High-speed recursive-separable image processing filters

A.V. Kamenskiy [1]
[1] *FSBEO HE «Tomsk State University of Control Systems and Radioelectronics»*
634050, Russia, Tomsk, Lenin avenue, building 40

Abstract

The development of modern technologies in the field of image formation leads to an increase in the size of the generated images, as a result the question of reducing the processing computational costs arises, and this is an important factor in the creation of real-time systems. The study provides a description of high-speed recursive-separable filters for improving the quality of images, which, due to the peculiarities of their implementation, can reduce the number of computational operations required for the image processing process. This type of filters is obtained from two-dimensional linear digital filters, which are modified by applying recursive and separable properties to them. The MATLAB environment computing method for implementation of these filters is described. An extensive performance research of the developed filters has been carried out at various sizes of the test image and on various experimental installations. The comparison with the classical two-dimensional convolution method of the developed filters is demonstrated, and it shows the time gain required for the image processing. The results obtained can be applied in biomedical image processing systems or in vision systems working in heavy weather conditions.

Keywords: image processing, recursive-separable algorithms, computational costs reduction, computer vision.

Citation: Kamenskiy AV. High-speed recursive-separable image processing filters. Computer Optics 2022; 46(4): 659-665. DOI: 10.18287/2412-6179-CO-1063.

Introduction

The digital image processing solves one of the most important problems of the modern world - it is the processing of various images, for example, images obtained from vision systems working in difficult weather conditions, or systems that solve problems for the formation of biomedical images (X-ray, computed tomography, etc.). When operating such systems, the main goal is to work in real time, or with minimal delay. The important factors of using digital image processing are the speed and efficiency as well as the video data analysis. As a result, the problem of reducing the number of computational operations maintaining the efficiency during the image processing arises.

In order to understand the essence of high-speed algorithms, we introduce the definition of high-speed performance. High-speed performance is the ability of devices, mechanisms and etc., to perform the work for which they are intended with the required speed [1]. Also, this property of processing algorithms can be formulated as the possibility of implementing classical linear filtering algorithms in a different representation due to any transformations, which leads to a decrease in the number of computational operations, simplification of the algorithm and, as a result, leads to a time gain, which is very important in various areas of human life which requires a computational costs reduction.

Book series written by T.S. Huang dedicated to digital image processing deals with three main areas of image processing: efficient encoding, restoration and improvement of the visual quality of images, pattern recognition. Many methods of restoration and improvement of the visual quality of images are based on the use of linear space-invariant filters [2, 3].

Two-dimensional linear filters have a very simple mathematical description, its operation is described by a discrete convolution of the input image with a mask of a given filter. A detailed description of the principles of operation of two-dimensional linear filters is described in the work of Kuriachiy M.I. [4].

Researches aimed at image processing computational costs reduction is carried out by many scientists. The issues of building a high-speed image processing system using digital filters and the System-On-Chip concept are considered by Zahojai O.I. and Soloshenko A.N. The use of digital filters makes it possible to implement high-speed image pre-processing and exposure control, which positively affects the representativeness of features in the image [5]. Methods for reducing the number of computational operations when performing digital filtering of images based on the operation of mathematical two-dimensional convolution are considered by Altman E.A. The paper considers known methods for decomposing a two-dimensional convolution into several convolutions with a smaller number of elements in relation to the problem of digital filtering [6]. M. Chobanu, in particular, is working on digital processing of multidimensional signals. In one of his works, the author describes an implemented hierarchical algorithm for coding multidimensional signals using an inseparable system. The developed method can become a breakthrough for such areas as three-dimensional television, medical technology, telecommunications, etc [7]. Sugimoto describes an efficient

