Development of Methods for Image Filtering in Noise and their Implementation for a Web Service

Dmytro Palahin
[1]Universite Sorbonne Paris Nord Institute Galilee,
99, avenue Jean-Baptiste Clément, 93430 Villetaneuse, France

E-mail: dmytro.palahin@gmail.com

Abstract. Known image processing software systems use filtering methods such as the Gaussian filter, the median filter, and others, which often do not have satisfactory performance in processing certain types of noise. This leads to a partial loss of the useful signal and a deterioration in image quality. The work is devoted to improving the quality of image filtering in various kinds of noise. A model of additive interaction of signals in additive impulse noise is proposed. A new method of least finite differences (MLFD) and its modification for image filtering has been developed. An interactive web service for the implementation of MLFD was created. Various filtering methods have been proposed to reduce the effect of impulsive noise on images. The proposed data processing is based on a priori information about the type of noise. The web service is a flexible and efficient solution for filtering digital images and is not available in well-known graphics packages. The new web service is a good compromise between the quality of filtering and the simplicity of the solution compared to complex and resource-intensive systems for graphic image processing.

Keywords — digital images, impulse noise, additive Gaussian noise, data filtering and recovery, gradient analysis

I. Introduction

Image recovery and filtering systems are an important and defining part of modern computer, information and telecommunication systems for surveillance, diagnostics, monitoring and control. The development of such systems is characterized by increased requirements for information processing, increasing complexity and functionality. The problems that arise in the improvement of this class systems are associated not only with technological upgrades, but also with the creation of advanced signal processing methods.

There is a significant group of technical systems that are designed to receive and process images and video data [1-3]. Distortion of data from various impulse noise can occur when transmitting images over communication channels [4-6]. The most common type of such interference is random additive noise, which is considered as statistically independent from the data. The interaction of useful signal and noise is characterized by impulse and fluctuation noise in telecommunications systems and communication channels, and requires specialized methods to remove them [7, 8].

Areas of digital image application processing methods are significantly expanded and are actively used in industry, art, medicine and more. Transmission of digital images through communication channels, the formation of better-quality images, satellite image processing automation, medical imaging processing etc. are the subject of modern research and development [9-18], including the use of various statistical methods [19, 20].

The brightness values of neighboring image pixels (for a color image take RGB-components) are processed in modern filtering methods when estimating the useful signal at any point in the image. This approach is based on the principle of similarity of the useful signal at these points [7, 9]. The concept of adjacent points is quite conditional and is chosen depending on the type of signal and noise. In addition, the level of influence of far and near points on the final decision made by the image recovery filter can be taken into account.

Various image filtering methods are used only for certain types of noise. Different optimality criteria are also used for data processing, which complicates their overall analysis and comparison [14, 15].

Classical filtering methods are known in the literature, such as the median filter, Gauss filter, Wiener filter [3, 7, 11] and others. These methods are widely used in well-known modern software packages for image processing. Such methods often do not have satisfactory efficiency for processing specific types of noise and often lead to partial loss of useful signal and quality of data processing.

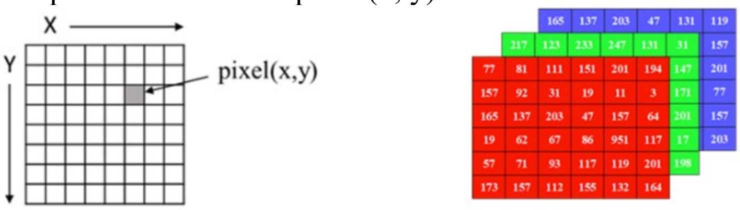
The main task of this paper is to develop and analyze the new methods of image processing – the method of least finite differences (**MLFD**) and its modification, which improves the results of data recovery compared to existing approaches. Interactive **Web Service** is created on the basis of the developed new methods of data recovery.

The proposed filtering methods allows us to effectively reduce impulse noise in images, taking into account a priori information about the noise without distorting the data. The results of image filtering in noise by the new methods have better characteristics compared to other classical methods that are widely used in many applications.

II. Methods Of Filtering Noisy Digital Images

Digital image filtering is based on the fact that the brightness of the image changes more slowly in spatial coordinates compared to the noise function [1, 7]. The brightness values of adjacent pixels (or the values of RGB components) are taken into account when processing the image frame. The brightness level of a pixel at some point in the frame is based on the principle of similarity of pixel values in some neighborhood. The brightness value of a pixel at some point in the image can be formed by the brightness value of the nearest pixels (neighboring pixels), or calculated as a more complex structure of far distant pixels. Then, the effect (or weight) of the pixels from different distances on the decisions made by the filter for a given point of the frame will be different. Digital image filtering is based on the rational selection of data from the operating point and its surroundings.

One frame of a digital image is represented as a two-dimensional matrix of dimension n×m (Fig.1.a), where the brightness of the pixel is denoted as pixel (x, y).



a) b)

Fig.1. a) Digital representation of the image in the form of a two-dimensional Data Set; b) Digital representation of a color image in RGB format

For black-and-white images, the pixel brightness level is in the range from zero (black) to 255 (white). Three two-dimensional arrays are used for the color image (one for RGB channels) (Fig. 1.b). The image of one frame for the studied pixel (x, y) with coordinates x, y and its neighbors is presented in Fig.2.a. Fig.2.b shows the transformation from a two-dimensional array to a one-dimensional one for computer data representation and processing.

a) b)

Fig.2. Two-dimensional digital image model (a) and its transformation into a one-dimensional array for computer data processing (b).

The result of linear filtering is defined as a linear combination of input data:

$$x^*(i,j) = \sum_{(i_l,j_l)\in S} \sum a(i_l,j_l) * y(i-i_l, j-j_l), \tag{1}$$

where $x^*(i,j)$ - the filtered value of the useful signal at a point with coordinates (i, j) on the set of points S, $a(i_l, j_l)$ - coefficients that represent the two-dimensional impulse response, $y(*)$ - input data of image.

The most common optimality criterion for finding the quality of image filtering is the minimum standard deviation and is represented by the functional:

$$E\left\{\left[x(i,j) - \sum_{(i_l,j_l)\in S} \sum a(i_l,j_l) * y(i-i_l, j-j_l)\right]^2\right\}, \tag{2}$$

where the symbol $E\{*\}$ indicates the operator of mathematical expectation.

