ADAPTIVE IMAGE DENOISING SOLUTION

Mihai NICOLAE[1]
Giorgiana Violeta VLĂSCEANU[2]
Costin-Anton BOIANGIU[3]

Abstract: *Digital images can be affected by many phenomena which corrupt the image, reducing its quality and which fall under the name of noise. Image noise is usually named after the distribution of the noise signal. As such, the noise signals are encountered and can be modeled by a Poisson, Gaussian, or even normal distribution or salt and pepper noise, which represents very high and very low impulse signals. This article aims to describe a simple voting image denoise algorithm that combines several filters that are specialized in certain types of digital image noise. The obtained results are compared with a well-established denoising technique, comparing the resulting image quality, and compute time.*

Keywords: *image denoising, Gaussian noise, salt and pepper noise, voting system, median filtering, bilateral filtering*

1. Introduction

For a digital camera sensor to capture light, it is equipped with charge wells, that when hit with photons they store and accumulate photoelectrons. As the electrons accumulate, the light intensity of the pixel corresponding to the charge well increases, but due to thermal effects, thermal electrons can also appear and get stored in the charge well. These electrons are characterized by a Gaussian distribution and are known as Gaussian noise [1]. The image can also be affected by salt & pepper noise, which can appear during the ADC conversion, the transmission of the image, or by shot noise caused in low light conditions by the discrete nature of photons, which is represented by a Poisson distribution.

Image denoise is the process of reducing or eliminating the noise from an image, thus bringing it closer to its original form. Image denoise can also be helpful when trying to add missing data to a picture, such as when we are dealing with shot noise or rendering optimization techniques.

[1] Engineer, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, mihai.nicolae1108@stud.acs.upb.ro
[2] Teaching Assistant, PhD Student, Eng., Computer Science and Engineering Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, giorgiana. vlasceanu@cs.pub.ro
[3] Professor, PhD, Eng., Computer Science and Engineering Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, costin.boiangiu@cs.pub.ro

The goal of this paper is to find a simple and intuitive image denoising algorithm that combines several known algorithms through a voting system. Such a system would decide how much and in what way each algorithm should contribute to the final image. In order to obtain this system, this research targets specific types of noise and tries to identify and remove them individually. This work will assume that the images used are grayscale, namely that they have an only one-color channel, although these concepts can be extended to color images.

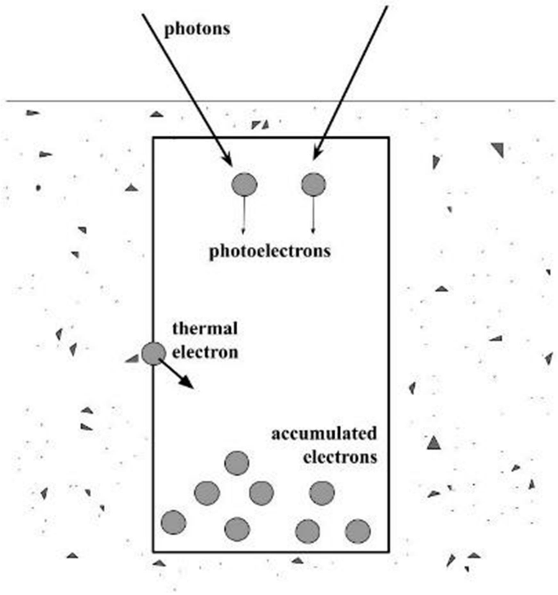

Figure 1 Charge Well Representation, Adapted from [1]

1.1. Previous work

Image noise is a severe problem affecting several areas of work and research, from computer tomography to astronomy. The noise models encountered in these various areas differ, be it Gaussian, speckle in ultrasound images, Rician noise in MRI images [4] or salt and pepper noise caused by transmission or ADC errors, as well as others.

The earliest attempts to reduce noise started with image processing in the spatial and frequency domain, but soon after they moved to the wavelet transform domain, much of the interest being attributed to Donoho [8].

To understand the dimension of image denoise, it is essential to give a broad categorization of the various techniques used in academia. In the area of probability theory, statistics differential equations, the list of denoising algorithms can contain spatial domain filtering, random fields, domain thresholding, statistical models, dictionary learning methods, diffusion methods, and hybrid methods [11].

