

MATLAB
开发实例系列图书

在线交流，有问必答
(详见封底勒口)



本书程序源代码
可在MATLAB中文论坛图书版块、
北航出版社“下载专区”免费下载

MATLAB

数字图像处理

——从仿真到C/C++代码的自动生成

赵小川 赵斌 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

MATLAB 开发实例系列图书

MATLAB 数字图像处理

——从仿真到 C/C++ 代码的自动生成

赵小川 赵 斌 编著

北京航空航天大学出版社

内 容 简 介

本书以 MATLAB 8. X 汉化版为工具,深入浅出地介绍了基于计算机视觉工具箱(Computer Vision System Toolbox)的数字图像处理的基本原理、实现方法、仿真过程,着重介绍了如何将仿真代码或模型快速地转化成为 C/C++ 代码。内容包括:MATLAB 基本操作、Visual Studio2010 使用入门、基于 MATLAB Coder 的 M 代码转换为 C/C++ 代码、MATLAB 计算机视觉工具箱、图像变换的仿真及其 C/C++ 代码的自动生成、图像特征提取的仿真及其 C/C++ 代码的自动生成。全书例程丰富、步骤详尽、注释完备、图文并茂。

本书适用于对数字图像技术感兴趣、打算系统学习的读者,也可作为电子信息工程、计算机科学技术相关专业的本科生、研究生的教材,以及本科毕业设计、研究生学术论文的资料,并可供工程技术人员参考使用。

图书在版编目(CIP)数据

MATLAB 数字图像处理 / 赵小川, 赵斌编著. -- 北京 :
北京航空航天大学出版社, 2015. 8
ISBN 978-7-5124-1844-8

I. ①M… II. ①赵… ②赵… III. ①数字图像处理—
Matlab 软件 IV. ①TN911.73

中国版本图书馆 CIP 数据核字(2015)第 176021 号

版权所有,侵权必究。

MATLAB 数字图像处理

——从仿真到 C/C++ 代码的自动生成

赵小川 赵 斌 编著

责任编辑 赵延永

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: goodtextbook@126.com 邮购电话:(010)82316936

北京兴华昌盛印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:20.5 字数:538 千字

2015 年 9 月第 1 版 2015 年 9 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1844-8 定价:45.00 元

前 言

随着信息处理技术和计算机技术的飞速发展,数字图像处理技术已在工业检测、航空航天、星球探测、军事侦察、公安防暴、人机交互、文化艺术等领域受到了广泛的重视并取得了众多成就。

近年来,MATLAB 软件针对数字图像处理技术推出了诸多新功能,计算机视觉工具箱(Computer Vision System Toolbox)便是其中典型的代表。与传统的图像处理工具箱(Image Processing Toolbox)相比,计算机视觉工具箱引入了基于系统对象(System Object)及基于模型的处理模式,使其处理速度更快、交互性更强,同时该工具箱的绝大多数函数、系统对象、模型支持代码转换,可自动生成可读、可运行、可移植的 C/C++ 代码。这无疑极大地方便了广大从事数字图像处理研究的科研工作者,提高了研发效率。

本书以 MATLAB 8. X 汉化版为工具,深入浅出地介绍了基于计算机视觉工具箱(Computer Vision System Toolbox)的数字图像处理的基本原理、实现方法、仿真过程,着重介绍了如何将仿真代码或模型快速地转化成为 C/C++ 代码。与传统的手写代码相比,MATLAB 自动生成的代码具有高效、规范、可读性强的特点;同时,稍加改进便可移植到不同的硬件平台上。

本书具有如下特色:

① 本书将基本原理、仿真实现、代码转换有机结合,实现了数字图像处理从原理到实现的无缝连接。

② 根据编者近些年来从事教学、科研的经验,列举了近百个关于数字图像处理采用 MATLAB 计算机视觉工具箱实现的实例,并附有详细注解;在每个例子中,都会有运行效果图,使读者有身临其境的感觉。