The synthesis of the optimal filter is to find an image processing algorithm in which the mean square of the error between the original $x(i, j)$ and filtered values $x^*(i,j)$ is minimal:

$$\min\xi(i,j) = x(i,j) - x^*(i,j).$$

The simplest option to filtering is to assign a new value to the central pixel as the arithmetic mean of all its neighbors, whose values differ from the central value by no more than a certain threshold. Examples of such filters are rectangular and circular averaging filter [7, 9]. More effective noise reduction can be achieved if the effect of the pixels on each other will decrease with distance from the law given by the Gaussian function. In this case, we are talking about the implementation of *Gaussian filtration*.

The use of *median filtering* is a standard way to reduce impulse noise. For each pixel in a certain neighborhood (window) is the median value and is assigned to the pixel that is processed. The value of the recovered image at the median filtering at any point (x, y) will look like:

$$\overline{f}(x,y) = med\{g(x,y)\}. \tag{3}$$

This type of filtering is widely used in many software applications for filtering and recovering data, including *Adobe PhotoShop, Paint, GIMP*, etc., which do not demonstrate satisfactory quality filtering images in impulse noise (see Fig.3).

Another example of using the widely used *GIMP* package also does not provide satisfactory results at different settings - no noise reduction, image blur is observed.

Other filtering tools were considered, including online services: *Remove Noise Online* (https://pinetools.com/remove-noise-image), *Image Noise Reduction Online* (https://online-photo-converter.com/image-noise-reduction) and others. They were involved in the image filtering experiment and did not show a satisfactory image recovery results – also no noise reduction, there is significant blurring of the image.

Original **Median filtration (radius 1)** **Median filtration (radius 3)**

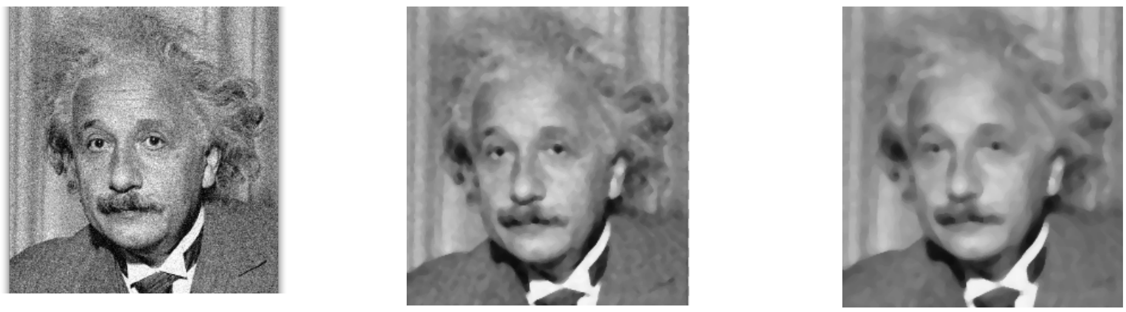


Fig.3. Results of filtering of a noisy image (noise type "Grayscale") by the median filter in the Adobe PhotoShop package (image taken from the source https://www.census.gov/history/www/homepage_archive/2019/march_2019.html)

The proposed known interactive tools do not provide high quality data processing of noisy data. Thus, there is a need to develop new methods and tools for implementing image filtering in impulse noise.

III. The Least Finite Difference Method For Filtering Impulse Noise In Images

All linear filtering algorithms smooth out sharp changes in the brightness of processed digital images. Such filtering procedures are optimal with normal distribution of observed data. The analysis shows that the actual image data that is processed have different character distributions.

One of the main reasons is the presence in the images of various boundaries, brightness differences, etc. Thus, the images are poorly represented in the form of Gaussian models of brightness distributions. This is one of the reasons for the poor transfer of boundaries in images when applying linear filtering.

Distortion in the form of impulse noise of various kinds is often observed in data and image processing. In this case, the images show white, black or gray dots, randomly scattered throughout the frame. The use of linear filtering in this case is inefficient - each of the input pulses responds in the form of a pulse characteristic of the filter, and their combination contributes to the propagation of the filtering error over the entire frame area.

The paper implements a new algorithm for impulse noise filtering based on the developed ***Least Finite Difference Method*** (**MLFD**). The proposed filtering method differs from the median filtering and is designed to reduce the impact of pulsed noise of different types while maintaining clarity and quality.

The proposed method demonstrates the replacement of the value studied pixel with coordinates (x, y) by a new value so that the difference in brightness (or RGB value for color images) of its value with possible neighbors (fig.6 - «+», «**X**») will be less. This principle is illustrated in Fig.4 and Fig.5.

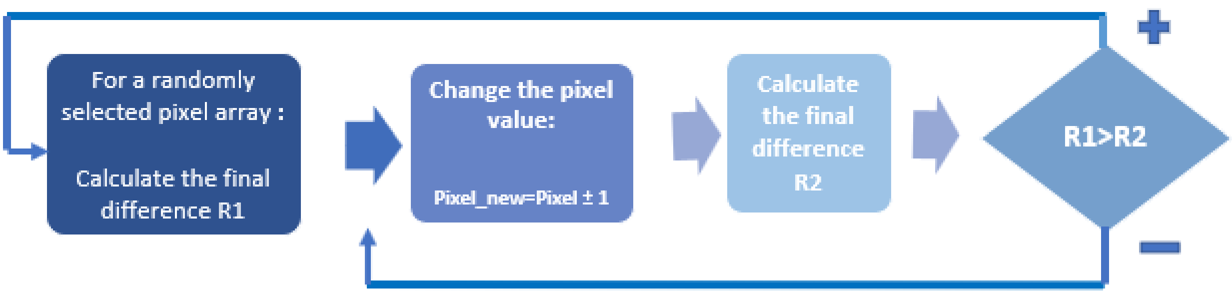


Fig.4. Algorithm for replacing the central pixel of two-dimensional digital image by the Method of Least Finite Difference (**MLFD**)

The pixel block as the kernel ("**3*3**") has been shown in the Fig.5. The central value of the pixel (highlighted in pink) is subject to filtering (**I**). Various combinations are considered as adjacent pixels, in particular the combination "+", indicated in Fig. 5.a (**II**), or a combination of "**X**", denoted as Fig.5.b (**II**). As a result of choosing one of these combinations of adjacent pixels, a vector of least differences is formed (**III**), which is calculated as the difference between the brightness levels of adjacent pixels and the central one, and its analysis is performed (**IV**). If the sum of the final differences takes less than the value when the center value is replaced (pink pixel), the center pixel is replaced with a new value (**V, a**), and if not, the previous pixel value will be the same (**V, b**).

