

基于逐行处理的高光谱实时异常目标检测

赵春晖, 邓伟伟, 姚浙峰

哈尔滨工程大学信息与通信工程学院, 黑龙江 哈尔滨 150001

摘要 实时处理可以缓解海量高光谱数据在存储及下行传输方面带来的巨大压力,在高光谱异常检测领域引起了研究人员的广泛关注。高光谱成像传感器通过推扫获取数据的方式已成为主流,因此,提出了一种基于逐行处理框架的高光谱实时异常目标检测算法。将局部因果窗模型引入 Reed-Xiaoli 异常检测算法中,通过滑动局部因果窗来检测异常目标,保证了实时处理的因果性。针对矩阵求逆过程复杂度过大的问题,在卡尔曼滤波器递归思想的基础上,利用 Woodbury 求逆引理,由前一时刻数据状态信息迭代更新当前数据的状态信息,避免了大矩阵的求逆运算,减少了算法的计算量。利用模拟和真实高光谱数据进行实验,结果表明,在保持检测精度不变的前提下,提出的实时算法的运算效率相比于原始算法得到显著提高。

关键词 遥感; 高光谱异常目标检测; 实时算法; 逐行处理; 局部因果窗

中图分类号 TP751.1 **文献标识码** A

doi: 10.3788/AOS201737.0128002

Hyperspectral Real-Time Anomaly Target Detection Based on Progressive Line Processing

Zhao Chunhui, Deng Weiwei, Yao Xifeng

*College of Information and Communication Engineering, Harbin Engineering University,
Harbin, Heilongjiang 150001, China*

Abstract Real-time processing can reduce the pressure of data storage and downlink transmission caused by the ever-expanding hyperspectral dataset, which has received more and more attention in hyperspectral anomaly detection. Since acquiring data with pushbroom has become main stream for hyperspectral imaging sensors, a real-time anomaly target detection method is proposed based on the framework of progressive line processing. In order to make sure the causality of real-time processing, the local causal window model is introduced into the Reed-Xiaoli anomaly detection algorithm, and the sliding local causal window is used to detect anomaly targets. In terms of the high computational complexity caused by the inversion of matrix, the recursive principle of the Kalman filter and the Woodbury's lemma are employed to update the status information of current data through iterating data status information at the previous moment, which avoids the inversion of large matrix. The simulated and real hyperspectral data are adopted for the experiment. The results show that under the premise of maintaining the detection accuracy, the proposed real-time algorithm improves the processing efficiency significantly compared with the original algorithm.

Key words remote sensing; hyperspectral anomaly target detection; real-time algorithm; progressive line processing; local causal window

OCIS codes 280.1100; 100.4145; 100.3190; 100.5010

1 引 言

随着光谱分辨率及光谱仪成像技术的不断提高,高光谱遥感图像处理技术得到了深入的研究与发展。高光谱遥感将图像的空间信息与光谱特征相结合^[1-3],探测地物光谱之间的差异,进行目标的检测与识别。

收稿日期: 2016-06-27; **收到修改稿日期:** 2016-09-12

基金项目: 国家自然科学基金(61405041,61571145)、黑龙江省自然科学基金重点项目(ZD201216)、哈尔滨市优秀学科带头人基金(RC2013XK009003)

作者简介: 赵春晖(1965—),男,博士,教授,博士生导师,主要从事高光谱遥感处理技术方面的研究。

E-mail: zhaochunhui@hrbeu.edu.cn

高光谱目标检测技术作为高光谱遥感图像处理中的一个主要分支,在井下勘探、军事侦察、农林监控等领域得到了广泛应用^[4]。然而,在多数情况下,图像数据的先验信息并不能准确得知,因此异常目标检测在实际应用中显得尤为重要。

常用的高光谱异常目标检测算法主要有 Reed-Xiaoli(RX)算法、低概率目标探测(LPTD)算法及均衡目标探测(UTD)算法等^[5],其中最为经典的是由 Reed 和 Xiaoli 提出的基于广义似然比假设检验模型的 RX 异常目标检测算法(RXD)^[6],这是一种基于样本协方差矩阵的经典 RX 算法,假设背景数据服从高斯分布的前提下,利用马氏距离来寻找异常目标。

然而,RXD 异常检测算法非常耗时,难以快速捕捉运动目标,并且数据处理时需要用到全部样本像元的统计信息,给数据的下行传输和后处理带来一定的压力。因此,Chen 等^[7]提出了一种基于逐像元处理的高光谱实时异常目标检测算法,在降低算法存储空间的同时,提高了算法的运算效率。Chang 等^[8]以逐波段处理的方式,提出了一种高光谱实时异常目标检测算法,通过对波段序列的递归更新,实现了算法的实时检测。目前,大多数高光谱实时异常检测算法都是在逐像元或逐波段的基础上完成的,而成像光谱仪空间成像方式主要为推扫式,利用面阵探测器逐行获取待测地物空间中样本像元的光谱信息,因此实现基于逐行处理的高光谱实时异常检测算法更具有实际价值。Rossi 等^[9]基于推扫式光谱仪提出了一种利用线性代数更新协方差矩阵的局部 RX 异常目标检测算法,但是该算法的更新过程不具有因果性,而且在算法推导过程中丢失了某些信息。

基于此,本文将局部因果窗模型、卡尔曼滤波器的递归思想和 Woodbury 矩阵引理相结合,提出了一种基于逐行处理方式的高光谱实时异常目标检测(RT-RXD)算法。局部因果窗的逐行滑动,保证了算法在检测时仅利用当前检测像元之前的数据信息,而未用到之后的数据信息,实现了数据的同时传输和处理。卡尔曼滤波器的递归思想和 Woodbury 矩阵引理使得算法无需重复计算数据的统计信息,大大降低了算法的计算量,提高了算子的运算效率。通过理论分析和实验仿真,对提出的实时算法的检测性能进行详尽论证。

2 经典 RX 异常检测算法

作为高光谱异常检测中的经典算法,基于广义恒虚警率的 RX 算子自被 Reed 与 Xiaoli 提出后^[6],得到了深入的研究。RX 算子将图像背景建模为多元高斯分布,以待测像元为中心,在局部检测窗口内利用马氏距离来区分背景与目标,实现了异常点的检测。传统检测窗口如图 1 所示,待检测像元位于内窗中,背景则分布于外窗中,通过窗口的移动,实现对待检测样本像元的处理。

