江西理工大學 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Digital Image Processing

数字图像处理

Lecture 011:

Review some concept based on matlab code

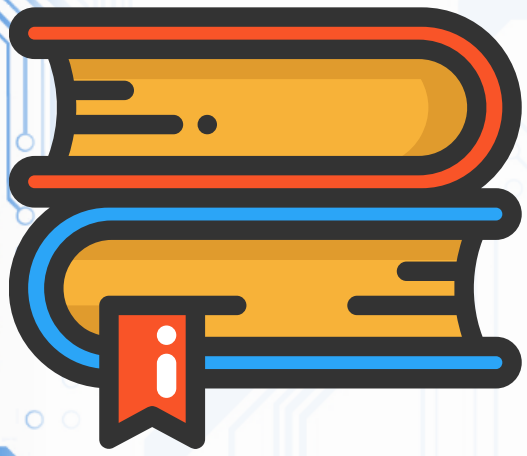
Dr Ata Jahangir Moshayedi

Prof Associate ,
School of information engineering Jiangxi university of science and technology, China

EMAIL: ajm@jxust.edu.cn

Autumn _2021

江西理工大學信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

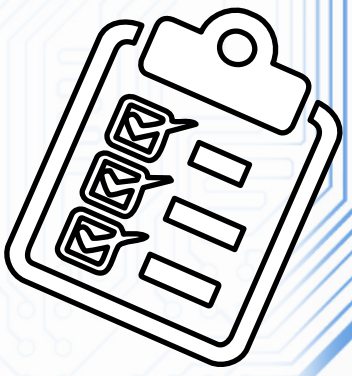


Digital Image Processing

LECTURE 11: Review some concept based on matlab code

Using MATLAB Image-processing tool-box_B

Agenda



- Converting image types
- Read in 8-bit intensity image
- Accessing pixel values
- Extract Red/Blue/Green Channel (1st Channel)
- Work with Pixels
- Image differencing using arithmetic subtraction
- Image multiplication and division
- All Arithmetic Operations
- Invert on image
- Logical operations on images
- Thresholding
- Point-based operations on images
- Logarithmic transform
- Exponential transform
- Power-law (gamma) transform

- Application: gamma correction
- Pixel distributions :histograms
- Histograms for threshold selection
- Adaptive thresholding
- Contrast stretching
- Histogram matching in practice
- Adaptive histogram equalization applied to a sample image
- Histogram operations on colour images

- 转换图像类型
- 读取 8 位强度图像
- 访问像素值
- 提取红/蓝/绿通道（第一通道）
- 使用像素- 使用算术减法的图像差分
- 图像乘法和除法-所有算术运算
- 图像反转
- 对图像的逻辑操作
- 阈值- 基于点的图像操作

-对数变换-指数变换-幂律（伽马）变换

- 应用：伽马校正
- 像素分布：直方图
- 阈值选择的直方图
- 自适应阈值-对比拉伸
- 实践中的直方图匹配
- 应用于样本图像的自适应直方图均衡
- 彩色图像的直方图操作

Multi channel image

多通道图像 Duō tōngdào túxiàng

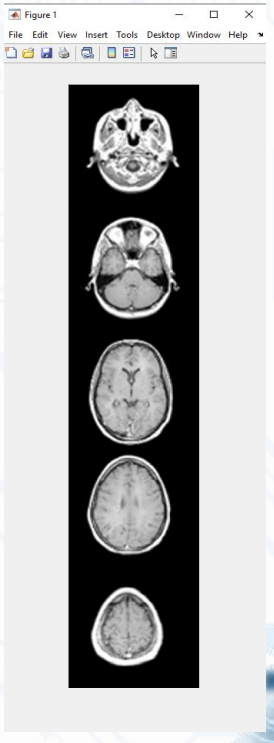
```
clc
clear all
p1 = imread('C:\Users\AJM\Desktop\f.tif',1);
p2 = imread('C:\Users\AJM\Desktop\f.tif',2);
p3 = imread('C:\Users\AJM\Desktop\f.tif',3);
p4 = imread('C:\Users\AJM\Desktop\f.tif',4);
p5 = imread('C:\Users\AJM\Desktop\f.tif',5);
pout=cat(1,p1,p2,p3,p4,p5)
imshow(pout)
```

Input

Workspace

Name	Value
p1	340x340x3 uint8
p2	340x340x3 uint8
p3	340x340x3 uint8
p4	340x340x3 uint8
p5	340x340x3 uint8
pout	1700x340x3 uint8

Output



Making multi dimensional for multi frame image

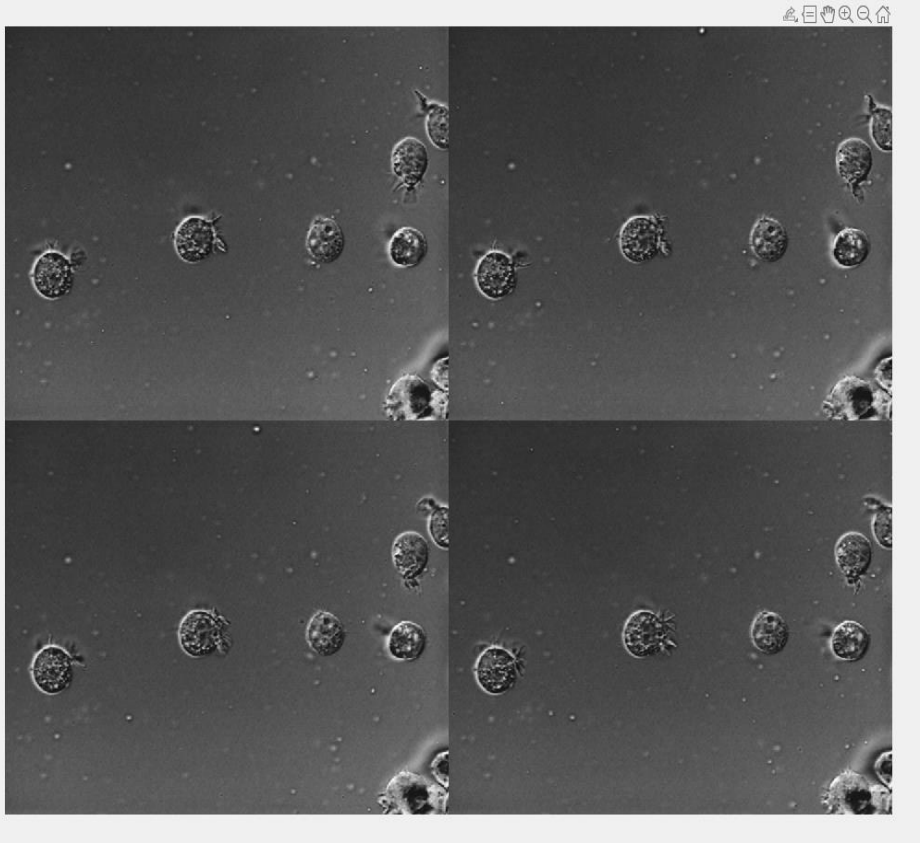
为多帧图像制作多维

Wèi duō zhēn túxiàng zhìzuò duōwéi

Showing all frame in one image

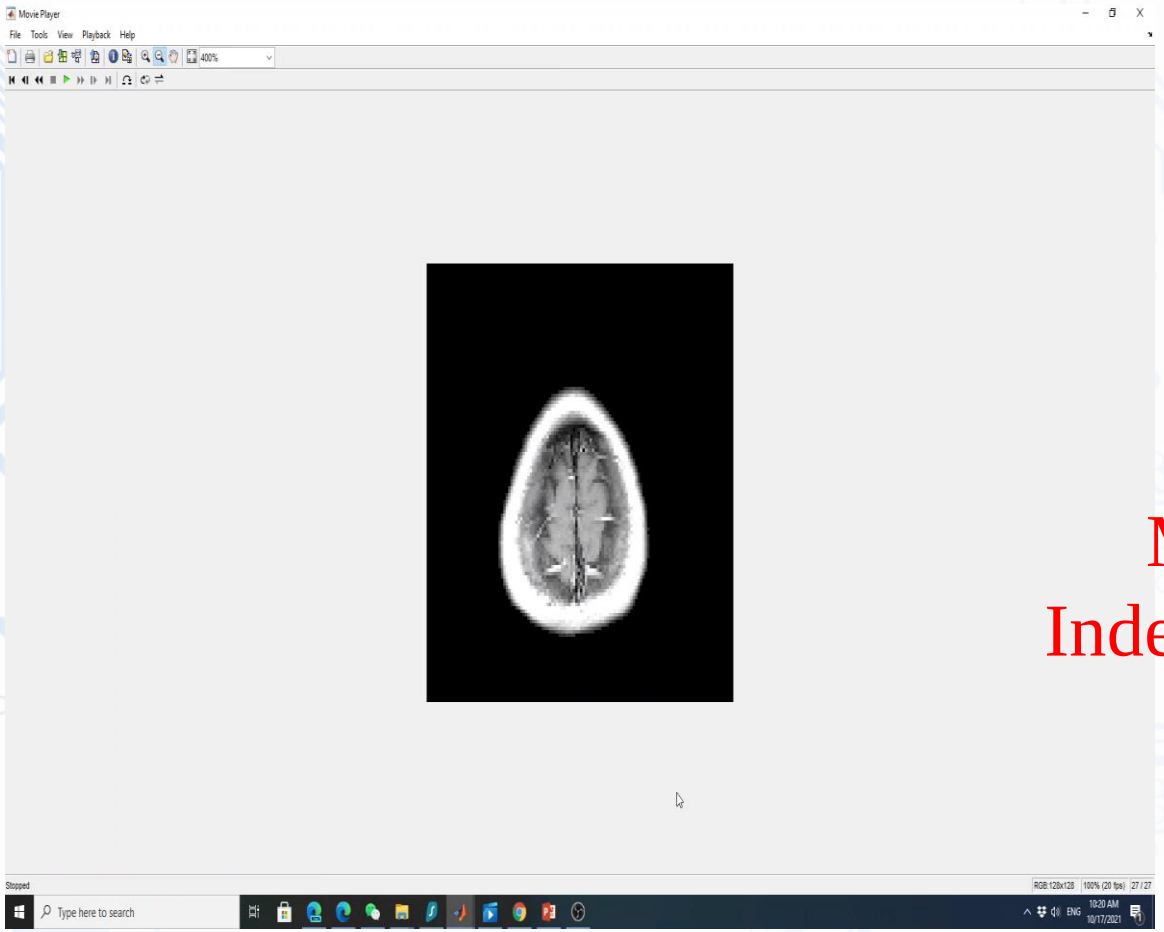
- 在一张图像中显示所有帧
- Zài yī zhāng túxiàng zhōng xiǎnshì suǒyǒu zhèng

```
clc
clear all

img1 = imread('AT3_1m4_01.tif');
img2 = imread('AT3_1m4_02.tif');
img3 = imread('AT3_1m4_03.tif');
img4 = imread('AT3_1m4_04.tif');
multi = cat(3,img1,img2,img3,img4);

montage(multi);
```



Make movie from multi frame image

- 从多帧图像制作电影
- Cóng duō zhēn túxiàng zhìzuò diànyǐng




```
load mri
mov = immovie(D,map);
implay(mov)
```

Make Movie from Indexed Image Sequence

Otsu Thresholding

大津阈值 Dàjīn yùzhí

- Converting a greyscale image to monochrome is a common image processing task.
- Otsu's method, named after its inventor Nobuyuki Otsu, is one of many binarization algorithms.
- Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background.
- The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum.

