

同态滤波的一种 GPU 提速实现方法

张楠^{1,2}, 王建立¹, 王德俊^{1,2}

(1 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2 中国科学院 研究生院, 北京 100039)

摘 要: 为了提高光照不均图像的增强速率, 提出了基于 GPU 平台的同态滤波并行算法. 根据同态滤波算法的并行性, 利用 CUDA 软硬件体系架构, 实现了同态滤波算法向 GPU 上的移植. 利用多幅不同分辨率图像作为测试数据, 对比 CPU 和 GPU 方案的计算效率. 实验结果表明, GPU 实现方案大幅度提升了计算效率.

关键词: 图像增强; 同态滤波; 图形处理器; 计算统一设备架构

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2011)01-0185-04

A Method of Using GPU to Accelerate Homomorphic Filter

ZHANG Nan^{1,2}, WANG Jian-li¹, WANG De-jun^{1,2}

(1 Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; 2 Graduate University of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: In order to speed up the enhancement of non-uniform illumination image, a parallel homomorphic filter method based on GPU is presented. According to the parallelity of the algorithm, employing the CUDA programming framework, homomorphic filter based on GPU was implemented. Using various resolution images as test dataset, compared the computational efficiency of CPU and GPU. The experiment result shows that GPU can effectively implement the fast algorithm.

Key words: image enhancement; homomorphic filter; Graphics Processing Unit (GPU); Compute Unified Device Architecture (CUDA)

1 引言

近年来, 可编程图形处理器的性能飞速发展, GPU 已经具有很强的并行计算能力, 并具有更好的可编程性, 程序开发更为高效灵活. 利用 GPU 加速图像处理的技术逐渐成为研究的热点^[1].

图像增强是图像处理技术中的一个重要研究内容, 如何快速有效的增强图像是图像分析中的一个难点. 经典的同态滤波算法是在频率域上进行的, 用适当的滤波函数对低频部分和高频部分施加不同的影响, 再将图像变换回时域, 算法易于实现^[2]. 但同态滤波算法包含傅里叶变换和大量循环体, 当图像分辨率较高时, 运算量很大, 串行算法难以实现实时

增强.

文中针对 CPU 同态滤波计算效率的不足, 使用 GPU 对同态滤波算法进行并行算法改造, 充分发挥了 GPU 的并行处理和浮点运算能力, 大大地提高了同态滤波的效率.

2 GPU 特性分析

2.1 硬件模型

GPU 采用单指令多线程 (Single Instruction, Multiple Thread, SIMT) 执行模型, 单一控制部件向每条流水线分派指令, 同样的指令处理部件同时执行. 如图 1 所示, GPU 包含多个多处理器 (Multiprocessor), 每个多处理器又包含多个处理器 (Pro-

cessor). 每个多处理器包含四类内存, 其中共享内存为片上内存, 而常量内存、纹理内存和设备内存(显存)均可与主机内存通信^[3]. 这种架构为图像处理算法的并行改造提供了良好的硬件平台.

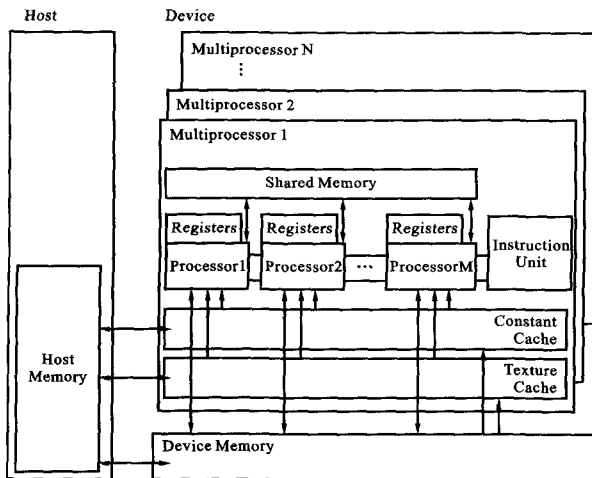


图1 硬件模型^[3]

2.2 编程模型

CUDA是NVIDIA在2007年推出的一种全新的GPU通用计算架构, 它包括一个硬件驱动程序和一个应用程序接口, 还提供了两个数学计算库CUFFT和CUBLAS. 抛开了以前的图形API方式, 而直接将GPU作为计算单元来管理和进行并行计算. CUDA编程语言是针对通用计算GPU的C语言开发环境, 使程序开发更为灵活高效.

