

数字图像处理

第一讲 Matlab速成

王伟强

中国科学院大学计算机科学与技术学院

内容大纲

- Matlab相关背景
- Matlab工作环境
- Matlab和图像处理工具箱
- Matlab编程摘要

初识Matlab?

- Matlab 意即矩阵实验室(**matrix laboratory**), 该软件由 **MathWorks** 公司开发。(<http://www.mathworks.com/>)
- MATLAB是一个以**数组**为基本数据元素的**交互式系统**。
- 它是一种用于**技术计算**的**高性能语言**
 - 这允许为许多技术计算问题, 特别是那些涉及矩阵表示的问题, 制定解决方案。
 - 相比标量非交互式语言(如C或Fortran), 编写MATLAB程序所需的时间更少。
- 它将**计算、可视化和编程**集成在一个易于使用的环境中。
- 它是**高校**(数学、工程和科学)和**企业**(研究、开发和分析)人员的一种强有力的算法开发工具。

初识Matlab?

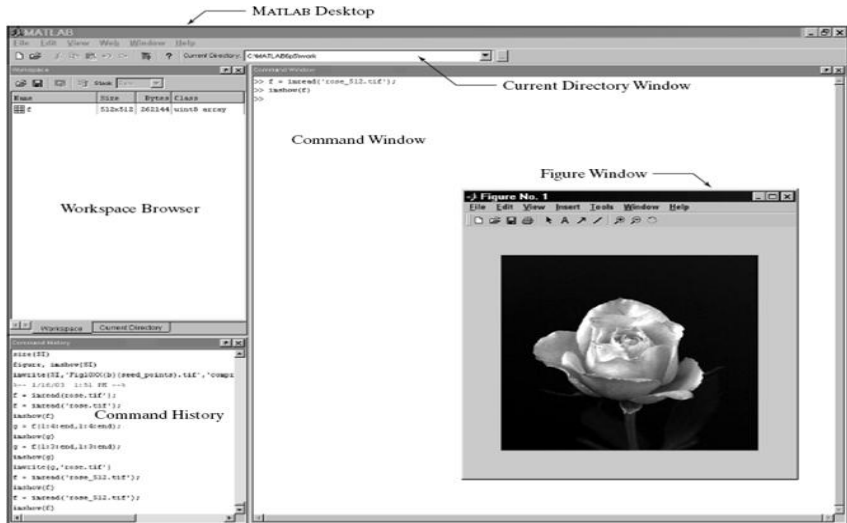
➤ Matlab的**典型应用**包括:

- 数学和计算
- 算法的开发
- 数据采集
- 建模、仿真和原型
- 数据分析、探索和可视化
- 科学与工程制图
- 应用程序开发, 包括图形用户界面构建

➤ **MATLAB**包含一系列用于各种应用程序的工具箱:

- 图像处理工具箱(IPT):扩展**MATLAB**环境用于解决数字图像处理问题
- 其他用来补充IPT的工具箱还有信号处理、神经网络、模糊逻辑和小波工具箱

Matlab 工作界面



配置环境：当前目录和搜索路径

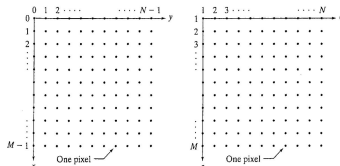
- 在MATLAB中运行的任何文件都必须位于当前目录或搜索路径上的目录中
- 默认情况下，搜索路径中包含MATLAB和MathWorks工具箱提供的文件
- 添加或修改搜索路径，是从桌面上的“文件”菜单中选择“设置路径”

使用MATLAB编辑器来创建.m文件

- MATLAB编辑器是一个专门用于创建m文件的文本编辑器，也是一个图形化MATLAB调试器。
- 源文件由扩展名.m表示
- 编辑器执行一些简单的检查，并使用颜色来区分不同的代码元素
- 文件必须在当前目录中，或在搜索路径中的目录中。

数字图像的表达

➤ 坐标约定



➤ 图像的矩阵存储形式

$$f(x, y) = \begin{pmatrix} f(0, 0), & f(0, 1), & \cdots, & f(0, N-1) \\ f(1, 0), & f(1, 1), & \cdots, & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0), & f(M-1, 1), & \cdots, & f(M-1, N-1) \end{pmatrix}$$

➤ Matlab中图像的矩阵存储形式

$$f = \begin{pmatrix} f(1, 1), & f(1, 2), & \cdots, & f(1, N) \\ f(2, 1), & f(2, 2), & \cdots, & f(2, N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M, 1), & f(M, 2), & \cdots, & f(M, N) \end{pmatrix}$$

读取一幅图像

➤ 图像存储格式

Format Name	Description	Recognized Extensions
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format [†]	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

[†] GIF is supported by `imread`, but not by `imwrite`.

