

# Automated recursive separable algorithm for medical image processing

A.A. Filimotseva<sup>1</sup>, A.V. Kamenskiy<sup>1</sup>, A.S. Zahlebin<sup>1</sup>

<sup>1</sup> FSBEU HE «Tomsk State University of Control Systems and Radioelectronics»,  
634050, Russia, Tomsk, Lenin avenue, building 40

## Abstract

In the modern world, there is an increasing demand for automated information processing systems that reduce the time spent on monotonous human work, increase the efficiency of algorithms and the time it takes to solve assigned tasks. The paper presents the development of an automated recursive-separable digital filter for processing digital medical images. The feature of the automated algorithm is its speed; due to its internal structure, the recursive-separable Laplacian Filter “double pyramid” performs separate processing by row and column of the image, using recirculators. The recirculator itself also affects the speed due to the use of recursion properties. Automation of the algorithm will increase the efficiency of digital image processing, due to automatic enumeration of filter coefficients, which will improve the quality of the original information almost without human intervention. The developed algorithm itself carries out the process of image quality assessment and repeats the digital processing procedure until it reaches the required values of one of the parameters: mean square error (MSE), peak signal-to-noise ratio (PSNR) and structural similarity index of image (SSIM). To test the developed automated algorithm, the software "ALF: Automated LDP Filter" was developed and implemented using QtDesigner and PySide6. The software module consists of three data entry areas, such as the mask size  $h$ , the coefficient for lifting  $A_1$  and the coefficient for increasing the central element  $A_2$ , two areas for displaying the image, the original and processed, seven buttons, such as input, output and saving the image, checking the image quality condition, three buttons for calculating the parameters MSE, PSNR and SSIM. As part of the testing, a study was carried out on the performance of the automated recursive separable digital filter using the example of endoscopic images obtained from a robotic surgical complex, which showed the effectiveness of its use, since it was possible to achieve the specified parameter values on the output image.

**Keywords:** automated algorithm, recursive separable filter, peak signal-to-noise ratio, root mean square error, structure similarity.

**Citation:** Filimotseva AA, Kamenskiy AV, Zahlebin AS. Automated recursive separable algorithm for medical image processing. *Computer Optics* 2025; 49(6): 1023-1029. DOI: 10.18287/2412-6179-CO-1597.

## Introduction

Digital image processing is one of the important areas of development of computer technologies used in modern telemedicine. Particular attention is paid to improving the quality of images, restoring damaged images. Image processing technology has a number of advantages, such as flexibility, adaptability, data storage and communicativeness - it has many rules for synchronous image processing. Two-dimensional and three-dimensional images can be processed in several dimensions (image processing methods were founded in the 1960s), these methods have been used in various fields, such as medicine, space, clinical research, art, etc. [1].

Medical imaging and diagnostic techniques have developed rapidly in recent decades and have become a very important part of disease diagnosis. Digital medical images (DMI) play an important role in transmitting information about the heart, brain, nerves, etc. to obtain an internal view of the human body. Numerous mathematical applications can be applied to medical images to conclude whether healthy tissue is infected or not. However, the loss of any specific area during medical imaging can lead to

disaster such as death. The main challenge in the process of medical imaging is to obtain an image without losing any significant information [2].

It should be noted that in a number of areas of DMI research, the conducted scientific research on the automation of image processing and analysis yield worse results compared to manual image analysis by specialists [3]. Existing software products for DMI processing yield good results, but they are usually foreign products with closed specialized software built into the analysis device and supplied with it. This foreign equipment is expensive. The development of domestic software for analyzing and improving the quality of visual information is an urgent task, the solution of which will allow processing large volumes of information obtained in the field of telemedicine [4].

In the modern world, many scientists are developing automated algorithms for processing and analyzing digital images that can increase human productivity. The work [5] presents Mitometer, an algorithm for fast, unbiased, and automated segmentation and tracking of mitochondria in live-cell two-dimensional and three-dimensional time-lapse images.

A number of modern studies such as an artificial intelligence-based algorithm for automated diabetic

retinopathy screening [6], methods for visual object recognition for automated digital modeling of existing buildings [7], a study on the impact of a real-time automatic quality control system on the detection of colorectal polyps and adenomas [8], an approach and dataset for describing document feature points using a small convolutional neural network [9], a method for segmenting breast cancer boundaries and pectoral muscles in mammograms using threshold-based and fully automated trainable approaches [10], a deep learning method for automatic medical image segmentation requiring minimal annotations [11], and an artificial bee colony algorithm and its application in medical image processing [12] present various methods for automating DMI processing algorithms and their application for various telemedicine tasks. In addition to the automation of image processing algorithms, algorithms for automated assessment of the quality of processed images are also being actively developed; for example, the work [13] presents an automated visual assessment.

