

# 课程实验2

---

- 编写matlab脚本实现如下功能：
  - (1) 读入一张真彩色图像；
  - (2) 编写matlab函数实现真彩色-》灰度图像的转换并调用输出结果，保存为新图像文件；
  - (3) 编写matlab函数统计一张灰度图像中每一个像素值的数量并输出二维统计图形。
  - (4) 编写matlab函数实现真彩色-》二值图像的转换并调用输出结果，保存为新图像文件；
  - (5) 编写matlab函数统计一张二值图像中的0/1像素数量并输出二维统计图形；

备注：不准调用matlab中与上述功能相关的自有函数。

注意一下imshow 在显示图像时的 数据类型:

uint8, double

二值化  $\rightarrow$  0,1  $\rightarrow$  imshow , 0,1 按照灰度显示, 看起来是很黑的图片

## imshow

显示图像

### 语法

```
imshow(I)
imshow(I,[low high])
imshow(I,[])
imshow(RGB)
imshow(BW)
imshow(X,map)
imshow(filename)
imshow(___,Name,Value)

himage = imshow( __ )
```

### 说明

imshow(I) 在图窗中显示灰度图像 I。imshow 使用图像数据类型的默认显示范围, 并优化图窗、坐标区和图像对象属性以便显示图像。

imshow(I,[low high]) 显示灰度图像 I, 以二元素向量 [low high] 形式指定显示范围。有关详细信息, 请参阅 [DisplayRange](#) 参数。

imshow(I,[]) 显示灰度图像 I, 根据 I 中的像素值范围对显示进行转换。imshow 使用 [min(I(:)) max(I(:)) ] 作为显示范围。imshow 将 I 中的最小值显示为黑色, 将最大值显示为白色。有关详细信息, 请参阅 [DisplayRange](#) 参数。

imshow(RGB) 在图窗中显示真彩色图像 RGB。

imshow(BW) 在图窗中显示二值图像 BW。对于二值图像, imshow 将值为 0 (零) 的像素显示为黑色, 将值为 1 的像素显示为白色。

## 用矩阵操作，避免循环

```
main.m x +
1 clear;clc;
2 A = rand(3,3,2)
3
4 B1 = 0.5 * A(:,:,1) + 0.5 * A(:,:,2)
5 for i=1:3
6     for j=1:3
7         B2(i,j) = 0.5 * A(i,j,1) + 0.5 * A(i,j,2);
8     end
9 end
10
11 B2
12
```

A(:, :, 1) =

0.0759	0.7792	0.5688
0.0540	0.9340	0.4694
0.5308	0.1299	0.0119

A(:, :, 2) =

0.3371	0.3112	0.6020
0.1622	0.5285	0.2630
0.7943	0.1656	0.6541

B1 =

0.2065	0.5452	0.5854
0.1081	0.7313	0.3662
0.6625	0.1478	0.3330

B2 =

0.2065	0.5452	0.5854
0.1081	0.7313	0.3662
0.6625	0.1478	0.3330

## 避免循环

```
1 - clc;clear;close all
```

```
3 - x = rand(3,4)
```

```
5 - mask = x > 0.5
```

```
7 - num = sum(sum(mask))
```

```
10 - num_for_loop = 0;
```

```
12 - for i=1:3
```

```
13 -     for j=1:4
```

```
14 -         if x(i,j) > 0.5
```

```
15 -             num_for_loop = num_for_loop + 1;
```

```
16 -         end
```

```
17 -     end
```

```
18 - end
```

```
20 - num_for_loop
```

x =

0.225921780972399	0.435698684103899	0.430207391329584	0.979748378356085
0.170708047147859	0.311102286650413	0.184816320124136	0.438869973126103
0.227664297816554	0.923379642103244	0.904880968679893	0.111119223440599

mask =

3×4 logical 数组

0	0	0	1
0	0	0	0
0	1	1	0

num =

3

num\_for\_loop =

3

## 有关 sum

```
Untitled2.m x +
1 - clc;clear;close all
2
3 - x = round(rand(3,4)*3)
4
5
6
7
8 - sum_0 = sum(x)
9
10 - sum_1 = sum(x,1)
11
12 - sum_2 = sum(x,2)
13
14 - sum_3 = sum(sum_0)
15
```

```
x =
     2     2     1     3
     1     2     3     3
     1     1     0     2

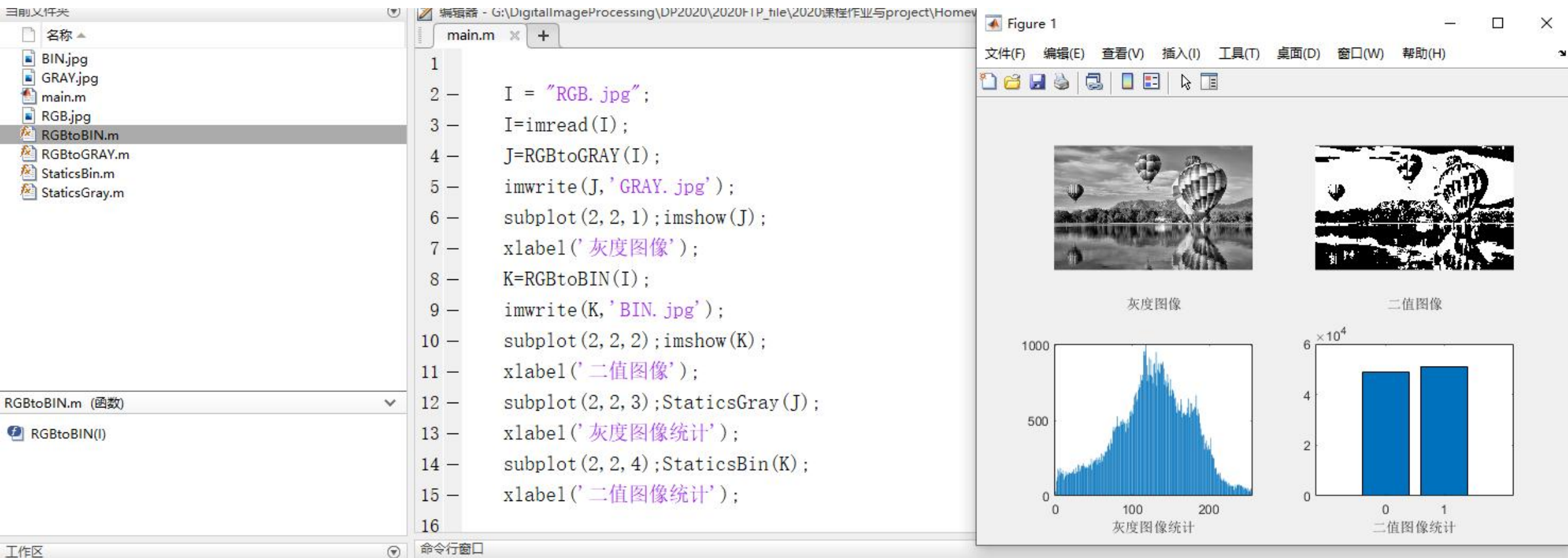
sum_0 =
     4     5     4     8

sum_1 =
     4     5     4     8

sum_2 =
     8
     9
     4

sum_3 =
```

# 一个样例



有些细节随意，例如图片文件名是否让用户输入，plot的图像分四个还是用subplot合在一起，等等