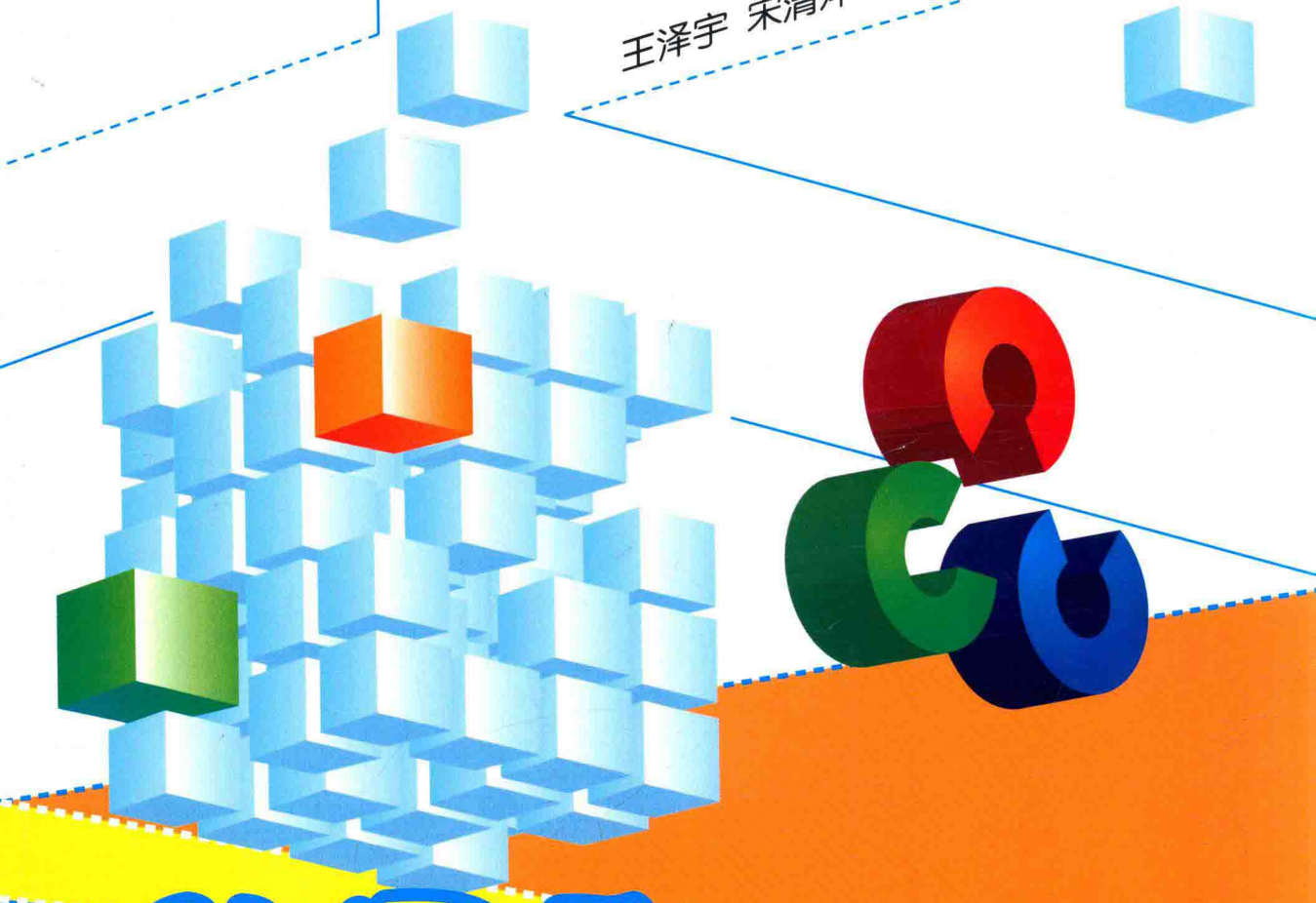
王泽宇 宋清洋 栾峰 编著



CUDA与OpenCV并行图像处理实战

清华大学出版社

CUDA与OpenCV并行图像处理实战

王泽宇 宋清洋 栾峰 编著

清华大学出版社
北京

内 容 简 介

本书主要介绍图像处理和GPU加速的基本原理、主要技术和典型应用。全书共分为5章，详细介绍了OpenCV的环境搭建，OpenCV在图像处理算法中的应用，OpenCV如何与CUDA进行编译，以及如何使用编译后的OpenCV库驱动GPU加速传统的图像处理算法。

本书可作为信号处理、通信工程、计算机应用、广播电视、自动控制、生物医学工程、地理信息等领域的工程技术人员，以及大专、本科院校相关专业的高年级学生研究图像处理技术的参考书。

图书在版编目(CIP)数据

CUDA与OpenCV并行图像处理实战/王泽宇，宋清洋，栾峰编著. —北京：清华大学出版社，2019
ISBN 978-7-302-51048-2

Ⅰ. ①C…　Ⅱ. ①王… ②宋… ③栾…　Ⅲ. ①图像处理软件—程序设计　Ⅳ. ①TP391.413

中国版本图书馆CIP数据核字(2018)第192050号

责任编辑：赵　凯　李　晔
封面设计：常雪影
责任校对：梁　毅
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：17.5　　**字　　数**：422千字
版　　次：2019年6月第1版　　**印　　次**：2019年6月第1次印刷
定　　价：69.00元