The aim of the research work is to develop and implement a software implementation of an automated recursive-separable algorithm for processing medical images. Tasks of the work:

1. Develop an automated algorithm;
2. Implement an application for using the developed algorithm;
3. Perform testing of the implemented algorithm.

### 1. Digital filter

The automated algorithm is implemented on the basis of a two-dimensional digital Laplacian Filter “double pyramid” (LDP) [14 – 15].

The two-dimensional digital LDP filter in the implementation of the size 7×7 (Fig. 1) is a filter in the center of which there is a positive aperture of 3×3 elements.

-1	-2	-3	-3	-3	-2	-1
-2	-4	-6	-6	-6	-4	-2
-3	-6	5	19	5	-6	-3
-3	-6	19	48	19	-6	-3
-3	-6	5	19	5	-6	-3
-2	-4	-6	-6	-6	-4	-2
-1	-2	-3	-3	-3	-2	-1

Fig. 1. Three-dimensional view of the LDP filter aperture

The structural diagram demonstrating the operating principle of the filter used is shown in Fig. 2.

The filter operates according to the following principle. In the second branch, a mask is formed, with a given dimension of the corresponding coefficients of the frame and line recirculators. Recirculators use the recursion property to search for each result obtained after the first sample is formed, which allows to reduce the image processing time. It is the main one and is fed to the adder with a negative

sign. In the first branch, a second mask is formed with the same dimension as the internal mask from the lower branch. In this case, it has a certain coefficient  $X$ , with which the convolution by the line and frame recirculators will occur. It is fed to the adder with a positive sign.

In the third branch, an additional mask is formed, which performs a similar function as the upper branch of the original filter. This branch has a coefficient  $A_2$ , which participates in increasing the central element. This branch is also fed to the adder with a positive sign. As a result, the matrix of the digital filter is formed.

### 2. Automated processing algorithm

The filter algorithm consists of the following steps:

1. Entering basic parameters. It is necessary to enter the input data:  $h$  – mask size value (odd);  $X$  – image to be processed;  $A_1$  – central aperture lift coefficient;  $A_2$  – central element magnification coefficient.

2. Determining the recirculator values for convolution of the processed image into the original matrix.

3. Calculating the main filter matrix, which consists of calculating the coefficients of two frame and two row recirculators.

4. Calculating the shift coefficients of the additional (internal) mask and calculating the size of this internal mask.

5. Calculating the coefficients of the frame and row recirculators for the second (additional) matrix.

6. Calculating the coefficient  $A_1$ , which is the main coefficient for convolution.

7. Convolution of the original image with these recirculators, taking into account the coefficient  $A_1$ , the central aperture lift regulator.

8. Formation of an additional matrix.

9. Surrounding the resulting matrix with zeros to the size of the main one.

10. Formation of a third matrix to implement the lifting of the central element.

11. Final calculations.

For the software implementation of the constructed automated recursive-separable algorithm for processing medical images, the Python programming language was chosen, where the algorithm of all processes occurring in the filter is written in the form of code. Fig. 3 shows a block diagram of the automated algorithm for processing digital medical images.

The initial digital image is fed to the input of the algorithm, the algorithm takes one color channel of the image and starts working. The mask size is entered from the keyboard, the entered value must be odd. To enter the parameters  $A_1$  and  $A_2$ , a nested loop was created that automatically iterates over the values in the range specified by the user. After entering the initial parameters, the image is processed by the algorithms described in the previous paragraph. When the image processing is finished, the image quality assessments such as the mean square error MSE, the structural similarity index SSIM and the peak signal-to-noise ratio PSNR are calculated.

MSE is the most common metric, and the closer its value to zero, the better. It is also called the deviation of the evaluator, which is the original image. The error is the difference between the estimated result and the evaluator [16]. PSNR is used to quantitatively evaluate the effectiveness of different

noise reduction methods. It is a measure of the peak error, used as a measure of reconstruction of lossy compression codes. The signal is the original image, and the noise is the introduced compression. The higher the value of the ratio, the more the reconstructed image is similar to the original [17].

