Multimedia Systems Lecture 4

LECTURER

MOUHSEN IBRAHIM

Contents

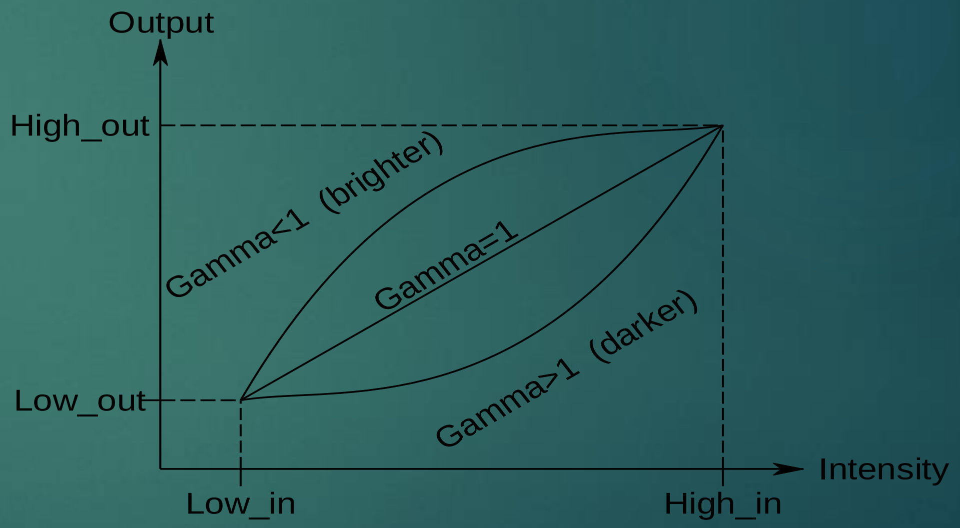
- imwrite Function
- Imadjust : gamma
- Thresholding
- Edge Detection
- Morphological Transformations

imwrite Function

- Write image to graphics file
- **Syntax**
  - *imwrite(A,filename,fmt)* writes the image A to the file specified by *filename* in the format specified by *fmt*
  - *imwrite(X,map,filename,fmt)* writes the indexed image in **X** and its associated colormap **map** to **filename** in the format specified by *fmt*
  - *imwrite(...,filename)* writes the image to filename
  - *imwrite(...,Param1,Val1,Param2,Val2...)* specifies parameters that control various characteristics of the output file. For example, if you are writing a JPEG file, you can specify the quality of the output image
- **Example**
  - A=imread('greens.jpg');
  - imwrite(A,'new_greens.jpg','Quality',25)
  - We choose the value of quality between 0 – 100
  - Compare between the two images in size and histogram.

Imadjust : gamma

- **Syntax**
  - J = imadjust(I,[low_in high_in],[low_out high_out],gamma)
- **Example**:
  - C=imread('circuit.tif');
  - imshow(C)
  - C_new=imadjust(C,[0 0.9],[],0.5);
  - imshow(C_new)



Thresholding

- To convert an image from gray scale to binary , we need to find a threshold (T) where values bigger than that threshold are white, and the rest is black
- Steps:
  - Convert the image to a gray scale image
  - The initial value of the T is calculated by

$$T = \frac{1}{2}\left(\min(L(:)) + \max(L(:))\right)$$

  - Depending on T value there are 2 sets of pixels

$$pixels = \begin{cases} G1 & f(x, y) \geq T \\ G2 & elsewhere \end{cases}$$

  - Calculate a new value of the threshold where
    - $T_{new} = {}^{1}/_{2}\,(Average_{G1} + Average_{G2})$
  - We repeat this operation until the difference between the last calculated thresholds is smaller than 0.2

Thresholding

- The next step is to give each pixel whose value is bigger than the threshold the value 1 (white) and the rest the value 0 (black)
- We can use the matlab function *graythresh to calculate the T* as it is shown in the following example.
    - gray_orginal=imread('cameraman.tif');
    - imshow(gray_orginal)
    - thresh=graythresh(gray_orginal);
    - binary=im2bw(gray_orginal,thresh);
    - imshow(binary)

Edge Detection

- Edge detection is an image processing technique for finding the boundaries of objects within images
- It works by detecting discontinuities in brightness
- Common edge detection algorithms include Sobel, Canny, Prewitt, Robert
- Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.
- In an image, an edge is a 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.
- To find edges, you can use the edge function. This function looks for places in the image where the intensity changes rapidly, using one of these two criteria:
  - Places where the first derivative of the intensity is larger in magnitude than some threshold
  - Places where the second derivative of the intensity has a zero crossing

Edge Detection

- I=imread(‘testpat1.tif ’);
- IEr = edge(I,‘roberts’);
- IEp = edge(I,‘prewitt’);
- IEs = edge(I,‘sobel’);
- IEc = edge(I,‘canny’);
- subplot(2,3,1), imshow(I); title(‘Original image’);
- subplot(2,3,2), imshow(IEr); title('modified image using Roberts filter');
- subplot(2,3,3), imshow(IEp); title('modified image using prewitt filter');
- subplot(2,3,4), imshow(IEs); title('modified image using sobol filter');
- subplot(2,3,5), imshow(IEc); title('modified image using canny filter');

Morphological Transformations

- Morphological transformations are some simple operations based on the image shape. It is normally performed on binary image
- **Strel**
  - Create morphological structuring element
- **Syntax**
  - SE = strel(*shape*,parameters)
- **Description**
  - creates a structuring element, SE, of the type specified by *shape*. Depending on *shape*, strel may take additional parameters.

Morphological Transformations

- **Dilation and Erosion:**
  - B=strel('square',4);
  - C=imread('circles.tif');
  - Imshow(C);
  - C2 = imdilate(C,B);
  - imshow(C2);
  - C3= imerode(C,B);

- **Opening and closing:**
  - d=strel('disk',5);
  - T=imread('circlesm.tif');
  - imshow(T);
  - T1=imopen(T,d);
  - imshow(T1);
  - T2=imclose(T,d);
  - Imshow(T2)

Morphological Transformations

- **Thinning and thickening:**
  - I= imread('testpat2.tif');
  - I2=bwmorph(I,'thin',5);
  - Imshow(I2);
  - I3=bwmorph(I,'thicken',5);
  - Imshow(I3)

- **Top-hat filtering**
  - F=imread('afmsurf.tif');
  - imshow(F);
  - se = strel('disk',15);
  - F1 = imtophat(F,se);
  - figure,imshow(F1);
  - K = imadjust(F1,stretchlim(F1));
  - imshow(K)

Exercise

- Remove the thin wires from the image “wires.gif ” and then remove the thick wires.
- Make the text in the image “text.bmp ” clearer
- Remove the noise from the image “finger_print.bmp”

THE END
GOOD LUCK