligence Research* and is 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, Maryland.