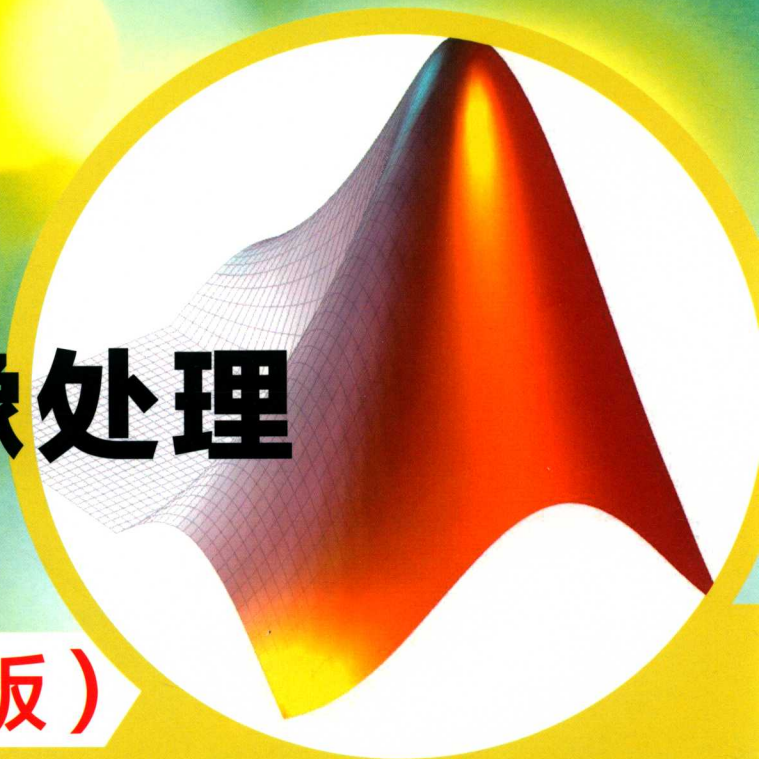


数字图像处理

基础与实践

(MATLAB版)



孙忠贵 编著

清华大学出版社

内容简介

本书以MATLAB软件为工具，系统地介绍了数字图像处理的基本原理、方法和应用。全书共分8章，主要内容包括：绪论、数字图像的表示、数字图像的变换、数字图像的增强、数字图像的复原、数字图像的分割、数字图像的识别、数字图像的压缩。本书可作为高等院校计算机专业及相关专业的教材，也可供从事图像处理工作的工程技术人员参考。

数字图像处理 基础与实践 (MATLAB版)

孙忠贵 编著

清华大学出版社
北京

内 容 简 介

本书主要介绍数字图像处理的基本内容及相应的 MATLAB 程序实现。主要内容包括图像的基本操作、图像的基本运算、图像变换、图像的形态学操作、图像增强、图像去噪、图像分割等。在内容组织上注重理论与实践的相辅相成,一方面通过对理论内容进行简明扼要的介绍,使读者能够顺利进入实践环节,另一方面通过实践操作使读者进一步理解并掌握相关内容的理论本质。本书提供了大量程序代码和编程素材。

本书适合作为高等院校数字图像处理及相关课程的本科生和研究生教材,同时也可供相关科研人员、工程技术人员参考。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数字图像处理基础与实践:MATLAB版/孙忠贵编著. —北京:清华大学出版社,2016
ISBN 978-7-302-45655-1

I. ①数… II. ①孙… III. ①数字图象处理—研究 IV. ①TN911.73

中国版本图书馆 CIP 数据核字(2016)第 285431 号

责任编辑:许 龙 刘远星

封面设计:常雪影

责任校对:赵丽敏

责任印制:宋 林

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

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×230mm 印 张:8.5 字 数:185千字

版 次:2016年11月第1版 印 次:2016年11月第1次印刷

印 数:1~2000

定 价:26.00元

产品编号:072222-01

随着计算机技术及信息技术的发展,数字图像在诸多领域中得到广泛应用,数字图像处理的学习也受到了越来越多的关注。数字图像的学习内容既涉及严谨的理论基础,又需要熟练的编程技能,将二者在知识体系上密切结合并相辅相成是本书写作的主要动机。

