High-performance discrete wavelet transform for JPEG XS standard

P.A. Lyakhov [1,2], M.V. Bergerman [2], N.N. Nagornov [1], A.S. Abdulsalyamova [2]
[1] Department of Mathematical Modeling, North-Caucasus Federal University, 355017, Stavropol, Russia, Pushkin st. 1;
[2] North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, 355017, Stavropol, Russia, Pushkin st. 1

Abstract

This paper presents a high-speed method for forward and inverse discrete wavelet transform (DWT) intended for the JPEG XS image compression standard. Unlike state-of-the-art approaches, which process pixels sequentially, the proposed algorithm employs the Winograd method to compute groups of 2-5 pixels in parallel within a single clock cycle. We determine the minimum fractional bit-widths required for fixed-point arithmetic to ensure reconstructed image quality with a peak signal-to-noise ratio (PSNR) of at least 40 dB. Hardware modeling using the OpenLane environment demonstrates that the proposed method increases throughput by up to 109% for forward DWT and up to 144% for inverse DWT compared to state-of-the-art techniques. The optimal configurations are 3-pixel fragments for forward and 4-pixel fragments for inverse transforms. The proposed DWT approach is recommended for real-time systems where processing speed is critical, particularly in medical imaging and satellite data processing.

Keywords: digital image processing, Le Gall filter, hardware modeling, Winograd computation.

Citation: Lyakhov PA, Bergerman MV, Nagornov NN, Abdulsalyamova AS. High-performance discrete wavelet transform for JPEG XS standard. Computer Optics 2026; 50(3): 1725. doi: 10.18287/COJ 1725.

Introduction

Digital image processing is widely used in many areas of science and technology [1]. Image and video processing standards need to be improved due to the increase in information transfer speeds and the size of data being delivered. The JPEG standard [2] is widely used for storing, transmitting, and processing graphic information on personal computers. This standard uses a special image compression algorithm and reduces the file size on the computing device. The JPEG standard uses discrete cosine transform (DCT) for image compression [3]. This method is a modification of the discrete Fourier transform, which is widely used in digital signal processing tasks. However, DCT performs compression with a loss of image quality.

In 2019, the International Organization for Standardization (ISO) introduced an image and video coding standard called JPEG XS, which allows lossless image compression [4]. This standard uses wavelets in its compression algorithm [5]. They are used in tasks of information compression [6], image analysis [7], video processing [8], noise reduction [9], and other tasks. Image compression in the JPEG XS standard is pre-processed by increasing the bit–depth of pixels. The next step is discrete wavelet transform (DWT). Then, the resulting images are quantized and rounded to integer pixel values. The final stage of image compression is Huffman entropy coding. The bitrate control unit controls the size of the entropy coding code stream after image compression. Image or video signal restoration is performed in the reverse order of encoding: entropy decoding; reverse quantization, inverse DWT, and bit–depth reduction. Fig. 1 shows the JPEG XS image encoding and decoding scheme.

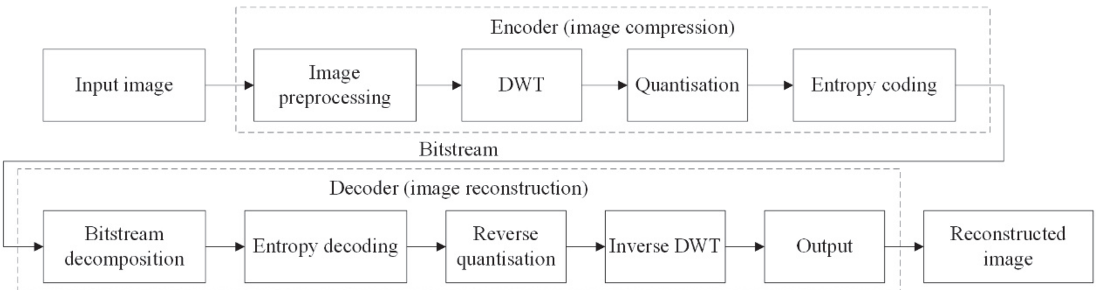

Fig. 1. Schematic diagram of JPEG XS image compression and reconstruction processing operation

New approaches and algorithms are being developed to improve the compression and restoration performance of images and videos for the JPEG XS standard at one of the processing stages. Thus, in [10], the authors presented methods for improving the visual characteristics of JPEG XS entropy coding. In [11] a modular scheme for parallel entropy coding and decoding was proposed for the development of a device on a programmable logic integrated circuit. The authors in [12] propose video coding techniques for efficient information transmission. Most of the work focuses on improving the performance of the entropy coding block for the JPEG XS codec.

The JPEG XS standard uses a 5/3 wavelet filter for image compression and restoration, which is a special case of the Le Gall filter [13]. The JPEG XS standard uses a lifting scheme [14] as the DWT method. The main advantage of this method is its low computational complexity. However, the disadvantage of this approach is the low pixel processing speed, since the computations are performed sequentially. To increase the processing speed, parallel computing methods are required. One such approach is to use of the Winograd method (WM), which allows several pixels to be calculated simultaneously [15]. The computational approach of this method has already been used in various fields of knowledge. The main reason for using this method is the technique of calculating groups of pixels, which reduces the number of multiplication operations by increasing the number of addition operations, thereby reducing computational complexity. This method is most often used when performing computations for various tasks in convolutional neural networks: detection of heart disease based on ECG signals [16], detection of computing clouds [17], detection of objects in an image [18], etc. The method is also used in the development of hardware acceleration systems. WM is also used in the development of hardware accelerators for high-performance matrix multiplication calculations [19].

