VOL 2 (2018) NO 4

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Salt and Pepper Noise: Effects and Removal

Jamil Al-Azzeh[#], Bilal Zahran[#], Ziad Alqadi[#]

[#] *Al- Balqa Applied University (BAU), Jordan*
E-mail: azzehjamil@gmail.com

***Abstract*— Noises degrade image quality which causes information losing and unsatisfying visual effects. Salt and Pepper noise is one of the most popular noises that affect image quality. In RGB color image Salt and pepper noise changes the number of occurrences of colors combination depending on the noise ratio. Many methods have been proposed to eliminate Salt and Pepper noise from color image with minimum loss of information. In this paper we will investigate the effects of adding salt and pepper noise to RGB color image, the experimental noise ratio will be calculated and the color combination with maximum and minimum numbers of occurrence will be calculated and detected in RGB color image. In addition this paper proposed a methodology of salt and pepper noise elimination for color images using median filter providing the reconstruction of an image in order to accept result with minimum loss of information. The proposed methodology is to be implemented, tested and experimental results will be analyzed using the calculated values of RMSE and PSNR.**

***Keywords*— RGB color image, salt and pepper noise, color combination, median filter, RMSE, PSNR.**

I. INTRODUCTION

Digital color image can be presented using deferent models. The most common model that used is RGB. In RGB color model colors described as a triple (Red, Green, and Blue). The RGB color space can be considered as a three-dimensional unit cube, in which each axis represents one of the primary colors [1, 2, 3]. Figure 1 shows the RGB color model, where any color can be represented in three values. For example (0, 0, 255) stands for Blue color.

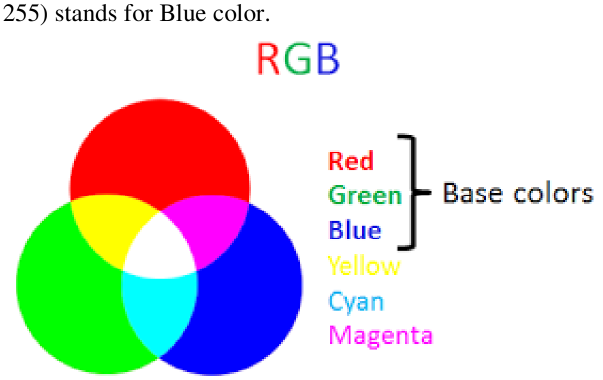

Fig. 1. RGB color model

Noise encountered into RGB color image caused by images captured sensing devices and transmitted communication noises reduce the visual quality of an image. One of the common noises is Salt and Pepper noise.

1.1 Salt-and-pepper noise

Salt-and-pepper noise is a sparsely occurring white and black pixels sometimes seen on images. Median filter or a morphological filter methods considered as a common reduction methods of this type noise of noise [4, 5].

Image noise may be defined as any change in the image signal, caused by external disturbance. Digital images are often corrupted by impulse noise also known as Salt and Pepper noise due to transmission errors [10]. Accordingly it is important to detect noisy pixels (using estimated calculations) and recover an efficient value for each, which known as image filters [6, 9]. The most commonly used filters are the Standard Median Filter (SMF), Adaptive Median Filter (AMF) [5], Decision Based Algorithm (DBA) [8], Progressive Switching Median Filter (PSMF) [12], and Detail preserving filter (DPF) [13]. The filtering algorithm varies from one algorithm to another by the approximation accuracy for the noisy pixel from its surrounding pixels [6, 9]. Median Filter (MF) is used widely because of its effective noise suppression capability [7]. Despite its main disadvantages of modifying both noisy and non-noisy pixels thus removing some fine details of the image. Image de-noising is the process of finding and recover unusual values in digital image, that represents unwanted information which spoils image quality.