产品编号：072020-01

序言

数字图像处理是实现计算机视觉的关键技术。在当前实际应用中，计算机视觉系统对实时性和准确率的要求越来越高，需要处理的数据量越来越多，涉及的计算量也越来越大，这使得目前个人计算机的数据处理能力不能满足实际需求。但随着图形处理器(Graphic Processing Unit，GPU)的急速发展，使用 GPU 进行加速计算通常能获得处理速度的大幅提升，因此使用 GPU 建立实时、准确、高效的计算机视觉系统成为必然趋势和研究热点。

然而，目前使用 GPU 做图像处理的相关中文资料普遍存在不足：

第一，资料较少且信息零散，不成系统。通常是这个博客中讲解几个技术点，那个论文中介绍几种算法，缺少完整的、条理清晰的归纳。

第二，较为复杂的环境搭建让很多初学者望而却步。

本书是一种非常适合初次接触并行图像处理技术的本科生和研究生的入门级读物。书中，首先以 OpenCV 和 CUDA 的基础知识为出发点，介绍并行处理数据的工作原理；然后详细介绍 OpenCV 和 CUDA 的环境搭建过程和其中可能遇到的问题及解决方案，帮助读者顺利完成开发环境搭建；最后通过丰富的算法和例程，由浅入深地介绍多个并行图像处理算法，为读者未来更深入地研究并行图像处理技术提供实践基础。

我由衷地希望有志加入数字图像处理技术开发的莘莘学子，能凭借此书更快速地打好基础，通过对 GPU 的并行运算能力的理解与应用，开启图像处理技术研发的大门。

隆克平

2019 年 3 月于北京科技大学

前言

随着大数据时代的来临，越来越多的图像需要实时处理。随之而来的使用C++编程的机器视觉库OpenCV以及驱动GPU的CUDA也变得越来越火热。

OpenCV是机器视觉领域非常著名的开源库，它几乎被应用到机器视觉的所有领域，其功能几乎涵盖每个研究方向。OpenCV包含了底层的图像处理、中层的图像分析以及高层的视觉技术。而且，其算法紧跟视觉前沿，将最新的算法纳入其中。特别是OpenCV 2系列的出现，可以使用C++进行编程，并且可以使用GPU为图像处理进行加速。OpenCV在图像界是相当重要的工具，也是很多图像领域研究人员极力推荐的库。

CUDA作为一种并行计算架构，是以GPU为数据并行计算设备的软硬件体系。CUDA以C语言为基础，可以直接用C语言写出在显示芯片上执行的程序，而不需要去学习特定的显示芯片的指令或特殊的结构。因此，CUDA被广泛应用在视频编解码、金融、地质勘探、科学计算等领域。

作为并行图像处理的入门级教材，本书将并行计算架构CUDA和机器视觉库OpenCV结合，以大量示例程序为主线，详细介绍了如何搭建OpenCV环境，如何使用Cmake编译CUDA和OpenCV，以及环境搭建过程中可能出现的错误和解决方案。编写本书的初衷是希望更多初步接触GPU和图像处理的读者可以快速搭建好环境并快速了解OpenCV和CUDA的基础知识，节省入门消耗的时间。

由衷感谢我的导师宋清洋对于我学业和生活上的支持与鼓励，以及对这本书的付出。感谢栾峰老师对我学业上的指点，没有他的指点也就不会有这本书的诞生。感谢我的好兄弟郑建斌和学姐包锡伟在我学习图像处理的过程中对我的指导。

真心希望读者可以轻松地入门并行图像处理技术。由于作者水平有限，书中难免有不足之处，恳请读者批评指正。

王泽宇

2019年3月于东北大学

目录

第 1 章　并行图像处理概述 …… 1

1.1　计算机的构成 …… 1
1.1.1　计算机硬件构成 …… 1
1.1.2　显卡和 GPU …… 3
1.1.3　显卡的发展史 …… 7
1.2　并行计算 …… 8
1.3　并行图像处理 …… 9
1.3.1　并行图像处理的应用背景 …… 9
1.3.2　并行图像处理的原理 …… 10
1.3.3　并行图像处理的加速效果 …… 11
1.4　并行图像处理硬件平台 …… 12
1.5　并行图像处理软件平台 …… 14
1.5.1　开发平台——Visual Studio …… 14
1.5.2　计算机视觉库——OpenCV …… 14
1.5.3　统一设备架构——CUDA …… 15
1.5.4　并行编程开发工具——TBB …… 15
1.5.5　跨平台编译工具——CMake …… 16
1.6　常用软硬件搭配方案 …… 16
1.7　本书介绍 …… 17
1.8　本章小结 …… 17
参考文献 …… 17

