MULTI-ALGORITHM IMAGE DENOISING

Georgiana-Rodica CHELU [1*]
Marius-Adrian GHIDEL [2]
Denisa-Gabriela OLTEANU [3]
Costin-Anton BOIANGIU [4]
Ion BUCUR [5]

ABSTRACT

In spite of the thorough research that has been done in the field of image denoising, a generic algorithm able to preserve the details of an image at an acceptable level has not been yet discovered. Most methods account for a specific class of noise and provide suitable results only if the implicitly-determined control parameters of the image correspond to the method's assumptions. Furthermore, many such methods reside on the presumption that noise is spatially-invariant and do not treat the other case.
The purpose of this paper is to analyze the classical methods used in image denoising, to observe their limitations in order to decide how mixing different algorithms might correct their undesired behaviors and to set the scene for a new method appropriate for image denoising that would yield better results on a more varied set of images.

KEYWORDS: *Image Denoising, Image Processing, Merging Technologies, Random Noise, Fixed Pattern Noise, Banding Noise, Salt-and-Pepper Noise.*

INTRODUCTION

Images are 2-dimensional representations of the visible light spectrum and are stored on computers as multi-dimensional arrays where the dimension depends on whether the image is colored (fig. 1) or black and white (fig. 2).

For simplicity, the third dimension is also encoded as a tuple of red, green and blue components, corresponding to the RGB color space.

A pixel is represented by a pair *(i, v(i))*, where *v(i)* is the value at *i*, the pixel's position, and it is the result of measuring the light intensity by using a charge-coupled device (CCD) matrix and a focus system (the lens).

[1*] corresponding author, Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, georgiana.chelu93@gmail.com
[2] Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, ghidel.ady@gmail.com
[3] Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, denisa.olteanu@asaff.ro
[4] Professor PhD Eng.,"Politehnica" University of Bucharest, Bucharest, Romania, costin.boiangiu@cs.pub.ro
[5] Associate Professor PhD Eng., "Politehnica" University of Bucharest, Bucharest, ion.bucur@cs.pub.ro,

In the case of black and white images, *v(i)* is a real value representing a shade of grey.

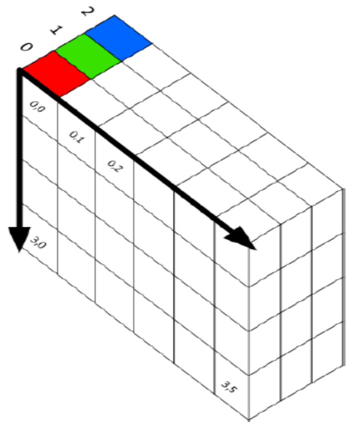


Figure 1. Color image. Image taken from [9]

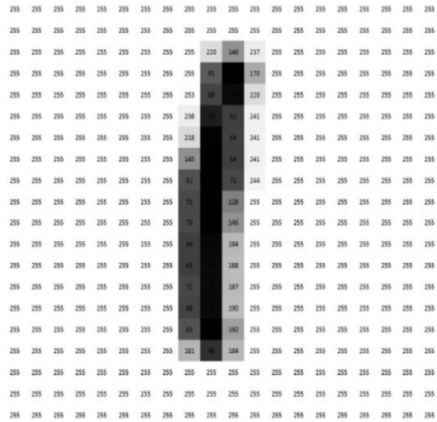
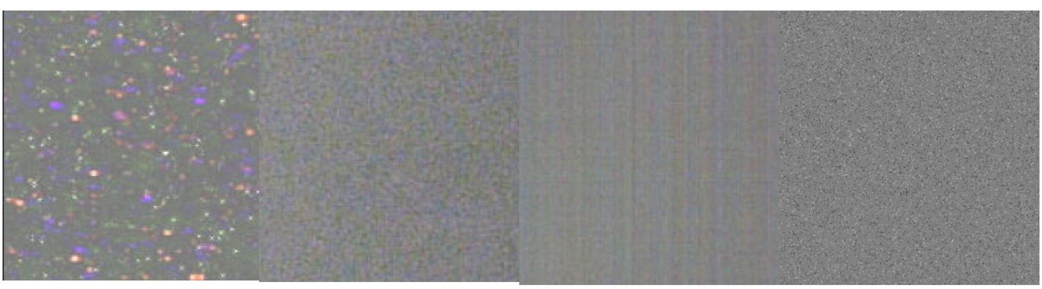


