ANOMALY DETECTION USING SLIDING CAUSAL WINDOWS

Yulei Wang[1, 2], student member, IEEE, Chunhui Zhao[1] and Chein-I Chang[2], Fellow, IEEE

[1]Information and Communication Engineering College, Harbin Engineering University, Harbin, China
[2]Remote Sensing Signal and Image Processing Laboratory, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250 USA

ABSTRACT

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

Index Terms—Causal anomaly detection, Causal matrix window, Causal window, Causal array window, K-RXD, R-RXD

1. INTRODUCTION

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

2. CAUSAL ANOMALY DETECTION

In this section we design a new type of anomaly detectors derived from K-RXD and R-RXD, which can adjust K or R

dynamically to capture sample by sample changes in background so as to achieve sample varying background suppression as opposed to sample invariant anomaly detectors such as K-RXD and R-RXD. It is called causal anomaly detectors which are defined as follows.

2.1 Causal Matrix Windows

Let $\{\mathbf{r}_i\}_{i=1}^{n}$ be a set of data sample vectors up to the currently being processed data sample vector $\mathbf{r}_n$. Assume that a sliding square window W is specified by a fixed size of $w^2 = w \times w$. This sliding window has its center located at the currently being processed sample $\mathbf{r}_n$ and moved along with $\mathbf{r}_n$ as the process goes on. A sliding causal matrix window with size given by $a = (w^2 - 1)/2$ is then defined by a window which includes only all data sample vectors in the square window W with size of $w^2 = w \times w$ that precede the $\mathbf{r}_n$ and have been visited, while a non-causal matrix window with size of $a = (w^2 - 1)/2$ includes only those future data sample vectors yet to be visited within the square window W. Fig. 1 illustrates its concept by specifying the window W with size of $w \times w = 5 \times 5$ and $a = (w^2 - 1)/2 = 12$ where the causal matrix window comprise of all the data sample vectors $\{\mathbf{r}_i\}_{i=n-a}^{n-1}$ with the size of a = 12 pixels and the non-causal matrix window is also made up of all the future data sample vectors yet to be processed in the W, $\{\mathbf{r}_i\}_{i=n+1}^{n+a}$ with size of a = 12 pixels.

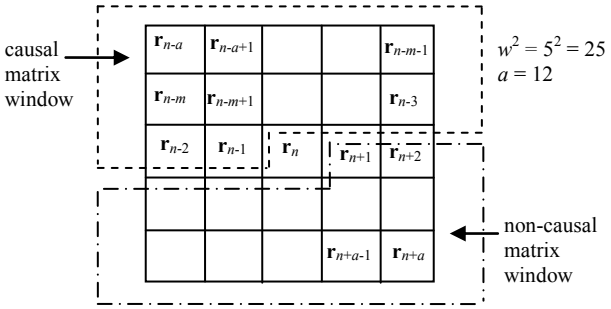


Figure 1. Causal and non-causal of a window with size $w \times w$

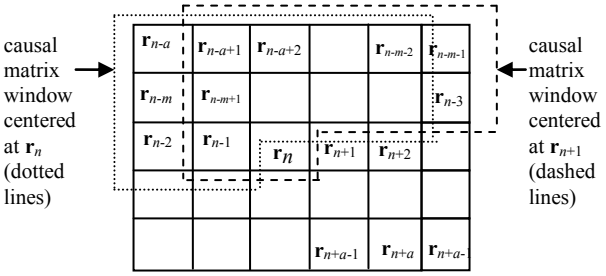


Figure 2. causal matrix windows at $\mathbf{r}_n$ and $\mathbf{r}_{n+1}$

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

2.2 Causal Array Windows

As we can see from Fig. 2, all data sample vectors excluded from the causal matrix window are not removed in sequence. For example, in Fig. 2 the $\mathbf{r}_{n-a}$, $\mathbf{r}_{n-m}$ and $\mathbf{r}_{n-2}$ in the causal matrix window centered at $\mathbf{r}_n$ are removed from the causal matrix window centered at $\mathbf{r}_{n+1}$, while $\mathbf{r}_{n-m-1}$ and $\mathbf{r}_{n-3}$ which are not included in the causal matrix window centered at $\mathbf{r}_n$ are now added to the causal matrix window centered at $\mathbf{r}_{n+1}$. Obviously, it requires bookkeeping to keep track of which pixels should be removed and which pixels should added as a causal matrix window moves on. In order to resolve this issue we can stretch out the causal matrix window in Fig. 1 as a linear array shown in Fig. 3.

