

外辐射源雷达多通道时域杂波抑制算法并行实现

陈伟 万显荣* 张勋 饶云华 程丰

(武汉大学电子信息学院 武汉 430072)

摘要: 直达波及杂波抑制是外辐射源雷达目标信息提取的关键步骤之一。扩展杂波相消批处理(ECA-B)算法是一种有效的时域杂波抑制算法,但算法空间和时间复杂度极大,在处理多通道(或多波束)数据时尤其是如此。结合图形处理器(GPU)数据吞吐量大、浮点运算能力强的优点,该文提出一种适用于GPU处理的多通道ECA-B时域杂波抑制并行算法。首先推导了多通道ECA-B算法的原理,避免了原算法分别对单通道进行单独处理的计算冗余问题。然后针对其中耗时最大的自相关矩阵计算,提出一种迭代计算方法,使时间和空间复杂度均降至常规方法的约 $1/K$ (K 为杂波自由度)。最后给出了改进算法的GPU并行实现方案。仿真和实测结果验证了算法的准确性和实效性。

关键词: 外辐射源雷达; 时域杂波抑制; 扩展杂波相消(ECA); 并行实现

中图分类号: TN958.57

文献标识码: A

文章编号: 2095-283X(2014)06-0686-08

DOI: 10.12000/JR14157

Parallel Implementation of Multi-channel Time Domain Clutter Suppression Algorithm for Passive Radar

Chen Wei Wan Xian-rong Zhang Xun Rao Yun-hua Cheng Feng

(School of Electronic Information, Wuhan University, Wuhan 430072, China)

Abstract: Cancellation of clutter and multi-path is one of the key steps in passive radar target information extraction. Extensive Cancellation Algorithm Batches (ECA-B) is an effective time-domain clutter suppression algorithm, but with high time and space complexity, and even higher with multi-channel (or multi-beam) data processing. Combining high memory throughput and tremendous computational horsepower of GPU graphics processor, this paper proposes a multi-channel ECA-B algorithm which is suitable for parallel implementation on GPUs. Firstly, the principle of multi-channel ECA-B algorithm is derived, avoiding the redundancy of processing each channel singly. Then an iterative calculation method is presented for reducing the biggest time-consuming calculation of the correlation matrix, so that time and space complexity are both reduced to $1/K$ (K is clutter's degree of freedom) of the conventional method. Finally, the full GPU parallel implementation of the algorithm is given. The simulation and experimental results verify the accuracy and effectiveness of the proposed algorithm.

Key words: Passive radar; Time domain clutter suppression; Extensive Cancellation Algorithm (ECA); Parallel implementation

1 引言

外辐射源雷达^[1-7]接收到的目标回波幅度弱,易被直达波、多径杂波及干扰淹没,因而在目标检测前需要进行杂波抑制处理,其中杂波抑制的实时处理是外辐射源雷达实时化关键步骤之一。常用的如LMS, NLMS, BLMS, RLS, LSL等闭环自适应滤波算法^[7-14]属于迭代类算法,处理时数据按一定

批次输入自适应滤波器,由于每批次数据量较少,算法时间和空间复杂度相对较低,易于在FPGA, DSP或CPU上实现。但该类算法面临收敛速度、稳定性和适应能力等难以确定的问题。文献[15,16]提出了基于最小二乘(Least Square, LS)算法的扩展杂波相消(Extensive Cancellation Algorithm, ECA)和扩展相消批处理算法(Extensive Cancellation Algorithm Batches, ECA-B),直接利用输入信号和期待响应的信息计算自适应权值,无需引入输出信息的反馈,避免了迭代类算法的收敛问题,因而不会引起输出误差的发散,是一种稳健的自适应算法。但由于其求解过程中需要计算高维复矩阵乘法和矩阵求逆,耗时较长;且为减少估计误差,通常

2014-12-10 收到, 2014-12-17 改回; 2014-12-23 网络优先出版
国家自然科学基金(61331012, 61371197), 教育部博士点基金
(20120141110077)和中央高校基本科研业务费专项资金
(2014212020204)资助课题

*通信作者: 万显荣 xrw@whu.edu.cn

需要提高样本数目以取得较好的滤波效果,导致空间复杂性增加。原理上, ECA 和 ECA-B 算法针对单通道数据处理,以 ECA 为例,采用常规算法,对单个通道处理时,时间和空间复杂度分别为 $O(NK^2 + K^3)$ 和 $O(NK)$,其中 N 为数据点数, K 为杂波子空间自由度。当通道数为 M 时,时间复杂度和空间复杂度分别为 $O(MNK^2 + MK^3)$ 和 $O(MNK)$,算法存储空间占用和计算量与通道数目成正比,若采用传统基于 FPGA, DSP 或 ARM 等微处理器的实时处理方案,存储空间和硬件计算能力需求难以满足。

近年来,随着大规模集成电路技术的发展,中央处理器(Central Processing Unit, CPU)和图形处理器(Graphic Processing Unit, GPU)性能与日俱增,为外辐射源雷达的实时信号处理提供了新的解决方案。相对于传统方案,其具有成本低廉、结构简单、开发容易等优点。国内外已有大量文献证实利用高性能 CPU 和 GPU 实现外辐射雷达信号实时处理的可行性^[9-13]。由于 GPU 能够并行计算,且具有存储空间充足、数据吞吐量大、浮点运行能力强等优点, ECA 和 ECA-B 算法更适合利用 GPU 予以实现。

