A New Easy Method of Enhancement of Low Contrast Image using Spatial Domain

S. Somorjeet Singh
Guest Lecturer, Dept. of Computer Science
Manipur University, Canchipur

Dr. H. Mamata Devi
Assoc. Prof., Dept. of Computer Science
Manipur University, Canchipur

Th. Tangkeshwar Singh
Asst. Prof.(Sr. Scale), Dept. of Computer Science
Manipur University, Canchipur

O. Imocha Singh
Assoc. Prof. & Head, Dept. of Comp. Science
Manipur University, Canchipur

ABSTRACT
For a method for enhancing an image, if it improves the quality enough or it nearly equals to the better method, the primary concern is the speed or the computational complexity and simplicity. Using global contrast stretching, low contrast image can be enhanced with an adjustable manner. Our method is based on a very simple algorithm. Its advantage is that the speed is comparatively fast and the implementation is easy.

Keywords
Global contrast enhancement, simple, easy and fast.

1. INTRODUCTION
Low contrast image is the image whose intensity levels of the pixels resides densely in a narrow range in the histogram of the image. The objects in this type of image are not clear or distinct. To improve the quality of the image and visual perception of human beings, different enhancement methods can be applied. Some methods work in frequency domain, some works in spatial domain and some works in fuzzy domain. For every method, it has some advantages and some disadvantages also. Sometimes it is required to implement a very simple, easy and fast enhancement method. It can be implemented in spatial domain expanding the narrow range of the intensity levels in the histogram and can be kept in the proper levels as far as possible by shifting the levels uniformly and adjusting the exceeding intensity level to the nearest minimum or maximum intensity level.

2. A VERY EASY METHOD FOR THE IMAGE CONTRAST ENHANCEMENT
The enhanced output image can be described as

$$F(x,y) = m * f(x,y) - n \qquad (1)$$

where ***F(x,y)=0*** if ***(m*f(x,y)–n)<0,*** or ***F(x,y)=1*** if ***(m*f(x,y)–n) > 1***, and ***F(x,y)*** is the output image, ***f(x,y)*** is the input image, ***m*** is a global multiplier and **n** is a global real value, and ***m>0, n>=0***.

When the input image ***f(x,y)*** is multiplied by the quantity ***m***, the intensity levels of the image are distributed uniformly with a factor ***m***. If ***m>1***, it makes the image higher contrast and if ***m<1***, it makes the image lower contrast. The output image is same as the input image when ***m=1*** and ***n=0***. If the input image is very low contrast, ***m*** is needed to be larger. And for the input image with not too low contrast, ***m*** should not be too large. Subtracting the value of ***n*** from the value of the product, ***m*f(x,y)***, all the intensity levels of the resulting image will be shifted towards the zero intensity level uniformly. When ***n*** equals to ***m*f(x,y)***, the minimum intensity level of the output image is zero. If ***n*** is greater than ***m*f(x,y)***, the intensity level of the output is adjusted as zero. By choosing different values of ***m*** and ***n***, we can get different output images having different intensities.

3. ALGORITHMS OF THE PROPOSED METHOD
The following two algorithms are for gray scale processing and the color image processing.

3.1 Steps for Gray scale processing
1. Read the original image.
2. If it is RGB image, convert it to Gray scale.
3. Choose an appropriate value of **m** and **n**.
4. Multiply all the pixels of the image by m and subtract n from each value.
5. If the resulting value is less than zero, adjust it as zero else if it is greater than one, adjust it as one.

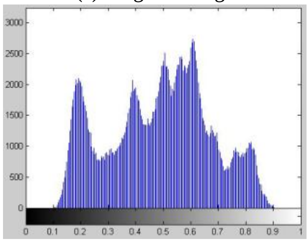
3.2 Steps for Color processing
1. Read the original image.
2. Convert it to **HSI** format.
3. Extract **H, S**, and **I** components from the **HSI** image.
4. Choose the appropriate values of **m** and **n**.
5. Multiply the pixel values of **I** by **m** and subtract **n** from each.
6. If the resulting value is less than zero, adjust it as zero else if it is greater than one, adjust it as one.
7. Combine the **H, S** and **I** components to form the **HSI** image
8. Convert it to **RGB** image.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Using Matlab, we have applied this method and found the following results.


