# 一 matlab图像处理工具箱

Image process toolbox. 含有一系列图像算法函数.

Imread

A = imread(filename,fmt)

Fmt可以是bmp,jpg,jpeg,tif等.

Imwrite

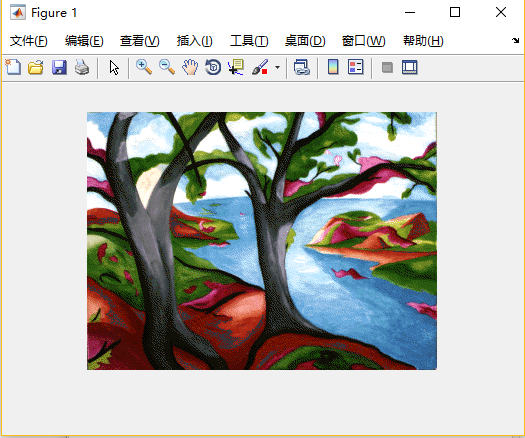
Imwrite(A,filename,fmt)

## 索引图转成灰度图

|  |
| --- |
| load trees  i5=ind2gray(X,map)  imshow(X,map) |

### 何为ind2gray

|  |
| --- |
| function I = ind2gray(varargin)  **##　索引图（map）转成灰度图.**   1. **索引图是为了解决老式显示器不能显示24位真彩色的问题.** 2. **伴随图像有一个map文件, 颜色由它定义.** 3. **根据索引值(map)找到最终颜色** 4. **索引图像是一种把像素值直接作为RGB调色板下标的图像(?)**   %IND2GRAY Convert indexed image to intensity image.  % I = IND2GRAY(X,MAP) converts the image X with colormap MAP  % to an intensity(**强度**) image I. IND2GRAY removes the hue and  % saturation information while retaining the luminance.  **%% 只保留lum信息, 丢失掉hue(色度)和saturation(饱和度)**   1. **色度**   **是物体反射光中占优势的波长决定,是由不同波长产生的不同感觉,是决定颜色本质的基本特征.**   1. **饱和度** 2. **是颜色鲜明程度, 深绿,深红等.** 3. **白色光越少,饱和度越大.** 4. **白色光越多,饱和度越小.**   %  % Class Support  % -------------  % X can be uint8, uint16, double, or single. **MAP is double**.  % I has the same class as X.  %  % Example  % -------  % load trees  % I = ind2gray(X,map);  % figure, imshow(X,map), figure, imshow(I);  %  % See also GRAY2IND, IMTOOL, IMSHOW, RGB2NTSC, RGB2GRAY, MAT2GRAY.    % Copyright 1992-2013 The MathWorks, Inc.      [a,cm] = parse\_inputs(varargin{:});    %initialize output matrix  I = a;    % calculate gray colormap  graycm = rgb2gray(cm); **%% rgb转成灰度colormap.**  graycm = graycm(:,1);  **%% 由rgb2gary后,graycm是128x3的,但是这3列每行都是同一个数值.**    % do transformation  if isa(a,'double') || isa(a,'single')  % Make sure A is in the range from 1 to size(cm,1)  a = max(1,min(a,length(graycm))); **%把原图a,大小限定在128以内.a是258x350的.因为colormap总共有128个.你的图像的值,不能超过128的索引.**  **%%从中索引出图像.**   1. **A的元素就是colormap的索引.** 2. **Colormap存的是灰度值.**   **I(:) = graycm(a);**  else  %convert graycm to appropriate class  graycm = images.internal.changeClass(class(a),graycm);    % get vector size for class  if isa(a,'uint8')  vs = 256;  else  vs = 65536;  end    % lut is equal to the cropped graycm (if longer than the vs for class)  % or the padded graycm (if smaller than the vs for class).  len = length(graycm);  lut = graycm(1:min(len,vs));  lut(len:vs) = graycm(end);    I(:) = intlut(a,lut);  end |



## 索引图二值化显示

|  |
| --- |
| load trees  BW6 = im2bw(X,map,0.4) %阈值0.4的转换.  imshow(X,map), title("Ë÷ÒýÍ¼")  figure, imshow(BW6), title("¶þÖµ»¯Í¼") |