CUDA的基本思想是支持大量的线程级并行, 并在硬件中动态地调度和执行这些线程. CUDA编程模型中, GPU被看作一个可以并行执行大量线程的计算设备, 作为主控CPU的协处理器执行运算. 如图2所示, 运行在GPU上的程序称为核(Kernel), 核以栅格(grid)的形式执行, 每个栅格由若干个线程块(block)组成, 每一个线程块又由最多512个线程(thread)组成. 各个线程通过自己的ID号, 取不同的数据并行执行Kernel.

图3为CUDA程序的编译流程^[4]. CUDA程序代码分为主机代码和设备代码两部分, 前者调用标准C编译器生成目标代码, 后者则由NVIDIA的C编译器生成相应的含有ptx(parallel thread execution)的目标代码, 之后两者链接生成完整的可执行程序.

3 并行同态滤波算法

3.1 同态滤波算法

同态滤波是以图像的照度-反射模型为处理

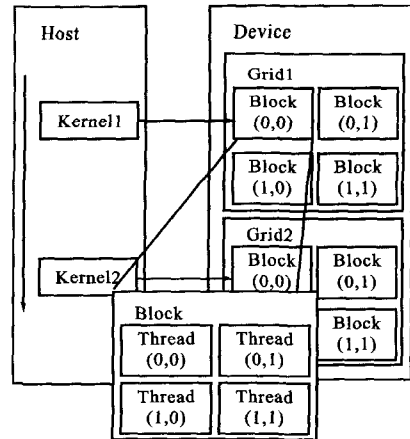


图2 线程划分

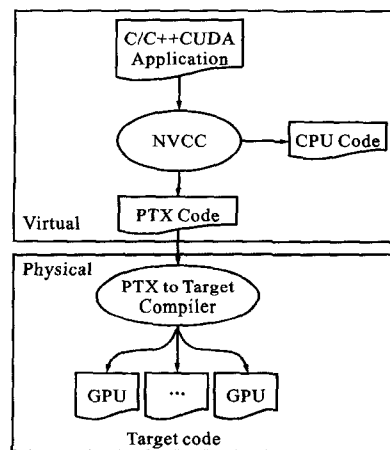


图3 CUDA程序的编译过程^[4]

基础, 利用压缩亮度范围和增强对比度来改善图像. 图像 $f(x, y)$ 可由照度分量 $i(x, y)$ 和反射分量 $r(x, y)$ 的乘积表示, 即 $f(x, y) = i(x, y)r(x, y)$.

反射分量反映图像内容, 随图像细节不同在空间上作快速变化, 在频谱上其能量集中于高频; 照度分量在空间上通常具有缓慢变化的性质, 其能量集中于低频.

同态滤波分为5个步骤, 如图4所示.

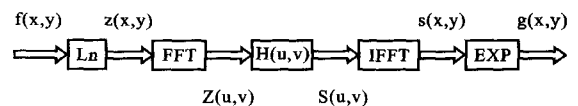


图4 同态滤波算法计算流程

3.2 并行同态滤波算法

图4中的每一步处理过程都具有数据的相关性, 对其进行并行处理只能在每一步处理过程内进行. 对数变换、滤波及反对数变换, 都属于图像处理操作中的点运算, 输出像素仅与当前输入像素有关, 而与图像中其它像素无数据交互, 因此这三个步骤均可以完全并行化, 特别适合GPU这种SIMT的

流式计算架构,用编写的核函数即可实现.对于傅里叶变换和反变换,可以调用 CUDA^[5] 提供的 CUFFT 库函数实现.

图 5 为同态滤波的 GPU 并行计算流程.