第 2 章　OpenCV 及环境搭建 …… 18

2.1　OpenCV 的发展历程 …… 18
2.2　开发平台——Visual Studio 2010 …… 18
2.2.1　Visual Studio 简介 …… 19
2.2.2　安装 Visual Studio 2010 …… 20
2.3　搭建 OpenCV 2.4.9 …… 23
2.3.1　第一步　OpenCV 的下载和安装 …… 24

2.3.2 第二步 OpenCV 的环境变量配置 …… 25
2.3.3 第三步 工程项目内包含目录的配置 …… 28
2.3.4 第四步 库目录的配置 …… 32
2.3.5 第五步 附加依赖项的配置 …… 33
2.3.6 第六步 清单项配置 …… 34
2.3.7 第七步 Release 配置 …… 35
2.3.8 第八步 加入 OpenCV 动态链接库 …… 36
2.3.9 第九步 环境测试 …… 38
2.4 OpenCV 基本架构 …… 40
2.5 OpenCV 环境搭建中常见的问题及解决方案 …… 44
2.5.1 无法启动程序 …… 44
2.5.2 文件缺少 MSVCP110D.dll …… 46
2.5.3 Cannot find or open the PDB file …… 49
2.5.4 文件缺少 tbb_debug.dll …… 52
2.5.5 应用程序无法启动 0xc000007b …… 52
2.5.6 找不到头文件 …… 54
2.5.7 无法打开 lib 文件 …… 55
2.5.8 指针越界 cv::Exception …… 57
2.5.9 x86 与 x64 类型冲突 …… 58
2.6 本章小结 …… 59
2.7 参考文献 …… 59

第 3 章 OpenCV 常用函数和应用实例 …… 60

3.1 OpenCV 常用函数 …… 60
3.1.1 Mat 类 …… 60
3.1.2 imread 函数 …… 64
3.1.3 imshow 函数 …… 65
3.1.4 imwrite 函数 …… 67
3.2 反向算法 …… 67
3.3 图像融合 …… 75
3.3.1 覆盖型图像融合 …… 75
3.3.2 线性图像混合 …… 77
3.3.3 动画效果的线性混合 …… 79
3.4 图像去噪 …… 83
3.4.1 均值滤波 …… 83
3.4.2 高斯滤波 …… 87
3.4.3 非局部均值滤波 …… 90
3.5 双目视觉测量物体深度 …… 99
3.5.1 双目视觉原理 …… 99

3.5.2 双目视觉标定 …… 99
3.5.3 OpenCV 实现 …… 111
3.6 本章小结 …… 126
3.7 参考文献 …… 126

第 4 章 GPU 和 CUDA 的介绍和应用 …… 128

4.1 CUDA 的介绍 …… 128
4.2 GPU 的内部结构 …… 129
4.2.1 GPU 内部结构的简单介绍 …… 129
4.2.2 GPU 的架构 …… 131
4.2.3 常见 GPU 的挑选 …… 135
4.3 并行处理介绍 …… 137
4.4 CUDA 环境搭建 …… 139
4.4.1 CUDA 的下载 …… 139
4.4.2 CUDA 的安装 …… 140
4.4.3 CUDA 在 VS 中的测试 …… 144
4.4.4 CUDA 项目的创建 …… 145
4.5 CUDA C 语言 …… 153
4.5.1 C 语言最小扩展集 …… 153
4.5.2 运行时库 …… 156
4.6 程序示例 …… 179
4.6.1 Hello World 实现 …… 179
4.6.2 参数传递 …… 180
4.6.3 同步函数 …… 182
4.7 线程层次 …… 185
4.7.1 核函数调用和线程层次介绍 …… 185
4.7.2 矢量求和 …… 189
4.7.3 数据较多的矢量求和 …… 192
4.7.4 不同维度线程索引 …… 194
4.8 GPU 的存储器 …… 200
4.8.1 寄存器 …… 201
4.8.2 局部存储器 …… 202
4.8.3 共享存储器 …… 202
4.8.4 常数存储器 …… 205
4.8.5 纹理存储器 …… 205
4.8.6 全局存储器 …… 206
4.8.7 页锁定存储器 …… 206
4.8.8 可分页存储器 …… 206
4.9 本章小结 …… 207

参考文献 …… 207

第 5 章　基于 GPU 的并行图像处理 …… 208

5.1　CMake 和 TBB 的安装 …… 208
5.1.1　安装 CMake …… 208
5.1.2　安装 TBB …… 209
5.2　并行 OpenCV 库的生成 …… 211
5.3　VS 内的 OpenCV 环境搭建及环境测试 …… 216
5.3.1　常用工程文件的配置 …… 216
5.3.2　分别配置项目文件 …… 224
5.4　GPU 图像处理实例 …… 231
5.4.1　反向算法 …… 231
5.4.2　图像加法、减法 …… 235
5.4.3　图像腐蚀、膨胀 …… 238
5.4.4　非局部均值算法 …… 249
5.5　本章小结 …… 267
参考文献 …… 267