Besides these methods, this listing also includes spatial adaptive filters, stochastic analysis, morphological analysis, statistical estimators, and order statistics [12].

The article [14] mentions that methods based on spatial domain filters are suitable for decreasing the high-frequency noise, in detriment of blurring the contrast information. In the case of dictionary learning, the method proposes unnecessary and overcomplete dictionaries. For this scenario, the load is computational heavy. Hybrid methods boost the quality of denoised images and have an ample increment in PSNR [10].

One of the most popular and efficient techniques mentioned in scientific research block-matching with 3D filtering (BM3D) [2]. BM3D reached the theoretical limit for image denoising, but AI techniques have the potential for advancement in the field [13]. Knaus and Zwicker obtain another remarkable result in this domain. They propose dual-domain image denoising (DDID), a simple algorithm that produces exceptional results [15].

A comprehensive survey of the various denoise techniques used in different domains of activity, as well as their evolution over time is presented in Motwani et al. [4].

In recent years, the focus moved to more exotic methods, such as 3D filtering in [2] and the use of deep neural networks for noise removal [3][9] that have promising results.

1.2. Problem motivation

Image denoising has many applications, from restoring old pictures or enhancing new ones, to correcting acquisition ADC errors on digital cameras or corruptions caused during image transfer. Furthermore, image denoise became even more of an interesting problem as it nowadays is an essential preprocessing step in real-time ray tracing, by approximating missing data that could not be otherwise computed in real-time.

2. Proposed method

As stated before, this paper recommends a method for filtering specific types of noise, precisely Gaussian and salt-and-pepper by identifying them individually and applying filtering techniques. The characteristics of the noise are adapted and then combined through a voting-based algorithm. The first step is removing salt and pepper noise and then proceeding with analyzing the image in order to remove the Gaussian noise.

2.1. Salt and pepper

Since salt and pepper noise is characterized by impulses of very high and very low pixel values, a popular and straightforward method of eliminating the noise is by using a median filter, which is a non-linear filter for each pixel in the original image. It computes and substitutes the original pixel with the median value in the neighborhood of that pixel (1), where it represents a user-defined neighborhood around location [6].

$$y[m,n] = median\{x[i,j],(i,j) \in \omega\} \quad (1)$$

Although this filter can successfully remove most or even all the salt and pepper noise, it has the disadvantage that it tends to degrade the rest of the original image as well. In this step, the mission is to replace only the noisy pixels of the image, leaving as many useful pixels unchanged as possible. The essential property of the salt and pepper noise is used to achieve that. It has values either very close to zero or one (the maximum intensity value). With that in mind, it can be proposed to create a mask equal in size to the original image, having values of one on the positions corresponding to noisy pixels (values very close to zero or one) and values of zero on the rest of the mask. After that, it can proceed to apply the median filter to a copy of the original image. Lastly, the pixels from the median image are copied to the original image only on the positions where there is a value of one on the mask. As expressed in (2), where y is the resulting image, x is the initial image, *t* is the median filtered image, and *i* is the mask. A comparison can be seen in Figure 2.

Figure 2. From left to right: a. Salt and pepper noise; b. median filtered image; c. image filtered with our method

$$y[m,n] = \begin{cases} x[m,n], i[m,n] = 1 \\ t[m,n], i[m,n] = 0 \end{cases} \quad (2)$$

Gaussian noise, on the other hand, has the property that it can be averaged out by applying an averaging filter such as a box filter or a Gaussian filter. Nevertheless, since these types of filters behave like a low-pass filter, all high-frequency signals in the image will be dampened, which can be observed in the resulting image as smoothed edges. Since this is an undesirable effect, denoise filters, like the bilateral filter, have been developed to preserve the edges.

The method assumes a Gaussian distribution of noise in the input image and uses a method from Gonzalez et al. [5] to identify the standard deviation of the Gaussian noise distribution, which it will then pass to the bilateral filter as a standard color deviation. For the pixel diameter and space standard deviation, the values of 25 pixels and 21 pixels are chosen empirically. First, it uses the input image from which the salt & pepper noise has been filtered out, and it creates a preliminary heavily filtered image by applying the bilateral filter with some predefined parameters. At this point, the intention is to remove the Gaussian noise from the image without concerns of losing image quality. Afterward, it can subtract the filtered image from the noisy image and obtain the high-frequency signal of the noisy image. The assumption here is to be mostly the Gaussian noise. After, it proceeds to compute the standard deviation of the noise map and use this value as the standard color deviation of bilateral filtering on the original image, having the other two parameters, the pixel neighborhood diameter and the spatial standard deviation set to predefined values.