### 何为im2bw

|  |
| --- |
| function BW = im2bw(varargin)  %IM2BW Convert image to binary image by thresholding.  **## img转二值化的图(含有0,1)**   1. **Img可以是索引图(indexed), intensity, RGB图.** 2. **二值化根据传入的阈值, 小于阈值的为0, 大于阈值的为1.**   % IM2BW is not recommended. Use IMBINARIZE instead.  %  % IM2BW produces binary images from indexed, intensity, or RGB images. To do  % this, it converts the input image to grayscale format (if it is not already  % an intensity image), and then converts this grayscale image to binary by  % thresholding. The output binary image BW has values of 1 (white) for all  % pixels in the input image with luminance greater than LEVEL and 0 (black)  % for all other pixels. (Note that you specify LEVEL in the range [0,1],  % regardless of the class of the input image.)  %  % BW = IM2BW(I,LEVEL) converts the intensity image I to black and white.  %  % BW = IM2BW(X,MAP,LEVEL) converts the indexed image X with colormap MAP to  % black and white.  %  % BW = IM2BW(RGB,LEVEL) converts the RGB image RGB to black and white.  %  % Note that the function GRAYTHRESH can be used to compute LEVEL  % automatically.  %  % Class Support  % -------------  % The input image can be uint8, uint16, single, int16, or double and it  % must be nonsparse. The output image BW is logical. I and X must be 2-D.  % RGB images are M-by-N-by-3.    %  % Example  % -------  % load trees  % BW = im2bw(X,map,0.4);  % figure, imshow(X,map), figure, imshow(BW)  %  % See also IMBINARIZE, GRAYTHRESH, OTSUTHRESH, ADAPTTHRESH, IND2GRAY,  % RGB2GRAY.    % Copyright 1992-2016 The MathWorks, Inc.    [A,map,level] = parse\_inputs(varargin{:});  **##　根据img的维度判断传入的img的类似．**   1. **Dim是3,img是RGB图.** 2. **Dim是2,img是indexed图(含有file和对应的map).** 3. **其他的是intensity(类似gray图).**   if ndims(A)==3,% RGB is given  A = rgb2gray(A);  elseif ~isempty(map),% indexed image is given  A = ind2gray(A,map);  end % nothing to do for intensity image    range = getrangefromclass(A); **%% 拿到img的元素的类型.**   * 1. **可以是**'numeric','logical','int64','uint64'   2. **double, single, or logical类型的会返回[0,1]**     if isinteger(A)  BWp = (A > range(2) \*level);    elseif islogical(A)  %A is already a binary image and does not require thresholding  warning(message('images:im2bw:binaryInput'))  BWp = A;  else % double or single  BWp = (A > level);  end    % Output:  if nargout==0 % Show results  imshow(BWp);  return;  end  BW = BWp; |

## 灰度图转索引图.

|  |
| --- |
| %% gray to indexed img  I = imread('lena.jpg')  %subplot(221);imshow(I)  Xnew = grayslice(I,16);  imshow(I)  figure, imshow(Xnew,jet(16)) |

### 何为grayslice

Grayslice用到了一个全局遍历,margin.

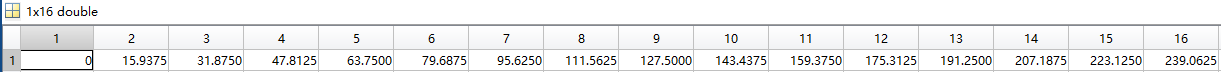
Margin描述了函数出入参数的个数.

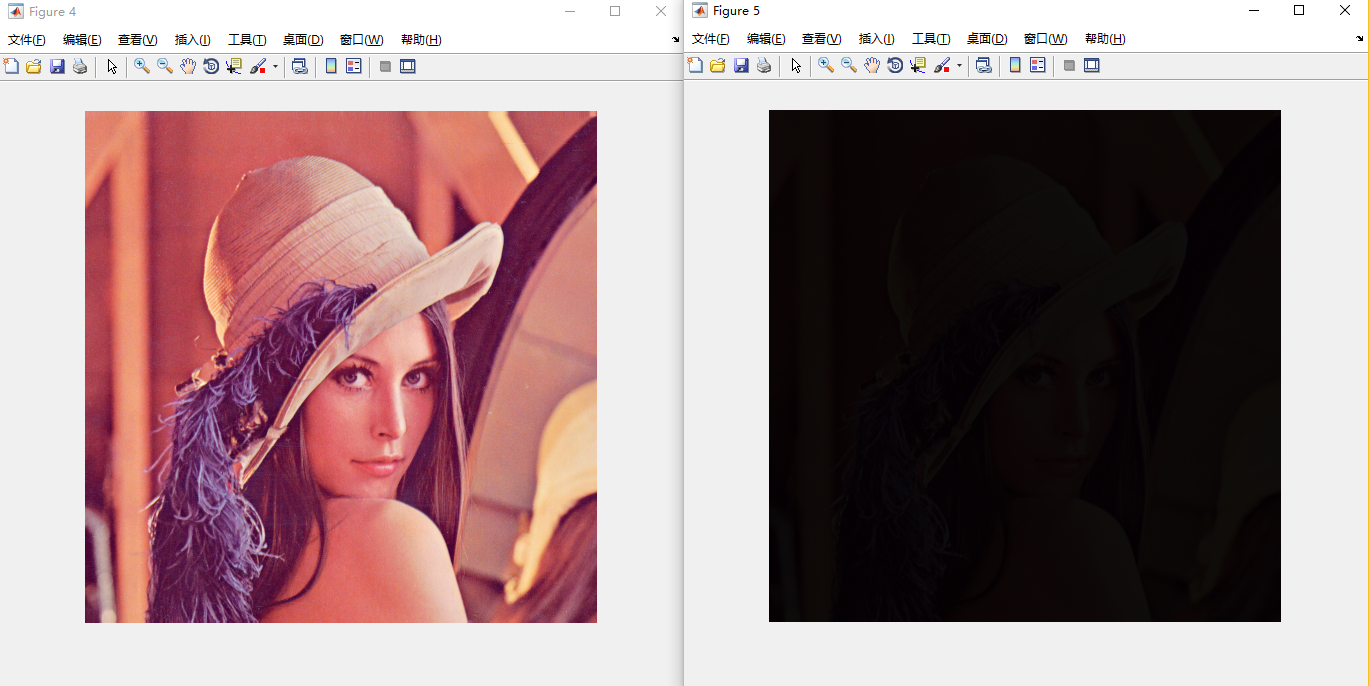
|  |
| --- |
| %NARGIN Number of function input arguments. |