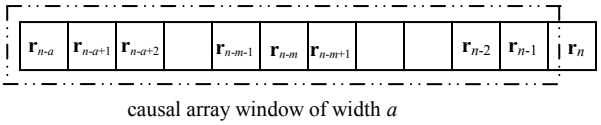


Figure 3. Causal array window with width specified by a

By virtue of Fig. 3 we define a causal array window of width a sliding along with a processed pixel $\mathbf{r}_n$ as a linear array, which embraces a pixels, $\{\mathbf{r}_i\}_{i=n-a}^{n-1}$, preceding the processed pixel $\mathbf{r}_n$. In other words, the causal array window of width a defined in Fig. 3 is formed by a linear array which consists of pixels preceding the current processed pixel $\mathbf{r}_n$. It is no longer a square window shown in Fig. 1. It should be also noted that the current pixel $\mathbf{r}_n$ is not included in the causal array window. So, when a causal array window moves along with the pixels, the linear array simply performs like a queue, first in and first out. Fig. 4 shows the causal array window at $\mathbf{r}_n$ depicted by dotted lines and the causal array window at $\mathbf{r}_{n+1}$ depicted by dashed lines where the farthest pixel $\mathbf{r}_{n-a}$ in the causal array window at $\mathbf{r}_n$ is removed from the causal array window, while the most recent data sample vector $\mathbf{r}_n$ is then added to the causal array window at $\mathbf{r}_{n+1}$.

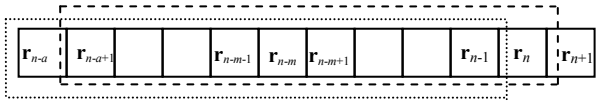


Figure 4. Causal array window at $\mathbf{r}_{n+1}$ with width specified by a

The difference between the causal matrix window shown in Fig. 2 and the causal array window shown in Fig. 4 is that the former includes $\mathbf{r}_n$, $\mathbf{r}_{n-m-1}$, $\mathbf{r}_{n-3}$ but excludes $\mathbf{r}_{n-a}$, $\mathbf{r}_{n-m}$ and $\mathbf{r}_{n-2}$, while the latter includes $\mathbf{r}_n$ but excludes $\mathbf{r}_{n-a}$.

2.3 Causal Anomaly Detection

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

$$\delta^{\text{CR-RXD}}(\mathbf{r}_n) = \mathbf{r}_n^T \widetilde{\mathbf{R}}(n)^{-1} \mathbf{r}_n \tag{4}$$

where $\mathbf{r}_n$ is the n^{th} data sample vector currently being processed. $\widetilde{\mathbf{R}}(n)$ is called "*causal*" sample correlation matrix

formed by data sample vectors if it is defined by $\widetilde{\mathbf{R}}(n)=(1/n)\sum_{i=1}^{n}\mathbf{r}_i\mathbf{r}_i^T$.

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

$$\delta^{\text{CK-RXD}}(\mathbf{r}_n)=\left(\mathbf{r}_n-\widetilde{\boldsymbol{\mu}}(n)\right)^T\widetilde{\mathbf{K}}(n)^{-1}\left(\mathbf{r}_n-\widetilde{\boldsymbol{\mu}}(n)\right) \tag{5}$$

where $\widetilde{\boldsymbol{\mu}}(n)=(1/n)\sum_{i=1}^{n}\mathbf{r}_i$ is the "*causal*" sample mean of samples $\{\mathbf{r}_i\}_{i=1}^{n-1}$ and $\widetilde{\mathbf{K}}(n)=(1/n)\sum_{i=1}^{n}(\mathbf{r}_i-\widetilde{\boldsymbol{\mu}}(n))(\mathbf{r}_i-\widetilde{\boldsymbol{\mu}}(n))^T$ is the *"causal"* covariance matrix of samples $\{\mathbf{r}_i\}_{i=1}^{n-1}$.