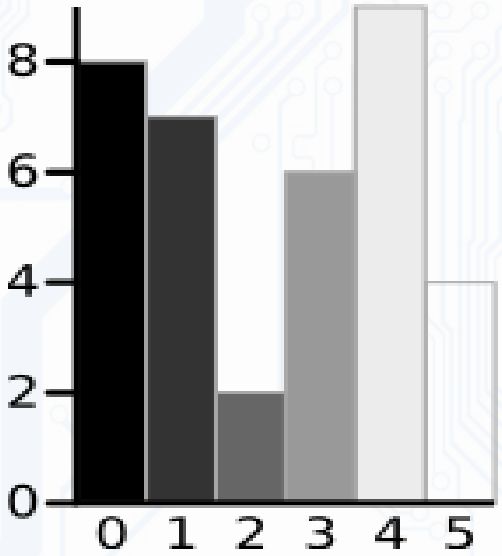
大津阈值 Dàjīn yùzhí

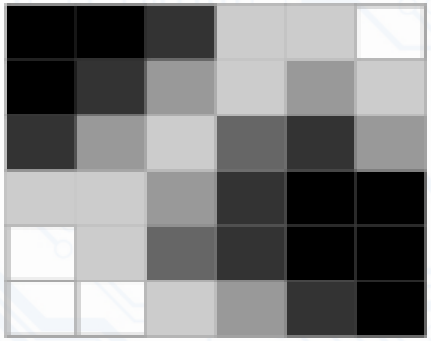
- - 将灰度图像转换为单色是一项常见的图像处理任务。- Otsu 的方法以其发明者 Nobuyuki Otsu 的名字命名，是众多二值化算法之一。
- -Otsu 的阈值方法涉及迭代所有可能的阈值并计算阈值每一侧的像素级别的扩散度量，即落在前景或背景中的像素。目的是找到前景和背景传播之和最小的阈值。

Otsu Thresholding

大津阈值 Dàjīn yùzhí

- The algorithm will be demonstrated using the simple 6x6 image shown below. The histogram for the image is shown next to it. To simplify the explanation, only 6 greyscale levels are used



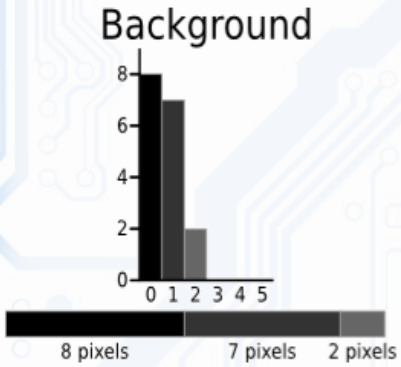


Otsu Thresholding_One example

大津阈值 Dàjīn yùzhí

- The calculations for finding the foreground and background variances (the measure of spread) for a single threshold are now shown. In this case the threshold value is 3.

现在显示了用于查找单个阈值的前景和背景方差（扩散度量）的计算。 在这种情况下，阈值为 3。



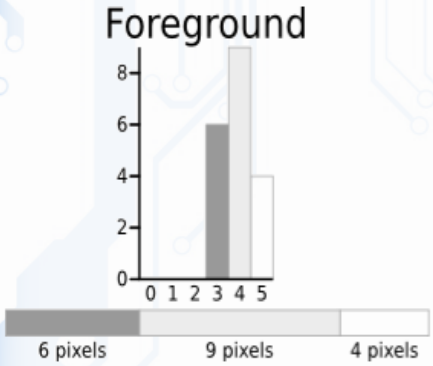
$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

$$\text{Variance} \quad \sigma_b^2 = \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17}$$

$$= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17}$$

$$= 0.4637$$



$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}$$

$$= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}$$

$$= 0.5152$$

Otsu Thresholding_ One example

- The next step is to calculate the 'Within-Class Variance'.
- This is simply the sum of the two variances multiplied by their associated weights.

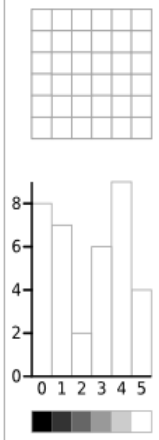
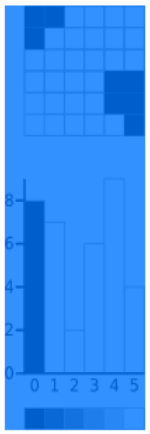
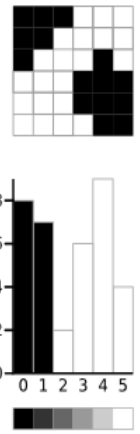
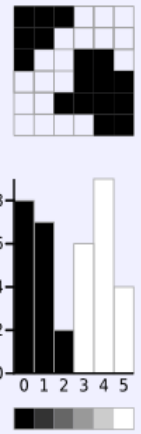
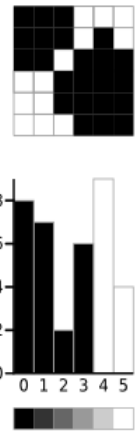
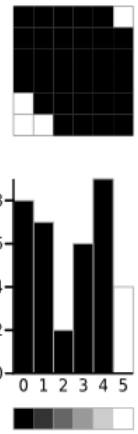
下一步是计算“类内方差”。 这只是两个方差的总和乘以其相关的权重。

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152$$
$$= 0.4909$$

- This final value is the 'sum of weighted variances' for the threshold value 3.
- This same calculation needs to be performed for all the possible threshold values 0 to 5.
- The table below shows the results for these calculations.
- The highlighted column shows the values for the threshold calculated above.

该最终值是阈值 3 的“加权方差总和”。需要对所有可能的阈值 0 到 5 执行相同的计算。 下表显示了这些计算的结果。 突出显示的列显示了上面计算的阈值。

Otsu Thresholding_One example

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	W_b = 0	W_b = 0.222	W_b = 0.4167	W_b = 0.4722	W_b = 0.6389	W_b = 0.8889
Mean, Background	M_b = 0	M_b = 0	M_b = 0.4667	M_b = 0.6471	M_b = 1.2609	M_b = 2.0313
Variance, Background	σ^2_b = 0	σ^2_b = 0	σ^2_b = 0.2489	σ^2_b = 0.4637	σ^2_b = 1.4102	σ^2_b = 2.5303
Weight, Foreground	W_f = 1	W_f = 0.7778	W_f = 0.5833	W_f = 0.5278	W_f = 0.3611	W_f = 0.1111
Mean, Foreground	M_f = 2.3611	M_f = 3.0357	M_f = 3.7143	M_f = 3.8947	M_f = 4.3077	M_f = 5.000
Variance, Foreground	σ^2_f = 3.1196	σ^2_f = 1.9639	σ^2_f = 0.7755	σ^2_f = 0.5152	σ^2_f = 0.2130	σ^2_f = 0
Within Class Variance	σ^2_W = 3.1196	σ^2_W = 1.5268	σ^2_W = 0.5561	σ^2_W = 0.4909	σ^2_W = 0.9779	σ^2_W = 2.2491

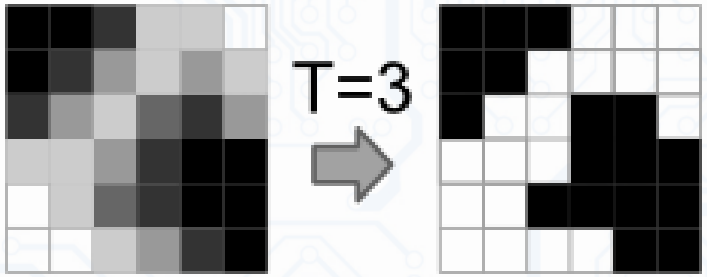
- It can be seen that for the threshold equal to 3, as well as being used for the example, also has the lowest sum of weighted variances.
- Therefore, this is the final selected threshold. All pixels with a level less than 3 are background, all those with a level equal to or greater than 3 are foreground.
- As the images in the table show, this threshold works well

可以看出，对于等于 3 的阈值，以及用于示例，也具有最低的加权方差总和。 因此，这是最终选定的阈值。 级别小于 3 的所有像素都是背景，级别等于或大于 3 的所有像素都是前景。 正如表中的图像所示，这个阈值效果很好

Otsu Thresholding _One example

- It can be seen that for the threshold equal to 3, as well as being used for the example, also has the lowest sum of weighted variances. Therefore, this is the final selected threshold.
- All pixels with a level less than 3 are background, all those with a level equal to or greater than 3 are foreground.
- As the images in the table show, this threshold works well

可以看出，对于等于 3 的阈值，以及用于示例，也具有最低的加权方差总和。 因此，这是最终选定的阈值。 级别小于 3 的所有像素都是背景，级别等于或大于 3 的所有像素都是前景。 正如表中的图像所示，这个阈值效果很好



- This approach for calculating Otsu's threshold is useful for explaining the theory, but it is computationally intensive, especially if you have a full 8-bit greyscale.
- The next section shows a faster method of performing the calculations which is much more appropriate for implementations.

- 这种计算 Otsu 阈值的方法对于解释该理论很有用，但它是计算密集型的，尤其是当您有一个完整的 8 位灰度时。
- 下一节展示了一种更适合实现的更快的计算方法。

Otsu Thresholding _One example

- By a bit of manipulation, you can calculate what is called the *between class* variance, which is far quicker to calculate.
- Luckily, the threshold with the maximum *between class* variance also has the minimum *within class* variance.
- So it can also be used for finding the best threshold and therefore due to being simpler is a much better approach to use.