针对外辐射源雷达多通道时域杂波抑制问题,本文提出适合 GPU 处理的多通道 ECA-B 杂波抑制算法,既充分利用了 ECA-B 算法稳定、鲁棒性好的优点,又解决了其多通道处理时时间和空间复杂度高的问题。首先简要阐述了 ECA 和 ECA-B 算法基本原理,通过扩充列向量的方法推导出能够同时进行多通道或多波束杂波抑制的算法,降低了算法复杂度。然后针对改进算法中自相关矩阵 R 计算和空间冗余,提出一种结合自相关矩阵的共轭对称性降低冗余的迭代计算方案。再次,通过合理设计 GPU 核函数,利用高带宽片内寄存器、共享存储器等 GPU 资源,减少不必要的访存,实现数据的重用,降低程序运行时间。最后,通过仿真和实测数据处理对算法进行验证。

2 基于最小二乘的多通道 ECA-B 算法

假设照射源发射信号为 $s(t)$,则外辐射源雷达参考通道和监测通道回波可以分别表示为 $s_{\text{ref}}(t)$, $s_{\text{sur}}(t)$ 。

$$s_{\text{ref}}(t) = s(t - \tau) + n_{\text{ref}}(t) \quad (1)$$

$$s_{\text{sur}}(t) = \sum_{i=1}^P \alpha_i s(t - \tau_i) + \sum_{i=1}^Q \beta_i s(t - \gamma_i) e^{j2\pi f_{di}t} + n_{\text{sur}}(t) \quad (2)$$

其中 $s(t)$ 为外辐射源发射信号, τ 为参考通道回波相

对发射信号延时, α_i, τ_i 分别为 P 条多径中第 i 条回波衰减和时延, $\beta_i, \gamma_i, f_{di}$ 分别为 Q 个目标中第 i 个目标的衰减、时延和多普勒频移。 $n_{\text{ref}}(t), n_{\text{sur}}(t)$ 分别为参考信号和监测信号中的噪声,一般情况下可以认为是高斯白噪声。式(2)中并没有对杂波干扰进行建模,实际上增加杂波干扰只需增加杂波子空间的自由度,也即杂波抑制滤波器的阶数^[10]。杂波抑制处理即估计出其中的 α_i, τ_i ,构造杂波子空间,然后从监测信号中减去即可得到目标回波,最后通过后续相干积累提取出目标信号的时延 γ_i 和多普勒 f_{di} 。把监测通道和参考通道信号的关系写成矩阵形式

$$\mathbf{s}_{\text{sur}} = \mathbf{X}_{\text{ref}} \mathbf{w}_{\text{sur}} + \mathbf{n}_{\text{sur}} \quad (3)$$

其中

$$\mathbf{s}_{\text{sur}} = \begin{bmatrix} s_{\text{sur}}(K-1) & s_{\text{sur}}(K-1) & \cdots & s_{\text{sur}}(N+K-2) \end{bmatrix}^T \quad (4)$$

$$\mathbf{X}_{\text{ref}} = \begin{bmatrix} s_{\text{ref}}(0) & s_{\text{ref}}(1) & \cdots & s_{\text{ref}}(K-1) \\ s_{\text{ref}}(1) & s_{\text{ref}}(2) & \cdots & s_{\text{ref}}(K) \\ \vdots & \vdots & \ddots & \vdots \\ s_{\text{ref}}(N-1) & s_{\text{ref}}(N-2) & \cdots & s_{\text{ref}}(N+K-2) \end{bmatrix} \quad (5)$$

$$\mathbf{w}_{\text{sur}} = \begin{bmatrix} w_{\text{sur}}(0) & w_{\text{sur}}(1) & \cdots & w_{\text{sur}}(K-1) \end{bmatrix}^T \quad (6)$$

$$\mathbf{n}_{\text{sur}} = \begin{bmatrix} n_{\text{sur}}(K-1) & n_{\text{sur}}(K) & \cdots & n_{\text{sur}}(N+K-2) \end{bmatrix}^T \quad (7)$$

$$\mathbf{s}_{\text{ref}} = \begin{bmatrix} s_{\text{ref}}(0) & s_{\text{ref}}(1) & \cdots & s_{\text{ref}}(N+K-2) \end{bmatrix}^T \quad (8)$$

利用最小二乘法容易得到抑制后信号为:

$$\tilde{\mathbf{s}}_{\text{sur}} = \mathbf{s}_{\text{sur}} - \mathbf{X}_{\text{ref}} (\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}})^{-1} \mathbf{X}_{\text{ref}}^H \mathbf{s}_{\text{sur}} \quad (9)$$

ECA 算法用于多通道杂波抑制处理时,需重复调用算法多次计算式(9),空间占用和计算量成倍增加。而计算过程中 $\mathbf{X}_{\text{ref}} (\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}})^{-1} \mathbf{X}_{\text{ref}}^H$ 部分是可以共用,利用矩阵乘法可以分块计算的性质,通过对 $\mathbf{s}_{\text{sur}}, \mathbf{w}_{\text{sur}}$ 和 \mathbf{n}_{sur} 按列扩展,避免了多次计算导致的计算冗余,仅仅增加部分计算量,即可实现同时对多通道数据进行杂波抑制。扩充后的 ECA 算法可以表示为:

$$\mathbf{S}_{\text{sur}} = \mathbf{X}_{\text{ref}} \mathbf{W} + \mathbf{N} \quad (10)$$