③ 本书在讲解的过程中,分享了作者的一些心得(以“经验分享”的形式出现),实用性强,有利于读者快速上手。

感谢寇宇翔、李喜玉、刘祥、李阳、肖伟、梁冠豪、葛卓、郅威、孙祥溪、常青在本书的资料整理及校对过程中所付出的辛勤劳动。

本书涉及的所有源程序将放到 MATLAB 中文论坛的读者在线交流平台上,供读者自由下载。这些源程序在 MATLAB 8. X 及 Visual Studio 2010 下经过了验证,均能够正确执行,读者可将自己的 MATLAB 版本更新至 MATLAB 8. X 及其以后的版本,以避免出现不必要的问题。本书读者在线交流平台网址:<http://www.ilovematlab.cn/forum-250-1.html>。

由于作者水平有限,书中的疏漏和不当之处,恳请广大读者和同行批评指正!作者邮箱 zhaoxch1983@sina.com。本书勘误网址:<http://www.ilovematlab.cn/thread-434967-1-1.html>。

赵小川

2015 年 6 月

目 录

第 1 章 MATLAB 基本操作	1
1.1 矩阵操作与运算	1
1.1.1 在 MATLAB 中生成矩阵	1
1.1.2 矩阵变形操作	6
1.1.3 矩阵的下标引用	9
1.1.4 获取当前矩阵信息	11
1.1.5 矩阵运算	14
1.1.6 矩阵关系比较	18
1.1.7 矩阵元素值取整	19
1.1.8 对矩阵进行逻辑运算	20
1.1.9 矩阵分解	21
1.1.10 查找矩阵中的最值	21
1.1.11 查找矩阵中的元素	22
1.2 MATLAB 编程基础	23
1.2.1 变量命名规则及其类型	23
1.2.2 基本程序结构	24
1.2.3 M 文件	32
1.2.4 函数句柄与匿名函数	39
1.2.5 MATLAB 编程技巧	40
1.3 基于 Simulink 的仿真	43
1.3.1 什么是 Simulink	43
1.3.2 Simulink 模块库介绍	43
1.3.3 创建一个简单的 Simulink 示例	48
1.3.4 对模块进行基本操作	53
1.3.5 信号线的操作	54
第 2 章 Visual Studio 2010 使用入门	55
2.1 Visual Studio 2010 简介	55
2.2 安装流程	57
2.3 Visual Studio 语言	58
2.4 编写一个“HelloWorld”程序	59
2.5 访问 MSDN 论坛	60
2.6 Visual Studio 2010 中的应用程序开发	61
2.6.1 管理解决方案、项目和文件	61
2.6.2 编辑代码和资源文件	66
2.6.3 解决方案生成和调试	69

第 3 章 基于 MATLAB Coder 的 M 代码转换成 C/C++ 代码	73
3.1 启动 MATLAB Coder	73
3.2 MATLAB Coder 使用典型实例	74
3.2.1 把 M 文件转换为 C 程序代码	74
3.2.2 将生成的代码在 VS 2010 中实现	79
3.2.3 生成特定硬件可以运行的代码	84
3.2.4 通过命令实现 C 代码的生成	85
第 4 章 MATLAB 计算机视觉工具箱	89
4.1 数字图像处理基础	89
4.1.1 什么是数字图像	89
4.1.2 数字图像处理的基本概念	90
4.1.3 数字图像的矩阵表示	93
4.2 MATLAB 数字图像处理基本操作	94
4.2.1 图像文件的读取	94
4.2.2 图像文件的写入(保存)	94
4.2.3 图像文件的显示	95
4.2.4 图像文件信息的查询	96
4.2.5 MATLAB 中的图像类型	97
4.3 基于系统对象(System Object)编程	97
4.4 计算机视觉系统工具箱及其功能模块介绍	100
4.4.1 概述	100
4.4.2 各功能模块介绍	101
第 5 章 图像变换的仿真及其 C/C++ 代码的自动生成	109
5.1 图像缩放变换	109
5.1.1 基本原理	109
5.1.2 基于 System Object 的仿真	111
5.1.3 基于 Blocks - Simulink 的仿真	113
5.1.4 C/C++ 代码的自动生成及其运行效果	114
5.2 图像的平移变换	124
5.2.1 基本原理	124
5.2.2 基于 System Object 的仿真	125
5.2.3 基于 Blocks - Simulink 的仿真	126
5.2.4 C/C++ 代码自动生成及运行效果	127
5.3 图像的旋转变换	137
5.3.1 基本原理	137
5.3.2 基于 System Object 的仿真	138
5.3.3 基于 Blocks - Simulink 的仿真	140
5.3.4 C/C++ 代码自动生成及运行效果	141
5.4 图像的傅里叶变换	150
5.4.1 基本原理	150