通过一些操作，您可以计算出所谓的类间方差，它的计算速度要快得多。 幸运的是，具有最大类间方差的阈值也具有最小类内方差。 所以它也可以用于寻找最佳阈值，因此由于更简单，它是一种更好的使用方法。

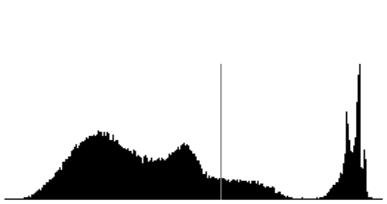
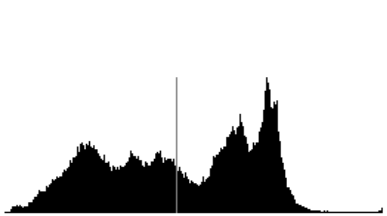
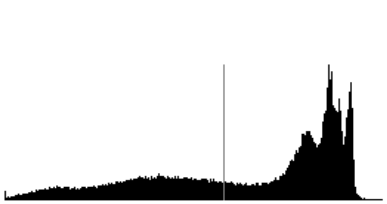
$$\begin{aligned}
\text{Within Class Variance} \quad & \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 \quad \text{(as seen above)} \\
\text{Between Class Variance} \quad & \sigma_B^2 = \sigma^2 - \sigma_W^2 \\
& = W_b(\mu_b - \mu)^2 + W_f(\mu_f - \mu)^2 \quad (\text{where } \mu = W_b\,\mu_b + W_f\,\mu_f) \\
& = W_b\,W_f\,(\mu_b - \mu_f)^2
\end{aligned}$$

The table below shows the different variances for each threshold value.

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
Within Class Variance	σ^2_W = 3.1196	σ^2_W = 1.5268	σ^2_W = 0.5561	σ^2_W = 0.4909	σ^2_W = 0.9779	σ^2_W = 2.2491
Between Class Variance	σ^2_B = 0	σ^2_B = 1.5928	σ^2_B = 2.5635	σ^2_B = 2.6287	σ^2_B = 2.1417	σ^2_B = 0.8705

Otsu Thresholding _One example

- Here are a number of examples of the Otsu Method in use.
- It works well with images that have a bi-modal histogram (those with two distinct regions).

Greyscale Image	Binary Image	Histogram
		
		
		

这里有一些正在使用的大津方法的例子。 它适用于具有双峰直方图（具有两个不同区域的图像）的图像。

Global image threshold using Otsu's method

使用 Otsu 方法的全局图像阈值

Shǐyòng Otsu fāngfǎ de quánjú túxiàng yùzhí

graythresh

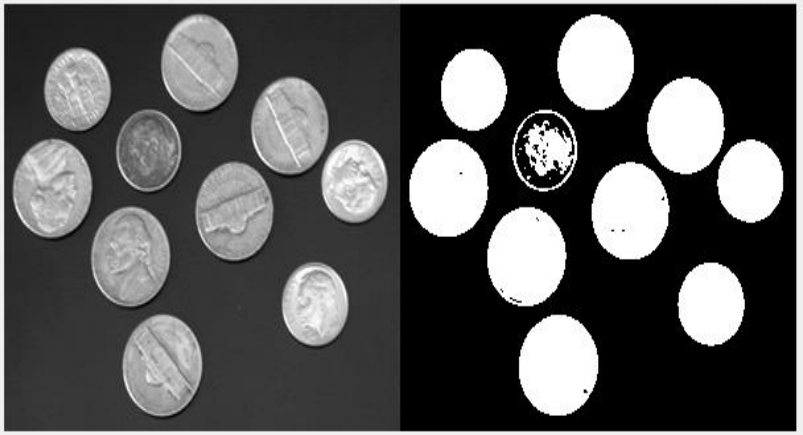
Syntax

- T = graythresh(I)
- [T,EM] = graythresh(I)

```
clc
clear all
    I = imread('coins.png');
    level = graythresh(I)
    BW = imbinarize(I,level);
    imshowpair(I,BW,'montage')
```

```
level =

    0.4941
```

Convert Intensity Image to Binary Image Using Level Threshold



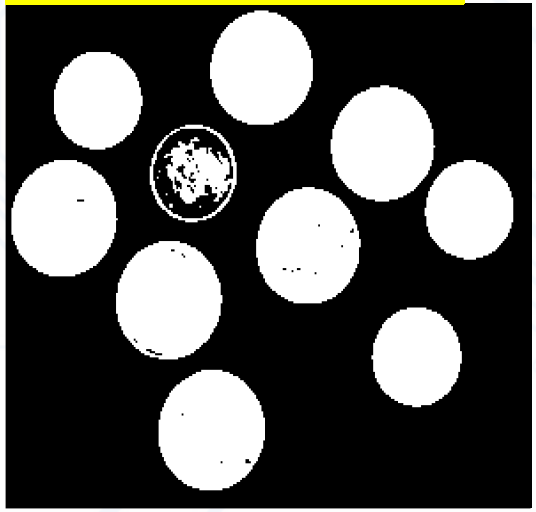
Global image threshold using Otsu's method

```
clc
clear all
    x = imread('coins.png');
    figure(1);imshow(x)
    x2=im2bw(x)
       figure(2);imshow(x2)
    x3=im2bw(x,graythresh(x));
       figure(3);imshow(x3)
```

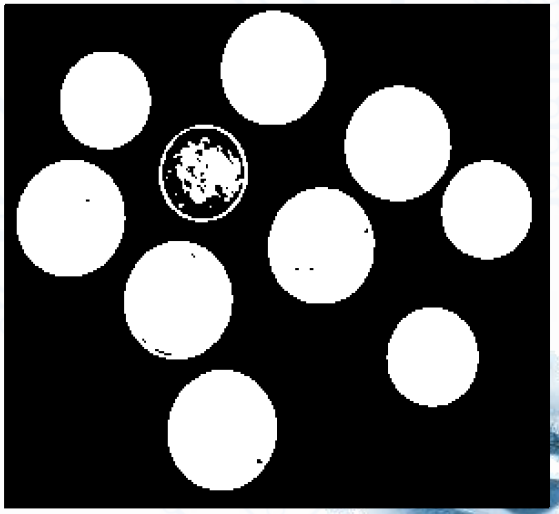
Im2bw:
Convert image to binary image, based on threshold

figure(1);imshow(x)

figure(2);imshow(x2)



figure(3);imshow(x3)



Converting image types

转换图像类型

Matlab also contains built in functions for converting different image types.

Here, we examine conversion to grey scale and the display of individual RGB colour channels from an image.

Matlab还包含用于转换不同图像类型的内置函数。
这里，我们检查从图像到灰度的转换和单个RGB颜色通道的显示。

```
D=imread('onion.png'); % Read in 8-bit RGB colour image.
Dgray = rgb2gray(D); % convert it to a grayscale image
subplot(2,1,1);
imshow(D);
axis image; % display both side by side
subplot(2,1,2);
imshow(Dgray);
```

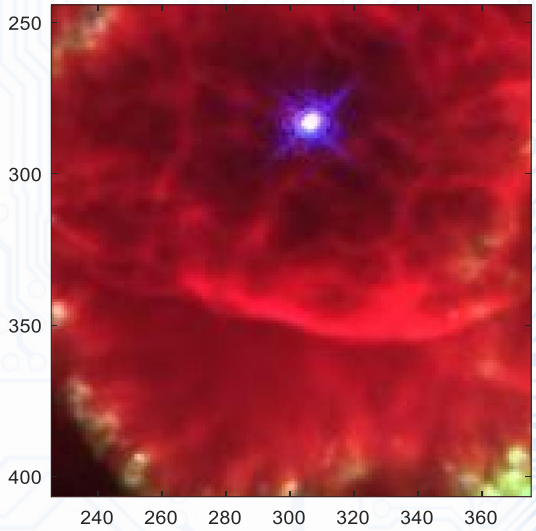
RGB to Intensity

Convert RGB image to indexed image

rgb2ind: Convert RGB image to indexed image

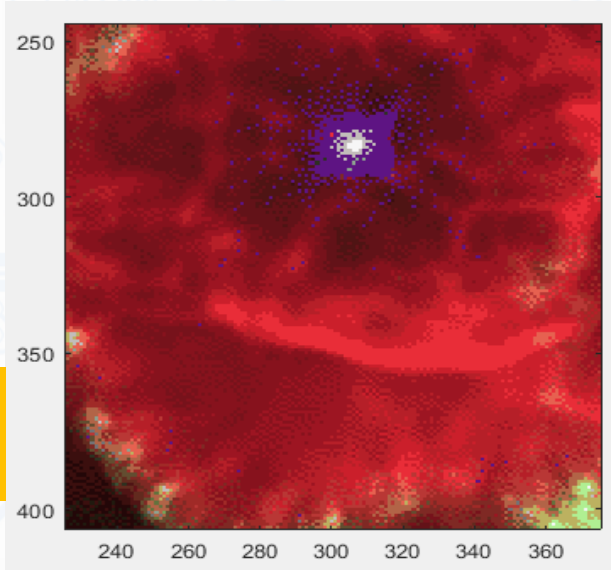
```
clc
clear all
RGB = imread('ngc6543a.jpg');
figure
imagesc(RGB)
axis image
zoom(4)
```

Read and display a truecolor uint8 JPEG image of a nebula




```
clc
clear all
RGB = imread('ngc6543a.jpg');
figure
imagesc(RGB)
axis image
zoom(4)
[IND,map] = rgb2ind(RGB,32);
figure
imagesc(IND)
colormap(map)
axis image
zoom(4)
```

Convert RGB to an indexed image with 32 colors.



