Digital Image Processing ECE 533

Assignment 4

Due date: March 11, in class

Department of Electrical and Computing Engineering, University of New Mexico.

Professor Majeed Hayat, hayat@ece.unm.edu

February 27, 2008

1 Image enhancement using intensity transformations

Consider the problem of mapping the normalized intensity levels \( r_k \in \{0, \frac{1}{L-1}, \ldots, 1\} \) into the normalized intensity levels \( s_k = r_k \).

1. Define a class of functions, with parameter \( \gamma, \gamma \geq 0 \), to implement the transformations shown in the Figs. 1(a) and (b). Use MATLAB to implement such intensity transformation.

2. The class of transformations

\[
s_k = \begin{cases} 
\frac{1}{1+(m/r_k)^\alpha}, & \alpha > 0 \\
0, & \alpha = \infty \text{ and } r_k < m \\
1, & \alpha = \infty \text{ and } r_k \geq m 
\end{cases}
\]  

with \( m \in [0,1] \), which is plotted in Fig. 1(c), can be used to produce contrast stretching. Implement in MATLAB the contrast stretching operation.
Test your codes for the transformations shown in the Figs. 1(a–c) using the sample images available at the following location: http://www.ece.unm.edu/~jpezoa/tmp. Make sure to plot the histogram after you apply the transformation and use it to comment on the quality of the produced image addressing issues such as the dynamic range, darkness/brightness concerns, and contrast.
Next, compare your Gamma transformation function with MATLAB’s `imadjust` command. [Useful commands in MATLAB: `imadjust`, `im2double`, `imread`.

2 Gray-level slicing

![Graphs](image)

Figure 2: Two different transformations for gray-level slicing. (a) Binary transformation or level thresholding. (b) Level boosting.

Consider transforming 8-bits intensity levels $r_k \in \{0, 1, \ldots, 255\}$ into another set $S$ with levels $s_k$ and with at most the same number of gray levels.

1. Use MATLAB to implement the transformations given in Figs. 2(a), display the transformed images, and comment on the quality of the transformed images. Use the images from Problem 1. In Fig. 2(a), the range of the highlighted interval is $[A, B] = [128, 240]$ and and the intensity values are $s_L = 10$ and $s_H = 200$. For the case of Fig. 2(b), the range is $[A, B] = [100, 130]$ while the intensity is set to $s_H = 200$.

[Useful command in MATLAB: `find`.]
3 Bit-level slicing

Explain what is meant by bit-level slicing and write a MATLAB code to generate the bit-level slice for the $k$th bit. What would be the effect on the histogram of an image if we set to zero the lower order bit planes? What would be the effect on the histogram if we set to zero the higher order bit planes? What would be the effect on the histogram if we invert the higher order bit planes? Answer these questions by looking at examples from the set of images considered in problem 1. [Useful command in MATLAB: bitand]

4 Histogram equalization and histogram specification

1. Explain why the discrete histogram equalization technique does not, in general, yield a flat histogram (This is Problem 3.5 in [1]).

2. Consider a digital image with $L$ gray-levels. Recall the histogram-equalization transformation, defined by:

$$s_k = T(r_k) = \frac{\sum_{j=0}^{k} n_j}{n}, \quad k = 0, 1, \ldots, L - 1,$$

where, $n$ is the total number of pixels in the image and $n_k$ is the number of pixels that have gray level $r_k$. Show that $T(r_k)$ satisfies the conditions required for any histogram equalization transformation, that is (i) $T(r_k)$ is single-valued and monotonically increasing in the discrete interval $[0, L - 1]$, and (ii) $0 \leq T(r_k) \leq 1$ for $r_k \in [0, L - 1]$. Show also that the inverse discrete transformation of (2) satisfies the conditions (i) and (ii) if none of the gray levels $r_k$ are missing. Is histogram equalization an example of point processing? Explain.