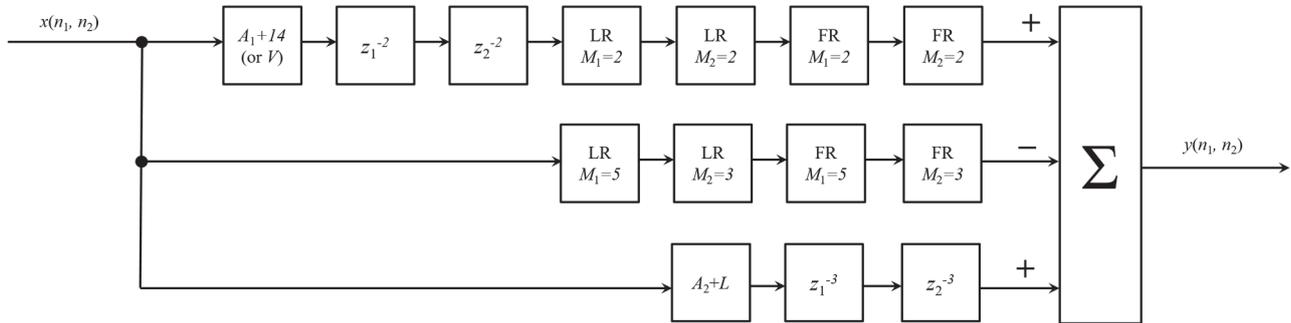


Fig. 2. Structural diagram of the LDP filter:

$x(n_1, n_2)$  – input data;  $14 (V)$  – coefficient for additional mask convolution;  $z_1^{-2}$  and  $z_2^{-2}$  – delay by two elements; LR – line recirculator, FR – frame recirculator;  $M_1$  – line direction,  $M_2$  – frame direction;  $z_1^{-3}$  – delay by three elements;  $A_1$  – coefficient of rise of the central aperture;  $A_2$  – coefficient of increase of the central element;  $y(n_1, n_2)$  – output data;  $L$  – coefficient of the complementary matrix

The structural similarity index of image (SSIM) is a relatively new method for measuring image similarity, used in areas ranging from image restoration/compression to object recognition [18].

Image processing ends with the formation of the final matrix with normalization of the image brightness, this matrix is larger in size than the matrix of the original image, therefore, to calculate image quality estimates, it is first necessary to crop the matrix to the desired size, thus obtaining a mask. To calculate the peak signal-to-noise ratio, it is necessary to know the root-mean-square error, therefore this parameter is initially calculated. After the MSE value is known, the PSNR is calculated.

The PSNR and MSE parameters provide a general integral assessment of image quality. It is believed that good image quality is ensured at  $\text{PSNR} > 40$  dB and  $\text{SSIM} > 0.98$  [19]. Accordingly, after calculating the image quality estimates, the obtained values are compared with the values at which the image is considered to be of good quality. If the value of the processed image satisfies the values of a good quality image, the image is saved with the parameters  $A_1$  and  $A_2$  indicated, the values of the processed image are output, and a message is output to the console stating that the image has been saved.

The original image ( $x(n_1, n_2)$ ) is processed by the line (LR) and frame (FR) recirculators using the values obtained during the calculations. The size of the first line and frame recirculator is denoted by  $M_1$ , and the size of the second line and frame recirculator is denoted by  $M_2$ . The calculation process is performed as follows: in the calculation algorithm, the minimum value of the recirculator is specified, for the LDP filter it is equal to 3 ( $t_1 = 3$ ), and then the difference between the user-entered value of the filter aperture size ( $h$ ) and the coefficient  $t_1$  is found. The result is divided by 2 ( $k = (h - t_1) / 2$ ). Then the obtained value is added to the coefficient  $t_1$ , thus the value of the coefficients of the first recirculators is calculated ( $M_1 = t_1 + k$ ). The value

of the coefficient of the second recirculators ( $M_2$ ) is calculated based on  $M_1$ , and the coefficient 2 is subtracted from it ( $M_2 = M_1 - 2$ ).