Figure 2. Grayscale image. Image taken from [9]

Each element of the CCD matrix counts the number of photons which are incident on it during the exposure time. In the case of a constant light source, it has been proven that the number of photons incident on each captor varies according to the central limit theorem, that is for *n* incident photons, the fluctuation is around $\overline{\quad}$. This accounts for **random noise** that is characterized by intensity and color fluctuations, it is determined by ISO speed and it is **always present** at any exposure length.

Apart from random noise, there are other two types of noise: fixed pattern noise and banding noise. **Fixed pattern noise**, commonly known as hot pixels, occurs when the not adequately cooled captors receive heat spurious photons [10]. What makes this type of noise easy to remove is that it will show almost the same distribution of hot pixels provided that the image is taken under the same conditions. For this reason, it will not make the subject of the paper.

Banding noise is camera-dependent and it is introduced when data is read from the camera sensor [10] or after significant noise reduction and it will also not make the subject of the research.



3.1. hot pixels 3.2. random noise 3.3. banding noise 3.4. salt-and-pepper

Figure 3. Images taken from [10]

Salt-and-pepper is another type of noise either caused by camera sensors, memory loss or by faulty conversions between analog and digital images. It is characterized by the fact that salt-and-pepper pixels can only take the maximum or the minimum values in the dynamic range.

Random noise is the most difficult to extract, as it is often hard to differentiate it from fine textures such as dirt and therefore removing the noise results in removing those details as well (fig. 4). Therefore, denoising often leads to other undesired effects such as blur, the staircase effect, the checkerboard effect, the wavelet outliers, etc. depending on the denoising algorithm applied. The difficulty of the problem is also increased due to the uneven distribution of random noise.

This does not come as a surprise when using algorithms that are either based on one of the noise models presented above or on a generic image smoothness model.

4.1. Image taken from [9]   4.2. Blur Effect Inflicted by Denoising using NLM

Figure 4

As it was argued before, the purpose of image denoising is to reconstruct an image that has been subject to noise. One can mathematically represent the noisy image y as a sum between the original image x and the noise n , a pixel:

$$y = + n \tag{1}$$

The aim of any denoising algorithm is to reduce or completely remove the noise, n(i), in order to obtain the original signal, x(i).

RELATED WORK

A possible class of algorithms consists in algorithms which filters pixel sets in which individuals have a certain degree of similarity. Such an algorithm was developed by Perona-Malik and it is related to anisotropic filtering. Another category of denoising methods uses training sets to derive image statistics about certain coefficients (Wavelet filtering) and attempt to modify these coefficients in order to diminish the noise.

Other methods such as the **Non-Local Means** algorithm use sampling techniques to evaluate the areas in an image that appear to be similar with respect to the structure, but not to the quantity of noise. No matter the approach, the aim is to obtain noise-free images without altering the original image.

Classical methods such as **Gaussian** or **Wiener filtering** (Yaroslavsky) work by separating the image into the two parts, the smooth and oscillatory part [8]. The drawback is that this might lead to losing the fine edges present in the original image which is un undesired outcome. Another algorithm that yields similar unwanted results when it comes to preserving the fine edges of an image is **Perona-Malik**, which filters the noise by using anisotropic diffusion. An improved approach could be robust anisotropic image smoothing.

A newer and improved approach uses local adaptive filters to analyze the image in a moving window, compute its spectrum for each position but only use the value at the central pixel of the window [2].

The **wavelet thresholding** approach assumes that noise is represented with small wavelet coefficients that should be discarded if under a certain threshold. The drawback of this method is that it outputs images in which important edge coefficients are also cancelled which again leads to a loss of fine details and the appearance of spurious pixels [2].

