

Journal of Image
and Graphics

中国图象图形学报



ISSN1006-8961
CN11-3758/TB

2012 **2**
Vol.17 No.

中国科学院遥感应用研究所
中国图象图形学学会主办
北京应用物理与计算数学研究所

中国图象图形学报

Zhongguo Tuxiang Tuxing Xuebao

2012年2月 第17卷 第2期(总第190期)

目次

综述

内容感知图像缩放技术综述 施美玲,徐丹(157)

图像处理和编码

引入纹理相似性的纺织品图像增强 杨学志,田晓梅,方静,卢洁(169)

保持边缘特征和增强对比度的图像缩放算法 计忠平,方美娥,王毅刚,吴向阳(178)

以图像认证为目的的分形编码数字水印算法 黄晓晴,于盛林(183)

结合运动矢量和像素递归的全局运动估计方法 赵亚湘,刘少强,樊晓平(191)

无缝栅格数据小波金字塔构建 马伯宁,冷志光,汤晓安,匡纲要(197)

图像分析和识别

伪彩色空间完全非结构化道路检测方法 胡晓辉,孙苗强,苏晓许(203)

未知环境下机器人障碍物检测技术 王文格,武凯宾,朱江,段文彬,许重阳(209)

局部 GAC 模型在医学图像分割中的应用 张建伟,方林,陈允杰,詹天明,罗春燕(215)

判别割(Dcut)的图像分割及其快速分割算法 邹小林,陈伟福,冯国灿(222)

改进的遮挡条件下瞳孔检测方法 潘林,魏丽芳,郑炳锜,余轮(229)

木材显微图像的局部水平集分割方法 汪杭军,祁亨年(236)

关键点检测的线要素综合算法 黄志坚,张金芳,徐帆江(241)

GPU 加速的高分辨率 DEM 图像地形特征线提取算法 刘洲俊,胡包钢(249)

改进 GVF 的自动 Snakes 模型 周亚男,程熙,骆剑承,沈占锋,胡晓东(256)

曲波变换用于磨粒图像不变矩的提取 张云强,张培林,任国全,王国德,徐超,李兵(263)

图像理解和计算机视觉

深度图像中基于轮廓曲线和局部区域特征的3维物体识别 吴莉婷,张宇,杨一平,田原(269)

机载 LiDAR 数据的 LZD 航带平差 王丽英,宋伟东(279)

计算机图形学

基于 Z 曲线的瓦片地图服务空间索引 聂云峰,周文生,舒坚,许虎(286)

虚拟现实和增强现实

GPU 加速的台风可视化方法 秦绪佳,张勤锋,陈坚,郑红波,徐晓刚(293)

中国图象图形学报

刊名题字: 宋 健 月刊(1996 年创刊)

第 17 卷 第 2 期

2012 年 2 月 16 日出版

主管单位 中国科学院
主 办 中国科学院遥感应用研究所
中国图象图形学学会
北京应用物理与计算数学研究所

主 编 李小文

编辑出版 《中国图象图形学报》编辑出版委员会

北京 9718 信箱 邮编 100101

电子信箱: jig@irsa.ac.cn

电话: 010-68407995 010-82614429

网 址: www.cjig.cn

印刷装订 北京北林印刷厂

广告经营许可证 京朝工商广字第 0346 号

总 发 行 北京报刊发行局

订 购 全国各地邮局

国外发行 中国国际图书贸易总公司

(中国国际书店)

(北京 399 信箱 邮编 100044)

Superintended by Chinese Academy of Sciences
Sponsored by Institute of Remote Sensing Application,
CAS China Society of Image and Graphics
Institute of Applied Physics and Computational
Mathematics
Chief editor LI Xiaowen
Editor, Publisher Editorial and Publishing Board
of Journal of Image and Graphics
(P. O. Box 9718, Beijing 100101, China)
E-mail: jig@irsa.ac.cn
Distributed by Beijing Bureau for Distribution of Newspapers
and Journals
Domestic All Local Post Offices in China
Foreign China International Book Trading Corporation
(P. O. Box 399, Beijing 100044, China)
Printed by Beijing Beilin Printing House

Journal of Image and Graphics

(Monthly, Started in 1996)

Vol. 17 No. 2 February 2012

Contents

Review

Survey on content-aware image resizing techniques Shi Meiling, Xu Dan (157)

Image Processing and Coding

Enhancement of textile image with texture similarity Yang Xuezhi, Tian Xiaomei, Fang Jing, Lu Jie (169)

Edge-preserving and contrast-enhancement image scaling Ji Zhongping, Fang Mei'e, Wang Yigang, Wu Xiangyang (178)

Watermarking method on fractal coding for image authentication Huang Xiaoqing, Yu Shenglin (183)

Global motion estimation method with motion vectors and pixel recursion Zhao Yaxiang, Liu Shaoqiang, Fan Xiaoping (191)

Seamless wavelet pyramid construction for raster data Ma Boning, Leng Zhiguang, Tang Xiao'an, Kuang Gangyao (197)

