LAB#1 Image Processing Lab (Tuesday, Feb. 7, 2006, E192)

Objectives:
1. To be familiar with a typical machine vision system, illumination, general functions, etc.
2. To learn how to use a Machine Vision System to take an image and store an image to a disk.
3. To learn how to use MATLAB Image Processing Toolbox to do digital image processing.

Components:
A DVT 600 Vision System, a computer, a ring illuminator, a LED illumination array, and some mechanical parts.

Procedures:
1. An overall review and demonstration of the basic functions and operations of the DVT vision system – a typical industrial machine vision system.
2. Change different illumination conditions to observe how the illumination affects the quality of an image.
3. Take an image using the DVT system and save your image in “xx.bmp” or “xx.tif” (xx is your file name) in a floppy disk (will be provided by Dr. Du).

Lab Report/Homework Assignment:
After you take an image (for this particular lab, the image is provided by Dr. Du, see the image below), you can go to E213 to do the following assignments (if you have MATLAB Image Processing Toolbox in your own computer, you can do the assignment at home).

1. Use a 3 x 3, 5 x5, and 11 x 11 low-pass filtering mask to process your image and compare three images.
2. Use a 3 x 3 median filter to processing your image.
3. Design an edge detector to detect the contour of the part.

In your lab report, please
(1) show all images (including the original image) you get from procedure 3;
(2) give your commands about these images. E.g. whether or not these images are improved, the designed edge detector works good or not, etc.;
(3) attach all MATLAB codes that you wrote for the image processing.

Please read attached materials before you do the assignment.
Getting Started – MATLAB Image Processing Toolbox

Basic functions:
This handout introduces the basic tasks and commands of reading and displaying an image, adjusting its contrast, and writing it back to disk.

1. Read and Display an Image (Use your own image)
   >> Img=imread('a:\ bonemarr.tif'); % Load your image from your disk
   >> imshow(Img)   % Show your image in MATLAB

2. Check the Image Storage Information (Use your own image)
   >> whos    % See how myimg.tif is stored into the MATLAB workspace.
   % MATLAB responds with
   % Name       Size            Bytes    Class
   % Img      480x640            307200   uint8 array

   Where “Img” -- your image name; “480x640” -- image size in pixel; “307200” -- image data size in Bytes; “uint8 array” -- Convert data to unsigned 8-bit integers. (This is a MATLAB function. See the online MATLAB Function Reference for its reference page.)

3. Pixel Value & Euclidean Distance (Use your own image)
   >> pixval on
   This command turns on interactive display of pixel info (x, y, and intensity) in the current image. A black bar at the bottom of the figure displays the (x,y) coordinates for the pixel that the cursor is currently over and the intensity info for that pixel.

   If you click on the image and hold down the mouse button while you move the cursor, PIXVAL also displays the Euclidean distance between the point you clicked on and current cursor location. PIXVAL draws a line between these points to indicate the distance being measured. When you release the mouse button, the line and the distance display disappear.

   You can move the display bar by clicking on it and dragging it to another place in the figure.
   >> pixval off
   This command turns interactive display off in the current figure. You can also turn off the display by clicking the button on the right side of the display bar.
   >> Img
   This command shows all pixel values in the image. If you want see the pixel value at (45, 69), just type in:
   >> Img(45,69)
   If you want to see the pixel values between x =36,…,40, and y=54, …, 69), just type in:
   >> I(36:40,54:69)

4. Image Processing
   1) Histogram (Use image ‘pout.tif’)
      >> imhist(Img)       % Display histogram of image data.
>> histeq(Img) % Enhance contrast using histogram equalization.

2) Edge Detection (Use image ‘rice.tif’)
   a) “Sobel” has a 3 x 3 mask: \( h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \).
   b) “Prewitt” has a 3 x 3 mask: \( h = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \).
   c) “Roberts” has a 3 x 3 mask: \( h = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \).
      >> NewImg = edge(I,'prewitt');
      This command uses “prewitt” mask for edge detection, the new image named “NewImg”.

3) Change a gray image into a binary image (Black-White Image) (Use image ‘rice.tif’)
   >> level = graythresh(Img)
   >> BW = im2bw(I,level);
   >> imshow(BW)
   The first command computes a global threshold (level) that can be used to convert an intensity image to a binary image with \textit{im2bw}. level is a normalized intensity value that lies in the range \([0, 1]\). The graythresh function uses Otsu’s method, which chooses the threshold to minimize the intraclass variance of the black and white pixels.
   The second one converts an image to a binary image, based on threshold

4) Median Filter (Use image ‘eight.tif’)
   >> N = imnoise(Img,’salt & pepper’,0.02);
   >> figure, imshow(N)
   The above commands add ‘salt & pepper’ noise on the image where 0.02 is the noise density. Then the median filter is used to remove the noise by following command:
      >> B = medfilt2(N);
      >> figure, imshow(B)

5) Smoothing Filter (e.g. Low-pass filter)
   >> I = imread('eight.tif');
   >> J = imnoise(I,’gaussian’,0,0.005);
   >> K = wiener2(J,[5 5]);
   >> imshow(J)
   >> figure, imshow(K)
   Note: Please repeat this problem using [3 3] and [9 9] instead of [5 5] in the third commands

\(^1\text{wiener2 lowpass filters an intensity image that has been degraded by constant power additive noise. wiener2 uses a pixel-wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.}

---

5. Multiple-Image Display (Use these commands to display multiple images in one figure)
   >>subplot(1,2,1), subimage(Img1);
   >>subplot(1,2,2), subimage(Img2)

   Display multiple images in the same figure.

6. Write the Image (If you want to store the processed image, use this command)
   >>imwrite (I2, 'MyNewImg.png');

   Write the processed image I2 to a disk. MATLAB recognizes several file version, such as
   “.tif”, “.bmp”, “png”, “.jpg”, etc.

======================================================== Lab Report Information ==============

The lab report for Lab#1 is to hand in your processed images in “4. Image Processing” part. Please also give your own comments about the processed images, including

   (1) How to determine a threshold using image’s histogram;
   (2) In three edge detection methods, which one or two is better for that particular image;
   (3) Why the median filter is good for removing ‘salt-pepper’ type of noise;
   (4) How the mask size (e.g. 3 x 3, 5 x 5, etc.) affect the image smoothing? Why wiener filter
       is good to remove gaussian noise.

* Please refer to the MATLAB HELP for more information.
* Please refer to the materials that I hand out to you or go to the course website to learn the
  theoretical stuff behind these processing algorithms.