其中 \mathbf{X}_{ref} 不变, $\mathbf{S}_{\text{sur}}, \mathbf{W}, \mathbf{N}$ 如式(10)~式(12)所示。其中 M 表示监测通道总数或波束总数。

$$\mathbf{S}_{\text{sur}} = \begin{bmatrix} \mathbf{s}_{\text{sur},0} & \mathbf{s}_{\text{sur},1} & \cdots & \mathbf{s}_{\text{sur},M-1} \end{bmatrix} \quad (11)$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_{\text{sur},0} & \mathbf{w}_{\text{sur},1} & \cdots & \mathbf{w}_{\text{sur},M-1} \end{bmatrix} \quad (12)$$

$$N = \begin{bmatrix} n_{\text{sur},0} & n_{\text{sur},1} & \cdots & n_{\text{sur},M-1} \end{bmatrix} \quad (13)$$

利用最小二乘算法,容易求得权矢量

$$\tilde{W} = (\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}})^{-1} \mathbf{X}_{\text{ref}}^H \mathbf{s}_{\text{sur}} \quad (14)$$

故监测信号中的直达波及多径分量为:

$$\tilde{\mathbf{s}}_{\text{sur}} = \mathbf{X}_{\text{ref}} \tilde{W} = \mathbf{X}_{\text{ref}} (\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}})^{-1} \mathbf{X}_{\text{ref}}^H \mathbf{s}_{\text{sur}} \quad (15)$$

监测信号杂波抑制后残余信号(包含目标信息)

为:

$$\begin{aligned} \tilde{\mathbf{s}}_{\text{sup}} &= \mathbf{s}_{\text{sur}} - \tilde{\mathbf{s}}_{\text{sur}} \\ &= \mathbf{s}_{\text{sur}} - \mathbf{X}_{\text{ref}} (\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}})^{-1} \mathbf{X}_{\text{ref}}^H \mathbf{s}_{\text{sur}} \quad (16) \end{aligned}$$

$\tilde{\mathbf{s}}_{\text{sur}}$ 相当于 \mathbf{s}_{sur} 在 \mathbf{s}_{ref} 向量上的投影,其中 $\mathbf{P} =$

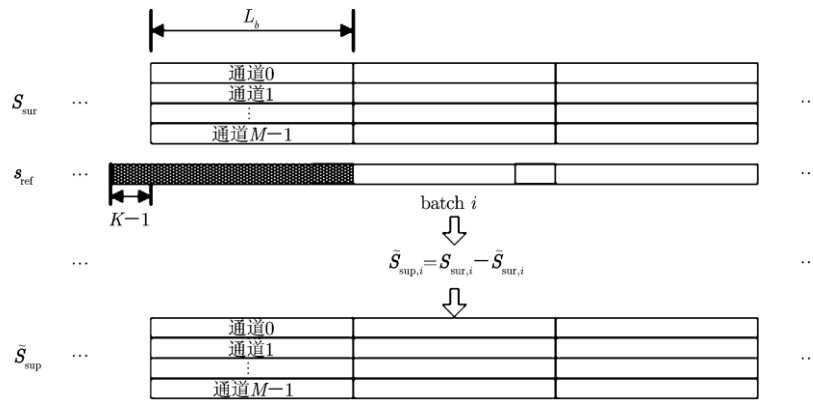


图1 多通道 ECA-B 扩展算法框图

Fig. 1 Block diagram of the multi-channel ECA-B algorithm

3 基于迭代计算的改进多通道 ECA-B 算法

3.1 计算量分析

式(16)中计算矩阵相乘既可以从左向右,也可以从右向左计算。设每段数据长度为 N , 自适应滤波器阶数为 K , 监测通道总数为 M , 则 \mathbf{X}_{ref} , \mathbf{s}_{sur} 分别为 $N \times K$ 和 $K \times M$ 矩阵, 显然直接计算 $\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}}$ 计算量为 $K^2(N-1)$ 次复数浮点加法和 K^2N 复数浮点乘法, K 阶矩阵求逆采用高斯消元法计算时间复杂度为 $O(K^3)$ 。经推导, 不难得到其余计算部分, 若从左往右直接计算量为 $N^2(K+M) + NK^2 - N^2 - NK$ 次复数浮点加法和 $N^2(K+M) + NK^2$ 次复数浮点乘法, 而从右向左直接计算量为 $KM(2N+K-2)$ 次复数浮点加法和 $KM(2N+K)$ 次复数浮点乘法。由于 $N \gg K \gg M$, 故后者计算量远小于前者, 故采用从右向左计算的方式, 分为 6 步进行, 如图 2 所示。同时可知 $\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}}$ 的计算量远远大于剩余部分, 且算法中存在 \mathbf{X}_{ref} 和 $\mathbf{X}_{\text{ref}}^H$, 以单精度浮点型数据为例, 其占用显存为 $8NK$ 字节, 取数据长度 $N = 256 \times 1024$, $K = 128$, 则保存 \mathbf{X}_{ref} 共需 256 MB 存储空间, 若进一步加大 K

$\mathbf{X}_{\text{ref}} (\mathbf{X}_{\text{ref}}^H \mathbf{X}_{\text{ref}})^{-1} \mathbf{X}_{\text{ref}}^H$ 称为投影算子, 所以 ECA 算法也称为正交子空间投影算法。

ECA-B 算法相当于对 ECA 算法的数据进行分块处理, 并在子块中利用 ECA 算法进行杂波抑制, 最终组合各个子块中的数据得到抑制结果, 如图 1 所示。通过分块处理, 不仅降低了每批次计算量和存储空间需求, 而且能够进一步抑制零频附近慢速目标引入的杂波, 并且由于子块数据长度变短, 使得其能更快速地适应信道变化。但从实现方式来看, ECA 和 ECA-B 算法实现和计算量基本相同, 只有子块数据长度的不同, 下文不做区分。

