CSL461: Digital Image Analysis

Semester I, 2017 – 2018

Programming Assignment 1: Contrast Enhancement Techniques

(***Due Date/Time: Tuesday, 12th Sep 2017, Midnight***)

- **Aim**

  To give students hands-on experience with implementing both basic and more advanced contrast enhancement techniques.

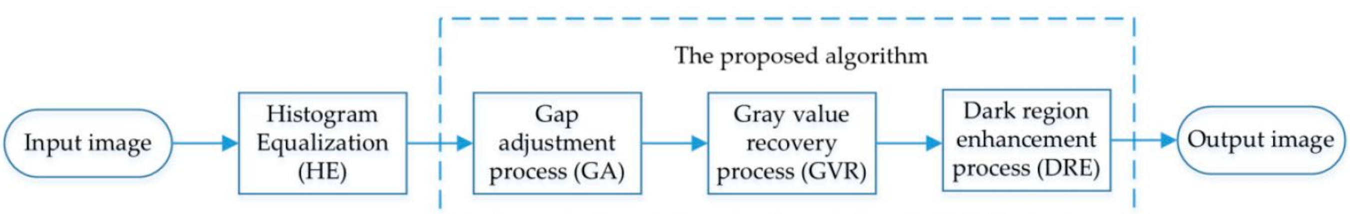
- **Introduction**

  - Histogram Equalization (HE) technique was discussed in class as one of the methods for achieving contrast enhancement. That method achieved improved contrast at the global level of the image.

  - Adaptive Histogram Equalization (AHE) is a modified version of HE that achieves local contrast enhancement by applying the same HE technique to a predefined neighborhood (block size) of individual pixels.

  - Contrast Enhancement with Gap Adjustment Histogram Equalization (CegaHE) is a further modified version that deals with the tendency of HE to over-enhance and cause loss of features.

    - Following is the gist of CegaHE extracted from the paper CegaHE.pdf. For more details and intuition behind the algorithm, read the paper, specifically, section 2!



    - Color Images: For color input images, convert from RGB space to YUV space. Then luminance (Y) component is used for further processing. For gray images, CegaHE utilizes gray values for further processing.

    - Gap Adjustment (GA): This process adjusts the gaps between two neighboring gray values in the HE histogram to alleviate the over-enhancement problem produced by HE, and generates processed images that meet the minimum requirement of human visual perception. The gap limiter equation is:

$$L(G) = Round\left(a \times [(G/127) - 1]^2\right) + b$$

where $L(G)$ denotes the limiting gap of each gray value, G is the gray value ranging from 0 to 255, and a and b are parameters that control the levels of enhancement.

    - After the image is processed by HE, suppose `hist(x)` denotes the number of pixels at gray value `x`, ranging from 0 to 255, and the gap between gray value `x` and the one previous `x` is `d`. If `d` is more than the value of the gap limiter, `L(x)`, `hist(x)` is moved backward by `d - L(x)` gray levels. After all `hist(x)` are moved backward, the GA image is obtained (example Figure 4a).

- Although you can experiment with values of `a` and `b`, empirical analysis from the paper seems to suggest that a = b = 3 provides good performance.

- Gray Value Recovery (GVR): This process alleviates the feature loss problem produced by HE. HE merges the gray values with low cumulative probabilities to the same gray value, resulting in the feature loss problem. To mitigate the problem, the GVR uses free pixels available after GA to recover the lost features.

- *Due to ambiguity in description of GVR in CegaHE.pdf, an equivalent method is adopted here from VCEA.pdf!!*

- The GVR process is as follows. It is assumed that `GAhist(x)`, `GVRhist(x)`, and `hist(x)` are the total number of pixels in the GA, GVR, and the original histograms at gray value x, respectively, where x ranges from 0 to 255. The GVR process first compares the GA histogram with the original histogram. When `GAhist(x)` is not zero, the GVR process determines the range of gray levels of the original histogram containing the sum of pixels, which is equal to `GAhist(x)`. Following this, the GVR process recovers the pixels in the particular range of gray levels from the original histogram, and repeats the same task until all the free spaces are used up. For example, when the value of `GAhist(x)` at gray value x is equal to the cumulative pixels from gray level m to n of the original histogram,

$$GAhist(x) = \sum_{X_i=m}^{n} hist(X_i)$$

- After the GVR process, the gray value x in `GAhist(x)` becomes the range $x, x+1, x+2, \dots, x+(n-m)$ in the GVR histogram. The pixels of each recovered gray value x can be expressed as:

$$GVRhist(x) = hist(m + x_i - x), \quad x_i = x, x+1, x+2, \dots, x+(n-m)$$

- Through this process, many compressed gray values are recovered.

- Dark region enhancement (DRE): Owing to the inferior light sensitivity of CCD/CMOS sensors, the image quality is poor, especially when the light is dim or insufficient. To compensate for this effect, the process enhances the textures in the dark regions of the images, and increases their contrast and clarity.