➤ 使用MATLAB IPT读取一幅图像

```
f=imread('bird.jpg');
```

- `Imread`从当前目录读取文件，如果失败，它将尝试在搜索路径中找到该文件
- 请注意分号(;). 如果把它擦掉,命令窗口将显示图像的矩阵值

获取图像的信息

- `size(f)`

```
ans = 230 352 3
```

- `whos f`

```
Name Size Bytes Class
```

```
f 230x352x3 242880 uint8 array
```

```
Grand total is 242880 elements using 242880 bytes
```

- `imfinfo('bird.jpg')` show more info than `whos`

```
ans = Filename: 'bird.jpg'
```

```
FileModDate: '25-Sep-2002 19:00:16'
```

```
FileSize: 7028
```

```
Format: 'jpg'
```

```
FormatVersion: "
```

```
Width: 352
```

```
Height: 230
```

```
BitDepth: 24
```

```
ColorType: 'truecolor'
```

```
FormatSignature: "
```

```
NumberOfSamples: 3
```

```
CodingMethod: 'Huffman'
```

```
CodingProcess: 'Sequential'
```

```
Comment: {'ACD Systems Digital Imaging'}
```

显示或写入图像

- `imshow(f)`
- `imshow(f, G)`
- `imshow(f,[low,high])`
- `imshow(f,[])`

示例程序 2.1

- `imwrite(f,'filename')`
- `imwrite(f,'filename','ext')`
- `imwrite(f,'filename','quality',q)`

质量指标 q 的范围是0到100，值越低，图像质量越差，压缩率越高

图像类型

➤ 强度图像/灰度图像

- 灰度图像是一个数据矩阵，它的值被缩放以表示强度。
- 当灰度图像的像素是uint8或uint16类型时，它们的取值是 $[0, 255]$ 或 $[0, 65535]$ 范围内的整数。
- 如果图像是double类型的，则像素值为浮点数。按照惯例，经过缩放之后double类型的灰度图像的像素值落在 $[0, 1]$ 范围内

➤ 二进制图像

- 二进制图像是由0和1组成的逻辑阵列。
- 数值阵列使用逻辑函数转换为二进制。
- 如果A包含0和1以外的元素，则使用逻辑函数将所有非零数值转换为逻辑真(1)，并将所有值为0的项转换为逻辑假(0)

➤ 索引图像

➤ RGB图像

索引图像和RGB图像

- 图像可以是索引图像的，也可以是真颜色图像。
- 索引图像将颜色作为索引数组存储到`colormap`图中。
- 真彩色图像不使用`colormap`；相反，每个像素的颜色值直接存储为RGB三元组。
- 在MATLAB中，真彩色图像对象的`CData`属性是一个三维($m \times n \times 3$)数组。这个数组由三个 $m \times n$ 矩阵组成，它们沿着第三维连接在一起。

Image Type	Double-Precision Data (double Array)	8-Bit Data (uint8 Array) 16-Bit Data (uint16 Array)
Indexed (colormap)	Image is stored as a two-dimensional (m-by-n) array of integers in the range [1, length(colormap)]; colormap is an m-by-3 array of floating-point values in the range [0, 1].	Image is stored as a two-dimensional (m-by-n) array of integers in the range [0, 255] (uint8) or [0, 65535] (uint16); colormap is an m-by-3 array of floating-point values in the range [0, 1].
True color (RGB)	Image is stored as a three-dimensional (m-by-n-by-3) array of floating-point values in the range [0, 1].	Image is stored as a three-dimensional (m-by-n-by-3) array of integers in the range [0, 255] (uint8) or [0, 65535] (uint16).

数据类型

Name	Description
double	Double-precision, floating-point numbers in the approximate range -10^{308} to 10^{308} (8 bytes per element).
uint8	Unsigned 8-bit integers in the range [0, 255] (1 byte per element).
uint16	Unsigned 16-bit integers in the range [0, 65535] (2 bytes per element).
uint32	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element).
int8	Signed 8-bit integers in the range [-128, 127] (1 byte per element).
int16	Signed 16-bit integers in the range [-32768, 32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648, 2147483647] (4 bytes per element).
single	Single-precision floating-point numbers with values in the approximate range -10^{38} to 10^{38} (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