5.4.2 基于 System Object 的仿真	152
5.4.3 基于 Blocks-Simulink 的仿真	154
5.4.4 C/C++代码自动生成及运行效果	156
5.5 图像的余弦变换	164
5.5.1 基本原理	164
5.5.2 基于 System Object 的仿真	166
5.5.3 基于 Blocks-Simulink 的仿真	167
5.5.4 C/C++代码自动生成及运行效果	169
5.6 图像腐蚀、膨胀	177
5.6.1 基本原理	177
5.6.2 基于 System Object 的仿真	179
5.6.3 基于 Blocks-Simulink 的仿真	181
5.6.4 C/C++代码自动生成及运行效果	183
5.7 图像的开运算、闭运算	201
5.7.1 基本原理	201
5.7.2 基于 System Object 的仿真	202
5.7.3 基于 Blocks-Simulink 的仿真	204
5.7.4 C/C++代码自动生成及运行效果	206
5.8 图像的中值滤波	220
5.8.1 基本原理	220
5.8.2 基于 System Object 的程序实现	220
5.8.3 基于 Blocks-Simulink 的仿真	222
5.8.4 C/C++代码自动生成及运行效果	223
5.9 图像的金字塔分解	233
5.9.1 基本原理	233
5.9.2 基于 System Object 的仿真	234
5.9.3 基于 Blocks-Simulink 的仿真	235
5.9.4 C/C++代码自动生成及运行效果	237
第 6 章 图像特征提取的仿真及其 C/C++代码的生成	247
6.1 图像的灰度直方图	247
6.1.1 基本原理	247
6.1.2 基于 System Object 的仿真	247
6.1.3 基于 Blocks-Simulink 的仿真	249
6.1.4 C/C++代码自动生成及运行效果	249
6.2 图像的色彩空间	258
6.2.1 常见的色彩空间	258
6.2.2 基于 System Object 的仿真	262
6.2.3 基于 Blocks-Simulink 的仿真	263
6.2.4 C/C++代码自动生成及运行效果	265
6.3 图像的角点检测	275

6.3.1	角点检测的基本原理	275
6.3.2	基于 System Object 的仿真	279
6.3.3	基于 Blocks - Simulink 的仿真	280
6.3.4	C/C++ 代码自动生成及运行效果	283
6.4	图像的边缘检测	291
6.4.1	基本原理	291
6.4.2	基于 System Object 的仿真	295
6.4.3	基于 Blocks - Simulink 的仿真	297
6.4.4	C/C++ 代码自动生成及运行效果	299
6.5	图像的信噪比	307
6.5.1	基本原理	307
6.5.2	基于 System Object 的仿真	307
6.5.3	基于 Blocks - Simulink 的仿真	308
6.5.4	C/C++ 代码自动生成及运行效果	309
兴趣 · 尝试 · 总结——浅谈学习 Computer Vision System Toolbox 心得		317
参考文献		320

若您对此书内容有任何疑问，可以凭在线交流卡登录 MATLAB 中文论坛与作者交流。

MATLAB 是 Matrix Laboratory 的简称,是一种用于数值计算、可视化及编程的高级语言和交互式环境。借助其语言、工具和内置数学函数,可以探求多种方法,比电子表格或传统编程语言(如 C/C++ 或 Java™)更快地求取结果。MATLAB 应用范围十分广泛,包括信号处理和通信、图像和视频处理、控制系统、测试和测量、计算金融学及计算生物学等众多应用领域。在各行业和学术机构中,有人数众多的工程师和科学家使用 MATLAB 这一技术计算语言。

1.1 矩阵操作与运算

在数学上,由 $m \times n$ 个数 x_{ij} ($i=1,2,3,\dots,m; j=1,2,3,\dots,n$) 排列而成 m 行 n 列的数表定义为 m 行 n 列矩阵。

在 MATLAB 中,矩阵类型的数据是以数组的形式存在,属于一种有序的数据组织形式。因此,在 MATLAB 中,矩阵是数组的一种表现形式。一维数组可看作向量,而二维或更高维度的数组可看作矩阵。

