文章编号:1671-4598(2018)02-0158-03 DOI:10.16526/j.cnki.11—4762/tp.2018.02.039 中图分类号:TN957.52 文献标识码:A

基于噪声可见性函数的SAR图像增强快速算法

朱逸飞,杨 国,王 强,吴 文
(南京理工大学 近程高速目标探测技术国防重点学科实验室,南京 210094)

摘要:针对SAR成像中图像模糊并伴有噪声的问题,结合噪声可见性函数,提出了一种SAR图像增强快速算法;该算法在图像分层的基础上,结合人眼视觉特性,引入噪声可见性函数,实现细节层图像的增益控制;根据GPU架构和存储结构特点,并行计算各个像素在基本层和细节层上的处理过程,完成该算法的并行优化设计与实现;实验结果表明,该算法能够有效提高图像质量,增强图像细节;同时,能够充分利用GPU的并行计算能力,有效提高SAR图像增强的实时性。

关键词:并行计算;噪声可见性函数;图像分层

Fast Algorithm for SAR Image Enhancement Based on NVF

Zhu Yifei, Yang Guo, Wang Qiang, Wu Wen
(Ministerial Key Laboratory of JGMT, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: According to the problem of image blurring with noise in SAR imaging, an adaptive enhancement algorithm for SAR image which combines with the noise visibility function is proposed. This algorithm realizes the adaptive processing of the detail layer, combined with the visual characteristics of human eyes and the noise visibility function, based on the image layering. The processing of each pixel in the base layer and the detail layer is calculated in parallel according to the characteristics of GPU architecture and memory, and the parallel optimization design of the algorithm is completed. The simulation results show that this algorithm can improve the image quality effectively, and at the same time, enhance the real-time performance of SAR image enhancement by making full use of GPU parallel computing ability.

Keywords: parallel computing; noise visibility function; image layering

0 引言

合成孔径雷达SAR(synthetic aperture radar)是一种高分辨率成像雷达,具有全天时、全天候、穿透性强等特点,广泛应用于军事和民用领域。但由于战场环境复杂、大气衰减、硬件设备干扰等原因,SAR成像结果会存在整体灰度分布低且集中、图像对比度低、含有噪声等问题,造成目标不清晰、图像模糊,影响图像识别效果,因此需要对SAR图像进行增强[1]。目前国内外利用数据处理的方法提高SAR图像质量主要分为两类[2]:一是在成像阶段利用系统回波数据,通过超分辨成像算法获得高分辨率的SAR图像,即SAR的超分辨成像方法;二是在成像后的图像数据基础上进行图像质量提高的处理。其中后者是本文的研究内容,典型算法如Kuan算法、Lee算法、双边滤波算法、偏微分算法等,这些算法能够滤除部分噪声并保留细节边缘和目标特征[3]。但是,典型的SAR图像增强算法复杂度较高,面对高分辨率图像时,需要的计算时间较长,在目标识别、跟踪及灾难评估等对实时性要求较高的领域,无法满足系统要求。

与中央处理器CPU(central processing unit)不同,图形处理器GPU(graphic processing unit)专为密集型、高度并行化的计算而设计,其中用于数据处理的晶体管数量远远大于缓存和逻辑控制部分,这使得GPU更适合处理无逻辑关系数据的并行运算[4]。2007年,NVIDIA公司推出通用并行计算架构CUDA(compute unified device architecture),目的是将GPU作为并行计算设备,进行通用并行计算[5]。CUDA为开发者提供了硬件的直接访问接口,可以方便地写出在GPU上执行的程序,而不用像过去的GPGPU架构,如OpenGL等,将计算映射到图形API中,大大降低了开发门槛和难度。自推出后,CUDA被广泛应用于计算机可视化、音视频编解码、流体力学模拟等领域,都能获得较高的加速比,有着良好的应用前景[6]。

本文提出一种基于噪声可见性函数的SAR图像增强快速算法及其基于GPU的并行化实现方法。该算法运用图像分层理论,分离SAR图像细节与噪声,在对图像细节层进行处理时,结合人眼视觉特性,引入噪声可见性函数,控制细节层增益系数,实现图像增强。在提高图像质量、增强图像细节的同时,充分运用GPU的并行计算特点,具有较高的实时性。

1 基于图像分层的增强算法

为了抑制图像噪声、增强图像细节、提高图像识别效果,Branchitta F等人提出了图像分层理论[7]。其基本思想为:首先,选择一种分层滤波器对原始输入图像进行处理,抹平图像细节和微小波动,得到图像基本层和细节层;其次,分别对图像基本层和细节层进行处理,因为基本层中不包含图像细节信息,所以处理过程中不必考虑细节损失问题,重点在于图像对比度的调整;由于细节层不仅包含原始图像的细节信息,还有噪声的存在,因此需要对细节层进行噪声抑制和细节增强;最后,将处理后的基本层和细节层融合并调整显示范围,得到输出图像。