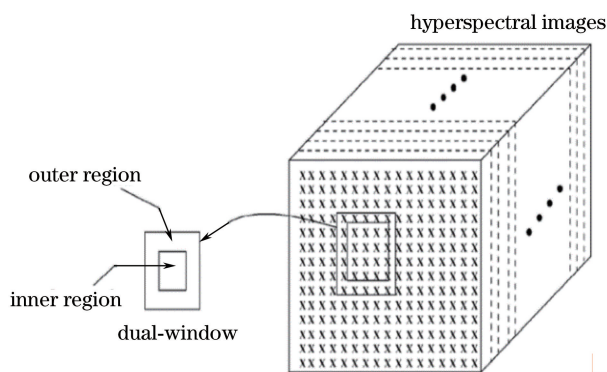


图 1 传统的局部双窗口模型

Fig. 1 Traditional local dual-window model

RX 算子可以看作是一个二值假设检验问题。具有 J 维波段的高光谱图像,第 i 个像元的光谱信号可表示为一个 J 维列向量 $\mathbf{x}_i = [x_{1i}, x_{2i}, \dots, x_{ji}]^T$,高光谱图像可以表示为一个二维 $M \times J$ 图像矩阵 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]^T$,每一个像元均为图像矩阵 \mathbf{X} 中的一个列向量, M 表示图像中所含样本像元的总数。二值假设检验可定义为

$$\begin{cases} \mathbf{x} | H_0: \mathbf{n} \in N(\boldsymbol{\mu}_b, \mathbf{C}_b) \\ \mathbf{x} | H_1: \mathbf{a}\mathbf{s} + \mathbf{n} \in N(\boldsymbol{\mu}_s, \mathbf{C}_b) \end{cases}, \quad (1)$$

式中在 H_0 条件下, $\mathbf{a} = \mathbf{0}$, 未检测到目标; 在 H_1 条件下, $\mathbf{a} > \mathbf{0}$, 检测到目标。 $\mathbf{s} = [s_1, s_2, \dots, s_J]^T$ 为目标的光谱信号, \mathbf{n} 为混有噪声的剩余背景信号, $\mathbf{a} = [a_1, a_2, \dots, a_J]^T$ 为信号的权值。在两种假设条件下, 二者均服从多维高斯分布, 其中 \mathbf{C}_b 为数据样本中未知的背景协方差矩阵, $\boldsymbol{\mu}_b$ 和 $\boldsymbol{\mu}_s$ 分别为数据样本的背景均值和目标均值。

由广义似然比检验可得, RX 异常检测算法的决策规则为

$$\delta^{\text{RXD}}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_b)^T \mathbf{C}_b^{-1} (\mathbf{x} - \boldsymbol{\mu}_b) \begin{cases} \geq \eta, & H_1 \text{ is tenable} \\ < \eta, & H_0 \text{ is tenable} \end{cases}, \quad (2)$$

式中 \mathbf{x} 为一个 $J \times 1$ 的待检测数据样本的光谱向量, η 为检测的门限值。

$$\boldsymbol{\mu}_b = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_i, \quad (3)$$

$$\mathbf{C}_b = \frac{1}{M} \sum_{i=1}^M (\mathbf{x}_i - \boldsymbol{\mu}_b) (\mathbf{x}_i - \boldsymbol{\mu}_b)^T. \quad (4)$$

3 实时 RX 异常检测算法

实时异常检测需要满足两个基本条件, 即因果性和时效性。因果性要求处理的数据仅与当前数据样本像元及其之前的所有数据样本像元有关, 而与之后尚未处理的数据样本像元无关。时效性则要保证算法的处理速度足够快, 以满足实际要求, 因为从实际应用的角度来说, 算法的时间延迟在不同的应用领域具有不同的定义。例如, 运动目标可能会突然出现并很快消失, 这就要求数据的处理时间非常短甚至可以忽略不计; 而农业中对于农作物的监测, 几分钟甚至几小时都可以认为是实时或近似实时的。为了满足上述条件, 采用 Woodbury 引理作为数学基础, 实现基于逐行处理的实时 RX 异常目标检测。

3.1 Woodbury 矩阵求逆引理

1950 年, Woodbury 提出了矩阵求逆引理, 也称 Woodbury 公式。Woodbury 指出可以利用原始矩阵的逆矩阵和 k 阶修正因子求得某些具有 k 阶修正因子的矩阵的逆矩阵^[10]。该引理的广泛应用解决了信号处理、自动控制和神经网络等领域的诸多难题^[10]。Woodbury 引理的简化形式为

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U} (\mathbf{I} + \mathbf{V}^T \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V}^T \mathbf{A}^{-1}, \quad (5)$$

式中 \mathbf{A} 为 $n \times n$ 的非奇异矩阵, \mathbf{U} 和 \mathbf{V} 均为 $n \times k$ 的矩阵。将 Woodbury 引理用于求解样本协方差矩阵的逆矩阵, 可有效降低算法的计算复杂度, 提高运算效率。

3.2 基于逐行处理的实时 RX 架构

目前, 基于局部检测的高光谱目标探测技术大多数采用传统的双窗口模型, 通过内外两个窗口将目标和背景进行分离, 以达到检测异常目标的目的。但传统的双窗口模型并不满足实时处理中因果性的要求, 为此, 采用一种具有因果特性的滑动窗模型^[11]。通过动态更新高光谱图像的局部协方差矩阵, 实现基于逐行处理方式的高光谱实时异常目标检测。

具有因果特性的滑动窗模型如图 2 所示。将整个矩形窗口分为左右两部分, 左半部分为当前待测像元之前的所有数据样本像元构成的因果矩阵窗(长度为 a , 宽度为 b), 右半部分为当前待测像元之后的所有数据样本像元构成的非因果矩阵窗。当图 3 中的因果滑动窗由检测像元 \mathbf{x}_n 滑动至 \mathbf{x}_{n+1} 时, 窗口中所包含的数据样本像元将发生变化, 如图 4 所示(实线框内是以 \mathbf{x}_n 为中心的因果滑动窗, 虚线框内是以 \mathbf{x}_{n+1} 为中心的因果滑动窗)。此时原窗口中顶行的所有元素 $(\mathbf{x}_{n-a}, \mathbf{x}_{n-m}, \mathbf{x}_{n-2})$ 将被移出, 而窗口外新的一行元素 $(\mathbf{x}_{n-a+5}, \mathbf{x}_{n-m+5})$ 将被移入当前的滑动窗中。值得注意的是, 移入和移出的数据样本像元均为当前检测像元之前的数据, 保证了数据实时处理的因果性。图 5 为局部逐行处理的 RX 实时异常检测算法示意图, 因果滑动窗沿推扫轨道的垂直方向进行滑动, 完成异常目标的检测。当因果滑动窗滑动至图像数据的两端时, 沿推扫轨道方向平移一行, 继续沿推扫轨道垂直方向进行滑动, 直至检测过程结束。因果滑动窗的环绕型移动避免了边界处理时对顶层窗口中数据信息的预存, 节约了算法的存储空间。