1.1.1 在 MATLAB 中生成矩阵

一般而言,在 MATLAB 中创建矩阵的方式有两种:一种是与枚举式直接赋值法相似,直接使用赋值语句对枚举矩阵的每个元素进行赋值;另一种则是 MATLAB 库函数中提供的创建特殊矩阵的基本指令。

1. 生成数值矩阵

(1) 实数值矩阵输入

任何矩阵(向量)都可以直接按行方式输入每个元素:同一行中的元素用逗号(,)或者用空格符来分隔,且空格个数不限;不同的行用分号(;)分隔;所有元素处于一方括号([])内;当矩阵是多维(三维以上),且方括号内的元素是维数较低的矩阵时,会有多重的方括号。图 1.1.1 和图 1.1.2 展示了如何输入一个矩阵。

```
>> a=[1,2,3] %输入完这一行,按回车键可得下面的结果

a =

     1     2     3
```

图 1.1.1 输入一个行矩阵

经验分享:输入矩阵时,逗号和分号应该在“半角”及“英文标点”格式下输入,否则会提示出错,如图 1.1.3 所示。

```
>> Matrix_B = [1 2 3;%可以在此行输入完后按回车键，再输入其他行
2 3 4;3 4 5] %此行输入完以后，按回车键可得下面的结果

Matrix_B =

     1     2     3
     2     3     4
     3     4     5
```

图 1.1.2 输入一个 3×3 矩阵

命令行窗口

```
>> A=[1 2; 4 5]
A=[1 2; 4 5]
|
错误：输入字符不是 MATLAB 语句或表达式中的有效字符。
```

图 1.1.3 MATLAB 提示输入字符出错

(2) 向量输入

由于向量可以看作是一维矩阵，所以生成矩阵的方法同样适用于生成向量。向量也可以通过如下方式生成：

$X = X_0:STEP:X_n$ % 产生以 X_0 为初始值，步长为 $STEP$ ，终值不超过 X_n 的数值型向量

经验分享：

- X_n 是该向量的最后一个分量的界限，不一定是最后一个数。
- 当 $X_0 < X_n$ 时，要求 $STEP > 0$ ；当 $X_0 > X_n$ 时，要求 $STEP < 0$ 。
- $STEP$ 省略时， $STEP = 1$ 。

如图 1.1.4 所示。

```
命令行窗口
>> a=1:3:14

a =

     1     4     7    10    13

>> b=20:-1:15.5

b =

    20    19    18    17    16

>> c=20:2:15.5

c =

Empty matrix: 1-by-0

>> d=1:pi^2

d =

     1     2     3     4     5     6     7     8     9
```

