Image Processing with MATLAB

Lecture 4: Images Enhancement Techniques

Dr.Eng. Hassan Mohamed

Hassan.hussein@feng.bu.edu.eg

Lecture Contents:

1. Pixel Values and Statistics
   - Data values for selected pixels (pixval, impixel)
   - Data values along a path in an image (improfile)
   - Contour plot of the image data (imcontour)
   - Histogram of the image data (imhist)
   - Summary statistics for the image data (mean2, std2, corr2)
   - Feature measurements for image regions (regionprops)
2. Intensity Adjustment
   - Adjusting Intensity Values to a Specified Range
   - Histogram Equalization
   - Contrast-Limited Adaptive Histogram Equalization

Image Types in the Toolbox:

Term	Definition
adaptive filter	Filter whose properties vary across an image depending on the local characteristics of the image pixels.
contour	Path in an image along which the image intensity values are equal to a constant.
edge	Curve that follows a path of rapid change in image intensity. Edges are often associated with the boundaries of objects in a scene. Edge detection is used to identify the edges in an image.
property	Quantitative measurement of an image or image region. Examples of image region properties include centroid, bounding box, and area.
histogram	Graph used in image analysis that shows the distribution of intensities in an image. You can use the information in a histogram to choose an appropriate enhancement operation. For example, if an image histogram shows that the range of intensity values is small, you can use an intensity adjustment function to spread the values across a wider range.
noise	Errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene.
profile	Set of intensity values taken from regularly spaced points along a line segment or multiline path in an image. For points that do not fall on the center of a pixel, the intensity values are interpolated.
quadtree decomposition	Image analysis technique that partitions an image into homogeneous blocks.

Getting Pixel Information with pixval:

- To use pixval, you first display an image and then enter the pixval command. pixval installs a black bar at the bottom of the figure, which displays the (x,y) coordinates for whatever pixel the cursor is currently over and the color data for that pixel.

Getting Pixel Information with impixel:

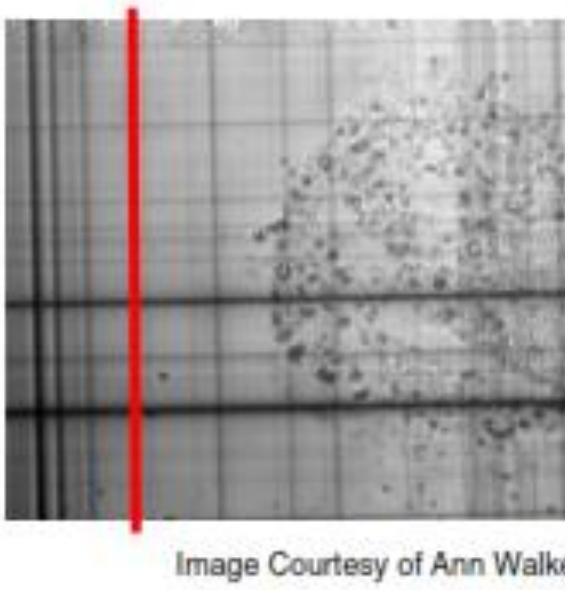
- impixel provides pixel information in a variable in the MATLAB workspace. If you call impixel with no input arguments, the function associates itself with the image in the current axes.
- **Return**: impixel returns the color values for the selected pixels, and the stars disappear.
- *imshow canoe.tif*
- *vals = impixel*

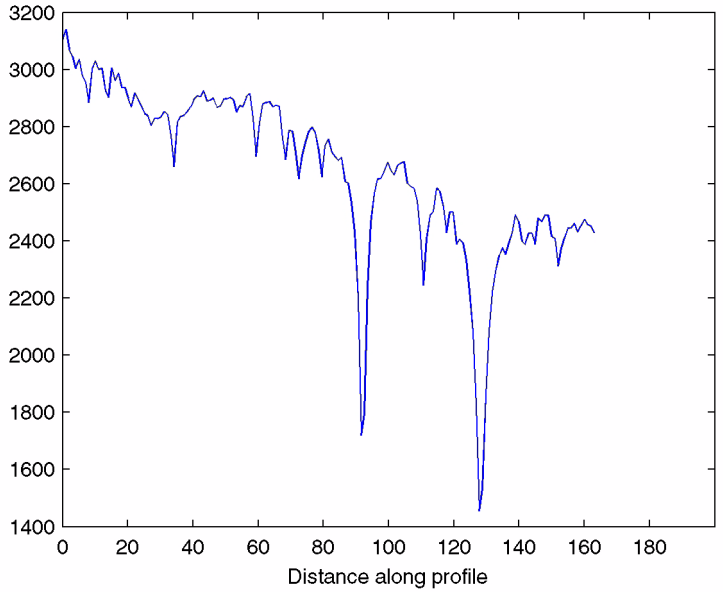
```
vals =

    0.1294    0.1294    0.1294
    0.5176         0         0
    0.7765    0.6118    0.4196
```