Gaussian filter with constant time that provides high accuracy, with the second-order shift property DCT-5, which outperforms existing algorithms over a wide range of scales in terms of accuracy, stability and computational cost. The proposed algorithmic solution for the analysis of multiscale images makes it possible to overcome the problem of increasing the "computational" Gaussian convolution [8]. The study described in paper “Compiling High Performance Recursive Filters”, the authors shows that parallel and locally oriented implementations of infinite impulse response filtering pipelines can be obtained using software transformations that are mechanized with a domain compiler. The main result of the work is automatic fragmentation of common pipelines, achieved using a set of simple code transformations [9]. A new global and recursive tracking three dimensional algorithm (GReTA) presented at the IEEE conference, is capable to reconstruct continuous trajectories for a large number of objects and longtime intervals, even with frequent optical "occlusions". This recursive algorithm is based on the idea of global optimization of the solution both in space and in time. The applicability of global optimization is limited by computational complexity, which increases exponentially quickly with a long sequence. It implements a significant reduction in computational complexity using a recursive divide-and-conquer strategy that first optimizes matches globally (over shorter time intervals) and then iterates to cover the entire time sequence [10].

From the analysis of modern research in the field of image processing aimed at reducing of computational operations number, it follows that this direction is actively developing all over the world and that it is a demanded direction of research. However, some tasks in the direction indicated above remain unresolved. In many works, only the feature of recursiveness is used in the implementation of image processing algorithms. Therefore, the author and his colleagues are conducting research on recursive-separable image processing algorithms. As their name implies, they use not only the feature of recursiveness, but also the feature of separability.

In this direction, studies are carried out and published that confirm the effectiveness of recursive-separable image processing algorithms, there are considered the principles of recursive-separable filters construction are considered and studies that confirm their effectiveness in processing the images captured by various systems are presented.

The novelty of the presented work is that it describes the software implementation of the recursive-separable algorithms and presents the results confirming the theoretical calculations for evaluating the gain of computational operations in image processing.

1. Principles of recursive-separable filters construction

This work solves the problem of reducing computational operations when using two-dimensional linear digital filters. The solution to this problem is achieved by modifying them through the use of recursive and separable properties.

The construction of recursive-separable algorithms requires generating recursive cells (recirculators). Fig. 1 shows the generating recursive cell along the n_1-line (line recirculator – LR), fig. 2 shows generating recursive cell along the n_2-frame (frame recirculator – FR), realizing the corresponding orthogonal processing directions "moving average". LR – carries out processing on a row of the input data matrix, and FR – on a column of this matrix, z – delay element.

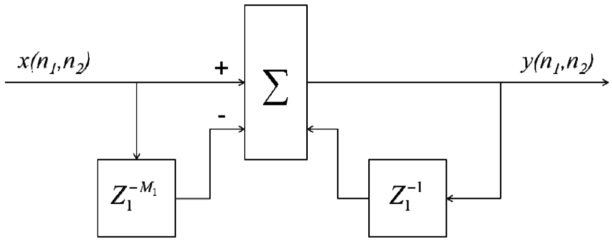

Fig. 1. Line recirculator

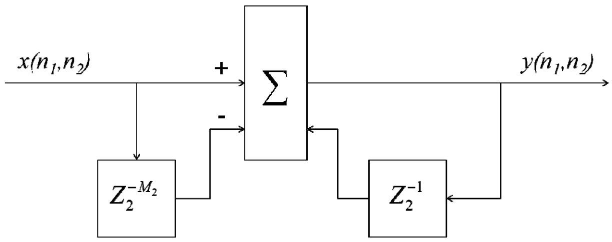

Fig. 2. Frame recirculator

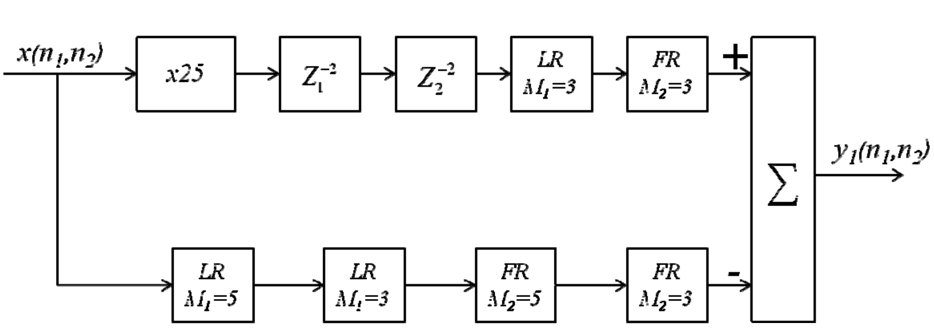

Рис. 3. Block diagram of two-dimensional truncated pyramid laplacian filter

The difference equation for the line recirculator will be:

$$y(n_1,n_2)=x(n_1,n_2)-x(n_1-M_1,n_2)+y(n_1-1,n_2), \quad (1)$$

and for the frame recirculator:

$$y(n_1,n_2)=x(n_1,n_2)-x(n_1,n_2-M_2)+y(n_1,n_2-1), \quad (2)$$

where $x(n_1, n_2)$ – input data, $y(n_1, n_2)$ – output data, n – current signal count number, including zero count ($n=0, 1, ...$), M_1 – line direction, M_2 – frame direction.

1.1. Implementation of the Laplacian Filter "Truncated Pyramid" (LTP)

This filter is constructed using the sum of two recursive branches; the principle of its construction is demonstrated in the block diagram of this filter shown in fig. 3, where z_1^{-2} – delay by two elements, $x(n_1, n_2)$ – input data, $y(n_1, n_2)$ – output data, n – current signal count number, including zero count ($n=0, 1, ...$), M_1 – line direction, M_2 – frame direction, LR – line recirculator, FR – frame recirculator.

The three-dimensional representation of the mask (aperture) for the LTP filter is shown in fig. 4.

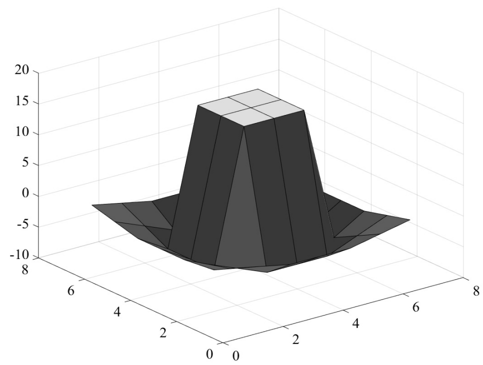

Fig. 4. The three-dimensional representation of the LTP aperture

Conducting theoretical calculations of the payoff by the number of computational operations. As an example, we took an aperture of 7×7 elements, which generally requires 49 multiplication operations and 48 addition/subtraction operations for a non-recursive filter design. With the recursive-separable version of the LTP filter construction, 13 addition/subtraction operations and one multiplication operation are required, which reduces the number of computational operations by 6.93 times. We chose of the 7×7 element aperture due to its difference from the classical 3×3 element apertures. Since with an increase in the size of the processing aperture, the gain in computational operations in the recursive-separable filters will increase in comparison with the non-recursive construction option.

1.2. Implementation of the Laplacian Filter "double pyramid" (LDP)

Let us give a block diagram for a recursively separable two-dimensional Laplacian Filter "Double Pyramid", shown in fig. 5, where z_1^{-3} – delay by three elements.

A three-dimensional view of the mask (aperture) of the LDP filter with a size of 7×7 elements is shown in fig. 6.

With a recursively separable implementation of the LDP filter, requires only 18 addition / subtraction operations and one multiplication operation, while for a 7×7 element aperture in the general case, 49 multiplication operations and 48 addition / subtraction operations are required for a non-recursive version of the filter construction, that is, with this variant of construction, the number of computational operations will be reduced by 5.11 times [11].