|  |
| --- |
| function bout=grayslice(I,z)  **%% 会处理两种情况:**   1. **传入N(描述了要用多少个阈值来生成map图).** 2. **传入vector(值在0和1之间), 描述的是阈值,图像会用vector的值作为阈值,返回索引图像.**   %GRAYSLICE Create indexed image from intensity image by thresholding.  % X=GRAYSLICE(I,N) thresholds the intensity image I using threshold values  % 1/n, 2/n, ..., (n-1)/n, returning an indexed image in X.  %  % X=GRAYSLICE(I,V), where V is a vector of values between 0 and 1, thresholds  % I using the values of V as thresholds, returning an indexed image in X.  %  % You can view the thresholded image using IMSHOW(X,MAP) with a colormap of  % appropriate length.  %  % Class Support  % -------------  % The input image I can uint8, uint16, int16, single or double, and must be  % nonsparse. Note that the threshold values are always between 0 and 1, even  % if I is of class uint8 or uint16. In this case, each threshold value is  % multiplied by 255 or 65535 to determine the actual threshold to use.  %  % The class of the output image X depends on the number of threshold values,  % as specified by N or length(V). If the number of threshold values is less  % than 256, then X is of class uint8, and the values in X range from 0 to N or  % length(V). If the number of threshold values is 256 or greater, X is of  % class double, and the values in X range from 1 to N+1 or length(V)+1.  %  % Example  % -------  % Use multilevel thresholding to enhance high intensity areas in the image.  %  % I = imread('snowflakes.png');  % X = grayslice(I,16);  % figure, imshow(I), figure, imshow(X,jet(16))  %  % See also GRAY2IND.    % Copyright 1993-2016 The MathWorks, Inc.    validateattributes(I,{'double','uint8','uint16','int16','single'},{'nonsparse','real'}, ...  mfilename,'I',1);    if nargin == 1 %% 如果传入了一个参数(只有img), 默认分成10的index.  z = 10;  else  validateattributes(z,{'double','uint8','uint16','int16','single'},{'nonsparse','real'}, ...  mfilename,'z',2);  if ~isa(z,'double')  z = double(z);  end  end    % Convert int16 data to uint16.  if isa(I,'int16')  I = int16touint16mex(I);  end    range = getrangefromclass(I);    if ( (numel(z) == 1) && ((round(z)==z) || (z>1)) )  % arg2 is scalar: Integer number of equally spaced levels.  n = z;  if isinteger(I)  z = range(2) \* (0:(n-1))/n;  **%% range是[1,255] range(2)是255**   1. **255\*(0, 1/16, 2/16, …, 15/16)**     else % I is double or single  z = (0:(n-1))/n;  **%% 生成分段的z值.**  end  else  % arg2 is vector containing threshold levels  n = length(z)+1;  if isinteger(I)  % uint8 or uint16  zmax = range(2);  zmin = range(1);  else  % double or single  maxI = max(I(:));  minI = min(I(:));  % make sure that zmax and zmin are double  zmax = max(1,double(maxI));  zmin = min(0,double(minI));  end  newzmax = min(zmax,sort(z(:)));  newzmax = newzmax';  newzmax = max(zmin,newzmax);  z = [zmin,newzmax]; % sort and threshold z  end    % Get output matrix of appropriate size and type  if n < 256  b = repmat(uint8(0), size(I));  else  b = zeros(size(I));  end    % Loop over all intervals, except the last  for i = 1:length(z)-1  % j is the index value we will output, so it depend upon storage class  **%% j是输出的位图的索引值.**   1. **找到I的元素值在z(i)和z(i+1)之间的元素, 返回他们的位置.** 2. **在file文件中要保存将在在map中查表的index值, 把上面的位置都极为j(对于uint8的,j=i-1的值)** 3. **将来显示的时候,根据b中每个位置的值,把这个值看成map的索引值, 而map中保存着,对应索引值的灰度值.**     if isa(b,'uint8')  j = i-1;  else  j = i;  end  d = find(I>=z(i) & I<z(i+1));  if ~isempty(d),  b(d) = j;  end  end    % Take care of that last interval  d = find(I >= z(end));  if ~isempty(d)  % j is the index value we will output, so it depend upon storage class  if isa(b, 'uint8'),  j = length(z)-1;  else  j = length(z);  end  b(d) = j;  end    if nargout == 0  imshow(b,jet(n))  return  end  bout = b; |

