文章编号:1001-9014(2016)06-0708-07

DOI:10.11972/j.issn.1001-9014.2016.06.013

基于局部核 RX 算法的高光谱实时检测

赵春晖*, 姚淅峰

(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:提出了一种基于 LKRX 检测器的实时异常检测算法. 利用局部因果滑动阵列窗,使检测系统保持因果性. 根据卡尔曼滤波器的递归思想,利用 Hermitian 矩阵分块求逆引理和 Woodbury 引理,将 LKRX 算法中核协方差矩阵以及其逆矩阵以递归方式更新,避免了数据的重复计算和逆矩阵的求解,大大降低了算法复杂度. 通过真实数据进行实验,结果表明,与 LKRX 算法相比,实时 LKRX 算法在保持相同检测精度的同时,消耗更少的计算时间;而与实时 RX 算法相比,实时 LKRX 算法能够检测到更多的异常目标.

关 键 词:高光谱图像处理;多项式 KRX 算法;实时异常检测;Hermitian 矩阵分块求逆引理;Woodbury 引理

中图分类号:TP751.1 **文献标识码**: A

Local kernel RX algorithm-based hyperspectral real-time detection

ZHAO Chun-Hui*, YAO Xi-Feng

(Information and Communication Engineering College, Harbin Engineering University, Harbin 150001, China)

Abstract: LKRX detector-based hyperspectral real-time anomaly detection algorithm was proposed. Using local causal sliding array window, the causality of detection system is remained. According to Kalman filter, by using Hermitian lemma and Woodbury's identity, the kernel covariance matrix and its inverse in KRX algorithm are updated recursively. This thereby leads to low computational complexity. Experimental results demonstrated that real-time KRX detector consumes less time in comparison with KRX detector by keeping the same detection performance, which detects more anomalies.

Key words: hyperspectral image processing, polynomial KRX algorithm, real-time anomaly detection, Hermitian lemma, Woodbury's identity

PACS: 42.30.-d

引言

高光谱目标检测是高光谱图像处理中的重要方向之一,利用其所含丰富的光谱信息,可以有效地将弱小的地物目标探测出来. 而异常检测由于不需要先验知识和信息,更加符合实际需求,其研究和发展受到了广泛的关注. Reed 和 Yu 提出的全局 RX 检测器[1]被认为是基准异常检测算法,它是基于广义似然比检验,通过计算当前像元与背景数据均值之间的马氏距离来检测异常目标. 而根据滑动窗设计的局部 RX 检测器[2-3],通过利用局部背景信息,可以有效地检测出湮没在全局中的局部异常目标. 但是 RX 算法仅仅利用了数据的低阶线性统计信息,没有挖掘更高阶的统计特性. 通过利用核技术[4],Kwon 提出了全局核 RX 检测器(Global Kernel RX Detector, GKRXD)[5],它将原始高光谱数据投影到更高维的特征空间,充分利用了波段间丰富的非线性信息,获得了更好的检测精度. 但是 GKRXD 中核协方差矩阵的维数等于全局背景中所含像元的个数,导致了高维矩阵的乘法运算和求逆运算,需要消耗大量的处理时间. 因此在此基础上,Kwon 利用滑动双窗设计了局部核 RX 检测器(Local Kernel RX

收稿日期:2016-04-25,**修回日期**:2016-09-29 **Received date**: 2016-04-25, **revised date**: 2016-09-29

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),中国博士后科学基金(2014M551221),中央高校基础研究基金(HEUCFl608)

Foundation items: Supported by National Natural Science Foundation of China (61405041, 61571145), the Key Program of Heilongjiang Natural Science Foundation (ZD201216), the Program Excellent Academic Leaders of Harbin (RC2013XK009003), the China Postdoctoral Science Foundation (2014M551221), and the Fundamental Research Funds for the Central Universities (HEUCFl408)

作者简介(Biography):赵春晖(1965-),男,黑龙江哈尔滨人,教授,工学博士,主要研究领域为高光谱图像处理,非线性信号处理.

* **通讯作者(Corresponding author)**: E-mail: zhaochunhui@hrbeu.edu.cn

Detector, LKRXD)[5],将双窗之间的像元作为局部背景数据,限制了核协方差矩阵的维度,显著地减少了运算时间.

近年来,面对检测运动目标的需求和大数据存储的压力,实时处理得到了较快的发展. 文献[6]提出了需要满足实时异常检测的因果性. 后来,文献[7-9]提出了全局实时 RX 检测器(Global Real-time RX Detector, GRT-RXD)和基于不同滑动窗设计的局部实时 RX 检测器(Local Real-time RX Detector, LRT-RXD),通过协方差逆矩阵的递归更新,避免了大量数据的重复计算,显示了较好的实时处理进程. 然而,这些实时算法都是以 RX 检测器为框架设计的,依然无法解决检测精度不高的问题. 为此,本文提出了基于 LKRXD 的实时异常检测算法.