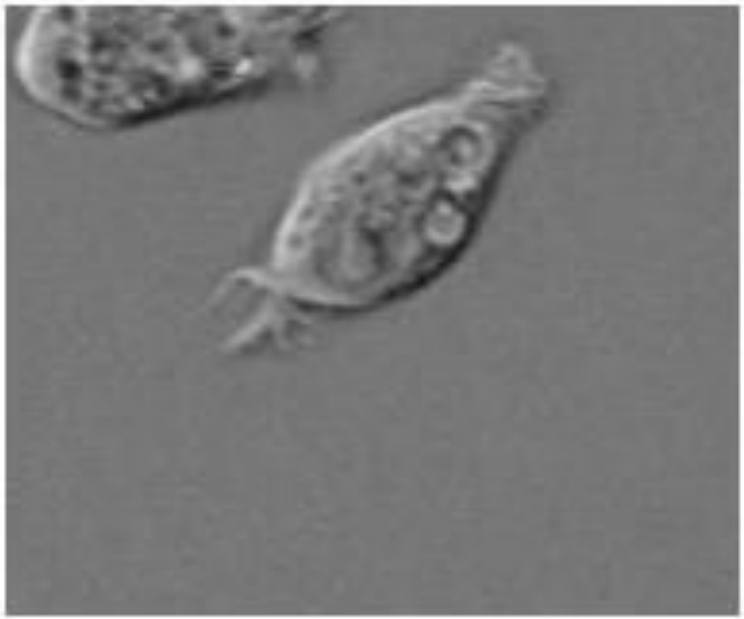
Read in 8-bit intensity image

读取 8 位强度图像

```
clc
close all
B=imread('cell.tif'); % Read in 8-bit intensity image of cell
figure();
 imshow(B); % examine grayscale image in interactive viewer
```

jiāo hù shì交互式

clc
全部关闭
B=imread（'cell.tif'）；%读取细胞的8位强度图像
图（）；
imshow（B）；%在交互式查看器中检查灰度图像



Convert image, increasing apparent color resolution by dithering

Convert Grayscale Image to Binary Image Using Dithering

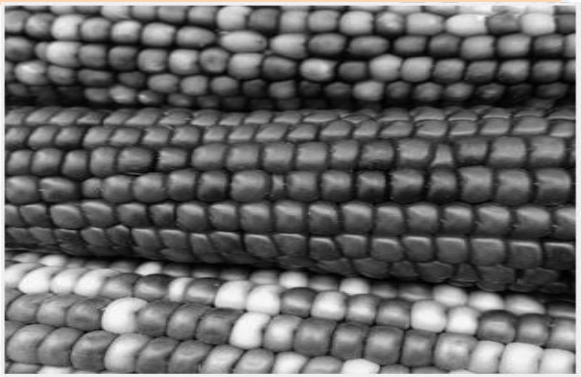
dither
Convert image, increasing apparent color resolution by dithering

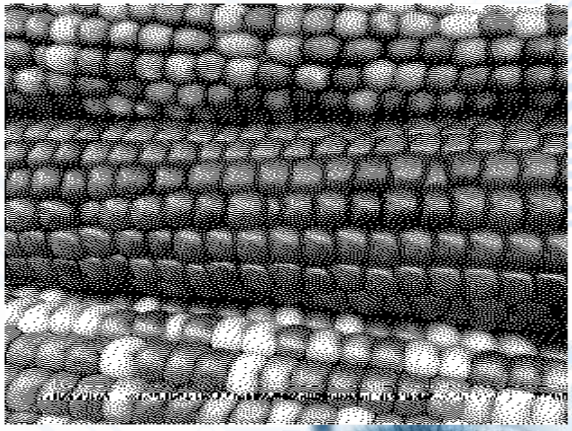
Syntax

X = dither(RGB,map)
X = dither(RGB,map,Qm,Qe)
BW = dither(I)

- Read the grayscale image from the corn.tif file into the MATLAB® workspace.
- The grayscale version of the image is the third image in the file.
- Display the grayscale image using imshow.

```
clc
clear all
corn_gray = imread('corn.tif',3);
figure(1);imshow(corn_gray)
corn_bw = dither(corn_gray);
figure(2);imshow(corn_bw)
```





Accessing pixel values

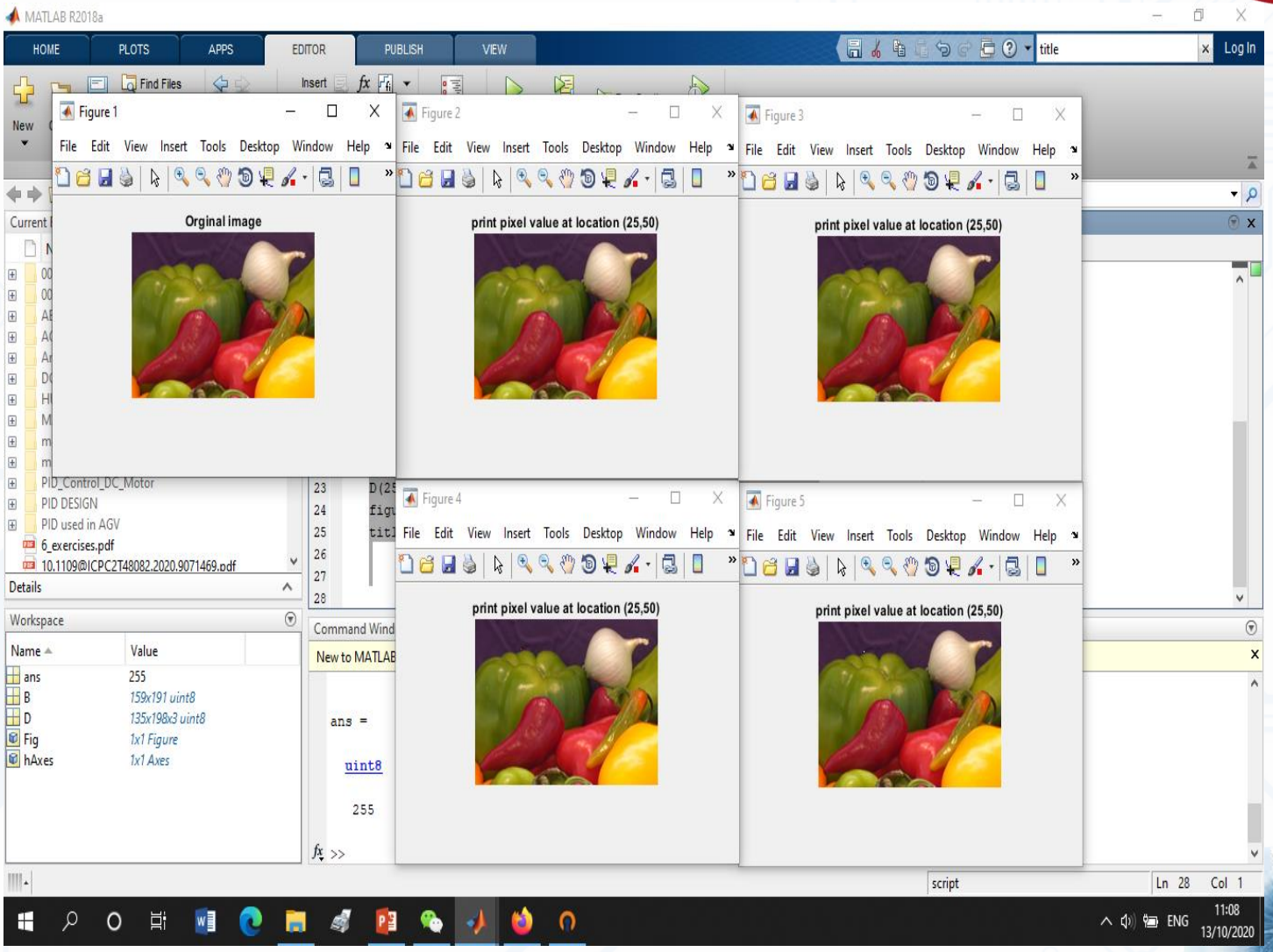
访问像素值

Matlab also contains a built-in interactive image viewer which can be launched using the imview function. Its purpose is slightly different from the other two: it is a graphical, image viewer which is intended primarily for the inspection of images and sub-regions within them.

Matlab还包含一个内置的交互式图像查看器，可以使用imview函数启动该查看器。它的用途与其他两个稍有不同：它是一个图形化的图像查看器，主要用于检查图像及其子区域。

```
clc
close all
B= imread('onion.png'); % Read in 8-bit intensity image of cell
figure();
 imshow(B); % examine grayscale image in interactive viewer
imshow(B); % examine RGB image in interactive viewer
B(25,50) % print pixel value at location (25,50)
B(25,50) = 255; % set pixel value at (25,50) to white
imshow(B); % view resulting changes in image
D(25,50, :) % print RGB pixel value at location (25,50)
D(25,50, 1) % print only the red value at (25,50)
D(25,50, :) = [255, 255, 255]; % set pixel value to RGB white
imshow(D); % view resulting changes in image'); % Read in 8-bit intensity image of cell
```

Accessing pixel values



Get RGB Pixel Values From Image Model

```
close all
clc
h = imshow('flamingos.jpg');
im = imagemodel(h)

r = 100;
c = 200;

pxValue = getPixelValue(im,r,c)

defaultPxInfoStr = getDefaultPixelInfoString(im)

pxInfoStr = getPixelInfoString(im,r,c)

defaultPxRegStr =
getDefaultPixelRegionString(im)

formatFcn = getPixelRegionFormatFcn(im);
pxRegStr = formatFcn(r,c)
```

% Get RGB Pixel Values From Image Model
% Pixel values obtained from an imagemodel object can be returned in several formats suitable for display in different interactive image processing tools.
% Create an image model associated with a color image.

% Select a pixel by specifying row and column coordinates.
This pixel has (row, column) coordinates (100, 200).

% Get the numeric value of the pixel using the getPixelValue function.

% Get the default pixel information string using the getDefaultPixelInfoString function. This string depends on the type of image but does not use the pixel values. The pixel information string is suitable for use with the Pixel Information tool.

% Using the same string format, get the pixel information string for the specified pixel by using the getPixelInfoString function.

% Get the default pixel region string using the getDefaultPixelRegionString function. This string depends on the type of image but does not use the pixel values. The pixel region string is suitable for use with the Pixel Region tool.

% There are two steps to get the pixel region string for the specified pixel in the same string format. First, get a function formatFcn that formats numeric pixel values by using the getPixelRegionFormatFcn function. Then, specify the row and column coordinate of the pixel as input arguments to formatFcn to get the formatted string.