Image Analysis and Recognition

Completely unstructured road detection for pseudo-color space Hu Xiaohui, Sun Miaoqiang, Su Xiaoxu (203)

Obstacle detection for robot in unknown environment
..... Wang Wenge, Wu Kaibin, Zhu Jiang, Duan Wenbin, Xu Chongyang (209)

Application of local GAC model for medical image segmentation
..... Zhang Jianwei, Fang Lin, Chen Yunjie, Zhan Tianming, Luo Chunyan (215)

Fast image segmentations of Deut Zou Xiaolin, Chen Weifu, Feng Guocan (222)

Improved method for the pupil measurement under occlusion Pan Lin, Wei Lifang, Zheng Bingkun, Yu Lun (229)

Segmentation method of wood microscopic image based on local level set Wang Hangjun, Qi Hengnian (236)

Algorithm of line generalization with key point detection Huang Zhijian, Zhang Jinfang, Xu Fanjiang (241)

High resolution DEM topographic feature line extraction algorithm using GPU Liu Zhoujun, Hu Baogang (249)

Automatic snakes model based on modified GVF Zhou Yanan, Cheng Xi, Luo Jiancheng, Shen Zhanfeng, Hu Xiaodong (256)

Invariant moment extraction by curvelet transform for wear particle images
..... Zhang Yunqiang, Zhang Peilin, Ren Guoquan, Wang Guode, Xu Chao, Li Bing (263)

Image Understanding and Computer Vision

Three-dimensional free-form object recognition based on contour curve and local surface patches in range images
..... Wu Liting, Zhang Yu, Yang Yiping, Tian Yuan (269)

Least Z-difference strip adjustment of airborne LiDAR data Wang Liying, Song Weidong (279)

Computer Graphics

Spatial index for tile map service based on Z curve Nie Yunfeng, Zhou Wensheng, Shu Jian, Xu Hu (286)

Virtual Reality and Augmented Reality

GPU accelerated typhoon visualization method Qin Xujia, Zhang Qinfeng, Chen Jian, Zheng Hongbo, Xu Xiaogang (293)

中图分类号: TP391 文献标志码: A 文章编号: 1006-8961(2012)02-0249-07

论文引用格式: 刘洲俊, 胡包钢. GPU加速的高分辨率DEM图像地形特征线提取算法[J]. 中国图象图形学报, 2012, 17(2): 249-255

GPU加速的高分辨率DEM图像 地形特征线提取算法

刘洲俊^{1,2}, 胡包钢¹

1. 中国科学院自动化研究所, 北京 100190; 2. 中国科学院研究生院, 北京 100049

摘要: 随着数据采集设备的发展, 数字地形分析中高分辨率数字高程模型(DEM)图像越来越普遍。目前已经存在一系列的曲线结构提取算法由于计算复杂度较高, 因此在针对高分辨率DEM图像提取地形特征线时效率较低。提出一种在图形处理器(GPU)上加速Steger曲线结构提取算法的策略, 利用图形处理器上计算统一设备架构(CUDA)的高度并行性来加速算法中计算密集的Hessian矩阵生成模块以及图像特征点提取模块, 对于百万像素级的DEM图像该算法可以获得5倍以上的加速比。

关键词: 数字高程模型; 地形特征线检测; 图形处理器; 计算统一设备架构

High resolution DEM topographic feature line extraction algorithm using GPU

Liu Zhoujun^{1,2}, Hu Baogang¹

1. Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China;

2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China

Abstract: High resolution digital elevation models (DEM) used in digital terrain analysis are becoming more and more prevalent. There are various curvilinear structure extracting algorithms, but the main limitation of them is the high computing cost, making them less efficient when extracting topographic feature lines from high resolution DEM images. We propose an efficient strategy to speed them up using Steger's curvilinear structure detection algorithm implemented on graphic processing units (GPU). We choose to speed up the most computation intensive modules of the algorithm (Hessian matrix generation and feature point detection) using NVIDIA's compute unified device architecture (CUDA). This method can achieve more than five times speedup compared with the original algorithm on central process units (CPU) for large scale DEM images with millions of pixels.

Key words: digital elevation model; topographic feature line detection; graphic processing unit; compute unified device architecture

0 引言

地理信息系统(GIS)使用的数字高程模型

(DEM)中包含各种地形地貌构造的特征信息, 其中地形特征线的自动提取作为GIS应用功能的一项重要内容, 在地形信息获取和分析中具有重要的应用价值, 而山脊线和山谷线是地形特征线的典型代表,

收稿日期: 2011-05-24; 修回日期: 2011-06-07

基金项目: 中国科学技术部国际合作重点项目(2007DFC10740)

第一作者简介: 刘洲俊(1984—), 女, 中国科学院自动化研究所计算机应用技术专业硕士研究生, 主要研究方向为计算机图形学。

E-mail: zhoujun.liu@ia.ac.cn

被广泛用在高逼真度地形表示、等高线高程的自动推算、等高线成组综合以及地形水文分析、地图制作等方面^[1-3]。目前已经存在一系列的地形特征线提取算法,包括基于地形曲面几何形态分析提取方法、基于地表水流模拟提取方法等。