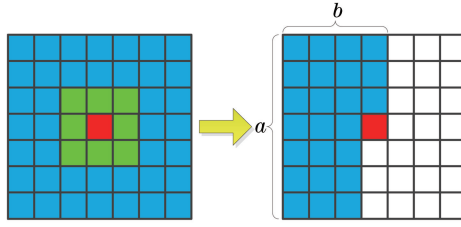


图 2 传统矩阵窗和因果滑动窗

Fig. 2 Traditional matrix window and causal sliding window

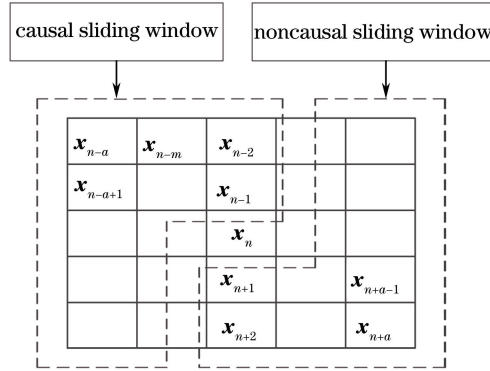


图 3 因果滑动窗模型

Fig. 3 Model of causal sliding window

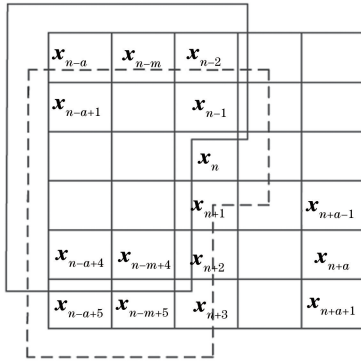


图 4 以 x_n 和 x_{n+1} 为中心的因果滑动窗

Fig. 4 Causal sliding window at centers of x_n and x_{n+1}

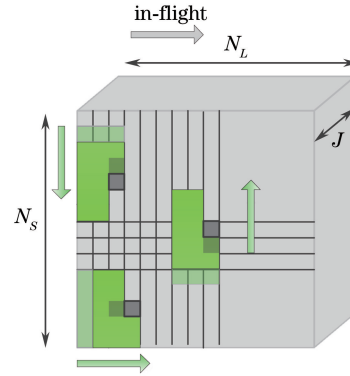


图 5 基于逐行处理的实时 RX 算法示意图

Fig. 5 Real-time RX algorithm diagram based on progressive line processing

3.3 RT-RXD 异常检测算子

为了表示算法的实时性,具有因果特性的 RX 算子(C-RXD)可以写为

$$\delta_w^{\text{C-RXD}}(\mathbf{x}_n) = [\mathbf{x}_n - \tilde{\boldsymbol{\mu}}(n)]^T \tilde{\mathbf{K}}(n)^{-1} [\mathbf{x}_n - \tilde{\boldsymbol{\mu}}(n)], \quad (6)$$

$$\tilde{\boldsymbol{\mu}}(n) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i, \quad (7)$$

$$\tilde{\mathbf{K}}(n) = \frac{1}{N} \sum_{i=1}^N [\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(n)] [\mathbf{x}_i - \tilde{\boldsymbol{\mu}}(n)]^T, \quad (8)$$

式中 \mathbf{x}_n 为当前局部检测窗中待检测样本像元的光谱特征, $\tilde{\boldsymbol{\mu}}(n)$ 为当前检测窗中数据样本像元的均值, N 为窗口中所含数据样本像元的总数, $\tilde{\mathbf{K}}(n)$ 为当前检测窗中数据样本像元的协方差矩阵。由(6)式可知, C-RXD算子虽能保证实时检测的因果性,但需要对局部窗中的协方差矩阵进行反复的矩阵求逆运算,使计算复杂度正比于矩阵的维数,降低了算法的运算效率。为此,本文利用 Woodbury 求逆引理,实现矩阵的递归更新。

假设 \mathbf{x}_{n-1} 为前一时刻局部检测窗中的待测样本像元, 将 $\tilde{\boldsymbol{\mu}}(n)$ 记作 $\tilde{\boldsymbol{\mu}}_w(n)$, 则当前待测样本像元为 \mathbf{x}_n 时, 具有实时递归特性的样本像元的均值可表示为

$$\tilde{\boldsymbol{\mu}}_w(n) = \frac{1}{N} \sum_{i \in n_w^{\text{th}}} \mathbf{x}_i = \frac{1}{N} \left(\sum_{i \in (n-1)_w^{\text{th}}} \mathbf{x}_i - \sum_{j=1}^L \mathbf{x}_j^q + \sum_{m=1}^L \mathbf{x}_m^a \right) = \tilde{\boldsymbol{\mu}}_w(n-1) - \frac{L}{N} \boldsymbol{\mu}(\mathbf{X}_q) + \frac{L}{N} \boldsymbol{\mu}(\mathbf{X}_a), \quad (9)$$

式中 $\tilde{\boldsymbol{\mu}}_w(n-1)$ 为前一时刻滑动窗中数据样本像元的均值, L 表示因果滑动窗中移入或移出的行中所含样本像元总数, \mathbf{X}_q 表示沿待测样本像元滑动时窗口中去除的一行数据样本像元 $\{\mathbf{x}_j^q\}_{j=1}^L$, \mathbf{X}_a 表示沿待测样本像元滑动时窗口中移入的一行数据样本像元 $\{\mathbf{x}_m^a\}_{m=1}^L$, $\boldsymbol{\mu}(\mathbf{X}_q)$ 表示滑动窗的移出行中所含样本像元的均值 $\left[\boldsymbol{\mu}(\mathbf{X}_q) = \frac{1}{L} \sum_{j=1}^L \mathbf{x}_j^q \right]$, $\boldsymbol{\mu}(\mathbf{X}_a)$ 表示滑动窗的加入行中所含样本像元的均值, $\boldsymbol{\mu}(\mathbf{X}_a) = \frac{1}{L} \sum_{m=1}^L \mathbf{x}_m^a$ 。

同理, 由(8)式可以推得具有实时递归特性的样本协方差矩阵为

$$\begin{aligned} \tilde{\mathbf{K}}_w(n) &= \frac{1}{N} \sum_{i \in n_w^{\text{th}}} [\mathbf{x}_i - \tilde{\boldsymbol{\mu}}_w(n)] [\mathbf{x}_i - \tilde{\boldsymbol{\mu}}_w(n)]^T = \\ &= \frac{1}{N} \left\{ \sum_{i \in (n-1)_w^{\text{th}}} [\mathbf{x}_i - \tilde{\boldsymbol{\mu}}_w(n-1)] [\mathbf{x}_i - \tilde{\boldsymbol{\mu}}_w(n-1)]^T \right\} - \\ &= \frac{1}{N} \left\{ \sum_{j=1}^L [\mathbf{x}_j^q - \boldsymbol{\mu}(\mathbf{X}_q)] [\mathbf{x}_j^q - \boldsymbol{\mu}(\mathbf{X}_q)]^T - \sum_{m=1}^L [\mathbf{x}_m^a - \boldsymbol{\mu}(\mathbf{X}_a)] [\mathbf{x}_m^a - \boldsymbol{\mu}(\mathbf{X}_a)]^T \right\} = \\ &= \tilde{\mathbf{K}}_w(n-1) - \frac{1}{N} \sum_{j=1}^L \mathbf{x}_j^q (\mathbf{x}_j^q)^T + \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_q) \boldsymbol{\mu}(\mathbf{X}_q)^T + \frac{1}{N} \sum_{m=1}^L \mathbf{x}_m^a (\mathbf{x}_m^a)^T - \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_a) \boldsymbol{\mu}(\mathbf{X}_a)^T = \\ &= \tilde{\mathbf{K}}_w(n-1) - \frac{1}{N} \mathbf{X}_q \mathbf{X}_q^T + \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_q) \boldsymbol{\mu}(\mathbf{X}_q)^T + \frac{1}{N} \mathbf{X}_a \mathbf{X}_a^T - \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_a) \boldsymbol{\mu}(\mathbf{X}_a)^T, \end{aligned} \quad (10)$$