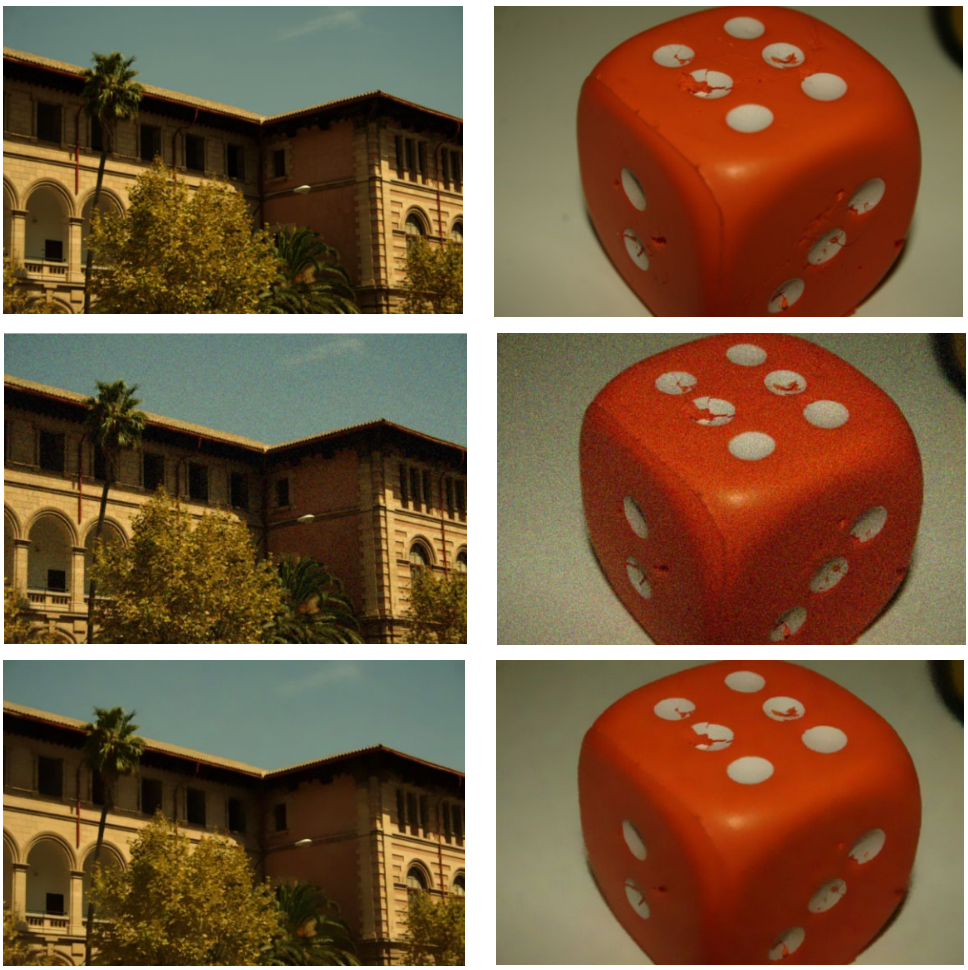


Figure 5: Non-local means applied on two images: first row, original image; second row, noisy image ($\sigma = 25$); third row: image denoised using the non-local means algorithm

Among the promising methods for a certain class of images is the **non-local means** approach proposed by Antoni Buades, Bartomeu Coll and Jean-Michel Morel as it is fairly easy to implement and yields qualitative results. The method suggests that one should choose a pixel, look for similar neighborhoods to the one surrounding the pixel and replace the pixel by the average of the centers of the neighborhoods. This is an algorithm based on exploiting the self-similarity that it can be seen in most natural images.

Total variation denoising techniques assume that images affected by noise have high total variation or the integral of its absolute gradient is high and therefore attempt to reduce the total variation. These algorithms yield impressive results as they remove the noise without affecting the edges as most of the above presented algorithms do [12].