The combination of adjacent pixels or the shape of the kernel may be different. Such settings may depend on the specific task and the noise parameters. In particular, the combination of the type "+" (Fig.5.a) - 4 neighboring pixels, the type "**X**" (Fig.5.b) - also 4 neighboring pixels, or their complete combination "+**X**" - 8 neighboring pixels can be considered. In addition, it is possible to modify this approach and consider other sizes and shapes of the studied kernel, in particular ("**3*3**"), ("**5*5**"), ("**7*7**") and others. with the appropriate selection of adjacent pixels (see Fig. 6)

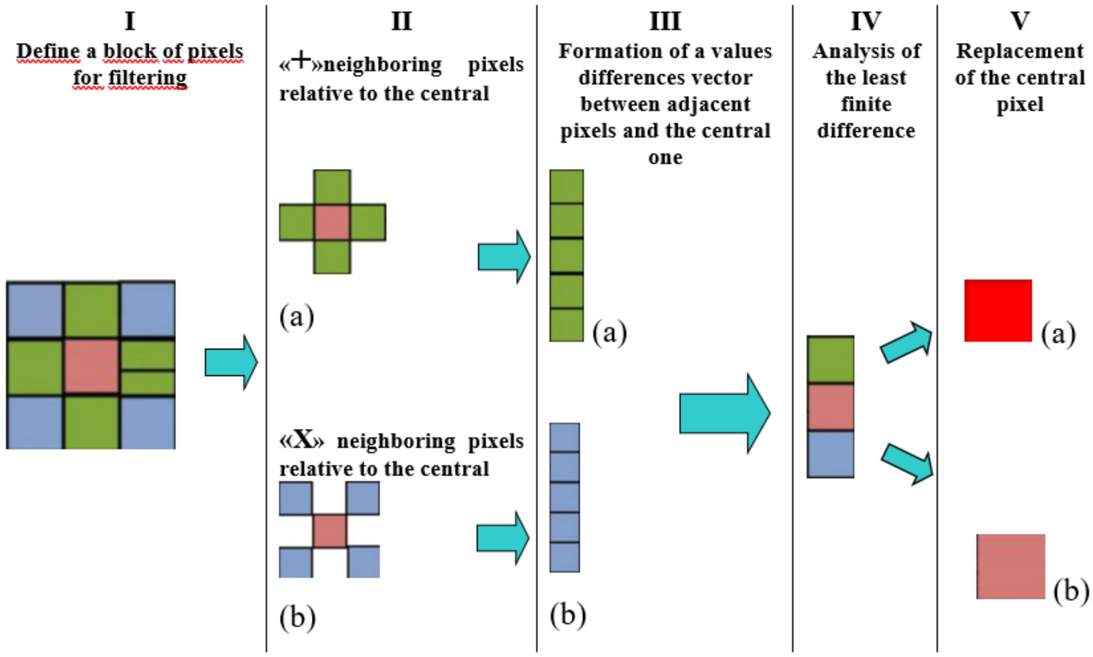


Fig.5. Algorithm for replacing the central pixel by the Method of Least Finite Difference (**MLFD**)

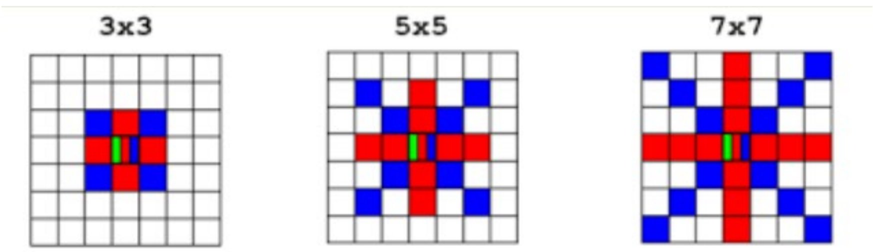


Fig.6. Shapes of the kernel and adjacent pixels such as "+", "X" and "+X", which are involved in data processing using the Method of Least Finite Differences.

Evaluation of the results effectiveness of image data recovery was performed according to the following criteria:

1) visual evaluation of the filtered image;

2) calculation of Standard Deviation (**SD**) for pixel intensity values:

$$\sigma = \sqrt{\frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij}^{f} - x_{ij}^{orig})^2}. \quad (4)$$

where x_{ij}^{f} - the brightness value of the filtered pixel, x_{ij}^{orig} - the brightness value of the pixel from the original image, *m, n* - the dimension of the image;

3) calculation of the Signal-to-Noise Ratio (***SNR***):

$$SNR = 10 \, log \frac{255^2}{\sigma^2}. \quad (5)$$

where σ^2 is the variance (power) of noise.

The web service for image filtering is proposed in the work, which is based on the use of a new *Method of Least Finite Difference*. This allows us to implement new filtering algorithms with better efficiency compared to known results and services.

IV. Results and Discussion

The application is implemented as a web service that allows us to select or upload images for filtering (see Figure 7, https://imagefiltr.herokuapp.com/index.html). The web service contains the following filtering tools, which are shown in Fig. 8 and in Table 1.