图 1.1.4 向量生成实例

若您对此书内容有任何疑问，可以凭在线交流卡登录 MATLAB 中文论坛与作者交流。

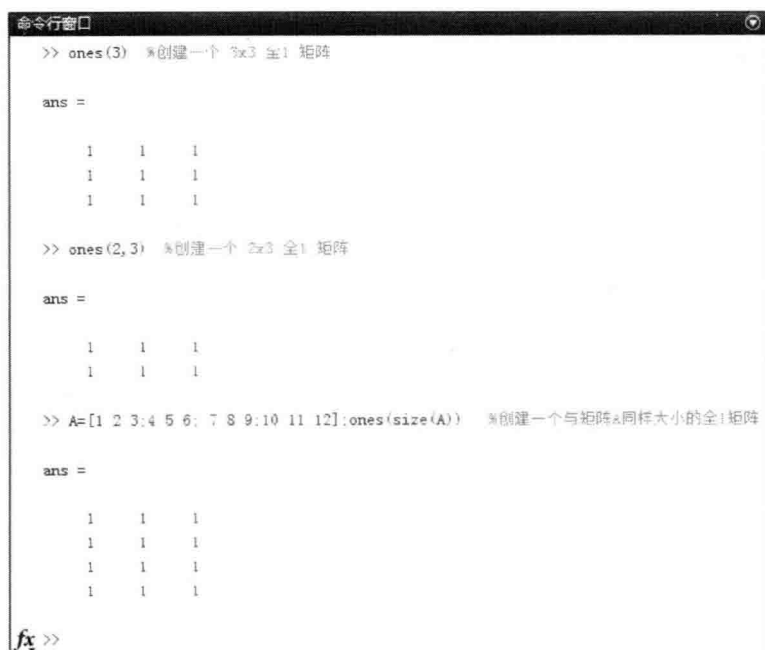
(3) 利用函数生成矩阵

特殊矩阵的生成函数如表 1.1-1 所列。

表 1.1-1 特殊矩阵的生成函数

函数名称	功能说明
ones(n)	创建一个 $n \times n$ 的全 1(元素)矩阵
ones(m,n,...,p)	创建一个 $m \times n \times \dots \times p$ 的全 1(元素)矩阵
ones(size(A))	创建一个与矩阵 A 同样大小的全 1(元素)矩阵
zeros(n)	创建一个 $n \times n$ 的全 0(元素)矩阵
zeros(m,n,...,p)	创建一个 $m \times n \times \dots \times p$ 的全 0(元素)矩阵
zeros(size(A))	创建一个与矩阵 A 同样大小的全 0(元素)矩阵
eye(n)	创建一个 $n \times n$ 的单位矩阵
eye(m,n)	创建一个 $m \times n$ 的单位矩阵
eye(size(A))	创建一个与矩阵 A 同样大小的单位矩阵
magic(n)	创建一个 $n \times n$ 的矩阵,其每一行、每一列的元素之和都相等
rand(n)	创建一个 $n \times n$ 的随机数矩阵,其元素为 0~1 之间均匀分布的随机数
randn(n)	创建一个 $n \times n$ 的正态分布随机数矩阵,其元素是零均值,单位方差的正态分布随机数
diag(x)	创建一个 n 维的对角方阵,它的主对角线元素值取自向量 x ,其余元素的值都为 0
triu(A)	创建一个与矩阵 A 大小相同的上三角矩阵,该矩阵的主对角线上的元素为 A 中相应的元素,其余元素为 0
tril(A)	创建一个与矩阵 A 大小相同的下三角矩阵,该矩阵主对角线上的元素为 A 中相应的元素,其余元素为 0

图 1.1.5 为部分特殊矩阵生成函数的使用实例。



```

命令窗口
>> ones(3) %创建一个 3x3 全1 矩阵

ans =

     1     1     1
     1     1     1
     1     1     1

>> ones(2,3) %创建一个 2x3 全1 矩阵

ans =

     1     1     1
     1     1     1

>> A=[1 2 3;4 5 6; 7 8 9;10 11 12];ones(size(A)) %创建一个与矩阵A同样大小的全1矩阵

ans =

     1     1     1
     1     1     1
     1     1     1
     1     1     1

fx >>
  
```

(a)

图 1.1.5 部分特殊矩阵生成函数的使用实例

若您对此书内容有任何疑问，可以凭在线交流卡登录 MATLAB 中文论坛与作者交流。

4

```

命令窗口
>> zeros(4) %创建一个 4x4 全0 矩阵

ans =

    0    0    0    0
    0    0    0    0
    0    0    0    0
    0    0    0    0

>> zeros(4,3) %创建一个 4x3 全0 矩阵

ans =

    0    0    0
    0    0    0
    0    0    0
    0    0    0

>> A=[1 2 3;4 5 6; 7 8 9;10 11 12];zeros(size(A)) %创建一个与矩阵A同样大小的全0矩阵

ans =

    0    0    0
    0    0    0
    0    0    0
    0    0    0

fx >>
    
```

(b)

```

命令窗口
>> eye(3) %创建一个 3x3 的单位矩阵

ans =

    1    0    0
    0    1    0
    0    0    1

>> magic(4) %创建一个 4x4 的 魔方阵

ans =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

>> rand(3) %创建一个 3x3 的随机数矩阵

ans =

    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575

fx >>
    
```

(c)

图 1.1.5 部分特殊矩阵生成函数的使用实例 (续)

```

命令窗口
>> randn(3) %创建一个 3x3 零均值，单位方差的 正态分布随机数 矩阵

ans =

    1.4090   -1.2075    0.4889
    1.4172    0.7172    1.0347
    0.6715    1.6302    0.7269

>> diag(1:2:8) %创建一个 以向量x=1:2:8为对角元素 的n维对角方阵

ans =

     1     0     0     0
     0     3     0     0
     0     0     5     0
     0     0     0     7

>> A=[1 2 3;2 3 4; 3 4 5];triu(A) %创建一个与矩阵A同大小的 上三角矩阵

ans =

     1     2     3
     0     3     4
     0     0     5

>> tril(A) %创建一个与矩阵A同大小的 下三角矩阵

ans =

     1     0     0
     2     3     0
     3     4     5

fx >>
  