式中 $\tilde{\mathbf{K}}_w(n-1)$ 为前一时刻滑动窗中样本像元的协方差矩阵。为了简化矩阵的求逆运算, 降低计算复杂度, 利用 4 次 Woodbury 矩阵引理对 $\tilde{\mathbf{K}}_w(n)$ 的逆矩阵进行实时递归更新。将(10)式中的前 4 项作为一个整体, 记为

$$\tilde{\mathbf{K}}_1(n-1) = \tilde{\mathbf{K}}_w(n-1) - \frac{1}{N} \mathbf{X}_q \mathbf{X}_q^T + \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_q) \boldsymbol{\mu}(\mathbf{X}_q)^T + \frac{1}{N} \mathbf{X}_a \mathbf{X}_a^T. \quad (11)$$

根据(5)式和(11)式, 令 $\mathbf{A} = \tilde{\mathbf{K}}_1(n-1)$, $\mathbf{U} = \mathbf{V} = \sqrt{(2L-1)/N} \boldsymbol{\mu}(\mathbf{X}_a)$, 推出(10)式的逆矩阵为

$$\begin{aligned} \tilde{\mathbf{K}}_w(n)^{-1} &= \left[\tilde{\mathbf{K}}_1(n-1) - \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_a) \boldsymbol{\mu}(\mathbf{X}_a)^T \right]^{-1} = \\ &= \tilde{\mathbf{K}}_1(n-1)^{-1} + \frac{[(2L-1)/N] \tilde{\mathbf{K}}_1(n-1)^{-1} \boldsymbol{\mu}(\mathbf{X}_a) \boldsymbol{\mu}(\mathbf{X}_a)^T \tilde{\mathbf{K}}_1(n-1)^{-1}}{1 - [(2L-1)/N] \boldsymbol{\mu}(\mathbf{X}_a)^T \tilde{\mathbf{K}}_1(n-1)^{-1} \boldsymbol{\mu}(\mathbf{X}_a)}. \end{aligned} \quad (12)$$

为了计算未知矩阵 $\tilde{\mathbf{K}}_1(n-1)^{-1}$, 将(11)式的前 3 项作为一个整体, 记为

$$\tilde{\mathbf{K}}_2(n-1) = \tilde{\mathbf{K}}_w(n-1) - (1/N) \mathbf{X}_q \mathbf{X}_q^T + [(2L-1)/N] \boldsymbol{\mu}(\mathbf{X}_q) \boldsymbol{\mu}(\mathbf{X}_q)^T. \quad (13)$$

根据(11)式和(13)式, 利用 Woodbury 引理, 令 $\mathbf{A} = \tilde{\mathbf{K}}_2(n-1)$, $\mathbf{U} = \mathbf{V} = \sqrt{1/N} \mathbf{X}_a$, 可以推出 $\tilde{\mathbf{K}}_1(n-1)^{-1}$ 为

$$\begin{aligned} \tilde{\mathbf{K}}_1(n-1)^{-1} &= [\tilde{\mathbf{K}}_2(n-1) + (1/N) \mathbf{X}_a \mathbf{X}_a^T]^{-1} = \\ &= \tilde{\mathbf{K}}_2(n-1)^{-1} - (1/N) \tilde{\mathbf{K}}_2(n-1)^{-1} \mathbf{X}_a [I + (1/N) \mathbf{X}_a^T \tilde{\mathbf{K}}_2(n-1)^{-1} \mathbf{X}_a]^{-1} \mathbf{X}_a^T \tilde{\mathbf{K}}_2(n-1)^{-1}, \end{aligned} \quad (14)$$

此时 $\tilde{\mathbf{K}}_2(n-1)^{-1}$ 为未知量, 将(13)式的前两项作为一个整体, 记为

$$\tilde{\mathbf{K}}_3(n-1) = \tilde{\mathbf{K}}_w(n-1) - (1/N) \mathbf{X}_q \mathbf{X}_q^T. \quad (15)$$

同理, 由(13)式和(15)式, 利用 Woodbury 引理, 令 $\mathbf{A} = \tilde{\mathbf{K}}_3(n-1)$, $\mathbf{U} = \mathbf{V} = \sqrt{(2L-1)/N} \boldsymbol{\mu}(\mathbf{X}_q)$, 求得 $\tilde{\mathbf{K}}_2(n-1)^{-1}$ 为

$$\begin{aligned} \tilde{\mathbf{K}}_2(n-1)^{-1} &= \left[\tilde{\mathbf{K}}_3(n-1) + \frac{2L-1}{N} \boldsymbol{\mu}(\mathbf{X}_q) \boldsymbol{\mu}(\mathbf{X}_q)^T \right]^{-1} = \\ & \tilde{\mathbf{K}}_3(n-1)^{-1} - \frac{[(2L-1)/N] \tilde{\mathbf{K}}_3(n-1)^{-1} \boldsymbol{\mu}(\mathbf{X}_q) \boldsymbol{\mu}(\mathbf{X}_q)^T \tilde{\mathbf{K}}_3(n-1)^{-1}}{1 + [(2L-1)/N] \boldsymbol{\mu}(\mathbf{X}_q)^T \tilde{\mathbf{K}}_3(n-1)^{-1} \boldsymbol{\mu}(\mathbf{X}_q)}. \end{aligned} \quad (16)$$

为求得 $\tilde{\mathbf{K}}_3(n-1)^{-1}$, 根据(5)式和(15)式, 令 $\mathbf{A} = \tilde{\mathbf{K}}_w(n-1)$, $\mathbf{U} = \mathbf{V} = \sqrt{1/N} \mathbf{X}_q$, 则有

$$\begin{aligned} \tilde{\mathbf{K}}_3(n-1)^{-1} &= [\tilde{\mathbf{K}}_w(n-1) - (1/N) \mathbf{X}_q \mathbf{X}_q^T]^{-1} = \\ & \tilde{\mathbf{K}}_w(n-1)^{-1} + (1/N) \tilde{\mathbf{K}}_w(n-1)^{-1} \mathbf{X}_q [I - (1/N) \mathbf{X}_q^T \tilde{\mathbf{K}}_w(n-1)^{-1} \mathbf{X}_q]^{-1} \mathbf{X}_q^T \tilde{\mathbf{K}}_w(n-1)^{-1}. \end{aligned} \quad (17)$$