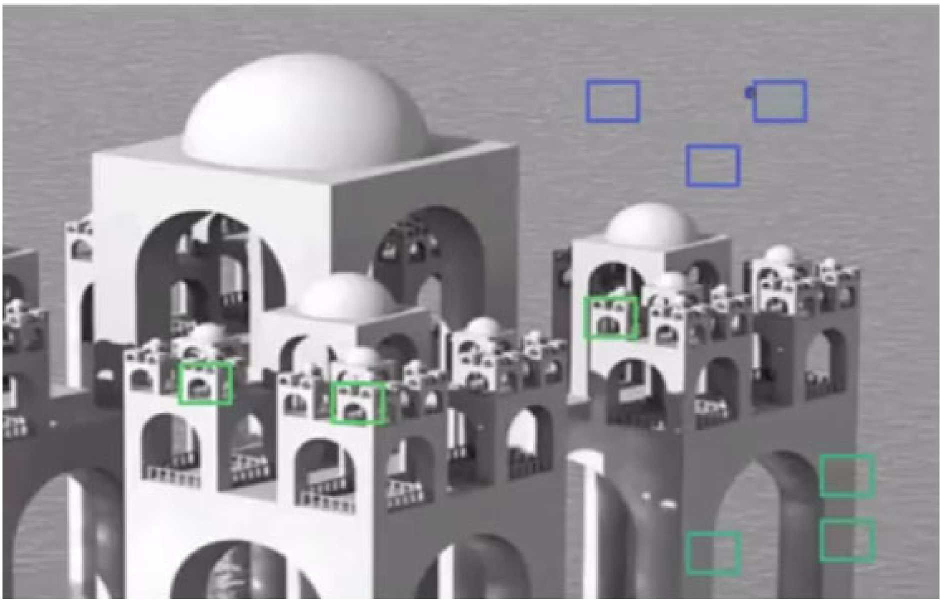


Figure 6: Examples of similar neighborhoods. Image taken from [12]

ALGORITHMS' ANALYSIS

No matter the method that is being used, the final goal is to obtain an image as free of noise as possible without loss of details and side effects such as blurring. But each and every algorithm presented above has its own specific limitations.

For example, anisotropic smoothing methods can preserve strong edges but cannot be used for smooth patterns and textures while methods based on wavelet coefficient statistics yield the expected results only for limited types of input images. The non-local means algorithm works well for repetitive image patches but underperforms when it comes to preserving details in non-repetitive areas as it treats the smooth regions and the edges in the same manner. Furthermore, it is worth observing what type of noise each algorithm is removing the best. Strategies that perform best for low noise levels may not perform as well for high noise levels and the same happens for uniformly or non-uniformly distributed types of noise.

As a conclusion, there is a strong need for simple, unsupervised, efficient denoising algorithms that performs well on all kinds of input images. A great number of algorithms performs similarly despite using different approaches and thus merging seems to be a suitable solution for improving the already existing methods. In what follows, a method based on mixing approaches that could account for some drawbacks of the current state-of-the-art denoising algorithms will be presented.

THE PROPOSED SOLUTION

A solution will be proposed in which five of the previously described algorithms will be selected and applied on various images. These images will be black and white images, colored and also on images with different levels of brightness/color uniformity.

Which algorithms performs better on each of these images can be categorized and assigned some weights. Having these weights the image can be split after some criteria:

1. Highlights/shadows and non-highlights/shadows
2. Uniform and non-uniform color

The average value of this point will be given by the weight of each algorithm on each region. The average can be tweaked by using removing extreme values or values that are outside of a calculated interval.

The expectation is to get a better image with each algorithm have a greater impact on each type of surface: e.g. on the sky, where all pixels have the same value the average of the neighboring pixels will give us the best approximation (this could also go for shadows). On the other hand if the neighboring pixels have really different values this algorithm will fail.

Figure 7: Sample images from the proposed dataset

Five algorithms have been chosen as input for further research and mixing-based improvements:

- the Non-Local Means implementation from OpenCV
- the TV - L1 (total variation denoising) from OpenCV
- the wavelet thresholding implementation from scikit
- the bilateral filtering from scikit
- and the median filtering from scipy

The image dataset was divided into 4 categories in order to better assess the performance of the algorithms, each subset being characterized by:

- dominant colors
- few colors
- many colors
- images containing people

THE PREPROCESSING STAGE

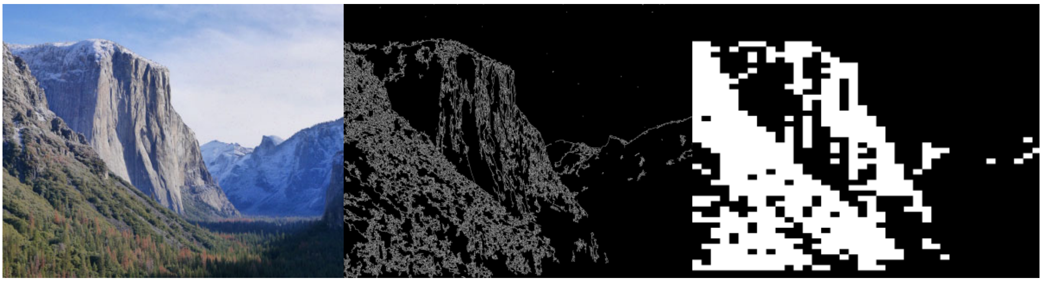
Two types of noise were added (salt and pepper and Gaussian noise) to each image.

It will have a lower weight assigned or even take it out of the average in case it underperforms.

Edge detection

Edge detection is a method used for identifying those points in an image at which the brightness of the image changes sharply. These points are grouped into a set of curves named edges. The Canny Edge Detection algorithm developed by John F. Canny from *OpenCV* was used. The first step of this approach is noise reduction, since edge detection is susceptible to noise. To reduce this susceptibility, the noise is removed by applying a 5x5 Gaussian filter. Then the algorithm computes the intensity gradient of the image. A Sobel kernel filter is applied in the horizontal and vertical directions to obtain the first derivatives in horizontal (Gx) and vertical (Gy) directions.

After obtaining the gradient magnitude and direction, the program scans the whole image in order to remove any unwanted pixels which may not be a part of an edge. The algorithm is then checking for each pixel if it is a local maximum in its neighborhood in the direction of gradient.



Original noisy image Obtained edges Obtained mask

Figure 8

The next step is called hysteresis thresholding and it decides which edges we can keep. For this, we need two threshold values, *min* and *max*. Any edge having the intensity gradient greater than *max* is definitely an edge. Those smaller than *min* are definitely not edges, so they can be discarded. Those that remain obey to the following rule: if they are connected to a "real-edge", they are a part of the edge. Otherwise, they are discarded.

Based on the edges that were obtained by applying this method, a mask was created that the program is using to apply the algorithm to certain areas of the image.

RESULTS AND CONCLUSIONS

Figure 9. The image with Salt-and-pepper noise

Figure 10. Image denoised with total variation algorithm

Figure 12 Denoised with total variation and Non-Local Means algorithms

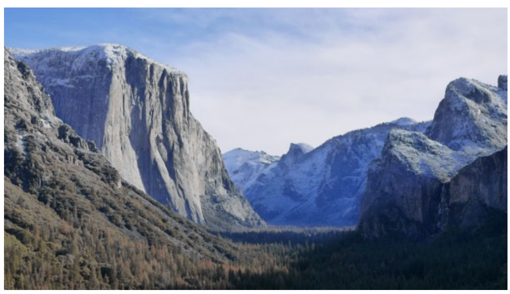


Figure 13. Denoised with bilateral filtering algorithm

Figure 11. Image denoised with Non-Local Means algorithm

Figure 14. Denoised with wavelet thresholding algorithm

To assess the results of applying the algorithm, the RMSE was used to compare the original image (before applying the noise) and the resulting image (the one obtained after applying the method to the noisy image).