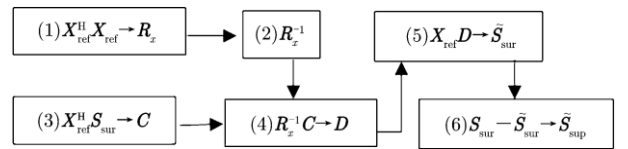


图2 式(16)计算流程

Fig. 2 Calculation flow of equation (16)

或 N 或者采用双精度浮点型数据以提高运算精度, 所需存储空间更高, 大大降低了算法的实用性。

3.2 自相关矩阵 R_x 的迭代计算方案

考虑到 \mathbf{X}_{ref} 的特殊结构, 即 $X_{\text{ref}}(i, j) = s_{\text{ref}}(i+j)$, $0 \leq i \leq N-1$, $0 \leq j \leq K-1$, 容易得到

$$\begin{aligned} R_x(i, j) &= \mathbf{X}_{\text{ref}}^H(:, i) \mathbf{X}_{\text{ref}}(:, j) = \sum_{k=0}^{N-1} s_{\text{ref}}^*(k+i) s_{\text{ref}}(k+j), \\ &0 \leq i, j \leq K-1 \quad (17) \end{aligned}$$

显然 $R_x(i, j) = R_x^*(j, i)$, 故只需要计算 R_x 矩阵的上三角部分或下三角部分, 通过共轭对称性可以直接得到另一部分的值, 通过此方法可以减少一半的计算量。以计算上三角部分为例, 当 $1 \leq i \leq j \leq K-1$ 时,

$$R_x(i-1, j-1) = \sum_{k=0}^{N-1} s_{\text{ref}}^*(k+i-1)s_{\text{ref}}(k+j-1) \quad (18)$$

$$R_x(i, j) - R_x(i-1, j-1) = s_{\text{ref}}^*(N+i-1)s_{\text{ref}}(N+j-1) - s_{\text{ref}}^*(i-1)s_{\text{ref}}(j-1) \quad (19)$$

由式(18), 式(19)可知, 关于 $R_x(i, j)$ 的计算有极大的计算冗余, 直接计算需要 N 次复数浮点乘法和 $N-1$ 次复数浮点加法, 而若利用对角元素的前一计算值 $R_x(i-1, j-1)$, 则只需要增加 2 次复数乘和 2 次复数加法即可得到。从而可以通过计算矩阵第 1 行或第 1 列值, 然后迭代计算剩余 $K-1$ 行或列值, 计算出 R_x , 显然此算法可以极大降低计算量。以先计算第 1 行为例, 实现示意图如图 3 所示, 具体计算流程如下:

- (1) 计算矩阵 R_x 第 $i=0$ 行值, 即 $R_x(0, j)$, 其中 $0 \leq j \leq K-1$;
- (2) 计算出第 $i=i+1$ 行的值。若列序号 $j < i$, 则 $R_x(i, j) = R_x^*(j, i)$, 否则通过式(19)计算 $R_x(i, j)$;
- (3) 重复步骤(2), 若 $i = K-1$ 则计算完毕后停止。

整理两种计算方法计算量如表 1 所示, 由于一般情况下 $N \gg K$, 故两种算法计算量之比近似为 K 。故改进方法从原理上可以大大减少计算量。

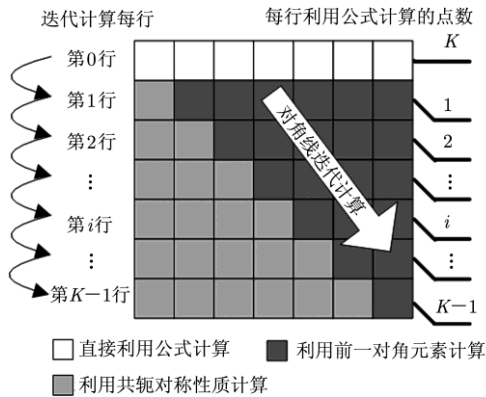


图 3 迭代计算 $R_x(i, j)$ 框图

Fig. 3 Block diagram of iterative algorithm of calculating $R_x(i, j)$

表 1 递推迭代法和矩阵相乘法计算 R_x 计算量

Tab. 1 Time complexity of calculating R_x by iterative algorithm and conventional matrix multiplication algorithm

运算	复数乘法	复数加法
矩阵相乘法 (方法 1)	NK^2	$(N-1)K^2$
递推迭代算法 (方法 2)	$NK + (K-1)K$	$(N-1)K + (K-1)K$
比值(方法 1/ 方法 2)	$\frac{NK}{N+K-1}$	$\frac{(N-1)K}{(N-1)+K-1}$

4 改进算法的 GPU 并行实现及优化

2007 年 2 月, NVIDIA 公司正式提出统一计算设备架构(Compute Unified Device Architecture, CUDA), 它结合图形处理器具有的数据吞吐量大, 浮点运算能力强的优点, 可以解决复杂的科学运算问题, 此外它利用广为熟悉的 C 语言编程, 可以大幅度缩短开发周期, 减少成本。CUDA 架构的公布, 使得基于 GPU 的通用计算成为可能。在 CUDA 编程模型中, CPU 作为主机端, 负责 GPU 设备端的资源分配、运行参数配置等工作, GPU 设备端按照粒度粗分为 Grid, Block 和 Thread, 主要用于并行计算。实际编程中, 将能够并行计算的相关部分写成一个或多个核函数(kernel), 由 CPU 控制多个 kernel 流程, GPU 控制每个 kernel 内部的调度。CPU 与 GPU 异步独立执行, CUDA 提供了一系列函数用于主机与设备端的同步, 以保证算法有序进行^[17]。