```

(d)

图 1.1.5 部分特殊矩阵生成函数的使用实例(续)

经验分享:对于同一个特殊矩阵生成函数而言,其存在不同的版本。不同版本的主要区别在于输入参数的不同。因此,为更好地使用这些特殊矩阵生成函数,用户通过 help 命令或查看 MATLAB 的帮助目录可以更加详细深入地了解这些特殊矩阵生成函数。

(4) 创建多维数组

方法 1:直接赋值法,如图 1.1.6 所示。

方法 2:调用 cat 函数。

函数:cat

格式: $A = \text{cat}(n, A_1, A_2, \dots, A_m)$

说明: $n=1$ 和 $n=2$ 时分别构造 $[A_1; A_2]$ 和 $[A_1, A_2]$, 都是二维数组, 而 $n=3$ 时可以构造出三维数组。

使用 cat 函数生成多维矩阵,如图 1.1.7 所示。

```

命令窗口
>> A1=[1,2,3;4,5,6;7,8,9];A2=A1';A3=A1-A2;
A5(:, :, 1)=A1; A5(:, :, 2)=A2; A5(:, :, 3)=A3;
>> A5

A5(:, :, 1) =

     1     2     3
     4     5     6
     7     8     9

A5(:, :, 2) =

     1     4     7
     2     5     8
     3     6     9

A5(:, :, 3) =

     0    -2    -4
     2     0    -2
     4     2     0
    
```

图 1.1.6 采用直接赋值法生成多维矩阵

```

命令窗口
>> A1=[1,2,3;4,5,6;7,8,9];A2=A1';A3=A1-A2;
>> A4=cat(3,A1,A2,A3)

A4(:, :, 1) =

     1     2     3
     4     5     6
     7     8     9

A4(:, :, 2) =

     1     4     7
     2     5     8
     3     6     9

A4(:, :, 3) =

     0    -2    -4
     2     0    -2
     4     2     0
    
```

图 1.1.7 使用 cat 函数生成多维矩阵实例

2. 如何生成复数矩阵

复数矩阵有两种生成方式,如图 1.1.8 和图 1.1.9 所示。

```

命令窗口
>> a=2.7;b=13/25;
C=[1,2*a+i*b,b*sqrt(a): sin(pi/4),a+5*b,3.5+1]

C =

    1.0000 + 0.0000i    5.4000 + 0.5200i    0.8544 + 0.0000i
    0.7071 + 0.0000i    5.3000 + 0.0000i    4.5000 + 0.0000i
    
```

图 1.1.8 复数矩阵的第一种生成方式

```

命令窗口
>> R=[1 2 3;4 5 6]; M=[11 12 13;14 15 16];
>> CN=R+i*M

CN =

    1.0000 +11.0000i    2.0000 +12.0000i    3.0000 +13.0000i
    4.0000 +14.0000i    5.0000 +15.0000i    6.0000 +16.0000i
    
```

图 1.1.9 复数矩阵的第二种生成方式

1.1.2 矩阵变形操作

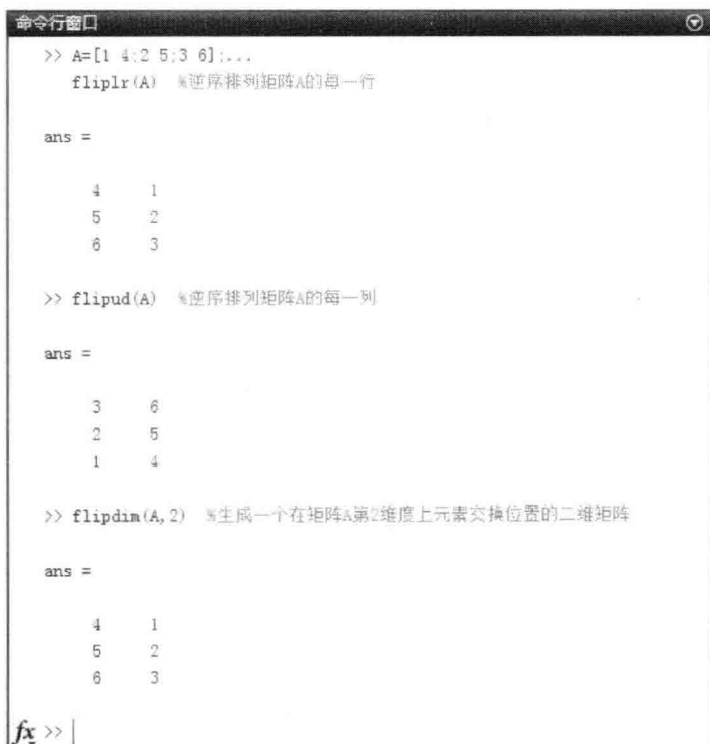
在具体的矩阵运算过程中,用户可能遇到需要改变矩阵形态的情况,包括改变矩阵的大小,甚至结构。具体来说,矩阵的变形主要有矩阵的旋转、矩阵维度的修改与矩阵元素的删除

