Multimedia
(Lab 03)

Spring, 2018

Department of Software

Yong Ju Jung (정용주)

가천대학교 소프트웨어학과
Gachon University

Summary

- In Previous Lab, you have exercised a basic pixel processing of images.

- In this lab, you will learn simple image processing techniques.
  - histogram equalization
  - image filtering techniques.

Useful Tip for Debug

- **Image Watch:
viewing in-memory images in the Visual Studio debugger**
  - You will learn how to visualize OpenCV matrices and images within Visual Studio 2012 (or better).

To bring up Image Watch, select View –> Other Windows –> Image Watch.

[Lab03-1]

- Histogram equalization to enhance the contrast of an image
  - Load a color image (using **cv::imread**)
  - Convert the original image to grayscale
    - cvtColor(src, src, CV_BGR2GRAY);
  - Equalize the Histogram by using the OpenCV function EqualizeHist
    - equalizeHist(src, dst);
  - Display the source and equalized images in a window (using **cv::imshow**)

Histogram Calculation

- Calculate & plot the histogram of your input image.

- calcHist(&image, 1, 0, Mat(), b_hist, 1, &histSize, &histRange, uniform, accumulate);
    - int histSize = 256; //from 0 to 255
    - Set the range of values (as we said, between 0 and 255)
        - float range[] = { 0, 256 } ; //the upper boundary is exclusive
        - const float* histRange = { range };
    - bool uniform = true; bool accumulate = false;

Draw a histogram

- // Draw the histogram
- int hist_w = 512; int hist_h = 400;
- int bin_w = cvRound((double) hist_w/histSize);
- Mat histImage(hist_h, hist_w, CV_8UC3, Scalar(0,0,0));

- normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, -1, Mat());

- for(int i = 1; i < histSize; i++)
  line(histImage, Point(bin_w*(i-1), hist_h - cvRound(b_hist.at<float>(i-1))) , Point(bin_w*(i), hist_h - cvRound(b_hist.at<float>(i))), Scalar(255, 0, 0), 2, 8, 0);

- namedWindow("calcHist Demo", WINDOW_AUTOSIZE);
- imshow("calcHist Demo", histImage);

[Lab03-2]

- Smoothing filter for an image
  - Download an image from E-Class. The file is Lena in gray-scale.
  - Load the image (using **cv::imread**)
  - Perform NxN smoothing (average) filtering for the image, as you learn in this lecture (see Figure in page 40 in lecture 08).
  - Display the result image in an OpenCV window (using **cv::imshow**)

[Lab03-3]

- Median filter for an image
  - Download an image from E-Class. The file is the noisy Lena image.
  - Load the image (using **cv::imread**)
  - Perform median filtering with NxN window for the image, as you learn in this lecture (slide #40 in lecture 08).
  - Display the result image in an OpenCV window (using **cv::imshow**)

[Lab03-4]

- Derivative filter to obtain image gradient for an image
  - Download a gray-scale image.
  - Load the image (using **cv::imread**).
  - Perform 3x3 **Sobel** filtering for the image.
  - Display the gradient magnitude map of the filter output in an OpenCV window (using **cv::imshow**)

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

[Lab03-5]

- Sharpening filter by using Laplacian filter
  - Download an image from E-Class. The file is Lena in gray-scale.
  - Load an image (using **cv::imread**)
  - Perform the sharpening filtering for the image by using a Laplacian filter, as you learn in this lecture (see Fig. 3.40 in page 39 in lecture 08).
  - Display the Laplacian image and the sharpened result image in OpenCV windows (using **cv::imshow**)

  - Try to use the blurred image as an input, which you have submitted for HW#1. Then perform the Sharping filtering using Laplacian filter.