MATLAB Based Image Processing Lab Experiments

Carlos Sandoval Zuria, Juan Manuel Ramírez, David Báez-López,
and G. Espinosa Flores-Verdad[1]
Departamento de Ingeniería Electrónica
Universidad de las Américas-Puebla
Cholula, Puebla 72820
[1]*Departamento de Electrónica*
Instituto Nacional de Astrofísica, Optica y Electrónica
Tonantzintla, Puebla
MEXICO
dbaez@ieee.org

Abstract - *This paper presents a set of experiments for a digital image processing course. This set of experiments is based on MATLAB. These experiments will allow any instructor to cover experiments in most of the topics treated in a regular image processing course to be completed successfully.*

I. Introduction

Motivation for a digital image processing course [1-3] arises mainly from two areas of applications, namely, a) an increase in the pictorial information available for human interpretation, and b) image processing for automatic and autonomous machine control. This is because vision is the most important human sense in terms of the amount of information it conveys and because of a good visualization is very important for the correct information converged in an image.

Many image processing circuits require dedicated software to perform their tasks. These packages usually are highly priced and are not easily modified by the final user. In this paper the basis for the image processing is MATLAB [4], a software package now available almost anywhere for other uses and that is used as the engine for the image processing experiments. The software written for the image processing experiments is available from the users at no cost to interested users and instructors.

Image processing applications include many topics, among which we can mention remote sensing, ultrasound images, meteorology, astronomy, inspection, radar, seismology, radiology, autonomous navigation, recognition, etc.

II. Image Processing Laboratory

MATLAB is a matrix oriented computing engine. Thus, it is almost perfect for image processing because images can be thought of as matrices. A definition for an image can then be the following:

An image is a NxN array of elements. Each element in the array is a number which represents the sampled intensity. The samples are named pixels (picture element). After sampling each pixel is quantized. Each intensity is assigned a number within a finite set of values, usually between 0 and K-1, where K=2B is the possible number of gray levels, each represented by B bits. The digitizing of images is now complete and each image is now an array which can be handled more appropriately as a matrix.
Our Image processing laboratory based on MATLAB has the capability to perform the following tasks:

1. Perform algebraic operations such as addition, subtraction, multiplication, and division. These operations can be used to perform operations on images such as noise reduction by using averages, movement detection, and algebraic masking.

2. Geometric transformations such as translation, rotation, and scaling

3. Space domain operations such as histogram modification (scaling, offset, amplitude change) non linear point operations (absolute value, squaring, square root, log scale compression, edge detection)

4. Binary image processing (thresholding, logic operations).

5. Non linear image processing such as morphologic operators (opening, closing). Structuring element choice. Dilation, erosion.

6. Frequency domain processing. Fourier transform, log compression.

7. Filtering by linear convolution. Filter design (low pass, high pass, band pass, band reject.). Gaussian Filters. Linear restoration. White noise non linear filtering.

8. Digital Image coding and compression. Compression measures, losses compression, entropy, optimal coding.

III. Suggested List of Experiments

This section provides a list of experiments which cover the topics of the previous section.

1. Point-to-point transformation. This laboratory experiment provides for thresholding an image and the evaluation of its histogram. The user can choose a threshold level to see the image showing only the pixels at that threshold.

2. Morphological operations I. This experiment is intended so students can appreciate the effect of morphological operations using a small structuring element on simple binary images. The operations that can be performed are erosion, dilation, opening, closing, open-close, close-open.

3. Morphological operations II. This experiment is designed to let students know how morphological functions change images by applying consecutive erosion and dilation operations.

4. Histogram equalization. This experiment illustrates the relationship among the intensities (gray levels) of an image and its histogram. It shows how to improve the image by equalizing the histogram.

5. Geometric transformations. This experiment shows image rotation, scaling, and translation.

6. Two-dimensional Fourier transform I. The purpose of this experiment is to provide an understanding of the harmonic content of an image using the discrete Fourier transform (DFT).

7. Two-dimensional Fourier Transform II. This experiment is designed so the student learns the concept of masking with the DFT.

8. Linear filtering using convolution. After completing this experiment every student should understand the concepts of filtering using linear convolution.

9. Highly selective filters. In this experiment students appreciate the effects on an image after a highly selective filter is applied to it.

10. Ideal filters in the frequency domain. This experiment allows students to appreciate the effects of filtering low and high frequencies in an image.

11. Non Linear filtering using convolutional masks. This experiment allows students to understand the effects of a median filter on an image corrupted with impulsive noise.