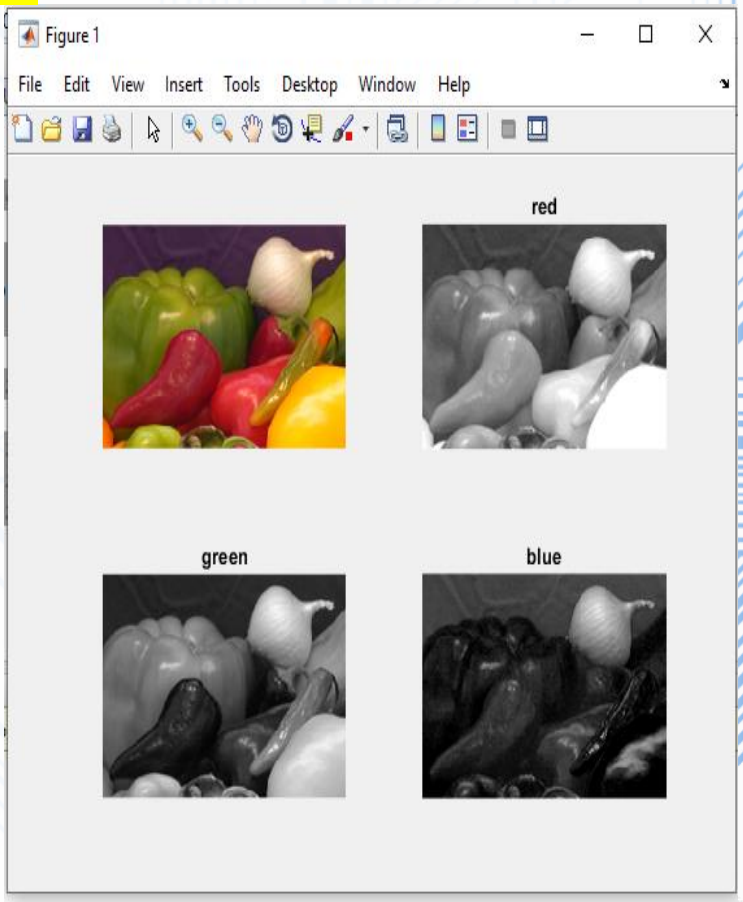
Get RGB Pixel Values From Image Model

```
im =

IMAGEMODEL object accessing an image with these properties:
       ClassType: 'uint8'
    DisplayRange: []
     ImageHeight: 972
       ImageType: 'truecolor'
      ImageWidth: 1296
    MinIntensity: []
    MaxIntensity: []

pxValue =

  1×3 uint8 row vector

   104    95    54

defaultPxInfoStr =

    '[R G B]'

pxInfoStr =
    '[104 95 54]'

defaultPxRegStr =
    'R:000
     G:000
     B:000'

pxRegStr =
  1×1 cell array

    {'R:104↵G: 95↵B: 54'}
```

Extract Red/Blue/Green Channel (1st Channel)

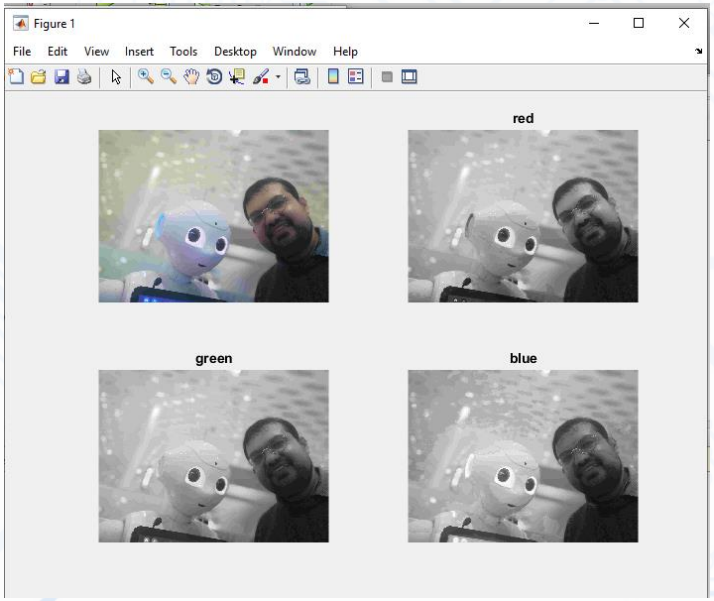
提取红/蓝/绿通道（第一通道）

```
close all
clc
clear all
D=imread('onion.png');      % Read in 8-bit RGB colour image.
Dred  = D(:,:,1);           % extract red channel   (1st channel)
Dgreen = D(:,:,2);          % extract green channel (2nd channel)
Dblue = D(:,:,3);           % extract blue channel  (3rd channel)
subplot(2,2,1); imshow(D); axis image;  % display all in 2x2 plot
subplot(2,2,2); imshow(Dred); title('red');     % display and label
subplot(2,2,3); imshow(Dgreen); title('green');
subplot(2,2,4); imshow(Dblue); title('blue');
```



Extract Red/Blue/Green Channel (1st Channel)

```
clc
clear all
D=imread('f:\ATA.JPG');       % Read in 8-bit RGB colour image.
Dred   = D(:,:,1);           % extract red channel   (1st channel)
Dgreen = D(:,:,2);           % extract green channel (2nd channel)
Dblue = D(:,:,3);            % extract blue channel  (3rd channel)
subplot(2,2,1); imshow(D); axis image;  % display all in 2x2 plot
subplot(2,2,2); imshow(Dred); title('red');     % display and label
subplot(2,2,3); imshow(Dgreen); title('green');
subplot(2,2,4); imshow(Dblue); title('blue');
```



Work with Pixels

使用像素

- Image addition in Matlab Matlab中的图像加法

```
clc
close all
clear all
A=imread('cameraman.tif');
% Read in image
subplot(1,2,1), imshow(A);
 % Display image
B = imadd(A, 100);
% Add 100 pixel values to image A
subplot(1,2,2), imshow(B);
 % Display result image B
```

Work with Pixels

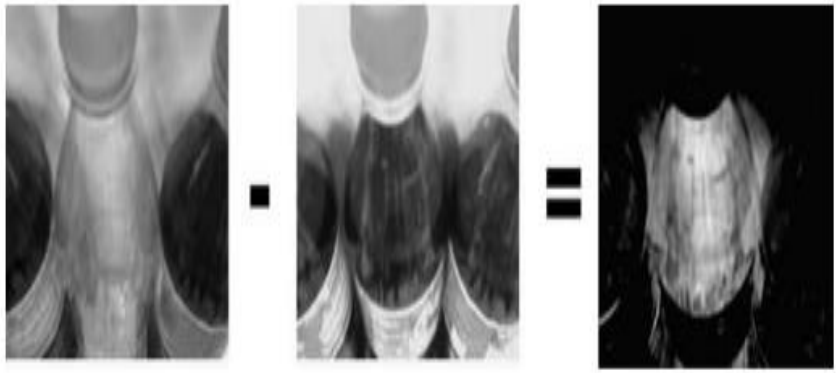
- Image addition in Matlab

```
clc
close all
clear all
A=imread('f:\ATA.JPG'); % Read in image
subplot(1,2,1), imshow(A); % Display image
B = imadd(A, 100); % Add 100 pixel values to image A
subplot(1,2,2), imshow(B); % Display result image B
```

Image differencing using arithmetic subtraction

使用算术减法的图像差分

```
clc
close all
clear all
B=imread('F:\B.jpg'); % Read in 2nd image
A=imread('F:\A.jpg');  % Read in 1st image
subplot(1,3,1), imshow(A); % Display 1st image
subplot(1,3,2), imshow(B); % Display 2nd image
Output = imsubtract(A, B); % subtract images
subplot(1,3,3), imshow(Output); % Display result
```



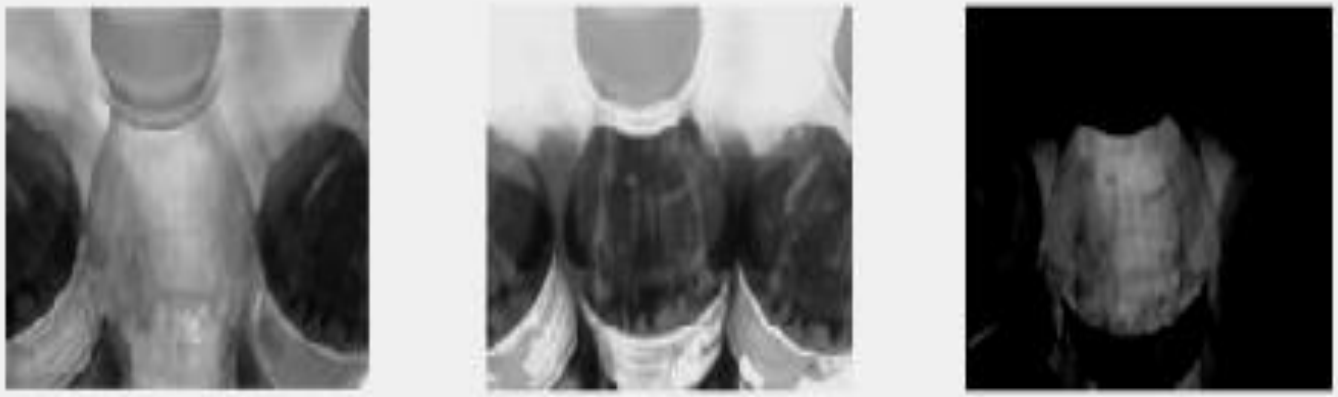


Image differencing using arithmetic subtraction

使用算术减法的图像差分

A useful variation on subtraction is the absolute difference I = .[$ia - ib$] between images . This avoids the potential problem of integer overflow when the difference becomes negative

减法的一个有用变体是绝对差I=[ia]−ib]在图像之间。
这避免了差为负时整数溢出的潜在问题

```
clc
close all
clear all
B=imread('F:\B.jpg'); % Read in 2nd image
A=imread('F:\A.jpg');  % Read in 1st image
subplot(1,3,1), imshow(A); % Display 1st image
subplot(1,3,2), imshow(B); % Display 2nd image
Output = imabsdiff(A, B); % subtract images
subplot(1,3,3), imshow(Output); % Display result
```

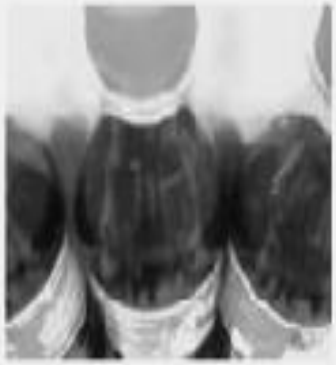




Image multiplication and division

图像乘除 tú xiàng chéng chú

- Multiplication and division can be used as a simple means of contrast adjustment and extension to addition/subtraction (e.g. reduce contrast to 25% ¼ division by 4; increase contrast by 50% ¼ multiplication by 1.5).
- This procedure is sometimes referred to as image colour scaling. Similarly, division can be used for image differencing, as dividing an image by another gives a result of 1.0 where the image pixel values are identical and a value not equal to 1.0 where differences occur.
- However, image differencing using subtraction is computationally more efficient. Following from the earlier examples, image multiplication and division can be performed in Matlab

Image multiplication and division

图像乘除 tú xiàng chéng chú

- 乘法和除法可作为对比度调整和加减法扩展的简单方法（例如，将对比度降低到25%¼乘4；将对比度提高50%¼乘1.5）。

- 此过程有时称为图像颜色缩放。类似地，除法可用于图像差分，因为将图像除以另一图像会得到1.0的结果，其中图像像素值相同，并且在出现差异时，值不等于1.0。

- 然而，使用减法进行图像差分在计算上更有效。根据前面的示例，可以在Matlab中执行图像乘法和除法

Image multiplication and division

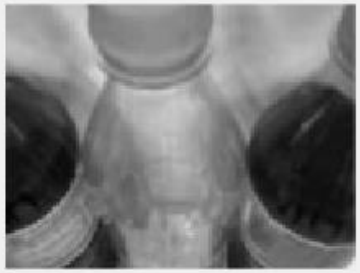
- However, image differencing using subtraction is computationally more efficient. Following from the earlier examples, image multiplication and division can be performed in Matlab as shown

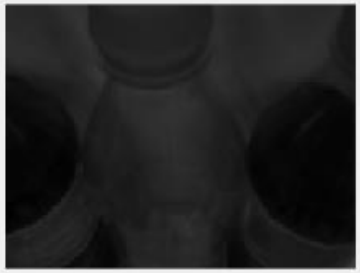
然而，使用减法进行图像差分在计算上更有效。根据前面的示例，可以在Matlab中执行图像乘法和除法，如图所示

Multiplying different images together or dividing them by one another is not a common operation in image processing.