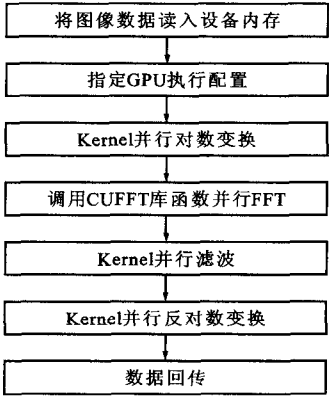


图 5 并行同态滤波流程

在 GPU 中进行任何计算,必须对 GPU 程序进行执行配置,即确定线程块数目和每个线程块中的线程数目,用以对要处理的问题进行划分.对数、滤波和反对数运算三个环节,各个像素各自计算,按照 CUDA 线程划分原则(见图 2),将 $M \times N$ 大小的图像划分为 gridDim 个子图像并行计算,每个子图像包含 blockDim 个像素,每个线程负责 1 个像素的计算,这样,点运算的并行计算就建立了.以 1024×1024 图像为例,在 GPU 上划分了 64×64 个线程块,每个线程块包含了 16×16 个线程,使 GPU 计算中每个线程块对应一个子图像,而每个线程对应其中一个像素.

FFT 部分计算选用 NVIDIA 开发的 CUFFT 库^[6].CUFFT 库是一个用 GPU 进行傅立叶变换的函数库,这是 NVIDIA 公司优化过的函数库,在进行大规模快速傅里叶变换的测试中,该函数库将提供 GPU 最佳的性能.CUFFT 模拟了 FFTW 库的运算机制,接口与 FFTW 库相似,不同的是 CUFFT 操作的数据储存在显存.

4 实验结果与分析

本文硬件平台为: NVIDIA 公司的 Geforce 8800GT GPU,显存 512MB,主频 1.5GHz; Intel Pentium Dual E2160 CPU,内存 1GB,主频 1.8GHz.

软件平台为:Windows XP 操作系统,程序开发环境为 Visual Studio 2005,CUDA Toolkit 1.1.

选择不同分辨率的室内光照不均图像,分别按照图 4、图 5 计算流程,进行基于 CPU 和 GPU 的同

态滤波算法,并统一采用单精度浮点数运算,以保证计算结果的可比性.其中 CPU 实现中的 FFT 计算部分选用了 FFTW 函数库,该库在高性能计算领域被广泛使用,是业界公认效率最高的 FFT 库^[7].

CPU 和 GPU 上所用的算法相同,区别仅是执行操作的线程数不同,因此 CPU 与 GPU 的增强效果是一致的.分别对分辨率为 256×256 、 512×512 、 1024×1024 及 2048×2048 的图像进行增强评价测试.图 6 为经过 GPU 并行同态滤波前后的增强图像.像质评价数据如表 1 所示.

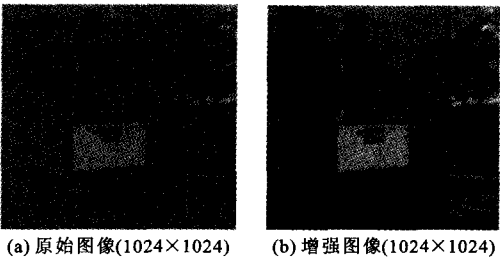


图 6 图像处理效果图

表 1 图像增强前后质量评价

图像	尺寸	亮度	对比度	熵
光照不足	1024×1024	10.52	12.94	4.83
增强后	1024×1024	37.38	26.11	5.53

图 7 为不同分辨率图像数据的 CPU 同态滤波和 GPU 同态滤波的时间对比,使用 CUDA 自带的时间测定函数来得到程序运行的时间,由于得到的时间只能为毫秒级,因此采用运行 10 次求平均时间的方式得到.其中 GPU 执行时间包含了数据加载与回传的时间.