1　局部核 RX 算法

令 $\boldsymbol{r}$ 表示待检测像元,通过将原始数据投影到更高维的特征空间,则 LKRXD 可以表示为

$$\delta^{\mathrm{LKRXD}}(\Phi(\boldsymbol{r})) = (\Phi(\boldsymbol{r}) - \boldsymbol{\mu}_{B\Phi})^{\mathrm{T}} \boldsymbol{K}_{B\Phi}^{-1} (\Phi(\boldsymbol{r}) - \boldsymbol{\mu}_{B\Phi}) \gtrless \eta \quad , \quad (1)$$

其中,$\Phi(\boldsymbol{r})$表示为特征空间中的待检测像元,$\boldsymbol{\mu}_{B\Phi}$和 $\boldsymbol{K}_{B\Phi}$分别表示特征空间中局部背景数据的估计均值和协方差矩阵. 通过特定的核化和推导,最终 LKRX 算法可简化为

$$\delta^{\mathrm{LKRXD}}(\Phi(\boldsymbol{r})) = (\boldsymbol{k}_r^{\mathrm{T}} - \boldsymbol{k}_{\mu}^{\mathrm{T}})^{\mathrm{T}} \boldsymbol{K}_B^{-1} (\boldsymbol{k}_r^{\mathrm{T}} - \boldsymbol{k}_{\mu}^{\mathrm{T}}) \quad , \quad (2)$$

其中,

$$\boldsymbol{k}_r^{\mathrm{T}} = k(\boldsymbol{r}, \boldsymbol{X}_B) - \left(\frac{1}{N}\sum_{i=1}^{N} k(\boldsymbol{r}, \boldsymbol{x}_i)\right) 1_{1\times \mathrm{N}} \quad , \quad (3)$$

$$\boldsymbol{k}_{\mu}^{\mathrm{T}} = \left(\frac{1}{N}\sum_{i=1}^{N} k(\boldsymbol{x}_i, \boldsymbol{X}_B)\right) - \left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) 1_{1\times \mathrm{N}} \quad , \quad (4)$$

$\boldsymbol{X}_B$ 是原始局部背景数据矩阵,包含 N 个像元样本,每个像元样本具有 L 个波段. $\boldsymbol{K}_B = k(\boldsymbol{X}_B, \boldsymbol{X}_B)$定义为核 Gram 矩阵.

通过执行点积运算,核方法避免了数据在特征空间中的处理[4]. 本文采用多项式函数作为核映射,其表达式如下:

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}_i)^d, d \in N \quad . \quad (5)$$

2　局部实时 KRX 算法

2.1　算法原理

实时处理要求满足时效性和因果性[8]. 从式(2)可以看出,$\boldsymbol{K}_B$, $\boldsymbol{K}_B^{-1}$ 以及 $\boldsymbol{k}_{\mu}^{T}$ 具有较大的计算复杂度,需要消耗大量的处理时间,很难满足时效性的要求. 而针对因果性,不难看出 LKRXD 的执行需要用到当前检测像元以后的像元信息,也不符合实时处理的条件. 因此,针对时效性和因果性,本文提出了局部实时 KRX 检测器(LRT-KRXD).

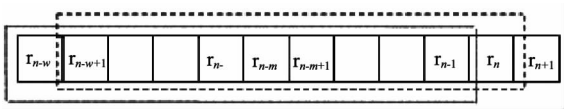

图 1　时刻和时刻局部因果滑动阵列窗

Fig. 1　Local causal array windows at $\boldsymbol{r}_n$ and $\boldsymbol{r}_{n+1}$

为了使处理数据满足因果性,本文引用了局部因果滑动阵列窗[9]. 图 1 展示了 n 时刻(待检测像元为 $\boldsymbol{r}_n$)由点线表示的局部因果滑动阵列窗和 $n+1$ 时刻(待检测像元为 $\boldsymbol{r}_{n+1}$)由折线表示的局部因果滑动阵列窗. 窗的大小为 w,当它从 n 时刻滑动到 $n+1$ 时刻,$\boldsymbol{r}_{n-w}$ 像元从局部因果滑动阵列窗中出来,而 $\boldsymbol{r}_n$ 像元进入,表现为一进一出. 并且局部因果滑动阵列窗只包含当前像元之前的 w 个像元信息,因此利用局部因果滑动阵列窗处理数据,保证了系统的因果性. 随着局部因果滑动阵列窗从 n 时刻移动到 $n+1$ 时刻,待检测像元也从 $\boldsymbol{r}_n$ 变为 $\boldsymbol{r}_{n+1}$,由此实现了高光谱图像的逐像元检测.

对于 LKRXD,为了满足实时处理的因果性,式(2)可以写成如下形式

$$\delta^{\mathrm{LRT\text{-}KRXD}}(\Phi(\hat{\boldsymbol{r}}_n)) = (\boldsymbol{k}_{\hat{r}}^{\mathrm{T}}(n) - \boldsymbol{k}_{\hat{\mu}}^{\mathrm{T}}(n))^{\mathrm{T}} \boldsymbol{K}_w^{-1}(n) (\boldsymbol{k}_{\hat{r}}^{\mathrm{T}}(n) - \boldsymbol{k}_{\hat{\mu}}^{\mathrm{T}}(n)) \quad , \quad (6)$$

其中,$\hat{\boldsymbol{r}}_n$ 是第 n 个待检测像元,$\boldsymbol{K}_w(n) = k(\boldsymbol{X}_w(n), \boldsymbol{X}_w(n))$定义为局部因果核 Gram 矩阵,$\boldsymbol{X}_w(n) = [\hat{\boldsymbol{r}}_{\mathrm{n-w}}, \hat{\boldsymbol{r}}_{\mathrm{n-w+1}}, \cdots, \hat{\boldsymbol{r}}_{\mathrm{n-1}}]$、$\boldsymbol{k}_{\hat{r}}^{\mathrm{T}}(n)$和 $\boldsymbol{k}_{\hat{\mu}}^{\mathrm{T}}(n)$都只利用了局部因果滑动阵列窗内的像元信息.