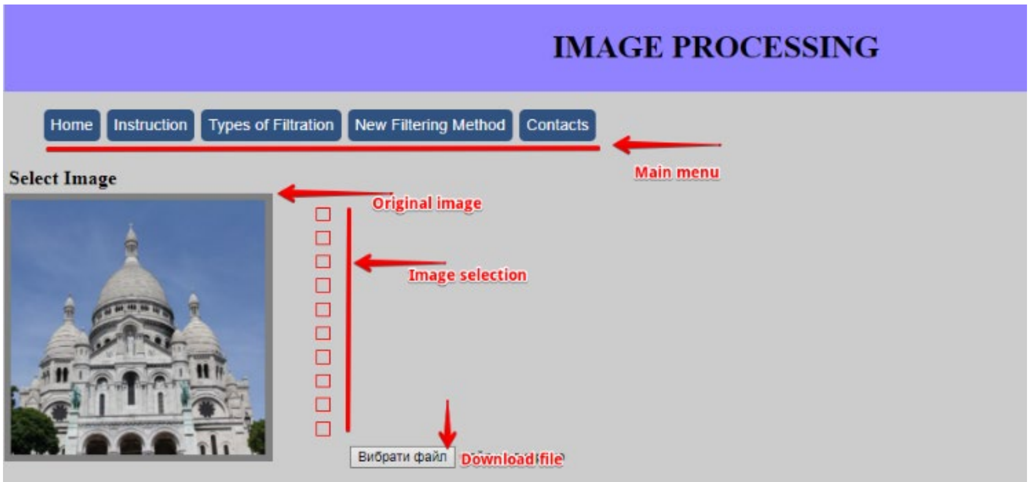


Fig.7. Home page of web application for image processing

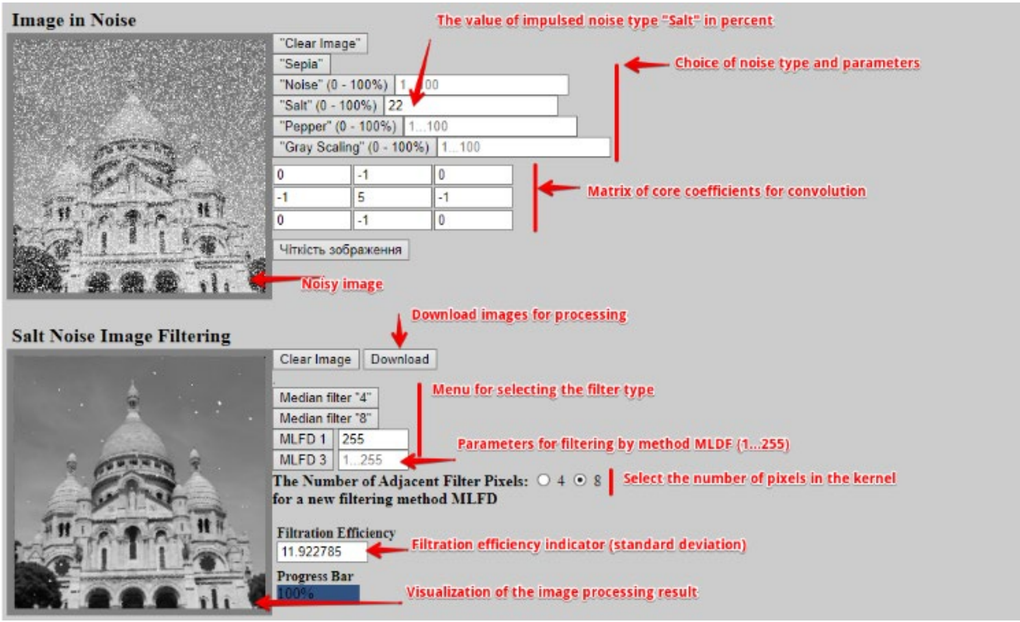


Fig.8. Image filtering tools and settings

TABLE 1 FILTERING PARAMETERS AND TYPE OF NOISE

#	Filter type	Filtering parameters	Type of Noise	Criterion of efficiency
1	Median -4	«+» - 4	• «Salt» • «Pepper» • «Grayscale»	Standard Deviation
2	Median -8	«+X» - 8		
3	MLFD -1	0…255 0…500		
4	MLFD -3	0…255 0…500		

The web service implements median filtering of the type "*Median-4*" and "*Median-8*" as ("+" - 4) and ("+ X" - 8). This type of filtering is used in many applications such as Adobe PhotoShop CS5. Median filtering is compared with the new *Method of Least Finite Differences*. The new algorithm can be used once (**MLFD-1**) or three times (**MLFD-3**). "Salt" type impulse noise (random emissions with a maximum brightness of 255), "Pepper" type impulse noise (random emissions with a minimum brightness of 0), "Grayscale" type impulse noise (random noise with a brightness in the range 0 ..255) are presented for modeling.

The results of image filtering in Salt noise (10%) for various filters are shown in Fig.9. The original non-noisy image (a) was used to estimate the Standard Deviation (**SD**) parameters for various filters.

The results of applying four filters are shown in fig. 9. As can be seen, the best result is obtained with single (e) and triple (f) use of the new **MLFD** method compared to classical median filtering (c) for 4 and (d) for 8 neighboring pixels.

As can be seen from the experiments, the Standard Deviation for different types of filters is different. The smallest **SD** value is obtained for **MLFD**. This result demonstrates the improved quality of image filtering compared to known approaches that are used in many applications. Experimental studies of image filtering by various methods were carried out with additive noise such as "Salt", "Pepper" and "Greyscale" from 20% to 50%. A significant advantage of the new **MLFD** data recovery method compared to median filtering is shown. The best image filtering result is obtained for highly noisy images (30% or more, Fig. 10).

Experiments were carried out for different types of noise - "**Grid**", "**Stripes**". Using well-known packages such as *Adobe PhotoShop* and *Gimp* does not give a quality result. The new **MLFD** method for 8 adjacent filter pixels gives the best result. In this case, the filtered image best matches the original image (see Fig.11).