综上所述, 可以通过(17)式, 由 $\tilde{\mathbf{K}}_w(n-1)^{-1}$ 推出 $\tilde{\mathbf{K}}_3(n-1)^{-1}$; 然后通过(16)式, 由 $\tilde{\mathbf{K}}_3(n-1)^{-1}$ 推出 $\tilde{\mathbf{K}}_2(n-1)^{-1}$; 接着通过(14)式, 由 $\tilde{\mathbf{K}}_2(n-1)^{-1}$ 推出 $\tilde{\mathbf{K}}_1(n-1)^{-1}$; 最后通过(12)式, 由 $\tilde{\mathbf{K}}_1(n-1)^{-1}$ 推出 $\tilde{\mathbf{K}}_w(n)^{-1}$, 由此实现了由 $\tilde{\mathbf{K}}_w(n-1)^{-1}$ 到 $\tilde{\mathbf{K}}_w(n)^{-1}$ 的实时递归更新。RT-RXD算法的具体实现过程如下:

- 1) 初始化, 确定因果滑动窗尺寸 a 和 b , 计算初始窗中 $\tilde{\boldsymbol{\mu}}_w(1)$ 、 $\tilde{\mathbf{K}}_w(1)$ 及其逆矩阵 $\tilde{\mathbf{K}}_w(1)^{-1}$;
- 2) 当算法运行至第 n 个检测像元时, 当前滑动窗中移入和移出行分别为 \mathbf{X}_a 和 \mathbf{X}_q , 对应的均值为 $\boldsymbol{\mu}(\mathbf{X}_a)$ 和 $\boldsymbol{\mu}(\mathbf{X}_q)$, 已获取的前一时刻的状态信息为 $\tilde{\boldsymbol{\mu}}_w(n-1)$ 和 $\tilde{\mathbf{K}}_w(n-1)^{-1}$;
- 3) 利用步骤 2) 中的信息, 通过 Woodbury 引理, 求出 $\tilde{\mathbf{K}}_w(n)^{-1}$ 的递归更新;
- 4) 将更新后的 $\tilde{\mathbf{K}}_w(n)^{-1}$ 代入 RT-RXD 算子中, 得到当前检测结果;
- 5) 若当前处理的行数少于高光谱数据的行数, 返回步骤 2); 否则, 检测过程结束, 输出 RT-RXD 检测结果。

利用 RT-RXD 算子计算样本 $\tilde{\mathbf{K}}_w(n)^{-1}$ 时, 需要计算初始样本的 $\tilde{\mathbf{K}}_w(1)^{-1}$, 此时滑动窗口尺寸的选择尤为重要。为避免奇异性过大而出现病态矩阵, 因果滑动窗中包含的样本像元的个数应大于样本像元的维数。值得注意的是, 因果滑动窗的环绕型运动轨迹使得 L 的取值并不固定, 当因果滑动窗纵向移动时, L 等于滑动窗口的宽度 b , 当因果滑动窗横向移动时, L 等于滑动窗口的长度 a 。

3.4 计算复杂度分析

采用高斯-若尔当反演(GJ)法对矩阵进行求逆运算^[12], 一个大小为 $m \times m$ 的矩阵, 利用 GJ 法对其求逆时需要的加法计算量 S 和乘法计算量 P 分别为

$$\begin{cases} S = \frac{m^3}{2} + m^2 - \frac{m}{2} + 2 \\ P = \frac{m^3}{2} - \frac{3m}{2} + 1 \end{cases}. \quad (18)$$

由(18)式可知, 矩阵求逆算法总的计算量与 m^3 成正比, 所以利用一个大小为 $n \times n$ 的矩阵的逆矩阵 ($n \leq m$) 递归更新该矩阵的逆矩阵时, 算法的计算量将大大降低。为对算法的复杂度进行量化分析, 表 1 分别统计了 RT-RXD 和 RXD 算子的加法计算量和乘法计算量。

表 1 算法复杂度分析

Table 1 Algorithm complexity analysis

	S	P
RXD	$J^3/2 + (N+1)J^2 + (N-5/2)J + 1$	$J^3/2 + (N+2)J^2 + 1/2J + 1$
RT-RXD	$(4L+7)J^2 + 2(4L^2 - 2L - 3)J + L^3 + 2L^2 - 5L + 3$	$(4L+11)J^2 + (4L^2 + 5)J + L^3 + 2L^2 - 3L + 2$

在算法运行过程中, 执行加法操作所耗费的时间非常短, 甚至可以忽略, 因此仅讨论光谱维数 J 和窗口宽度 b 对乘法计算量的影响, 如图 6、7 所示。显然, 随着 J 的增大, 与 RXD 算法相比, RT-RXD 算法的计算量上升趋势更为缓慢, 算法的有效性更加显著。当光谱维数一定时, 随着 b 的不断增大, RT-RXD 和 RXD 算法的计算量分别呈 e 指数和线性增长, 因此过大的窗口尺寸将制约实时算子的有效性。

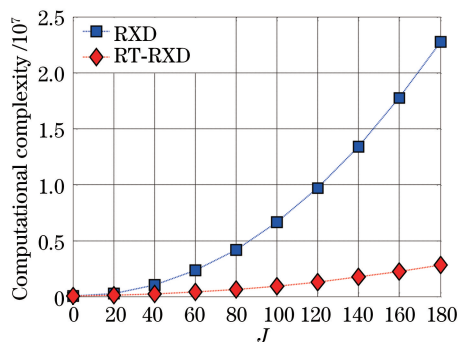


图 6 算法计算量与光谱维数的关系 ($a=37, b=17$)

Fig. 6 Relationship between computational complexity and spectral dimension

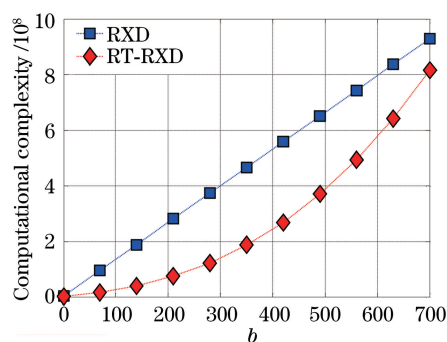


图 7 算法计算量与窗口宽度的关系 ($a=37, J=189$)