针对规则格网 DEM 数据,将格网等价于图像像素,DEM 高程值等价于图像灰度值时,借助图像处理或者计算机视觉领域的曲线结构提取算法,可以完成地形特征线的自动提取。目前在该领域有 3 类主流的曲线结构提取算法,第 1 类算法基于图像边缘提取^[4-5],将曲线结构看作一个具有平行边缘的区域,首先用边缘检测算法提出图像的边缘信息,然后从中分析出曲线结构的位置。第 2 类算法是通过提取图像中的脊线来提取曲线结构^[6-7],将灰度图像看成高度场,使用微分几何的思想提取出高度场中的脊线和谷线。第 3 类算法认为曲线结构是一个退化的 2 维区域,使用基于区域分割的思想^[8-9]来提取曲线结构的所有像素。

选取有代表性的由 Steger 提出的脊线检测算法^[6]作为研究基础来展开研究工作,将该算法命名为基于 Hessian 矩阵的脊线检测算法。该算法具有提取精度高、鲁棒性好和算法可调参数少的优点。但该算法属于计算密集型算法,对于越来越普遍出现的高分辨率 DEM 图像进行地形特征线提取时,算法效率较低。经过分析,该算法的主要模块具有天然的可并行性,所以将该算法在图形处理器(GPU)上进行并行加速,采用真实的高分辨率 DEM 灰度图像进行测试,能够有效地提高算法的运行效率。

1 计算统一设备架构

基于图形处理器的通用计算在视频编码或者计算机视觉应用中,针对大数据量表现出实时性方面的优势^[10-13]。Kinsner 等人将离散尺度空间特征提取算法采用 GPU 技术加速^[14]。因此,通过分析原算法的计算瓶颈,并进行算法可并行性分析,进而利用 GPU 强大的计算优势来提高算法的实时性是一条提高算法效率的有效途径。

采用计算统一设备架构(CUDA)作为算法并行化的实现平台。简单介绍 CUDA 的硬件结构和软件运行机制^[15],作为实现算法的基础。

CUDA 编程模型由作为主机端的 CPU 和作为设备端的 GPU 组成(如图 1 所示)。运行在 GPU 上

的 CUDA 代码称之为内核函数(kernel),在一个网格(grid)上运行内核函数时,网格先被分成若干个相互之间不能进行通信的线程块;每个线程块中又含有若干个线程。总之,CUDA 编程模型是通过粗粒度级的线程块和细粒度的线程两级并行来完成大规模数据的计算任务的。

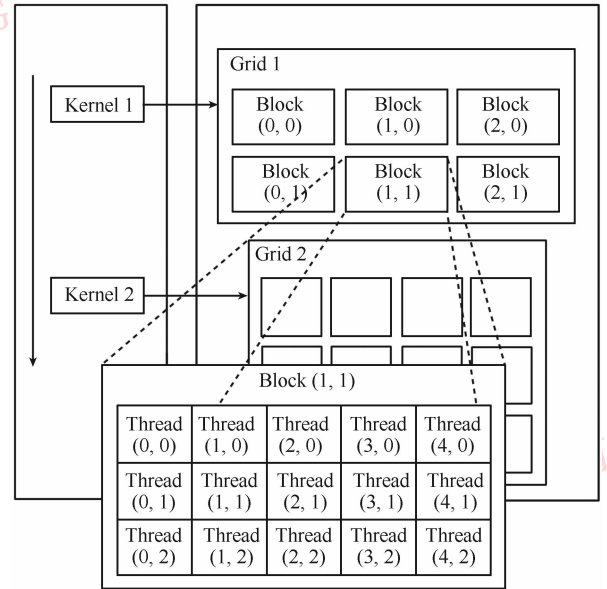


图 1 CUDA 编程模型^[16]

Fig. 1 CUDA program model

CUDA 存储结构由以下几部分组成,分别是: 1)所有线程都可以进行读写访问的全局存储器; 2)供每个线程块进行读写操作的共享存储器,同一个线程块内部的线程之间可以通过共享内存实现相互通信;3)所有线程都可以访问的快速只读存储器:常数存储器和纹理存储器;4)每个线程独享的寄存器和局部存储器。从访问速度来讲,寄存器和共享存储器是 GPU 片上的高速缓存器;常数存储器和纹理存储器位于全局存储器上但具有缓存加速,因而访问速度较快;局部存储器亦位于全局存储器上,访问速度很慢。

2 基于 Hessian 矩阵的脊线检测算法^[6]

2.1 1 维线检测

理想状态下,一条宽为 2ω , 像素对比度为 h 的待检测线的截面 1 维模型为

$$f_b(x) = \begin{cases} h & |x| \leq \omega \\ 0 & |x| > \omega \end{cases} \quad (1)$$