Intensity Profile:

- The improfile function calculates and plots the intensity values along a line segment or a multiline path in an image. You can supply the coordinates of the line segments as input arguments, or you can define the desired path using a mouse. In either case, improfile uses interpolation to determine the values of equally spaced points along the path.
- *I = fitsread('solarspectra.fts');*
- *imshow(I,[])*
- *improfile*





RGB Image with Line Segment Drawn with improfile:

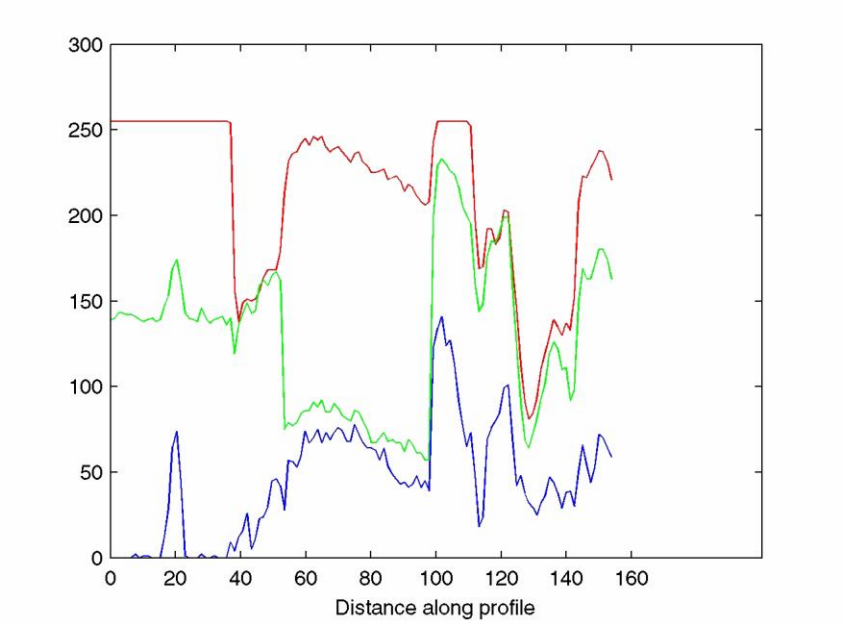
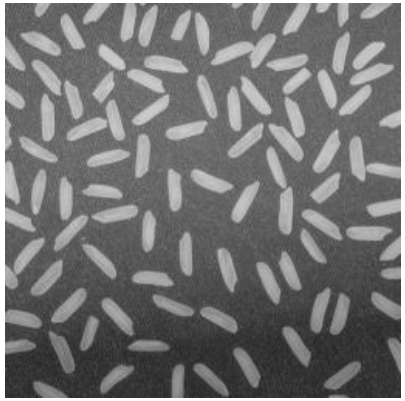


Image Contours:

- You can use the toolbox function imcontour to display a contour plot of the data in an intensity image. This function is similar to the contour function in MATLAB, but it automatically sets up the axes so their orientation and aspect ratio match the image.
- *I = imread('rice.png');*
- *imshow(I)*
- *figure, imcontour(I,3)*