为了实现实时处理的时效性,需要推导 $\boldsymbol{K}_w(n)$ 和 $\boldsymbol{K}_w^{-1}(n)$的更新式. $\boldsymbol{K}_w(n)$的核矩阵运算表示为

$$\boldsymbol{K}_w(n) = k(\boldsymbol{X}_w(n), \boldsymbol{X}_w(n)) = \begin{bmatrix} \alpha_{n-w,n-w} & \alpha_{n-w,n-w+1} & \cdots & \alpha_{n-w,n-1} \\ \alpha_{n-w+1,n-w} & \alpha_{n-w+1,n-w+1} & \cdots & \alpha_{n-w+1,n-1} \\ \vdots & \vdots & & \vdots \\ \alpha_{n-1,n-w} & \alpha_{n-1,n-w+1} & \cdots & \alpha_{n-1,n-1} \end{bmatrix} = \begin{bmatrix} \rho & \alpha \\ \alpha^{\mathrm{T}} & \boldsymbol{K}_{bw} \end{bmatrix} \quad . \quad (7)$$

利用 Hermitian 求逆引理[10-11],$\boldsymbol{K}_w^{-1}(n)$表示为

$$\boldsymbol{K}_w^{-1}(n) = \begin{bmatrix} \rho & \alpha \\ \alpha^{\mathrm{T}} & \boldsymbol{K}_{bw} \end{bmatrix}$$

$$= \begin{bmatrix} \rho^{-1} + \alpha\rho^{-1}\boldsymbol{\kappa}\rho^{-1}\alpha^{\mathrm{T}} & -\alpha\rho^{-1}\boldsymbol{\kappa} \\ -\boldsymbol{\kappa}\rho^{-1}\alpha^{\mathrm{T}} & \boldsymbol{\kappa} \end{bmatrix} = \begin{bmatrix} a & c \\ c^{\mathrm{T}} & \boldsymbol{\kappa} \end{bmatrix} \quad ,(8)$$

$$\boldsymbol{\kappa} = (\boldsymbol{K}_{bw} - \alpha^{\mathrm{T}}\rho^{-1}\alpha)^{-1}$$

其中,$\boldsymbol{\kappa}$ 可以由 $\boldsymbol{K}_w^{-1}(n)$ 获得. 根据式(8),可以推得

$$\begin{aligned} (\boldsymbol{K}_{bw} - \alpha^{\mathrm{T}}\rho^{-1}\alpha)^{-1} &= K \\ \boldsymbol{K}_{bw} &= (\boldsymbol{\kappa}^{-1} + \alpha^{\mathrm{T}}\rho^{-1}\alpha) \quad . \quad (9) \\ \boldsymbol{K}_{bw}^{-1} &= (\boldsymbol{\kappa}^{-1} + \rho^{-1}\alpha^{\mathrm{T}}\alpha)^{-1} \end{aligned}$$

为了计算 $\boldsymbol{K}_{bw}^{-1}$,需要引入 Woodbury 引理[12],它的表达式如下:

$$(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^{\mathrm{T}})^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{u}} \quad . \quad (10)$$

由于 ρ^{-1} 是一个常数,利用 Woodbury 引理,式(9)可以转变为:

$$\boldsymbol{K}_{bw}^{-1} = (\boldsymbol{\kappa}^{-1} + \rho^{-1}\alpha^{\mathrm{T}}\alpha)^{-1} = \boldsymbol{\kappa} - \frac{\rho^{-1}\boldsymbol{\kappa}\alpha^{\mathrm{T}}\alpha\boldsymbol{\kappa}}{1 + \rho^{-1}\alpha\boldsymbol{\kappa}\alpha^{\mathrm{T}}} . \quad (11)$$

因为 $\boldsymbol{\kappa}$ 可以由 $\boldsymbol{K}_w^{-1}$ 获得,通过式(11),可以推出 $\boldsymbol{K}_{bw}^{-1}$ 可以由 $\boldsymbol{K}_w^{-1}$(n)得到.

同理,$\boldsymbol{K}_w(n+1)$ 的核矩阵运算可以表示为

$$\boldsymbol{K}_w(n+1) = k(\boldsymbol{X}_{n+1,w}, \boldsymbol{X}_{n+1,w})$$

$$= \begin{bmatrix} \beta_{n-w+1,n-w+1} & \beta_{n-w+1,n-w+2} & \cdots & \beta_{n-w+1,n} \\ \beta_{n-w+2,n-w+1} & \beta_{n-w+2,n-w+2} & \cdots & \beta_{n-w+2,n} \\ \vdots & \vdots & & \vdots \\ \beta_{n,n-w+1} & \beta_{n,n-w+2} & \cdots & \beta_{n,n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{fw} & \beta^{\mathrm{T}} \\ \beta & \chi \end{bmatrix}$$

. (12)

由式(7)和(12)可以得出 $\boldsymbol{K}_{bw} = \boldsymbol{K}_{fw}$,因此 $\boldsymbol{K}_{bw}^{-1} = \boldsymbol{K}_{fw}^{-1}$. 再次利用 Hermitian 矩阵分块求逆引理,$\boldsymbol{K}_w^{-1}(n+1)$ 可以表示为:

$$\boldsymbol{K}_w^{-1}(n+1) = \begin{bmatrix} \boldsymbol{K}_{fw} & \beta^{\mathrm{T}} \\ \beta & \chi \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{K}_{fw}^{-1} & 0^{\mathrm{T}} \\ 0 & 0 \end{bmatrix} + \frac{1}{\chi + \beta\varepsilon}\begin{bmatrix} \varepsilon^{\mathrm{T}}\varepsilon & \varepsilon^{\mathrm{T}} \\ \varepsilon & 1 \end{bmatrix} \quad .(13)$$

$$\varepsilon = -\boldsymbol{K}_{fw}^{-1}\beta^{\mathrm{T}}$$

$\boldsymbol{K}_{bw}^{-1}$ 可以由 $\boldsymbol{K}_w^{-1}(n)$ 得到,并且 $\boldsymbol{K}_{bw}^{-1} = \boldsymbol{K}_{fw}^{-1}$,所以 $\boldsymbol{K}_{fw}^{-1}$ 可以由 $\boldsymbol{K}_w^{-1}(n)$ 得到;通过式(13),则由 $\boldsymbol{K}_{fw}^{-1}$ 可以推出 $\boldsymbol{K}_w^{-1}(n+1)$. 因此可以得出结论:由 $\boldsymbol{K}_w^{-1}(n)$ 可以推出 $\boldsymbol{K}_w^{-1}(n+1)$,这样就实现了 $\boldsymbol{K}_w^{-1}(n)$ 的递归更新.