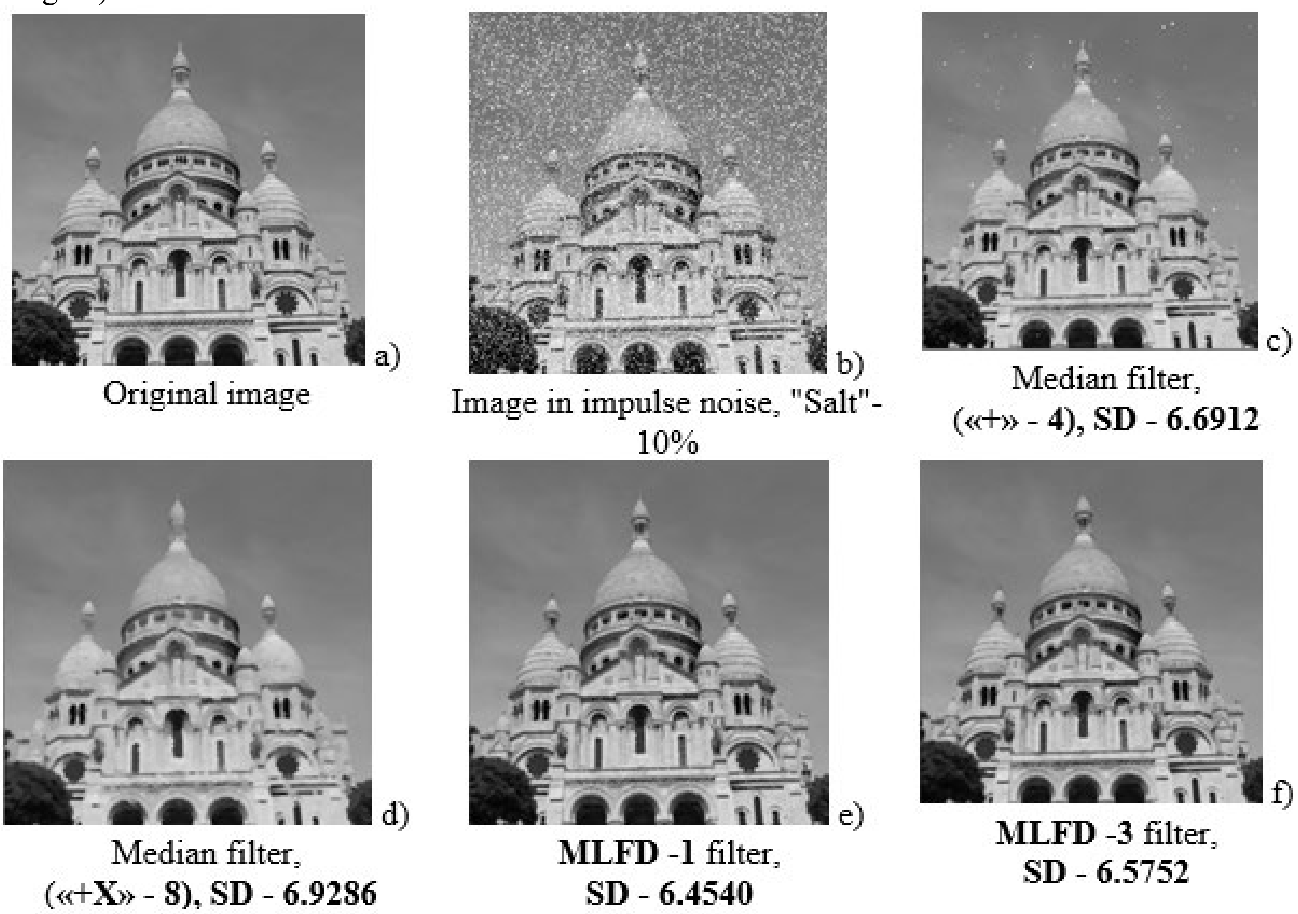


Fig.9. The result of images filtering in impulse noise "Salt" at 10%.

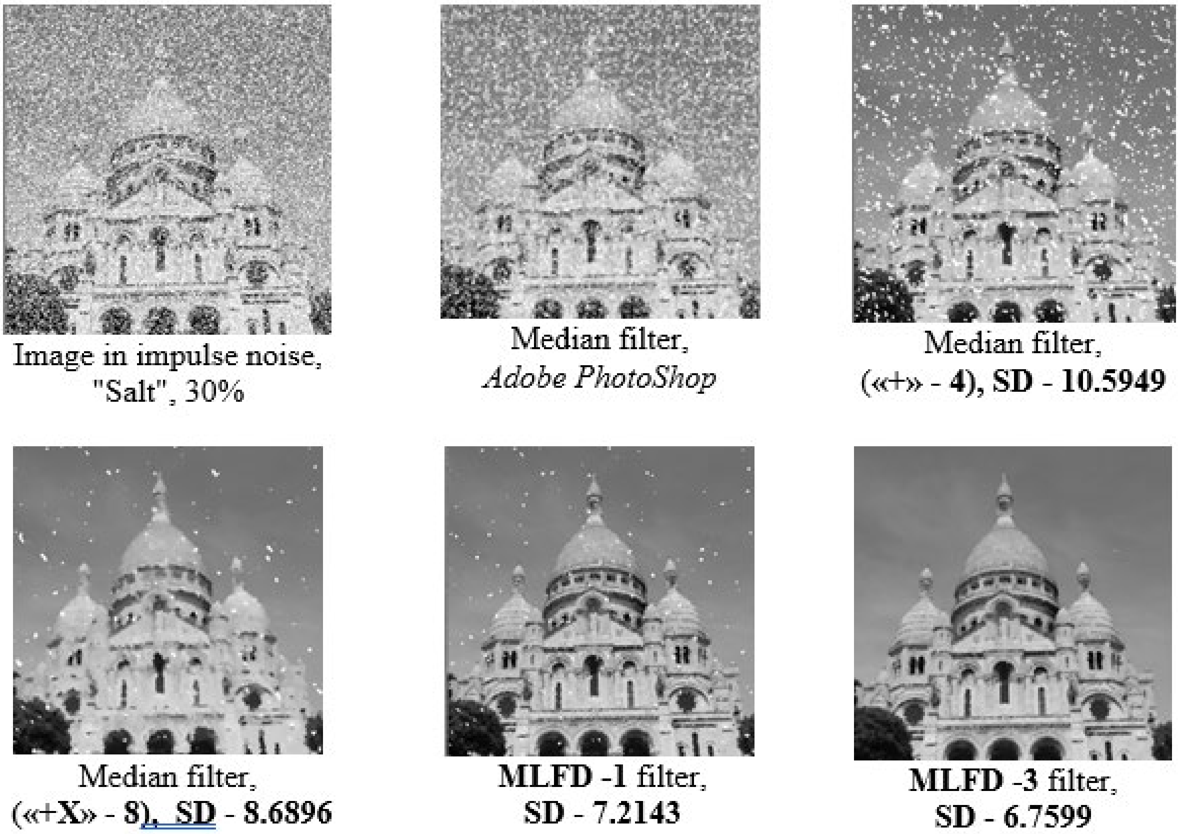


Fig.10. The result of images filtering in impulse noise "Salt" at 30%.