This paper proposes WM-based method for high-performance 1D forward and inverse DWT of JPEG XS images. The accuracy of WM coefficients is determined using computer simulation to obtain high-quality reconstructed images. Hardware modeling is performed using calculated coefficient sizes for forward and inverse DWT, and a comparison with state-of-the-art DWT approaches is also carried out.

The rest of the paper is organized as follows. Section 1 describes the state-of-the-art wavelet processing methods. In Section 2 presents the proposed DWT method. Section 3 describes the accuracy analysis of the proposed DWT method for high-quality image processing. Section 4 describes the hardware modelling of the proposed method, presents the simulation results, discusses the obtained results and describes the feature of the proposed method for performing DWT. Section 5 presents conclusions and suggests promising areas of application for the proposed DWT method.

1. Preliminaries

This section describes state-of-the-art DWT methods: direct method and lifting scheme. The direct method is the standard method, which is performed using a convolution operation with step size of 1. The Lifting scheme is the next generation of wavelet signal and image processing.

1.1. Direct DWT

An image in digital form is denoted as $I(x, y)$, where x represents the number of pixels in width and y represents the number of pixels in height. Wavelet transform with direct method performs filtering of signals in which even-order filters are used. 1D wavelet filtering by width by direct method is performed by the formula (1) and by height by the formula (2):

$$\tilde{I}(a,b) = \sum_{i=0}^{k-1} I(a, b-i)F(i), \tag{1}$$

$$\tilde{I}(a,b) = \sum_{i=0}^{k-1} I(a-i, b)F(i), \tag{2}$$

where I and $\tilde{I}$ are the original and processed images with the with row number a and column number b, F is a wavelet filter of order k. Image filtering in wavelet processing is performed by two computational channels:

1.On the low-pass filter LD;

2.On the high-pass filter HD.

After convolution using both filters, downscaling is performed in steps of 2 (↓2). The result of the processing is a low-frequency $\tilde{I}_{LD}$ and high-frequency $\tilde{I}_{HD}$ image.

Le Gall's filter coefficients for performing the forward DWT using the direct method are presented in formulas (3) and (4):

$$F_{Direct,LD} = \left(\frac{1}{4}; \frac{3}{4}; \frac{3}{4}; \frac{1}{4}\right), \tag{3}$$

$$F_{Direct,HD} = \left(-\frac{1}{4}; -\frac{3}{4}; \frac{3}{4}; \frac{1}{4}\right). \tag{4}$$

The direct method of inverse DWT is performed as follows. First, upsampling is performed with the same step as in forward DWT. In this case, the step is equal to 2 (↑2). This operation involves inserting zeros between each pixel in the analyzing images. Then, the low-frequency F_{LR} and high-frequency F_{HR} wavelet filters are applied to the post-upsampled images. Finally, the results of the two filtered images are combined by summing the pixel values at the corresponding coordinates. If all calculations are performed without any loss of accuracy, the reconstructed image $\tilde{I}$ will be identical to the original image I. Fig. 2 shows the process of performing a full DWT on an image using the direct method.

Le Gall's filter coefficients for performing inverse DWT are given in formulas (5) and (6):

$$F_{Direct,LR} = \left(-\frac{1}{4}; \frac{3}{4}; \frac{3}{4}; -\frac{1}{4}\right), \tag{5}$$

$$F_{Direct,HR} = \left(-\frac{1}{4}; \frac{3}{4}; -\frac{3}{4}; \frac{1}{4}\right). \tag{6}$$

1.2. Lifting scheme DWT

The lifting scheme is another DWT approach. This method is widely used in image processing, data compression, noise filtering and other digital signal processing tasks. The essence of the lifting scheme is as follows: in the first stage, the original signal is divided into two parts: the even part of the image I_a and the odd part of the image I_b. This can be done by simply alternating the elements of the array. The second stage, an attempt is made to predict the values of the odd part based on the values of the even part. This is done using some predictive operator P. The result of the computation will be the high-frequency image $\tilde{I}_{HD}$. In the third stage, the values of the even part are updated taking into account prediction errors. For this purpose, the update operator U is used. The result of calculations in this operator is a low-frequency image $\tilde{I}_{LD}$.