The Salt & Pepper noise is generally caused by defect of camera sensor, software failure, or hardware failure in image capturing or transmission. Due to this situation, Salt & Pepper noise model, only a proportion of all the image pixels are corrupted whereas other pixels are non-noisy [12]. A standard

Salt & Pepper noise value may be either minimum (0) or maximum (255). The typical intensity value for pepper noise is close to 0 and for salt noise is close to 255. Furthermore, the unaffected pixels remain unchanged.

$$\eta(x, y) = \begin{Bmatrix} 0, & Pepper\ noise \\ 255, & Salt\ noise \end{Bmatrix} \quad (1)$$

1.2 Median filtering

Is a nonlinear method widely used to remove 'salt and pepper' type noise. The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighboring pixels within predefined window size. The median is calculated by first sorting all the pixel values from the window, and then replacing the pixel being considered with the middle (median) pixel value. Then the window slides, pixel by pixel over the entire image.

The following (Figure 2) example shows the application of a median filter to a simple 2 dimensional signal. A window size of 3x3 is used, with one entry immediately preceding and following each entry.

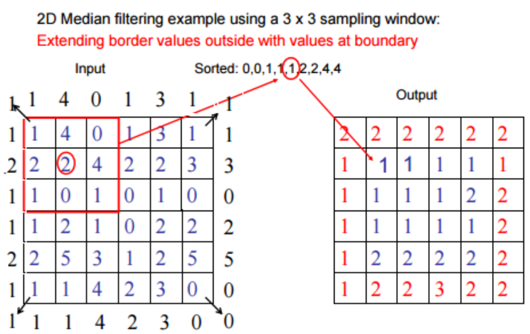

Fig 2. Median filter example

1.3 PSNR and RMSE

PSNR uses a standard mathematical model to measure a quality of reducing noise from the image. It is commonly used in the development and analysis of compression algorithms, and for comparing visual quality between different compression systems. PSNR is calculated by the following formula:

$$PSNR = 20*\log10\ (255 / RMSE) \quad (2)$$

Where RMSE is the square root of the mean squared error for the entire image.

PSNR is calculated using the first-captured image as the reference image, and the filtered image. The higher the PSNR value is the higher quality of noise reduction

II. Salt and Pepper Noise Effects

First part of this paper we will introduce an experimental method to investigate the affects a Salt and Pepper noise affecting RGB color image. The proposed method can implemented applying the following steps.

1. Acquire the source RGB image.
2. Add Salt and Pepper noise to the image varying the noise ratio.
3. Construct a RGB columns matrix by reshaping the color image.
4. Find the unique_colors combination by applying the Matlab function unique.
5. Construct an array with accumulation to find the repetition of each color combination by applying the Matlab function accumarray.
6. Find the color combination with maximum repetition.
7. Find the color combination with minimum repetition.
8. Detect the locations of the pixels with maximum repletion.
9. Detect the locations of the pixels with minimum repletion.

The program was implemented in deferent ways as follows:

1. First we create a black RGB color image with size (200x250x3).

Here we select deferent values of noise ratio associated with imnoise Matlab function; Table 1 shows the implementation results of this phase. Here the experimental noise ratio will be calculated as follows:
Experimental noise ratio= Color combination with max occurrence/number of colors in the image.
Number of colors in the image= 50000.

TABLE I
RESULTS OF PHASE 1

Noise ratio	Number of color combinations	Color combination with max occurrence	Combination value	Experimental noise ratio
0	1	50000	0 , 0, 0	0
0.001	4	49908	0 , 0, 0	0.0018
0.002	4	49849	0 , 0, 0	0.0030
0.003	5	49742	0 , 0, 0	0.0052
0.004	5	49722	0 , 0, 0	0.0056
0.005	6	49652	0 , 0, 0	0.0070
0.006	4	49564	0 , 0, 0	0.0087
0.007	5	49458	0 , 0, 0	0.0108
0.008	7	49359	0 , 0, 0	0.0128
0.009	6	49304	0 , 0, 0	0.0139
0.010	6	49256	0 , 0, 0	0.0149
0.020	8	48469	0 , 0, 0	0.0306
0.030	8	47793	0 , 0, 0	0.0441
0.040	7	46992	0 , 0, 0	0.0602
0.050	7	46321	0 , 0, 0	0.0736
0.060	8	45693	0 , 0, 0	0.0861
0.070	8	44818	0 , 0, 0	0.1036
0.080	8	44235	0 , 0, 0	0.1153
0.090	8	43543	0 , 0, 0	0.1291
0.100	8	42912	0 , 0, 0	0.1418

