KKU ENGINEERING JOURNAL 2016;43(S1):125-129 Research Article

KKU Engineering Journal

https://www.tci-thaijo.org/index.php/kkuenj/index

Detecting generalized salt and pepper noise image based on standard deviation

Ratthasart Thawong, Supap Sakha and Watcharin Klongdee*

Department of Mathematics, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand.

Received April 2016
Accepted June 2016

Abstract

The goal of this research is to detect generalized salt and pepper noise in grayscale and color images, using strip window and standard deviation. We compared the performance of algorithms between proposed algorithm (PA), Rank-Ordered Absolute Differences (ROAD), and Rank-Ordered Logarithmic Differences (ROLD) in term of Peak Signal to Noise Ratio (PSNR). We found that PA is better than ROAD and ROLD.

Keywords: Noise detection, Generalized salt and pepper noise, Standard deviation

1. Introduction

Impulse noise is spotty noise caused from the change of the light intensity of the pixel differing from neighbor pixels that appear in an image. The causes of impulse noise are various such as communication errors, dilapidated pixel in digital camera sensors, locations of error memory in hardware, transmission in a noisy channel, or timing in analog-to-digital conversion. Salt and pepper noise is one of the impulse noises which have been widely studied. It happens easily in general and affects the image analysis. The corrupted pixel of this noise is changed to black pixel or white pixel. There are several algorithms for detecting and restoring this noise. For example, the algorithms using median filter which are Adaptive Median Filter (AMF)[1], Decision Based Algorithm (DBA)[2], and Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF)[3], and the algorithms using moving averaging filter which are Improved Mean Filter (IMF)[4] and Mean-Strip Window algorithm (MSW)[5]. However, the above algorithms are still unable to detect the noise effectively, since the number of corrupted pixels for random-valued impulse noise or generalized salt and pepper noise as changed randomly in the range of 0 to 255. Rank-Ordered Absolute Differences (ROAD) algorithm [6] was proposed to detect this noise by finding the distance between processes pixel and the neighbor pixels. If the sum of the distance is greater than a threshold, the processes pixel is considered corrupted pixel. A drawback of this algorithm is that when noise values close to their neighbors, then the noise cannot be detected. After that, Rank-Ordered Logarithmic Difference (ROLD) algorithm [7] was proposed to avoid the problem of ROAD by using the logarithmic function on the absolute difference of the distance. This algorithm can detect the noise better when the corrupted pixel is closer to their neighbors.

Although, there are many excellent noise detectors for detecting salt and pepper noise. This paper is interested in detecting and restoring of the generalized salt and pepper noise that also cover the detection of the salt and pepper noise. The proposed algorithm was divided into two parts. For the first part, we considered a strip window in computing the standard deviation to detect noise. For another part, we restored the noise by MDBUTMF and IMF. We considered the grayscale and color images are that corrupted by generalized salt and pepper noise and then compared the performance of the proposed algorithm and other algorithms by using Peak Signal to Noise Ratio (PSNR).

2. Materials and methods

2.1 Noise model

This section presents the methodology to construct the generalized salt and pepper noise in grayscale and RGB images. For convenient, throughout this paper, we let $\mathcal{T} = \{0,1,2,\dots,255\}$ be the set of the grayscale image intensity, such that 0 is black pixel and 255 is white pixel.

2.1.1 Generalized salt and pepper noise model

Let $S_g = [s_{i,j}]_{m\times n}$ be an original grayscale image matrix. The corrupted image matrix $X_g = [x_{i,j}]_{m\times n}$ of salt and pepper noise by α percent is defined by

$$x_{i,j} = \begin{cases} \text{uniformly random integer from the set } \mathcal{T} & \text{if } p_{i,j} < L, \\ s_{ij} & \text{elsewhere,} \end{cases}$$

where $p_{i,j}$ is a uniformly random integer from the set $\mathcal{T}$ and $L = \frac{256\alpha}{100} = 2.56\alpha$.

*Corresponding author. Tel.: +6683 356 9335
Email address: kwatch@kku.ac.th
doi: 10.14456/kkuenj.2016.38

In RGB image, let $C_{rgb} = [R, G, B]$ be an original color image matrix such that $R = [r_{ij}]_{m\times n}$, $G = [g_{ij}]_{m\times n}$, and $B = [b_{ij}]_{m\times n}$. The corrupted image matrix $X_{rgb} = [R_0, G_0, B_0]$ of salt and pepper noise by α percent, where $R_0 = [r^*_{ij}]_{m\times n}$, $G_0 = [g^*_{ij}]_{m\times n}$, and $B_0 = [b^*_{ij}]_{m\times n}$, is defined by

$$[r^*_{ij}, g^*_{ij}, b^*_{ij}] = \begin{cases} d \in \mathcal{T}^3 & \text{if } p_{i,j} < 2.56\alpha, \\ [r_{ij}, g_{ij}, b_{ij}] & \text{elsewhere,} \end{cases}$$

where d, $p_{i,j}$ are uniformly random integer from the set $\mathcal{T}$.