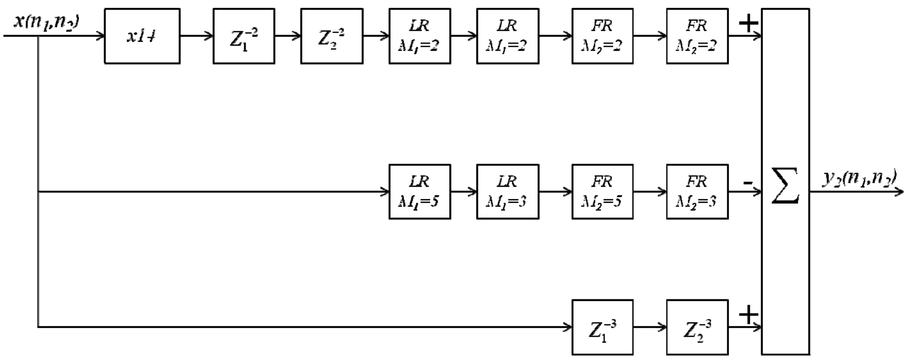

Fig. 5. The block diagram of two-dimensional LDP filter

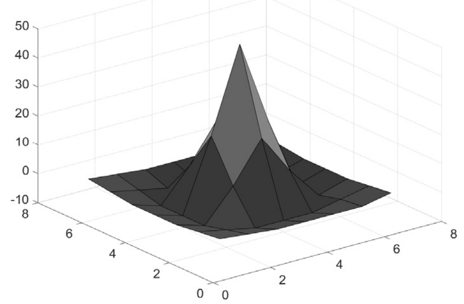

Fig. 6. Three-dimensional view of the aperture of the LDP filter

2. Software implementation of the developed algorithms

The next stage of the research was the software implementation of the developed recursive-separable algorithms. For the implementation of this idea, the MATLAB programming environment was used, which, after implementation, allows the developed algorithms to be adapted into C / C++ code.

For the developed algorithms to work, it is required to set a number of input parameters:

- X – processed image;
- A_1 – coefficient for the rise of the central aperture of 3×3 elements;

- A_2 – coefficient for increasing the central element of the matrix by a given value.

Coefficient A_1 serves to lift the central element of the original mask with a size of 3×3 elements. In this case, there is not a simple addition of a given value to each value of the central element of 3×3 size, but its proportional increase, shown in fig. 7.

$$\underset{A_1=0}{\begin{bmatrix} 5 & 19 & 5 \\ 19 & 48 & 19 \\ 5 & 19 & 5 \end{bmatrix}} \rightarrow \underset{A_1=1}{\begin{bmatrix} 6 & 21 & 6 \\ 21 & 52 & 21 \\ 6 & 21 & 6 \end{bmatrix}} \rightarrow \underset{A_1=2}{\begin{bmatrix} 7 & 23 & 7 \\ 23 & 56 & 23 \\ 7 & 23 & 7 \end{bmatrix}}$$

Fig. 7. Changing the central element of the LDP filter mask

As can be seen from fig. 7 there is a proportional increase in the entire central matrix of the filter. By changing the A_1 coefficient by an integer value of 1; 2; 3, etc., we increase the sum of the coefficients of the original matrix each time it increases by 16 units.

Coefficient A_2 adds the specified value to the central element of the filter mask.

2.1. Implementation of the laplacian filter "Truncated Pyramid"

The first step is to determine the sum of the matrix coefficients to find the coefficient by which the final image will be divided to normalize the brightness. Next step is the processing along the first branch of the structural diagram shown in fig. 3. The original image is multiplied by 25, and in the case of raising the central aperture by the coefficient A_1, the required value is added to the multiplier 25. After that, the image is processed by a line recirculator with an aperture size $M_1 = 3$ and a frame recirculator with an aperture size $M_2 = 3$.

At the end of the filtering procedure, the image is supplemented with "zeros", since the summation of matrices of the same dimension is required for correct operation.

Further, the processing procedure takes place along the second branch of the structural diagram. According to it, the image is processed by two line recirculators with aperture sizes $M_1 = 5$ and $M_1 = 3$, after which the it is processed by two frame recirculators with aperture sizes $M_2 = 5$ and $M_2 = 3$.

