UDC 004.315, 004.92, 519.6 **doi:** https://doi.org/10.20998/2522-9052.2024.2.05

Andrey Ivashko[1], Andrey Zuev[1], Dmytro Karaman[1], Miha Moškon[2]

[1] National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine
[2] University of Ljubljana, Ljubljana, Slovenia

FPGA-BASED IMPLEMENTATION OF A GAUSSIAN SMOOTHING FILTER WITH POWERS-OF-TWO COEFFICIENTS

Abstract. **The purpose of the study** is to develop methods for synthesizing a Gaussian filter that ensures simplified hardware and software implementation, particularly filters with powers-of-two coefficients. Such filters can provide effective denoising of images, including landscape maps, both natural and synthetically generated. The study also involves analyzing of methods for FPGA implementation, comparing their hardware complexity, performance, and noise reduction with traditional Gaussian filters. **Results.** An algorithm for rounding filter coefficients to powers of two, providing optimal approximation of the constructed filter to the original, is presented, along with examples of developed filters. Topics covered include FPGA implementation, based on the Xilinx Artix-7 FPGA. Filter structures, testing methods, simulation results, and verification of the scheme are discussed. Examples of the technological placement of the implemented scheme on the FPGA chip are provided. Comparative evaluations of FPGA resources and performance for proposed and traditional Gaussian filters are carried out. Digital modeling of the filters and noise reduction estimates for noisy images of the terrain surface are presented. The developed algorithm provides approximation of Gaussian filter coefficients as powers of two for a given window size and maximum number of bits with a relative error of no more than 0.18. Implementing the proposed filters on FPGA results in a hardware costs reduction with comparable performance. Computer simulation show that Gaussian filters both traditional and proposed effectively suppress additive white noise in images. Proposed filters improve the signal-to-noise ratio within 5-10 dB and practically match the filtering quality of traditional Gaussian filters.

Keywords: Gaussian filter, noise filtering; powers-of-two coefficients; hardware implementation; quality and performance estimation; simulation training complex; computer-generated environment; procedural landscape generation enhancement.

Introduction

The recent progress in science and technology has significantly influenced the emergence of new types of specialized vehicles. Their complexity has increased, along with the requirements for the quality of their operation. This necessitates a qualitative improvement in the training of operators for such equipment. The most modern means of training operators are simulation training complexes (STCs), which use computer-generated environments during the learning process. They are free from the drawbacks of traditional training methods, have a relatively low cost, both in terms of the complex development and its maintenance, are safe for humans, and allow the modeling of any situations that may arise during the operation of the real vehicle.

At the same time, the most essential component of any STC is the visual aspect of the learning process [1, 2]. This aspect allows conveying the principles of operating a vehicle to the future operator, enabling an appropriate and quick response to various situations that may arise during the operation of the vehicle. It reflects the results of the operator's actions, facilitating control over the learning process by the instructor. The main part of the visualization system involves modeling a plausible and realistic landscape, which should allow for physical simulation of movement and accurately simulate the surrounding environment, including lighting and weather conditions in different natural-climatic zones.

In the creation of the landscape model, it is crucial to require minimal resources and take up a small amount of time. This becomes particularly relevant given the requirements for the size of the landscape in training complexes (exceeding 16-25 square kilometers, determined by exercise areas). Therefore, it is expedient to use various methods of automated landscape synthesis, such as [1, 3].

Many automated algorithms for procedural landscape generation [4, 5] synthesize landscapes in a combined manner, using existing height maps of real landscapes as a foundation, which allows for obtaining a realistic model [6, 7]. However, existing sets of landscape height maps (Digital Elevation Models, DEM), such as SRTM and ASTER, have insufficient resolution in the horizontal plane and exhibit a significant level of noise. This necessitates additional filtering of the original height map data before its use in landscape synthesis.

Images, during their formation, storage, and transmission, are typically subject to various random disturbances or noise. The most common type of noise is random additive noise, which is statistically independent of the source image. The additive noise model is employed when the output signal from an imaging system can be considered as the sum of the useful signal and some random signal (noise). The additive noise model effectively describes the effects of film grain, photochemical defects, thermal noise in sensors, charge transfer noise, analog-to-digital converter quantization noise, video signal amplifier noise, dirt, dust on the sensor, and so on.

