AN IMAGE ENHANCEMENT METHOD BASED ON A S-SHARP FUNCTION AND PIXEL NEIGHBORHOOD INFORMATION

Libao Yang, Suzelawati Zenian*, Rozaimi Zakaria

Faculty of Science and Natural Resources, Universiti Malaysia Sabah,
88400 Kota Kinabalu, Sabah, Malaysia.

** Corresponding author: Suzelawati Zenian*
Email: suzela@ums.edu.my

ABSTRACT. *Image enhancement is a significant field in image processing. This paper proposes an image enhancement method based on an S-sharp function of grayscale transformation and neighborhood information. Firstly, a function is established based on the sine function. Then, the image threshold is added into the function. Finally, the result grayscales are modified by parameter* b_{ij}*, where parameter* b_{ij} *is determined by the image pixel neighborhood information. In general, in the result image, each pixel grayscale is determined by both the sine function with threshold and the parameter* b_{ij}*. In the experiment results, the* ***NIEM*** *method (we proposed) achieves better performance than the comparison algorithms. It gets the smallest MSE and the highest PSNR, SSIM. In image Lena test, MSE value:330.8151, PSNR value:22.9350, and SSIM value: 0.9451. In image Pout test, MSE value:132.0988, PSNR value:26.9218, and SSIM value: 0.9604.*

KEYWORDS. Image enhancement, S-sharp function, Standard deviation, Threshold.

INTRODUCTION

Image enhancement is an important field in image processing. The purpose of image enhancement is to improve the visual effect of the input image and turn the fuzzy image into a clear image, thus laying a solid foundation for the subsequent image analysis and image understanding. Image enhancement methods may be categorized into two broad classes: transform domain methods and spatial domain methods. The techniques in the first category are based on modifying the frequency transform of an image. However, computing a two-dimensional transform for a large array (image) is a very time-consuming task even with fast transformation techniques and is not suitable for real-time processing. The techniques in the second category directly operate on the pixels. Contrast enhancement is one of the important image enhancement techniques in the spatial domain. Gray transformation is the simplest and most effective image enhancement method. Various authors have proposed various methods based on histogram equalization. Daeyeong, et al (2017) proposed an adaptive contrast enhancement algorithm considering both preservations of the shape of a one-dimensional (1-D) histogram and statistical information on the gray-level differences between neighboring pixels obtained by a 2-D histogram. Veluchamy, M., and Subramani, B. (2020) proposed an efficient method called fuzzy dissimilarity adaptive histogram equalization with gamma correction. Pal and King (1980) proposed a method of image enhancement by computer using the fuzzy set theoretic approach. They used Zadeh's intensification operator (an S-sharp function) in membership modification. Yang Ciyin, and Huang Lianqing (2002) proposed a kind of sine nonlinear grey level transformation. According to the characteristics of the infrared images, Gong et al., (2012) proposed an image enhancement method based on sine grayscale transformation (an S-sharp function). Jose-Luis Lisani (2020) proposed the technique based on a logarithmic mapping function. Zhang and Feng (2020) proposed an image enhancement algorithm based on a quadratic function (an S-sharp function) for gray value stretching.

This paper proposes an image enhancement method based on an S-sharp function of grayscale transformation and neighborhood information. The image enhancement method consists of 3 steps: establish function, add image threshold into function, and modify the result using pixel neighborhood information. This method achieves a good performance in the experiment.

METHODOLOGY

2.1 An improved sine grayscale transformation

Gong et al. (2012) proposed an improved sine grayscale transformation. The transformation function is written as follows:

$$f(x_{i,j}) = \begin{cases} (x_{i,j}-a)[\sin(\dfrac{x_{i,j}-a}{q-a})]^{k_{i,j}} + a, & a \le x_{i,j} \le q \\ (x_{i,j}-b)[\sin(\dfrac{x_{i,j}-q}{b-q})]^{k_{i,j}} + b, & q < x_{i,j} \le b \end{cases} \quad \textbf{(1)}$$

In formula (1), a is the minimum of pixels gray, b is the maximum of pixels gray, q is image threshold, $k_{i,j} = \alpha \times \dfrac{|x_{i,j} - \overline{x_{i,j}}|}{\overline{x_{i,j}}}$, a is a constant, $\overline{x_{i,j}}$ is the average grayscale of pixels in window W_t, $\overline{x_{i,j}} = \dfrac{1}{m}\sum_{i,j \in W_t} x_{i,j}$, m is the size of window W_t.