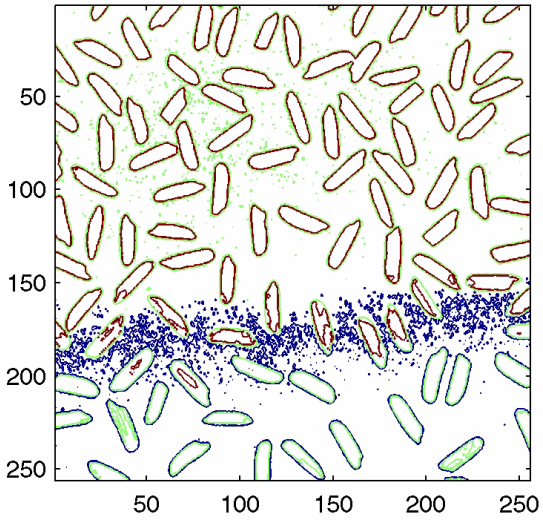
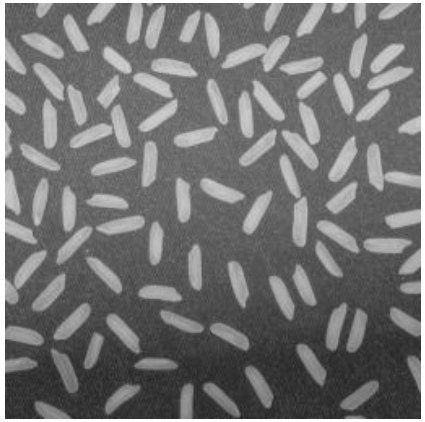
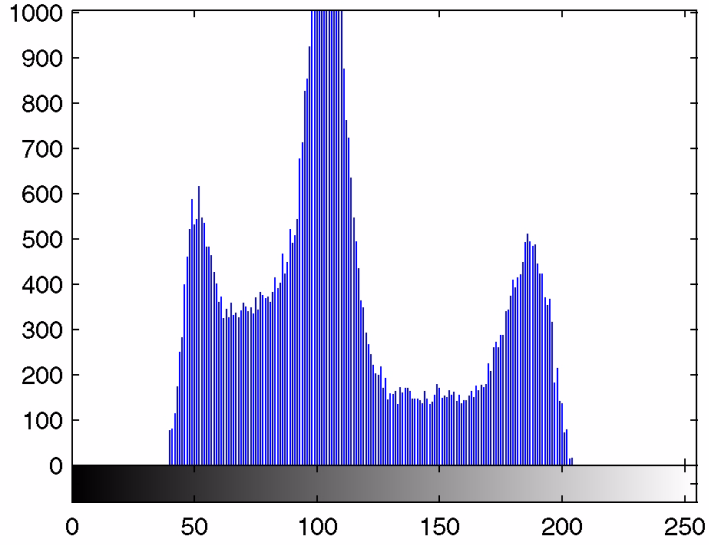


Image Histogram:

- An image histogram is a chart that shows the distribution of intensities in an indexed or intensity image. The image histogram function imhist creates this plot by making n equally spaced bins, each representing a range of data values. It then calculates the number of pixels within each range.
- *I = imread('rice.png');*
- *imshow(I)*
- *figure, imhist(I)*



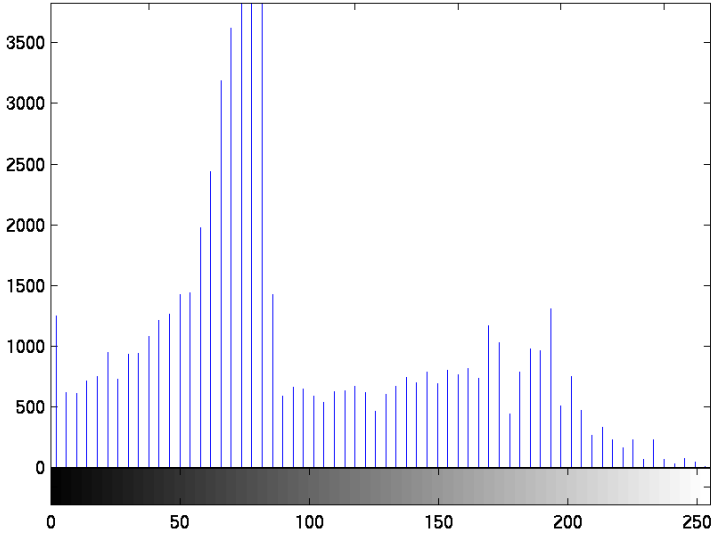


Summary Statistics & Region Property Measurement:

- You can compute standard statistics of an image using the *mean2, std2,* and *corr2* functions. *mean2* and *std2* compute the mean and standard deviation of the elements of a matrix. *corr2* computes the correlation coefficient between two matrices of the same size.
- You can use the *regionprops* function to compute properties for image regions. For example, *regionprops* can measure such properties as the area, center of mass, and bounding box for a region you specify.