Visually, noise in an image appears as randomly positioned raster elements (dots) with sizes close to the pixel size.

The noise differs from the image with a lighter or darker shade.

Some common types of noise [8, 9] include:

- Gaussian noise – intensity variation following a normal distribution;
- "Salt and Pepper" – random isolated black or white points in the image;

- Speckle noise – a type of multiplicative noise that visually produces a "grainy" appearance in the image, and it is caused by the overlay of coherent waves, energetic interference when reflecting rays in the equipment.

To improve the quality of a noisy image (cleaning from noise), a wide range of smoothing filters is applied [10], including moving average, Gaussian, median, binomial and bilateral filters [11].

Gaussian filtering

A filter that is relatively simple to implement yet provides satisfactory results for solving many practical tasks is the Gaussian filter [12]. Gaussian filtering is applied both directly for noise suppression and as a preliminary step in tasks such as edge detection, for example, in the Canny method [13], highlighting ridges and valleys [14, 15].

Gaussian filtering is carried out through convolution with a kernel of size $N \times N$ (where N is odd), derived from surface formed by rotating of the curve of the normal distribution (Gaussian distribution) around the vertical axis and described by the expression:

$$G_{ij} = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(M-i)^2+(M-j)^2}{\sigma^2}}, \quad i,j = \overrightarrow{1,N}, \qquad (1)$$

where $M = (N+1)/2$ and σ – the standard deviation of the normal distribution, determining the scale of the transformation.

Since the elements of the Gaussian matrix calculated according to (1) are real numbers, rounding errors inevitably occur when smoothing the image by computing the convolution. On the other hand, dealing with real numbers in floating point format slows down calculations in the software and complicates the implementation in hardware.

The traditional method to address this problem is to approximate the elements of the matrix and replace them with integers. For example, a known approach is the use of a two-dimensional smoothing filter with integer coefficients in the popular Canny algorithm, used for edge detection in images [16, 17]. The kernel of such a filter with dimensions 5×5 has the form:

$$G^{Canny} = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}.$$

A series of integer approximations of Gaussian matrices are known [18, 19].

For dimensions of 3×3, 5×5 and 7×7 the most commonly used approximations are:

$${}^3G^{int} = \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix};$$

$${}^5G^{int} = \frac{1}{273} \cdot \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix};$$

$${}^7G^{int} = \frac{1}{1003} \times$$

$$\times \begin{bmatrix} 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 3 & 13 & 22 & 13 & 3 & 0 \\ 1 & 13 & 59 & 97 & 59 & 13 & 1 \\ 2 & 22 & 97 & 159 & 97 & 22 & 2 \\ 1 & 13 & 59 & 97 & 59 & 13 & 1 \\ 0 & 3 & 13 & 22 & 13 & 3 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \end{bmatrix}.$$

In [20], integer approximations of Gaussian smoothing filters of higher dimensions are also considered. In [21], an alternative approach is proposed, replacing Gaussian filters with the filtration of several consecutively connected box filters (averaging image elements within a two-dimensional window). There are also proposals to construct smoothing filters from binomial coefficients (elements of Pascal's triangle) [22, 23]. The drawbacks of these methods include insufficiently accurate approximation of the Gaussian filter and, more importantly, the need for a significant number of computationally intensive multiplication operations.

Gaussian filters with powers-of-two coefficients

Significant simplification of the filter hardware and software implementation can be achieved by approximating the filter kernel elements with powers of two. In this case, computationally intensive multiplication operations are replaced with shift operations. Some publications [20, 24] consider approximations of Gaussian filters using powers of two as coefficients. However, these works do not provide a methodology for filter synthesis and lack a quantitative estimation of the reduction in software and hardware complexity.

An algorithm has been developed to find the optimal power-of-two approximation of the Gaussian filter kernel based on the mean squared error criterion. The algorithm includes the following steps:

1. Specify the maximum power of two exponent n and compute $P = 2^n$, which should not exceed the maximum element of the kernel matrix (central element).