将图像中线特征的截面函数 $f_b(x)$ 与高斯平滑核函数及其微分形式(式(5)–(7))的卷积定义为如下响应函数:

$$r_b(x, \sigma, \omega, h) = g_\sigma(x) * f_b(x) \quad (2)$$

$$r'_b(x, \sigma, \omega, h) = g'_\sigma(x) * f_b(x) \quad (3)$$

$$r''_b(x, \sigma, \omega, h) = g''_\sigma(x) * f_b(x) \quad (4)$$

式中,

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (5)$$

$$g'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma^2}} \quad (6)$$

$$g''_\sigma(x) = \frac{x^2 - \sigma^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{x^2}{2\sigma^2}} \quad (7)$$

从图2^[6]可得,响应函数 r'_b 对于任意 σ 均满足 $r'_b(x, \sigma, \omega, h) = 0 \Leftrightarrow x = 0$ 。显著特征线的检测是由二阶方向导数 $r''_b(x, \sigma, \omega, h)$ 在 $x = 0$ 处的幅值决定的。当尺度空间 $\sigma = \omega / \sqrt{3}$ 时 $r''_b(x, \sigma, \omega, h)$ 在 $x = 0$ 处可以取得负的最大值。

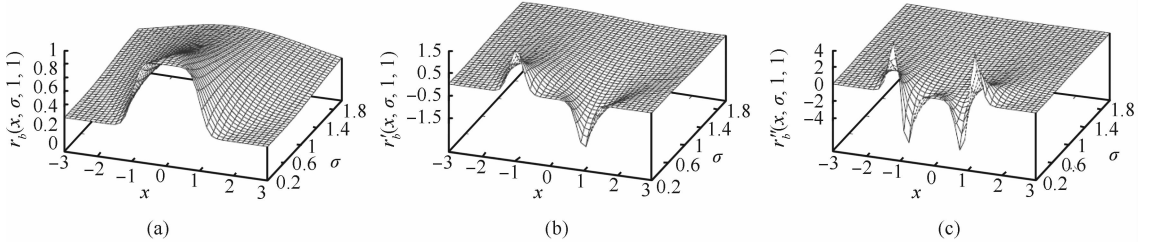


图2 棒状截面 f_b 与高斯核函数微分卷积的尺度空间行为 $x \in [-3, 3]$ 且 $\sigma \in [0.2, 2]$

Fig. 2 Scale-space behavior of the parabolic line f_b when convolved with the derivatives of $x \in [-3, 3]$ and $\sigma \in [0.2, 2]$

2.2 2维线检测

由1维线检测的性质,将图像 I 分别卷积高斯核函数以及其相应的一阶偏导数和二阶偏导数得到

$$I_x = g_{x,\sigma}(x, y) * I = g_\sigma(y) g'_\sigma(x) * I \quad (8)$$

$$I_y = g_{y,\sigma}(x, y) * I = g'_\sigma(y) g_\sigma(x) * I \quad (9)$$

$$I_{xx} = g_{xx,\sigma}(x, y) * I = g_\sigma(y) g''_\sigma(x) * I \quad (10)$$

$$I_{xy} = g_{xy,\sigma}(x, y) * I = g'_\sigma(y) g'_\sigma(x) * I \quad (11)$$

$$I_{yy} = g_{yy,\sigma}(x, y) * I = g''_\sigma(y) g_\sigma(x) * I \quad (12)$$

进而获得图像上每个像素点对应的 Hessian 矩阵

$$H(x, y) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (13)$$

每个像素点 Hessian 矩阵特征值绝对值最大时,特征向量 (n_x, n_y) 表示该点脊线的法线方向。沿着 (n_x, n_y) 方向,图像 I 一阶方向导数为零求得脊线点的相对亚像素位置 (p_x, p_y) 。

$$(p_x, p_y) = (tn_x, tn_y) \quad (14)$$

$$t = -\frac{I_x n_x + I_y n_y}{I_{xx} n_x^2 + 2I_{xy} n_x n_y + I_{yy} n_y^2} \quad (15)$$

定义线的强度,沿着 (n_x, n_y) 方向,图像 I 二阶方向导数的绝对值,表示线的显著程度。

2.3 连线算法

经过2.2节中算法的处理,图像中的每个像素得到3个数据,脊点的相对亚像素位置坐标 $(p_x,$

$p_y)$, 线条法线方向 (n_x, n_y) 和线条强度(沿线条法线方向二阶方向导数的绝对值)。

线的起点为二阶方向导数最大的像素点,即先由最显著的线连起,在给定邻域内,假设 d 表示两点亚像素位置的距离, $d = \|p_2 - p_1\|_2$, β 表示两点处线方向的角度差, $\beta = |\alpha_2 - \alpha_1|$, 当 $d + c\beta$ (通常取 $c = 1$) 取得最小值时,另一个点为可连接点,按此规则递归连线。连线中止条件是邻域内已经无脊线点或者最佳可连接点已属于另一条线。

定义上限阈值 T_H 和下限阈值 T_L 。确定是脊线起点的条件是二阶方向导数的绝对值大于上限阈值 T_H , 确定线上后续点的条件是二阶方向导数的绝对值需要大于下限阈值 T_L , 这样可以降低图像噪声以及去掉一些不显著的线。