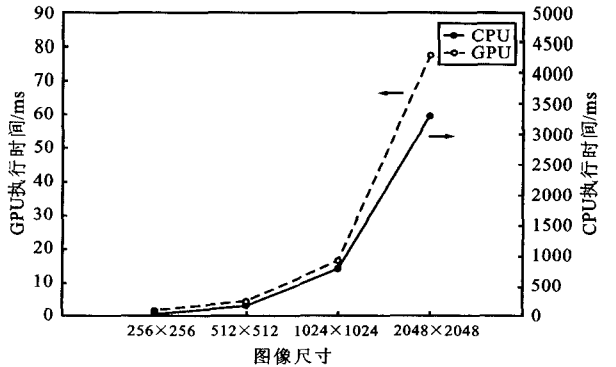


图 7 GPU 与 CPU 执行时间对比

如图 7 所示,左边 y 轴对应 GPU 算法执行时间,最小刻度 10ms,而右边 y 轴对应 CPU 算法执行时间,最小刻度 500ms.图中数据显示,在 GPU 上执行同态滤波算法相比 CPU 有明显的加速效果,

可以在 16ms 时间内处理 1024×1024 分辨率的光照不均图像,相比 CPU 加速比可达 49 倍.随着图像尺寸的增大,并行处理的性能提升倍数也随之增大,但并非线性增长,而是随着输入图像尺寸的增大,增幅趋于平缓,性能提升稳定在 42 倍左右^[8].

5 结束语

根据同态滤波算法的高并行性,充分利用 GPU 的硬件并行架构,及高度优化的 CUFFT 库,在 GPU 上实现了并行同态滤波算法.实验结果表明,相比 CPU,算法的 GPU 实现有很大的效率提升,在本平台上,对 1024×1024 图像进行处理时,速度可达 16ms,较大程度地提高了图像增强速度,可以实时增强高分辨率光照不均图像.这种并行化加速的思想对其他具有并行性的图像处理算法也有很好的借鉴.

参考文献:

- [1] Nvidia. NVIDIA official site [EB/OL]. [2009-10-09]. <http://www.nvidia.com>
 - [2] 陈春宁,王延杰.在频域中利用同态滤波增强图像对比度[J].微计算机信息,2007,23(2/3):264-266.
 - [3] Nvidia. NVIDIA CUDA programming guide Version 1.1 [EB/OL]. [2009-10-10]. www.nvidia.com/object/cuda_home.html.
 - [4] 多相复杂系统国家重点实验室多尺度离散模拟项目组.基于 GPU 的多尺度离散模拟并行计算[M].北京:科学出版社,2009.
 - [5] 阮军,韩定定.基于 CUDA 的快速 DCT 变换实现方法[J].微电子学与计算机,2009,26(8):201-205.
 - [6] Nvidia. CUFFT_Library_1.1 [EB/OL]. [2009-10-06]. http://www.nvidia.com/object/cuda_home.html
 - [7] Frigo M, Johnson S. FFTW: an adaptive software architecture for the FFT[C]// ICASSP Conference Proceedings. USA: Washington, 1998:1381-1384.
 - [8] 伍尤富.基于离散平稳小波变换的图像同态增强[J].微电子学与计算机,2008,25(2):151-153.
- 作者简介:**
张楠 女,(1984-),博士研究生.研究方向为基于 GPU 的图像处理.
王建立 男,(1971-),研究员,博士生导师.研究方向为光电跟踪技术.
王德俊 男,(1983-),硕士研究生.研究方向为图像处理.
-
- (上接第 184 页)
- [7] Christian Martyn Jones, Satnam Singh Dlay, Raouf Gorgui Naguib. Berger check prediction for concurrent error detection in the braun array multiplier[C]// The Third IEEE International Conference on Electronics, Circuits, and Systems. Italy: Bari, 1996: 81-84.
 - [8] Leblebici Y, Ozdemir H, Kepkep A, et al. A compact high-speed (31,5) parallel counter circuit based on capacitive threshold-logic gates[J]. Journal of Solid-State Circuits, 1996, 31(8):1177-1183.
 - [9] Jien-Chung Lo, Suchai Thanawastien, Rao T R N. Concurrent error detection in arithmetic and logical operations using berger codes[C]// The 9th Symposium on Computer Arithmetic. Orlando: Florida, 1989: 233-240.
 - [10] Lo J C, Thanawastien S, Rao T R N, et al. An SFS berger check prediction ALU and its application to self-checking processor designs[J]. IEEE Journal of Computer-Aided Design of Integrated Circuits and Systems, 1992, 11(4):525-540.
- 作者简介:**
张洵颖 男,(1981-),博士.研究方向为高性能容错微处理器设计、多核同构/异构 SoC 设计.