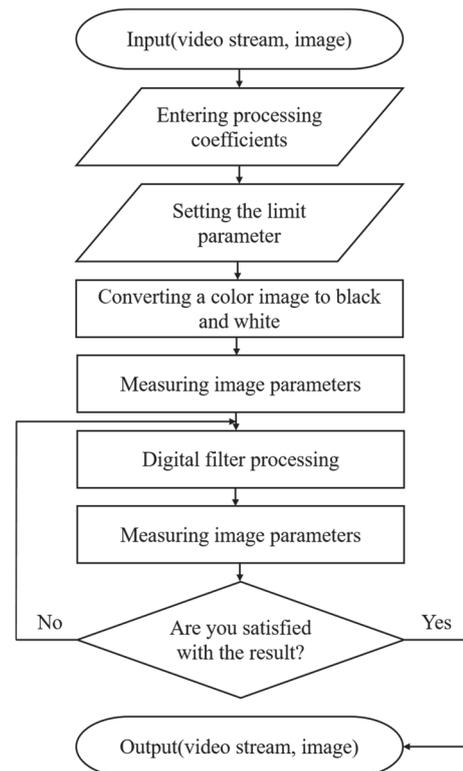


Fig. 3. Block diagram of the automated recursive-separable algorithm

Next, the algorithm is launched, constructing a matrix of values, after processing the original image by the recirculator LR ( $M_1$ ). To begin with, the recirculator  $Y_1$ , itself is designated, it takes the size of one row and the number of columns equal to  $M_1$ , which was calculated earlier. Then the algorithm determines the values of the number of columns and rows in the original image  $X$ , as well as the

number of columns in the recirculator, and writes them as  $i_1, j_1$  and  $i_2$ .

Next, the algorithm builds the matrix  $X_1$  from the original image and supplements it with zeros on the right to the final size. Then it determines the coefficients  $k$  and  $m$ . They are necessary for building the basis for recording the future result, the zero matrix  $y_1$ .

When all the initial data is ready, the code starts a two-level cycle. First, it selects a row of the  $y_1$ , matrix, then a column, and fills the column with the necessary values. So, in the first column there will be a value equal to the value of the same element in the  $X_1$  matrix. In the second column, the value of the element in the  $X_1$  matrix will also be taken into account, and in any case, whether there is a certain number or zero, the algorithm will accept it and add the previous element of the  $y_1$  matrix to it. Such operations will be performed until it passes the middle and reaches a certain element, when the value begins to change. Then it will set values with the elements equal to the value of the  $X_1$  element plus the values of the previous element and minus the value of the  $X_1$  element, which is in the first column.

The result is a matrix ( $y_1$ ), where the number of rows will be equal to  $j_1 \times i_1 + i_2 - 1$ .

The same thing happens with the second row recirculator. The only difference is that instead of the original image, the final matrix ( $y_1$ ), obtained after the first recirculator, is taken (Fig. 4).

```

Y2=np.ones((1,t22)) #
i1=np.size(y1, axis=1)
j1=np.size(y1, axis=0)
i2=np.size(Y2, axis=1)
nulik12 = np.zeros((j1, i2-1))
X2 = np.hstack((y1, nulik12))
k=j1
m=i1+(i2-1)
y2=np.zeros((k,m))
for k in range(0, j1):
    for m in range(0, i1+(i2-1)):
        if m==0:
            y2[k,m]=X2[k,m]
        if m>0 and m<i2+1:
            y2[k,m]=X2[k,m]+y2[k,m-1]
        if m-i2>=0:
            y2[k,m]=X2[k,m]-X2[k,m-i2]+y2[k,m-1]

```

Fig. 4. Fragment of the code for the operation of the second line recirculator

Similar work is performed by frame recirculators (FR). Similar to the second row recirculator, the matrix obtained as a result of work at the previous stage ( $y_2$ ) is taken as a basis. The matrix is surrounded by zeros from above and below to the final size, and in the cycle itself the work is performed by columns, and not by rows, as before. That is, first a column is selected, filled with elements, and only after that it moves on to another column.

The second frame recirculator operates in the same way. However, this time the final matrix obtained after the first frame recirculator ( $y_3$ ), respectively, and after two row recirculators, i.e. the result of all previous convolutions, is taken as the initial matrix and the basis for convolution. As

a result, the matrix of the result of all operations in the filter branch ( $y_4$ ) is obtained.

The process of calculating the central mask shift factor is based on the use of the  $h$  value, i.e. the mask aperture size.