After performing this processing, the procedure of multiplying the original image by the coefficient A_2, if this coefficient is specified, takes place. After that, the procedure of summing intermediate images and the formation of the final image after processing takes place.
The code fragment is given below:

W = X*A2; *% Original image multiplied by a coefficient A2*

i1 = size (W, 2); *% number of columns in W matrix*

j1 = size (W, 1); *% number of lines in W matrix*

W0 = zeros (j1 + 6, i1 + 6); *% zeros matrix*

W0 (4:4 + j1–1, 4:4 + i1–1) = W; *% W matrix surrounded by zeros*

rez = (W0 + (Z*f* – y5)) / (A2 + S); *% Summation of the difference between the images after the upper (Zf) and lower (y5) parts of the filter and the image with the A2 coefficient and normalization of the image using the A2 coefficient and the coefficient of the 3×3 central element plucked when lifting (S).*

2.2. Implementation of the laplacian filter "Double Pyramid"

At the first step, the sum of the matrix coefficients is ccalculated to determine the coefficient by which the final image will be divided to normalize the "digital" brightness level. Next step is the processing along the first branch of the structural diagram shown in fig. 5. The original image is multiplied by 14, and in the case of raising the central aperture by a coefficient A_1 the required value is added to the multiplier 14. Then the image is processed by a lowercase recirculator with aperture sizes $M_1 = 2$ and two frame recirculators with aperture size $M_2 = 2$.

At the end of the filtering procedure, the image is supplemented with "zeros", since the summation of matrices of the same dimension is required for correct operation.

Further, the processing procedure takes place along the second branch of the structural diagram. On it, the input image is processed by two line recirculators with aperture sizes $M_1 = 5$ and $M_1 = 3$, after which the processing process takes place with two frame recirculators with aperture sizes $M_2 = 5$ and $M_2 = 3$.

After this processing, starts the procedure of multiplying the original image by the coefficient $(1 + A_2)$. After that, the procedure of summing intermediate images and the formation of the final image after processing takes place. The code fragment is given below:

i0 = size (X, 2); *% number of columns in y4 matrix*

j0 = size (X, 1); *% number of lines in y4 matrix*

Zf0 = zeros (j0 + 6, i0 + 6); % zeros matrix

Zf0 (4:4 + j0 – 1, 4:4 + i0 – 1) = X*(A2 + 1); *% W matrix surrounded by zeros*

rez = (Zf – y7 + Zf0) / (S + A2). *Summation of the difference between the images after the upper (Zf) and lower (y7) parts of the filter and the image with the A2 coefficient and normalization of the image using the A2 coefficient and the coefficient of the 3×3 central element plucked when lifting (S).*

2.3. Implementation of the classical two-dimensional convolution CTC

The algorithm of classical two-dimensional convolution was implemented (CTC). The implementation of this algorithm is presented in the code below:

S = sum (sum (Y)); *% filter matrix sum of coefficients*

i1 = size (X, 2); *% number of columns in X matrix*

j1 = size (X, 1); *% number of lines in X matrix*

i2 = size (Y, 2); *% number of columns in Y matrix*

j2 = size (Y, 1); *% number of lines in Y matrix*

k = j1 + (j2 – 1);

m = i1+ (i2 – 1);

y = zeros (k, m); *% variable to write the resulting value*

for k = 1: j1 + (j2 – 1); *% selection of processing line*

for m = 1: i1 + (i2 – 1); *% selection of processing column*

y (k, m) = sum (sum (Z3 (k:k + (j2 – 1), m:m + (i2 – 1)).*Y)); *% convolution of the original image surrounded by zeros Z3 and the filter matrix (Y)*
end
end
rez = y / S;

3. Experimental research

The study was divided into two stages. The first "Basic" stage is a general check of the efficiency of the developed algorithms with respect to the classical two-dimensional convolution. The second "Confirming" stage was done to confirm the results obtained in the first experiment. In an experimental study, all the developed filter algorithms were converted into internal MATLAB functions to simplify the experiment.

3.1. "Basic" experiment to assess the effectiveness of the developed algorithms

For experimental studies, images of various dimensions and formats were used, namely 640×480"TIF", 1280×720"BMP", 1280×1024"TIF", 1920×1080"BMP", 3000×2000"JPG", 5472×3078"JPG", 5616×3744"JPG". In the course of the experiment, a comparison was made between the implementation of the classical two-dimensional convolution and the developed algorithms for two-dimensional recursive-separable filters.