(a) Original image


(b) Processed with m=1.28,n=0.168


(c) Histogram of original image

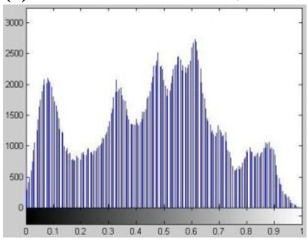

(d) Histogram of Processed image

Fig. 1. Gray scale images and histograms

The images shown in **Fig. 1(a)** and **1(b)** are the original image and the processed image (with **m=1.28**, and **n=0.168**) respectively. The **Fig. 1(c)** and **1(d)** are the histograms of the original image and the processed image respectively. The quality of the processed image is much improved from the original image to a great reasonable extent. The objects in the processed image are more cleared and more perceptible to human view.

In **Fig. 2**, some more images are shown with the processed images with their corresponding values of **m** and **n**. **Fig. 2(a)**, **2(c)** and **2(e)** are the original images and the **Fig. 2(b), 2(d)** and **2(f)** are the processed images. In these processed output images also, the images are more pronounced as well as enhanced and more acceptable to human perception.

And a color image and its processed output image are shown in **Fig. 3. Fig. 3(a)** is the original image and **Fig. 3(b)** is the processed image.

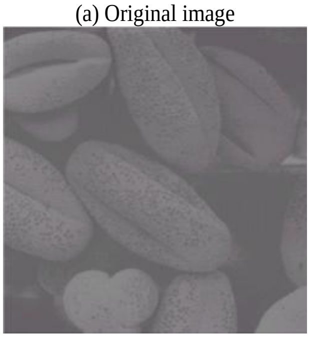
By the observation of the processed output images, we can say that this method is reasonably applicable for some images. This method is very simple and very easy to implement. It has low computational complexity and not too complicated. So, the processing speed is reasonably fast. It is even very less in complexity as compared to the Histogram equalization [1, 2, 3]. No probability density function or hard calculation is required for our proposed method whereas probability density function **(PDF)** is required for histogram equalization [2]. Here, the histogram equalization transformation for the interval [0, L-1] as shown in eqn. (2) requires complex calculations.

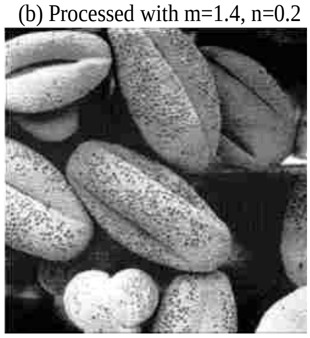
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad (2)$$

where, $\boldsymbol{k = 0, 1, 2, \ldots, L\text{-}1}$, $\boldsymbol{r_k}$ is a pixel in the input image, $\boldsymbol{s_k}$ is the corresponding pixel in the output image, $\boldsymbol{p_r(r_j)}$ is the **PDF** of the $\boldsymbol{r_j}$ and ***MN*** is the total number of pixel in the image.


(a) Original image


(b) Processed with m=1.4, n=0.2


(c) Original image


(d) Processed with m=6.76,n=2.61


(e) Original image

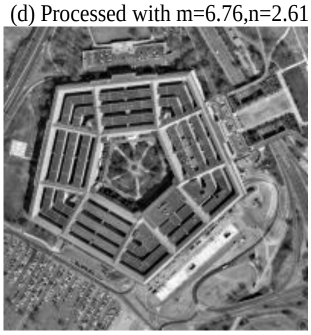
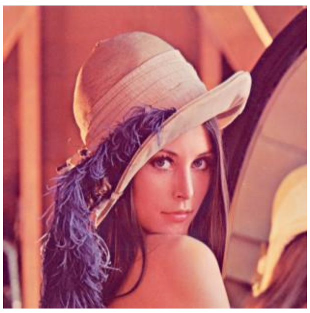

(f) Processed with m=2.04,n=0.66

Fig.2. Some more gray scale images with corresponding enhanced images


(a) Original image


(b) Processed with m=1.4, n=0.24

Fig. 3. Color image with enhanced image

Our proposed algorithm is much less in complexity as compared to gamma correction [2, 4]. In the equation for Power-Law

Table 1. A comparative statement of the different methods

	Histogram Equalized Applications	**Power-Law (Gamma Correction) Applications**	**Method Suggested by Lee**	**Method by Abdel-Ouahab *et al.*, [7]**	**Our proposed method**
Function or Equation Type	$O(n^2)$ problem, by eqn. (2)	Power function, by eqn. (3)	$O(n^2)$ problem, by eqn. (4)	Nonlinear equation by eqn. (5) and eqn. (6)	Linear function, by equation (1)
No. of Iterations required	More number of iterations required	Number of iterations is same as our method	More number of iterations required	More number of iterations required	Lesser number of iterations required
Complexity of Calculation of each pixel	More calculations done repeatedly	Calculation of power is more complex than multiplication or subtraction. When the value of γ is larger, calculation is even more complex.	More calculations done repeatedly	More calculations done repeatedly	Calculations are simple as only one multiplication and one subtraction are involved for a pixel.