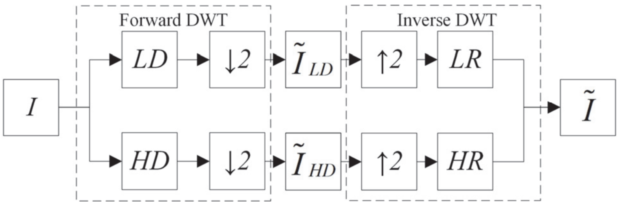

Fig. 2. Schematic of full wavelet filtering of the image by direct method

Le Gall's filter coefficients for performing forward DWT using the lifting scheme are given in formulas (7) and (8):

$$DWT_{LS,LD} = \left(-\frac{1}{8}; \frac{2}{8}; \frac{6}{8}; \frac{2}{8}; -\frac{1}{8}\right), \tag{7}$$

$$DWT_{LS,HD} = \left(-\frac{1}{2}; 1; -\frac{1}{2}\right). \tag{8}$$

The inverse DWT lifting scheme is performed in reverse order. The value of the low-frequency image is subtracted from the high-frequency image passed through the operator U. The value obtained after passing through the P operator is added to the low-frequency image. To obtain the reconstructed image, the processed data is summed using the concatenation operator C. This operation involves performing an addition operation on the corresponding pixels on the row and column of the processed images. If all calculations were performed without loss of accuracy, the reconstructed image will be identical to the original. The full DWT image processing by lifting scheme is shown in Fig. 3.

Le Gall's filter coefficients for performing inverse DWT using the lifting scheme are given in formulas (9) and (10):

$$DWT_{LS,LR} = \left(-\frac{1}{8}; -\frac{2}{8}; \frac{6}{8}; -\frac{2}{8}; -\frac{1}{8}\right), \tag{9}$$

$$DWT_{LS,HR} = \left(\frac{1}{2}; 1; \frac{1}{2}\right). \tag{10}$$

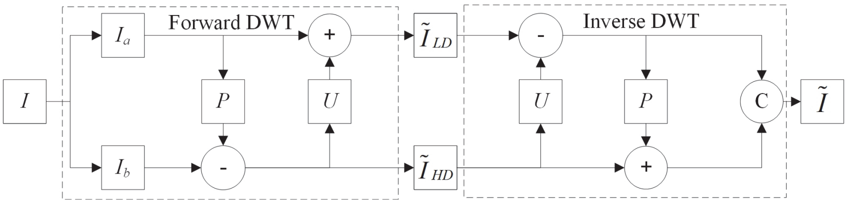

Fig. 3. DWT image processing by lifting scheme method

2. Winograd method for forward and inverse DWI

The WM is an alternative to the direct method for performing DWT [15]. WM for 1D data processing is denoted as $F(m, r)$, where m indicates the size of the image fragments to be processed, r is the order of the wavelet used. For forward DWT a third number $F(m, r, d)$ is used, where d denotes the degree of sampling rate reduction. In our case, $d = 2$. 1D wavelet filtering using the Winograd method is performed according to formula (11):

$$\tilde{I} = A^T\left((GK) \odot (B^T I)\right), \tag{11}$$

where $\tilde{I}$ is a fragment of the processed image (size), A^T, G and B^T are transformation matrices, K is filter coefficients, $\odot$ is the matrix elementwise multiplication operator, I is a fragment of the original image. The GK product is calculated in advance with the corresponding analyzing filters. The values of the transformation matrices A^T, G and B^T depend on the number of image fragments and the filter order. The work in [20] shows that the values of the coefficients of the transformation matrices influence the selection of Lagrange polynomial points for constructing the Vandermonde matrix. The filter coefficients K for WM calculations for JPEG XS standard images correspond to the coefficients of the Le Gall filter according to formulas (2-5).

To perform wavelet transform with decimation using WM, the image is divided into two parts: the image with even I_a and odd I_b signal parts, as in the lifting scheme. Then, a multiplication operation is performed in parallel between the divided signals and the matrix B_D^T. In the next step, piecewise multiplication is performed with the corresponding filters for odd signals $GK_{LD,odd}$, $GK_{HD,odd}$, and even signals $GK_{LD,even}$, $GK_{HD,even}$. The obtained results are multiplied by the matrix A_D^T, for each computational channel, the addition of the corresponding processed low-frequency signals ($\tilde{I}_{LD,odd}$ and $\tilde{I}_{LD,even}$) and high-frequency signals ($\tilde{I}_{HD,odd}$ and $\tilde{I}_{HD,even}$) is performed in the last step of the wavelet transform.

Before performing the inverse DWT with WM, image upsampling is carried out, where the resolution of the images is increased by a factor of 2. In this process, odd pixels in the images are filled with values from the original images, and the even pixels are assigned a value of "0". Next, the low-frequency and high-frequency image fragments are multiplied in parallel by a matrix B_R^T. The product GK is pre-calculated, provided that all coefficients of the original filter are known. After that, the corresponding values of GK_{LR} and GK_{HR} are multiplied element-wise by the matrix $B_R^T I$. The next step involves multiplying the resulting values with matrix A_R^T, which calculates the group of pixels of the image fragment. The final computational step to obtain the reconstructed image $\tilde{I}$ is to add the corresponding pixels from the low-frequency and high-frequency images.

A schematic diagram illustrating the full DWT for image processing using WM is shown in Fig. 4.