When putting both filters together, the result is an adaptive solution for filtering both types of image noise, represented in Figure. 3. In contrast, the results of applying the combined algorithms can be observed in Figure. 4.

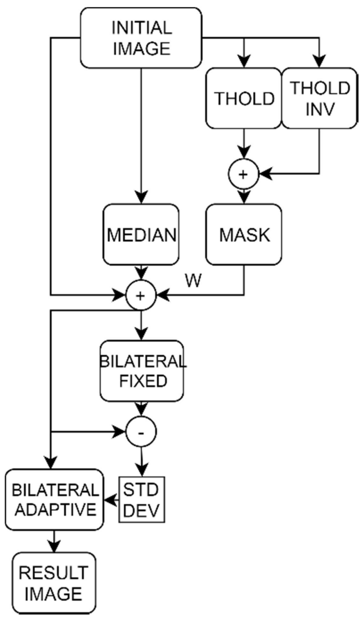

Figure 3. Proposed solution

3. Performance measurements

The focus of this section is to compare the result of the proposed algorithm with the Fast Non-Local Mean Denoise (FNLMD) implemented in OpenCV. We are interested in the MSE, PSNR, and compute time for the two algorithms, given an image of a specific size and standard deviation for the Gaussian noise map, which we combine with the original image through an averaging. The images are grayscale single-channel, ranging from 490x733 to 3840x2160 pixels in size. The results are represented in Table 1, and a sample of the resulting pictures can be observed in Figure 5.

Figure 4. From left to right: a. Image affected by salt and pepper and Gaussian noise; b. Image filtered without salt and pepper noise filter; c. Image filtered using both filtering methods

The observation here is that the proposed method performs well for small amounts of Gaussian noise, but it is outperformed by FNLMD as the standard deviation of the noise increases. It also performs very well when removing salt and pepper noise, removing it almost completely, while FNLMD fails to remove most of this type of noise. Another aspect that stands out is the compute time, which can be up to two orders of magnitude smaller for usual images.

Figure 5. From left to right: a. Image affected by salt and pepper and Gaussian noise; b. Image filtered without salt and pepper noise filter; c. Image filtered using OpenCV FNLMD

Algorithm	**FNLMD and Voting Performance Comparison**				
	Image size	***σ***	***MSE***	***PSNR***	***Time(s)***
Noisy	490x733	0.05	0.0102	19.883	N/A
	490x733	0.15	0.029	15.258	N/A
	1920x1080	0.05	0.0109	19.588	N/A
	1920x1080	0.15	0.0304	15.159	N/A
	3840x2160	0.05	0.011	19.494	N/A
	3840x2160	0.15	0.0307	15.116	N/A

Voting-Based	490x733	0.05	0.0016	27.845	0.325
	490x733	0.15	0.007	21.305	0.267
	1920x1080	0.05	0.003	25.068	0.625
	1920x1080	0.15	0.009	20.107	0.626
	3840x2160	0.05	0.003	24.25	2.447
	3840x2160	0.15	0.01	19.826	2.519
FNLMD	490x733	0.05	0.005	22.704	9.346
	490x733	0.15	0.003	25.109	8.74
	1920x1080	0.05	0.008	20.508	37.555
	1920x1080	0.15	0.008	20.816	44.374
	3840x2160	0.05	0.007	21.139	173.623
	3840x2160	0.15	0.009	20.430	175.532

Table 1 FNLMD and voting algorithm performance comparison

4. Conclusions

The proposed voting-based image denoise algorithm performs well for removing Gaussian or salt and pepper noise from an image. When compared to a well-established algorithm, such as FNLMD, it can obtain similar or even better results in certain situations, such as when removing salt and pepper or small amounts of Gaussian noise.

As future work, the algorithm can be improved by replacing the current solution for Gaussian noise removal with another that would only require the noise distribution as input. A promising future approach could be using the method described in [7].