研究基于图像分层的增强算法,其重点在于分层滤波器、基本层图像处理方法和细节层图像增强方法的选择。本文采用

收稿日期:2017-11-27; **修回日期**:2017-12-11。
基金项目:国防重点项目。
作者简介:朱逸飞(1993-),男,江苏扬州人,硕士研究生,主要从事信号处理、并行计算方向的研究。

双边滤波作为分层滤波器，对 SAR 图像进行双边滤波，得到抹平图像细节和微小波动后的基本层，并与原图像相减，得到包含细节信息和大量噪声的细节层；对基本层采用 $\gamma<1$ 的伽马变换；通过噪声可见性函数控制细节层增益系数。算法框架如图 1 所示。

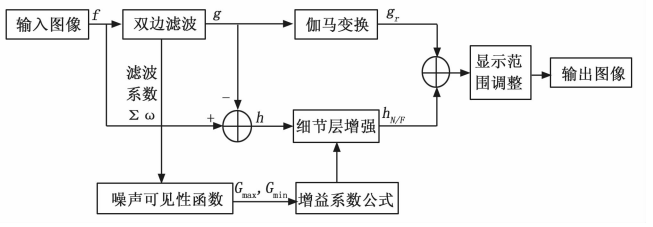


图 1　基于图像分层的增强算法框架

双边滤波（Bilateral filter）结合空域信息和灰度相似性，对图像进行基于空间分布的高斯滤波，是一种非线性的滤波方法。邻域内离中心像素较远的像素占有的权重较小，不会对中心像素的值造成太大影响，这样有利于图像细节边缘像素值的保存，达到保边去噪的目的。

设 $f(x,y)$ 为原图像，$g(x,y)$ 为原图像经双边滤波后的结果，即基本层，其表达式为：

$$g(x,y)=\frac{\sum_{(i,j)\in S_{x,y}}\omega(i,j)f(i,j)}{\sum_{(i,j)\in S_{x,y}}\omega(i,j)} \tag{1}$$

式中，$S_{x,y}$ 表示以像素 (x,y) 为中心，半宽为 N 的邻域，$\omega(i,j)$ 为两部分系数的乘积：

$$\omega(i,j)=\omega_s(i,j)\omega_r(i,j) \tag{2}$$

式中，$\omega_s(i,j)$ 为空域滤波器系数，$\omega_r(i,j)$ 为值域滤波器系数：

$$\omega_s(i,j)=e^{-\frac{|i-x|^2+|j-y|^2}{2\sigma_s^2}} \tag{3}$$

$$\omega_r(i,j)=e^{-\frac{|f(i,j)-f(x,y)|^2}{2\sigma_r^2}} \tag{4}$$

对基本层 $g(x,y)$ 采用 $\gamma<1$ 的伽马变换，拉伸图像灰度范围，提高对比度。伽马变换，也叫做幂律变换，是一种非线性的灰度拉伸变换。设变换后的基本层为 $g_\gamma(x,y)$，变换形式为：

$$g_\gamma(x,y)=C*g(x,y)^\gamma \tag{5}$$

式中，γ 为变换系数，C 为控制像素变化范围的常数。

2　基于 NVF 的细节层增强

在研究水印估计问题时，S. Voloshnovskiy 等人提出了噪声可见性函数 NVF（noise visibility function）[8-9]：

$$NVF(x,y)=\frac{\eta(x,y)}{\eta(x,y)+\sigma_N^2(x,y)} \tag{6}$$

式中，$\eta(x,y)$ 为像素 (x,y) 的权函数，$\sigma_K^2(x,y)$ 为图像的局部方差。

归一化处理后，式（6）变为：

$$NVF(x,y)=\frac{1}{1+\theta\cdot m(x,y)} \tag{7}$$

式中，θ 为常数，$m(x,y)$ 为噪声掩膜函数，与图像的局部方差有关，可用于衡量图像细节。

由式（7）可以看出，$NVF(x,y)$ 可以表示图像中各像素点对噪声的敏感程度，与图像局部能量成反比。越是平坦的区域，$m(x,y)$ 越小，$NVF(x,y)$ 越接近 1，图像在该像素处较为敏感，噪声对人眼刺激较大，视觉效果降低；越是变化剧烈的区域，$m(x,y)$ 越大，$NVF(x,y)$ 越接近 0，图像在该像素处允许较大的噪声，其对图像视觉效果的影响不明显，噪声表现出低可见性，不易被察觉。

将原图像 $f(x,y)$ 与基本层 $g(x,y)$ 相减得到细节层图像 $h(x,y)$。$h(x,y)$ 中包含了原图像中的细节信息，必须进行保留或增强，但细节层中又不可避免地存在原图像中的大部分噪声。因此，在对细节层进行处理时，不能直接简单地放大，这会造成平坦区域噪声的过度放大。