The aperture size value is increased by one ( $h + 1$ ), thereby bringing it to parity. Then a check is made, i.e. in this case, the value is filtered for the possibility of dividing by 4. If the value is divisible by 4, it is divided, and as a result, the shift factor is obtained for the elements  $z_1^{-n}$  and  $z_2^{-n}$ .

The process of calculating the recirculator coefficient for the central mask of a filter with an aperture of  $7 \times 7$  elements is based on the use of the  $h$  value. The aperture size value is increased by one ( $h + 1$ ), thereby bringing it to parity. Then a check is made, that is, in this case, the value is filtered for the possibility of dividing by 4. If the value is divisible by 4, then it is divided, and as a result, the coefficient  $M_1 = M_2$ , is obtained, for LR and FR in the first branch of the filter.

Next, the convolution coefficient  $V$  is calculated (in the special case of  $7 \times 7$  elements, it is equal to 14). It should be calculated based on the value required to achieve the goal at which the sum of the internal and external elements of the final matrix will be equal to zero. A fragment with the calculation of the convolution coefficient is shown in the Fig. 5.

```

S0=(t11*t22)*(t11*t22)
P=(t33*t33)*(t33*t33)
A=S0//P
A=round(A)

```

Fig. 5. Code snippet with calculation of convolution coefficient

The coefficient for convolution is calculated according to the following principle. First, the sum of the elements of the main matrix is found. It is the multiplication of the coefficients of the recirculators. Thus, the summation is not of the matrix that was obtained during processing of the received image, but of the matrix calculation when feeding a unit as the original image.

Since the final summation must be carried out with matrices of the same dimensions, the resulting matrix is surrounded by zeros up to the size of the main one.

The resulting matrices, when summed, will not give a sum of elements equal to zero, therefore a third matrix is formed.

The formation of this matrix occurs according to the following principle. First, the coefficient  $L$  is found. It is based on finding the missing value, therefore, first, the sum of the elements of the additional matrix is calculated. The found coefficient  $V$  is also taken into account, it is multiplied by the sum of the elements when convolving the unit with an additional mask, that is, as a result, the sum of the matrix elements is obtained when convolving the coefficient  $V$  instead of the unit.

Then the missing value is found. For this purpose, the code refers to the value of the sum of the elements of the first matrix when convolving one. The found sum of the elements of the additional matrix is subtracted from it, and as a result, the missing value is obtained. It is the coefficient  $L$ . To begin with, the original image is multiplied by

the sum  $A_2+V$ , then the process of adjusting the obtained element to the size of the main matrix occurs. A zero matrix is constructed with the same size as the main one, which will serve as the basis. And then the shift coefficient is determined by adding one to the second value of the re-circulators of the main matrix. And ultimately, a zero matrix is formed, in the middle of which the central element or matrix is placed.

The final stage of the filter operation is the summation of all the obtained masks. However, when the central matrix and the central element are raised, the image brightness values are violated; in order to normalize it, the final sum of the matrices is divided by the sum of the elements that were added using  $A_1$  and  $A_2$ . The resulting matrix will be the result of the filter operation [20].

### 3. Implementation of an application for using an automated algorithm

To use the automated recursive separable algorithm for processing medical images, a software module [21] was implemented using QtDesigner and PySide6.

The software module consists of three areas for inputting data, such as the mask size  $h$ , the coefficient for lifting  $A_1$  and the coefficient for increasing the central element  $A_2$ , two areas for displaying the image, the original and processed, seven buttons, such as input, output and saving the image, checking the image quality condition, three buttons for calculating the MSE, PSNR and SSIM parameters.

To operate the buttons, the previously developed and described algorithm was divided into functions. A certain function is responsible for the operation of a certain button.

A separate function was written for the windows that appear on the user's screen, which report the successful loading, output and saving of the image, display a window with the results of calculating the MSE, PSNR and SSIM parameters, and also display the result of checking the image quality conditions (Fig. 6).

The function for the "Insert image" (кнопка «Вставить изображение») button specifies the dimensions of the window into which the image will be inserted, as well as messages that will appear in the event of a successful or unsuccessful attempt to load the image.

The function for the "Calculate MSE" (кнопка «Рассчитать MSE») button specifies the lines for displaying messages on the user's screen, and also specifies all the necessary parameters for calculating the value.

The function for the "Calculate PSNR" button (кнопка «Рассчитать PSNR») specifies the lines with messages for windows with notifications and specifies the necessary parameters. To calculate PSNR, the necessary parameter is MSE; if it is not calculated, an error will occur.