2.2 Gray transformation function based on the quadratic function

Zhang and Feng (2020) proposed an image enhancement algorithm based on a quadratic function for gray value stretching in which the gray value of images is self-adjusted with two quadratic functions. The function defined as follows:

$$f(x_{i,j}) = \begin{cases} a, & x_{i,j} = a \\ \dfrac{(x_{i,j}-a)^2}{q-a} + a, & a \le x_{i,j} \le q \\ -\dfrac{(x_{i,j}-b)^2}{b-q} + b, & q < x_{i,j} \le b \\ b, & x_{i,j} = b \end{cases} \quad \textbf{(2)}$$

In formula (2), a is the minimum of pixels gray, b is the maximum of pixels gray, q is image threshold.

2.3 Neighborhood standard deviation

Image $I=\{x_{i,j}\mid i=1,2,3,\cdots,m,\ j=1,2,3,\cdots,n\}$, where $x_{i,j}$ is the grayscale of the pixel in row i and column j of the image. Now, $\sigma=\mathrm{SD}(a,b,c,d)$ denotes the standard deviation of a,b,c,d. σ_{ij} is the neighborhood standard deviation of the pixel inrow i and column j. According to the location of each pixel in the image (edge and non-edge), the σ_{ij} is calculated as follows:

$$\sigma_{ij}=SD\begin{pmatrix} x_{i-1,j-1} & x_{i-1,j} & x_{i-1,j+1} \\ x_{i,j-1} & x_{i,j} & x_{i,j+1} \\ x_{i+1,j-1} & x_{i+1,j} & x_{i+1,j+1} \end{pmatrix}, \quad \begin{matrix} 2<i<m-1 \\ 2<j<n-1 \end{matrix},$$

$$\sigma_{i1}=SD\begin{pmatrix} x_{i-1,1} & x_{i-1,2} \\ x_{i,1} & x_{i,2} \\ x_{i+1,1} & x_{i+1,2} \end{pmatrix}, \quad \sigma_{in}=SD\begin{pmatrix} x_{i-1,n-1} & x_{i-1,n} \\ x_{i,n-1} & x_{i,n} \\ x_{i+1,n-1} & x_{i+1,n} \end{pmatrix}, \quad 1<i<m,$$

$$\sigma_{1j}=SD\begin{pmatrix} x_{1,j-1} & x_{1,j} & x_{1,j+1} \\ x_{2,j-1} & x_{2,j} & x_{2,j+1} \end{pmatrix}, \quad \sigma_{mj}=SD\begin{pmatrix} x_{m-1,j-1} & x_{m-1,j} & x_{m-1,j+1} \\ x_{m,j-1} & x_{m,j} & x_{m,j+1} \end{pmatrix}, \quad 1<j<n,$$

$$\sigma_{11}=SD\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix}, \sigma_{1n}=SD\begin{pmatrix} x_{1,n-1} & x_{1,n} \\ x_{2,n-1} & x_{2,n} \end{pmatrix}, \sigma_{m1}=SD\begin{pmatrix} x_{m-1,1} & x_{m-1,2} \\ x_{m,1} & x_{m,2} \end{pmatrix}, \sigma_{mn}=SD\begin{pmatrix} x_{m-1,n-1} & x_{m-1,n} \\ x_{m,n-1} & x_{m,n} \end{pmatrix}$$

2.4 Define a S∗-sharp function

According to the image characteristics of sine function, the S∗-sharp function is established by magnifying and translating the sine function $f(x)=\sin(x), x\in[-\pi/2,\pi/2]$. The S∗-sharp function as follow:

$$f(x_{i,j})=255\times\frac{1}{2}\times[\sin(\frac{x_{i,j}}{255}\times\pi-\frac{\pi}{2})+1] \tag{3}$$