增强细节层图像时，通过 NVF 区分图像的平坦与变化剧烈的区域，对不同的区域采用不同的增益系数。设细节层像素点 (x,y) 的增益系数为 $G(x,y)$，对于平坦区域的像素点，NVF 趋近于 1，$G(x,y)$ 需要取较小值，避免噪声被增强；对于变化剧烈的区域，NVF 趋近于 0，$G(x,y)$ 可以取较大值，在增强细节时不用考虑噪声的问题。设 $G_{\max}$ 为 $G(x,y)$ 的最大值，$G_{\min}$ 为 $G(x,y)$ 的最小值，构造增益系数公式如下：

$$G(x,y)=G_{\min}+(1-NVF(x,y))(G_{\max}-G_{\min}) \tag{8}$$

在双边滤波中，其滤波系数 $\sum_{(i,j)\in S_{x,y}}\omega(i,j)$ 可以很好地表征图像细节，图像局部越平坦，$\sum_{(i,j)\in S_{x,y}}\omega(i,j)$ 越大；反之，$\sum_{(i,j)\in S_{x,y}}\omega(i,j)$ 越小。设计归一化系数 $k(x,y)$：

$$k(x,y)=\frac{\sum_{(i,j)\in S_{x,y}}\omega(i,j)}{\sum_{(i,j)\in S_{x,y}}\omega_s(i,j)} \tag{9}$$

且有：

$$m(x,y)=k(x,y)^{-1}-1 \tag{10}$$

因为 $k(x,y)\in[0,1]$，无需调整范围，所以取 $\theta=1$。由式（7）、式（8）和式（10）可得：

$$G(x,y)=G_{\min}+(1-k(x,y))(G_{\max}-G_{\min}) \tag{11}$$

增强后的细节层 $h_{NVF}(x,y)$ 可表示为：

$$h_{NVF}(x,y)=h(x,y)\cdot G(x,y) \tag{12}$$

3　基于 CUDA 的图像增强算法实现

3.1　CUDA 编程模型

GPU 拥有强大的并行计算能力，但早期的 GPU 编程存在很多困难，如图形 API、交互接口复杂等，这些都限制了 GPU 在通用计算上的发展[10]。CUDA 的出现解决了这一问题，其包含一整套软硬件体系，支持通用并行计算。由于 CUDA 采用类 C 语言进行开发，具有高性能计算指令开发能力，开发者能够在 GPU 强大的并行计算能力基础上建立更高效率的密集数据计算方案。

一个完整的 CUDA 程序包含在 CPU（主机端，host）中执行的部分和在 GPU（设备端，device）中执行的部分。如图 2 所示，主机端负责数据的输入输出、内存分配、主机与设备间的数据传输；而在设备端中运行的核函数（kernel）承担着并行计算的任务[11]。由于 CPU 对显存中的数据进行存取时只能通过 PCI Express 接口，其理论带宽为双向各 4 GB/s，速度较慢，若频繁地进行存取显存的操作会降低 GPU 计算效率[12]。

在并行线程总数确定的情况下，为了有效利用 GPU 的计

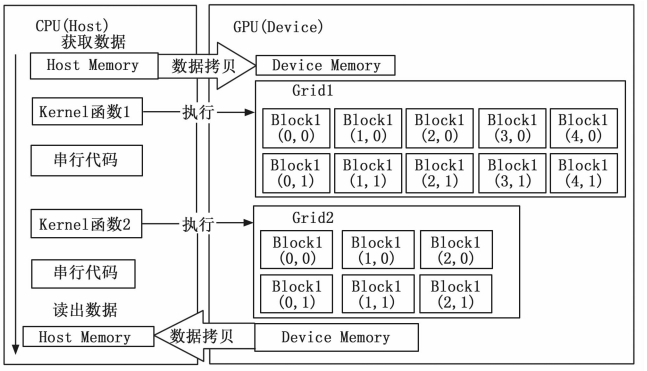


图 2 CUDA 编程模型

算资源，需要对核函数进行合理的线程配置，根据 GPU 规格和需要计算的数据规模，合理地设定线程块 Block 的数量和每个线程块中线程 Thread 的数量。每个线程块中线程数量应为该 GPU 的 warp 大小的整数倍[13]。warp 为最小单指令多线程执行单元，以 32 个线程为一组，并行创建、调度和执行[14]，考虑存储器大小的限制，本文设定每个线程块中线程数量为 256，并根据不同的图像大小划分线程块数量。

3.2 并行化图像增强算法实现

上文中图像增强算法复杂度较高，对于高分辨率图像而言，串行实现方法实时性较低。该算法中像素点间的运算相对独立，数据依赖程度较低，具有较高的内在并行性，适合 GPU 并行实现。在并行化双边滤波算法的基础上[15]进行线程内的分层处理、基本层处理和基于 NVF 的细节层增强，并行完成各像素的全部计算过程，实现算法并行化设计。