(Gamma correction) transformations as shown in eqn. (3), needs to calculate the power factors of the intensity levels [2].

$$s = cr^{\gamma} \tag{3}$$

where, c and γ are positive constants, r is the pixel in input image and s is the corresponding pixel in the output image.

Again, the local contrast stretching method as suggested by Lee [5, 6] as shown in eqn. (4). The modified gray level, $\breve{g}(r,c)$ of the original gray level, $g(r,c)$ of a pixel (r,c) is given by

$$\breve{g}(r,c) = \hat{g}(r,c) + k[g(r,c) - \hat{g}(r,c)] \tag{4}$$

where, k is a global multiplier. If $\hat{g}(r,c)$ is the local mean, the calculation is so complex as compared to our method. Even if it is global mean, at least one mean is to be calculated which is not required in our method. And our method is more adjustable to get different desired output.

Abdel-Ouahab *et al.*, [7] proposed image contrast enhancement technique based on 2D Teager-Kraiser Operator. The method employed complex computations as shown in eqn. (5) and eqn. (6) for computing, $C(i, j)$, a real number lying in the unit interval, and $\bar{E}(i, j)$, mean edge gray value for measuring local contrast in a neighborhood, N_{ij} of a pixel at (i, j).

$$C(i, j) = \frac{|I(i, j) - \overline{E}(i, j)|}{|I(i, j) + \overline{E}(i, j)|} \tag{5}$$

and

$$\overline{E}(i, j) = \frac{\sum_{(k,l)\in N_{ij}} \psi(k,l) I(k,l)}{\sum_{(k,l)\in N_{ij}} \psi(k,l)} \tag{6}$$

where, $\Psi(k, l)$ represents the edge strength information for a pixel
(k, j). A comparative statement is shown in the **Table 1**.

Probably, our method is the fastest enhancement method till now. At any case, the proposed algorithm will be one of the easiest and fastest enhancement methods. Another advantage of this method is that it can be added as a tool in photo editing software like Photoshop or any existing image processing software, by attaching two sliding bars- one is for adjusting the value of **m**, and another is for adjusting the value of **n**. Besides these sliding bars, we can arrange another alternative facility for typing the values of **m** and **n** directly from the keyboard. By adjusting these two values, the user can choose his/her most desired output images.

5. CONCLUSION

Global enhancement is beneficial and it cannot be neglected because there is the limitation of local enhancement like producing artifacts. However, global enhancement is not always sufficient for various image processing tasks. The combination of global and local enhancement techniques and better improvement of the presented algorithm is the subject of our future works.

6. REFERENCES

[1] A. Suneetha, Dr. A. Sri Krishna, "A New Method of Image Enhancement in Spatial Domain Using Histogram Equalization, Smoothing and Fuzzy Technique", IJCST Vol. 2, SP 1, December 2011.

[2] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", 3rd edition, Prentice Hall, 2008.

[3] Hyunsup Yoon, Youngjoon Han, and Hernsoo Hahn, Image Contrast Enhancement based Sub-histogram Equalization technique without Over-equalization Noise", IJEEE 3:6 2009.

[4] Kh. Manglem Singh, Romen Singh, Rupachandra Singh and O. Imocha Singh, "Image Enhancement by Adapted Power Law Transformation", BUJICT, September 2010.

[5] J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics", IEEE Trans. On Pattern Analysis and Machine Intelligence, PAMI-2:165-,1980.

[6] J. S. Lee, "Refined filtering of image noise using local statistics", Computer Graphics and Image Processing, 15:380-,1981.

[7] Abdel-Ouahab BOUDRAA, El-Hadji Samba DIOP, "Image Contrast Enhancement Based on 2D Teager-Kraiser Operator", ICIP, IEEE 978-1-4244-1764, 2008.