In the function for the "Calculate SSIM" button (кнопка «Рассчитать SSIM»), the required parameter for calculation is the resulting matrix obtained as a result of the algorithm. If the matrix is not calculated, an error occurs.

The function for the "Output image" button (кнопка «Вывести изображение») is similar to the image input

function. It also specifies the size of the window in which the processed image will be output.

The function for the "Condition check" button (кнопка «Проверка условия») requires the previously calculated PSNR and SSIM values, otherwise the user will see an error message.

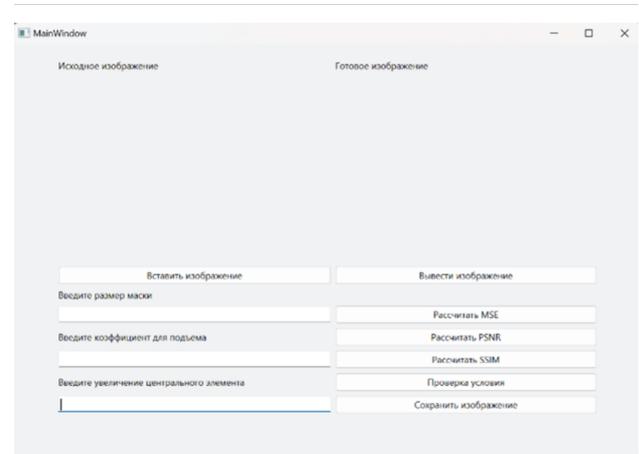


Fig. 6. Appearance of the software module

The function for the "Save image" button (кнопка «Сохранить изображение») allows you to save an image with a unique number using the coefficient values entered at the beginning of the algorithm.

### 4. Testing the automated algorithm

The performance test of the developed and implemented automated algorithm will consist of assessing its impact on a number of parameters.

As part of the experimental study, the desired parameters of the PSNR and SSIM metrics will be set, which the algorithm must achieve during its operation, and a fixed value of the filtering coefficient  $A_1 = 1$ . As a result of the algorithm's operation, the operating time will be obtained, during which it was possible to achieve the specified values of the PSNR and SSIM metrics, as well as the value of the additional filtering coefficient  $A_2$ .

The study will be conducted on a computing platform with the following characteristics: processor – Intel(R) Pentium(R) CPU 5405U @ 2.30GHz; RAM – 4 GB; system type – 64-bit.

In the original image, the values of the parameters were as follows: PSNR = 13.994; SSIM = 0.452.

Fig. 7 Shows a test image that will be used in the experiment.

The parameter values were obtained using the equations below. All equations were written in the code and calculated sequentially.

The mean square error is measured using Eq. (1):

$$MSE = \frac{1}{M \times N} \sum_{i,j} [I_1(i,j) - I_2(i,j)]^2, \quad (1)$$

where  $M \times N$  – image size,  $I_1(i,j)$  – original image,  $I_2(i,j)$  – processed image.



Fig. 7. Test image

The peak signal-to-noise ratio is calculated using the following Eq. (2):

$$PSNR = 10 \log \left[ \frac{I^2}{MSE} \right], \quad (2)$$

where  $I$  – the maximum value of an image pixel. If the input image has an eight-bit integer data type, this value will be 255.

The most common equation for measuring SSIM between  $x$  and  $y$  vectors is:

$$SSIM(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma, \quad (3)$$

where  $l$  – comparison of luminance signals (Eq. 4),  $c$  – comparison of contrast signals (Eq. 5),  $s$  – structural correlation of signals (Eq. 6).

$$l(x, y) = (2\mu_x\mu_y + c_1) / (\mu_x^2 + \mu_y^2 + c_1), \quad (4)$$

$$c(x, y) = (2\sigma_x\sigma_y + c_2) / (\sigma_x^2 + \sigma_y^2 + c_2), \quad (5)$$

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_{xy} + c_3}, \quad (6)$$

where  $\mu_x, \mu_y$  – average values of  $x$  and  $y$ ,  $\sigma_x^2, \sigma_y^2$  – variance of  $x$  and  $y$ ,  $c_1, c_2, c_3$  – constant values for cases where  $\sigma$  and  $\mu$  are too small.

Substituting all the data we obtain the final result.