12. Entropy as a compression measure. This experiments introduces students to entropy as a compression measurement to the DPCM compression measure.

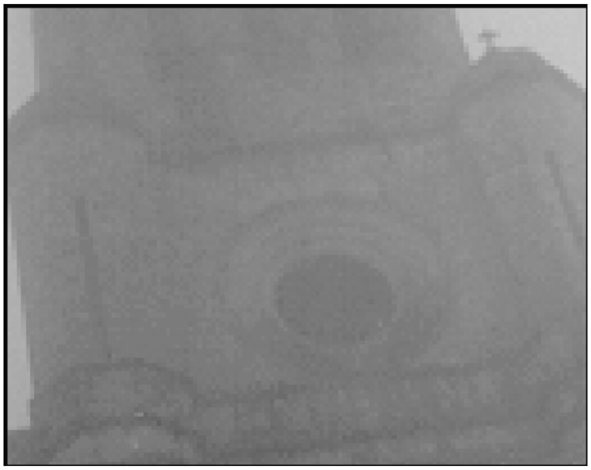
13. Edge detection. This experiment enables students to understand the concept of edge detectors and their operation in noisy images.

It is worth mentioning that in some of these experiments, the functions defined have been completely written in MATLAB code and in other cases use have been made of internal MATLAB functions such as rand, show-img, fft2, ifft2, fftshift, etc.

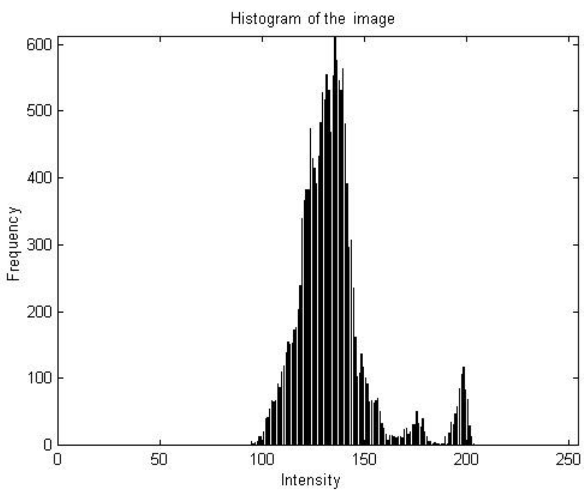
IV. Examples

Our first example is related to experiment No. 1 about image enhancement by use of point-to-point transformations. The goal of enhancement techniques is to process a given image so that the result is more suitable than the original image. An histogram provides statistical information about the use of the range of gray levels in a digital image (e.g., the histogram of a dark image is concentrated in he loer gray levels). Fig. 1 a shows a typical Mexican church tower and Fig. 1b shows its histogram. Here students can observe that most of the pixel's intensities are between 120 and 200. In other words, this image has a small dynamic range. To obtain further information about the image we can choose a value of the gray level and observe the image formed by pixels at that value. For example, choosing first as a threshold the value of 129 and then the value of 140, the resulting images in Figs. 2a and 2b are obtained. Note that clearly there are more pixels with intensity 140 and, therefore, Fig. 2b has more definition. Note that Fig. 2b shows a glimpse of a tower while Fig. 2a does not. Further processing can be done with the histogram. This is done in the next example.

A related experiment is Experiment No. 4 for histogram equalization. The purpose of histogram equalization is to optimize the use of the dynamic range. Fig. 3 shows an original image and its histogram. Note that both ends of the histogram are not used meaning that there are neither black nor white pixels in the figure. Fig. 4 shows the result on the image after histogram equalization. It can be noticed the contrast in the equalized image now that the range of gray levels is completely covered.



(a)



(b)

Fig. 1. a) Mexican church tower.
(b) Histogram for Fig. 1a.

Fig. 5 shows the church tower from Fig. 1 after applying histogram equalization. Note that the contrast improves drastically since in the original image grey levels are clustered around the center values and now they are distributed throughout all the range. This example shows how a simple concept can be used to improve images. This is in general a concept for non linear signal processing.

(a)

(b)

Fig. 2. a) Threshold at 129.
b) Threshold at 140.

Our third example (experiment No. 11) shows the effects of a median filter on an impulsive noise corrupted image. The function of a median filter is to replace the value of each pixel by the median of the gray levels in a neighborhood of that pixel. The use of median filters to remove impulsive filters is particularly effective. The size of the neighborhood to evaluate the median is defined by the user. Fig. 6 shows the noisy image and the restored image. (peppers).