Fig. 7 Relationship between computational complexity and window width

4 实验仿真与分析

4.1 AVIRIS 模拟数据实验

模拟数据是以某种特定目的、根据真实高光谱数据的光谱特性、人为合成得到的高光谱图像,对实验分析具有较高的实用价值。为了证明算法的可行性,实验选用一组由 USGS 网站公布的美国内华达州一个赤铜矿矿区的真实图像数据。图像大小为 $350 \text{ pixel} \times 350 \text{ pixel}$,包含 224 个波段,在去除 1~3,105~115 和 150~170 波段范围内的水吸收波段和低信噪比波段后,剩余的 189 个波段用于实验分析。真实图像数据及其地物分布如图 8 所示,5 种地物分别是明矾石、水铵长石、方解石、高岭石和白云母,分别以 A, B, C, K, M 表示,如图 8(b)所示。

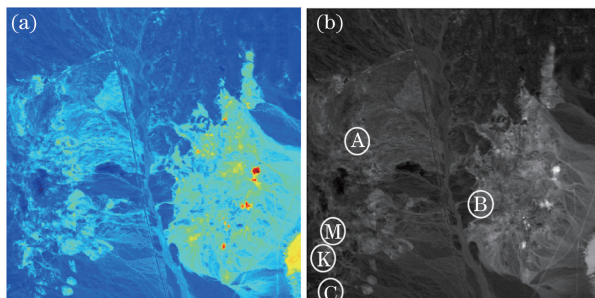


图 8 赤铜矿矿区的 (a) AVIRIS 图像数据及 (b) 真实地物分布

Fig. 8 (a) AVIRIS image and (b) distribution of true ground of cuprite

根据 5 种地物的光谱特性,由文献[13-14]中的数据合成过程生成 25 个异常点,如图 9 所示。合成数据的每一行由 5 个具有相同光谱特性的地物组成,每一列由具有相同大小形状的目标组成。合成数据的第 1、2 列分别是大小为 $4 \text{ pixel} \times 4 \text{ pixel}$ 和 $2 \text{ pixel} \times 2 \text{ pixel}$ 的纯像元,第 3 列是大小为 $2 \text{ pixel} \times 2 \text{ pixel}$ 的混合像元,第 4、5 列都是大小为 $1 \text{ pixel} \times 1 \text{ pixel}$ 的亚像元。将 25 个异常目标嵌入到一个大小为 $200 \text{ pixel} \times$

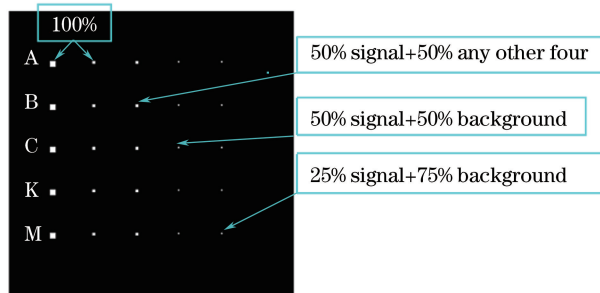


图 9 异常目标分布

Fig. 9 Distribution of anomaly objects

200 pixel 的合成数据图像中,相应位置的背景像元用目标像元替代,将信噪比为 20:1 的加性高斯噪声加入数据背景中。

接收机工作特性(ROC)曲线是高光谱目标检测中常用的一种评价手段。通过改变假设检验模型中选取的阈值 η ,得到由虚警概率 p_f 和检测概率 p_d 构成的关系曲线。在 ROC 曲线无法做出评价时,ROC 曲线下所覆盖的面积(AUC)常被用作二值分类器的另一性能指标。AUC 的数值越大,表明分类器的工作性能越好。为了验证窗口尺寸对算法检测性能的影响,表 2 给出了不同窗口尺寸下 RT-RXD 算法的 AUC。显然,随着窗口尺寸的增大,RT-RXD 算法的检测性能不断提高,当窗口尺寸达到长度 $a=37$ 、宽度 $b=17$ 时,检测性能达到最优并趋于平稳。

表 2 RT-RXD 算法在不同因果窗口尺寸下的 AUC

Table 2 AUC of RT-RXD algorithm with different causal window sizes

Window size	$a=23, b=9$	$a=23, b=11$	$a=23, b=13$	$a=29, b=15$	$a=35, b=15$	$a=37, b=17$	$a=37, b=19$
AUC	0.9790	0.9802	0.9820	0.9874	0.9884	0.9930	0.9930

实验选取最优局部窗对 RXD 和 RT-RXD 算法进行仿真分析。实验检测灰度图如图 10 所示,与 RXD 算法相比,RT-RXD 算法对第 1 列数据的检测效果更好,但对第 4 列数据的检测结果并不明显。然而,如图 11、12 所示,分别从两种算法的阈值分割图及三维峰度图可以看出,两种算法的检测效果基本上是一致的,均能够准确地检测出异常像元的位置。这是因为 RT-RXD 算法在检测中第 1 列异常目标的检测值较大,对弱异常的检测带来一定影响,造成视觉干扰。

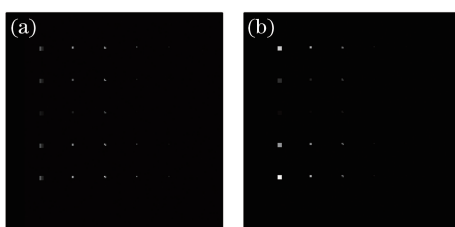


图 10 检测灰度图。(a) RXD; (b) RT-RXD

Fig. 10 Detection results of grayscale image. (a) RXD; (b) RT-RXD

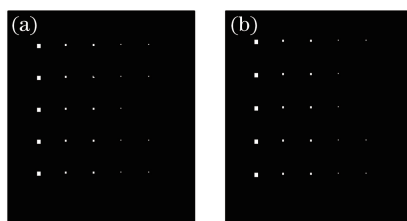


图 11 阈值分割图。(a) RXD; (b) RT-RXD

Fig. 11 Detection results after threshold segmentation. (a) RXD; (b) RT-RXD

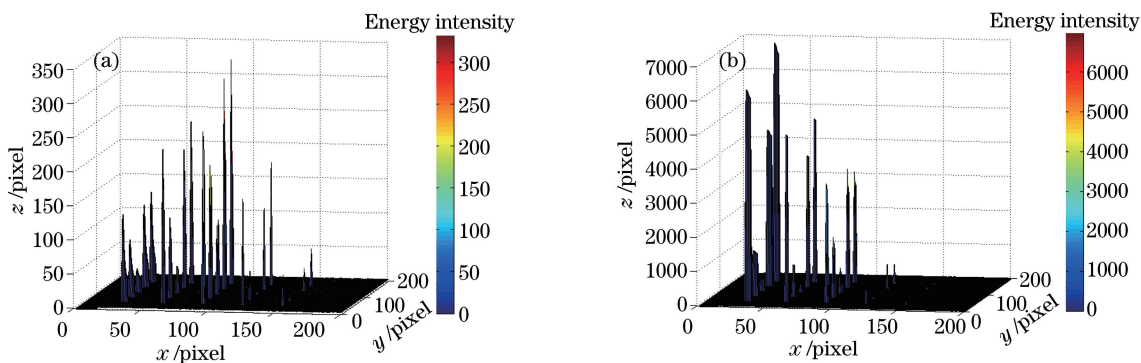


图 12 三维峰度图。(a) RXD; (b) RT-RXD

Fig. 12 Detection results of three-dimensional kurtosis image. (a) RXD; (b) RT-RXD