为了进一步降低计算复杂度,$\boldsymbol{k}_{\mu}^{\mathrm{T}}(n)$ 也可以进行递归更新,其推导过程如下:

$$\begin{aligned} \boldsymbol{k}_{\mu}^{\mathrm{T}}(n) &= \left(\frac{1}{w}\sum_{i=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \boldsymbol{X}_{n,w})\right) - \left(\frac{1}{w^2}\sum_{i=n-w}^{n-1}\sum_{j=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_j)\right)1_{1\times n} \\ &= \boldsymbol{k}_{m1}(n) - k_{m2}(n)1_{1\times n} \quad , \quad (14) \end{aligned}$$

$$\begin{aligned} \boldsymbol{k}_{m1}(n) &= \frac{1}{w}\sum_{i=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \boldsymbol{X}_{n,w}) \\ &= \left[\frac{1}{w}\sum_{i=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_{n-w}), \frac{1}{w}\sum_{i=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \boldsymbol{X}_{n,2:w})\right] \quad ,(15) \\ &= \left[\frac{1}{w}\left(\sum_{i=n-w}^{n-1}\alpha_{i,n-w} + \rho\right), \boldsymbol{k}_{mf}\right] \end{aligned}$$

$$\begin{aligned} \boldsymbol{k}_{m1}(n+1) &= \frac{1}{w}\sum_{i=n-w+1}^{n} k(\hat{\boldsymbol{r}}_i, \boldsymbol{X}_{n+1,w}) \\ &= \left[\frac{1}{w}\sum_{i=n-w+1}^{n} k(\hat{\boldsymbol{r}}_i, \boldsymbol{X}_{n+1,1:w-1}), \frac{1}{w}\sum_{i=n-w+1}^{n} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_n)\right] \\ &= \left[\frac{1}{w}(w\boldsymbol{k}_{mf} - \alpha + \beta), \frac{1}{w}\left(\sum_{i=n-w+1}^{n-1}\beta_{i,n+1} + \chi\right)\right] \\ &\quad ,(16) \end{aligned}$$

$$\begin{aligned} \boldsymbol{k}_{m2}(n+1) &= \frac{1}{w^2}\sum_{i=n-w+1}^{n}\sum_{j=n-w+1}^{n} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_j) \\ &= \frac{1}{w^2}\sum_{i=n-w}^{n-1}\sum_{j=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_j) - \frac{1}{w^2}\sum_{j=n-w}^{n-1} k(\hat{\boldsymbol{r}}_{n-w}, \hat{\boldsymbol{r}}_j) \\ &\quad - \frac{1}{w^2}\sum_{i=n-w}^{n-1} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_{n-w}) + \frac{1}{w^2}\sum_{i=n-w+1}^{n} k(\hat{\boldsymbol{r}}_n, \hat{\boldsymbol{r}}_j) \\ &\quad + \frac{1}{w^2}\sum_{i=n-w+1}^{n} k(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_n) + k(\hat{\boldsymbol{r}}_{n-w}, \hat{\boldsymbol{r}}_{n-w}) - k(\hat{\boldsymbol{r}}_n, \hat{\boldsymbol{r}}_n) \\ &= \frac{1}{w^2}(w^2 k_{m2}(n) - 2\sum_{i=n-w}^{n-1}\alpha_{i,n-w} + 2\sum_{i=n-w+1}^{n}\beta_{i,n} + \rho - \chi) \\ &\quad ,(17) \end{aligned}$$

其中,$\boldsymbol{X}_{n,2:w}$ 包含 n 时刻局部因果滑动阵列窗中 2^{th} 到 w^{th} 的像元向量,$\boldsymbol{X}_{n+1,1:w-1}$ 包含 $n+1$ 时刻局部因果滑动阵列窗中 1^{th} 到 $(w-1)^{\mathrm{th}}$ 的像元向量. 由 $\boldsymbol{k}_{m1}(n)$ 可以得到 $\boldsymbol{k}_{mf}$,通过式(16)可以进一步推出 $\boldsymbol{k}_{m1}(n+1)$. 根据式(17),$k_{m2}(n+1)$ 可以由 $k_{m2}(n)$ 获得. 因此,$\boldsymbol{k}_{\mu}^{T}(n)$ 可以通过式(14)递归更新.