The main experiment was carried out on a platform with the following characteristics: CPU Intel(R) Core(TM) i7-6700HQ, 2.60 GHz, RAM 16 Gb. Tab. 1 shows the results of evaluating the performance for an image of 640×480 pixels in the TIF format.

Tab. 1. 640×480 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	2.9763	0.3335	0.4078
2	3.1284	0.3415	0.4103
3	2.8980	0.3409	0.4341
4	2.9135	0.3400	0.4336
5	3.1218	0.3263	0.4112
6	2.9331	0.3363	0.4252
7	3.1134	0.3458	0.4262
8	2.9509	0.3398	0.4321
9	3.0765	0.3404	0.4229
10	3.1313	0.3431	0.4226
∑/10	3.0243	0.3388	0.4226

The experimental results show that when using algorithms as internal MATLAB functions LTP filter showed a gain in performance by 8.93 times, and the LDP filter shown gain by 7.16 times of the standard image convolution process.

At the second stage of the experiment, an image of size 1280×720 pixels in BMP format, and the results of evaluating the performance are presented in tab. 2.

The experimental results show that when using the algorithms as internal functions of MATLAB, LTP filter showed a gain in performance by 8.05 times, and the LDP filter by 6.16 times relative to the standard process of CTC.

Tab. 2. 1280×720 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	8.8017	1.0512	1.3827
2	9.0167	1.0790	1.4665
3	9.0490	1.0872	1.4674
4	9.0299	1.0770	1.4479
5	8.6302	1.1427	1.4493
6	9.2393	1.1195	1.4566
7	8.6713	1.1492	1.4867
8	8.8479	1.1271	1.4843
9	8.8951	1.1056	1.3770
10	9.1414	1.1557	1.4685
∑/10	8.9323	1.1094	1.4487

At the third stage of the experiment, an image with a size of 1280×1024 pixels in TIF format was processed, and the results of measuring the performance are presented in tab. 3.

Tab. 3. 1280×1024 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	13.0535	2.2556	3.0545
2	12.5253	2.3112	3.0014
3	12.6339	2.3169	3.0544
4	13.0610	2.3473	3.0408
5	12.4212	2.3107	3.0451
6	13.0539	2.2863	3.0531
7	12.6420	2.3213	3.0104
8	12.7094	2.3137	3.0539
9	12.5054	2.2990	2.9758
10	12.5232	2.3659	3.0316
∑/10	12.7129	2.3128	3.0321

The experimental results show that when using the algorithms as internal functions of MATLAB, LTP filter showed a gain in performance by 5.50 times, and the LDP filter by 4.19 times relative to the standard process of CTC.

In the fourth experiment, an image with a size of 1920×1080 pixels in BMP format was used as an input; the results of measuring the performance are presented in tab. 4.

The experimental results show that when using the algorithms as internal functions of MATLAB, LTP filter showed a gain in performance by 7.86 times, and the LDP filter by 6 times relative to the standard process of CTC.

Tab. 4. 1920×1080 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	20.0636	2.5910	3.3868
2	19.5431	2.5272	3.3630
3	20.6623	2.4855	3.2446
4	19.8688	2.6223	3.3809
5	20.6524	2.6059	3.3308
6	19.7270	2.6182	3.3626
7	19.3841	2.5296	3.3210
8	20.4544	2.5618	3.3707
9	20.2557	2.6296	3.3982
10	20.6269	2.4376	3.3878
∑/10	20.1238	2.5609	3.3546

In the fifth experiment, an image of 3000×2000 pixels in JPG format was used as an input; the measurement results are presented in tab. 5.