2. Using formula (1), calculate the Gaussian matrix ${}^NG$ of the given size N. Determine the central element of the matrix $C = {}^NG_{(N+1)/2,(N+1)/2}$.

3. Calculate the values $k_{min} = \lceil 0.75 \cdot P/C \rceil$ and $k_{max} = \lfloor 1.5 \cdot P/C \rfloor$. These values define the range in which the relationship $[k \cdot C] = 2^n$ holds. Here, brackets $\lceil\ \rceil$, $\lfloor\ \rfloor$, $[\ ]$ denote rounding to the nearest greater, nearest smaller, and nearest integer, respectively.

4. Multiply by k and round to the nearest integer: $G^{p2} = [k \cdot G]$. The resulting integer matrix is then divided by the normalization coefficient

$$s = \sum_{i=1}^{N} \sum_{j=1}^{N} G_{ij}^{p2}.$$

5. For each of the obtained matrices, calculate the degree of deviation d of the original Gaussian kernel matrix from the rounded-to-powers-of-two matrix

$$d = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (G_{ij} - G_{ij}^{p2})^2}{\sum_{i=1}^{N} \sum_{j=1}^{N} (G_{ij})^2}}.$$

6. Choose the rounded matrix for which the deviation d is minimal, and consider it as the optimal one.

According to the proposed algorithm, a series of filters were calculated, and the maximum deviation of a filter with rounded coefficients from the original did not exceed 0.18.

We provide examples of the obtained filter matrices with coefficients in the form of powers of two, Gaussian filters ${}^{N,n}G^{p2}$ with powers-of-two coefficients (GFP2), for various values of dimension N and maximum power n. The value of σ, the standard deviation of the normal distribution in formula (1), is chosen as the mean squared deviation $\sigma = (N-1)/4$.

$$ {}^{3,3}G^{p2} = \frac{1}{12}\cdot\begin{bmatrix} 0 & 1 & 0 \\ 1 & 8 & 1 \\ 0 & 1 & 0 \end{bmatrix}; $$

$$ {}^{3,6}G^{p2} = \frac{1}{12}\cdot\begin{bmatrix} 1 & 8 & 1 \\ 8 & 64 & 8 \\ 1 & 8 & 1 \end{bmatrix}; $$

$$ {}^{5,3}G^{p2} = \frac{1}{44}\cdot\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 4 & 8 & 4 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}; $$

$$ {}^{5,6}G^{p2} = \frac{1}{324}\cdot\begin{bmatrix} 1 & 4 & 8 & 4 & 1 \\ 4 & 16 & 32 & 16 & 4 \\ 8 & 32 & 64 & 32 & 8 \\ 4 & 16 & 32 & 16 & 4 \\ 1 & 4 & 8 & 4 & 1 \end{bmatrix}; $$

$$ {}^{7,3}G^{p2} = \frac{1}{148}\cdot\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 4 & 4 & 2 & 1 \\ 1 & 4 & 8 & 8 & 8 & 4 & 1 \\ 1 & 4 & 8 & 8 & 8 & 4 & 1 \\ 1 & 4 & 8 & 8 & 8 & 4 & 1 \\ 1 & 2 & 4 & 4 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}; $$

$$ {}^{7,6}G^{p2} = \frac{1}{1156}\cdot\begin{bmatrix} 1 & 4 & 8 & 8 & 8 & 4 & 1 \\ 4 & 16 & 32 & 32 & 32 & 16 & 4 \\ 8 & 32 & 64 & 64 & 64 & 32 & 8 \\ 8 & 32 & 64 & 64 & 64 & 32 & 8 \\ 8 & 32 & 64 & 64 & 64 & 32 & 8 \\ 4 & 16 & 32 & 32 & 32 & 16 & 4 \\ 1 & 4 & 8 & 8 & 8 & 4 & 1 \end{bmatrix}; $$

