

## 5.1 图像处理.

(2022.5.29)

### ▷ [1- 图片的读写和显示]

• `pic = imread("image.jpg");` % 读写

% 工作区中 `200*200`: 像素点的个数 / 对应矩阵的大小.

`uint8`: `unsigned int 8` (无符号整型)

% 矩阵元素值的范围: `0:255` (`0`-黑色, `255`-白色)

• `imshow(pic);` % 显示.

• `temp = pic(60:100, 60:100);`

`imshow(temp);`

% 显示部分图像

• 生成随机图像.

`m = randi([0,255],[400,400]);`

`m = uint8(m);` % 转换为 `uint8` 格式.

`imshow(m);`

• 生成渐变图像.

```
for i = 1:256
    for j = 1:400
        bw(i,j) = i-1; % 逐项递增
    end
end
```

⇒ `bw = uint8(bw);`  
`imshow(bw);` 即可!

• 保存图像

`imwrite(bw, "image\bw.jpg");` % 存取至特定路径.

% 支持格式: `jpg`, `png`, `jpeg`, `tif`, `bmp`, `ico`, ...

## ▷ [2-彩色图, 灰度图和二值化]

### (1) RGB分离与合并

分离

- 首先读取图像 `pepper = imread("peppers.png");`  
`imshow(pepper);`
- 利用矩阵的索引获取三通道  
`R = pepper(:, :, 1)`  
`G = pepper(:, :, 2)`  
`B = pepper(:, :, 3)`
- 利用 `subplot` 命令同时显示多张图片即可。

合并

- 定义 `rgb`  
`rgb(:, :, 1) = R;`  
`rgb(:, :, 2) = G;`  
`rgb(:, :, 3) = B;`
- `imshow(rgb);`

### (2) 彩色图转灰度图

`rgb2gray(pepper);` % 内置语句 `rgb2gray`

% 原理: 计算 R, G, B 分量加权求和 → 转化为灰度值

### (3) 二值化 (`imbinarize`) % 应用: 去图片阴影

(法一) `[row, col] = size(pepper_gray);`

作 row, col 的二重循环

{	<code>pepper_gray(i, j) &gt; 128</code>	记为 1
	otherwise	记为 0

→ 控制阈值

(法二) `pepper_gray = rgb2gray(pepper);`

`bw = imbinarize(pepper_gray);` % 系统内置二值化函数

`imshow(bw);`

→ 参数

method (二值化方法) { global (默认)  
adaptive

Sensitivity (敏感度因子) % 默认 0.5 ([0, 1] 之间)

Foreground Polarity { bright (默认) (亮)  
dark (前景比背景暗)



### 17 [3- 图像处理函数]

#### (1) 调整图像大小 (imresize) % 拉伸

`imresize(I, 0.5);` % 0.5 → 传入数值, 缩放倍数 (scale)

`imresize(pic, [row, col]);` % 调整大小 row\*col.

#### (2) 旋转图像 (imrotate)

`imrotate(pic, angle);` % angle 为旋转角度.

#### (3) 图像的加减乘除

加 `imadd(I, 50)` % 提亮图像, 为矩阵中每个元素加 50.


`imadd(I, J)` % 叠加, 类似于发光图层 (I).

减 `imsubtract(I, J)`

乘 `immultiply(I, scale)` } % 矩阵的点乘/除.

除 `imdivide(I, scale)` }

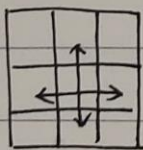
#### (4) 直方图/直方图均衡化.

`imhist(I);` %  统计不同灰度像素点的数量

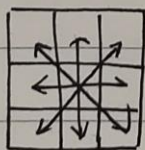
`J = histeq(I);` } % 灰度均衡化 ↓

`imhist(J);`

#### (5) 标注连通分量.



4连通



8连通

处理二值化矩阵

% 按照左图所示方向搜索, 若非零则划为一个区块!

区块会进行编号!

`bwlabel(BW, 4)`

% 应用: 找图像中图形 (如找硬币) 的个数.

`bwlabel(BW, 8)`