第1章 并行图像处理概述

1.1 计算机的构成

本节将从计算机的组成部分来普及计算机的硬件知识，将介绍计算机的中央处理器、主板等硬件，并详细介绍本书后面并行处理所基于的硬件——GPU(Graphics Processing Unit)。

1.1.1 计算机硬件构成

计算机的硬件系统可以按照工作原理和实际应用两种方式进行分类。

按照工作原理可以分为五大部分：运算器、控制器、存储器、输入设备和输出设备。运算器和控制器合称中央处理器，也就是 CPU(Central Processing Unit)，它是计算机工作的核心。存储器包括内存和外存，内存用来存储运行中的临时数据；外存用于存储应用程序和用户数据，硬盘光盘等都属于外存。输入设备用于给计算机输入程序或数据，如键盘、鼠标、扫描仪等。输出设备是将计算机处理后的结果送到外部设备，如显示器、打印机等。

从实际应用分为两方面：内部设备和外部设备。内部设备分为 CPU、内存、主板、显卡、声卡、网卡、光驱、电源。外部设备分为显示器、键盘、鼠标、音箱、耳机、打印机、扫描仪、手写板等与计算机相关的设备。

中央处理器，即 CPU，是一块超大规模的集成电路，是一台计算机的运算核心(Core)和控制核心(Control Unit)，它的主要功能是解释计算机指令以及处理计算机软件中的数据，CPU 的处理方式是传统的串行处理，如图 1-1 所示为 CPU 的外观。

显卡即 GPU，又被称作图形处理器、显示核心、视觉处理器、显示芯片，是一种专门在计算机、工作站、游戏机和一些移动设备(如平板电脑、智能手机等)上进行图像运算工作的微处理器，GPU 的内部结构与 CPU 不同，它处理数据的方式是并行处理。在多数情况下，显卡指的是包含了 GPU 芯片和辅助芯片的电路、风扇、接口等部分的一个完整的、可以直接使用的硬件设备，而 GPU 指的是做图像处理计算的 GPU 芯片。显卡的外

图 1-1 CPU 外观图

貌如图 1-2 所示，显卡内部的 GPU 芯片如图 1-3 所示。

图 1-2　计算机显卡

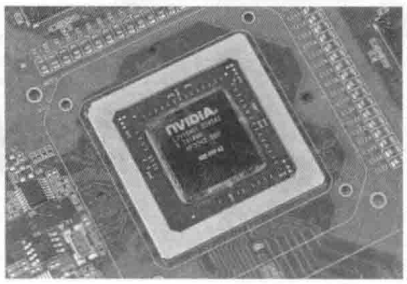


图 1-3　显卡内的 GPU 芯片

内存是计算机运行程序的地方。通常使用计算机运行的所有程序都是在内存中执行的，因此内存容量的大小对计算机的运行速度影响也比较大。计算机使用久了，运行出现卡顿现象，多数原因是内存容量不够大。内存所处的硬件被称作内存条，内存条的外观如图 1-4 和图 1-5 所示。在内存条生产领域，较为知名的企业有三星、金士顿、英飞凌、现代、Smart 等。近几年，这些内存条生产商逐步采用了小型的内存条来代替传统的大内存条，俗称小卡和大卡。小卡与大卡的区别主要体现在小卡宽度比大卡窄，小卡也可以插在原本大卡的接口上，不过两边没有扣子卡住小卡。小卡的优势是体积小，给其他硬件提供了更大的空间，不过其劣势就是集成度太高，散热等性能不如大卡，同样内存容量的大卡和小卡，从稳定性角度来看，大卡更胜一筹。

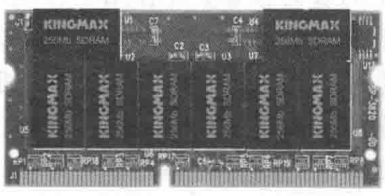


图 1-4　笔记本电脑中的内存条

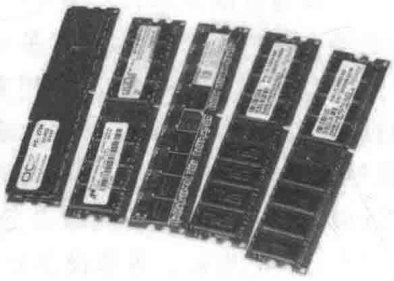


图 1-5　台式机中的老式内存条(大卡)

硬盘作为构成计算机硬件系统的存储设备，具有非常重要的地位。可以说，没有硬盘，计算机就无法正常工作。硬盘集机、电、磁于一体，结构相当复杂，图 1-6 和图 1-7 分别为硬盘的外部结构和内部结构。硬盘主要分成固态硬盘(Solid State Drive，SSD)、传统硬盘(Hard Disk Drive，HDD)和混合硬盘(Hybrid Hard Drive，HHD)三大类。固态硬盘是使用固态电子存储芯片阵列制成的硬盘，由控制单元和存储单元(Flash 芯片、DRAM 芯片)组成。固态硬盘的优点是：读写速度快；抗摔性好；低功耗；无噪声；工作温度范围大；轻便。缺点是：容量小；寿命有限；售价高。传统硬盘就是常说的机械硬盘。机械硬盘