- First, the DRE segments the GVR image into the dark and the bright regions by using its mean. Pixel values less than the mean value imply that the pixels lie in the dark region. Then, the DRE computes the gradient magnitude at each pixel and sums up the gradient values of the gray values in the dark region of the image.

For example, $n_{i,j}$ represents the value of each pixel in an image and the size of the image is $W \times H$ pixels. The gradient value $G_{n_{i,j}}$ of $n_{i,j}$ is equal to the sum of $\left|GH_{n_{i,j}}\right|$ and $\left|GV_{n_{i,j}}\right|$. $\left|GH_{n_{i,j}}\right|$ and $\left|GV_{n_{i,j}}\right|$ are equal to $\left|n_{i,j+1} - n_{i,j-1}\right|$ and $\left|n_{i+1,j} - n_{i-1,j}\right|$, respectively:

$$G_{n_{i,j}} = \left|GH_{n_{i,j}}\right| + \left|GV_{n_{i,j}}\right| = \left|n_{i,j+1} - n_{i,j-1}\right| + \left|n_{i+1,j} - n_{i-1,j}\right| \tag{2}$$

where $0 \leqslant i \leqslant W - 1$ and $0 \leqslant j \leqslant H - 1$. Suppose $G(x)$ is the gradient sum of the xth gray value of the dark region. The DRE computes the gradient probabilities, $P(x)$, of the gray values of the dark region:

$$P(x) = G(x) / \sum_{r=0}^{mean-1} G(r) \qquad 0 \leqslant x \leqslant mean - 1 \tag{3}$$

where $mean$ is defined as:

$$mean = \sum_{s=0}^{255} s \times p(s) \tag{4}$$

where $p(s)$ is the ratio of the total pixels of each gray level to the total pixels of the whole image. Further, the DRE adopts the remaining free pixels, rfp, to enhance the textures in the dark region. In order to avoid over-enhancement, the DRE utilizes only a part of the remaining free pixels, R_{rfp}, based on the ratio of the number of pixels in the dark region to the total number of pixels. Suppose $D(x)$ is the total number of pixels at gray value x. The R_{rfp} is defined as:

$$R_{rfp} = \left[\left(\sum_{x=0}^{mean-1} D(x) / \sum_{x=0}^{255} D(x) \right) \right] \times rfp \tag{5}$$

Based on the gradient probabilities of each gray value in the dark region, the DRE distributes R_{rfp} to enhance these gray values. The allocated gray level, $Space(x)$, of the xth gray value is defined as:

$$Space(x) = R_{rfp} \times P(x) \qquad 0 \leqslant x \leqslant mean - 1 \tag{6}$$

Finally, the DRE expands the gaps of the gray values in the dark region based on $Space(x)$.

- **Implementation**
  - You will be implementing each of the above described contrast enhancement techniques:
    - `[f] = myHistEqual( inImg, nBins );`
    - `[f] = myAHE( inImg, nBins, wSize );`
    - `[f] = myCegaHE( inImg, nBins, ANY OTHER REQUIRED PARATMERS!! );`

  Where:

  `inImg` – Input Image

  `nBins` – Number of bins of the histogram

  `wSize` – Window Size

1. Show the performance of these algorithms on sample input images (`TestPA1.m`)
    a. For each sample image, display the results in 2 x 3 subplot with top row: **original image**, **Histogram Equalized, AHE output and bottom row: GA-CegaHE output, GVR-CegaHE output, DRE-CegaHE output**
    b. There should be a pause for user to evaluate the results
    c. When resumed, the figure should be closed and new figure should be displayed with next sample image.
    d. This process then should be repeated for all sample images
2. Comment on the trends you observe for different values of input parameters of the algorithms in README file.

- **Submitting your work:**
    - All source files and class files as one tar-gzipped archive.
        - When unzipped, it should create a directory with your ID. Example: **P2008CS1001–PA1** (NO OTHER FORMAT IS ACCEPTABLE!!! Case sensitive!!!)
        - ***Negative marks if the TA has to manually change this to run his/her scripts!!***
    - Source / class files should include the following: (Case-Sensitive file names!!)
        - `TestPA1.m, myHistEqual.m, myAHE.m and myCegaHE.m`
        - **README** (Should provide all the necessary details to the TA for ease of grading – what works, what doesn't, any assumptions made, etc.)
        - *Any other utility functions you might end up creating (example: GA.m, GVR.m, DRE.m, etc)*
    - ***Negative marks for any problems/errors in running your programs***
    - Submit/Upload it to Moodle

- **Grading Scheme : 70 Points (Total)**
    - Histogram Equalization (10 Points)
    - Adaptive Histogram Equalization (10 Points)
    - CegaHE – Gap Adjustment (10 Points)
    - CegaHE – Gray Value Recovery (10 Points)
    - CegaHE – Dark Region Enhancement (10 Points)
    - Test Script (10 Points)
    - README (5 Points)
    - Code + Comments (5 Points)