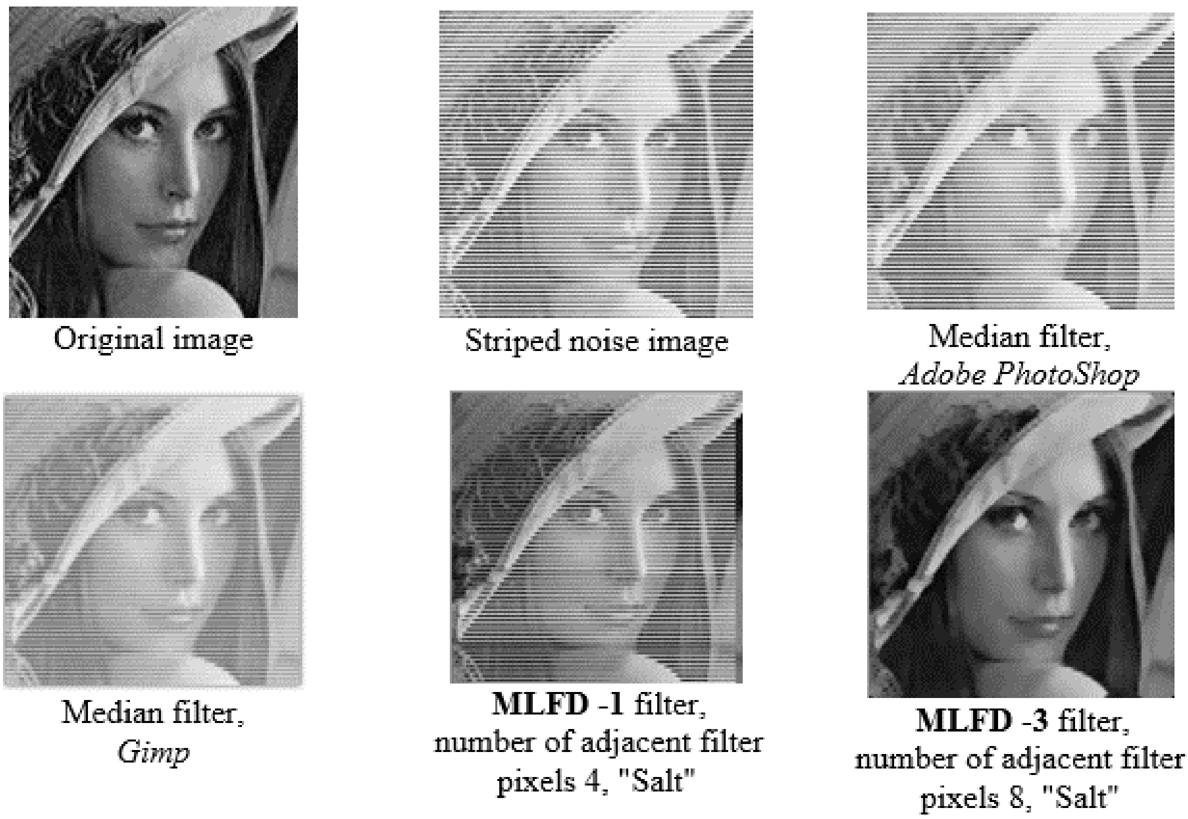


Fig.11. The result of images filtering in aditive noise as "Stripes" (image taken from the source http://decsai.ugr.es/cvg/CG/base.htm)

Analysis of filtration efficiency by different methods is shown in fig.12. The new **MLFD** method improves the filtering result, especially for very noisy images. Known filtering methods that are used in well-known graphics applications do not show good results.

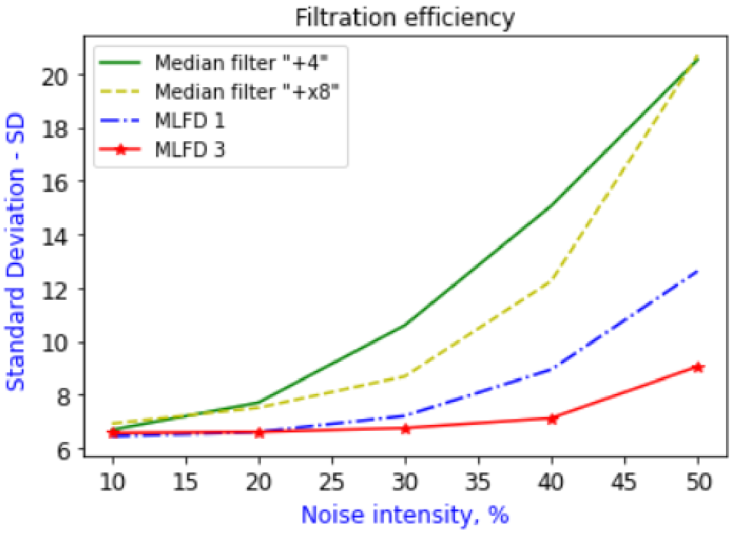


Fig.12. Characteristics of standard deviation parameter from noise intensity (%) by different methods

V. DEVELOPMENT OF A MODIFIED METHOD OF LEAST FINITE DIFFERENCES (MMLFD)

Let's consider the improvement of a new image filtering method (**MLDF**) under the influence of additive Gaussian noise. Experimental studies of image filtering have shown that additive Gaussian noise (Fig.13.b) cannot be filtered by classical methods using Adobe PhotoShop or built-in median filters (Fig.13.c, d). Even the proposed **MLFD** does not give the desired results, leaving white spots after its treatment (Fig.13.e).