将不同的图像相乘或相除不是图像处理中常见的操作

```
clc
close all
clear all
B=imread('F:\B.jpg'); % Read in 2nd image
A=imread('F:\A.jpg');  % Read in 1st image
subplot(1,4,1), imshow(A); % Display 1st image
subplot(1,4,2), imshow(B); % Display 2nd image
Output = immultiply(A,1.5);     % multiple image by 1.5
subplot(1,4,3), imshow(Output); % Display result
Output = imdivide(A,4);         % divide image by 4
subplot(1,4,4), imshow(Output); % Display result
```



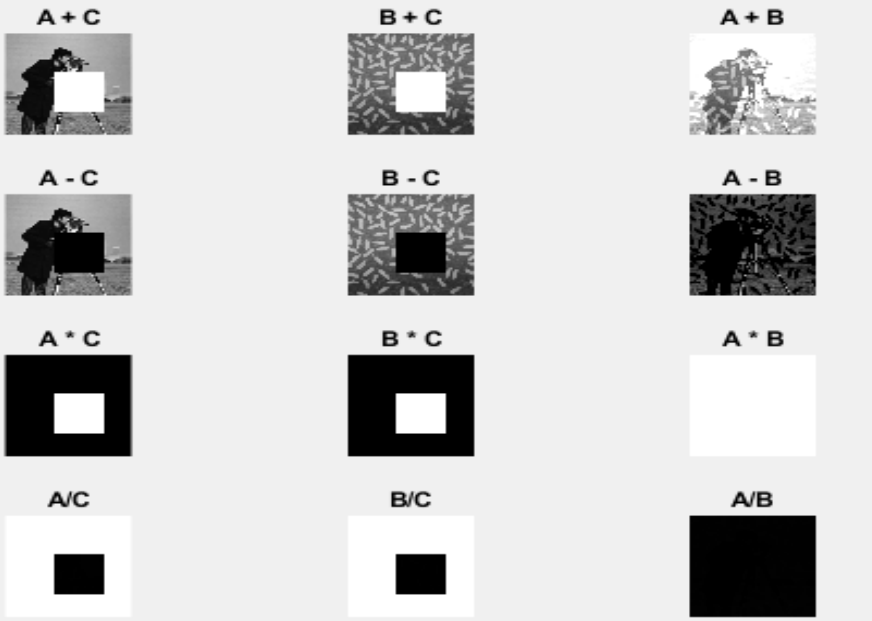



All Arithmetic Operations

所有算术运算

suǒ yǒu suàn shù yùn suàn

```
clc;
clear all;
close all;
A=imread('cameraman.tif');
subplot(5,3,1)
imshow(A)
title('Image A');
B=imread('rice.png');
subplot(5,3,2)
imshow(B)
title ('Image B');
C=zeros(size(A));
for (x= 100: 200)
    for (y=100: 200)
        C(x,y)=255;
    end
end
subplot(5,3,3),imshow(C),title('Image C');
C=uint8(C);
% Arithmetic Operations
lr13=imadd(A,C);figure,subplot(4,3,1),imshow(uint8(lr13)),title('A + C');
lr14=imadd(B,C);subplot(4,3,2),imshow(uint8(lr14)),title('B + C');
lr15=imadd(A,B);subplot(4,3,3),imshow(uint8(lr15)),title('A + B');
lr16=imsubtract(A,C);subplot(4,3,4),imshow(uint8(lr16)),title('A - C');
lr17=imsubtract(B,C);subplot(4,3,5),imshow(uint8(lr17)),title('B - C');
lr18=imsubtract(A,B);subplot(4,3,6),imshow(uint8(lr18)),title('A - B');
lr19=immultiply(A,C);subplot(4,3,7),imshow(uint8(lr19)),title('A * C');
lr20=immultiply(B,C);subplot(4,3,8),imshow(uint8(lr20)),title('B * C');
lr21=immultiply(A,B);subplot(4,3,9),imshow(uint8(lr21)),title('A * B');
lr22=imdivide(A,C);subplot(4,3,10),imshow(uint8(lr22)),title('A/C');
lr23=imdivide(B,C);subplot(4,3,11),imshow(uint8(lr23)),title('B/C');
lr24=imdivide(A,B);subplot(4,3,12),imshow(uint8(lr24)),title('A/B');
```



Invert on image

反转图像

fǎn zhuǎn tú xiàng

```
clc
close all
clear all
A=imread('cameraman.tif'); % read in image
subplot(1,2,1), imshow(A); % display image
B = imcomplement(A); % invert the image
subplot(1,2,2), imshow(B); % display result image B
```

Logical operations on images

对图像的逻辑操作

- OR/XOR Logical OR (and XOR) is useful for processing binary-valued images (0 or 1) to detect objects which have moved between frames. Binary objects are typically
produced through application of thresholding to a grey-scale image.
- . AND Logical AND is commonly used for detecting differences in images, highlighting target regions with a binary mask or producing bit-planes through an image,
- These operations can be performed in Matlab as following

- OR/XOR逻辑OR（和XOR）用于处理二值图像（0或1）以检测在帧之间移动的对象。二值对象通常是通过对灰度图像应用阈值生成的。

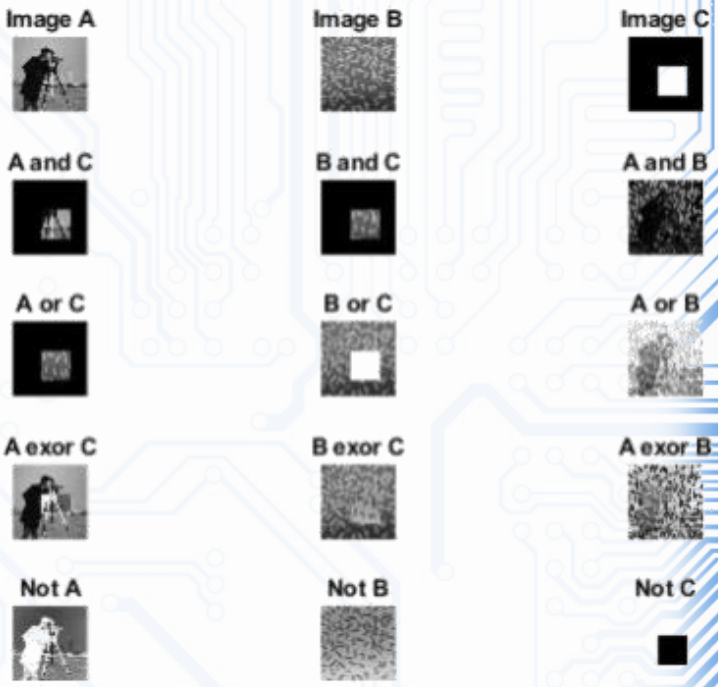
- . 和逻辑AND通常用于检测图像中的差异，用二值掩码突出显示目标区域或通过图像生成位平面，

- 这些操作可以在Matlab中执行，如下所示

Logical Operations

逻辑运算 luó jí yùn suàn

```
% 1.To perform Basic Image Processing Operations
%   a. Arithmetic and Logical Operations
clc;
clear all;
close all;
A=imread('cameraman.tif');
subplot(5,3,1),imshow(A),title('Image A');
B=imread('rice.png');
subplot(5,3,2),imshow(B),title ('Image B');
C=zeros(size(A));
for (x= 100: 200)
    for (y=100: 200)
        C(x,y)=255;
    end
end
subplot(5,3,3),imshow(C),title('Image C');C=uint8(C);
% Logical Operations
lr1=bitand(A,C);subplot(5,3,4),imshow(lr1),title('A and C');
lr2=bitand(B,C);subplot(5,3,5),imshow(lr2),title('B and C');
lr3=bitand(A,B);subplot(5,3,6),imshow(lr3),title('A and B');
lr4=bitor(A,C);subplot(5,3,7),imshow(lr2),title('A or C');
lr5=bitor(B,C);subplot(5,3,8),imshow(lr5),title('B or C');
lr6=bitor(A,B);subplot(5,3,9),imshow(lr6),title('A or B');
lr7=bitxor(A,C);subplot(5,3,10),imshow(lr7),title('A exor C');
lr8=bitxor(B,C);subplot(5,3,11),imshow(lr8),title('B exor C');
lr9=bitxor(A,B);subplot(5,3,12),imshow(lr9),title('A exor B');
lr10= bitcmp(A);subplot(5,3,13),imshow(lr10),title('Not A');
lr11= bitcmp(B);subplot(5,3,14),imshow(lr11),title('Not B');
lr12= bitcmp(C);subplot(5,3,15),imshow(lr12),title('Not C');
```



Thresholding

阈值

yù zhí

Thresholding produces a binary image from a grey-scale or colour image by setting pixel values to 1 or 0 depending on whether they are above or below the threshold value. This is commonly used to separate or segment a region or object within the image based upon its pixel values, In Matlab, this can be carried out using the function im2bw and a threshold in therange0 to 1, as shown .