本书介绍数字图像若干研究方向的基本知识,在介绍这些理论知识的同时嵌入了必要的程序示例,力求使读者通过动手实践进一步理解并掌握数字图像研究的理论本质,同时也掌握相应的图像处理 MATLAB 编程技能。

本书各章最后以实验报告形式设计了实验环节,这部分内容在编写过程中注意了实用性、趣味性及启发性。相信通过完成实验报告,能够激发读者对图像处理的学习兴趣及研究热情。另外,本书中涉及的部分图片、代码及其他参考资料等在网上有电子资源,读者可通过扫描封底二维码下载。

全书共分为 9 章,涵盖了数字图像处理的若干研究内容,如图像的基本操作、图像的基本运算、图像变换、图像的形态学操作、图像增强、图像去噪、图像分割等。

本书的写作结合了编者多年来在数字图像处理方面的教学实践与研究经验,并受到山东省应用型人才培养特色名校建设工程及山东省自然科学基金(ZR2014FM032)的资助。全书的完成,参考和引用了大量同行的研究成果,在此向原作者表示感谢,并且受到诸多文献、资源的启发,书中引用得不完全,请原作者及各位同行多多谅解。

由于编者水平所限,书中难免出现不当之处,敬请各位读者批评指正。

作者

2016 年 7 月

第 1 章 绪论	1
1.1 基本概念	1
1.2 MATLAB 基础	3
1.2.1 MATLAB 工具箱安装	3
1.2.2 几个常用的 MATLAB 命令	3
1.2.3 MATLAB 的内联函数与匿名函数	5
1.2.4 MATLAB GPU 编程基础	6
1.3 本书内容	8
本章实验	9
实验一 MATLAB 工具箱安装	9
实验二 MATLAB 基本操作	9
第 2 章 图像的基本操作	10
2.1 数字图像的离散化表示	10
2.1.1 灰度图像	10
2.1.2 彩色图像	12
2.2 数字图像的读、写和显示	13
2.2.1 图像的读取	13
2.2.2 图像的显示	14
2.2.3 图像的保存	16
2.3 邻域操作与块操作	16
2.3.1 图像的邻域操作	16
2.3.2 图像的块操作	18
本章实验	19
实验一 灰度图像的基本操作	19
实验二 彩色图像的基本操作	20
实验三 图像的邻域操作	21

第 3 章 图像的基本运算	22
3.1 代数运算	22
3.1.1 图像的叠加函数	23
3.1.2 绝对值差函数	23
3.1.3 图像的减法运算	25
3.1.4 图像的乘法运算	25
3.1.5 图像的除法运算	26
3.1.6 图像的线性组合	27
3.1.7 图像的求补运算	27
3.2 几何变换	28
3.2.1 改变图像大小	28
3.2.2 图像的剪切	29
3.2.3 图像的旋转	30
3.2.4 图像的几何形变	31
本章实验	33
实验一 图像亮度的自适应调整	33
实验二 使用基本运算添加图像水印	33
实验三 仿射变换的 MATLAB 实现	35
第 4 章 图像的频域操作	36
4.1 傅里叶变换	36
4.1.1 连续傅里叶变换	37
4.1.2 离散傅里叶变换	38
4.1.3 离散傅里叶变换的实现	38
4.1.4 快速卷积的离散傅里叶变换实现	40
4.2 离散余弦变换	42
4.2.1 离散余弦变换定义	42
4.2.2 离散余弦变换的实现	43
4.3 小波变换	44
4.3.1 小波变换的定义	44
4.3.2 离散小波变换的实现	47
4.4 Hough 变换与 Radon 变换	49
4.4.1 Hough 变换	49
4.4.2 Radon 变换及实现	50