Tab. 5. 3000×2000 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	57.4257	7.6080	9.9401
2	57.0687	7.2609	10.0507
3	58.1517	7.6518	9.9815
4	58.8604	7.3207	10.0094
5	58.3374	7.5059	9.8032
6	58.1652	7.6048	9.7607
7	57.4552	7.4921	9.7831
8	58.9665	7.4803	9.7285
9	57.1110	7.4686	9.9751
10	57.2676	7.6252	9.9318
∑/10	57.8809	7.5018	9.8964

The experimental results show that when using the algorithms as internal functions of MATLAB, LTP filter showed a gain in performance by 7.72 times, and the LDP filter by 5.85 times relative to the standard process of CTC.

In the sixth experiment, an image with a size of 5472×3078 pixels in JPG format was used as an input; the results of measuring the performance are presented in tab. 6.

Tab. 6. 5472×3078 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	160.8304	21.1836	27.5091
2	167.2055	20.9470	28.1730
3	159.8483	21.4212	27.6421
4	160.3264	20.7794	27.5637
5	162.2150	21.1521	28.3405
6	163.6381	21.5628	27.9274
7	160.5580	21.1151	27.3746
8	162.0221	20.9400	27.0660
9	162.3684	21.1567	28.2759
10	164.5893	20.4278	28.3244
∑/10	162.3602	21.0686	27.8197

The experimental results show that when using the algorithms as internal functions of MATLAB, LTP filter showed a gain in performance by 7.71 times, and the LDP filter by 5.84 times relative to the standard process of CTC.

In the seventh experiment, an image of 5616×3744 pixels in JPG format was used as an input; the measurement results are presented in tab. 7.

The experimental results show that when using the algorithms as internal functions of MATLAB, LTP filter showed a gain in performance by 6.81 times, and the LDP filter by 5.22 times relative to the standard process of CTC.

Analyzing the results obtained, we can conclude that when comparing two implementations of the classical and recursive-separable convolution, the gain in time costs varies depending on the dimension of the original image, but on average it is saved on the gain calculated in paragraph 1 by a factor of 7 for the LTP filter and in 5 times for the LDP filter with a scanning multi-element aperture size of 7×7 elements.

Tab. 7. 5616×3744 Image processing

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	200.8406	30.0880	39.2395
2	198.4460	29.9074	39.2885
3	200.9667	29.0851	38.1552
4	201.6156	29.0606	37.9196
5	207.4295	29.9186	39.8160
6	207.6623	29.2552	37.9889
7	205.7146	30.3672	39.4183
8	202.2933	30.0239	39.4351
9	200.5313	30.3891	39.2587
10	203.9872	29.7963	38.5208
∑/10	202.9487	29.7891	38.9041

3.2. "Confirming" experiment to evaluate the effectiveness of the developed algorithms

To confirm the conclusions obtained in the previous paragraph, it is necessary to repeat the results obtained. Therefore, it is necessary to conduct a similar study of performance on other computing platforms. For this, another 3 other platforms were chosen for carrying out a computational experiment. Since it follows from the previous study that the gain in performance does not actually depend on the dimension and format of the image being processed, it is possible to carry out a confirmation experiment on one image size (1920×1080) and with fewer repetitions of the processing procedure, reducing it to 5 iterations.

The first experiment was carried out on a platform with the following characteristics: CPU Intel ×64, 2.70 GHz, RAM 8 Gb. The results are shown in Table 8.

From the analysis of the results obtained, it follows that the constructed adaptive filters are faster than the classical convolution. So, when working with the algorithms as internal functions of MATLAB, the adaptive filters LTP and LDP outperform the classical convolution by 6.57 and 5.41 times, respectively.

Tab. 8. Image processing experiment №1

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	3.832	0.583	0.746
2	4.006	0.560	0.683
3	3.788	0.603	0.702
4	3.788	0.615	0.717
5	3.667	0.545	0.683
∑/5	3.816	0.581	0.706

The second experiment was carried out on a platform with the following characteristics: CPU Intel ×64 2.30 GHz, RAM 16 Gb. The results are shown in tab. 9.

Tab. 9. Image processing experiment №2

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	4.056	0.651	0.821
2	4.147	0.643	0.831
3	4.051	0.682	0.804
4	4.105	0.605	0.815
5	4.083	0.634	0.833
∑/5	92,4	643	820,8

So, when working with the algorithms as internal functions of MATLAB, the adaptive filters LTP and LDP outperform the classical convolution by 6.36 and 4.98 times, respectively.