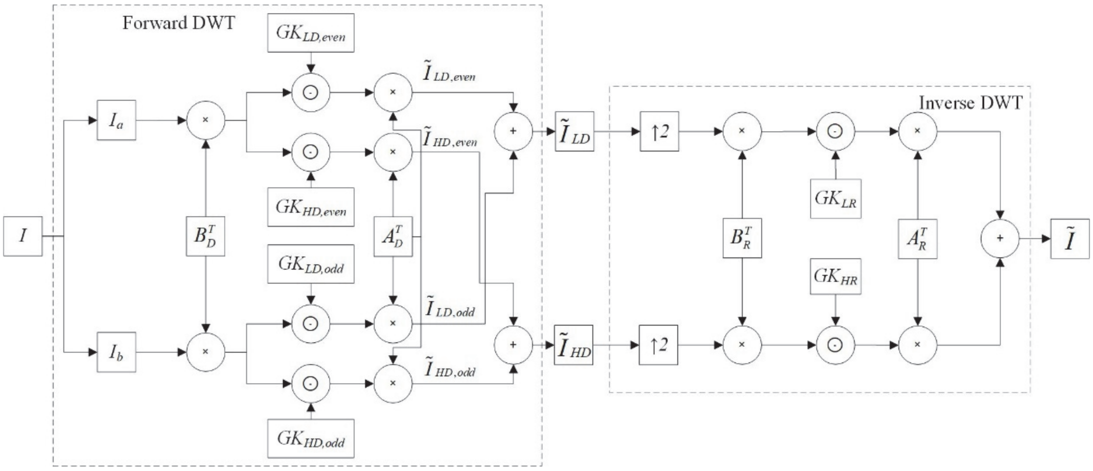

Fig. 4. Proposed schematic diagram of the full DWT using WM

3. Accuracy analysis of forward and inverse DWT using Winograd method

We used fixed-point arithmetic for DWT calculations. This approach has a number of advantages, including high accuracy for small number ranges, fast computation, and ease of implementation. However, there are also disadvantages, particularly regarding accuracy when working with fractional numbers, especially those that are divisible by numbers that are not powers of two. Such calculations can lead to a loss of accuracy, as some fractions may have an infinite number of digits in binary representation. There are various methods for improving accuracy, such as increasing the number of bits used to store the fractional part or using special number formats.

For the proposed DWT method, sets of Lagrange polynomials with positive and negative values of degree two were selected due to their low time complexity, based on theoretical estimates from [21]:

$$L \in \{0,\ 1, -1, 2, -2, \dots, 2^n, -2^n, 2^{n+1}, -2^{n+1}, \dots, \infty\}.$$

The values of the GK matrices are presented in Appendices A and B for forward and inverse DWT based on the selected Lagrange points set, respectively.

We used the Peak Signal-to-Noise Ratio (PSNR) metric to evaluate the quality of processed images after applying DWT. This metric evaluates the quality of the processed image compared to the quality of the original image. PSNR is calculated using formula (11):

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{12}$$

where MAX_I is the maximum pixel brightness value, MSE is the mean–square error which is calculated using formula (12):

$$\mathrm{MSE} = \frac{1}{xy}\sum_{i=0}^{x-1}\sum_{j=0}^{y-1}\left|I(x,y) - \tilde{I}(x,y)\right|^2. \tag{13}$$

Modeling of the operation of the DWT images was performed in MATLAB version R2021b. For the simulation, we used an image with the size of 7680×4320 pixels, which has a color depth of 8 bits and 3 color channels: red, green, blue (RGB). Fig. 5 shows the image used to compare the forward and inverse DWT methods.

A processed image is considered high quality if the PSNR metric is equal to or greater than 40 dB for images with a bit–depth of 8 bits [22]. If we consider the WM coefficients used to multiply the GK matrices in the forward DWT for groups of 2 and 3 pixels, which are described in detail in Appendix A, we can observe that all the denominators of these coefficients are multiples of two raised to the second power. This indicates that 2 bits are sufficient for the fractional part for accurate calculations.

The results of the software modeling are presented in Tab. 1. The analysis indicates that to perform a forward DWT with WM for groups of 2 and 3 pixels, an additional 2 bits are required for the fractional part to ensure quality preservation. For inverse DWT applied to segments of 2 and 3 pixels, an additional 3 bits and 7 bits, respectively, must be used to obtain high-quality images. In addition, when processing images from groups of 4 and 5 pixels, 7 and 12 bits must be added for the forward DWT, and the same values should be used for the inverse DWT.

Fig. 5. Source image for wavelet processing

Tab. 1. PSNR results of inverse DWT images processed by the proposed Winograd method

Method	DWT	Bits for the fractional part of a binary number	PSNR, dB
F(2,4)	Inverse	2	37.0871
	Inverse	3	46.6113
F(3,4)	Inverse	4	25.1008
	Inverse	5	31.1231
	Inverse	6	37.1138
	Inverse	7	43.0627
F(4,4)	Forward and inverse	3 and 3	7.3319
	Forward and inverse	4 and 4	23.8625
	Forward and inverse	5 and 5	24.6229
	Forward and inverse	6 and 6	25.0959
	Forward and inverse	7 and 7	42.7271
F(5,4)	Forward and inverse	6 and 6	12.3971
	Forward and inverse	7 and 7	14.7865
	Forward and inverse	8 and 8	21.8067
	Forward and inverse	9 and 9	24.3237
	Forward and inverse	10 and 10	27.6388
	Forward and inverse	11 and 11	35.2019
	Forward and inverse	12 and 12	40.1445

4. Hardware modeling and discussion

Hardware modelling was performed using OpenLane hardware environment [23]. The Yosys library is used for synthesizing and optimizing logic circuits from hardware description languages in OpenLane. Yosys is an open-source Verilog synthesis tool. It converts Verilog circuit descriptions into various formats for further processing in the design of microchips, FPGAs and ASICs. The modified modeling parameters are presented in Tab. 2. The remaining parameters are set to default values. Proprietary codes were used to implement state-of-the-art and proposed methods, as it was necessary to conduct simulations under identical conditions for the sake of experimental purity. The characteristic “Performance” defined by the number of pixels processed per second is used to analyze the proposed method based on group pixel processing. This parameter is calculated as the ratio between the computational delay and the number of pixels processed. This metric shows the average delay spent on processing 1 pixel, since other methods (direct and lifting scheme) process 1 pixel at a time, while the WM processes several pixels (2 or more) at a time. The results are shown in Tables 3 – 4 and figs. (6) – (13).