Also, a further development may be the integration of the proposed algorithm in a voting-based image processing pipeline, using elements from [16-18].

Acknowledgements

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0689 / „Lib2Life- Revitalizarea bibliotecilor și a patrimoniului cultural prin tehnologii avansate” / "Revitalizing Libraries and Cultural Heritage through Advanced Technologies", within PNCDI III.

References

[1] Peter Corke, *Robotics Vision and Control*, Springer, 2011, DOI: 10.1007/978-3-642-20144-8.

[2] Dabov Kostadin, Foi Alessandro, Katkovnik Vladimir, Egiazarian Karen, *Image restoration by sparse 3D transform-domain collaborative filtering*, 2008 https://doi.org/10.1117/12.766355.

[3] Y. Chen, T. Pock, *Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1256-1272, 1 June 2017. DOI: 10.1109/TPAMI.2016.2596743.

[4] Motwani Mukesh, Gadiya Mukesh, Motwani Rakhi, Harris Frederick, *Survey of image denoising techniques*, 2004.

[5] Barry R. Masters, Rafael C. Gonzalez, Richard Woods, Digital Image Processing, Third Edition, Journal of Biomedical Optics 14 (2), 2009, DOI: 10.1117/1.3115362.

[6] Nikos Drakos, Computer Based Learning Unit, University of Leeds, Ross Moore, Mathematics Department, Macquarie University, Sydney, URL: http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node2.html, Accessed in April 2020.

[7] V.R. Vijaykumar, Vanathi P.T., Kanagasabapathy Prathapan*, Fast and Efficient Algorithm to Remove Gaussian Noise in Digital Images*, IAENG International Journal of Computer Science. 37. 78-84, 2010.

[8] D. L. Donoho, *De-noising by soft-thresholding*, IEEE Transactions on Information Theory, vol. 41, no. 3, pp. 613-627, May 1995, DOI: 10.1109/18.382009.

[9] Junyuan Xie, Linli Xu, Enhong Chen, *Image denoising and inpainting with deep neural networks*, 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA, 341–349, 2012.

[10] Fan L., Zhang F., Fan H., *Brief review of image denoising techniques*, Vis. Comput. Ind. Biomed. Art 2, 7, 2019, DOI: 10.1186/s42492-019-0016-7.

[11] Hong-Ying Yang, Xiang-Yang Wang, Pan-Pan Niu, Yang-Cheng Liu, *Image denoising using nonsubsampled shearlet transform and twin support vector machines*, Neural Networks, Volume 57, 2014, Pages 152-165, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2014.06.007.

[12] M. Elad, M. Aharon, *Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries*, IEEE Transactions on Image Processing, vol. 15, no. 12, pp. 3736-3745, Dec. 2006, DOI: 10.1109/TIP.2006.881969.

[13] C. Knaus, M. Zwicker, *Progressive Image Denoising*, IEEE Transactions on Image Processing, vol. 23, no. 7, pp. 3114-3125, July 2014, DOI: 10.1109/TIP.2014.2326771.

[14] Bhawna Goyal, Ayush Dogra, Sunil Agrawal, B.S. Sohi, Apoorav Sharma, *Image denoising review: From classical to state-of-the-art approaches*, Information Fusion, Volume 55, 2020, Pages 220-244, ISSN 1566-2535, DOI: 10.1016/j.inffus.2019.09.003.

[15] C. Knaus and M. Zwicker, *Dual-domain image denoising*, 20th IEEE ICIP, Sep. 2013, pp. 440–444.

[16] Costin-Anton Boiangiu, Radu Ioanitescu, Razvan-Costin Dragomir, *Voting-Based OCR System*, The Journal of Information Systems & Operations Management, Vol. 10, No. 2, 2016, pp. 470-486.

[17] Costin-Anton Boiangiu, Mihai Simion, Vlad Lionte, Zaharescu Mihai – *Voting Based Image Binarization*, The Journal of Information Systems & Operations Management, Vol. 8, No. 2, 2014, pp. 343-351.

[18] Costin-Anton Boiangiu, Paul Boglis, Georgiana Simion, Radu Ioanitescu, *Voting-Based Layout Analysis*, The Journal of Information Systems & Operations Management, Vol. 8, No. 1, 2014, pp. 39-47.