本章实验	52
实验一 利用傅里叶变换进行文本定位	52
实验二 离散余弦变换练习	53
实验三 小波变换练习	53
实验四 Radon 变换练习	54
第 5 章 形态学图像处理	55
5.1 数学形态学的集合论基础	55
5.2 二值形态学	56
5.3 灰度形态学	57
5.4 形态学操作的 MATLAB 实现	58
本章实验 利用 imerode 和 imdilate 函数实现图像的开、闭操作	60
第 6 章 图像的空域增强	62
6.1 基本知识	62
6.2 灰度变换	63
6.3 直方图处理	64
6.3.1 直方图	64
6.3.2 直方图均衡化	65
6.4 空域滤波	66
6.4.1 平滑滤波	67
6.4.2 锐化滤波	70
本章实验	73
实验一 直方图均衡化编程实现	73
实验二 灰度变换与直方图均衡化比较	74
实验三 图像空域滤波的块操作实现	74
第 7 章 图像的频域增强	75
7.1 低通滤波	75
7.1.1 傅里叶变换低通滤波	76
7.1.2 小波变换低通滤波	78
7.2 高通滤波	79
本章实验	83
实验一 巴特沃斯滤波器的快速算法实现	83
实验二 频域滤波器的图像滤波	83

第 8 章 图像去噪	84
8.1 噪声模型及实现	84
8.1.1 两类常见的噪声模型	84
8.1.2 噪声添加的实现	85
8.2 噪声滤除及效果评价	86
8.3 细节保持的滤波器	88
8.3.1 灰度最小方差滤波器	88
8.3.2 k 近邻(KNN)平滑滤波器	89
8.3.3 自适应中值滤波器	89
8.3.4 双边滤波器	92
8.3.5 非局部均值滤波器	92
本章实验	93
实验一 不同滤波器的滤波性能比较	93
实验二 细节保持滤波器的实现	94
第 9 章 图像分割	95
9.1 点检测与线检测	95
9.1.1 点检测	96
9.1.2 线检测	97
9.2 边缘检测	100
9.2.1 图像不连续性的数学刻画	100
9.2.2 Roberts 算子	102
9.2.3 Sobel 算子	103
9.2.4 Prewitt 算子	103
9.2.5 拉普拉斯算子	104
9.2.6 LOG 算子与 DOG 算子	104
9.2.7 Canny 算子	106
9.2.8 形态学算子	108
9.3 基于灰度阈值的图像分割	108
9.3.1 全局阈值分割	109
9.3.2 局部阈值分割和多阈值分割	113
9.4 区域生长法与分裂合并法进行图像分割	114
9.4.1 区域生长法	114
9.4.2 分裂合并法	115

9.5 使用分水岭法进行图像分割	119
本章实验	121
实验一 基于 Radon 变换的灰度图像线检测	121
实验二 不同检测算子的性能比较	121
实验三 阈值分割算法比较	122
实验四 图像的四叉树分解	122
实验五 分水岭算法进行图像分割	123
附录 A 实验报告参考模板	124
参考文献	126

第 1 章

绪 论

► 内容提要

本章介绍数字图像的基本概念、数字图像的离散化表示、数字图像处理及软件环境配置。

► 知识要点

- ◇ 图像、数字图像、像素及数字视频的定义。
- ◇ MATLAB 环境配置。
- ◇ 最常用的几个 MATLAB 函数命令。
- ◇ 内联函数与匿名函数。
- ◇ MATLAB GPU 编程基础。
- ◇ 数字图像处理所研究的主要内容。

► 教学建议

- ◇ 本章教学安排建议 6 课时左右。
- ◇ 先修知识主要有线性代数、数学实验(MATLAB)、计算机基础等。
- ◇ MATLAB 常用函数是全书编程的基本工具,用 2 课时左右进行复习巩固。
- ◇ 内联函数与匿名函数用 2 课时左右进行练习。
- ◇ 考虑到具体实验环境,MATLAB GPU 编程基础部分可选学,用 2 课时左右。

1.1 基本概念

人们对图像是很熟悉的,然而想对其给出一个科学的定义却不简单^[1],诸多研究者一直进行着这方面的尝试。一个关于图像的最古老的定义是柏拉图提出的,他说:“我们首先把影子称为图像,然后把人们在水中或在模糊的、光滑的和闪亮的物体表面看到的反光和所有相似的再现看成图像”^[2];图像是“对物体的表达、表象、模仿,一个生动的视觉描述,为了表