Hardware simulation results show that for one iteration of the proposed forward DWT significantly increases hardware costs, ranging from 3.83 to 55.62 times, depending on the number of pixels processed. Additionally, power consumption rises from 7.85 times to 3250.69 times, as the number of processed pixels increases. The device delay also experiences an increase, ranging from 35.56 % to 154.13 %. Despite these drawbacks, the performance of the proposed method improves considerably, from 47.54 % to 109.07 %.

The proposed method for single iteration of inverse DWT significantly increases hardware costs, depending on the number of pixels processed, from 3.20 to 57.24 times. In addition, power consumption increases with the number of pixels, escalating from 6.49 to 2432.85 times. The device's delay also increases, with a rise between 13.11 % and 130.68 %. However, the performance of the proposed method improves, ranging from 76.82 % to 144.43 %.

The proposed approach increases hardware and power costs as the number of processed pixel fragments increases. Computational redundancy is reduced in the inverse DWT based on the proposed method by processing pixel groups consisting of an even number. The values of even pixels in such an image are zero when the sampling frequency increases. The more pixels an image fragment contains, the higher the hardware and power costs. Analysis of the simulation results for processing 5 pixels shows that its performance is inferior to that of the forward DWT for processing 3 pixels and the inverse DWT for processing 4 pixels. In addition, the hardware costs for processing 5 pixels are significantly higher. Therefore, further increasing the number of pixel groups for DWT is not practical. Optimal performance is achieved with WM F(3,4,2) for forward DWT and F(4,4) for inverse DWT.

One of the main advantages of the proposed method is the simultaneous processing of multiple pixels. Calculations are performed in parallel, unlike the lifting scheme. This increases the performance of calculations. High computational complexity is the main disadvantage of the proposed approach, as the number of operations increases. The more pixels are selected for processing, the more calculations will be performed. This will affect both hardware and power consumption. The proposed DWT method is effective in devices where computing speed is a key parameter. Modification of the proposed approach is a promising area for further research, which will reduce hardware costs and power consumption while maintaining the same level of computing speed.

Tab. 2. Parameters for hardware modeling of wavelet processing

Parameter	**Value**
CLOCK_TREE_SYNTH	False
CLOCK_PORT	Null
PL_TARGET_DENSITY	0.6
FP_CORE_UTIL	20

Tab. 3. Forward DWT hardware modeling results of state-of-the-art and proposed methods

Method / **Parameter**	**State-of-the-art methods**		**Proposed methods**			
	Direct	Lifting scheme	F(2,4,2)	F(3,4,2)	F(4,4,2)	F(5,4,2)
Area, μm^2	3,475	2,268	8,696	17,252	50,979	126,156
Delay, ns	7.27	6.30	8.54	9.04	13.21	16.01
Performance, Mpixels / s	137.552	158.730	234.192	331.858	302.801	312.305
Power, mW	0.655	0.363	2.850	7.030	155.000	1,180.000

Tab. 4. Inverse DWT hardware modeling results of state-of-the-art and proposed methods

Method / **Parameter**	**State-of-the-art methods**		**Proposed methods**			
	Direct	Lifting scheme	F(2,4)	F(3,4)	F(4,4)	F(5,4)
Area, μm^2	6,115	3,120	9,996	33,425	45,661	178,575
Delay, ns	8.07	7.40	8.37	11.87	12.11	17.02
Performance, Mpixels / s	123.916	135.135	238.949	252.738	330.306	293.772
Power, mW	1.250	0.485	3.150	47.300	60.600	1,180.000

Conclusion

This paper proposes an approach to forward and inverse DWT of images using WM for JPEG XS standard. Calculations were done in fixed–point format for high-quality processing image. The hardware simulation results showed that the high performance of forward DWT was shown by the proposed WM for processing 3 pixels and inverse DWT for processing 4 pixels. The proposed approach showed a performance increase of 109.07 % for forward DWT, but hardware and power consumption increased by 7.61 times and 19.37 times, respectively. The proposed WM showed a performance increase of 144.43 % for inverse DWT, but increased hardware and power consumption by 14.63 times and 124.95 times, respectively. The proposed approaches are suggested to be used in devices where speed is a critical parameter, for example, in medical or satellite image processing. A promising direction for further research is the development of devices for image processing by the proposed DWT method, operating in real time.

Acknowledgements

Research is section 3 was funded by the Russian Science Foundation (Project No. 24-71-10016). The rest of the paper was funded by the Russian Science Foundation (Project No. 23-71-10013).