However, taking into account the specifics of the noise in this case, it was decided to improve the filter using the following Modified Method of Least Finite Differences (**MMLFD**). The idea of the method is that we randomly select an image pixel and add to the brightness value of the selected pixel a certain random value rnd, which does not exceed the specified amplitude amplitude:

$$p\_new\ [i,j] = p\_new\ [i,j] + rnd, \tag{6}$$

where $p\_new\ [i,j]$ – [i,j]- an element of the pixel array of the image with changing noise;
rnd – random value that does not exceed the value level *amplitude.*

The change in the pixel value is accepted if the final difference for the [i, j] -th pixel of the original array p [i, j] is greater than the final difference for the [i, j] -th pixel of the array p_new [i, j]. The generalized algorithm and block diagram are presented in fig. 14, 15.

Note that changes in pixel brightness are possible in both directions. These changes will depend on the overall picture of the brightness distribution in the graphics array.

The image "evolves" to a smoother image by gradually changing the brightness of random points. The final brightness distribution is achieved after a large number of iterations of the program.

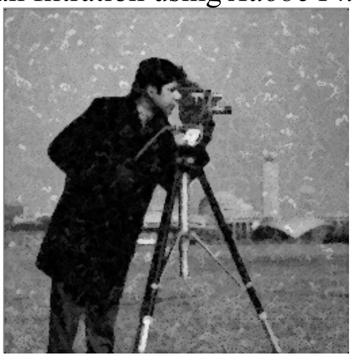
The results of the application of the modified **MMLFD** method are shown in Fig.13. The new method allows us to filter additive Gaussian noise, which cannot be removed by classical methods when using Adobe PhotoShop or built-in median filters

a) Original image

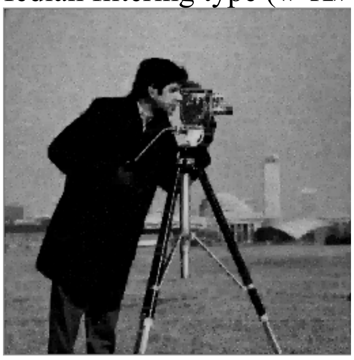
b) Image in additive Gaussian noise

c) Median filtration using *Adobe PhotoShop*

d) Median filtering type («+**X**» **- 8)**



e) Filtration method
MLFD **-3,** number of adjacent filtering pixels - 8,
"Salt"



f) Filtration method MMLFD**,**
the number of adjacent filter pixels - 8,
amplitude 2, filtration depth 10.

Fig.13. The result of image filtering in additive Gaussian noise (image taken from the source http://decsai.ugr.es/cvg/CG/base.htm)

Algorithm: Modified method of least finite differences (MMLFD)

For a randomly selected pixel array $p\_new\ [i,j]$:

1. Calculate the final difference for the [i, j]-th pixel of the array $p\,[i,j]$:

$$R_1 = |p[i,j] - p[i-1,j]| + |p[i,j] - p[i+1,j]| + |p[i,j] - p[i,j-1]| + \\ + |p[i,j] - p[i,j+1]| \ .$$

2. Generate a random variable

$$rnd = -random(amplituda).$$

3. If $p\_new\ [i,j] + rnd - p[i,j] > -amplituda$,

then $p\_new\ [i,j] = p\_new\ [i,j] + rnd.$

4. Calculate the final difference for the [i, j] -th pixel of the array $p\_new\ [i,j]$:

$$R_2 = |p\_new[i,j] - p\_new[i-1,j]| + |p\_new[i,j] - p\_new[i+1,j]| + |p\_new[i,j] - p\_new[i,j-1]| + |p\_new[i,j] - p\_new[i,j+1]| \,.$$

If $R_1 < R_2$, then $p\_new\ [i,j] = p\_new\ [i,j] - rnd$.

Fig.14. The generalized algorithm of the Modified Method of Least Finite Differences

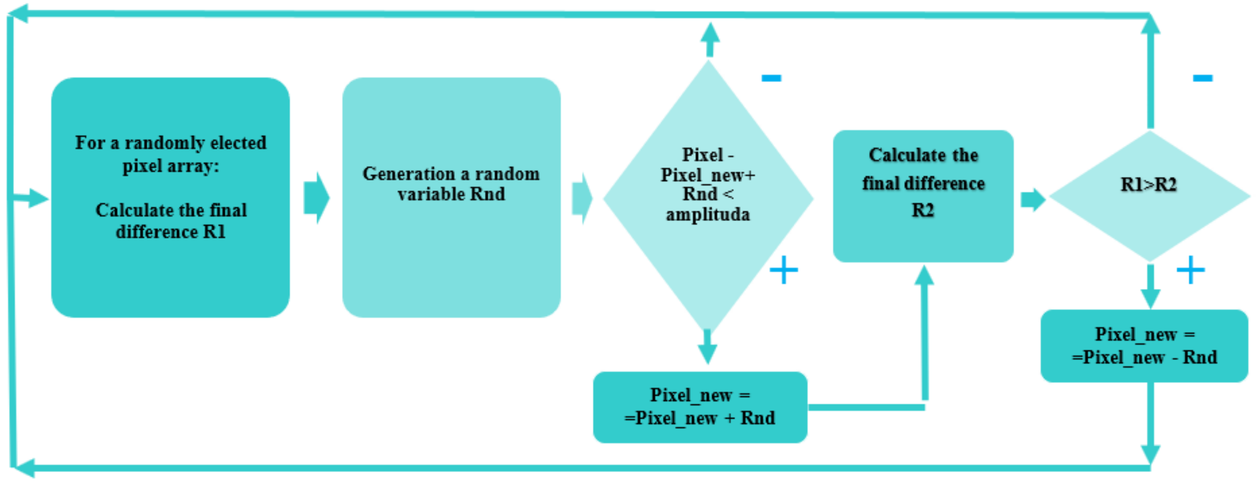


Fig.15. Block diagram of the Modified Method of Least Finite Differences

VI. Conclusions

An interactive web service for filtering images is developed in the article (https://imagefiltr.herokuapp.com/index.html). Data processing is based on the developed new method of least finite differences, which allows you to improve the data recovery procedure. The obtained results of the efficiency of image filtering were improved compared to known methods, for example, with a median filter. This type of filtering is a typical tool for many graphic applications.