2.2 算法流程及复杂度分析

由 n 时刻信息递归更新 $n+1$ 时刻 $\delta^{\mathrm{LRT\text{-}KRXD}}(\Phi(\hat{\boldsymbol{r}}_{n+1}))$ 的具体算法流程设计如下:

(1) 利用式(8),从 n 时刻核协方差逆矩阵 $\boldsymbol{K}_w^{-1}(n)$ 中提取 $\boldsymbol{\kappa}$;

(2) 将步骤(1)得到的 $\boldsymbol{\kappa}$ 带入式(11)求出 $\boldsymbol{K}_{bw}^{-1}$;

(3) 由于 $\boldsymbol{K}_{bw}^{-1} = \boldsymbol{K}_{fw}^{-1}$,将步骤(2)得到的 $\boldsymbol{K}_{bw}^{-1}$ 带入式(13),这样就可以求出 $n+1$ 时刻的核协方差逆矩阵 $\boldsymbol{K}_w^{-1}(n+1)$;

(4) 利用式(15)和(16),可以由 n 时刻 $\boldsymbol{k}_{m1}$ 求出 $n+1$ 时刻 $\boldsymbol{k}_{m1}(n+1)$;同理利用式(17),可以由 n 时刻 $\boldsymbol{k}_{m2}(n)$ 求出 $n+1$ 时刻 $\boldsymbol{k}_{m2}(n+1)$. 再利用式(14)的求解形式,就可以得到 $n+1$ 时刻的估计均

值 $\boldsymbol{k}_{\hat{\mu}}^{\mathrm{T}}(n+1)$;

(5) 根据步骤(3)得到的 $\boldsymbol{k}_{\omega}^{-1}(n+1)$ 和步骤(4)得到的 $\boldsymbol{k}_{\hat{\mu}}^{\mathrm{T}}(n+1)$,利用式(6)的求解形式,就可以求出 $n+1$ 时刻 $\delta^{\mathrm{LRT\text{-}KRXD}}(\Phi(\hat{\boldsymbol{r}}_{n+1}))$.

算法的复杂度主要取决于所包含乘法项的个数. 为了更加直观地分析 LRT-KRXD 的时效性,表 1 分别给出了执行一次像元检测 LKRXD 和 LRT-KRXD 中四项计算成分所包含的乘法个数及复杂度. 由于避免了大量的核运算求解和矩阵逆处理,LKRXD 中 $\boldsymbol{K}_w$ 和 $\boldsymbol{K}_w^{-1}$ 的复杂度分别由 3 阶降到 2 阶. 因此,相比于 LKRXD,LRT-KRXD 的执行需要更少的处理时间.

表 1　四种成分的计算复杂度

Table 1　Computational complexity of four components

	LKRXD		LRT-KRXD	
	乘法数量	复杂度	乘法数量	复杂度
$\boldsymbol{K}_{\hat{r}}^{\mathrm{T}}$	$\omega(L+1)$	$O(\omega L)$	$\omega(L+1)$	$O(\omega L)$
$\boldsymbol{K}_{\hat{\mu}}^{\mathrm{T}}$	2ω	$O(\omega)$	ω	$O(\omega)$
$\boldsymbol{K}_w$	$\omega^2 L$	$O(\omega^2 L)$	$(\omega-1)L$	$O(\omega L)$
$\boldsymbol{K}_w^{-1}$	ω^3	$O(\omega^2)$	$3\omega^2-4\omega+2$	$O(\omega^2)$

3　实验

为了验证 LRT-KRXD 工作的有效性,利用两幅真实高光谱数据进行了实验.

3.1　数据描述

3.1.1　AVIRIS 数据

AVIRIS 数据是由 AVIRIS 高光谱仪拍摄的圣地亚哥海军基地图像. 该图像的大小为 400 × 400 像素,覆盖了 0.4 ~ 1.8 μm 的波长范围,光谱分辨率为 10 nm,空间分辨率为 3.5 m. 它拥有 224 个波段,去掉水吸收和信噪比低的波段后,剩余 126 个波段用于后续处理. 我们截取 51 × 50 像素的图像用于实验分析,其中包含的三架飞机作为异常. 其第一波段图像和真实地物分布如图 2 (a)和(b)所示.

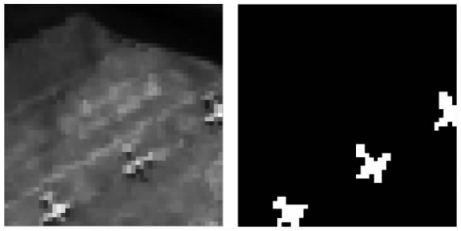

图 2　(a) 第一波段图像,(b) 真实地物分布图

Fig. 2　(a) The image scene of the first band, (b) ground truth of the image

3.1.2　ROSIS 数据

ROSIS 数据是由 ROSIS 光谱仪拍摄的,它覆盖了意大利北部的 Pavia 中心. 该光谱仪的波长范围是 0.43 ~ 0.86 μm,光谱分辨率为 4 nm,空间分辨率为 1.3 m. 该图像的大小为 1 096 × 715 像素,可用波段数为 102. 我们选择其中 105 × 105 像素的图像用于实验处理. 其第一波段图像和真实地物分布如图 3 (a)和(b)所示.

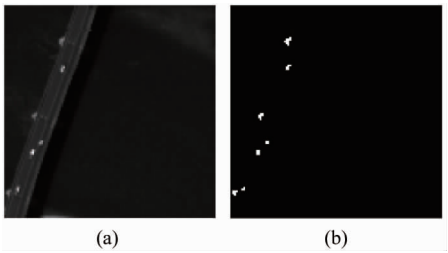

图 3　(a) 第一波段图像,(b) 真实地物分布图

Fig. 3　(a) The image scene of the first band, (b) ground truth of the image

3.2　核参数 d 对 LRT-KRXD 的影响

关于多项式核参数 d 对 LRT-KRXD 的影响进行了实验. 利用交叉验证,在 AVIRIS 数据实验中,局部因果滑动阵列窗的长度选为 90;在 ROSIS 数据实验中,局部因果滑动阵列窗的长度选为 70.