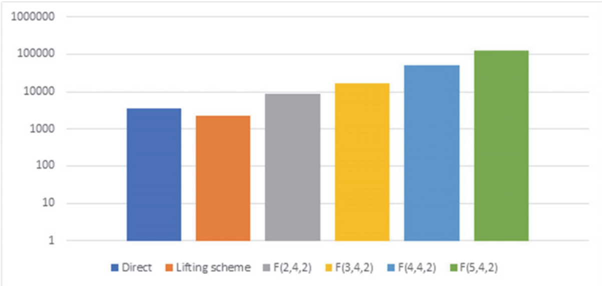

Fig. 6. Results of forward DWT modeling of hardware costs

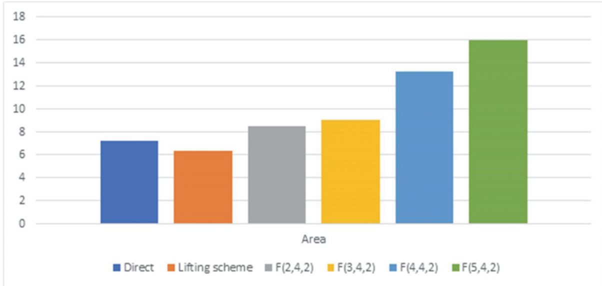

Fig. 7. Results of forward DWT modeling of computational delay

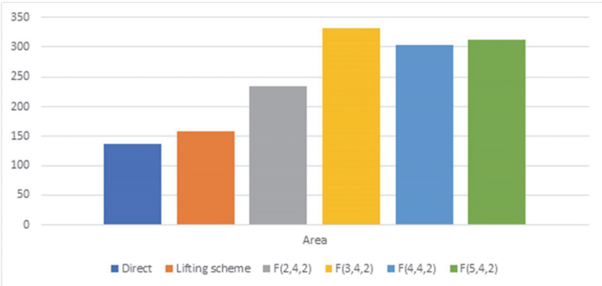

Fig. 8. Results of forward DWT modeling of image processing performance

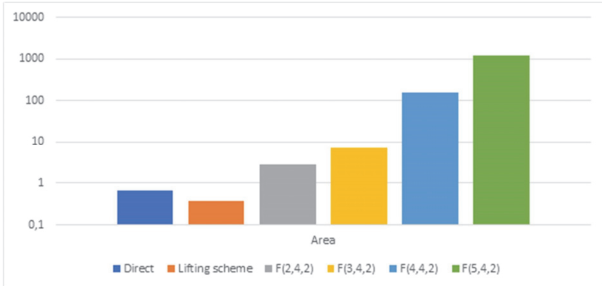

Fig. 9. Results of forward DWT modeling of power consumption

Fig. 10. Results of inverse DWT modeling of hardware costs

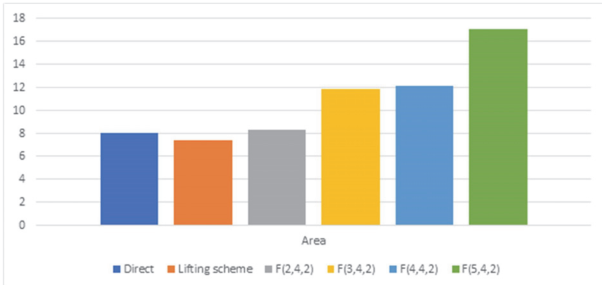

Fig. 11. Results of inverse DWT modeling of computational delay

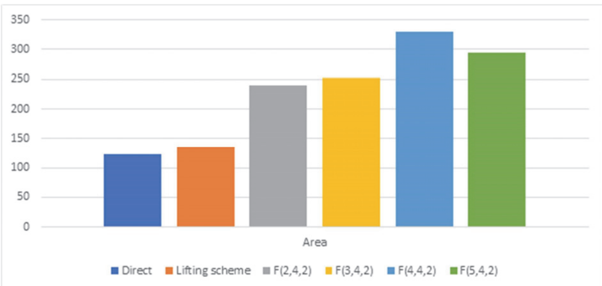

Fig. 12. Results of inverse DWT modeling of image processing performance

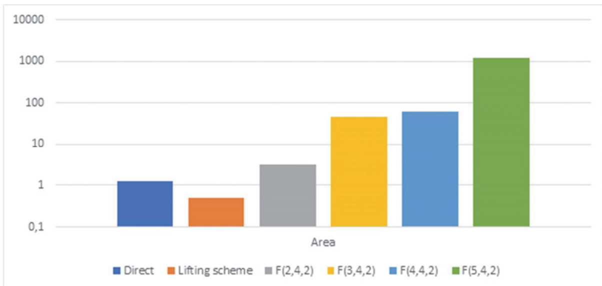

Fig. 13. Results of inverse DWT modeling of power consumption

References

[1] Gonzalez RC, Woods RE. Digital image processing. 4th ed. Harlow: Pearson Education; 2018.
[2] Rao KR, Domínguez HO, Subbarayappa S. JPEG series. 1st ed. Gistrup, Denmark: River Publishers; 2021. 346 p.
[3] Sai SV, Zinkevich AV, Fomina ES. Comparison of discrete cosine and wavelet transforms in RAW image compression systems. Computer Optics 2022; 46(6): 929-938. doi:10.18287/2412-6179-CO-1094.
[4] Descampe A, Richter T, Ebrahimi T, Foessel S, Keinert J, Bruylants T, Pellegrin P, Buysschaert C, Rouvroy G. JPEG XS – a new standard for visually lossless low-latency lightweight image coding. Proc IEEE 2021; 109(9): 1559-1577. doi:10.1109/JPROC.2021.3080916.
[5] Guo T, Zhang T, Lim E, López-Benítez M, Ma F, Yu L. A review of wavelet analysis and its applications: challenges and opportunities. IEEE Access 2022; 10: 58869-58903. doi:10.1109/ACCESS.2022.3179517.