数据类型和图像类型之间的转换

- 通常我们将图像称为“数据-图像类型”的图像。

例如“unit8强度图像”仅指代像素值是unit8类型的强度图像。

- 数据类之间的转换很简单。一般语法是：

$$B = \text{data classname}(A)$$

其中数据类名是数据类表第一列中的名称之一。例如，假设A是一个uint8类型的数组。精度更高的double类型数组B是由命令 $B = \text{double}(A)$ 生成的。这种转换在整本书中都很常见，因为MATLAB希望数值计算中的操作数是双精度浮点数。

数据类之间的类型转换

- 对于uint8, logical数据类型, 类型转换可以轻松地无损实现。
- 对于double类型, 将其转换为任何目标类型时, 任何超过目标类型表示范围的取值将由对应的边界值替代。
- 如果将double类型的矩阵转换为整型, 所有的小数转换为整数时都将忽略小数部分
- 如果将double类型的矩阵转换为逻辑类型的矩阵, 非0的数值型数据转换成逻辑真, 即1; 等于0的数值转换成逻辑假, 即0。

数据类型和图像类型之间的转换

- 工具箱提供了特定的函数进行尺度放缩，为图像数据类型转换做准备。
- IPT中的下列函数可用于进行图像中数据类型的转换。

Name	Converts Input to:	Valid Input Image Data Classes
im2uint8	uint8	logical, uint8, uint16, and double
im2uint16	uint16	logical, uint8, uint16, and double
mat2gray	double (in range [0, 1])	double
im2double	double	logical, uint8, uint16, and double
im2bw	logical	uint8, uint16, and double

数据类型和图像类型之间的转换

- 函数 `im2uint8` 可以检测输入的数据类型，并为工具箱执行所有必要的缩放以将数据识别为有效的图像数据。
- 例如，考虑下面的 2×2 大小的`double`类型图像，这可能是一个中间计算的结果

```
f= -0.5 0.5  
    0.75 1.5  
G=im2uint8(f)  
G=  
    0 128  
   191 255
```

数据类型和图像类型之间的转换

- 若需要将任意double类型的矩阵转换为放缩到[0,1]区间的double类型矩阵，可以使用mat2gray函数实现该功能，其基本语法为：

$G = \text{mat2gray}(A, [A_{\min}, A_{\max}])$

其中图像G中像素取值在区间[0,1]之间，0代表纯黑，1代表纯白。参数Amin和Amax的存在使得A中所有小于等于Amin值的像素值变成0，所有大于等于Amax的像素值变成1。如果不显式给出Amax和Amin，

$G = \text{mat2gray}(A)$

则Amax和Amin会被设置为A中实际最大和最小的像素值。

数据类型和图像类型之间的转换

➤ 例如:

```
A=[128,300;-12,66.98];
```

```
G=mat2gray(A,[0,255])
```

```
G=
```

```
0.5020 1.0000  
0 0.2627
```

```
G=mat2gray(A)
```

```
G=
```

```
0.4487 1.0000  
0 0.2531
```

➤ 数组索引

为了方便数组处理和提高编程效率，MATLAB提供了一系列的强有力的索引组合：

1. 向量索引
2. 矩阵索引
3. 常用函数

➤ 向量索引

$1 \times N$ 的数组被称为行向量。

- $v=[1\ 2\ 3\ 4]$
- $w=v'$
- $w(1:end)$
- $\text{zeros}(1,4)$

➤ 矩阵索引

矩阵在MATLAB中可以很方便地表示：将一系列行向量用方括号括起来，行向量之间用逗号隔开。

- $A=[1\ 2\ 3;4\ 5\ 6]$
- $A=\text{rand}(4,4)$
- $A=\text{magic}(5)$
- $A=5*\text{ones}(3,3)$
- $A(1,2)$
- $A(:,2:4)$
- $\text{sum}(A)$ 与 $\text{sum}(A(:))$ 不一样
- $\text{mean}(A)$ 和 $\text{mean}(A(:))$
- $\text{max}(A)$ 和 $\text{max}(A(:))$ 或者 $\text{max}(\text{max}(A))$
- $\text{min}(A)$ 和 $\text{min}(A(:))$ 或者 $\text{min}(\text{min}(A))$

常用函数

➤ 将3x4的矩阵调整为2x6矩阵:

```
A=  1  4  7 10  
    2  5  8 11  
    3  6  9 12  
B = reshape(A,2, 6)  
B =  
  
    1  3  5  7  9 11  
    2  4  6  8 10 12
```

M-Function编程简介

➤ 操作符

- `+, -, *, ./, /, \, ^, .^,`
- `xor, any, all`
- 系列函数, 如 `iscell, ischar`
- `imadd, imsubtract, immultiply, imdivide`