接收机操作特性(ROC)能够定量分析检测概率和虚警概率之间的关系,而 ROC 曲线下面积(AUC)也是判别检测性能的重要标准,AUC 越接近 1,表明算法性能越好. 图 4 展示了两幅数据中不同数值 d 的 AUC 柱状图. 在 AVIRIS 数据中,其 AUC 随着 d 的增加而递减,当 d 的数值为 1 时,它拥有最大的曲线下面积. 在 ROSIS 数据中,当 d 从 1 上升到 2 时,其 AUC 达到最高,而随着 d 的继续增加,它的 AUC 逐渐下降.

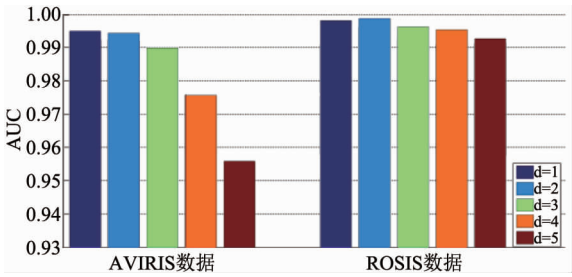

图 4　两幅数据中不同数值 d 的 AUC 柱状图

Fig. 4　Bars of AUC with different d in two datasets

3.3　局部因果滑动阵列窗长对 LRT-KRXD 的影响

关于局部因果滑动阵列窗的长度 w 对 LRT-

KRXD的影响进行了实验. 通过交叉验证,在AVIRIS数据实验中,多项式核参数d的数值设置为1;在ROSIS数据实验中,多项式核参数d的数值设置为2.

不同的高光谱数据对局部因果滑动阵列窗长度的要求是不一样的. 如图5所示,针对AVIRIS数据,由于地物分布较为复杂,在窗长 w 从20上升到80期间,其AUC增加较为明显,当窗长 w 大于或等于90时,AUC达到最高并基本保持不变;而对于ROSIS数据,它的地物分布较为简单,受窗长 w 的影响不大,当窗长 w 从20上升到70,其AUC略有增加,随着窗长 w 继续上升,则AUC保持稳定.

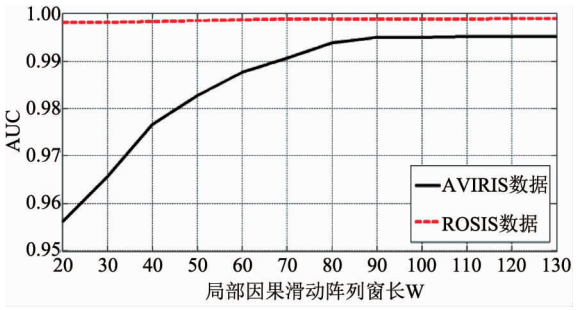

图5 两幅数据中不同局部因果滑动阵列窗长的AUC曲线图

Fig. 5 Curves of AUC with different local causal sliding array window width in two datasets

3.4 LRT-KRXD的检测性能

为了验证LRT-KRXD的有效性,在本节,将LRT-KRXD与两种实时异常检测算法LRT-RXD、GRT-RXD及其原算法LKRXD进行了实验对比. 在AVIRIS数据实验中,LRT-KRXD中的多项式核参数 d 设置为1,其局部因果滑动阵列窗长 w 选为90;LRT-RXD的局部因果滑动阵列窗长 w 设为300;LKRXD中的多项式核参数 d 选择为1,其滑动双窗的大小设置为5/11. 而在ROSIS数据实验中,LRT-KRXD中的多项式核参数 d 设置为2,其局部因果滑动阵列窗长 w 选为70;LRT-RXD的局部因果滑动阵列窗长 w 设为300;LKRXD中的多项式核参数 d 选择为2,其滑动双窗的大小设置为5/11.

图6给出了不同数据下四种算法的ROC曲线. 对于AVIRIS数据和ROSIS数据,LRT-KRXD的ROC曲线基本和其原算法LKRXD完全一样,并且在较低的虚警概率下,其检测概率就已经达到1. 而相比于核算法,两种实时RX算法(GRT-RXD和LRT-RXD)则具有较低的检测精度.

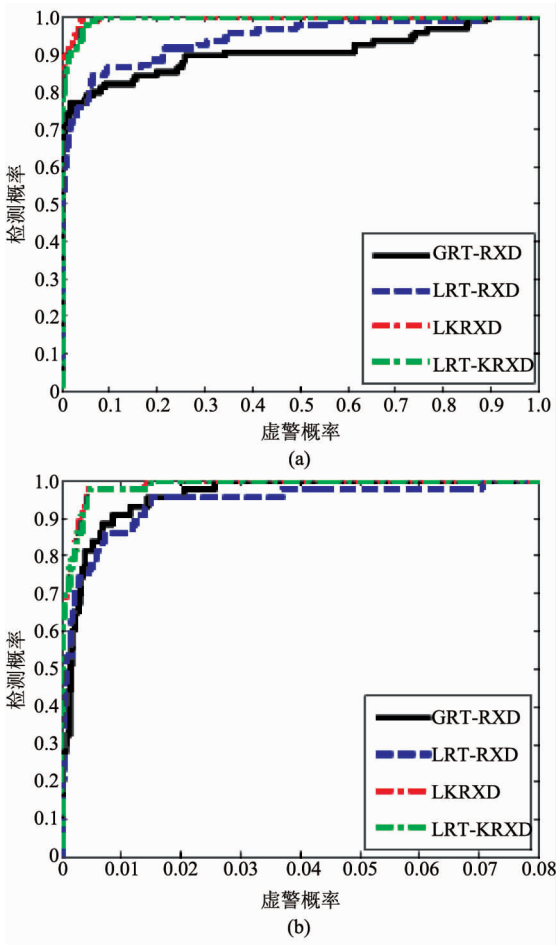

图6 接收机操作特性(ROC)曲线 (a) AVIRIS数据,(b) ROSIS数据

Fig. 6 Receiver operation characteristic (ROC) curves (a) AVIRIS dataset, (b) ROSIS dataset