若您对此书内容有任何疑问,可以凭在线交流卡登录 MATLAB 中文论坛与作者交流。

等。MATLAB 提供了一系列可以改变矩阵大小与结构的库函数,见表 1.1-2。图 1.1.10 所示是几个矩阵变形操作的库函数演示。

表 1.1-2 矩阵变形操作的库函数

函数名称	功能说明
fliplr(A)	逆序排列矩阵 A 的每一行
flipud(A)	逆序排列矩阵 A 的每一列
flipdim(A,dim)	生成一个在第 dim 维矩阵 A 内的元素交换位置的多维矩阵
rot90(A)	生成一个由矩阵 A 逆时针旋转 90 度而得到的新矩阵
reshape(A,m,n)	生成一个 $m \times n$ 的矩阵,其元素以线性索引的方式从矩阵 A 中顺序取得
repmat(A,n)	以矩阵 A 为基本模板,生成一个以矩阵 A 为块矩阵的新矩阵,新矩阵维数是 $\text{size}(A) \times n$
shiftdim(A,n)	矩阵 A 的维数移动 n 步。(n 为正数,则左移;n 为负数,则右移)
cat(dim,A,B)	将矩阵 A 与矩阵 B 在第 dim 个维度上(行方向、列方向或第 dim 维方向上)拼接成新的矩阵
permute(A,order)	根据向量 order 来改变矩阵 A 中的维数顺序
ipermute(A,order)	进行 permute 命令的逆变换
sort(A)	对一维或二维矩阵进行升序排列,并返回排序后的矩阵。当 A 是二维矩阵时,对矩阵 A 的每一列分别进行排序



```

命令窗口
>> A=[1 4;2 5;3 6];...
      fliplr(A)  %逆序排列矩阵A的每一行

ans =

     4     1
     5     2
     6     3

>> flipud(A)  %逆序排列矩阵A的每一列

ans =

     3     6
     2     5
     1     4

>> flipdim(A,2)  %生成一个在矩阵A第2维度上元素交换位置的二维矩阵

ans =

     4     1
     5     2
     6     3
  
```

(a)

图 1.1.10 矩阵变形操作的库函数使用演示

若您对此书内容有任何疑问，可以凭在线交流卡登录 MATLAB 中文论坛与作者交流。

8

```

命令窗口
>> rot90(A) %将矩阵A 逆时针 旋转90度生成新矩阵

ans =

     4     5     6
     1     2     3

>> reshape(A,2,3) %将矩阵A中的元素以 线性索引 方式重新组成一个2x3新矩阵

ans =

     1     3     5
     2     4     6

>> repmat(A,2) %以矩阵A为一个块矩阵，生成以块为一个基本单元的新矩阵

ans =

     1     4     1     4
     2     5     2     5
     3     6     3     6
     1     4     1     4
     2     5     2     5
     3     6     3     6

fx >>

```

(b)

```

命令窗口
>> shiftdim(A,1) %矩阵A (3x2维)的维数左移1步，则得2x3维新矩阵

ans =

     1     2     3
     4     5     6

>> B=rand(3,2);cat(2,A,B) %矩阵B是一个3x2随机数矩阵，在第2维度，即行的方向上拼接成新矩阵

ans =

     1.0000     4.0000     0.7922     0.0357
     2.0000     5.0000     0.9595     0.8491
     3.0000     6.0000     0.6557     0.9340

>> permute(A,[2 1]) %用向量[2 1]表示互换矩阵A的第一维度与第二维度，输出效果为矩阵A的转置

ans =

     1     2     3
     4     5     6

>> C=[3 1.2 4;6 5];sort(C) %将矩阵C的每一个列按降序进行排列

ans =

     2     1
     3     4
     6     5

fx >>