The structural similarity index is calculated using Eq. (7):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \quad (7)$$

In the experiment, different levels of PSNR and SSIM metrics were set, with a fixed value of the filter coefficient  $A_1 = 1$ . After that, the automated recursive separable algorithm was launched, within the framework of which it performed a step-by-step enumeration of the filter coefficient  $A_2$  and an assessment of the specified parameters. In cases where it was not possible to achieve the specified values of the PSNR and SSIM metrics, repeated processing was performed, but with a different value of the filter coefficient  $A_2$  until the desired result was achieved. As a result, information was also issued on the time spent on achieving the required result, these results are presented in Tab. 1.

Analyzing the obtained data, it can be seen that the higher the required values of the PSNR and SSIM

parameters, the more time the algorithm needs to find the optimal coefficient. These results are logical and consistent.

Tab. 1. Test results of the automated recursive-separable algorithm

PSNR, dB	SSIM	Time hh:mm:ss	Coef. $A_2$
15	0.35	00:00:26	$A_2 = 1$
20	0.45	00:01:18	$A_2 = 5$
25	0.55	00:04:22	$A_2 = 19$
30	0.65	00:09:58	$A_2 = 47$
35	0.75	00:20:27	$A_2 = 95$
38	0.85	00:30:46	$A_2 = 141$
40	0.98	01:58:55	$A_2 = 571$

By analyzing the works [14 – 15] it can be concluded that the algorithm's performance has improved significantly, with the LDP filter showing a 6-fold increase in performance compared to the classical two-dimensional convolution.

In addition to the quantitative and time assessment, a visual assessment of the image quality was also carried out. Each image was saved when the parameters specified in the table were reached and did not acquire third-party artifacts. Fig. 8 shows a test image after processing by an automated algorithm.



Fig. 8. The result of the algorithm operation with  $PSNR \geq 35$  and  $SSIM \geq 0.75$

### Conclusion

The developed automated algorithm of the LDP filter operation allows achieving the required results without user participation, in particular, increasing such image quality assessments as PSNR and SSIM. In the process of automating the algorithm, the ability to enumerate the lift coefficients and increase the central element until the value becomes equal to the parameters specified by the user was added to it.

As a result of testing the algorithm, the dependence of the growth of PSNR, SSIM, the algorithm operation time and the value of the filter coefficient was demonstrated. However, at this stage there is a significant drawback in the speed of the algorithm, with the initially low quality of the original images. In the framework of further work, it will be eliminated by optimizing the algorithms of the automated filter.

During the implementation of the automated algorithm, software was also developed that allows testing the algorithm and performing a visual assessment of the results obtained.