2.2 Proposed algorithms

We separated the proposed algorithm into two parts, detection and restoration noise. In the noise detection, we considered the strip window sized 7×1 to find the corrupted pixel by measuring the dispersion by the standard deviation. This window can move through each pixel of the corrupted image and the pixels in the window are not excessive to calculate. For restoration, we shall consider two algorithms which are MDBUTMF and IMF.

2.2.1 Proposed algorithm (PA) for the noise detection

The proposed algorithm for noise detection can be explicated as follows:

Step 1. Read the corrupted image, denote by D, of size $m \times n$.

Step 2. Consider each column of the matrix D and let the strip window sized 7×1 denoted by E, such that the processes pixel is the center position of this window. The average (μ) and standard deviation (σ) of this window without the center position are calculated by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} e_i\,, \quad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i - \mu)^2}$$

where N is the number of information in the strip window without the center position and e_i is an element belong to each E.

Step 3. Consider the center position of the strip window, if it does not belong to $[\lfloor \mu - \sigma \rfloor, \lceil \mu + \sigma \rceil]$, then it is considered a noise.

Step 4. Repeat step 2 to step 3 for the remaining window.

In color image, D = [R G B] where $R = [r_{ij}]_{m\times n}$, $G = [g_{ij}]_{m\times n}$, and $B = [b_{ij}]_{m\times n}$, we will detect the noise of each matrix in D by using the steps above. If at least two matrices have the same corrupted pixels, this pixel is considered as noise in D.

2.2.2 MDBUTMF algorithm for denoising

This algorithm for denoising can be explicated as follows:

Step 1. Consider a square window sized 3×3 that the corrupted pixel is the center of the window.

Step 2. If not all of the pixels in the square window are corrupted pixels, then replace the center of this window by median value of the uncorrupted pixels in the window. Otherwise, if all are corrupted, the center of this window is replaced by its mean value.

Step 3. Repeat step 1 to step 2 for the remaining window.

2.2.3 IMF algorithm for denoising

This algorithm for denoising can be explicated as follows:

Step 1. Consider a square window sized 3×3 that consists of the corrupted pixel is in the window.

Step 2. If not all of the pixels in the square window are corrupted pixels, then replace all the corrupted pixels of this window by mean value of the uncorrupted pixels in the window. Otherwise, if all are corrupted, the corrupted pixels of this window are replaced by its mean value.

Step 3. Repeat step 1 to step 2 for the remaining window. Now, new grayscale and color images noise are removed.

3. Results

This section, presents the simulation results of the proposed algorithm. We used grayscale and color images like ‘Lena.jpg’ and ‘Peppers.jpg’ with the sized of 512×512 pixels which was corrupted by generalized salt and pepper noise densities from 10% to 60%. Then calculated the performance of the proposed algorithm and compared with ROAD and ROLD in terms of PSNR, where the higher PSNR value indicates better performance. The PSNR is as follows:

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}}$$

where MSE of the grayscale image is defined by:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(D_{ij} - O_{ij})^2$$

For the color image MSE is defined by:

$$\mathrm{MSE} = \frac{1}{3MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(D_{ij} - O_{ij})^2$$

where $M \times N$ is the size of the original image, $D_{i,j}$ is the pixel value at (i, j) of the denoising image, and $O_{i,j}$ is the pixel value at (i, j) of the original image.

The Comparison of the PSNR values of the proposed algorithm with other algorithms in the grayscale images and the color images at different noise densities from 10% to 60% are shown in Table 1 to 4. Table 1 and Table 2 presents the performance of the proposed algorithm and other algorithms in detecting noise in ‘Peppers.jpg’ and ‘Lenna.jpg’ grayscale images. The results show that the PSNR values from using the proposed algorithm with the MDBUTMF algorithm and the proposed algorithm with the IMF algorithm are higher than other algorithms in all density of noise.