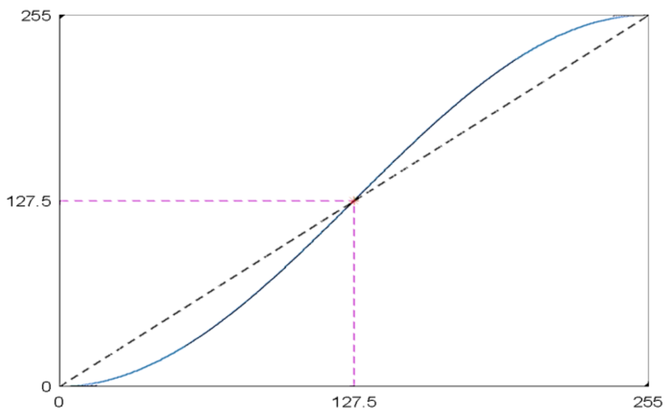

Figure 1. S∗-sharp function

2.5 Add image threshold into the S∗-sharp function(S-sharp function)

The S-sharp function in formula (3) has an inflection point at (127.5,127.5). In Fig. 1, the S-sharp function makes the grayscale smaller when they are less than 127.5, and makes the grayscale bigger when they are more than 127.5. To use the inflection point of formula (3) more flexibly, now the inflection point is corresponding to the image threshold (T). A variant of formula (3) is carried as follow:

$$f(x_{i,j}) = \begin{cases} T \times [\sin(\frac{x_{i,j}}{T} \times \frac{\pi}{2} - \frac{\pi}{2}) + 1], & 0 \le x_{i,j} \le T \\ (255-T) \times \sin(\frac{x_{i,j}-T}{255-T} \times \frac{\pi}{2}) + T, & T < x_{i,j} \le 255 \end{cases} \tag{4}$$

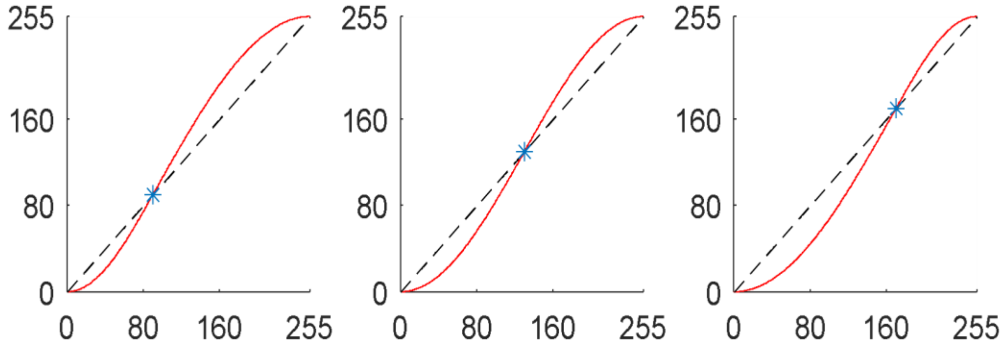

Figure 2. S-sharp function（T=90, 130, 170）

2.6 Modify the grayscale base on pixel neighborhood information

Using formula (4), the same grayscale in the input image corresponds to the same gray value in the output image. Now, for each pixel grayscale processed by formula (4), we modify the pixel grayscale base on its neighborhood information(σ_{ij}). In the input image, for the pixel $x_{i,j}$, the larger the value of σ_{ij} shows the pixel grayscales in pixel $x_{i,j}$'s neighborhood changing strongly. This study proposes the image enhancement method (we note **NIEM**) consists of two steps. The two steps are as follows:

Step 1: Compute $f(x_{i,j})$ using formula (4);

Step 2: Modify the $f(x_{i,j})$. Move $f(x_{i,j})$ vertically towards $y = x$, so that the distance from $y = x$ is b times as much as the original ($0 \le b \le 1$), and get $y_{i,j}$. Where parameter b and σ_{ij} are negatively correlated.

In experiments, parameter b is computed by a function with σ_{ij} . Fig.3 shows that the change of grayscale (0 ~ 255) at different parameter b .

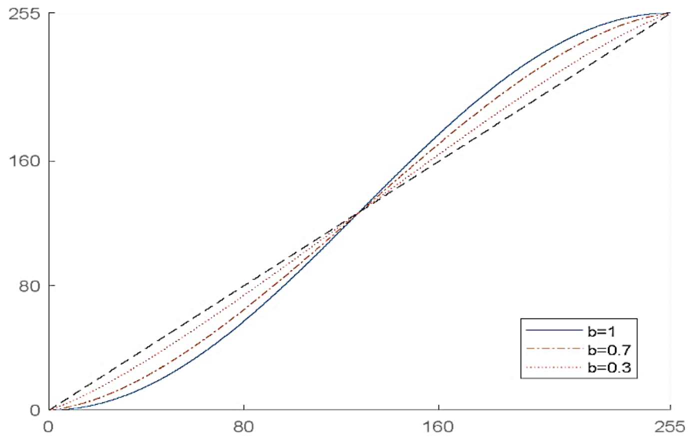

Figure 3. An example of S-sharp function modified by b = 0.3,

EXPERIMENTAL RESULTS AND DISCUSSION

This study uses Mean Squared Error (MSE), Peak Signal-Noise Ratio (PSNR), and Structural Similarity Index Measure(SSIM) to evaluate the image enhancement effect. The smaller MSE or the higher PSNR (SSIM) indicates a better enhancement effect (Thung and Raveendran, 2009).