阈值化通过将像素值设置为1或0（取决于它们是否高于或低于阈值），从灰度或彩色图像生成二值图像。这通常用于根据像素值分离或分割图像中的区域或对象，在Matlab中，可以使用函数im2bw和0到1范围内的阈值来执行，如图所示。

```
close all
clc
clear all
I=imread('trees.tif');  % Read in 1st image
T=im2bw(I, 0.1); % perform thresholding
subplot(1,3,1), imshow(I); % Display original image
subplot(1,3,2), imshow(T); % Display thresholded image
```

Point-based operations on images

基于点的图像操作

- The dynamic range of an image is defined as the difference between the smallest and largest pixel values within the image. We can define certain functional transforms or mappings that alter the effective use of the dynamic range.
- These transforms are primarily applied to improve the contrast of the image. This improvement is achieved by altering the relationship between the dynamic range of the image and the grey-scale (or colour) values that are used to represent the values.
- In general, we will assume an 8-bit (0 to 255) grey-scale range for both input and resulting output images, but these techniques can be generalized to other input ranges and individual channels from colour images

Point-based operations on images

基于点的图像操作

- 图像的动态范围定义为图像中最小和最大像素值之间的差值。我们可以定义某些函数变换或映射来改变动态范围的有效使用。

- 这些变换主要用于提高图像的对比度。这种改进是通过改变图像的动态范围和用于表示值的灰度（或颜色）值之间的关系来实现的。

- 一般来说，我们将假设输入和结果输出图像的灰度范围为8位（0到255），但这些技术可以推广到其他输入范围和彩色图像的单个通道

Logarithmic transform

对数变换 duì shù biàn huàn

```
close all
clc
clear all
I=imread('cameraman.tif'); % Read in image
subplot(2,2,1), imshow(I); % Display image
Id=im2double(I);
Output1=2*log(1+Id);
Output2=3*log(1+Id);
Output3=5*log(1+Id);
subplot(2,2,2), imshow(Output1); % Display result images
subplot(2,2,3), imshow(Output2);
subplot(2,2,4), imshow(Output3);
```

Exponential transform

指数变换

zhǐ shù biàn huàn

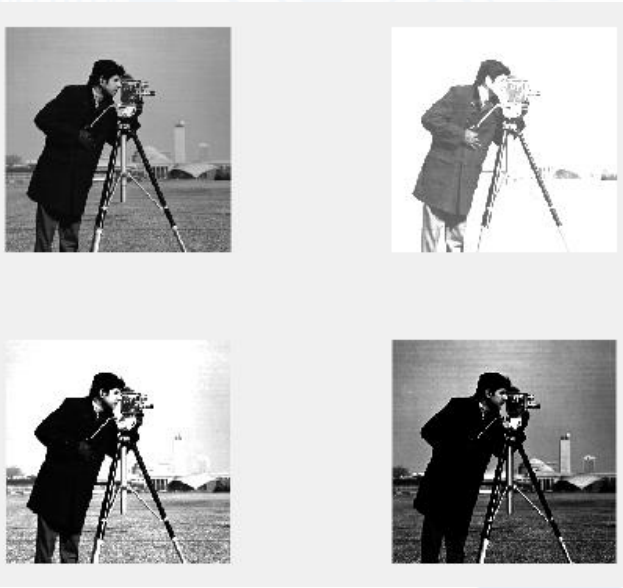
```
close all
clc
clear all
I=imread('cameraman.tif'); % Read in image
subplot(2,2,1), imshow(I); % Display image
 Id=im2double(I);
Output1=4*(((1+0.3).^(Id)) - 1);
Output2=4*(((1+0.4).^(Id)) - 1);
Output3=4*(((1+0.6).^(Id)) - 1);
subplot(2,2,2), imshow(Output1); % Display result images
subplot(2,2,3), imshow(Output2);
subplot(2,2,4), imshow(Output3);
```

Power-law (gamma) transform

幂律（伽马）变换

mì lǜ （gā mǎ ）biàn huàn

```
close all
clc
clear all
I=imread('cameraman.tif'); % Read in image
subplot(2,2,1), imshow(I); % Display image
Id=im2double(I);
Output1=2*(Id.^0.5);
Output2=2*(Id.^1.5);
Output3=2*(Id.^3.0);
subplot(2,2,2), imshow(Output1); % Display result images
subplot(2,2,3), imshow(Output2);
subplot(2,2,4), imshow(Output3);
```



Application: gamma correction

应用：伽马校正

A common application of the power-law transform is gamma correction. Gamma correction is the term used to describe the correction required for the nonlinear output curve of modern computer displays. When we display a given intensity on a monitor we vary the analogue voltage in proportion to the intensity required

幂律变换的一个常见应用是伽马校正。Gamma校正是一个术语，用于描述现代计算机显示器非线性输出曲线所需的校正。当我们在监视器上显示给定的强度时，我们会根据所需的强度按比例改变模拟电压

```
clc
clear all
close all
A=imread('cameraman.tif'); % Read in image
subplot(1,2,1), imshow(A); % Display image
B=imadjust(A,[0 1],[0 1],1./3);
 % Map input grey values of image A in range 0-1 to an
% output range of 0-1 with gamma factor of 1/3 (i.e. r = 3).
% Type $>>$ doc imadjust for details of possible syntaxes
subplot(1,2,2), imshow(B); % Display result.
```

Pixel distributions :histograms

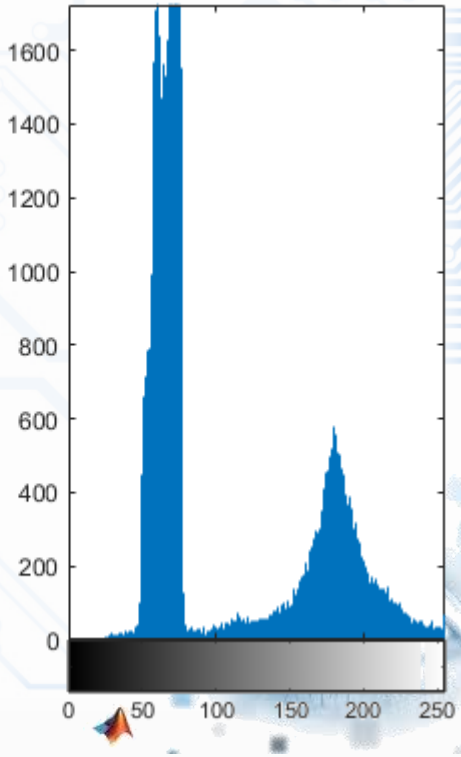
像素分布：直方图

zhí fāng tú

```
close all
clc
clear all
I=imread('coins.png'); % Read in image
subplot(1,2,1), imshow(I); % Display image
subplot(1,2,2), imhist(I); % Display histogram
```

where we see a histogram plot with two distinctive peaks: a high peak in the lower range of pixel values corresponds to the background intensity distribution of the image and a lower peak in the higher range of pixel values (bright pixels) corresponds to the foreground objects (coins).

其中，我们看到具有两个不同峰值的直方图图：较低像素值范围内的高峰值对应于图像的背景强度分布，较高像素值范围内的低峰值（明亮像素）对应于前景对象（硬币）。

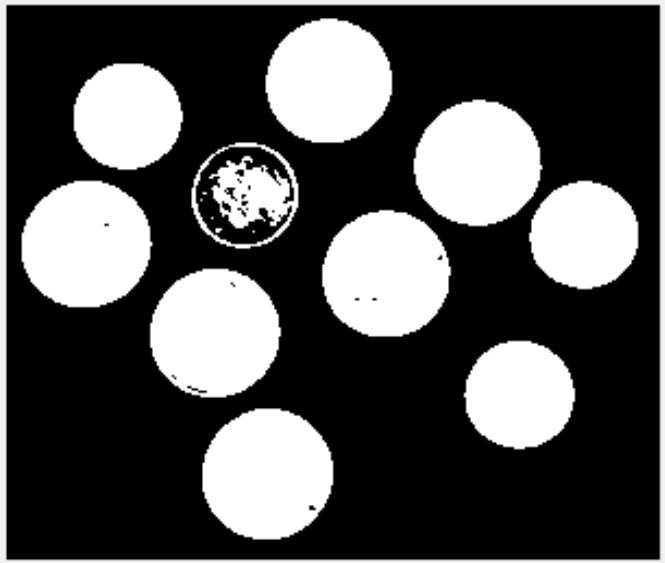


Histograms for threshold selection

阈值选择的直方图

- In Matlab, we can use the image histogram as the basis for calculation of an automatic threshold value. 在Matlab中，我们可以使用图像直方图作为计算自动阈值的基础。
- The function graythresh in Example exploits the Otsu method, which chooses that threshold value which minimizes the interclass statistical variance of the thresholded black and white pixels. 示例中的函数graysthresh利用了Otsu方法，该方法选择的阈值使阈值黑白像素的类间统计方差最小化。

```
close all
clc
clear all
I=imread('coins.png'); % read in image
level = graythresh(I); % get OTSU theshold
It = im2bw(I, level);  % theshold image
imshow(It);            % display it
```



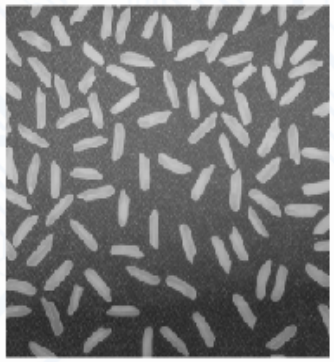
Adaptive thresholding

自适应阈值 zì shì yīng yù zhí

Adaptive thresholding is designed to overcome the limitations of conventional, global thresholding by using a different threshold at each pixel location in the image. This local threshold is generally determined by the values of the pixels in the neighbourhood of the pixel. Thus, adaptive thresholding works from the assumption that illumination may differ over the image but can be assumed to be roughly uniform in a sufficiently small, local neighbourhood

自适应阈值是为了克服传统全局阈值的局限性而设计的，它在图像中的每个像素位置使用不同的阈值。该局部阈值通常由像素附近的像素值确定。
因此，自适应阈值的工作原理是假设图像上的照明可能不同，但可以假设在足够小的局部邻域中大致均匀

```
close all
clc
clear all
I=imread('rice.png'); % read in image
Im=imfilter(I,fspecial('average',[15 15]),'replicate'); % create mean image
It = I - (Im + 20); % subtract mean image (+ constant C=20)
Ibw=im2bw(It,0);    % threshold result at 0 (keep +ve results only)
subplot(1,2,1), imshow(I); % Display image
subplot(1,2,2), imshow(Ibw); % Display result
```



Contrast stretching

对比拉伸 duì bǐ lā shēn

- Image histograms are also used for contrast stretching (also known as normalization) which operates by stretching the range of pixel intensities of the input image to occupy a larger dynamic range in the output image.