3 基于 Hessian 矩阵脊线检测算法 GPU 技术实现

基于 Hessian 矩阵的脊线检测算法,其主要流程包括:1) 高斯卷积,计算图像一阶和二阶偏导数;2) Hessian 矩阵计算,求取特征值和特征向量,确定脊点的方向、位置和强度;3) 脊点连线。高斯卷积计算和 Hessian 矩阵计算这两部分是针对图像中每个像素进行的,像素与像素之间的操作相互独立,因

此具有天然的可并行性,连线部分不适合并行的原因将在后面进行论述。

前述式(9)一(13)高斯卷积求取图像一阶及二阶导数可分解为分别进行一次行卷积和一次列卷积操作,如下所示:

$$I_x = g_{x,\sigma}(x,y) * I \leftrightarrow h_1 = g'_\sigma(x) * I \quad (16)$$

$$I_x = g_\sigma(y) * h_1 \quad (17)$$

$$I_y = g_{y,\sigma}(x,y) * I \leftrightarrow h_2 = g_\sigma(x) * I \quad (18)$$

$$I_y = g'_\sigma(y) * h_2 \quad (19)$$

$$I_{xx} = g_{xx,\sigma}(x,y) * I \leftrightarrow h_3 = g''_\sigma(x) * I \quad (20)$$

$$I_{xx} = g_\sigma(y) * h_3 \quad (21)$$

$$I_{xy} = g_{xy,\sigma}(x,y) * I \leftrightarrow h_4 = g'_\sigma(x) * I \quad (22)$$

$$I_{xy} = g'_\sigma(y) * h_4 \quad (23)$$

$$I_{yy} = g_{yy,\sigma}(x,y) * I \leftrightarrow h_5 = g_\sigma(x) * I \quad (24)$$

$$I_{yy} = g''_\sigma(y) * h_5 \quad (25)$$

由式(16)(18)(20)(22)(24)可得,5次行卷积计算中只有3次有效操作,因为 $h_4 = h_1, h_5 = h_2$ 。因此,首先将高斯行卷积操作由GPU的一个内核函数 $kernel_1$ 来完成,函数的输入是DEM灰度图像 I 和卷积核函数 $g_\sigma, g'_\sigma, g''_\sigma$,输出行卷积的结果 h_1, h_2, h_3 ,并将结果存储在全局显存中。

由式(17)(19)(21)(23)(25)可得,5次列卷积中没有冗余操作,因此,将这5次列卷积操作写入GPU的另一个内核函数 $kernel_2$ 。函数的输入是行卷积的结果和卷积核函数 $g_\sigma, g'_\sigma, g''_\sigma$,输出是DEM灰度图像 I 的一阶偏导数和二阶偏导数 $I_x, I_y, I_{xx}, I_{xy}, I_{yy}$ 。

通过上述两次卷积操作得到式(14)对应的Hessian矩阵,针对每一个像素Hessian矩阵计算亦由GPU的一个内核函数 $kernel_3$ 完成。函数的输入是图像的偏导 $I_x, I_y, I_{xx}, I_{xy}, I_{yy}$,输出的是绝对值最大的特征值 ev ,该像素是否为脊点的标记 $ismax$,脊点的位置坐标 (p_x, p_y) 和方向向量 (n_x, n_y) 。

注意到 $kernel_2$ 函数的输出图像 I 的一阶偏导数和二阶偏导数 $I_x, I_y, I_{xx}, I_{xy}, I_{yy}$ 是 $kernel_3$ 函数的输入,因此内核函数 $kernel_2$ 和 $kernel_3$ 可以合并到一个内核函数中完成。这么做的优点是将减少 $I_x, I_y, I_{xx}, I_{xy}, I_{yy}$ 先写入全局显存,再由全局显存读入到线程的寄存器和局部存储器,减少大规模数据读写,提高了算法的效率。同时需要特别注意的是 $kernel_1$ 函数的输出行卷积计算结果 h_1, h_2, h_3 虽然也是 $kernel_2$ 函数的输入,但是这部分却不能像内核函数 $kernel_2$ 和 $kernel_3$ 那样合并

由一个内核函数完成。这是因为在内核函数 $kernel_2$ 中进行列卷积计算时除了要知道线程中对应像素点处的行卷积结果还同时需要知道在一个卷积半径内,其他像素点行卷积的结果。因为在划分线程时选取的是每个线程完成一个像素点的计算,所以,为了获得卷积半径内其他像素点行卷积的计算结果,将会遇到两种情况:1)卷积半径内,进行其他像素点行卷积计算的线程和中心像素点行卷积计算的线程位于同一个线程块内时,首先计算结果不是同步完成的,另外结果存储在线程的寄存器和局部存储器中是无法读取的;2)进行其他像素点行卷积计算的线程和中心像素点行卷积计算的线程位于不同的线程块内,由于不同线程块间不能进行通信,再加上同时还有第一种情况分析的原因,卷积半径内其他像素点行卷积计算结果数据更是无法获得。因此 $kernel_1$ 函数和 $kernel_2$ 函数将只能由2个GPU内核函数来分别计算。