2. Second we create a white RGB color image with size (200x250x3).
Here we select deferent values of noise ratio associated with imnoise Matlab function; table 2 shows the implementation results of this phase.

TABLE III
RESULTS OF PHASE 2

Noise ratio	Number of color combinations	Color combination with max occurrence	Combination value	Experimental noise ratio
0	1	50000	255, 255, 255	0
0.001	4	49935	255, 255, 255	0.0013
0.002	4	49868	255, 255, 255	0.0026
0.003	5	49764	255, 255, 255	0.0047
0.004	4	49707	255, 255, 255	0.0059
0.005	4	49649	255, 255, 255	0.0070
0.006	4	49536	255, 255, 255	0.0093
0.007	6	49479	255, 255, 255	0.0104
0.008	4	49412	255, 255, 255	0.0118
0.009	6	49293	255, 255, 255	0.0141
0.010	6	49232	255, 255, 255	0.0154
0.020	7	48569	255, 255, 255	0.0286
0.030	7	47860	255, 255, 255	0.0428
0.040	7	47092	255, 255, 255	0.0582
0.050	8	46264	255, 255, 255	0.0747
0.060	8	45677	255, 255, 255	0.0865
0.070	8	44855	255, 255, 255	0.1029
0.080	8	44201	255, 255, 255	0.1160
0.090	8	43411	255, 255, 255	0.1318
0.100	8	42974	255, 255, 255	0.1405

From the obtained results shown in Table 1 and Table 2 we can see that the experimental calculated noise ratio is closed to a theoretical one which is approved in Matlab, this is shown in Figure 3.

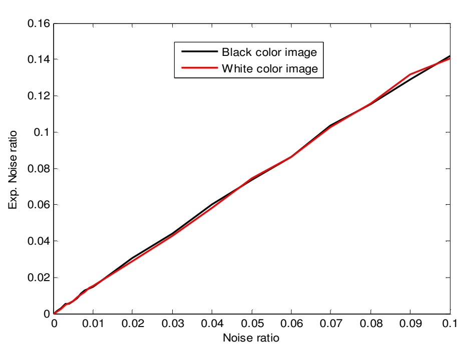

Fig. 3. Relationship between theoretical and experimental noise ratios

3. Third phase of implementation by taking a real RGB color image peppers.png
Here we select deferent values of noise ratio associated with imnoise Matlab function; Table 3 shows the implementation results of this phase.
From the obtained results shown in Table 3 we can see that adjusting the noise ratio value leads to changing the number of color combinations, the value of the color combination with minimum repetition, and the value of color combination with maximum repetition. Figure 4 shows the relationship between noise ratio and the number of color combinations:

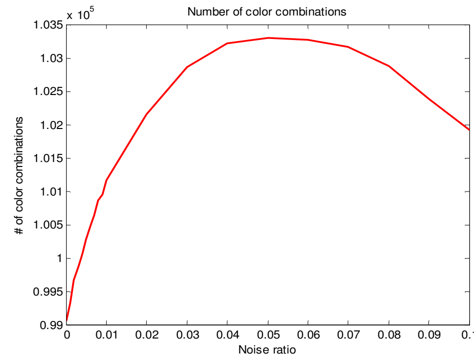

Fig 4. Relationship between number of color combinations and noise ratio

TABLE IIIII
RESULTS OF PHASE 3

Noise ratio	Number of color combinations	Color combination with min occurrence	Number of occurrences	Color combination with max occurrence	Number of occurrences
0	99059	5 , 10, 15	1	254, 254, 254	344
0.001	99320	0, 8, 30	1	254 , 254, 254	342
0.002	99670	0 , 3, 11	1	254 , 254, 254	343
0.003	99873	0 , 0 , 60	1	254 , 254, 254	343
0.004	100065	0, 5, 14	1	254 , 254, 254	343
0.005	100267	0 , 0, 26	1	254 , 254, 254	333
0.006	100493	0 , 0 , 35	1	254 , 254, 254	339
0.007	100634	0, 0, 46	1	254 , 254, 254	339
0.008	100856	0, 0, 38	1	254 , 254, 254	336
0.009	100957	0, 0, 23	1	254 , 254, 254	331
0.010	101164	0, 0, 3	1	254 , 254, 254	335
0.020	102148	0, 0, 6	1	254 , 254, 254	323
0.030	102864	0, 0, 5	1	254 , 254, 254	323
0.040	103214	0, 0, 12	1	254 , 254, 254	302
0.050	103301	0, 0, 8	1	254 , 254, 254	300
0.060	103273	0, 0, 4	1	254 , 254, 254	287

0.070	103161	0, 0, 2	1	255, 255, 255	278
0.080	102870	0, 0, 3	1	255, 255, 255	286
0.090	102385	0, 0, 5	1	255, 255, 0	274
0.100	101915	0, 0, 5	1	255, 255, 255	282

Next figure shows the locations of the color combinations with minimum and maximum repletion.

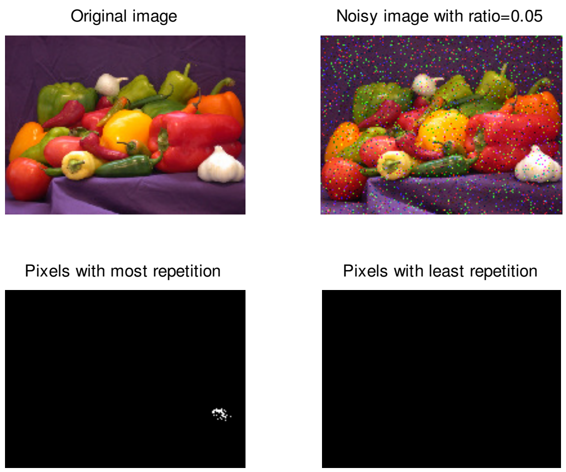

Fig. 5. Locations of the color combinations with minimum and maximum repletion

III. Salt and Pepper Noise Effects

A Matlab code was written to implement the proposed method, deferent RGB color images with deferent sizes were affected by salt and pepper noise with deferent value of noise ratio, Figure 6 shows the results of de-noising the image peppers.png with noise ratio equal 0.1. Our method used to reduce/eliminate salt and pepper noise from RGB color image can implemented applying the following steps:

1. Acquire RGB color image
2. Get the dimensions of the image
3. Extract the individual red, green, and blue color channels.
4. Apply Median Filter the channels.
5. Find the noise in the red.
6. Get rid of the noise in the red by replacing with median.
7. Find the noise in the green.
8. Get rid of the noise in the green by replacing with median.
9. Find the noise in the blue.
10. Get rid of the noise in the blue by replacing with median.
11. Reconstruct the noise free RGB image.

The RGB color image peppers.png then was treated several times varying the noise ratio from 0.001 to 0.1. The source image each time was affected with salt and pepper noise, and then filtered by median filter with 3x3 mask. Each time we calculate RMSE and PSNR, Table 4 shows the experimental results of this part of implementation.