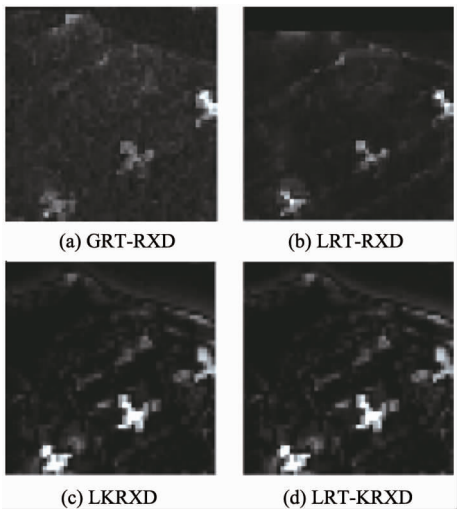

图7 AVIRIS数据实验灰度图

Fig. 7 The gray image of experiment in AVIRIS dataset

图7和图8分别给出了AVIRIS数据和ROSIS数据中四种算法的灰度结果. LRT-KRXD和LKRXD

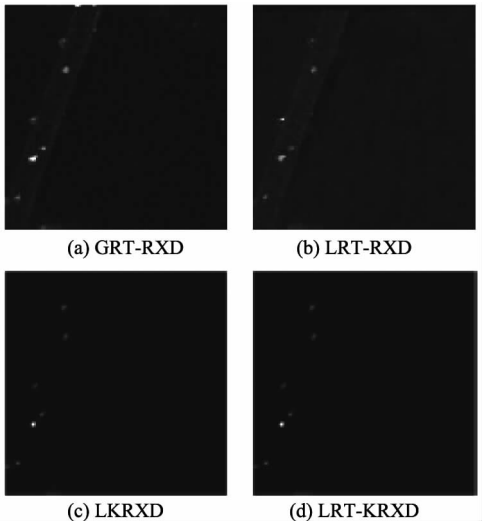

图 8　ROSIS 数据实验灰度图
Fig. 8　The gray image of experiment in ROSIS dataset

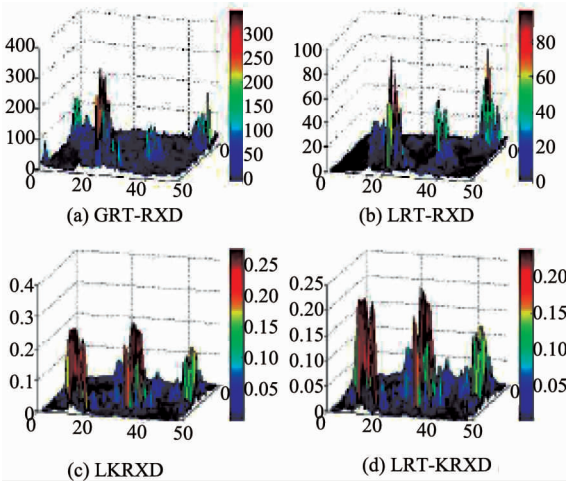

图 9　AVIRIS 数据实验 3D 图
Fig. 9　The 3D plots of experiment in AVIRIS dataset

的灰度输出基本相同,并且 GRT-RXD 和 LRT-RXD 的灰度输出依然还受到背景的干扰,而 LRT-KRXD 和 LKRXD 灰度图像中的异常目标却更加凸显.

为了对实验结果做出更加形象化的描述,图 9 和图 10 展示了两幅数据中四种算法的 3 维(3D)输出结果. 可以看出 LRT-KRXD 和 LKRXD 的 3D 图基本一致,并且相对于 GRT-RXD 和 LRT-RXD,它们具有较强的背景抑制能力.

图 11 和 12 分别给出了 AVIRIS 数据和 ROSIS 数据中 LRT-KRXD 的实时处理进程. 利用局部因果滑动阵列窗,LRT-KRXD 实现了实时的逐像元检测. 实时处理随着检测的执行拥有不同的背景抑制能力. 如图 12 所示,(a)、(b)和(c)都清晰地检测到弱

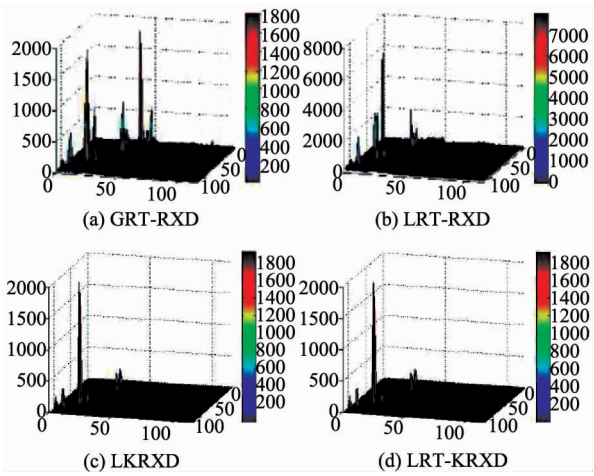

图 10　ROSIS 数据实验 3D 图
Fig. 10　The 3D plots of experiment in ROSIS dataset

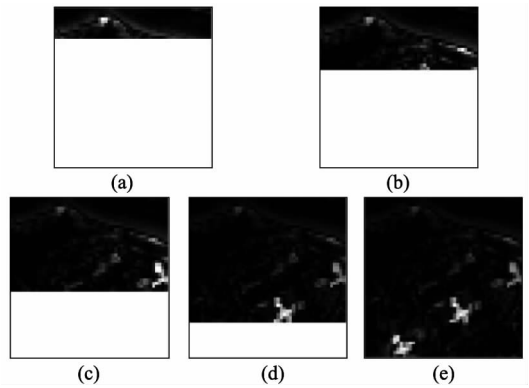

图 11　AVIRIS 数据 LRT-KRXD 实时处理进程
Fig. 11　Real-time progressive procedures of LRT-KRXD in AVIRIS dataset