内核函数 $kernel_2$ 和 $kernel_3$ 合并之后,总输出结果除了原来 $kernel_3$ 的6个输出值 $ismax, ev, p_x, p_y, n_x, n_y$ 之外,由于后续运算中需要输出图像的一阶偏导数 I_x 和 I_y ,因此将需要事先分配 $8 \times image_width \times image_height \times sizeof(float)$ 大小的显存空间,对比,内核函数 $kernel_2$ 和 $kernel_3$ 合并之前,对照前述每一步的输出,通过每次复用前一步已经分配的显存空间,则总共需要分配 $6 \times image_width \times image_height \times sizeof(float)$,由于处理的是高分辨率图像,所以内核函数 $kernel_2$ 和 $kernel_3$ 分开执行将可以处理相比于合并后更大的图像。

综上所述,通过将内核函数 $kernel_2$ 和 $kernel_3$ 合并,可以获得算法更高的执行效率, $kernel_2$ 和 $kernel_3$ 分开执行,则可以处理更大的图像,两者各有长处,可以根据实际需求灵活选用。

由于会有大规模数据的反复读取操作,利用全局显存提供的纹理显存和常量显存实现优化。纹理显存和常量显存虽位于全局显存上,但是提供缓存机制,由此读取速度明显快于全局显存。纹理显存适合大数据量的只读操作,常量显存适合小数据量的只读操作。本算法中,输入图像 I 、行卷积的计算结果 h_1, h_2, h_3 和列卷积的输出结果 $I_x, I_y, I_{xx}, I_{xy}, I_{yy}$,都可以通过存入纹理显存实现优化代码的目的,对于卷积核函数 g_σ 、

g'_σ 、 g''_σ 通过实现在 CPU 上计算完成,写入常量显存完成。同时,纹理显存内部已经加入了计算图像卷积时的边界处理,因此在代码实现上,不用像 CPU 程序专门去实现图像卷积的边界处理部分。

在单个脊点被提取出来后,需要将它们连接成线段。Steger 提出的连线算法首先从脊点中选取二阶导数值大于上限阈值 T_H 的点作为线段生长的种子点。每一条线段从当前二阶导数最大的种子点开始生长,不断的加入邻域内最优的脊点,直到邻域内没有脊点,或者最优的脊点已经加入了其他的线段,该线段连线结束。整个连线算法每次生长一条线段,直到所有脊点都加入线段为止。该算法内部耦合程度较高,数据依赖性强,不适合用 GPU 进行并行加速。具体原因有以下几点:

1) 线段的起始位置取决于先前的连线结果。每条线段的起始点是当前未加入线段的所有种子点中二阶导数最大的点。而未加入线段的种子点的数目是动态变化的,取决于之前加入线段的种子点的情况。如果打破这一依赖性假设,即线段的起始点可以是当前未加入线段的任意种子点,结果中线段的形态和数目均与原始算法不同。

2) 线段的终止条件影响了算法的实现。线条生长的一个提前终止条件是该线段邻域内最优的脊点已经加入了其他的线段。这就表示线段的生长与它周围线段的生长有着密切的关系,线段之间必须进行频繁的通信来确定生长终止条件是否发生。如果取消这个提前终止条件,即线段生长过程只有在线段邻域内没有脊点时终止。这样不仅无法检测到原始算法中的线段接合点,而且会导致线段之间出现大量重复连线,极大的影响连线算法的效率。

3) 算法需要动态内存分配机制。每条线段在生长时不断的加入新的脊点,储存线段数据的内存需要动态的增长。在 CPU 上有合适的内存动态增长和重分配机制。但是目前 CUDA 框架下不支持在 GPU 的显存上进行动态显存分配。这就限制了连线算法在 GPU 上进行并行加速的可能性。每条线段所含的脊点数目是不确定的,如果按最大可能性在显存上预分配,如果有 N 个脊点,每条线

段需预分配 $N \times N$ 个单元,而在我们的测试图像中,检测到的脊点数目 N 一般大于 10 000,很容易出现显存溢出的问题,因此从硬件支持上连线算法也是无法完成并行的。

4 实验结果

采用 Inter Core2 Duo CPU 2.93 Hz Windows XP 操作系统,NVIDIA Geforce GTX275 显卡,使用 CPU 提供的计时函数用于整体运行时间的纪录。在以下实验中,算法尺度空间参数 σ 设置为 3.0,上限阈值 T_H 为 0.5,下限阈值 T_L 为 0.05。部分实验结果如图 3 及图 4 所示。

通过采用不同尺寸 DEM 灰度图像进行脊线检测测试,从图 3 和图 4 的实验结果反映出,采用 GPU 技术加速基于 Hessian 矩阵的脊线检测算法,很好地继承了该算法检测精度高、鲁棒性好和算法可调参数少的优势。