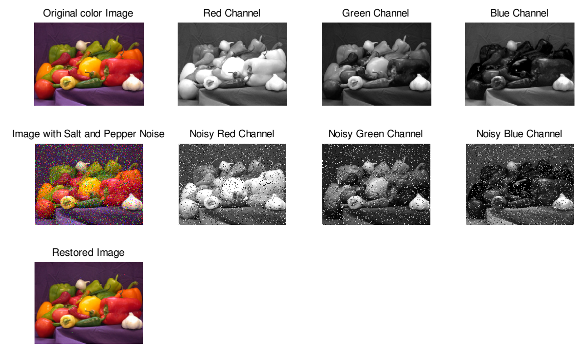

Fig. 6. De-noising peppers.png

From the obtained results shown in Table 4 we can raise the following facts:

1. Median filter provides a high quality of salt and pepper noise reduction and elimination, because of the high values of PSNR and low values of RMSE.
2. Increasing noise ration leads to decreasing PSNR and increasing RMSE, thus reducing the quality of the median filter.
3. The relationship between PSNR and noise ratio is closed to linear as shown in Figure 7.
4. The relationship between RMSE and noise ratio is also closed to linear relationship as shown in Figure 8.
5. The quality of median filter in noise reduction decreased when the noise ratio increased.

TABLE IVV
EXPERIMENTAL RESULTS OF FIRST PART OF IMPLEMENTATION

Noise ratio	PSNR	RMSE
0.001	140.2946	0.0033
0.005	130.5774	0.008
0.01	125.3474	0.0146
0.015	120.8481	0.021
0.02	118.1216	0.027
0.025	115.4155	0.0345
0.03	113.897	0.0408
0.035	111.7628	0.0478
0.04	110.0766	0.0534
0.045	109.36	0.0592
0.05	108.2835	0.0678
0.055	107.0779	0.0749
0.06	104.6116	0.0848
0.065	103.3155	0.0929
0.07	103.2653	0.0988
0.075	101.1348	0.106
0.08	99.4528	0.115
0.085	99.1635	0.117
0.09	97.8145	0.1335

0.095	97.4672	0.1391
0.1	95.6972	0.1481

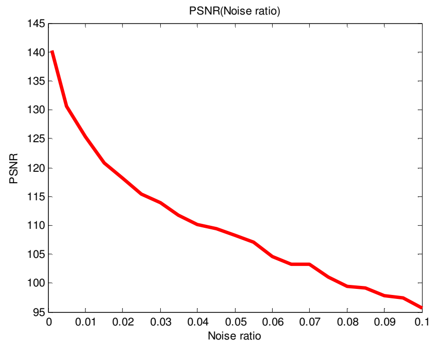

Fig 7. Relationship between PSNR and noise ratio

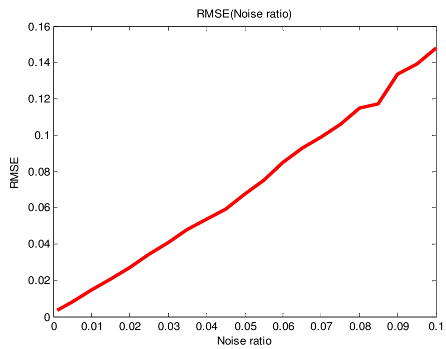

Fig 8. Relationship between RMSE and noise ratio

The second part of implementation was implemented taking deferent color images with deferent size fixing the noise ratio to 0.1. Table 5 shows the experimental results of this part:

TABLE V
EXPERIMENTAL RESULTS OF PHASE 2 OF IMPLEMENTATION

Image	Rows	Columns	Colors	PSNR	RMSE
Peppers.png	384	512	3	95.6972	0.1481
fabric.png	480	640	3	83.1603	0.4595
football.jpg	256	320	3	86.1364	0.3510
Birds.jpg	371	508	3	69.6713	0.5551
blood.jpg	156	322	3	90.1108	0.1602

From the obtained results shown in table 5 we can see that there is no fixed relationship between PSNR and image size, RMSE and image size, and this quite correct due to the nature of salt and pepper noise and due to the steps 6m8m and 10

IV. CONCLUSIONS

The proposed method of investigation the effects of Salt and Pepper noise on RGB color image was implemented several times using deferent images and deferent values of noise ratio, and from the obtained results we can conclude the following facts:

- ✓ The experimental calculated noise ratio is more accurate comparing with the noise ratio proposed in matlab.
- ✓ The experimental calculated noise ratio is very closed to the noise ratio proposed in Matlab.
- ✓ Varying the values of noise ratio leads to some changes in the number of color combinations.
- ✓ Varying the values of noise ratio leads to e change in the color combination of minimum repletion..
- ✓ Varying the values of noise ratio leads to e change in the color combination of maximum repletion..
- ✓ Using this method we can detect the locations of color combination with minimum repetition.
- ✓ Using this method we can detect the locations of color combination with maximum repetition.