图 1-6　硬盘的外部结构

使用了传统的工艺，虽然读写速度和稳定性不如 SSD，但是因为造价成本低，所以仍然占有很大的市场。混合硬盘是将 SSD 和 HDD 两种硬盘混合在一起，它既包含了 HHD 的大容量，又有 SSD 的闪存模块，目前市面上常用的笔记本电脑都是使用这种硬盘。

主板，又叫主机板、系统板或母板。它安装在机箱内，是计算机最基本的也是最重要的部件之一。主板一般为矩形电路板，上面安装了组成计算机的主要电路系统，一般有 BIOS 芯片、I/O 控制芯片、键和面板控制开关接口、指示灯插接件、扩充插槽、主板及插卡的直流电源供电接插件，通常情况下还会焊接上声卡、网卡等装置，CPU 作为处理数据的核心，也会焊接在主板上。其外观如图 1-8 所示。

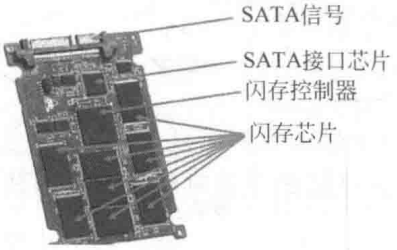


图 1-7　固态硬盘的内部结构

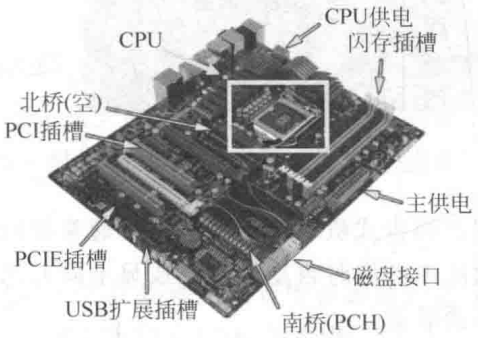


图 1-8　主板外观

1.1.2　显卡和 GPU

1.1.1 节已经对计算机中的显卡做了简单介绍，本节将针对后面并行处理需要用的硬件平台显卡和 GPU 做详细的介绍。

1. 显卡的分类

显卡的全称是显示接口卡，是计算机最基本的配置之一。显卡作为计算机的重要组成部分，承担输出显示图形的任务。用途是将计算机系统所需要的显示信息进行转换驱动，并向显示器提供信号，控制显示器的正确显示。没有显卡，计算机中的图像就无法处理，更无法在显示器上输出。

显卡目前分两种——集成显卡和独立显卡。

独立显卡简称为独显，是一个独立于主板之外的硬件，独立显卡具备单独的显存，不占用系统内存(但是独立显存不够用时可以共享内存为显存)，技术上优于集成显卡。独显由于拥有独立的一套运行环境，这使得其核心运算有很大的发挥空间，因此相对于集成显卡有更好的性能。如图 1-9 所示为独立显卡的配套硬件及其 GPU 芯片。

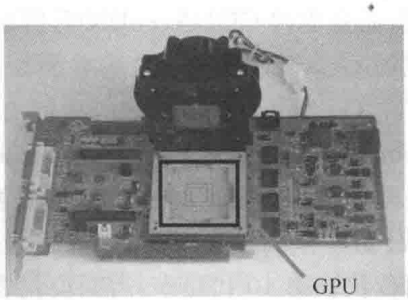


图 1-9　独立显卡 GPU 芯片

集成显卡简称集显，不带有显存，性能一般都比不过同等级的独立显卡。集成显卡分为两种：一种是焊接在主板的北桥中；另外一种是封装在 CPU 中，与 CPU 放在一起工作，被称为核心显卡，简称核显。两种不同的集成显卡因为没有独立的显存，所以工作时的显存都是从内存中分享而来，当运行较大型的程序或

游戏时，会占用相当一部分内存。近些年核显的性能得到了巨大的飞跃，核心显卡对内存的依赖越来越严重，在一定程度上也会影响 CPU 的性能。如图 1-10 和图 1-11 所示为计算机中焊接在北桥中的集成显卡。

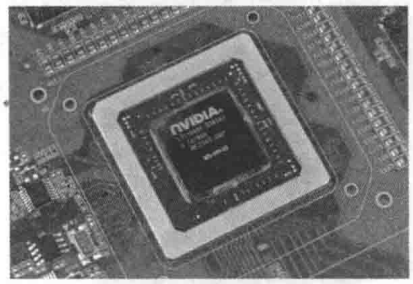


图 1-10　集成显卡

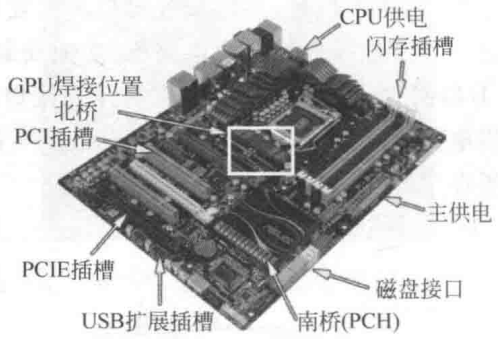


图 1-11　主板上的集成显卡

现在的台式机和笔记本电脑中通常都同时拥有独立显卡和集成显卡，有一些会智能地在显示简单图像时自动使用集成显卡降低功耗，在需要高清画质等时候自动使用独立显卡提高画面效果。