采用 GPU 技术实现脊线检测算法,除了首先要保证运算结果的准确性,最重要的工作是通过第 3 节采取一系列的提高算法效率的实现方法,针对较大尺寸图像,算法效率比原先在 CPU 上运行有显著改善。

表 1 列出了当输入 DEM 灰度图像尺寸逐渐增加时,采用基于 Hessian 矩阵脊线检测算法的 CPU 和 GPU 技术实现的性能比较。

通过 GPU 技术来达到提高算法效率目的,有如下特点,GPU 技术的并行运算能力是随着要处理的数据量的增大来反映出其优势的。从表 1 可以看出,当输入为较小尺寸图像时,相比于 CPU 的算法实现效率,加速比平均为 2 左右。随着输入图像尺寸的增加,采用 GPU 技术后的算法在效率上表现出越来越明显的优势,对于一幅 15 M 像素规模的图像,加速比平均为 5.7,对于在地形分析中普遍存在的这种较大尺寸 DEM 图像,本章提出的采用 GPU 技术实现具有更好的加速效果。受限于实验采用的 GPU 显存大小,在 NVIDIA Geforce GTX275 上最高能运行 20 M 像素规模的图像,并且,在处理较大尺寸图像时,大规模数据在 CPU-GPU 之间拷贝的开销也会逐渐增大,所以最后加速比约为 6,算法效率有了显著提高。

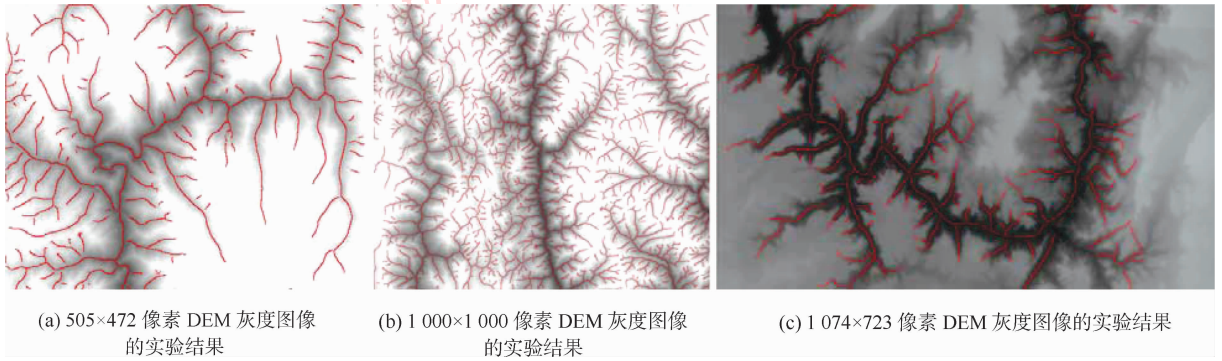


图 3 采用本文算法在不同 DEM 灰度图像的特征线提取结果

Fig. 3 Results on different DEM images using our feature line extraction algorithm

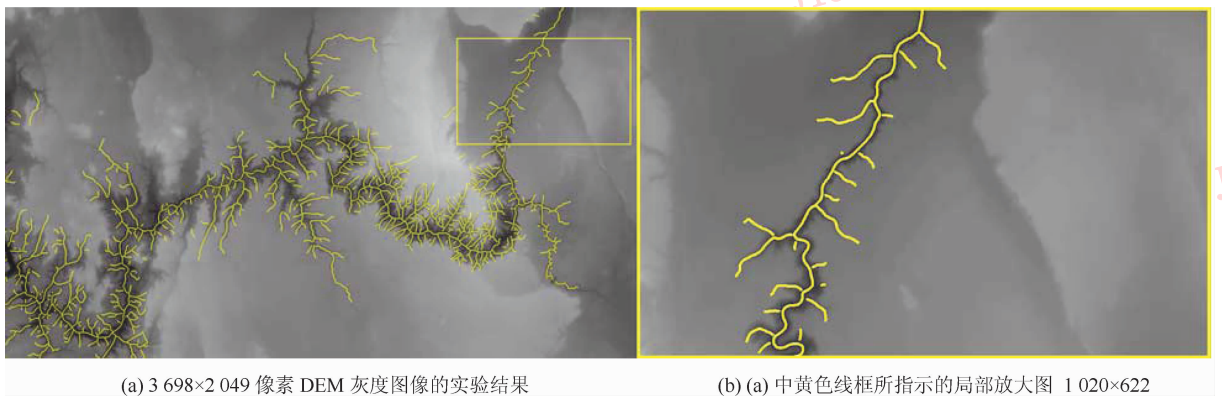


图 4 采用本文算法在更多测试 DEM 灰度图像的特征线提取结果及局部放大

Fig. 4 More results using our feature line extraction algorithm and some partial enlarged details