After using median filter to reduce salt and pepper noise was proposed, implemented and tested. It was shown that:

- Median filter provides a high quality of salt and pepper noise reduction and elimination, because of the high values of PSNR and low values of RMSE.
- Increasing noise ration leads to decreasing PSNR and increasing RMSE, thus reducing the quality of the median filter.
- The relationship between PSNR and noise ratio is closed to linear relationship.
- The relationship between RMSE and noise ratio is also closed to linear relationship.
- The quality of median filter in noise reduction decreased when the noise ratio increased.

There is no fixed relationship between PSNR and image size, RMSE and image size, and this quite correct due to the nature of salt and pepper noise.

REFERENCES

[1] Majed O. Al-Dwairi, Ziad A. Alqadi, Amjad A. AbuJazar and Rushdi Abu Zneit, Optimized True-Color Image Processing, World Applied Sciences Journal 8 (10): 1175-1182, 2010 ISSN 1818-4952 © IDOSI Publications, 2010.

[2] Zahran B., Jamil Al-Azzeh , Alqadi Z., Alzoghoul M. Khawatreh M, A Modified LBP Method To Extract Features From Color Images, Journal of Theoretical and Applied Information Technology; May 2018., vol. 96 Issue 10, p3014.

[3] Akram A. Moustafa and Ziad A. Alqadi, Color Image Reconstruction Using A New R'G'I Model, Journal of Computer Science 5 (4): 250-254, 2009 ISSN 1549-3636 © 2009 Science Publications.

[4] Firas Ajil Jassim, Kriging Interpolation Filter to Reduce High Density Salt and Pepper Noise, World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 3, No. 1, 8-14, 2013.

[5] D. Puig and M. Angel García,"Determiningoptimal window size for texture feature extraction methods", IX Spanish Symposium on Pattern Recognition and Image Analysis, Castellon, Spain, May, vol. 2, pp. 237-242, 2001.

[6] E. H. Issaks, and R. M. Srivastava, An Introduction to Applied Geostatistics. Oxford: Oxford University Press, 1989.

[7] F. A. Jassim, "Image Denoising Using Interquartile Range Filter with Local Averaging", International Journal of Soft Computing and Engineering (IJSCE), vol. 2, Issue 6, pp: 424 – 428, January 2013.

[8] K. Vasanth, S. Karthik, and S. Divakaran, "RemovalofSalt&Pepper Noise using Unsymmetrical Trimmed Variants as Detector", European Journal of Scientific Research, vol. 70, no.3, pp. 468-478, 2012.

[9] M. R. Stytz and R. W. Parrott. Using kriging for 3d medical imaging. Computerized Medical Image., Graphics, vol. 17, no. 6, pp.421–442, 1993.

[10] N.Alajlan,M.Kamel,E.Jernigan,"DetailPreservingimpulsenoise removal", International journal on Signal processing: image communication, vol 19, pp. 993-1003, 2004.

[11] T. Vimala,"SaltAndPepperNoiseReductionUsing MdbutmFilter WithFuzzyBasedRefinement",Volume 2, Issue 5, May 2012.

[12] W. C. M. Van Beers and J. P. C. Kleijnen "Kriging for interpolation in random simulation",Journal of the Operational Research Society , vol. 54, pp. 255–262, 2003.

[13] W. K. Pratt, Digital Image Processing, Fourth Edition, John Wiley & Sons, Inc., Publication, 2007.