2. NVIDIA 显卡

显卡的厂商主要有英伟达(NVIDIA)和 AMD 两家公司，现在的 AMD 显卡是原本制作 GPU 的 ATI 公司，后被 AMD 公司收购合并成一家，这两家的显卡俗称为 N 卡(NVIDIA 公司的显卡)和 A 卡(AMD 公司的显卡)。虽然苹果、Intel 等公司也在开发显卡，但市场份额远不如这两家。因为本书后续用到的 CUDA 都是使用了 NVIDIA 公司的独立显卡，所以接下来将对目前市面上常见的 NVIDIA 公司的独立显卡进行介绍。

目前 NVIDIA 公司的独立显卡主要分成三大类：Tesla 系列、Quadro 系列和 GeForce 系列。Tesla 系列显卡是专用的服务器级别的显卡，常用于大规模并行计算，超长时间的连续运行也不会大幅降低运行速度，非常适用于机器学习，但是一般价格很高昂，代表显卡有 Tesla K40 和 Tesla K80，如图 1-12 所示为 Tesla 显卡的外形。有人将 Quadro 显卡也称为 Q 卡，Q 系列的显卡是专业的图形设计显卡，在制图方面被广泛使用。其价格低于 Tesla 显卡，长时间持续工作的续航能力也低于 Tesla 显卡，如图 1-13 所示为 Quadro 显卡。GeForce 系列就是市面上最常见的游戏显卡。GeForce 系列的中文名称是“精视”，主要面对大众用户，用途主要为娱乐，支持市面上绝大多数的 3D 游戏，也是 NVIDIA 公司销售量最高的类型，到现在为止的最新版为 GTX 1080Ti。相对于专业显卡，GeForce 系列相对便宜，而且支持 CUDA、cuDNN，所以现在做深度学习一般也会使用 GTX 1080Ti。市面上常见的 GeForce 系列显卡基本都是 GTX 系列。如前面提到的 GTX 1080Ti 和 GTX 1080。其中 GTX 是定位性能级，从前有 GT 和 GTX 两大类，GT 的全称为 Graphics Processor protoType，定位为次高端显卡；GTX 全称为 Graphics Processor prototype eXtreme，定位为高端显卡。随着显卡技术的进步，GT 系列已经渐渐为低端显卡，而现在市面上常见的 GeForce 系列显卡也都是 GTX 开头的显卡。GTX 显卡后的编号，以 GTX 1080Ti 为例，前两位代表第 10 代显卡，后两位代表性能，其实本质上是不同的架构和工艺，有没有 Ti 代表是否为增强版。从性能角度，GTX 1080Ti 优于 GTX 1080，GTX 1080 优于 GTX 1070，

GTX 1080 优于 GTX 980。

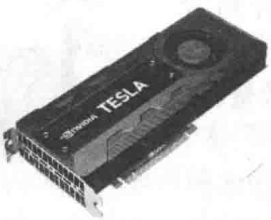


图 1-12　Tesla 显卡

图 1-13　Quadro 显卡

以上介绍的都是台式机的独立显卡和集成显卡，笔记本电脑因为没有较大的空间，所以其独立显卡也制作得非常小巧，并且焊接在主板上，如图 1-14 所示为笔记本电脑中的独立显卡。笔记本电脑中的独立显卡没有台式机中独立显卡那么好的风扇、辅助电路等，所以性能远不如同名称台式机的独立显卡，但是这种独立显卡也比普通的集成显卡性能要好，而且这种独立显卡也支持 CUDA 并行计算。

3. 性能参数介绍

显卡的显存也称为帧缓存，它的作用是存储显卡芯片处理过或者即将提取的渲染数据，同计算机的内存一样，显存是显卡用来存储要处理的图形信息的部件，如图 1-15 所示。

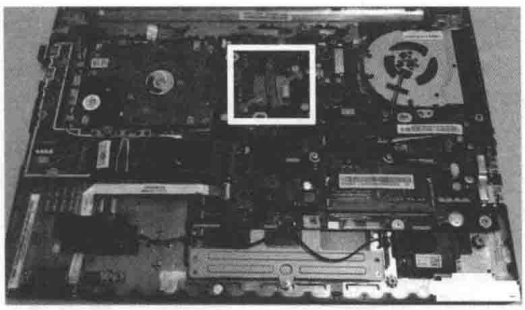


图 1-14　笔记本上的独立显卡

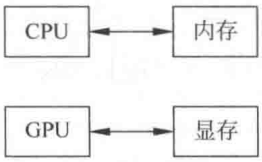


图 1-15　GPU 与显存的关系

显存位宽是显存在一个时钟周期内所能传送数据的位数，位数越大则瞬间所能传输的数据量越大，是显存的好坏的重要参数之一。显存带宽＝显存频率×显存位宽÷8，那么在显存频率相当的情况下，显存位宽决定显存带宽的大小。同样，显存频率为 500MHz 的 128 位和 256 位显存，那么它俩的显存带宽将分别为：128 位显存带宽＝500MHz×128÷8＝8GB/s，而 256 位显存带宽＝500MHz×256÷8＝16GB/s，是 128 位的 2 倍，可见显存位宽在显存数据中的重要性。