Obtained analysis of the efficiency of image filtering in the conditions of additive impulse noise. The evaluation of data recovery quality was carried out visually and by calculating standard deviation parameters for various signal/noise ratios. The obtained results showed that the new approach to data processing is better than the use of well-known graphic editors.

In addition, the proposed solutions in the form of an Internet service are a good compromise between the simplicity of implementation and the quality of filtration compared to powerful, but complex and resource-intensive image processing systems.

Abbreviations

MLFD - Method of Least Finite Differences
SD - Standard Deviation
SNR **-** Signal-to-Noise Ratio
MMLFD - Modified Method of Least Finite Differences

REFERENCES

[1] Chris Solomon, Toby Breckon, “Fundamentals of Digital Image Processing”, Wiley Blackwell (2011)
[2] Bernd Jähne, “Digital Image Processing”, Springer (2005)
[3] M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis, and Machine Vision”, 4th Ed, Cengage Learning (2015)
[4] T.V.Torubka, V.Y.Puyda, "Method of removing additive noise in the images of aircraft", Bulletin of the National University "Lviv Polytechnic" - Computer Systems and Networks, №773, pp.131-136 (2013)
[5] J.J.Bilynsky, P.M.Ratushny, "Methods and system for processing low-contrast images for evaluating the microcapillaries of human limbs", Vinnytsia, VNTU (2012)
[6] R.N.Kvetny, I.V.Bogach, O.R. Boyko and others, "Computer modeling of systems and processes. Calculation methods", part 2, Vinnytsia, VNTU (2013)
[7] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", 4th Edition, Pearson, NY (2018)
[8] Van Trees, Harry L., and Kristine L. Bell, "Bayesian Bounds for Parameter Estimation and Nonlinear Filtering", Tracking, Piscataway, NJ: IEEE (2007)
[9] Bekir Aksoy, Osamah Khaled Musleh Salman, “A New Hybrid Filter Approach for Image Processing”, *Sakarya University Journal of computer and Information Sciences* vol. 3, no. 3, pp.334-342 (2020)
[10] X.Deng, Y.Ma and M.Dong, “A new adaptive filtering method for removing salt and pepper noise based on multilayered PCNN,” *Pattern Recognit. Lett*., vol.79, pp. 8-17, 2016
[11] Wilhelm Burger, Mark.J.Burge, "Principles of Digital Image Processing. Advanced Methods", Springer-Verlag London (2013)
[12] Vibhor Sharma, Dheresh Soni, Deepak Srivastava, "Filtration Based Noise Reduction Technique in an Image", 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 18-19 April (2019)
[13] S.K. Sathua, A. Dash, A. Behera, “Removal of salt and pepper noise from gray-scale and color images: an adaptive approach”, *Int. J. Comp. Sci. Technol. (IJCST),* 5 (1), pp. 117-126 (2017)
[14] Chih-Yu Hsu, Ta-Shan Tsui, Shyr-Shen Yu and Kuo-Kun Tseng, "Salt and Pepper Noise Reduction by Cellular Automata", *International Journal of Applied Science and Engineering*, vol. 9, no. 3, pp. 143-160 (2011)
[15] Y.Zhang, Y.Liu, X.Li and C. Zhang, "Salt and pepper noise removal in surveillance video based on low-rank matrix recovery", *Computational Visual Media*, vol. 1, no. 1, pp. 59-68 (2015)
[16] Z. Gao, “Ana adaptive median filtering of salt and pepper noise based on local pixel distribution”, *Int. Conf. Transport. Logistics, Inform. Commun., Smart City (TLICSC)*, Adv. Intell. Syst. Res., 161, pp. 473-483 (2018)
[17] L.Jin, H.Liu, W.Zhang, and E.Song, "Video oriented filter for impulse noise reduction", *Journal of Visual Communication and Image Representation*, vol. 55, pp. 1-11 (2018)
[18] Anwar Shah Javed, Iqbal Bangash, Abdul Waheed, Khan ImranAhmed, Abdullah Khan, Asfandyar Khan, Arshad Khan, “Comparative analysis of median filter and its variants for removal of impulse noise from gray scale images”, Journal of King Saud University - Computer and Information Sciences, vol. 34, pp. 505-519 (2022)
[19] V.Palahin, O.Ivchenko, D.Vedernikov, "Statistical estimation of parameters of non-Gaussian correlated random processes", Cherkasy (2018)
[20] V.Palahin, J.Juhár, "Joint Signal parameters estimation in non-Gaussian noise by the method of polynomial maximization", *Journal of Electrical Engineering,* vol. 67, no.3, pp.217–221 (2016)

FIGURE LEGENDS

Fig.1. a) Digital representation of the image in the form of a two-dimensional Data Set; b) Digital representation of a color image in RGB format

Fig.2. Two-dimensional digital image model (a) and its transformation into a one-dimensional array for computer data processing (b).

Fig.3. Results of filtering of a noisy image (noise type "Grayscale") by the median filter in the Adobe PhotoShop package (image taken from the source https://www.census.gov/history/www/homepage_archive/2019/march_2019.html)

Fig.4. Algorithm for replacing the central pixel of two-dimensional digital image by the Method of Least Finite Difference (MLFD)

Fig.5. Algorithm for replacing the central pixel by the Method of Least Finite Difference (MLFD)

Fig.6. Shapes of the kernel and adjacent pixels such as "+", "X" and "+X", which are involved in data processing using the method of least finite differences.

Fig.7. Home page of web application for image processing

Fig.8. Image filtering tools and settings

Fig.9. The result of images filtering in impulse noise "Salt" at 10%.

Fig.10. The result of images filtering in impulse noise "Salt" at 30%.

Fig.11. The result of images filtering in dditive noise as "Stripes" (image taken from the source http://decsai.ugr.es/cvg/CG/base.htm)

Fig.12. Characteristics of standard deviation parameter from noise intensity (%) by different methods

Fig.13. The result of image filtering in additive Gaussian noise (image taken from the source http://decsai.ugr.es/cvg/CG/base.htm)

Fig.14. The generalized algorithm of the Modified Method of Least Finite Differences

Fig.15. Block diagram of the Modified Method of Least Finite Differences