为了对算法的检测性能进行定量评价,图 13 给出了 RXD 与 RT-RXD 算法的 ROC 曲线,可以看出,两种算法的检测性能相近。在保证相对一致的算法检测精度的前提下,RXD 算法只能给出检测的最终结果,而 RT-RXD 算法却可以给出实时检测的处理过程,如图 14 所示。随着检测过程的推进,RT-RXD 算法逐渐检测出图像中异常目标的准确位置。异常检测算法的实时化为数据的实时传输提供了理论支持。

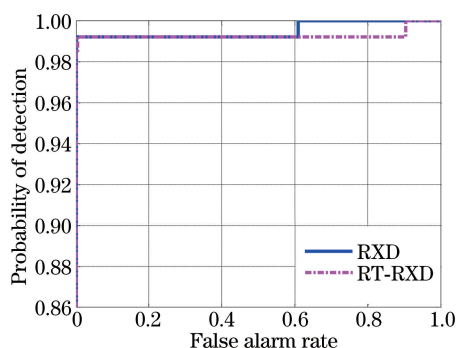


图 13 ROC 曲线性能分析

Fig. 13 Performance analysis of ROC curves

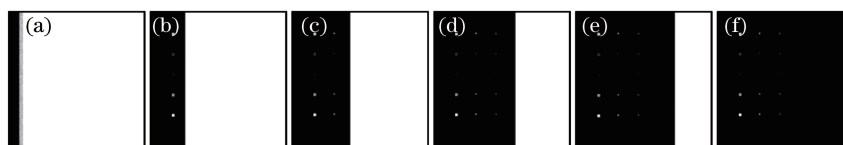


图 14 实时处理进程。(a) 未检测到目标；(b) 检测到的第 2 行目标；(c) 检测到的第 3 行目标；
(d) 检测到的第 4 行目标；(e) 检测到的第 5 行目标；(f) 检测到的第 6 行目标

Fig. 14 Real-time processing. (a) Target undetected; (b) detected target of line 2; (c) detected target of line 3;
(d) detected target of line 4; (e) detected target of line 5; (f) detected target of line 6

4.2 真实数据实验

为了验证所提实时算法的实用性,实验选用一幅取自意大利北部帕维亚城市中心的真实高光谱图像数据,图像大小为 $115 \text{ pixel} \times 115 \text{ pixel}$,波长范围为 $0.43 \sim 0.86 \mu\text{m}$,去除水吸收波段及低信噪比波段后,剩余 102 个波段用于实验分析。其中桥、水和阴影构成了图像数据的背景,桥上的车辆则为异常目标。图 15 即为高光谱伪彩色图像及真实地面异常目标分布。

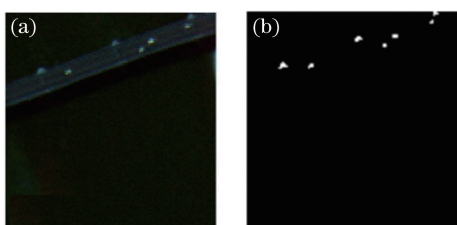


图 15 (a) 高光谱伪彩色图像；(b) 真实地物分布图

Fig. 15 (a) Hyperspectral pseudo color image; (b) true ground distribution

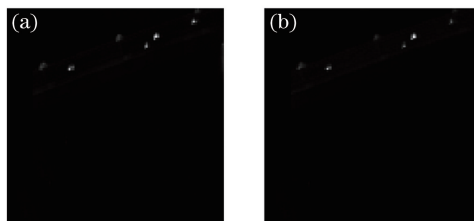


图 16 灰度图。(a) RXD; (b) RT-RXD

Fig. 16 Grayscale images. (a) RXD; (b) RT-RXD

为了验证真实数据中窗口参数设定对算法检测性能的影响,表 3 给出了不同窗口尺寸下 RT-RXD 算法的 AUC。同样,随着窗口尺寸的增大,算法的检测性能不断提高,当窗口尺寸设定为长度 $a=37$ 、宽度 $b=15$ 时,AUC 达到最大并保持不变。

表 3 RT-RXD 算法在不同窗口尺寸下的 AUC
Table 3 AUC of RT-RXD algorithm with different causal window sizes

Window size	$a=21, b=7$	$a=27, b=11$	$a=29, b=11$	$a=31, b=15$	$a=35, b=15$	$a=37, b=15$	$a=37, b=17$
AUC	0.9822	0.9844	0.9888	0.9932	0.9987	0.9988	0.9988

选取长度 $a=37$ 、宽度 $b=15$ 的局部窗进行真实数据实验分析。由图 16 可知,RXD 和 RT-RXD 算法的检测效果基本相同。两种算法的 ROC 曲线如图 17 所示,在同等虚警概率下,两种算法的检测概率基本相同;在虚警概率不高于 0.5% 的情况下,所提实时算法具有较高的检测概率。这就意味着对于虚警概率要求较低的情况,RT-RXD 算法具有更好的适应性,实用价值更高。

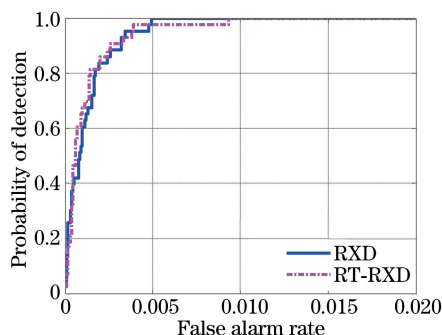


图 17 ROC 曲线性能分析

Fig. 17 Performance analysis of ROC curves

图 18 为实时算法的检测过程,可以看出,随着检测过程的进行,RT-RXD 算法逐渐检测到异常目标。但是当算法执行到图 18(a)时,检测到的前两个异常目标均较明亮,随着检测过程的结束,前两个异常目标被后续检测到的强异常目标所干扰,亮度变暗。

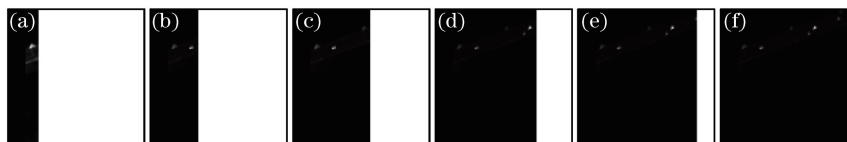


图 18 实时处理进程。(a) 检测到的第 1 行目标;(b) 检测到的第 2 行目标;(c) 检测到的第 3 行目标;
(d) 检测到的第 4 行目标;(e) 检测到的第 5, 6 行目标;(e) 检测到的第 7 行目标