4.1 改进算法的 GPU 并行实现

对于矩阵求逆运算, 并没有现成的核函数可以进行直接调用, 因此必须自己编写 kernel 函数进行求逆。文献[18]介绍了 GPU 利用 Gauss-Jordan 顺序消去法进行矩阵求逆的详细流程, 实际上可以采用 Gauss-Jordan 原地求逆算法, 这样可以减少一半的显存空间和计算量。图 4 为矩阵原地求逆实现示意图, 但由于矩阵求逆占用时间并不多, 在此不再赘述。类似地, 步骤(4)只涉及到 $K \times K$ 和 $K \times M$ 矩阵的相乘, 由于 K 与 M 相对较小, 计算耗时较短, 可直接利用线性代数库 CUBLAS^[19]中的 cublasCgemm 函数。同理, 步骤(6)计算量很小, 耗时短, 直接利用 CUBLAS 库中 cublasCaxpy 函数计算。

计算步骤(3), 步骤(5)矩阵相乘的运算涉及到 X_{ref} , 而其可以完全用 s_{ref} 构成, 若用 CUBLAS 中 cublasCgemm 函数进行计算, 将存在极大的空间冗

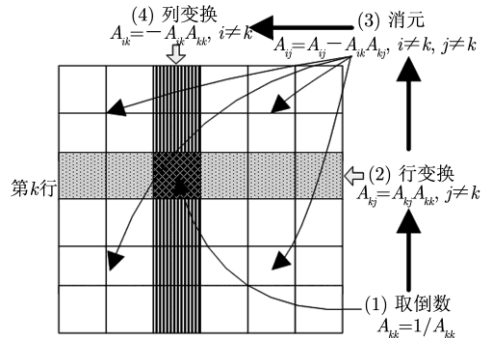


图 4 Gauss-Jordan 原地求逆算法框图

Fig. 4 Block diagram of Gauss-Jordan in-place matrix inverse algorithm

余。所以需要自己编写 kernel 函数直接利用 s_{ref} 计算, 此时不仅可以降低显存占用, 而且可以减少 GPU 访问全局显存的次数, 进一步缩短耗时。由于 $N \gg K$, 通过该方法约降低空间复杂度到原来的 $1/K$ 。同样的在计算步骤 1 中计算 R_x 第 1 行或第 1 列过程中, 也需自己设计 kernel 函数实现, 以降低算法对显存需求。

图 5 为改进算法的详细流程图。其中涉及到 X_{ref} 或 X_{ref}^H 的 3 个部分, 分别用 3 个 kernel 实现, 用于计算步骤(1)中 $X_{ref}^H(1,:)$ X_{ref} , 步骤(3)中 $X_{ref}^H S_{sur}$ 和步骤(5)中 $X_{ref} D$, 其中计算时不需要先利用 s_{ref} 形成 X_{ref} , 而是直接利用 s_{ref} 进行计算, 这样大大减少了从全局显存中加载数据量的大小与次数, 且由于无需形成 X_{ref} , 显存占用大大降低。这 3 个 kernel 实现流程类似, 以计算 $X_{ref}^H S_{sur}$ 为例, GPU 核函数详细描述如图 6 所示。计算 R_x 矩阵的后 $K-1$ 列, 通过调用 1 个 kernel, 在内部循环 $K-1$ 次实现。矩阵求逆由于行变换, 消元和列变换存在依赖性, 每次变换使用 3 个 kernel 实现, 迭代运行 K 次。

4.2 算法优化

算法的优化是一个十分重要的问题, 特别是对实时性要求较高的情况下, 对程序的优化是必不可少的, CPU 端应用程序如此, GPU 端 kernel 函数更是不容忽略。同 CPU 端程序优化不同, GPU 端

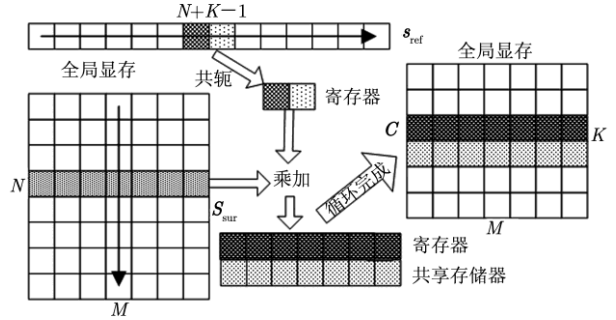


图 6 步骤(3) GPU 端 kernel 函数实现框图

Fig. 6 Block diagram of kernel function on GPU of step (3)

程序优化主要考虑能够利用 GPU 的各种资源, 如具有高带宽的片内寄存器和共享存储器, 能够对全局数据加速访问的纹理缓存, 常量缓存和 L1 和 L2 缓存等, 优化存储器访问和指令选取, 从而获得最大化的利用率、峰值带宽和最大化的指令吞吐率。此外对包含多个流处理器(Stream Multiprocessor, SM)的 GPU 要考虑各个 SM 资源的合理配置, 尽量使其负载均衡^[17]。根据 Amdahl 定律^[20], 主要对其中耗时较大的步骤 1, 3, 5 进行优化, 尽量使用寄存器来计算, 首先把数据从全局存储器加载至共享存储器, 然后从共享存储器依次读入寄存器, 在寄存器中进行计算以保证速度, 同时每个 Thread 计算多个结果, 实现数据重用, 减少了对全局存储器的访问, 并且利用纹理缓存和数据预取技术^[21]加速对全局存储器的访问, 最终得到较为理想的结果。