The PSNR value in the color images are represented in Table 3 and Table 4. These tables, it shows that the PSNR values of the proposed algorithm that are higher than other algorithms in all density of noise. This simulation results show that the performance of the proposed algorithm is reliable and stable to detect salt and pepper noise corrupting in the grayscale and color images. In addition, the proposed algorithm also shows better human perception than other algorithms as shown in Figure 1, 2, 3, and 4, respectively.

Table 1 The Comparison of PSNR values for various algorithms from 'Peppers.jpg' grayscale image at different percentage of noise density

Algorithms	Density of Noise in %					
	10	20	30	40	50	60
ROLD+IMF	28.035	26.894	25.027	24.286	22.823	20.335
ROAD+IMF	29.983	27.828	24.302	23.543	22.270	20.219
ROLD+MDBUTMF	33.100	30.567	27.478	25.518	23.546	21.282
ROAD+MDBUTMF	32.799	29.689	27.472	25.543	23.532	21.309
PA+IMF	31.098	29.709	28.348	26.127	23.633	20.870
PA+MDBUTMF	33.486	32.165	30.739	28.477	25.557	22.279

Table 2 The Comparison of PSNR values for various algorithms from 'Lenna.jpg' grayscale image at different percentage of noise density

Algorithms	Density of Noise in %					
	10	20	30	40	50	60
ROLD+IMF	27.851	26.633	24.773	24.119	22.947	20.604
ROAD+IMF	32.040	29.096	24.747	23.951	22.796	20.882
ROLD+MDBUTMF	33.149	30.890	27.493	25.918	24.317	22.119
ROAD+MDBUTMF	33.441	30.457	27.551	25.967	24.414	22.270
PA+IMF	32.334	30.182	28.702	27.032	24.779	21.926
PA+MDBUTMF	33.976	32.607	31.245	29.388	26.706	23.262

Table 3 The Comparison of PSNR values for various algorithms from 'Peppers.jpg' color image at different percentage of noise density

Algorithms	Density of Noise in %					
	10	20	30	40	50	60
ROLD+IMF	27.512	25.840	24.409	23.333	21.510	18.297
ROAD+IMF	27.176	25.132	23.291	22.221	20.636	17.984
ROLD+MDBUTMF	31.201	28.546	26.227	24.678	22.621	19.647
ROAD+MDBUTMF	29.512	26.561	26.327	24.763	22.549	19.607
PA+IMF	30.418	28.687	26.743	24.008	21.003	17.934
PA+MDBUTMF	32.546	30.695	28.996	26.229	22.753	19.051

Table 4 The Comparison of PSNR values for various algorithms from 'Peppers.jpg' color image at different percentage of noise density

Algorithms	Density of Noise in %					
	10	20	30	40	50	60
ROLD+IMF	28.085	26.360	24.423	23.635	22.153	19.527
ROAD+IMF	29.890	26.619	24.615	23.823	22.634	21.166
ROLD+MDBUTMF	31.906	29.358	26.792	25.265	23.549	20.341
ROAD+MDBUTMF	30.324	27.149	26.913	25.389	23.722	21.424
PA+IMF	30.931	29.038	27.747	25.591	22.951	20.108
PA+MDBUTMF	33.452	31.212	30.168	27.993	24.843	21.319

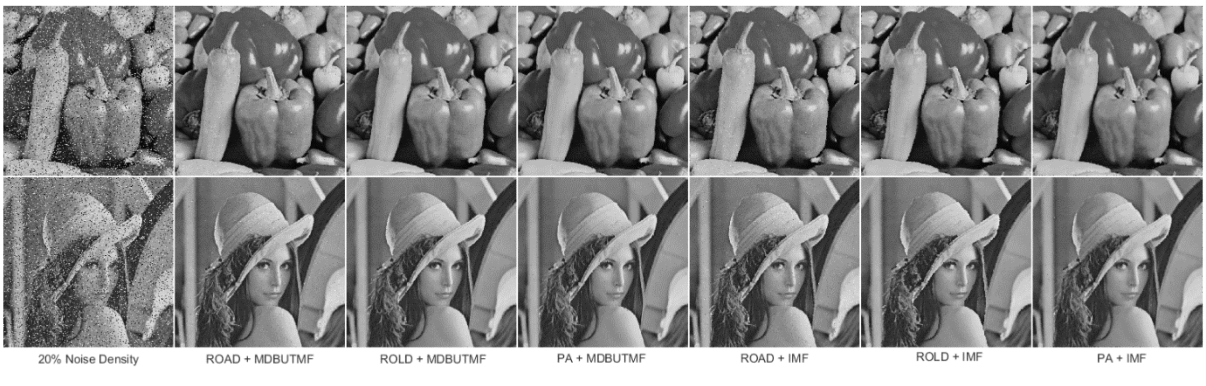

Figure 1 The Results of the proposed algorithm and various algorithms for 'Lenna.jpg' and 'Peppers.jpg' grayscale images corrupted with 20% noise density