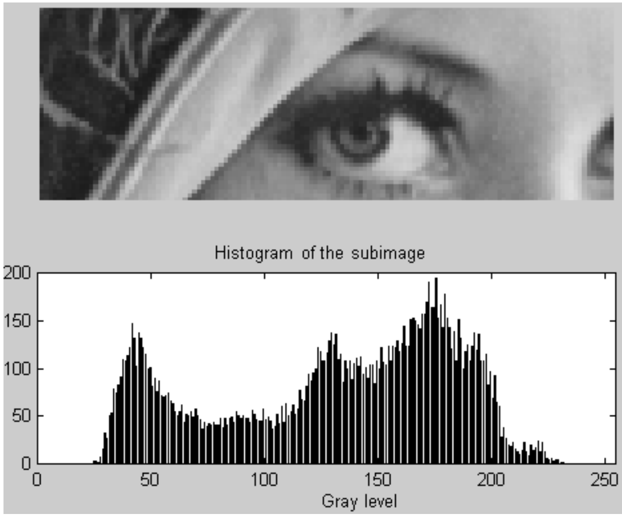


Fig. 3 Portion of Lena and its histogram.

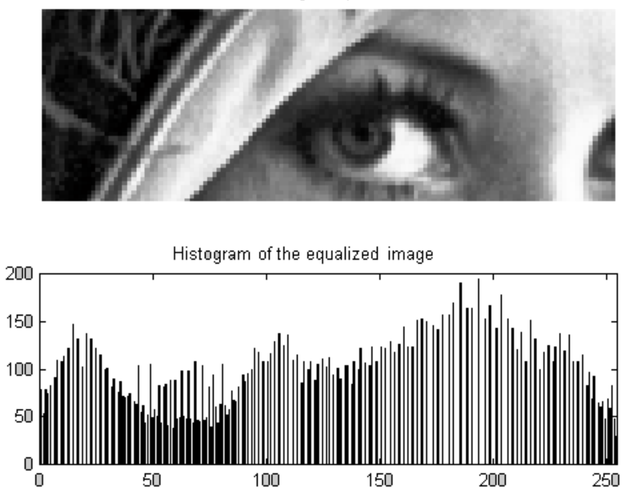


Fig. 4 This figure shows image from Fig. 3 after histogram equalization.

Conclusions

A set of image processing experiments based on MATLAB has been developed. This set of experiments covers most of the topics in a regular image processing course. An additional use of this set would be to use it in a regular Digital Signal Processing course, where image processing occupies a small part, to show digital image processing applications.

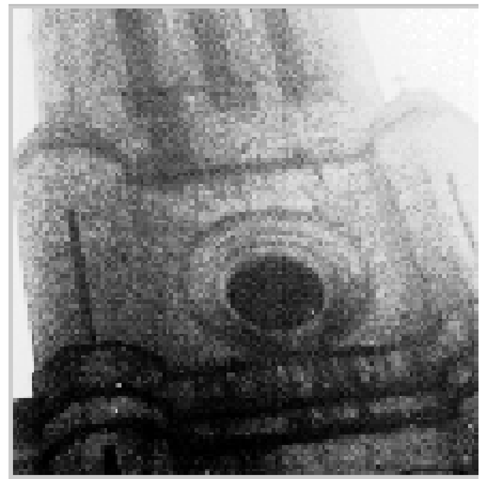


Fig. 5. Equalized church tower. The image contrast has improved after equalization.

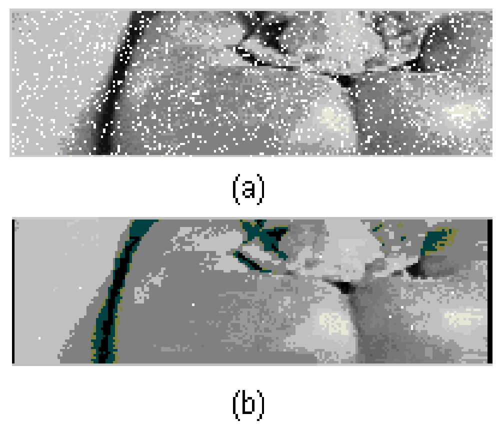


Fig. 6. Noise removal by applying non linear filtering. a) Image with impulsive noise added, b) filtered image .

References

[1] A.C. Bovik, Digital Image Processing Course Notes, Dept. of Electrical Engineering, U. of Texas at Austin, 1995.

[2] J.C. Russ, The Image Processing Handbook, CRC Press, Boca Raton, FL., 1992.

[3] R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, Mass., 1992.

[4] MATLAB, v. 4.0, The MathWorks, Inc., Natick, MA., 1995.