Operator	Name	MATLAB Function	Comments and Examples
+	Array and matrix addition	plus(A, B)	$a + b$, $A + B$, or $a + A$.
-	Array and matrix subtraction	minus(A, B)	$a - b$, $A - B$, $A - a$, or $a - A$.
.*	Array multiplication	times(A, B)	$C = A .* B$, $C(I, J) = A(I, J) * B(I, J)$.
*	Matrix multiplication	mtimes(A, B)	$A * B$, standard matrix multiplication, or $a * A$, multiplication of a scalar times all elements of A .
./	Array right division	rdivide(A, B)	$C = A ./ B$, $C(I, J) = A(I, J) / B(I, J)$.
.\	Array left division	ldivide(A, B)	$C = A .\ B$, $C(I, J) = B(I, J) / A(I, J)$.
/	Matrix right division	mrdivide(A, B)	A / B is roughly the same as $A * \text{inv}(B)$, depending on computational accuracy.
\	Matrix left division	mldivide(A, B)	$A \backslash B$ is roughly the same as $\text{inv}(A) * B$, depending on computational accuracy.
.^	Array power	power(A, B)	If $C = A.^B$, then $C(I, J) = A(I, J)^B(I, J)$.
^	Matrix power	mpower(A, B)	See online help for a discussion of this operator.
.'	Vector and matrix transpose	transpose(A)	$A.'$. Standard vector and matrix transpose.
'	Vector and matrix complex conjugate transpose	ctranspose(A)	A' . Standard vector and matrix conjugate transpose. When A is real $A.' = A'$.
+	Unary plus	uplus(A)	$+A$ is the same as $0 + A$.
-	Unary minus	uminus(A)	$-A$ is the same as $0 - A$ or $-1 * A$.
:	Colon		Discussed in Section 2.8.

Function	Description
<code>imadd</code>	Adds two images; or adds a constant to an image.
<code>imsubtract</code>	Subtracts two images; or subtracts a constant from an image.
<code>immultiply</code>	Multiplies two images, where the multiplication is carried out between pairs of corresponding image elements; or multiplies a constant times an image.
<code>imdivide</code>	Divides two images, where the division is carried out between pairs of corresponding image elements; or divides an image by a constant.
<code>imabsdiff</code>	Computes the absolute difference between two images.
<code>imcomplement</code>	Complements an image. See Section 3.2.1.
<code>imlincomb</code>	Computes a linear combination of two or more images. See Section 5.3.1 for an example.

Operator	Name
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

TABLE 2.6
Relational operators.

Operator	Name
&	AND
	OR
~	NOT

TABLE 2.7
Logical operators.

Function	Comments
xor (exclusive OR)	The xor function returns a 1 only if both operands are logically different; otherwise xor returns a 0.
all	The all function returns a 1 if all the elements in a vector are nonzero; otherwise all returns a 0. This function operates columnwise on matrices.
any	The any function returns a 1 if any of the elements in a vector is nonzero; otherwise any returns a 0. This function operates columnwise on matrices.

TABLE 2.8
Logical functions.

Function	Description
<code>iscell(C)</code>	True if C is a cell array.
<code>iscellstr(s)</code>	True if s is a cell array of strings.
<code>ischar(s)</code>	True if s is a character string.
<code>isempty(A)</code>	True if A is the empty array, [].
<code>isequal(A, B)</code>	True if A and B have identical elements and dimensions.
<code>isfield(S, 'name')</code>	True if 'name' is a field of structure S.
<code>isfinite(A)</code>	True in the locations of array A that are finite.
<code>isinf(A)</code>	True in the locations of array A that are infinite.
<code>isletter(A)</code>	True in the locations of A that are letters of the alphabet.
<code>islogical(A)</code>	True if A is a logical array.
<code>ismember(A, B)</code>	True in locations where elements of A are also in B.
<code>isnan(A)</code>	True in the locations of A that are NaNs (see Table 2.10 for a definition of NaN).
<code>isnumeric(A)</code>	True if A is a numeric array.
<code>isprime(A)</code>	True in locations of A that are prime numbers.
<code>isreal(A)</code>	True if the elements of A have no imaginary parts.
<code>isspace(A)</code>	True at locations where the elements of A are whitespace characters.
<code>issparse(A)</code>	True if A is a sparse matrix.
<code>isstruct(S)</code>	True if S is a structure.

TABLE 2.9

Some functions that return a logical 1 or a logical 0 depending on whether the value or condition in their arguments are true or false. See online help for a complete list.