3. RECURSIVE CAUSAL ANOMALY DETECTION

Technically speaking, (4) can be implemented in real time. However, the causal sample correlation, $\widetilde{\mathbf{R}}(n)$ in (4) varies with pixels to be processed and must be re-calculated each time when comes to a new pixel. This processing time generally goes beyond time constraints required for real time implementation. In order to resolve this issue, this section derives a cursive causal information update equation which only needs to update causal anomaly detection by including innovations information provided by the new data sample vector and its correlation with processed information obtained from previous data sample vectors.

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

$$\widetilde{\mathbf{R}}_a(n+1)=\left[\left(\widetilde{\mathbf{R}}_a(n)-\mathbf{r}_{n-a}\mathbf{r}_{n-a}^T\right)+\mathbf{r}_n\mathbf{r}_n^T\right]. \tag{6}$$

Now, in order to calculate the inverse of $\widetilde{\mathbf{R}}_a(n+1)$, i.e., $\widetilde{\mathbf{R}}_a^{-1}(n+1)$ we repeatedly make use of the following Woodbury matrix identity [3] twice

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

to first bring out $\mathbf{r}_n\mathbf{r}_n^T$ with $\mathbf{A}=\left(\widetilde{\mathbf{R}}_a(n)-\mathbf{r}_{n-a}\mathbf{r}_{n-a}^T\right)$ and $\mathbf{u}=\mathbf{v}=\mathbf{r}_n$; then bring out $\mathbf{r}_{n-a}\mathbf{r}_{n-a}^T$ by letting $\mathbf{A}=\widetilde{\mathbf{R}}_a(n)$ and $\mathbf{u}=\mathbf{v}=\mathbf{r}_{n-a}$ as follows:

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

where $\left(\widetilde{\mathbf{R}}_a(n)-\mathbf{r}_{n-a}\mathbf{r}_{n-a}^T\right)^{-1}$ can be further updated recursively by

$$\left(\widetilde{\mathbf{R}}_a(n)-\mathbf{r}_{n-a}\mathbf{r}_{n-a}^T\right)^{-1}=\widetilde{\mathbf{R}}_a^{-1}(n)-\frac{\left[\widetilde{\mathbf{R}}_a^{-1}(n)\mathbf{r}_{n-a}\right]\left[\mathbf{r}_{n-a}^T\widetilde{\mathbf{R}}_a^{-1}(n)\right]}{1+\mathbf{r}_{n-a}^T\widetilde{\mathbf{R}}_a^{-1}(n)\mathbf{r}_{n-a}} \tag{9}$$

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

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

4. REAL IMAGE EXPERIMENTS

In order to see whether this theory works in real hyperspectral imagery, a size of 200×74 pixels HYDICE image scene shown in Fig. 5(a) along with its ground truth provided in Fig. 5(b) where the center and boundary pixels of objects are highlighted by red and yellow respectively is used for the experiment. There are several advantages of using this HYDICE image scene in Fig. 5(a). The scene has various sizes of objects that can be used to evaluate ability of an anomaly detector in detecting anomalies with different sizes, an issue that has not been really addressed in many reports. Most importantly, the clean natural background and targets make visual assessment more easily to see various degrees of background be suppressed by an anomaly detector.

In order to verify the effectiveness of local causal anomaly detectors, two commonly used global anomaly detectors, K-RXD and R-RXD are implemented first, shown in Fig. 5(c) - (d). The global anomaly detectors have very good performance especially for the panels of the upper part. But they cannot be implemented in real-time due to the calculation of global covariance matrix or correlation matrix formed by the entire image data samples.

Anomaly detection using causal sliding windows is implemented in a real-time and causal manner. This causal anomaly detector is different from the commonly used dual local detectors with inner window and outer window centered by the pixel being processed. Due to the need of real-time process, the local window is designed causally which only uses pixels in a fixed size causal array window up to the data sample being processed. It should be noted that the width of the causal array window must be greater or equal to the total band number to avoid a singularity problem in the inversion of the correlation matrix. In order to see how anomaly detection using causal array windows of various widths from 200 up 900 with step size of 100 pixels, Fig. 6(a-h) shows the detection abundance fractional maps.