具体流程如下：

1）获取图像数据，初始化变量，申请 GPU 显存空间：cudaMalloc（（void * *）&Rdata，（width * height * sizeof (double)））；

2）完成 CPU 与 GPU 间的数据通信；

3）定义线程配置：

Dim3 grid（blockx，blocky，1）；

Dim3 block（threadx，thready，1）；

4）启动 kernel 函数，进行并行化双边滤波和分层，根据式（11）计算细节层增益系数，完成细节层和基本层处理以及图像融合；

5）实现图像输出，释放内存和显存空间。

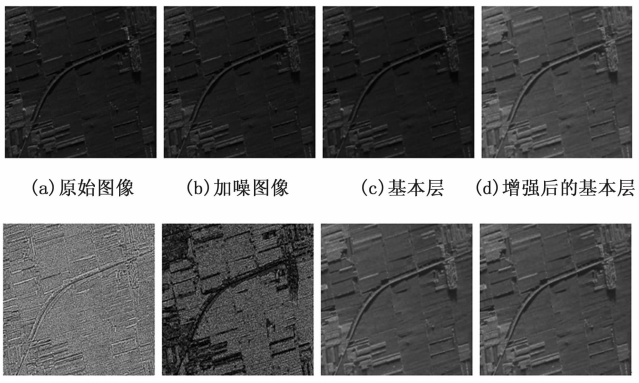
4 实验结果与分析

本文将从算法效果和加速性能两方面进行实验。硬件平台：CPU 为 Intel Core i5 6400，4 核，主频 2.70 GHz；GPU 为 NVIDIA GeForce GTX 960，1024 核，运行频率 1.22 GHz；编译环境为 Visual Studio 2012 和 CUDA 7.5。

4.1 算法效果

本次实验中，考虑视觉效果和计算复杂度，取双边滤波参数 $\sigma_s = 40$、$\sigma_r = 20$，邻域半宽 $N = 5$，基本层伽马变换系数 $\gamma = 0.5$，细节层增益系数最大值 $G_{\max} = 1.5$，最小值 $G_{\min} = 1$。算法实验效果如图 3 所示。

图 3（a）为 1024×1024 的原始 SAR 图像，对图像添加噪声以模拟 SAR 成像结果，结果如图 3（b）所示。图 3（c）为经双边滤波分层后得到的未处理的基本层，图像的细节和微小



(a)原始图像 (b)加噪图像 (c)基本层 (d)增强后的基本层

(e)细节层 (f)增强后的细节层 (g)串行结果 (h)并行结果

图 3 算法效果

波动被抹平；图 3（d）为经过伽马变换后的基本层，图像对比度得到增强；图 3（e）为未处理的细节层，图像细节和噪声同时存在；图 3（f）为经过增强后的细节层，可以看出，在细节增强的同时，部分噪声得到了抑制；图 3（g）和图 3（h）分别为本文所提算法的串行实现计算结果和并行化实现方法的计算结果，可以看到，图像对比度提高的同时，噪声得到了抑制，农田部分的纹理变得更加清晰，房屋、河流等轮廓也变得十分明显。

本文通过峰值信噪比 PSNR 来客观评价图像质量[16]，如式（13）所示。其中 m、n 为图像维度，$f_0(i,j)$ 表示原图像，$f_1(i,j)$ 表示待评价图像。$PSNR$ 值越大，说明失真越少，图像质量越高。结果如表 1 所示。

$$PSNR = 10\log_{10}\frac{255^2 \times m \times n}{\sum_{i=1}^{m}\sum_{j=1}^{n}[f_1(i,j) - f_0(i,j)]^2} \quad (13)$$

表 1 图像 PSNR 值比较

	带噪声图像	双边滤波结果	串行实现结果	并行实现结果
$PSNR$/dB	21.528	26.874	29.223	29.223

从表 1 可以看出，本文所提算法能够有效增强图像细节，提高目标识别能力，有较好的增强效果，并且对图像质量的改善能力优于单独的双边滤波；串行实现和并行实现不会影响算法处理效果，两者得到的图像质量是一致的。

4.2 加速性能

分别对不同尺寸的 SAR 图像进行加速性能测试，记录算法串行实现和并行实现的计算时间。为了减小误差，提高准确程度，取 10 次运算结果的平均值，计算加速比。实验结果如表 2 所示。

表 2 计算时间对比

图像尺寸	串行实现/ms	并行实现/ms	加速比
384×384	2138	63	33.937
512×512	3801	81	46.926
640×640	5939	115	51.643
768×768	8552	151	56.636
896×896	11640	175	66.514
1024×1024	15227	201	75.756

（下转第 164 页）