图像直方图还用于对比度拉伸（也称为标准化），其通过拉伸输入图像的像素强度范围来操作，以在输出图像中占据更大的动态范围。

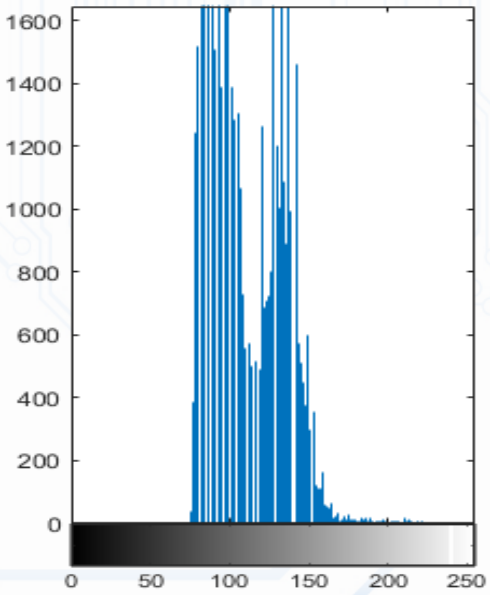
```
close all
clc
clear all
I=imread('pout.tif'); % read in image
Ics = imadjust(I,stretchlim(I, [0.05 0.95]),[]); % stretch
contrast using method 1
subplot(1,2,1), imshow(I); % display image
subplot(1,2,2), imshow(Ics); % display result
```

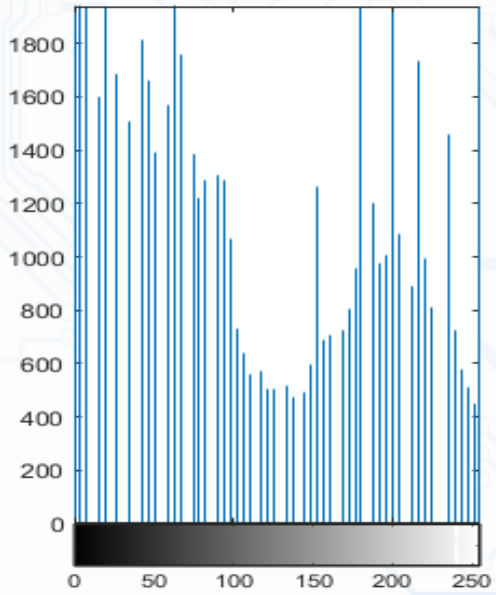
Here we use the stretchlim() function to identify the c and d pixel values at the 5th and 95th percentile points of the (normalized) histogram distribution of the image. The imadjust() function is then used to map the this range to the (default) maximum quantization range of the image

在这里，我们使用stretchlim（）函数来识别图像（标准化）直方图分布的第5个和第95个百分位点处的c和d像素值。然后使用imadjust（）函数将此范围映射到图像的（默认）最大量化范围

Contrast stretching

```
close all
clc
clear all
I=imread('pout.tif'); % read in image
Ics = imadjust(I,stretchlim(I, [0.05 0.95]),[]);
% stretch contrast using method 1
figure();subplot(1,2,1), imshow(I); % display image
subplot(1,2,2), imshow(Ics); % display result
figure();subplot(1,2,1), imhist(I); % display input hist.
subplot(1,2,2), imhist(Ics); % display output hist.
```





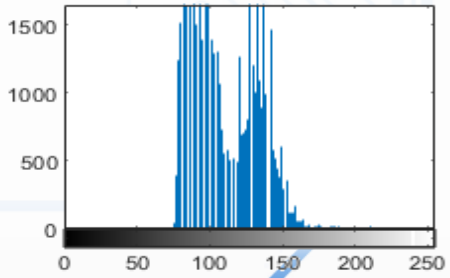
Histogram equalization

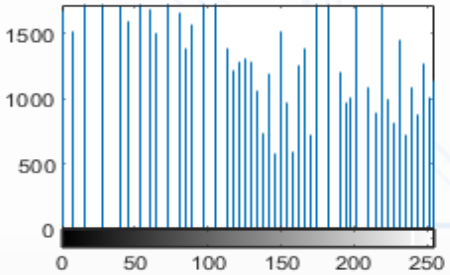
直方图均衡 zhí fāng tú jun héng

The second contrast enhancement operation based on the manipulation of the image histogram is histogram equalization. This is one of the most commonly used image enhancement techniques.

第二种基于图像直方图操作的对比度增强操作是直方图均衡化。这是最常用的图像增强技术之一。

```
close all
clc
clear all
I=imread('pout.tif'); % read in image
Ieq=histeq(I);
subplot(2,2,1), imshow(I); % display image
subplot(2,2,2), imshow(Ieq); % display result
subplot(2,2,3), imhist(I); % display hist. of image
subplot(2,2,4), imhist(Ieq); % display hist. of result
```



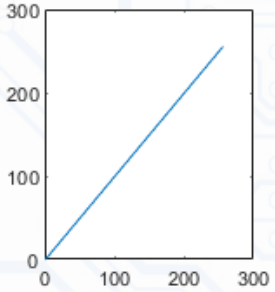


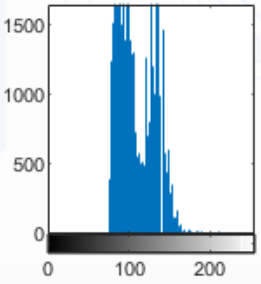
Histogram matching in practice

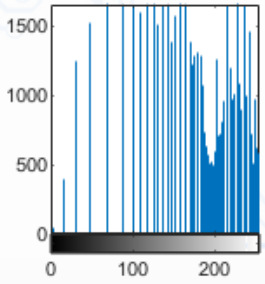
实践中的直方图匹配

```
close all
clc
clear all
I=imread('pout.tif');
pz=0:255;
% Define ramp-like pdf as desired output histogram
Im=histeq(I, pz);
 % supply desired histogram to perform matching
subplot(2,3,1), imshow(I); % display image
subplot(2,3,2), imshow(Im); % display result
subplot(2,3,3), plot(pz); % display distribution t
subplot(2,3,4), imhist(I); % display hist. of image
subplot(2,3,5), imhist(Im); % display hist. of result
```

Histogram matching extends the principle of histogram equalization by generalizing the form of the target histogram. It is an automatic enhancement technique in which the required transformation is derived from a (user-) specified target histogram distribution.
In practice, the target histogram distribution t will be extracted from an existing reference image or will correspond to a specified mathematical function with the correct properties. In Matlab, histogram matching can be achieved as in Example 3.20. In this example, we specify a linearly ramped PDF as the target distribution t and we can see the resulting output together with the modified image histograms in Figure







Adaptive histogram equalization applied to a sample image

应用于样本图像的自适应直方图均衡

```
clc
clear all
close all
I = imread('pout.tif');  % read in image
I1 = adapthisteq(I,'clipLimit',0.02,'Distribution','rayleigh');
I2 = adapthisteq(I,'clipLimit',0.02,'Distribution','exponential');
I3 = adapthisteq(I,'clipLimit',0.08,'Distribution','uniform');
subplot(2,2,1), imshow(I); subplot(2,2,2), imshow(I2); % diplay orig. + output
subplot(2,2,3), imshow(I2); subplot(2,2,4), imshow(I3); % diplay outputs
```

Histogram operations on colour images

彩色图像的直方图操作

- Until this point, we have only considered the application of our contrast manipulation operations on single-channel, grey-scale images. Attempting to improve the contrast of colour images is a slightly more complex issue than for grey-scale intensity images. At first glance, it is tempting to consider application of histogram equalization or matching independently to each of the three channels (R,G,B) of the true colour image. However, the RGB values of a pixel determine both its intensity and its chromaticity (i.e. the subjective impression of colour). Transformation of the RGB values of the pixels to improve contrast will, therefore, in general, alter the chromatic (i.e. the colour hue) content of the image.

- 到目前为止，我们只考虑了对比度操作在单通道灰度图像上的应用。试图改善彩色图像的对比度比灰度图像的对比度稍微复杂一些。乍一看，很容易考虑直方图均衡化或独立于真彩色图像的三个通道（R，G，B）中的每一个的匹配。然而，像素的RGB值决定其强度和色度（即颜色的主观印象）。因此，为了提高对比度而变换像素的RGB值通常会改变图像的色度（即色调）内容。

Histogram operations on colour images

彩色图像的直方图操作

```
close all
clc
clear all
I=imread('autumn.tif'); % Read in image
Ihsv=rgb2hsv(I); % Convert original to HSV image, I2
V=histeq(Ihsv(:,:,3)); % Histogram equalise V (3rd) channel of I2
Ihsv(:,:,3)=V; % Copy equalised V plane into (3rd) channel I2
Iout=hsv2rgb(Ihsv); % Convert I2 back to RGB form
subplot(1,2,1), imshow(I);
subplot(1,2,2), imshow(Iout);
```

Student Task_06: DIP

- 请帮我翻译部分的朋友鼓掌
- Qǐng bāng wǒ fānyì bùfèn de péngyǒu gǔzhǎng

Repeat all the examples in this PPT with your personal photo

用个人照片重复此PPT中的所有示例

Yòng gèrén zhàopiàn chóngfù cǐ PPT zhōng de suǒyǒu shìlì

Solve the Question shared in mooc

解决mooc分享的问题

Send for Next lecture

发送下一个讲座

THE DISTANCE
BETWEEN YOUR
DREAMS AND REALITY
IS DISCIPLINE.

MAY-BRITT MOSER
Nobel Prize in Physiology or Medicine 2014

"I learned at an early age that work makes you happy."

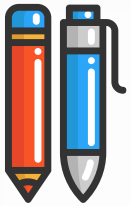
Reference

1. www.mathworks.com/products/**image**/
2. www.mathtools.net/MATLAB/Imaqe Processing/
3. Fundamentals of Digital Image Processing A Practical Approach with Examples in Matlab Chris Solomon, Kent, Canterbury,
4. Introduction to MATLAB, Kadin Tseng,Boston University, Scientific Computing and Visualization
5. Images taken from Gonzalez & Woods, Digital Image Processing (2002)

江西理工大学
Jiangxi University of Science and Technology
信息工程学院
School of information engineering

Digital Image Processing

THANK YOU



"BE HUMBLE. BE HUNGRY. AND ALWAYS BE THE HARDEST WORKER IN THE ROOM."