达其他事物而引入的事物”^[3]；图像是用各种观测系统以不同形式和手段观测客观世界而获得的，可以直接或间接作用于人眼并进而产生视觉和知觉的实体^[4]；图像是对客观存在物体的一种相似性的生动模仿与描述，是物体的一种不完全的、不精确的，但在某种意义上是适当的表示^[5-7]。因此，使用“图象”或许比“图像”更为合理^[4]。

图像直观、丰富的特点决定了其可承载大量信息，故有“一图胜千言”“百闻不如一见”等说法。图 1-1 所示为一幅风景图像^①，从图中可以看到天空、白云、水面、帆船等景色。尽管这幅图像展现的内容很有限，想把它们全部用文字描述清楚却十分困难。

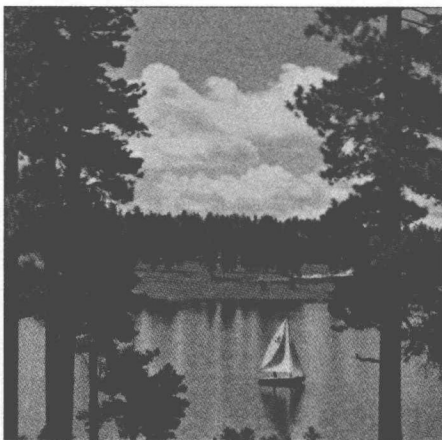


图 1-1 风景图像

根据图像的表达方式，可将其再细分为物理图像、数字图像、数字视频等。

物理图像是指物质或能量的实际分布^[6]。根据是否能被人眼所感知，又可将物理图像分为可见图像与不可见图像。可见图像是能为人类视觉所感知的通常意义下的图像。例如，光学图像的光谱波段就能被人的肉眼所感知，属可见图像。在医学诊断中使用的伽马射线波段在可见光之外，其原始的能量分布属不可见图像。最终的医学图像是将上述不可见的物理量通过可视化手段转换成人眼能够方便识别的图像形式。

数字图像是用数字阵列来表示的图像。数字阵列中的元素称为像素。

数字视频是连续播放的数字图像序列。与单幅图像相比，数字视频可看成多幅数字图像在时间轴的叠加，其中每幅图像称为数字视频的一帧。图 1-2 是一个数字视频中连续的 4 帧。为了使视频播放保持视觉上的连续性，帧与帧之间往往具有很大的信息冗余。

由于数字视频的每一帧都是静态图像，因此一些数字图像处理方法很容易应用到数字视频的处理领域。

① 除特殊说明，本书所用图像均来自 MATLAB 图像处理工具箱或南加利福尼亚大学的 USC-SIPI image database 图像库(<http://sipi.usc.edu/database/>)。



图 1-2 数字视频的帧序列示例

1.2 MATLAB 基础

MATLAB 是“Matrix Laboratory”两个词的缩写组合,意为“矩阵实验室”。是由美国 Mathworks 公司发布的主要面对科学计算、可视化以及交互式程序设计的软件系统。除基本模块外, MATLAB 还提供了许多可选工具箱,如图像处理工具箱、优化工具箱、统计工具箱、控制系统工具箱等。工具箱的使用能够大大提高用户在具体领域的编程效率,从而节省出更多的时间用于创新思维。有人把 MATLAB 称为继机器语言、汇编语言、高级语言之后的“第四代计算机编程语言”。

1.2.1 MATLAB 工具箱安装

MATLAB 中可以添加大量工具箱,这些工具箱可以是 MATLAB 官方提供的专业工具箱,也可以是第三方的共享代码。以图像处理软件包 DIPUM^[8]为例,介绍 MATLAB 工具箱的安装步骤:

第一步,将解压后的工具箱复制到 MATLAB 安装目录下的 toolbox 子目录。

第二步,在 MATLAB 运行界面的 HOME 选项卡中单击 Set Path 快捷按钮,打开 Set Path 窗口,如图 1-3 所示。

第三步,单击 Set Path 窗口中的 Add Folder 按钮,选择 DIPUM 所对应的目录,如图 1-4 所示。

第四步,依次单击上面窗口中的 Save 和 Close 按钮,完成 DIPUM 工具箱的安装。

1.2.2 几个常用的 MATLAB 命令

为方便后续内容的程序调试,几个常用的 MATLAB 命令列举如下:

“help 命令名称/程序名称”用来在命令窗口中显示相应 MATLAB 命令或文件的帮助信息。

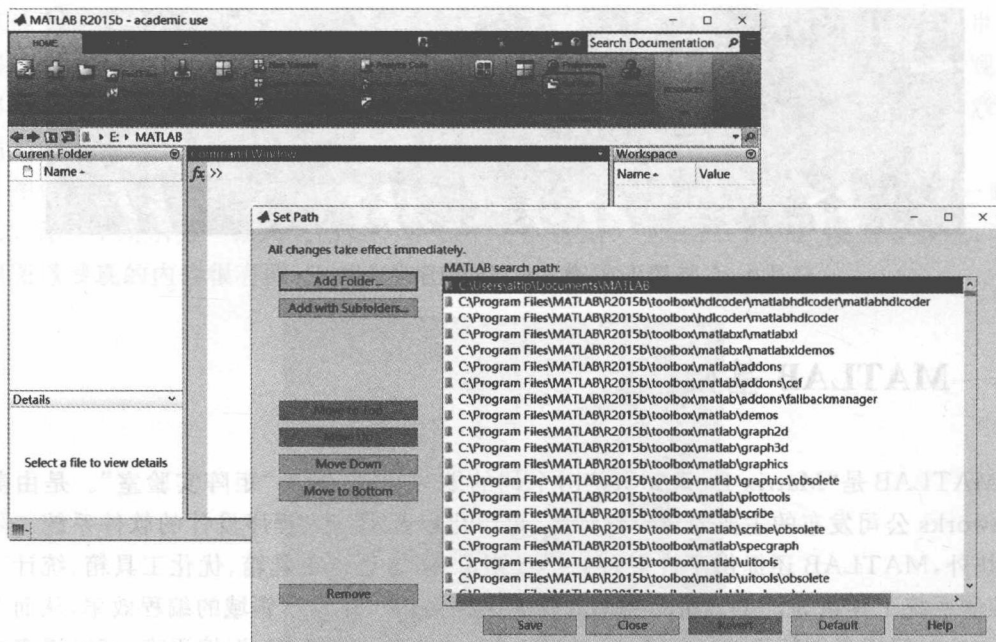


图 1-3 MATLAB 工具箱安装示例第二步

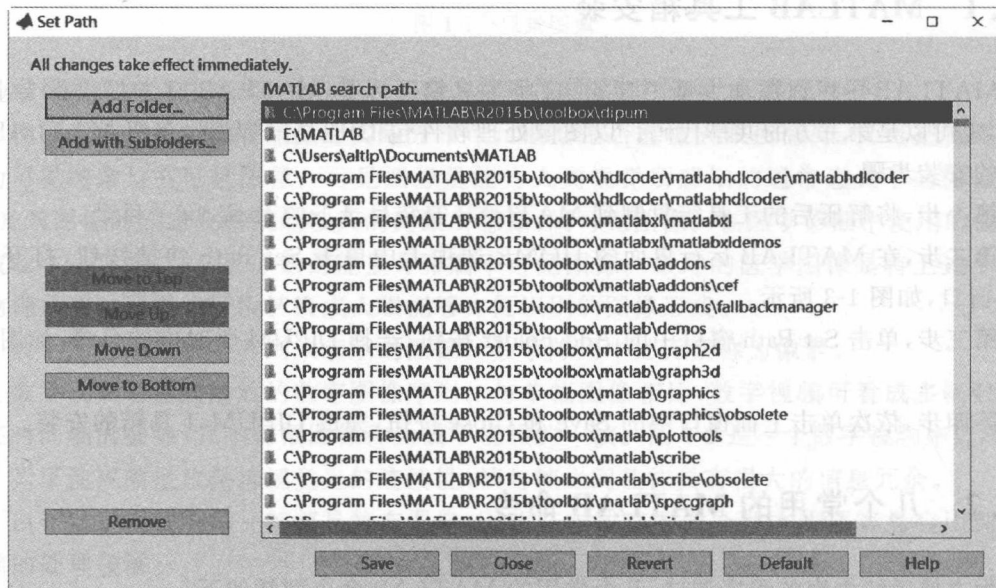


图 1-4 MATLAB 工具箱安装示例第三步

“doc 命令名称/程序名称”用来在帮助窗口中以网页形式获取 MATLAB 命令或 M 程序的帮助信息,显示风格较“help”更为丰富和灵活。若在命令窗口中只输入“doc”,则直接打开帮助网页,操作十分方便。