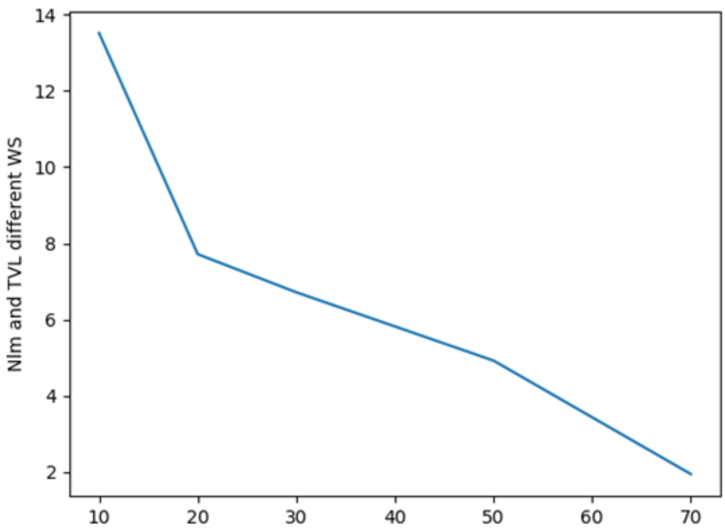


Figure 15. Processing results interpretation (Ox: Number of windows; Oy: RMSE)

The standard deviation is computed between the original image and the result image. A higher number of windows leads to a better denoising. The drawback is the computation time that increases in an exponential way with the number of the windows. The two denoising algorithms used are Non-Local Means and TV - L1 (total variation denoising).

Several voting-based processing methods have been tried over time [13][14][15][16]. The presented approach, despite that fact that does not make use of a real voting system in order to select the final result, has the same mechanisms presented in the voting-based applications: the use of totally different sub-optimal approaches to solve a specific problem and, in the end, the intelligent merge of every output into the final result.

ACKNOWLEDGEMENT

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 si a patrimoniului cultural prin tehnologii avansate" / "Revitalizing Libraries and Cultural Heritage through Advanced Technologies", within PNCDI III.

REFERENCES

[1] K. Sivaramakrishnan and T. Weissman, "Universal denoising of discrete-time continuous-amplitude signals," in "Proceedings of the IEEE International Symposium on Information Theory"

[2] A. Buades, B. Coll And J.M. Morel, "Image Denoising Algorithms, With A New One"

[3] Antoni Buades, Bartomeu Coll, Jean Michel Morel, "On image denoising methods"

[4] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising"

[5] Kamakshi Sivaramakrishnan, Tsachy Weissman, "A Context Quantization Approach to Universal Denoising"

[6] G. Motta, E. Ordentlich, I. Ramirez, G. Seroussi, and M. J. Weinberger, "The DUDE framework for continuous tone image denoising"

[7] Nima Khademi Kalantari, Pradeep Sen, "Removing the Noise in Monte Carlo Rendering with General Image Denoising Algorithms"

[8] Hyuntaek Oh, "Bayesian ensemble learning for image denoising"

[9] Ratan, Rajeev. "Mastering Computer Vision with OpenCV in Python". Udemy, Inc. Web. November 2017

[10] "Digital Camera Image Noise" Digital Camera Image Noise: Concept and Types, Available at: www.cambridgeincolour.com/tutorials/image-noise.htm. Accessed on: 1 March 2018

[11] Estrada, Francisco. Fleet, David. Jepson, Allan. "Stochastic image denoising"

[12] Glasner, Daniel. Bagon, Shai. Irani, Michal. "Super-Resolution from a Single Image". ICCV. 2009.

[13] Costin-Anton Boiangiu, Radu Ioanitescu, Razvan-Costin Dragomir, "Voting-Based OCR System", The Proceedings of Journal ISOM, Vol. 10 No. 2 / December 2016 (Journal of Information Systems, Operations Management), pp 470-486, ISSN 1843-4711

[14] Costin-Anton Boiangiu, Mihai Simion, Vlad Lionte, Zaharescu Mihai – "Voting-Based Image Binarization" - , The Proceedings of Journal ISOM Vol. 8 No. 2 / December 2014 (Journal of Information Systems, Operations Management), pp. 343-351, ISSN 1843-4711

[15] Costin-Anton Boiangiu, Paul Boglis, Georgiana Simion, Radu Ioanitescu, "Voting-Based Layout Analysis", The Proceedings of Journal ISOM Vol. 8 No. 1 / June 2014 (Journal of Information Systems, Operations Management), pp. 39-47, ISSN 1843-4711

[16] Costin-Anton Boiangiu, Radu Ioanitescu, “Voting-Based Image Segmentation”, The Proceedings of Journal ISOM Vol. 7 No. 2 / December 2013 (Journal of Information Systems, Operations Management), pp. 211-220, ISSN 1843-4711.