Function	Value Returned
ans	Most recent answer (variable). If no output variable is assigned to an expression, MATLAB automatically stores the result in ans.
eps	Floating-point relative accuracy. This is the distance between 1.0 and the next largest number representable using double-precision floating point.
i (or j)	Imaginary unit, as in $1 + 2i$.
NaN or nan	Stands for Not-a-Number (e.g., $0/0$).
pi	3.14159265358979
realmax	The largest floating-point number that your computer can represent.
realmin	The smallest floating-point number that your computer can represent.
computer	Your computer type.
version	MATLAB version string.

TABLE 2.10

Some important variables and constants.

M-Function 编程简介

➤ 语句关键字

If, for, while, break, continue, switch, return, try...catch

Statement	Description
if	if, together with else and elseif, executes a group of statements based on a specified logical condition.
for	Executes a group of statements a fixed (specified) number of times.
while	Executes a group of statements an indefinite number of times, based on a specified logical condition.
break	Terminates execution of a for or while loop.
continue	Passes control to the next iteration of a for or while loop, skipping any remaining statements in the body of the loop.
switch	switch, together with case and otherwise, executes different groups of statements, depending on a specified value or string.
return	Causes execution to return to the invoking function.
try...catch	Changes flow control if an error is detected during execution.

TABLE 2.11
Flow control
statements.

- **MATLAB** 语言专门针对矩阵运算进行了优化。尽可能地利用这一优势可以极大地加快计算速度。
 - ✓ 循环向量化
 - ✓ 数组预分配

循环向量化

- 向量化是指将for循环和while循环转换为等价的向量或者矩阵操作。
- 循环向量化之后代码长度将明显缩短，这不仅可以带来计算速度的显著提高，还能提高代码可读性。

例如: $f(x) = A \sin(x/2\pi)$

- for x= 1:M
 $f(x) = A \sin((x-1)/(2 * \pi));$
end

下面的代码更加高效:

- x=0:M-1;
 $f = A \sin(x/(2 * \pi));$

循环向量化

- 类似地，使用`meshgrid`函数可以对 $f(x, y) = A \sin(ux + vy)$ 的计算过程进行优化

- `[C,R]=meshgrid(c,r)`

这个函数将行向量`c`和`r`指定的域转换为数组`C`和`R`，可以用于两个变量的函数的估值和三维曲面的绘制(注意，在`meshgrid`的输入和输出中都将列放在了第一个参数的位置)。

例子: `c=[0,1] r=[0 1 2]`
 `>>[C R]=meshgrid(c,r)`
 `C = 0 1`
 `0 1`
 `0 1`
 `R = 0 0`
 `1 1`
 `2 2`

循环向量化

➤ 常用函数

$h = R.^2 + C.^2$

$h =$

0 1

1 2

4 5

数组预分配

- 在处理数字或逻辑数组时，预分配是指创建特定维度的元素全部为0的数组。
- 例如，如果我们要处理图像f和g，两幅图像的大小都是 1024×1024 ，预分配的语句如下

```
f=zeros(1024); g=zeros(1024);
```

- 在处理大型数组时，预分配有助于减少内存碎片。动态内存分配可能会导致内存碎片化，这带来的结果是，在计算过程中内存中可用空间有很多，但是没有足够的连续内存来容纳一个大变量。通过允许MATLAB在计算开始时为大型数据结构保留足够的内存，预分配有助于防止这种情况的发生。

➤ disp(argument)

如果参数是数组，则显示其内容。如果参数是文本字符串，则 disp 显示字符串中的字符。

例如：

```
A=[1 2; 3 4];  
disp(A);  
sc='Digital image processing';  
disp(sc);
```

➤ `t=input('message')`
`t=input('message', 's')`

例子:

```
t=input('Enter your data:', 's')
class(t)
size(t)
n=str2num(t)
size(n)
class(n)
```

交互式 I/O

```
➤ >>t='12.6, x2y, z';  
>>[a,b,c]=strread(t,'%f%q%q','delimiter',',')  
a =  
    12.6000  
b =  
    'x2y'  
c =  
    'z'
```

➤ strcmp(s1, s2)

如果字符串s1和s2相同，则返回1，否则返回0

例子：

```
strcmp(b,'x2y')
```

```
strcmp(b,'z')
```


元胞/单元数组(Cell arrays)和结构体(structure)

➤ 元胞数组

元胞数组是一个多维数组，元胞数组里面的元素是其他数组元素的复制。

例如：`c={' gauss'; [1 0;0 1],3}`

➤ 结构体

结构体同元胞数组非常类似，在某种意义上，它们允许将不同数据的集合分组到单个变量中。

例如：

`S.char string='gucas'; S.matrix=[1 0;0 1]; S.scalar=3;`