Adjusting Intensity Values to a Specified Range:

- You can adjust the intensity values in an image using the imadjust function, where you specify the range of intensity values in the output image.
- *I = imread('pout.tif');*
- *J = imadjust(I);*
- *imshow(J)*
- *figure,*
- *imhist(J,64)*



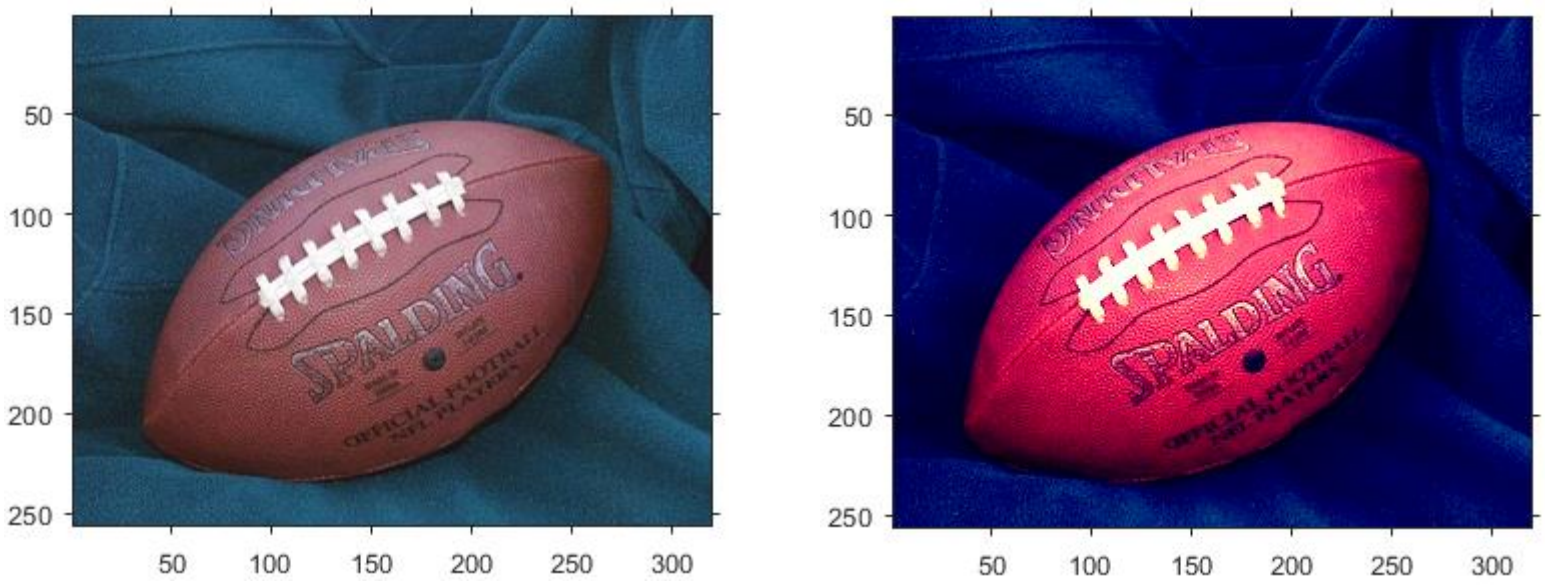
Specifying the Adjustment Limits:

- *I = imread('cameraman.tif');*
- *J = imadjust(I,[0 0.2],[0.5 1]);*
- *imshow(I)*
- *figure, imshow(J)*