3. Explain the following MATLAB code that implements the histogram equalization given in equation (2).

```matlab
img1 = imread('phobos.jpg');
bins1 = 0:255;
hist1 = hist(double(img1(:)),bins1)./length(img1(:));
```
CDF1 = cumsum(hist1);
img1eq = zeros(size(img1));
for i = bins1
    img1eq(find(img1==i)) = CDF1(i+1);
end
img1eq = uint8(255*img1eq);
hist1eq = hist(img1eq(:),bins1)./length(img1eq(:));

4. Use this code to perform histogram equalization of the image phobos.jpg and any other images from Problem 1.

5. Histogram specification or matching: Write a MATLAB code to implement the algorithm proposed in pages 94–101 in the text [1]. Use the histogram specified in the website\(^1\) to perform the histogram specification of the image phobos.jpg. Use the specified histogram to transform other images as well. Comment on your results. What can you do with histogram specification that you cannot do with histogram equalization?

6. Change the specified histogram thrice to (1) brighten, (2) reduce contrast, and (3) and darken the images. Explain your ideas and comment on your results.

5 Spatial filtering

The purpose of this problem is to compare the performance of the linear and the order-statistics spatial filters under two different scenarios. Pick some sample image, and using the MATLAB command `imnoise` generate two noisy versions of the image, one with salt and pepper noise and the other with Gaussian white noise. Specify linear and order-

\(^1\)You will find the following files: HistogramV6.mat, HistogramV7.mat, and Histogram.m. The first two files are binary MATLAB files that were saved for MATLAB versions 6.x and 7.x, respectively. Use the command `load` to open them. The third file is an ASCII file containing the histogram, to load the data use `Histogram`. The data contained in the files is THE SAME, we saved in three different format to avoid you any problem reading the data. Once you read a file you will load two double variables: `z` and `pz`, which contain the gray levels and the frequencies of appearing of the gray levels, respectively.
statistics filters with different mask sizes, say for example, $3 \times 3$, $9 \times 9$, and $15 \times 15$. Apply both types of filters using the same mask size to the noisy images and compare their performance in terms of: MSE, roughness coefficient ($\rho$), $Q$-index, and general visual properties.

MATLAB codes for the metrics $\rho$ and $Q$-index are available at the website [also see Appendix]. Comment also on the running-time of the types of filters.

Prove that a median filter is nonlinear.

[Useful commands in MATLAB: `imfilter`, `imnoise`, `filter2`, `conv2`.

6 Laplacian spatial filtering

Use MATLAB to reproduce the images in Figure 3.40, 3.41, 3.43, 3.45, and 3.46 of the text and comment on your results. The original images are available at http://www.ece.unm.edu/~jpezoa/tmp.

[Useful commands in MATLAB: `imfilter`, `imnoise`, `filter2`, `conv2`.

7 Change detection using image subtraction

Identify the change in frame 2 from frame 1 using the images `scene1.jpg` and `scene2.jpg`. Highlight and superimpose the changes in frame 2 and display your results.

References


A Image quality indices.

The roughness parameter $\rho$ is computed for any image $I$ using

$$\rho(I) \triangleq \frac{\|h_1 \ast I\|_1 + \|h_2 \ast I\|_1}{\|I\|_1},$$

where $h_1(i, j) = \delta_{i-1,j} - \delta_{i,j}$ and $h_2(i, j) = \delta_{i,j-1} - \delta_{i,j}$, respectively, $\delta_{ij}$ is the Kronecker delta, $\|I\|_1$ is the $\ell^1$-norm of $I$, and $\ast$ represents discrete convolution. Note that $\rho$ is zero for a uniform image and it increases with the pixel-to-pixel variation in the image.

The image quality index $Q$ was introduced by Wang and Bovik as a metric to compare two images [3]. The index is designed to regard any image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. Mathematically, the image quality index $Q$ is defined by [3]

$$Q = \frac{4}{\bar{T}_1 \bar{T}_2} \frac{\sigma_{I_1} \sigma_{I_2} \sigma_{I_1 I_2}}{(\bar{T}_1^2 + \bar{T}_2^2)(\sigma_{I_1}^2 + \sigma_{I_2}^2)},$$

where $\bar{T}_1$ and $\sigma_{I_1}^2$ ($\bar{T}_2$ and $\sigma_{I_2}^2$) are the spatial sample mean and the spatial sample variance of one image (of the other image), respectively, and $\sigma_{I_1 I_2}$ is the covariance between the two images. The dynamical range of the index $Q$ is $[-1, 1]$ with 1 representing the best performance.

Note that the $\rho$ index uses only one image, while the $Q$-index requires a reference image.