### Grayslice实验

中途计算的z值





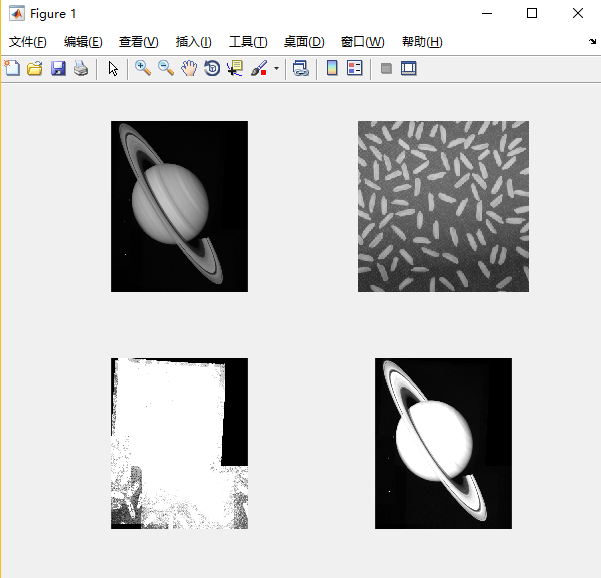
# 二 图像代数运算

## 图像相加

|  |
| --- |
| function ImgAdd  Ibackground=imread('pears.png');%¶ÁÈëµÚÒ»¸öÍ¼Ïñ  Ibackground=imresize(Ibackground,[200,200]);%µ÷ÕûÍ¼Ïñ³ß´ç´óÐ¡  subplot(221),imshow(Ibackground);  title('Í¼1');  J=imread('peppers.png');%¶ÁÈëµÚ¶þ¸öÍ¼Ïñ  J=imresize(J,[200,200]);%确保 两个图的尺寸是一样的.  subplot(222),imshow(J);  title('Í¼2');  K1=**imadd**(Ibackground,J);% 相加  subplot(223),imshow(K1);  title('Í¼ÏñÓëÍ¼ÏñÏà¼Ó');  K2 = imadd(J,100);  subplot(224),imshow(K2);  title('Í¼ÏñÓë³£ÊýÏà¼Ó'); |

## 图像相乘

|  |
| --- |
| function ImgSub  img1=imread('saturn.png');%¶ÁÈëµÚÒ»¸öÍ¼Ïñ  img1=rgb2gray(img1);  subplot(221),imshow(img1);  img2=imread('rice.png')  %img2=rgb2gray(img2); %±¾À´ÊÇgrayµÄ  subplot(222),imshow(img2);  [h,w]=size(img1)  img2=imresize(img2,[h,w])  img\_mul1 = **immultiply**(img1,img2)  subplot(223),imshow(img\_mul1 );  img\_mul2 = immultiply(img1,2)  subplot(224),imshow(img\_mul2 );  %% **immultiply 两种用法**   1. 传入两个图,同尺寸,两个img的对应的元素相乘(max clip为225),需要是gray图. 2. 传入图和倍数, 对应位置扩大常数倍数. |



## 图像的逻辑运算

如下的逻辑运算:

1. 一个前提是,图像的逻辑运算包含: 逻辑与,逻辑或,非,异或.
2. 所以要求图像是0,1的二进制,不然和逻辑运算无关.
3. 采用im2bw函数二值化.
4. 还要求两幅图的尺寸一致.

与:

Output = img1 & img2

或:

Output = img1 | img2

非:

Output = ~img1

异或:

Output = xor(img1 ,img2)