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (x_{i,j} - y_{i,j})^2 \tag{5}$$

$$PSNR = 10 \times \log_{10} \frac{(2^n - 1)^2}{MSE} dB \tag{6}$$

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{7}$$

In formula (6), n = 8 (the test image is an 8bits image). In formula (7), μ_x is the mean of x, μ_y is the mean of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y. c_1 and c_2 are constants (Wang et al., 2004; Wang and Simoncelli, 2011).

In this section, the test image(**Lena** and **Pout)** processed by enhancement methods including formula (1)(set α = 10), (2), (3), (4), and **NIEM**. In experiments, the threshold T is computed by the Otsu method (Otsu, 1979), then the parameter b value under the gray level of each pixel is calculated by formula (8).

$$b = b_{ij} = \begin{cases} \ln[e - \frac{\sigma_{ij} - \sigma_{\min}}{\sigma^* - \sigma_{\min}} \times (e - e^{\frac{1}{2}})], & \sigma_{ij} \le \sigma^* \\ \ln[e^{\frac{1}{2}} - \frac{\sigma_{ij} - \sigma_{\max}}{\sigma^* - \sigma_{\max}} \times (e^{\frac{1}{2}} - e^{\frac{1}{3}})], & \sigma_{ij} > \sigma^* \end{cases} \tag{8}$$

Where $\sigma^* = SD$ (all pixels in image), $\sigma_{\min} = \min\{\sigma_{ij} \mid i = 1,2,3,\cdots,m, j = 1,2,3,\cdots,n\}$, $\sigma_{\max} = \max\{\sigma_{ij} \mid i = 1,2,3,\cdots,m, j = 1,2,3,\cdots,n\}$.

Table 1. PSNR, MSE, and SSIM values

	Image Lena				
	(a,b)	(a,c)	(a,d)	(a,e)	(a,f)
PSNR	20.5951	21.6286	22.6591	22.5440	**22.9350**
MSE	566.9886	446.9082	352.5062	361.9781	**330.8151**
SSIM	0.7965	0.9341	0.9301	0.9365	**0.9451**
	Image Pout				
	(a,b)	(a,c)	(a,d)	(a,e)	(a,f)
PSNR	24.4769	26.2251	24.3008	26.1890	**26.9218**
MSE	231.9457	155.0835	241.5456	156.3794	**132.0988**
SSIM	0.7339	0.9521	0.9483	0.9551	**0.9604**

Note: a is original image, b is processed by formula (1), c is processed by formula (2), d is processed by formula (3), e is processed by formula (4), f is processed by **NIEM**.

Table 1 shows that compared with other algorithms, **NIEM** gets the smallest MSE and the highest PSNR, SSIM. In image Lena test, MSE value:330.8151, PSNR value:22.9350, and SSIM value: 0.9451. In image Pout test, MSE value:132.0988, PSNR value:26.9218, and SSIM value: 0.9604. This means that NIEM gets better performance than the other four methods. The SSIM value processed by formula (3), (4), and NIEM is gradually rising, so that it is necessary and effective to add threshold and image neighborhood information. In Fig.4, Fig.4(b) and Fig.4(c) are over-enhanced and Fig.4(f) and the original image are visually more similar. In conclusion, NIEM image enhancement method, which not only can use S-sharp functions for grayscale transformation but also consider the neighborhood information (standard deviation: σ_{ij}) of each pixel, is a more flexible and efficient algorithm. In the future study, image histogram will be used as image information in the image enhancement method.