核心频率指的是显卡显示核心的工作频率，它在一定程度上反映出了显示核心的性能。可以将其类比为 CPU 的主频，数值越高，性能越好。

显存频率是指默认情况下，该显存在显卡上工作时的频率，以兆赫兹(MHz)为单位。显存频率一定程度上反映着该显存的速度。在一定程度上可以类比为计算机的内存频率。

4. GPU-Z 查看显卡参数

显卡的性能参数一般在购买显卡附赠的说明书上会有详细说明，或者网上也有同型号显卡的性能参数。如果在不确定自己显卡的型号，或者想给显卡做一个详细的测试，则需要

较专业的测试软件——GPU-Z，其 Logo 如图 1-16 所示。

GPU-Z 是一款非常出名的处理器识别工具，软件约占 4Mb，非常轻便[1]。GPU-Z 可以检测如下参数：

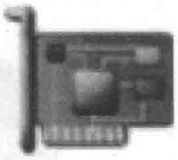


图 1-16 GPU-Z 的 Logo

① 检测显卡的型号、制作工艺、核心面积、晶体管数量、渲染器数量及生产厂商。

② 检测光栅和着色器处理单元数量及 DirectX 支持版本。

③ 检测 GPU 核心、着色器和显存运行频率，显存类型（生产厂商）。

④ 检测像素填充率和材质填充率速度。

⑤ 实时监测 GPU 温度、GPU 使用率、显存使用率以及风扇转速等相关信息。

⑥ 检查显卡的插槽类型和显卡所支持的附加功能与显卡驱动信息及系统版本。

主要特性有：

① 支持 NIVIDIA、ATI、AMD 和 Intel 的显卡或图形设备。

② 显示适配器型号和 GPU 型号，并且能显示其具体信息。

③ 显示显卡加速，默认时钟频率以及 3D 时钟频率（如果设备支持）。

④ 软件内包括了一个 GPU 负载测试，来监测显卡运行在 PCI-Express lane 上的状态。

⑤ 结果信息经过验证。

⑥ GPU-Z 可以备份显卡的 BIOS。

⑦ 不需要安装，但支持安装操作。

⑧ 支持 Windows XP/Vista/Windows 7/Windows 8/Windows 10。

如图 1-17 所示为 GPU-Z 检测 GTX 1070 的结果截图，GPU-Z 的一大优势是测量的很多参数都是基于硬件当时的状态，所以造假的 GPU 很难在 GPU-Z 的测试下鱼目混珠。

图 1-17 GPU-Z 检测 GTX 1070 显卡

1.1.3 显卡的发展史

了解显卡的基础知识后，本节将介绍显卡的发展历史。

早期的计算机因为没有图形界面，所以没有考虑到图像显示的问题。随着计算机的发展和图形操作系统的不断开发(最典型的是 Microsoft 公司的 Windows 系列)，图像界面已经普及，专门应对图像显示的 GPU 也在此期间得到高速的发展。

最早的个人计算机的图形处理单元是 IBM 公司在 1981 年推出的 CGA(Color Graphics Adapter)和在 1984 年推出的 EGA(Enhanced Graphics Adapter)。EGA 能同时显示 16 色，分辨率可以达到 640×350 像素，帧缓存达到 256KB。

IBM 公司在 1987 年又推出了 VGA(vidio graphics array)。VGA 实现了在单个芯片上包含图形的所有操作，如硬件的平滑滚动、屏幕的分割和扫描等操作。由于 Windows 操作系统的广泛使用和快速发展的趋势，各种 2D 和 3D 的图像处理芯片也相继出现。

当时高档的图形工作站霸占了 CAD 工作站的整个主流市场，硅谷公司(SGI)在 1992 年推出了 OpenGL，使其成为了第一个 2D 和 3D 操作系统的 API。1994 年，Matrox 公司为了发展个人计算机的 CAD 市场，推出了第一个应用于计算机的 3D 图形加速器 Matrox Impression。

1998 年，NVIDIA 公司推出了实时交互视频(Real-time Interactive Video and Animation accelerator，RIVA)和动画加速器(Twin Texel，TNT)。

1999 年，NVIDIA 公司发布了 GeForce 256，它实现了在芯片上集成变换、光照、建材、纹理和染色引擎，同时能够与 OpenGL 1.2 和 DirectX 7.0 兼容，达到了 SGI 的高端专业 3D 工作站的水平。2001 年，NVIDIA 推出的 GeForce3 优化了固定流水线的工作模式，首次提出了顶点着色器和像素着色器的概念，初步实现了部分电路的用户可编程性，其可编辑性使图像效果的实现不再受显卡的固定渲染管线限制，从而使多个顶点和多个像素点流入处理单元，分别通过同一程序进行独立的处理。

