Digital Image Processing

Presented by:

Dr. Moe Moe Myint

Information Technology Department

Technological University (Kyaukse), Myanmar

moemoemyint@moemyanmar.ml

www.slideshare.net/MoeMoeMyint

- Only Original Owner has full rights reserved for copied images.
  - This PPT is only for fair academic use.

Dr. Moe Moe Myint
Information Technology Department
Technological University (Kyaukse)

Pixel Value and Statistic (Lab 5)

Create histograms, contour plots, and get statistics on image regions

Objectives

- To display histogram of image data and pixel color values

Required Equipment

- Computers with MATLAB software and Projector

Practical Procedures

- Read the image and display it
- Use *histeq* for enhancing the contrast of images
- Use *impixel* for pixel color values
- Use *improfile* for pixel-value cross-sections along line segments
- Use *imcontour* for create contour plot of image data

Pixel Values and Statistics

corr2	2-D correlation coefficient
imhist	**Display histogram of image data**
impixel	**Pixel color values**
improfile	**Pixel-value cross-sections along line segments**
imcontour	**Create contour plot of image data**
mean2	Average or mean of matrix elements
regionprops	Measure properties of image regions
std2	Standard deviation of matrix elements

Histogram of Image Data

- An image histogram is a chart that shows the distribution of intensities in an indexed or grayscale 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.
- **imhist(I)** displays a histogram for the image

Examples

Read image and display it.

```
I = imread('rice.png');
imshow(I)
figure, imhist(I)
```

adapthisteq

- Contrast-limited adaptive histogram equalization (CLAHE)

Syntax

J = adapthisteq(I)

J = adapthisteq(I,param1,val1,param2,val2...)

- **histeq** enhances the contrast of images by transforming the values in an intensity image, or the values in the colormap of an indexed image, so that the histogram of the output image approximately matches a specified histogram.

Examples

- Enhance the contrast of an intensity image using histogram equalization.

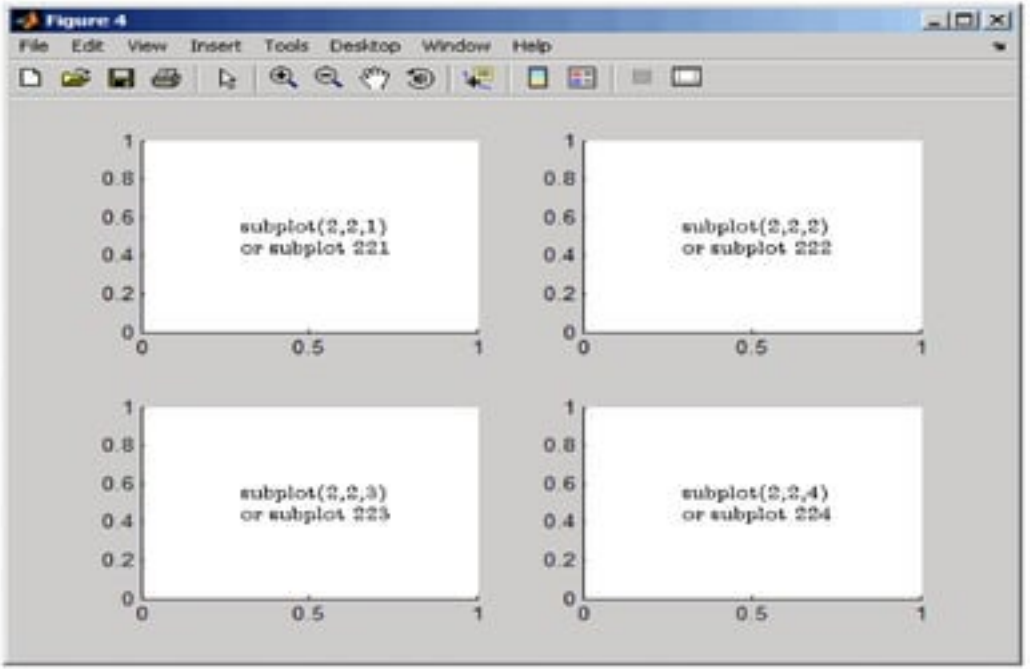
```
I = imread('tire.tif');
J1 = histeq(I);
J2 = adapthisteq(I);
imshow(I), title('Original Image');
figure, imshow(J1); title('Histogram Equalization');
figure, imshow(J2); title('Adaptive Histogram Equalization');
```

Example

```
clc,clear all, close all;
I=imread('pout.tif');
figure;
subplot(1,2,1);imshow(I);
subplot(1,2,2);imhist(I);
imh=imadjust(I,[0.3;0.6],[0.0,1.0]);
imh1=histeq(I);
figure;
subplot(2,2,1);imshow(imh);title('Stretched Image');
subplot(2,2,2);imhist(imh);
subplot(2,2,3);imshow(imh1);title('Histeq Image');
subplot(2,2,4);imhist(imh1);
```

subplot

- Create axes in tiled positions



- **impixel** returns the red, green, and blue color values of specified image pixels. In the syntax below, impixel displays the input image and ***waits for you to specify the pixels with the mouse.***

1. Display an image.

```
imshow canoe.tif
```

2. Call impixel. When called with no input arguments, impixel associates itself with the image in the current axes.

```
impixel
```

3. Select the points you want to examine in the image by clicking the mouse. impixel places a star at each point you select.

4. When you are finished selecting points, press Return. impixel returns the pixel values in an n-by-3 array, where n is the number of points you selected. The stars used to indicate selected points disappear from the image.

```
pixel_values =
0.1294   0.1294   0.1294
0.5176        0        0
0.7765   0.6118   0.4196
```

improfile

- The intensity profile of an image is the set of intensity values taken from regularly spaced points along a line segment or multi

```
I = fitsread('solarspectra.fts');

imshow(I,[]);

improfile
```

improfile displays a plot of the data along the line. Notice the peaks and valleys and how they correspond to the light and dark bands in the image.

The example below shows how improfile works with an RGB image. Use imshow to display the image in a figure window. Call improfile without any arguments and trace a line segment in the image interactively. In the figure, the black line indicates a line segment drawn from top to bottom. Double-click to end the line segment.

```
imshow peppers.png

improfile
```

The improfile function displays a plot of the intensity values along the line segment. The plot includes separate lines for the red, green, and blue intensities. In the plot, notice how low the blue values are at the beginning of the plot where the line traverses the orange pepper.

imcontour

- You can use the toolbox function imcontour to display a contour plot of the data in a grayscale image.
- This example displays a grayscale image of grains of rice and a contour plot of the image data:

1. Read a grayscale image and display it.

```
I = imread('rice.png');
imshow(I)
```

2. Display a contour plot of the grayscale image.

```
figure, imcontour(I,3)
```

Questions?