```

(c)

图 1.1.10 矩阵变形操作的库函数使用演示(续)

经验分享:表 1.1-2 中的 reshape 函数功能说明里提到的“线性索引”方式指的是，在 MATLAB 中，矩阵中的元素沿着从左到右，从上到下的列的方向依次编号，以这种编号方式寻访矩阵中每个元素的方式。此外，在使用 cat 函数的时候，必须确保矩阵 A 与矩阵 B 在拼接的维度 dim 上具有相同的长度，否则将产生错误。

1.1.3 矩阵的下标引用

元素操作是 MATLAB 矩阵操作的重要组成部分。下标引用为元素操作提供了必要的途径。在 MATLAB 中，一般二维矩阵元素的数字索引方式有单下标索引和双下标索引。单下标索引方式是沿着矩阵列方向的，采用列元素优先的原则，自左向右，自上而下地为矩阵中的每个元素设定单下标索引值，如图 1.1.11 所示。双下标索引方式是通过二元数对与二维矩阵元素在矩阵中的行列位置的对应关系对元素进行索引，如图 1.1.12 所示。MATLAB 提供了基于这两种矩阵元素寻访方式的索引表达式，如表 1.1-3 所列。图 1.1.13 是几个寻访矩阵元素的索引表达式使用演示。

$$A = \begin{bmatrix} 1_{(1)} & 2_{(2)} & 3_{(3)} \\ 4_{(4)} & 5_{(5)} & 6_{(6)} \\ 7_{(7)} & 8_{(8)} & 9_{(9)} \end{bmatrix}$$

图 1.1.11 单下标索引值排布方式

$$A = \begin{bmatrix} 1_{(1,1)} & 2_{(1,2)} & 3_{(1,3)} \\ 4_{(2,1)} & 5_{(2,2)} & 6_{(2,3)} \\ 7_{(3,1)} & 8_{(3,2)} & 9_{(3,3)} \end{bmatrix}$$

图 1.1.12 双下标索引值排布方式

表 1.1-3 寻访矩阵元素的索引表达式

索引表达式	功能说明
$A(n)$	按照单下标索引方式返回矩阵 A 的单下标为 n 的元素
$A(:, n)$	返回二维矩阵 A 的第 n 列列向量
$A(i, :)$	返回二维矩阵 A 的第 i 行行向量
$A(:, m:n)$	返回二维矩阵 A 从第 m 列到第 n 列所有向量构成的子矩阵
$A(i:j, :)$	返回二维矩阵 A 从第 i 行到第 j 行所有向量构成的子矩阵
$A(i:j, m:n)$	返回二维矩阵 A 从第 i 行到第 j 行与从第 m 列到第 n 列所有向量交集构成的子矩阵
$A(:)$	返回一个由矩阵 A 所有列向量依次拼成的列向量
$A(i:j)$	返回一个由 $A(:)$ 中第 i 到第 j 个元素构成的行向量
$A(i_1:i_2)$	返回一个由 $A(:)$ 中第 i_1, i_2 等元素构成的行向量
$A(:, [m_1 m_2 I])$	返回一个由矩阵 A 中第 m_1, m_2 等列向量构成的子矩阵
$A([i_1 i_2 I], :)$	返回一个由矩阵 A 中第 i_1, i_2 等行向量构成的子矩阵
$A([i_1 i_2 I], [m_1 m_2 I])$	返回一个由矩阵 A 中第 i_1, i_2 等行向量与第 m_1, m_2 等列向量交集元素构成的子矩阵

经验分享:在图 1.1.13 中，示例 $A([1\ 2\ 3], \text{end})$ 中的 end 是 MATLAB 中的一个关键字，用于表示该维中的最后一个元素，在该示例中表示最后一列。读者可自行构建一个 $m \times n$ 矩阵 A 并在命令行窗口中分别输入 $A(1:m, 2:n)$ 与 $A(:, 2:\text{end})$ ，比较二者输出结果，从而体会关键字 end 的含义。