Fig. 18 Real-time processing. (a) Detected target of line 1; (b) detected target of line 2; (c) detected target of line 3;
(d) detected target of line 4; (e) detected target of line 5,6; (f) detected target of line 7

4.3 算法时效性分析

为了验证 RT-RXD 算法的时效性,表 4 对有递归更新的 RT-RXD 算法和无递归更新的 RXD 算法在模拟和真实高光谱数据中总的运行时间进行了分析讨论。实验运行环境:主频为 3.50 GHz 的 Intel Core i7-4770 处理器,安装内存为 16 GB,仿真软件为 Matlab R2014。为了避免系统脉冲误差对实验结果的干扰,在相同条件下选用 5 次检测结果的平均值作为实验检测的结果值。与 RXD 算法相比,RT-RXD 算法大大减少了算法的运算时间,提高了算法的运算效率。

表 4 时间复杂度分析
Table 4 Time complexity analysis

	Computation time /s		Speedup
	RXD	RT-RXD	
Synthetic data	352.599	61.216	5.760
Real data	32.937	7.702	4.276

为了进一步验证 RT-RXD 算法在时效性上的优势,图 19 为 RXD 与 RT-RXD 算法处理每行数据样本像元所需要的时间。由于 RT-RXD 算法实现了矩阵的递归更新,避免了大矩阵的求逆运算,因此相比于 RXD 算法,具有更快的运算速度。

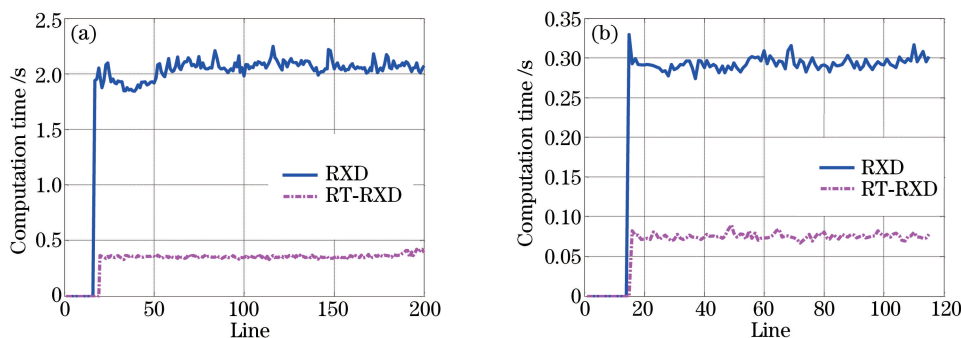


图 19 算法时间复杂度分析。(a) 模拟数据; (b) 真实数据

Fig. 19 Time complexity analysis of the proposed algorithm. (a) Synthetic data; (b) real data

5 结 论

提出了一种基于逐行处理方式的高光谱实时异常目标检测算法。引入局部因果窗的概念,利用卡尔曼滤波器的递归思想及 Woodbury 矩阵引理,实现了算法的实时递归更新。通过实时处理,可以发现检测过程中的微弱目标。对算法复杂度进行定量分析,理论上验证了所提实时算法通过对复杂矩阵的递归更新,能够显著降低算法的计算复杂度。实验结果表明,与非实时算法相比,在不降低算法检测精度的条件下,所提实时算法大大减少了运算时间,提高了算子的执行效率,在高光谱异常目标检测领域将有更广阔的应用前景。

参 考 文 献

- [1] Wang Xiaofei, Yan Qiujing, Zhang Junping, *et al.* Super-resolution reconstruction algorithm based on relevance vector machine for hyperspectral image[J]. Chinese J Lasers, 2014, 41(s1): s114001.
王晓飞, 阎秋静, 张钧萍, 等. 基于相关向量机的高光谱图像超分辨率算法[J]. 中国激光, 2014, 41(s1): s114001.
- [2] Wang Xiaofei, Yan Qiujing. An ensemble learning algorithm for one-class classification of hyperspectral images[J]. Acta Optica Sinica, 2014, 34(s2): s211002.
王晓飞, 阎秋静. 基于集成学习的高光谱图像一类分类算法[J]. 光学学报, 2014, 34(s2): s211002.
- [3] Zhao Chunhui, You Wei, Qi Bin, *et al.* Real-time anomaly detection algorithm for hyperspectral remote sensing by using recursive polynomial kernel function[J]. Acta Optica Sinica, 2016, 36(2): 0228002.
赵春晖, 尤伟, 齐滨, 等. 采用多项式递归核的高光谱遥感异常实时检测算法[J]. 光学学报, 2016, 36(2): 0228002.
- [4] Wang Liguang, Zhao Chunhui. Hyperspectral image processing technology [M]. Beijing: National Defence Industry Press, 2013.
王立国, 赵春晖. 高光谱图像处理技术[M]. 北京: 国防工业出版社, 2013.
- [5] Zhang Bing, Gao Lianru. Hyperspectral image classification and target detection[M]. Beijing: Science Press, 2011.
张兵, 高连如. 高光谱图像分类与目标探测[M]. 北京: 科学出版社, 2011.
- [6] Reed I S, Yu X. Adaptive multi-band CFAR detection of an optical pattern with unknown spectral distribution[J]. IEEE Transactions on Acoustics Speech and Signal Processing, 1990, 38(10): 1760-1770.
- [7] Chen S Y, Wang Y L, Wu C C, *et al.* Real-time causal processing of anomaly detection for hyperspectral imagery[J]. IEEE Transactions on Aerospace and Electronic Systems, 2014, 50(2): 1511-1534.
- [8] Chang C I, Li Y, Hobbs M C, *et al.* Progressive band processing of anomaly detection in hyperspectral imagery[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2015, 8(7): 3558-3571.
- [9] Rossi A, Acito N, Diani M, *et al.* RX architectures for real-time anomaly detection in hyperspectral images[J]. Journal of Real-Time Image Processing, 2014, 9(3): 503-517.
- [10] Zhang Xianda. Matrix analysis and applications[M]. Beijing: Tsinghua University Press, 2004.
张贤达. 矩阵分析与应用[M]. 北京: 清华大学出版社, 2004.

-
- [11] Chang C I, Wang Y L, Chen S Y. Anomaly detection using causal sliding windows[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2015, 8(7): 3260-3270.
- [12] Kossi A, Acito N, Diani M, *et al.* RX architecture for real-time anomaly detection in hyperspectral images[J]. Journal of Real-Time Image Processing, 2014, 9(3): 503-517.
- [13] Wang J, Chang C I. Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2006, 44(9): 2601-2616.
- [14] Chang Y C, Ren H, Chang C I, *et al.* How to design synthetic images to validate and evaluate hyperspectral imaging algorithms[C]. SPIE, 2008, 6966: 69661P.