表 1 CPU 和 GPU 上算法性能比较

Table 1 Performance comparison on CPU and GPU

图像大小	时间/s		
	CPU 实现	GPU 技术实现	加速比
505 × 472	0.125	0.063	1.98
1 000 × 1 000	0.578	0.219	2.64
2 000 × 2 000	1.719	0.328	5.24
3 698 × 2 049	3.265	0.594	5.5
7 396 × 2 048	6.515	1.156	5.64
3 698 × 4 098	6.547	1.125	5.82
5 000 × 4 000	8.547	1.485	5.76

5 结 论

使用 GPU 技术加速基于 Hessian 矩阵的脊线检测算法,通过将算法中计算密集的图片卷积模块和

Hessian 矩阵特征值计算模块用 CUDA 框架进行并行,有效提高了脊线检测算法的效率。其中,对较大尺寸的 DEM 图像可以取得约 6 倍的加速比。

继续优化现有 GPU 加速技术,提高 GPU 技术的加速比。可能的措施有减少 GPU 与 CPU 之间的数据传输,利用 GPU 和 CPU 进行异步操作,分配合适的线程块大小和数目等。另外是改善 GPU 显存的利用率,使得能够处理更大尺寸的图片。目前能处理的 DEM 图像数据受限于实验用的 GPU 显存,未来对于更大规模的图像,例如 40 M 像素规模以上,可以采取图像分块的思想,用来处理更大规模的图像。最后,随着 GPU 硬件技术和 CUDA 框架的发展,最新的 GPU 硬件和 CUDA 框架已经有一些新的特性,例如 Fermi 显卡的 Streaming 技术能有效提升 CPU-GPU 之间的传输效率,未来可以利用 GPU 的最新技术来改善现有 GPU 加速的性能。

参考文献 (References)

- [1] Jenson S K, Domingue J O. Extraction topographic structure from digital elevation data for geographic information system analysis [J]. Photogrammetric Engineering and Remote Sensing, 1988, 54(11): 1593-1600.
- [2] Aumann G, Ebner H, Tang L. Automatic derivation of skeleton lines from digitized contours [J]. ISPRS Journal of Photogrammetry and Remote Sensing. 1991, 46:259-268.
- [3] Zhang Y J. Landscape Automatic General Research Based on DEM Skeleton Feature Line [D]. Xi'an: Northwest University, 2005. [张宇军. 基于DEM骨架特征点线的地貌自动综合研究[D]. 西安:西北大学, 2005.]
- [4] Guru D S, Shekar B H, Nagabhushan P. A simple and robust line detection algorithm based on small eigenvalue analysis [J]. Pattern Recognition Letters, 2004, 25(1):1-13.
- [5] Koller T M, Gerig G, Szekely G, et al. Multi-scale detection of curvilinear structures in 2-D and 3-D image data [C]// Proceedings of the 5th International Conference on Computer Vision. Boston; MA, 1995:864-869.
- [6] Steger C. An unbiased detector of curvilinear structures [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(2): 113-125.
- [7] Jacob M, Unser M. Design of steerable filters for feature detection using canny-like criteria [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2004, 26(8):1007-1019.
- [8] Liu L, Zhang D, You J. Detecting wide lines using isotropic nonlinear filtering [J]. IEEE Trans. Image Process, 2007, 16(6):1584-1595.
- [9] Li S X, Chang H X, Zhu C F. Fast curvi-linear structure extraction and delineation using density estimation [J]. Computer Vision and Image Understanding, 2009, 113(6):763-775.
- [10] Owens J D, Luebke D, Govindaraju N, et al. A survey of general-purpose computation on graphics hardware [J]. Computer Graphics Forum. 2007, 26(1):80-113.
- [11] Sudipta N S, Frahm J M, Pollefeys M, et al. Feature tracking and matching in video using programmable graphics hardware [J]. Machine Vision and Applications. 2007, 22(1): 207-217.
- [12] Li W, Wei X, Kaufman A. Implementing lattice Boltzmann computation on graphics hardware [J]. The Visual Computer, 2004, 19(7-8): 1432-2315.
- [13] Kruger J, Westermann R. Linear algebra operators for GPU implementation of numerical algorithms [C]//SIGGRAPH'03: ACM SIGGRAPH. San Diego, California; ACM 2003 Article, 2003:908-916.
- [14] Kinsner M, Capson D, Spence A. Scale-space ridge detection with GPU acceleration [C]// Electrical and Computer Engineering CCECE 2008. Canada; University Postal Outlet, 2008:1527-1530.
- [15] NVIDIA. NVIDIA CUDA: Compute Unified Device Architecture [EB/OL]. [2011-5-20]. http://developer.download.nvidia.com/compute/cuda/2_0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf.
- [16] Zhao K Y. CUDA thread implement model analysis (one) recruit-resolution of GPU. [EB/OL]. (2010-10-19) [2011-05-20]. <http://www.bianceng.cn/PC/Hardware/201010/19605.htm>. [赵开勇. CUDA线程执行模型分析(一)招兵——GPU的革命[EB/OL]. (2010-10-19) [2011-5-20]. <http://www.bianceng.cn/PC/Hardware/201010/19605.htm>.]