“type 程序名称”用于在命令窗口中显示 MATLAB 程序内容。

“edit 程序名称”用于在 MATLAB 程序编辑器中打开相应程序以进行编辑。

“pcode 程序名称”用于将 MATLAB 和 M 程序转化为扩展名为 .p 的加密格式。

“clc”命令用于清除命令窗口所显示的内容。

“clear”命令用于清除内存变量。

“cat”命令用于按指定方向连接矩阵。

“save”和“load”命令则分别用于将内存变量放入外存和将放入外存的内存变量重新读入内存。

“tic”和“toc”命令,用来记录 MATLAB 命令执行的时间。常用“tic”来保存当前时间,而后使用“toc”来记录程序完成时间。

上面命令的具体实例可通过 doc 函数获取,建议读者结合实例,加深对这些命令的掌握。

1.2.3 MATLAB 的内联函数与匿名函数

此处对 MATLAB 的内联函数和匿名函数实现方法进行介绍, MATLAB 引入这两个函数主要是为了减少程序访问外存时间,提高运算效率。

1. 内联函数

内联(inline)函数是 MATLAB 7 以前经常使用的一种构造函数对象的方法。在命令窗口、程序或函数中创建局部函数时,通过使用 inline 构造函数,而不用将其储存为一个 M 文件,同时又可以像使用一般函数那样调用它。常用语法为

```
fhandle = inline(expr, arg1, arg2, ...)
```

其中, fhandle 是调用该函数的函数句柄; expr 为函数表达式; arg_i 为函数参数。由于内联函数是储存于内存中而不是在 M 文件中,因此省去了文件访问的时间,加快了程序的运行效率。

需注意,内联函数在使用过程中还会受到一些制约。首先,不能在内联函数中调用另一个 inline 函数;其次,只能由一个 MATLAB 表达式组成,并且只能返回一个变量。

2. 匿名函数

匿名函数(anonymous function)是 MATLAB 7.0 版提出的一种全新的函数描述形式,

和内联函数类似,可以让用户编写简单的函数而不需要创建 M 文件,匿名函数具有 inline 函数的所有优点,并且还具有一些独有特点,运行效率也比 inline 函数高。定义一个匿名函数常用语法是

```
fhandle = @(arglist) expression
```

其中, fhandle 依然是相应的函数句柄; arglist 为参数列表。

例 1-1 分别采用内联函数和匿名函数定义

$$f(x, y) = x^2 + y^2$$

并测试运行。

```
>> f1 = inline('x^2 + y^2', 'x', 'y'); ①
>> f1(2,3)
ans =
    13
>> f2 = @(x,y) x^2 + y^2;
>> f2(2,3)
ans =
    13
```

匿名函数会让前面的内联函数逐步退出 MATLAB 舞台,事实上在设计这种类型的函数时就带有这一目的,目前保留内联函数无疑会使程序具有更好的向下兼容性。

1.2.4 MATLAB GPU 编程基础

GPU 英文全称为 Graphic Processing Unit,中文翻译为“图形处理器”或“图像处理单元”。与 CPU 相比,GPU 更适合大规模并发计算,在处理图像时往往具有更高的工作效率。

从 MATLAB 2013 版本开始,MATLAB 可以直接调用 GPU 进行并行计算。只要数据是 gpuArray 格式的,许多 MATLAB 内置函数就会自动地调用 GPU 进行计算。下面介绍 MATLAB 的 GPU 编程时的几个基础函数。

1. GPU 设备确认

(1) gpuDeviceCount 函数用于获取当前设备上的 GPU 数目,语法格式为:

```
n = gpuDeviceCount
```

(2) gpuDevice 函数用于选择 GPU 设备,语法格式为:

```
D = gpuDevice or gpuDevice()
```

① 也可忽略参数而写成 f1=inline('x^2+y^2');