5 仿真和实测处理结果

信号处理的软硬件配置如表 2 所示。仿真数据为采样率为 312.5 kHz 的 8 路 FM 信号, 1 路为参考通道, 其余 7 路为监测通道。仿真时各个监测通道中多径和目标参数设置相同, 详细参数如表 3 所示。

其中数据点数 $N=256 \times 1024$ (以下简称 256 k), 自适应滤波阶数 $K=128$, 同时抑制通道数 $M=7$ 。

表 2 软硬件配置情况

Tab. 2 Hardware and software configuration

硬件配置	Intel Core i7 950 @ 3.07 GHz, 4 GB @DDR3 1333 MHz 内存, 1280 MB 显存, GeForce GTX 570 显卡, PCI-E 2.0 X16 接口
软件配置	Windows XP Professional SP3, GPU 驱动版本号 332.88 Visual Studio 2010 SP1 旗舰版, CUDA 6.0

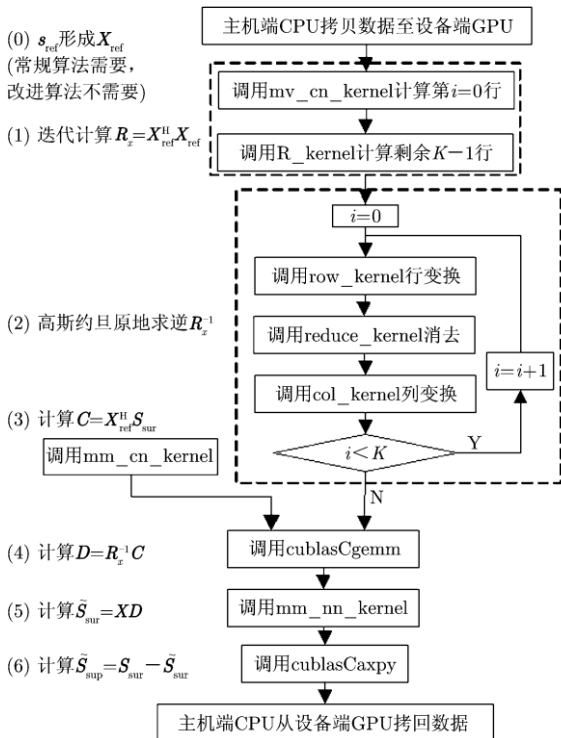


图 5 多通道改进杂波抑制算法 GPU 实现流程图

Fig. 5 Flow chart of multi-channel improved clutter suppression algorithm on GPU

表 3 仿真信号详细参数

Tab. 3 Detailed parameters of simulated signal

	直达波	多径 1	多径 2	多径 3	多径 4	目标 1	目标 2
信噪比 SNR(dB)	60	50	40	30	20	-20	-30
时延序号	1	14	23	37	59	105	179
多普勒频率(Hz)	0	0	0	0	0	25	-50

分别利用 CPU 和 GPU 实现多通道 ECA-B 算法并对仿真数据进行处理，通过相干匹配处理得到距离多普勒谱。图 7 为仿真信号杂波抑制前通道 1 距离多普勒谱，由于直达波和多径杂波的影响，目标被遮蔽不可见。图 8 为杂波抑制后通道 1 距离多普勒谱，直达波和多径杂波被抑制，目标凸显可见。图 9 给出了 CPU 与 GPU 程序计算绝对误差。表 4 和表 5 给出了算法各部分详细耗时和加速比，其中 CPU 端和 GPU 端程序运行时间均在 release 模式下

测得。CPU 端编译器设置为“使速度最大化(/O2)”，利用高精度测时函数 QueryPerformanceFrequency 和 QueryPerformanceCounter 多次运行取平均求得。GPU 端则使用 NVIDIA 官方提供的 NVIDIA Visual Profiler^[17]直接测出结果，再多次取平均，具有更高的准确性。从结果来看，多通道 ECA-B 算法同时对 7 个通道数据(每个通道点数为 256 k 点，抑制距离元数为 128)进行杂波抑制耗时仅 17.215 ms。