Figure 2 The Results of the proposed algorithm and various algorithms for 'Lenna.jpg' and 'Peppers.jpg' grayscale images corrupted with 50% noise density

Figure 3 The Results of the proposed algorithm and various algorithms for 'Lenna.jpg' and 'Peppers.jpg' color images corrupted with 20% noise density

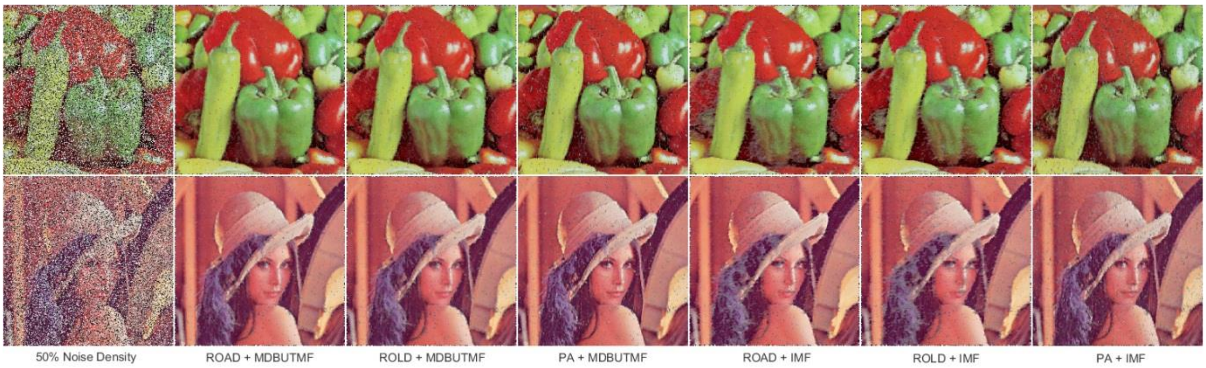

Figure 4 The Results of the proposed algorithm and various algorithms for 'Lenna.jpg' and 'Peppers.jpg' color images corrupted with 50% noise density

4. Conclusions

We proposed the new algorithm for detecting the generalized salt and pepper noise in the grayscale and color images by measuring the dispersion using the standard deviation. The simulation results show that the performance of the proposed algorithm is better than the compared algorithms. The proposed algorithm can accurately detect the position of the noise, so denoising is effective and can maintain the uncorrupted pixel of the image. Therefore, the proposed algorithm obtains the highest PSNR value for both the grayscale and color images. In statistical, there are many ways for data dispersion measuring. Standard deviation is one of those ways we interested and studied. This way is a good result in case of high density of noise, but it is not desirable. In further studies, we will consider another method in data dispersion measurement to improve our algorithm.

5. Acknowledgement

This research was supported by the research capability enhancement program through graduate student scholarship, Faculty of Science, Khon Kaen University.

6. References

[1] Hwang H, Haddad RA. Adaptive Median Filters: New Algorithms and Results. IEEE Trans Image Process 1995;4:499-502.

[2] Srinivasan KS, Ebenezer D. A New Fast and Efficient Decision Based Algorithm for Removal of High Density Impulse Noise. IEEE Signal Process Lett 2007;14:189-192.
[3] Esakkirajan S, Veerakumar T, Subramanyam AN, and PremChand CH. Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetric Trimmed Median Filter. IEEE Signal Process Lett 2011;18:287-290.
[4] Malviya S, Amhia H. Image Enhancement Using Improved Mean Filter at Low and High Noise Density. International Journal of Emerging Engineering Research and Technology 2014;2:45-52.
[5] Sakha S, Ieosanurak W, Pongkitwitoon M, Klongdee W. Noise Reduction Based on Standard Deviation of Strip Window. Far East Journal of Electronics and Communications 2016;16:89-100.
[6] Garnett R, Huegerich T, Chui C, He W. A Universal Noise Removal Algorithm with an Impulse Detector. IEEE Trans Image Process 2005;14:1747-1754.
[7] Dong Y, Chan RH, and Xu S. A Detection Statistic for Random-Valued Impulse Noise. IEEE Trans Image Process 2007;16:1112-1120.