如果当前还未设置选择的 GPU, 则选择默认的 GPU, D 是返回对象; 如果已经设置了 GPU, 则返回设置的 GPU 对象。

```
D = gpuDevice(Idx);
```

表示选择 Idx 对应的 GPU 设置(分别用 1 和 2 等表示), D 依然为返回对象。

(3) reset 函数用于清空 GPU 内存, 语法格式为:

```
reset(gpudev)
```

与 MATLAB 的 clear 功能类似, gpudev 是 gpuDevice 返回的对象。

例 1-2 获取当前计算机的 GPU 数目, 指定 GPU 设备, 并练习清空所指定 GPU 的内存。

```
>> n = gpuDeviceCount
>> D = gpuDevice
>> reset(D)
```

2. GPU 与 CPU 之间的数据交互

(1) gpuArray 用于将 CPU 内存数据传到 GPU 内存中, 语法格式为:

```
GX = gpuArray(X)
```

其中, X 为 CPU 内存变量; GX 为得到的 GPU 内存变量。

(2) gather 函数用于将 GPU 内存数据传到 CPU 内存中, 语法格式为:

```
X = gather(GX)
```

其中, GX 为 GPU 内存变量; X 为得到的 CPU 内存变量。

例 1-3 编程实现 CPU 与 GPU 之间内存变量的交互。

```
X = rand(10, 'single'); % 定义在 CPU 上的一个 10×10 的随机初始化数组
GX = gpuArray(X); % 在 GPU 初始化数组 GX, 并且将 X 的值赋给 GX
GX2 = GX .* GX; % GPU 上执行数组对应位置的点乘
CX2 = gather(GX2); % 将 GPU 上的运算结果 GX2 返回到 CPU 的 CX2
```

例 1-4 编程比较 CPU 与 GPU 对大规模计算的运行效率。

```
A1 = rand(12000, 400);
B1 = rand(400, 12000);
tic
f1 = sum(A1.' .* B1, 1);
tCPU = toc

A2 = rand(12000, 400, 'gpuArray');
B2 = rand(400, 12000, 'gpuArray');
```

```
tic
    f2 = sum(A2.' * B2, 1);
tGPU = toc
```

上面程序的运行结果为:

```
tCPU =
    0.0414
tGPU =
    3.0278e - 04
CGTime =
    136.6621
```

由程序运行结果可以看出,针对这一计算任务,在实验所用的计算机上,GPU的运算速度达到CPU的130余倍。另外需要注意,本程序在调用随机数函数rand时使用“gpuArray”参数,直接生成了GPU内存变量。

1.3 本书内容

数字图像处理就是用计算机处理数字图像,其着重强调在图像之间进行变换,即处理的对象和结果均为图像。

本书将介绍数字图像处理的一些基本内容,主要包括图像的基本操作、图像的基本运算、图像变换、图像的形态学操作、图像增强、图像去噪、图像分割等。

图像的基本操作,主要是图像的读取、显示与存储操作。

图像的基本运算,包括图像的代数运算和几何变换两部分。

图像变换是指通过数学映射的方法,将空域中的信息转换到变换域,如频域、时频域等。通过对图像变换域信息进行处理,再进行逆变换,最终获取处理后的图像。

图像的形态学操作是指采用数学形态学的仿生方法对图像进行处理。

图像增强是为了改善图像的视觉效果,突出图像中有用的部分(即所感兴趣的部分)。

图像去噪目的是将退化图像原有信息进行恢复,使之更接近实际。

需要注意,图像增强与图像去噪的目的不同,前者是为了满足用户的主观偏爱,而后者则倾向于逼近客观真相。

图像分割则是将一幅图像分成若干个有意义或感兴趣区域的过程。通过图像分割,有利于将图像中有意义的特征部分提取出来,而这些特征是进一步进行图像分析和理解的基础^[7]。

由于篇幅所限,本书主要对图像处理的最基本方法进行介绍。相信通过对本书的学习,读者在进行相关文献阅读及解决实际问题时,能有一个较为清晰的问题背景及物理直觉,有利于对问题的正确分析和最终解决。