The third experiment was carried out on a platform with the following characteristics: CPU AMD ×64 2.2 GHz, RAM 8 Gb. The results are shown in tab. 10.

Tab. 10. Image processing experiment №3

№	Processing time, s		
	MATLAB function		
	CTC	LTP	LDP
1	19.152	3.556	5.002
2	18.375	3.121	3.712
3	18.425	2.858	3.652
4	21.701	2.895	3.754
5	20.035	3.056	3.838
∑/5	19.538	3.097	3.922

From the analysis of the results obtained, it follows that the constructed adaptive filters are faster than the classical convolution. So, when working with the algorithms as internal functions of MATLAB, the adaptive filters LTP and LDP outperform the classical convolution by 6.31 and 4.89 times, respectively.

Conclusion

The conducted experimental study demonstrates the feasibility of using the developed recursive-separable algorithms to reduce the computational costs required for the image processing process. From the presented study, it follows that the reduction in computational costs does not change significantly when using various computing platforms and is on average 7 times for the LTP filter and 5 times for the LDP filter with a scanning multi-element aperture of 7×7 elements.

These filters have feature of adaptability, which allows you to change the size of their multidimensional scanning aperture. It should be noted that with an increase in the size of the aperture, the number of computational operations remains unchanged. In further studies of these algorithms, it is planned to prove these statements. Also, in the course of further research of the developed algorithms, it is planned to consider the possibility of their parallelization, in order to further reduce computational operations.

Acknowledgments

The research was carried out at the expense of the Russian Science Foundation grant No. 21-79-10200 at TUSUR.

References

[1] Efremova TF. New dictionary of the Russian language. Explanatory and derivational. Moscow: "Russkii Jazyk" Publisher, 2000.
[2] Huang TS. Two-dimensional digital signal processing i: linear filters. Top Appl Phys 1981; 42: 210.
[3] Huang TS. Picture processing and digital filtering. 2nd ed. Springer-Verlag; 1979.
[4] Kuriachiy, M.I. Digital signal processing: Textbook for high schools with UMO [In Russian]. Tomsk: "TUSUR" Publisher; 2009.
[5] Zahojai OI, Soloshenko AN. High-speed image processing with FPGA-based digital filters. Collection of Scientific Papers of the Donbass State Technical University 2013; 39: 250-254.
[6] Altman EA. Zaharenko EI, Vaseeva TB. Application of the two-dimensional convolution decomposition method in the implementation of digital filters. Scientific Bulletin of NSTU 2017; 69(4): 95-104.
[7] Chobanu M, Volkov M. Digital processing of multidimensional signals: prerequisites for a breakthrough. Electronics: Science, Technology, Business. Microprocessor and Computer Technology 2007: 3: 64-74.
[8] Sugimoto K, Kamata S. Fast Gaussian filter with second-order shift property of DCT-5. Proc IEEE Int Conf on Image Processing (ICIP) 2013: 514-518.
[9] Chaurasia G, Ragan-Kelley J, Paris S, Drettakis G, Durand F. Compiling high performance recursive filters. HPG '15 Proc 7th Conf on High-Performance Graphics 2015: 85-94.
[10] Attanasi A, Cavagna A, Castello LD, Giardina I, Jelic A, Melillo S, Parisi L, Pellacini F, Shen E, Silvestri E, Viale M. GReTA – a novel global and recursive tracking algorithm in three dimensions. IEEE Trans Pattern Anal Mach Intell 2015; 37(12): 2451-2463.
[11] Sai SV, Kamensky AV, Kuryachy MI. Modern methods of analyzing and improving the digital images quality. Khabarovsk: Publishing house of the Pacific State University; 2020.

Author's information

Andrey Victorovich Kamenskiy, born 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: andru170@mail.ru .

Code of State Categories Scientific and Technical Information (in Russian – GRNTI): 47.51.39
Received October 18, 2021. The final version – January 26, 2022.