NVIDIA 公司的主要竞争对手 ATI(Array Technology Industry inc，曾在图形渲染技术上领先，在 2006 年被 AMD 公司收购)公司在 2002 年推出了第一个符合 DirectX 9.0 规范的加速器。在该加速器中，顶点着色器和像素着色器可以方便、灵活地实现循环和长浮点数的运算。DirectX 9.0 进一步加强了像素着色器和顶点着色器的功能，使原来的汇编级的语言发展成为 C 语言风格的高级语言 HLSL，进而产生了类似于 CPU 的程序编译的概念。在同一时期，OpenGL 也根据用户的需要不断地改进，发展成为 OpenGL 1.5 以及后来的 OpenGL 2.0。并在此过程中形成了类似 C 语言风格的高级染色语言 GLSL。

2006 年，Microsoft 公司推出了 DirectX 10，其试图统一各种染色器，在理论上消除处理的瓶颈。此时，NVIDIA 公司推出了通用并行架构(Compute Unified Device Architecture，CUDA)，AMD/ATI 推出了 CAL/CTM 和后来的 Stream SDK 作为 GPU 编程模型的抽象层。这些 GPU 编程框架使用户可以利用 GPU 强大的计算能力进行通用计算，实现 GPGPU 计算。

2010 年，NVIDIA 公司推出的 GPU Fermi 架构集成了大约 30 亿个晶体管，其拥有 512 个 CUDA 核心，存储器接口达到 384 位宽，存储器峰值带宽达到了 230GB/s，其主要应用于实时图形处理和大规模并行计算领域。同时，AMD 采用 40nm 工艺推出了 Radeon 系列，有

20亿晶体管，也转向通用计算和移动图形计算领域的研究。此外，NVIDIA公司的CUDA编程框架和国际媒体处理标准协会KHRONOS推出的并行计算语言标准OpenCL都在很大程度上加速了GPGPU的发展。

2016年，NVIDIA公司推出了显卡GTX 1080和CUDA 7.5，在2017年又推出了GTX1080Ti和CUDA 8.0。

在这段发展史中，需要记住的时间节点是在2010年，在那一年NVIDIA做出了一款很实用的CUDA，这个CUDA已经可以做一些很全面的开发。这个时间点也说明，如果想使用CUDA，尽量使用2010年以后推出的NVIDIA显卡，虽然最近AMD公司推出的新版显卡也可以支持CUDA，但因为架构不同，兼容性还是不如NVIDIA自己的显卡。

图1-18所示为显卡发展的简单历史。

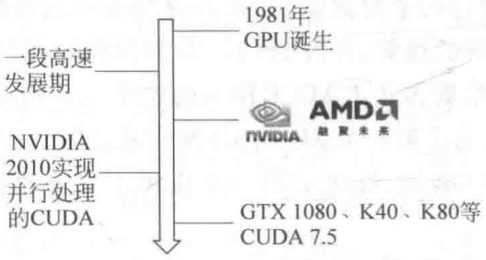


图1-18 显卡发展历史的简略图

1.2 并行计算

并行计算(Parallel Computing)或称平行计算是相对于串行计算来说的。它是一种一次可执行多个指令的算法，目的是提高计算的速度，通过扩大问题求解规模，解决大型而复杂的计算问题。并行计算可分为时间上的并行和空间上的并行。时间上的并行就是指流水线技术，而空间上的并行则是指用多个处理器并发的执行计算。

从广义上来讲，并行计算包括时间上的并行处理和空间上的并行处理，时间上的并行主要指在程序执行多条指令时，重叠进行操作的一种准并行处理实现技术，另外，多任务分时分复用也算是一种并行处理技术。空间上的并行则是用多个单元并发的执行计算，这些处理单元可以是多点式分布的CPU或者GPU，或者是一个GPU内部的许多处理线程，还可以是其他的异构形式。可见，时间上的并行并不能算是严格意义上的并行计算技术，而如何更有效地在空间上并行处理则是现在主流的研究方向[2]。

在并行处理进入公众视野之前，传统的串行处理是数据处理的主流。传统的串行计算就如同是沙漏中的沙子，堆积如山的数据只能排队等着被处理器逐一处理。所谓的多任务并行处理，也只是对不同任务分配不同的时间段来处理，这种时分复用的处理方法，并没有从本质上提高整体运算的速度。为了解决庞大数据量和有限处理能力之间的矛盾，研究者们发明了多核技术甚至是多处理器技术，目的也类似于在大量的沙粒下面多开通一些通道，让沙粒更快地流下，如图1-19左图所示，这也是

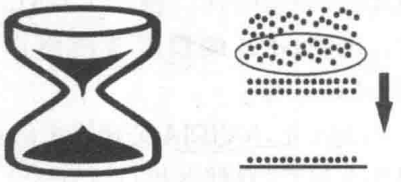


图1-19 并行处理示意图