According to our experiments, the detection result was very poor using causal array window width = 200, while the performance began to improve as the causal array window width increases. When the causal array window width becomes very large, the detection performance were similar

by visual inspection as shown in Fig. 6(e)-(h), with the causal array window width greater than or equal to 600.

5. CONCLUSION

Anomaly detection using causal sliding windows has not been explored in the past. Despite that many local and adaptive anomaly detectors have been proposed to claim to be real time they actually not causal detectors. So, technically speaking they cannot be implemented in real time. This is particular true for those which use sliding windows. This paper develops a new approach to designing anomaly detection using causal sliding windows to satisfy the necessity of real time processing and further provides a feasibility of hardware implementation for future FPGA design.

6. ACKNOWLEDGMENT

This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation, the National Natural Science Foundation of China (No.61077079, No.61275010), the Ph.D. Programs Foundation of Ministry of Education of China (No.20132304110007), the Key Program of Heilongjiang Natural Science Foundation (No.ZD201216) and the Fundamental Research Funds for the Central Universities (No.HEUCF1208) and the excellent academic leader Foundation of Harbin City in China (RC2013XK009003).

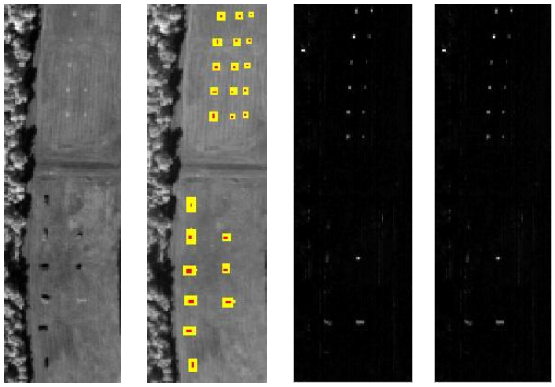
7. REFERENCES

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

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

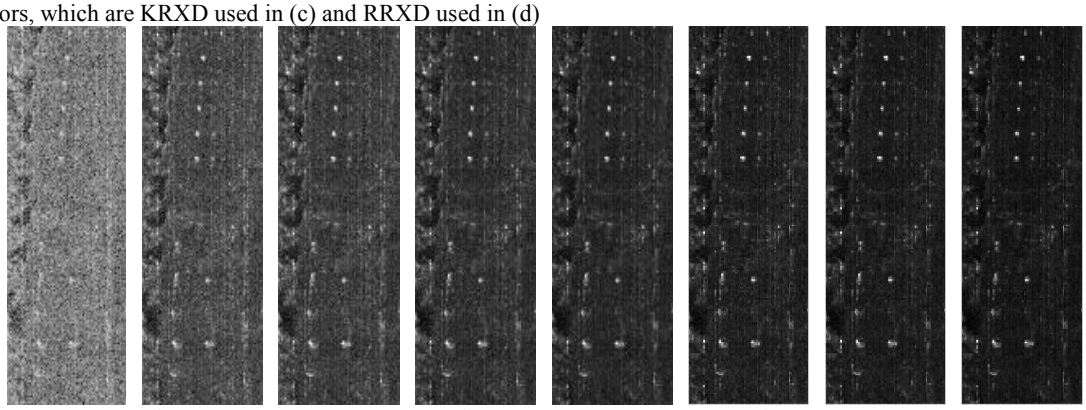
[3] H.V. Poor, *An Introduction to Detection and Estimation*, Springer-Verlag, 1994.

[4] C.-I Chang and M. Hsueh, "Characterization of anomaly detection for hyperspectral imagery," *Sensor Review*, vol. 26, no. 2, pp. 137-146, 2006.



(a) (b) (c) (d)

Figure 5. (a-b) HYDICE panel-vehicle scene in (a) and its ground truth in (b), (c-d) abundance fractional maps by two commonly used global anomaly detectors, which are KRXD used in (c) and RRXD used in (d)



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6. Detection abundance fractional maps by causal anomaly detectors with different causal array window width where (a) width = 200 pixels, (b) width 300, (c) width = 400, (d) width = 500, (e) width = 600, (f) width = 700, (g) width = 800, (h) width = 900.