[6] Álvarez-Cortés S, Serra-Sagristà J, Bartrina-Rapesta J, Marcellin MW. Regression wavelet analysis for near-lossless remote sensing data compression. IEEE Trans Geosci Remote Sens 2020; 58(2): 790-798. doi:10.1109/TGRS.2019.2940553.

[7] Meyer A, Brand F, Kaup A. Learned wavelet video coding using motion compensated temporal filtering. IEEE Access 2023; 11: 113390-113401. doi:10.1109/ACCESS.2023.3295512.

[8] Sahoo GR, Freed JH, Srivastava M. Optimal wavelet selection for signal denoising. IEEE Access 2024; 12: 45369-45380. doi: 10.1109/access.2024.3377664

[9] Stokfiszewski K, Wieloch K, Yatsymirskyy M. An efficient implementation of one-dimensional discrete wavelet transform algorithms for GPU architectures. J Supercomput 2022; 78: 11539-11563. doi:10.1007/s11227-021-04023-3.

[10] Richter T, Keinert J, Descampe A, Rouvroy G. Entropy coding and entropy coding improvements of JPEG XS. In: 2018 Data Compression Conference (DCC); Snowbird, UT, USA; 2018. pp. 87-96. doi:10.1109/DCC.2018.00018.

[11] Tian S, Song Q, He J, Zhang W, Wang Y, Lin Y. Implementation of JPEG XS entropy encoding and decoding on FPGA. J Real-Time Image Process 2024; 21(3-4): 1-17. doi:10.1007/s11554-023-01388-4. ← (уточнить том/номер по финальной версии)

[12] Foessel S, Richter T. Coding techniques in JPEG XS for efficient video production and contribution. In: Proceedings of the 2nd Mile-High Video Conference (MHV'23); 2023. pp. 72-77.

[13] Le Gall D, Tabatabai A. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In: Proc IEEE Int Conf Acoust Speech Signal Process (ICASSP-88); 1988; vol. 2. pp. 761-764.

[14] Sweldens W. The lifting scheme: a construction of second generation wavelets. SIAM J Math Anal 1997; 29(2): 511-546. doi:10.1137/S0036141095289051.

[15] Winograd S. Arithmetic complexity of computations. 1st ed. Philadelphia, PA: SIAM; 1980. 96 p.

[16] Anand R, Reddy A, Babu G, et al. Winograd transform-based fast detection of heart disease using ECG signals and chest X-ray images. IEEE Access 2025; 13: 57119-57140. doi:10.1109/ACCESS.2025.3555518.

[17] Kai G, Ximing Y, Yu P, Liansheng L. A Winograd-based CNN accelerator for cloud detection on FPGA. In: 2022 China Automation Congress (CAC); Xiamen, China; 2022. pp. 3966-3970. doi:10.1109/CAC57257.2022.10054855.

[18] Bao C, Xie T, Feng W, Chang L, Yu C. A power-efficient optimizing framework FPGA accelerator based on Winograd for YOLO. IEEE Access 2020; 8: 94307-94317. doi:10.1109/ACCESS.2020.2995330.

[19] Wang J, Hou J, Zhang F, Meng Y, Su Y, Yang C. An efficient and parallelism-scalable large integer multiplier architecture using least-positive form and Winograd fast algorithm. IEEE Trans Circuits Syst II Express Briefs 2025; 72(1): 248-252. doi:10.1109/TCSII.2024.3457494.

[20] Lyakhov P, Abdulsalyamova A, Semyonova N, Nagornov N. On the computational complexity of 2D filtering by Winograd method. In: 11th Mediterranean Conference on Embedded Computing (MECO); Budva, Montenegro; 2022. pp. 1-4. doi:10.1109/MECO55406.2022.9797173.

[21] Lyakhov PA, Nagornov NN, Semyonova NF, Abdulsalyamova AS. Development of digital image processing algorithms based on the Winograd method in general form and analysis of their computational complexity. Computer Optics 2023; 47(1): 68-78. doi:10.18287/2412-6179-CO-1146. [web:114]

[22] Rao KR, Yip PC. The transform and data compression handbook. 1st ed. Boca Raton, FL: CRC Press; 2001. 399 p.

[23] The OpenROAD Project. OpenLane. 2020. Available at: https://github.com/The-OpenROAD-Project/OpenLane.