### References

- [1] Grechishkina EI, Yaduta AZ. Analysis of digital processing of medical images [In Russian]. Collection of selected articles based on the materials of scientific conferences of the State Research Institute "National Development". Saint-Petersburg; 2021: 32-35.
- [2] Sagheer SVM, George SN. A review on medical image denoising algorithms. Biomed Signal Process Control 2020; 61: 102036. DOI: 10.1016/J.BSPC.2020.102036.
- [3] Pugin EV, Zhiznyakov AL, Titov DV. Segmentation of images of blood vessels of the fundus using fuzzy image representation [In Russian]. News of the South-West State University 2018; 1(76): 6-17. DOI: 10.21869/2223-1560-2018-22-1-6-17.
- [4] Shagalova PA, Sokolova ES, Ryndov SN. Automation of blood microscopy image processing [In Russian]. Proc Int Conf on Computer Graphics and Vision "Graphicon" 2022; 32: 1157-1164.
- [5] Lefebvre AEYT, Ma D, Kessenbrock K, Lawson DA, Digmán MA. Automated segmentation and tracking of mitochondria in live-cell time-lapse images. Nat Methods 2021; 18(9): 1091-1102. DOI: 10.1038/s41592-021-01234-z.
- [6] Heydon P, Egan C, Bolter L, et al. Prospective evaluation of an artificial intelligence-enabled algorithm for automated diabetic retinopathy screening of 30 000 patients. Br J Ophthalmol 2021; 105(5): 723-728. DOI: 10.1136/bjophthalmol-2020-316594.
- [7] Czerniawski T, Leite F. Automated digital modeling of existing buildings: A review of visual object recognition methods. Autom Constr 2020; 113: 103131. DOI: 10.1016/j.autcon.2020.103131.
- [8] Su J, Li Z, Shao X, Ji C, Ji R, Zhou R, et al. Impact of a real-time automatic quality control system on colorectal polyp and adenoma detection: a prospective randomized controlled study (with videos). Gastrointest Endosc 2020; 91(2): 415-424.e4. DOI: 10.1016/j.gie.2019.08.026.
- [9] Sheshkus A, Chirvonaya A, Arlazarov VL. Tiny CNN for feature point description for document analysis: approach and dataset. Computer Optics 2022; 46(3): 429-435. DOI: 10.18287/2412-6179-CO-1016.
- [10] Zebari DA, Zeebaree DQ, Abdulazeez AM, Haron H, Hamed HNA. Improved threshold based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images. IEEE Access 2020; 8: 203097-203116. DOI: 10.1109/ACCESS.2020.3036072.
- [11] Wang S, Li C, Wang R, et al. Annotation-efficient deep learning for automatic medical image segmentation. Nat Commun 2021; 12: 5915. DOI: 10.1038/s41467-021-26216-9.
- [12] Öztürk Ş, Ahmad R, Akhtar N. Variants of Artificial Bee Colony algorithm and its applications in medical image processing. Appl Soft Comput 2020; 97: 106799. DOI: 10.1016/j.asoc.2020.106799.
- [13] Xue Z, Angara S, Levitz D, Antani S. Analysis of digital noise and reduction methods on classifiers used in automated visual evaluation in cervical cancer screening. Proc SPIE 2022; 11950: 1195008. DOI: 10.1117/12.2610235.
- [14] Kamenskiy AV. High-speed recursive-separable image processing filters. Computer Optics 2022; 46(4): 659-665. DOI: 10.18287/2412-6179-CO-1063.
- [15] Rylov KA, Kupriyanova KS, Kamenskiy AV. Influence of unequal apertures of digital filters laplacian "Truncated pyramid" and "Double pyramid" on the accuracy of television measuring systems. Program Comput Soft 2024; 50: 238-248. DOI: 10.1134/S036176882470004X.
- [16] Nechayev AA. Using power mean for image quality assessment. Int J Open Inf Technol 2024; 12(6): 65-75.
- [17] Nikin VV, Garina SV. Overview of methods and tools for evaluating the quality of frames in video file. International Journal of Professional Science 2020; 11: 56-63.
- [18] Kartashevsky VG, Mamyshev NR. Video quality assessment. MPQM, SSIM, NQM metrics [In Russian]. Development of modern technologies: Theoretical and practical aspects 2022; 30-34.
- [19] Sai SV. A method for assessing photorealistic image quality with high resolution. Computer Optics 2022; 46(1): 121-129. DOI: 10.18287/2412-6179-CO-899.
- [20] Sai SV, Kamenskiy AV, Kuryachy MI. Modern methods of analyzing and improving the digital images quality [In Russian]. Khabarovsk: Publishing house of the Pacific State University; 2020.
- [21] Filimotseva AA, Kamenskiy AV. ALF: Automated LDP Filter [In Russian]. Certificate of state registration of the computer program No. 2024665224 of June 27, 2024.

### Author's information

**Alina Anatolievna Filimotseva**, born in 2002, graduated from Tomsk State University of Control Systems and Radioelectronics, specialty 11.03.02 "Video Information Technologies". Research interests: digital image processing, programming, biomedical technologies. E-mail: [alyssa403@mail.ru](mailto:alyssa403@mail.ru)

**Andrey Victorovich Kamenskiy**, birthed in 1992, Graduated from Tomsk State University of Control Systems and Radioelectronics in 2015 with a specialist degree in "Audiovisual Engineering", code 210312. PhD in Engineering Science (2020). Senior Researcher in Television Automation Laboratory of Television and Control Department. Field of expertise: digital image processing, programming, biomedical technologies. E-mail: [andrul70@mail.ru](mailto:andrul70@mail.ru)

**Alexander Sergeevich Zakhlebin**, birthed in 1993, in 2017 graduated from Tomsk State University of Control Systems and Radio Electronics with a degree in "Radio Engineering". PhD in Engineering Science. Senior lecturer of Television and Control department of TUSUR, senior research associate in television automation laboratory of TUSUR. Field of expertise: 3d images, photogrammetric image processing. E-mail: [aerosnimok@gmail.com](mailto:aerosnimok@gmail.com)

*Revised August 28, 2024. The final version – January 24, 2025.*