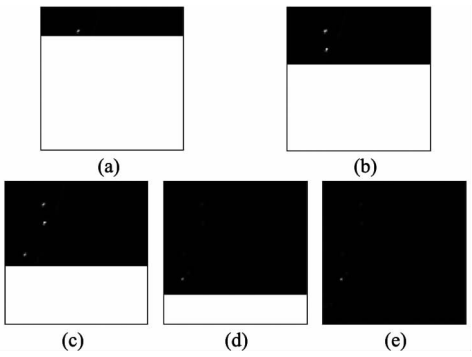

图 12　ROSIS 数据 LRT-KRXD 实时处理进程
Fig. 12　Real-time progressive procedures of LRT-KRXD in ROSIS dataset

异常,而强异常出现后,它们又变得十分模糊. 这种渐进式的实时方法可以有效地检测到湮没在全局中的弱异常.

3.5　计算时间分析

为了验证所提算法 LRT-KRXD 的时效性,将

LRT-KRXD与GRT-RXD、LRT-RXD和LKRXD在处理时间上做了对比.用于实验仿真的电脑环境是64位操作系统,INTEL酷睿I7-4720HQ的CPU,主频为3.5GHz以及内存为16G;仿真软件是MATLAB 2014A.表2给出了四种检测器在AVIRIS数据和ROSIS数据实验中所用的时间.和其原始算法LKRXD相比,LRT-KRXD在AVIRIS数据实验中加速了近33倍,在ROSIS数据实验中加速了42倍.

表2 四种检测器的计算时间
Table 2 Computing time of four detectors

	总处理时间/s	
	AVIRIS数据	ROSIS数据
GRT-RXD	0.764	2.769
LRT-RXD	0.824	3.352
LKRXD	42.349	210.523
LRT-KRXD	1.289	4.975

4 结论

提出了基于多项式核局部RX算法的实时处理,通过引用Hermitian矩阵分块求逆引理和Woodbury引理,推导出了核协方差逆矩阵和估计背景均值的递归更新公式,降低了算法复杂度,将实时异常检测从RX检测器移植到了LKRX检测器,在满足时效性的同时,也获得了更好的检测精度.并且对LKRXD和LRT-KRXD的计算复杂度进行了分析和对比,从理论上证明了利用递归更新公式可以显著降低算法的复杂度.通过对LRT-KRXD进行实验仿真,所提算法在保持LKRXD的检测精度下,大大地提高了检测效率.

致谢

本论文的实验数据是由美国威斯康星州立大学国际知名遥感和高性能计算领域专家Bormin Huang教授提供,并且在理论和实验中得到了他的指导,谨致谢意.

References

[1] Reed I S, Yu X. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution [J]. *IEEE Transactions on Acoustics Speech & Signal Processing*. 1990, **38**(10): 1760-1770.

[2] Taitano Y P, Geier B A, Bauer K W. A locally adaptable iterative RX detector [J]. *Journal on Advances in Signal Processing*. 2010, **2010**(1): 1-10.

[3] Molero J M, Garzon E M, Garcia I, *et al*. Analysis and optimizations of global and local versions of the RX algorithm for anomaly detection in hyperspectral data [J]. *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*. 2013, **6**(2): 801-814.

[4] Schölkopf B, Smola A. *Learning with kernels* [M]. MIT Press, Cambridge, MA, 2002.

[5] Kwon H, Nasrabadi N M. Kernel RX-algorithm: a nonlinear anomaly detector for hyperspectral imagery [J]. *IEEE Transactions on Geoscience & Remote Sensing*. 2005, **43**(2): 388-397.

[6] Chen S Y, Wang Y, Wu C C, *et al*. Real-time causal processing of anomaly detection for hyperspectral imagery [J]. *Aerospace & Electronic Systems IEEE Transactions on*. 2014, **50**(2):1511-1534.

[7] 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*. 2012, **9**(3):503-517.

[8] ZHAO Chun-Hui, WANG Yu-Lei, LI Xiao-Hui. A real-time anomaly detection algorithm for hyperspectral imagery based on causal processing [J]. *Journal of Infrared and Millimeter Waves*(赵春晖, 王玉磊, 李晓慧. 一种新型高光谱实时异常检测算法. **红外与毫米波学报**),2015, **34**(1):114-121.

[9] Zhao C, Wang Y, Qi B, *et al*. Global and local real-Time anomaly detectors for hyperspectral remote sensing imagery [J]. *Remote Sensing*. 2015, **7**(4):3966-3985.

[10] Noble B, Daniel J W. *Applied Linear Algebra* [M]. New Jersey: Prentice-Hall, 1997.

[11] Olver P J, Shakiban C. *Applied Linear Algebra* [M]. New Jersey: Prentice-Hall, 2006.

[12] Kailath T. *Linear Systems*, *Englewood Cliffs* [M]. New Jersey: Prentice-Hall, 1980.