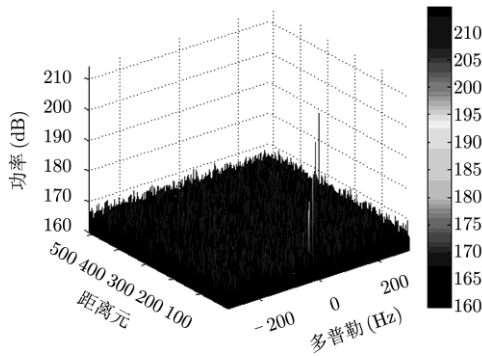


图 7 杂波抑制前通道 1 距离多普勒谱

Fig. 7 Range-Doppler map of channel 1 before clutter suppression

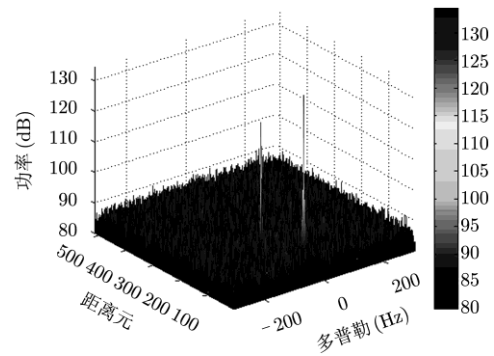


图 8 杂波抑制后通道 1 距离多普勒谱

Fig. 8 Range-Doppler map of channel 1 after clutter suppression

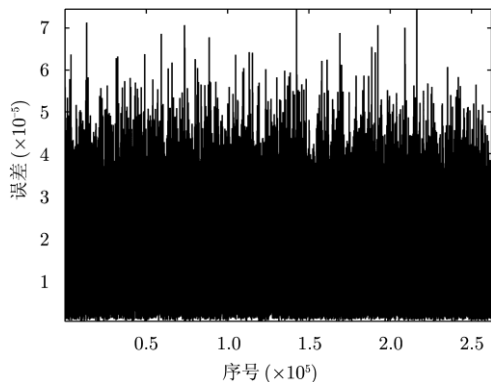


图 9 CPU 与 GPU 计算绝对误差

Fig. 9 Absolute computing error between CPU and GPU

上述改进算法的总耗时只是纯计算时间，若加上数据从 CPU 到 GPU 的传输耗时(5.4 ms)，总耗

时为 22.6 ms(GPU 端抑制结果由于要继续进行匹配滤波处理，因此理论上无需拷回 CPU 端)，最终加速比为 152.0X。

2013 年 11 月在武汉大学外辐射源雷达实验基地进行了利用 FM 广播的外场探测实验，图 10 为某段 FM 实测信号杂波抑制前距离多普勒谱，其中信号中心频率为 103.8 MHz，带宽为 500 kHz。采用改进算法对 7 个通道同时进行杂波抑制(每个通道数据点数为 256 k 点，约 0.524 s)，抑制距离元数为 256，采用双精度浮点计算，总耗时仅为 116.5 ms。文献[22]给出了在 GTX480 实现 ECA 杂波抑制的耗时结果，其采用双精度浮点运算，抑制距离元数为 96，对 240 k 点杂波抑制耗时为 21.10 s。对比可知，本文算法：(1)无需构造参考矩阵 \mathbf{X} ；(2)

表4 GPU常规和改进算法各步骤(序号参见图5)耗时(μs)Tab. 4 Time cost for each step (index refer to Fig. 5) of GPU conventional and improved algorithm (μs)

步骤	0	1	2	3	4	5	6
常规	3043.7	58748.1	1207.3	7184.6	16.6	7814.7	317.7
改进	-	1651.1		7022.1		7000.6	

表5 算法总耗时与加速比

Tab. 5 Total time cost and speedup of different algorithms on CPU and GPU

	CPU		GPU	
	常规	改进	常规	改进
总耗时(ms)	134646.5	3436.2	78.3	17.2
加速比(GPU/CPU)	-	1.0X	43.9X	199.6X

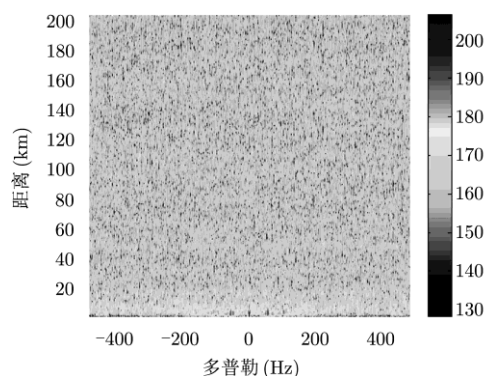


图10 实测FM信号通道1距离多普勒谱(抑制前)

Fig. 10 Range-Doppler map of channel 1 of real FM signal before clutter suppression

不需要对矩阵进行共轭转置;(3)由于仅对直达波和多径杂波进行抑制,不需要对参考矩阵进行多普勒频移;(4)利用迭代算法大大降低自相关矩阵的计算时间;(5)最大化利用GPU资源,合理安排kernel函数,并高度优化算法,故性能提升显著。图11为该段数据杂波抑制后通道1距离多普勒谱,抑制后多个目标凸显,证实方法的有效性。

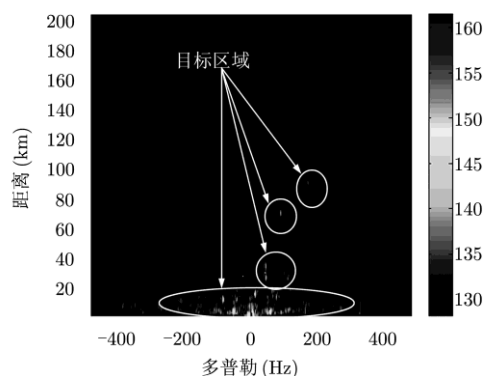


图11 实测FM信号通道1距离多普勒谱(抑制后)

Fig. 11 Range-Doppler map of channel 1 of real FM signal after clutter suppression

6 总结

本文首先对ECA-B算法通过向量列扩展,使其能够对多通道同时处理,然后针对多通道杂波抑制算法耗时最大的计算自相关矩阵 R 步骤中的时间冗余和空间冗余,提出迭代计算的新算法,将其时间和空间复杂度均降到常规算法的 $1/K$ (K 为杂波自由度),并结合CUDA在GPU上予以实现。仿真和实测数据处理表明,相对于常规算法,该改进算法能够极大减小显存需求,缩短计算时间,达到外辐射源雷达杂波抑制的实时处理要求。本文算法已运用于外辐射源雷达实时处理系统中,并取得较好的效果。

参考文献

- [1] 万显荣. 基于低频段数字广播电视信号的外辐射源雷达发展现状与趋势[J]. 雷达学报, 2012, 1(2): 109-123.
Wan Xian-rong. An overview on development of passive radar based on the low frequency band digital broadcasting and TV

signals[J]. *Journal of Radars*, 2012, 1(2): 109-123.