Appendix A

Tab. A. GK coefficients for performing forward DWT of image fragment using Winograd method

Method	$\mathrm{G}K^T_{LD,odd}$	$\mathrm{G}K^T_{LD,even}$	$\mathrm{G}K^T_{HD,odd}$	$\mathrm{G}K^T_{HD,even}$
$F(2,4,2)$	$\frac{1}{4};1;\frac{3}{4}$	$\frac{3}{4};1;\frac{1}{4}$	$\frac{1}{4};-\frac{1}{2};-\frac{3}{4}$	$\frac{3}{4};-\frac{1}{2};-\frac{1}{4}$
$F(3,4,2)$	$\frac{1}{4};\frac{1}{2};-\frac{1}{4};\frac{3}{4}$	$\frac{3}{4};\frac{1}{2};\frac{1}{4};\frac{1}{4}$	$\frac{1}{4};-\frac{1}{4};\frac{1}{2};-\frac{3}{4}$	$\frac{3}{4};\frac{1}{4};\frac{1}{2};-\frac{1}{4}$
$F(4,4,2)$	$\frac{1}{8};\frac{1}{2};-\frac{1}{12};\frac{7}{24};\frac{3}{4}$	$\frac{3}{8};\frac{1}{2};\frac{1}{12};\frac{5}{24};\frac{1}{4}$	$\frac{1}{8};-\frac{1}{4};\frac{1}{6};-\frac{5}{24};-\frac{3}{4}$	$\frac{3}{8};\frac{1}{4};\frac{1}{6};\frac{1}{24};-\frac{1}{4}$
$F(5,4,2)$	$\frac{1}{16};\frac{1}{6};-\frac{1}{12};\frac{7}{96};-\frac{5}{96};\frac{3}{4}$	$\frac{3}{16};\frac{1}{6};\frac{1}{12};\frac{5}{96};\frac{1}{96};\frac{1}{4}$	$\frac{1}{16};-\frac{1}{12};\frac{1}{6};-\frac{5}{96};\frac{7}{96};-\frac{3}{4}$	$\frac{3}{16};\frac{1}{12};\frac{1}{6};\frac{1}{96};\frac{5}{96};\frac{3}{4}$

Appendix B

Tab. B. GK coefficients for performing inverse DWT of image fragment using Winograd method

Method	$\mathrm{G}K^T_{LR}$	$\mathrm{G}K^T_{HR}$
$F(2,4)$	$-\frac{1}{8};\frac{1}{2};0;\frac{3}{8};-\frac{1}{4}$	$\frac{1}{8};0;\frac{1}{3};-\frac{1}{24};-\frac{1}{4}$
$F(3,4)$	$-\frac{1}{16};\frac{1}{6};0;\frac{3}{32};\frac{13}{96};-\frac{1}{4}$	$\frac{1}{16};0;\frac{1}{3};-\frac{1}{96};\frac{9}{32};-\frac{1}{4}$
$F(4,4)$	$-\frac{1}{64};\frac{1}{18};0;\frac{3}{64};\frac{13}{576};-\frac{1}{576};-\frac{1}{4}$	$\frac{1}{64};0;\frac{1}{15};-\frac{1}{192};\frac{3}{64};-\frac{3}{320};-\frac{1}{4}$
$F(5,4)$	$-\frac{1}{256};\frac{1}{90};0;\frac{1}{128};\frac{13}{1152};-\frac{1}{4608};\frac{11}{2560};-\frac{1}{4}$	$\frac{1}{256};0;\frac{1}{45};-\frac{1}{1152};\frac{3}{128};-\frac{3}{2560};\frac{25}{4608};-\frac{1}{4}$

Author's information

Pavel Alekseyevich Lyakhov (b. 1988) graduated from Stavropol State University, specialty "Mathematics" in 2009. Head of Department of Mathematical Modeling, North-Caucasus Federal University. Head of Department of Modular Computing and Artificial Intelligence, North-Caucasus Center for Mathematical Research, North-Caucasus Federal University. Research interests are digital signal and image processing, artificial intelligence, neural networks, modular arithmetic, parallel computing, high-performance computing, digital circuits and hardware accelerators.
E-mail: *ljahov@mail.ru*

Maxim Valer'evich Bergerman (b. 1997) in 2021 graduated from the Master's program in "Applied Mathematics and Informatics" at North Caucasus Federal University. Currently, he is a PhD of Computer Science. Since 2021, he has been serving as a junior researcher at the North-Caucasus Center for Mathematical Research, a regional scientific and educational institution affiliated with the Faculty of Mathematics and Computer Science named after N.I. Chervyakov North-Caucasus Federal University. Research interests are digital signal and image processing, high-performance computing, modular arithmetic, digital circuits and hardware accelerators. E-mail: *maxx07051997@inbox.ru*

Nikolay Nikolaevich Nagornov (b. 1992) graduated from North-Caucasus Federal University, specialty "Applied Mathematics and Computer Science" in 2014. PhD of Computer Sciences. Associate professor of Mathematical Modeling department, North-Caucasus Federal University. Research interests are digital image processing, modular arithmetic, parallel computing, high-performance computing, digital circuits and hardware accelerators. E-mail: *sparta1392@mail.ru*

Albina Shihaevna Abdulsalyamova (b. 2000) in 2024 graduated from the Master's program in "Applied Mathematics and Informatics" at North-Caucasus Federal University. Research interests are digital image processing, high-performance computing, digital circuits and hardware accelerators. E-mail: *a.abdulsalyamova@mail.ru*

Received April 30, 2025. The final version – September 22, 2025.