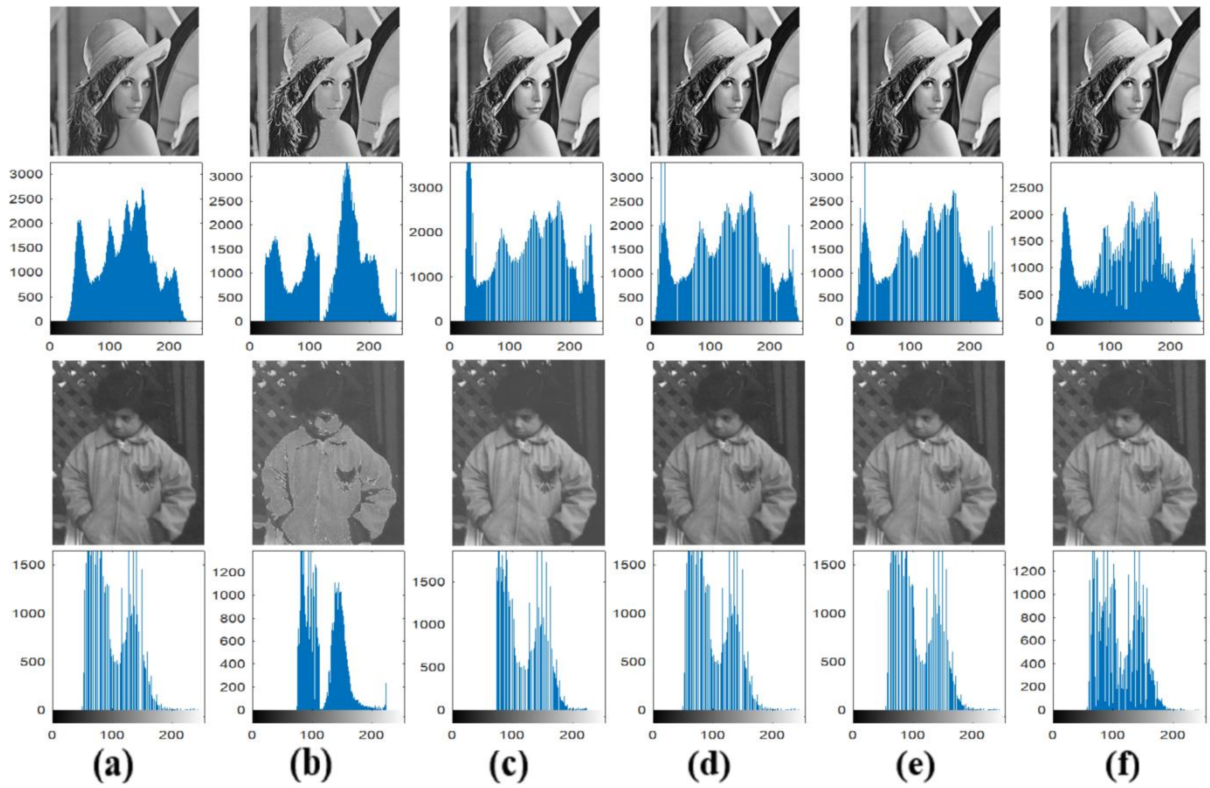

Figure. 4. (a) is the original image Pout, (b) is processed by formula (1), (c) is processed by formula (2), (d) is processed by formula (3), (e) is processed by formula (4), (f) is processed by **NIEM.**

REFERENCES

Daeyeong, Kim, Changick, and Kim. (2017) Contrast enhancement using combined 1-d and 2-d histogram-based techniques. IEEE Signal Processing Letters, 24(6), 804-808.

Magudeeswaran Veluchamy, Bharath Subramani. (2020) Fuzzy dissimilarity color histogram equalization for contrast enhancement and color correction. Applied Soft Computing, 89(1):106077.

Pal, S. K. and King, R. A. (1980) Image enhancement using fuzzy set, Electronics letters 16(10), 376-378.

Yang Ciyin, Huang Lianqing. (2002) X-ray image enhancement based on sinusoidal grayscale transformation. Optical Technology, 05,407-408.

Gong, C., Luo, C. and Yang, D. (2012) Improved image enhancement algorithm based on sine gray level transformation. Video Engineering 13, 60–63.

Lisani, J. L. . (2020) Local contrast enhancement based on adaptive logarithmic mappings. Image Processing On Line, 10, 43-61.

Zhang, Y. R. K. Y. and Feng, C. (2020). Image enhancement algorithm based on quadratic function and its implementation with fpga. Modern Electronics Technique 43(8), 72-76,81.

Thung, K.-H. and Raveendran, P. (2009) A survey of image quality measures. 2009 international conference for technical postgraduates (TECHPOS), IEEE, pp. 1-4.

Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. (2004) Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4), 600-612.

Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. (2011) The ssim index for image quality assessment, http://www.cns.nyu.edu/ lcv/ssim/ (https://ece.uwaterloo.ca/~z70wang/research/ssim/).

Otsu, N..(1979) A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66.