- [2] Kuschel H and O'Hagan D. Passive radar from history to future[C]. IEEE 11th International Radar Symposium (IRS), Vilnius, Lithuania, 2010: 1-4.
- [3] Howland P E, Griffiths H D, and Baker C J. Passive Bistatic Radar (Chapter in Bistatic Radar)[M]. Germany: Emerging Technology Wiley, 2008: 247-311.
- [4] Howland P E, Maksimiuk D, and Reitsma G. FM radio based bistatic radar[J]. *IEE Proceedings-Radar, Sonar and Navigation*, 2005, 152(3): 107-115.
- [5] 吴海洲, 陶然, 单涛. 基于DTTB照射源的无缘雷达直达波干扰抑制[J]. 电子与信息学报, 2009, 31(9): 2033-2038.
Wu Hai-zhou, Tao Ran, and Shan Tao. Direct-path interference suppression for passive radar based on DTTB illuminator[J]. *Journal of Electronics & Information Technology*, 2009, 31(9): 2033-2038.
- [6] 张各各, 王俊, 刘玉春. 一种基于数字电视地面照射源的外辐射源雷达快速杂波相消算法[J]. 电子与信息学报, 2013, 35(1): 36-40.

- Zhang Ge-ge, Wang Jun, and Liu Yu-chun. A fast cancellation algorithm for DTTB-based passive radar system[J]. *Journal of Electronics & Information Technology*, 2013, 35(1): 36-40.
- [7] Li Ji-chuan, Zhao Yao-dong, and Lü Xiao-de. The impact of step selection in NLMS algorithm on low velocity target detecting for passive radar[C]. IET International Radar Conference, Xi'an, 2013: 1-4.
- [8] Palmer J E and Searle S J. Evaluation of adaptive filter algorithms for clutter cancellation in passive bistatic radar[C]. 2012 IEEE Radar Conference (RADAR), Atlanta, USA, 2012: 0493-0498.
- [9] Rosa L, Re L, and Lallo D. Passive coherent locator signal processor on IBM Cell Broadband Engine (Cell BE)[C]. 2009 IEEE Radar Conference, Pasadena, CA, USA, 2009: 1-6.
- [10] Bernaschi M, Di Lallo A, Fulcoli R, *et al.* Combined use of Graphics Processing Unit (GPU) and Central Processing Unit (CPU) for passive radar signal & data elaboration[C]. 2011 IEEE International Radar Symposium (IRS), Hamburg, Germany, 2011: 315-320.
- [11] Malanowski M. Comparison of adaptive methods for clutter removal in PCL radar[C]. 2006 IEEE International Radar Symposium, Krakow, Poland, 2006: 237-240.
- [12] Szumski K, Malanowski M, Kulpa J, *et al.* Real-time software implementation of passive radar[C]. 2009 European Radar Conference, Rome, Italy, 2009: 33-36.
- [13] Zhao Y D, Zhao Y K, Lu X D, *et al.* Block NLMS cancellation algorithm and its real-time implementation for passive radar[C]. 2013 IET International Radar Conference, Xi'an, 2013: 1-5.
- [14] Cardinali R, Colone F, Ferretti C, *et al.* Comparison of clutter and multipath cancellation techniques for passive radar[C]. 2007 IEEE Radar Conference, Waltham, MA, USA, 2007: 469-474.
- [15] Colone F, Cardinali R, and Lombardo P. Cancellation of clutter and multipath in passive radar using a sequential approach[C]. 2006 IEEE Radar Conference, Verona, NY, USA, 2006: 393-399.
- [16] Colone F, O'hagan D W, Lombardo P, *et al.* A multistage processing algorithm for disturbance removal and target detection in passive bistatic radar[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2009, 45(2): 698-722.
- [17] NVIDIA. NVIDIA CUDA C Programming Guide, Version 6.0, 2014. 2.
- [18] 刘丽, 沈杰, 李洪林. 基于 GPU 的矩阵求逆性能测试和分析[J]. 华东理工大学学报: 自然科学版, 2010, 36(6): 812-817.
- Liu Li, Shen Jie, and Li Hong-lin. Performance testing and analysis for matrix in version base on GPU[J]. *Journal of East China University of Science and Technology: Natural Science Edition*, 2010, 36(6): 812-817.
- [19] NVIDIA. CUBLAS Library, Version 6.0, 2014. 2.
- [20] Amdahl G. Validity of the single processor approach to achieving large-scale computing capabilities[C]. AFIPS Conference, 1967, 30: 483-485.
- [21] Lai Jun-jie and Sez nec A. Performance upper bound analysis and optimization of SGEMM on Fermi and Kepler GPUs[C]. 2013 IEEE/ACM International Symposium on Code Generation and Optimization (CGO), 2013, DOI: CGO. 2013. 6494986.
- [22] Mathew J, Inggs M, and Petri D. Real time processing of networked passive coherent location radar system[J]. *International Journal of Electronics and Telecommunications*, 2011, 57(3): 363-368.

作者简介



陈伟(1989-),男,硕士研究生,研究方向为雷达信号处理、实时信号 GPU 并行处理等。



万显荣(1975-),男,博士,教授,博士生导师,研究方向为外辐射源雷达系统、高频雷达系统及雷达信号处理等。

E-mail: xrwan@whu.edu.cn