Image Courtesy of MIT

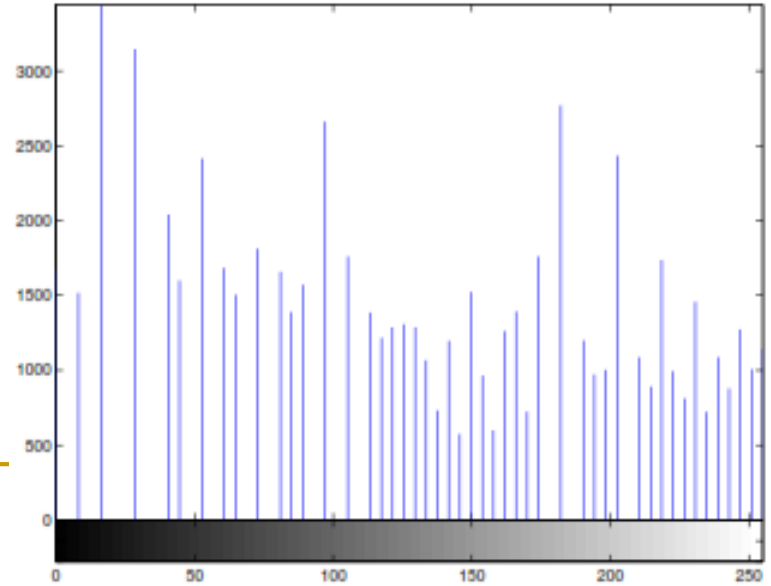
Adjust Contrast of RGB Image:

- Adjust the contrast of the RGB image, specifying contrast limits.
- *RGB2 = imadjust(RGB,[.2 .3 0; .6 .7 1],[]);*
- *figure*
- *imshow(RGB2)*



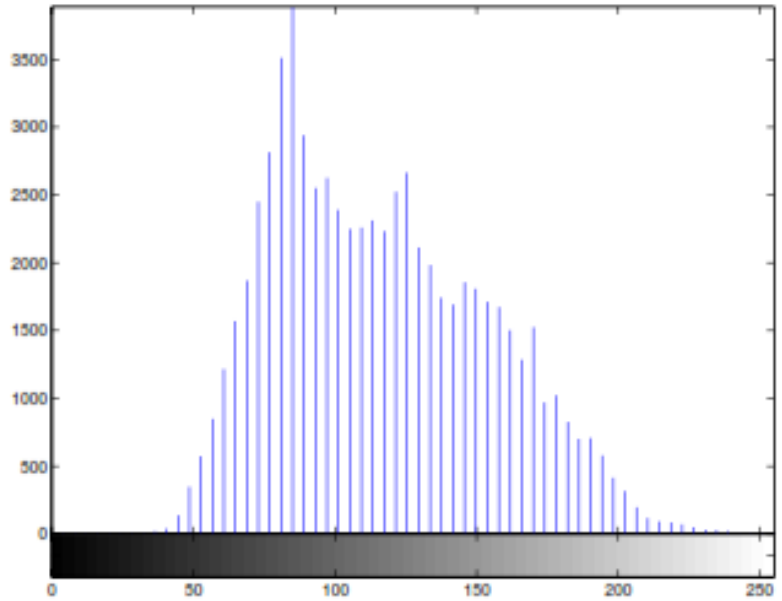
Histogram Equalization:

- The process of adjusting intensity values can be done automatically by the histeq function. histeq performs *histogram equalization, which involves* transforming the intensity values so that the histogram of the output image approximately matches a specified histogram. (By default, histeq tries to match a flat histogram with 64 bins, but you can specify a different histogram instead.
- *I = imread('pout.tif');*
- *J = histeq(I);*
- *imshow(J)*
- *figure, imhist(J,64)*



Contrast-Limited Adaptive Histogram Equalization:

- As an alternative to using *histeq*, you can perform contrast-limited adaptive histogram equalization (CLAHE) using the *adapthisteq* function. While *histeq* works on the entire image, *adapthisteq* operates on small regions in the image, called tiles. Each tile's contrast is enhanced, so that the histogram of the output region approximately matches a specified histogram.
- *I = imread('pout.tif');*
- *J = adapthisteq(I);*
- *imshow(I)*
- *figure, imshow(J)*



Supplementary files:

- MATLAB Tutorial:

http://www.mathworks.com/products/matlab/matlab_tutorial.html

- MATLAB documentation:

http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml

Please don’t use this presentation without getting a permeation from its original owner

Dr.Eng. Hassan Mohamed