

CPU-GPU异构平台上平面光源检测方法的 并行化设计与实现

张志源

（西安交通大学 电信学院 计算机33，陕西 西安）

2017年5月

目录

1	绪论	4
1.1	背景与意义	4
1.2	研究现状	4
1.3	论文主要内容	5
1.4	论文组织框架	6
2	项目需求和整体解决方案	6
2.1	项目概况	6
2.2	多线程加速设计方案	6
2.3	GPU通用计算的应用	6
2.3.1	blank	7
3	GPU图像检测算法的实现	7
4	并行化设计的整体实现与测试	7
4.1	OpenCV和CUDA环境搭建	7
4.1.1	软硬件环境	7
4.1.2	Cmake重新编译OpenCV	7
4.1.3	Visual Studio 2013下对项目的配置方法	7
4.2	多线程优化的性能测试	7
4.3	在GPU下的图像检测算法性能测试	7
4.4	多线程条件下GPU函数的执行情况	7

5	总结与展望	7
5.1	论文工作总结	7
5.2	下一步研究内容	7
5.3	本章小结	7

1 绪论

1.1 背景与意义

智能制造是当前中国产业变革的主攻方向，2015年首次提出的“中国制造2025”，是中国政府实施制造强国战略第一个十年的行动纲领，旨在发展高技术含量的制造行业，改变中国制造业“大而不强”的局面^[1]。而智能检测是智能制造中的关键环节之一；能否满足智能检测中的实时性要求，直接影响了智能制造流水线中的生产效率。

在智能手机的生产线上，在手机液晶屏幕的质量检测这一环节，缺少直接有效的方法来实现自动化，因此只能安排专人来把关^[2]。笔者所参与的项目，主要的工作是使用基于图像识别的方法，在生产线上实现自动化设备来检测手机屏幕缺陷，以此代替传统人工检测。

为了满足实时性的要求，最根本的途径是提高检测算法的执行速度，这也是笔者的主要工作和研究重点；如果算法的时间开销过大，检测系统将难以和现场的生产设备对接。如何提高检测环节中复杂算法的执行效率，就是一个亟待解决、且有广泛应用价值的问题。

1.2 研究现状

在有限的计算资源下，利用并行计算是提高算法执行效率的最直接的方法。在该项目下，对同一组手机屏幕的一次检测中需要拍摄多张图片，也就意味着要在短时间内对多张图片执行相同的检测算法，在这个地方可以利用多核CPU的多线程并发，对每张图片分配一个线程执行算法。

如果深入到处理器的计算性能，在大规模并行计算领域，GPU（Graphics Processing Unit，图像处理器）和CPU（Central Processing Unit，中央处理器）相比，展现了更强大的浮点运算能力。GPU的可编程性在未开始发展时，开发人员要借助复杂的计算机图形学API来对GPU进行编程，这对非专业人员造成极大的困难^[3]。而近年来，GPU在计算性能不断提高的

同时，它的可编程性也在不断提高，意味着GPU可以在通用计算领域得到更广泛的应用；像这一类的GPU被称为通用GPU，即GPGPU（General Purpose GPU）^[4]。目前应用较广泛的GPGPU平台主要有CUDA（Compute Unified Device Architecture，统一计算设备架构）、OpenCL（Open Graphics Library，开放计算语言）。CUDA是显卡厂商NVIDIA推出的、基于他们公司生产的GPU开发出来的，使用C语言来设计需要的程序，对所进行的计算进行分配和管理^[3]。

借助CUDA的架构来对算法进行移植是目前常用的解决性能问题的方法。问题在于，对现有图像处理算法进行移植和优化仍然需要比较强的专业和理论基础，如果只是针对特定算法倒是可行，但是要用上述方法对该项目的检测算法进行移植，涉及到复杂的处理流程，移植算法需要很长的学习和研发周期。考虑到易用性和友好程度，这里着重关注OpenCV（Open Source Computer Vision Library）的GPU模块；这个模块最早在NVIDIA公司支持下进行开发，并于2011年春正式发行^[4]。目前为止也更新了大量由CUDA代码编写的图形处理算法，这也就意味着开发人员可以使用这些通用的算法API来利用GPU进行计算，免去了繁杂的算法设计和优化。由于OpenCV的开源特性，专家和爱好者可以共同维护和开发OpenCV的GPU模块，并不断完善，在图形处理方面有着良好的发展前景。

本次毕设的主要工作就是在CPU-GPU异构平台上，利用CUDA架构对原有的算法进行移植，利用GPU进行加速，并尝试与CPU多核多线程并行结合，探索其性能提高的方法。

1.3 论文主要内容

blank

1.4 论文组织框架

blank

2 项目需求和整体解决方案

2.1 项目概况

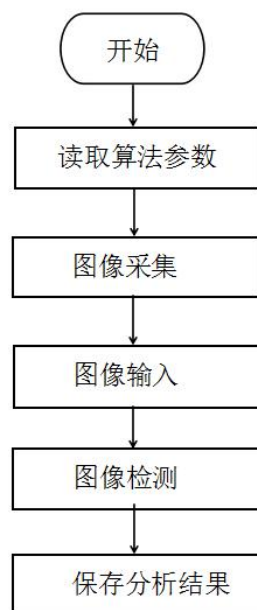


图 1: 手机屏幕缺陷检测整体流程

2.2 多线程加速设计方案

2.3 GPU通用计算的应用

blank

2.3.1 blank

blank

3 GPU图像检测算法的实现

4 并行化设计的整体实现与测试

4.1 OpenCV和CUDA环境搭建

4.1.1 软硬件环境

4.1.2 Cmake重新编译OpenCV

4.1.3 Visual Studio 2013下对项目的配置方法

4.2 多线程优化的性能测试

4.3 在GPU下的图像检测算法性能测试

4.4 多线程条件下GPU函数的执行情况

5 总结与展望

5.1 论文工作总结

5.2 下一步研究内容

5.3 本章小结

参考文献

- [1] 国务院印发.中国制造2025[EB/OL].http://news.xinhuanet.com/politics/2015-05/19/c_1115331338.html.
- [2] 易松松. 基于机器视觉的手机面板缺陷检测方法研究[D]. 哈尔滨工业大学, 2016.
- [3] 吕向阳. 基于CPU+GPU的图像处理异构并行计算研究[D]. 南昌大学, 2014.
- [4] 王锋,杜云飞,陈娟. GPGPU性能模型研究[J].计算机工程与科学,2013,35(12):1-7.
- [5] OpenCV官网CUDA主页面[EB/OL].<http://opencv.org/platforms/cuda.html>.
- [6] 标准C++库参考文档[EB/OL].<http://www.cplusplus.com/reference>.
- [7] Shane Cook.CUDA Programming_A Developer's Guide to Parallel Computing with GPUs[EB/OL].<http://www.nvidia.com>
- [8] CUDA_C_Programming_Guide[EB/OL].<http://www.nvidia.com>