$$ {}^{9,6}G^{p2} = \frac{1}{1636}\cdot\begin{bmatrix} 1 & 4 & 8 & 8 & 8 & 8 & 8 & 4 & 1 \\ 4 & 8 & 16 & 16 & 16 & 16 & 16 & 8 & 4 \\ 8 & 16 & 32 & 32 & 32 & 32 & 32 & 16 & 8 \\ 8 & 16 & 32 & 64 & 64 & 64 & 32 & 16 & 8 \\ 8 & 16 & 32 & 64 & 64 & 64 & 32 & 16 & 8 \\ 8 & 16 & 32 & 64 & 64 & 64 & 32 & 16 & 8 \\ 8 & 16 & 32 & 32 & 32 & 32 & 32 & 16 & 8 \\ 4 & 8 & 16 & 16 & 16 & 16 & 16 & 8 & 4 \\ 1 & 4 & 8 & 8 & 8 & 8 & 8 & 4 & 1 \end{bmatrix}. $$

It is easy to see that multiplying by the provided matrices requires only shifts and just one final multiplication during normalization.

FPGA implementation

There are a number of publications [25, 26] considering the problems of FPGA implementation of image denoising filters, including Gaussian filters [27, 28]. These works attempt to optimize the FPGA structure and estimate hardware costs and performance for various circuit implementation options. At the same time, these articles do not consider the features of filters with powers-of-two coefficients and the advantages that they provide.

For hardware implementation, a universal module was developed, the structural diagram of which is shown in Fig. 1.

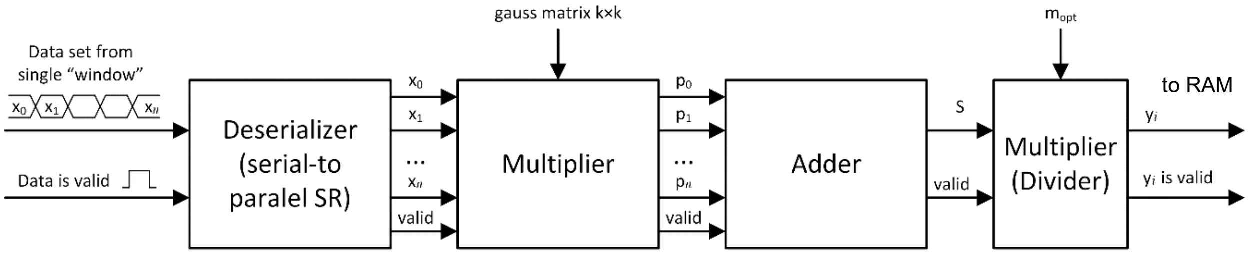


Fig. 1. Structural diagram of the implemented filtering module

The module consists of the following blocks: a deserializer (accumulative register-converter of sequentially incoming data), element-wise matrix multiplier, adder, and arithmetic divider.

The deserializer receives a packet consisting of sequentially incoming data from the current window (byte blocks) and places them in a register with parallel access to the outputs, so that the data of the entire window is simultaneously available for subsequent processing. An interface similar to AXI-Stream is used at the input, a popular bus interface used in embedded systems. The output data is accompanied by a data-ready signal (valid) on the block's outputs.

The multiplier performs element-wise multiplication of window data and constant coefficients from the Gaussian matrix of the corresponding size. The multiplication results in the form of a multi-bit binary vector are passed to the adder.

The adder performs arithmetic addition of all matrix elements.

The sum result is passed to the divider block, where arithmetic division by the coefficient before the matrix is performed.

The output of the scheme is the value from the 8 least significant bits of the division result.

The output values of the scheme, after processing each window's data, are stored in a specially allocated memory buffer and then, after processing all packets, form an array of filtered data.

The proposed scheme is scalable for any required size of the processed data block (window or packet) and the corresponding Gaussian matrix.

Functional verification and technological synthesis of the described scheme were performed for all variants of Gaussian matrices with integer coefficients of dimensions 3×3, 5×5, 7×7 and 9×9.

Example of the modeling and verification process for the scheme working with 3×3 matrices is shown in Fig. 2. The screenshot displays the timing diagrams of sequential packet transmission and processing.

The Xilinx Artix-7 FPGA chip (XC7A35Tftg256-1) was chosen as the target platform for implementation. An overall cost estimate based on technological synthesis results is provided in Table 1.

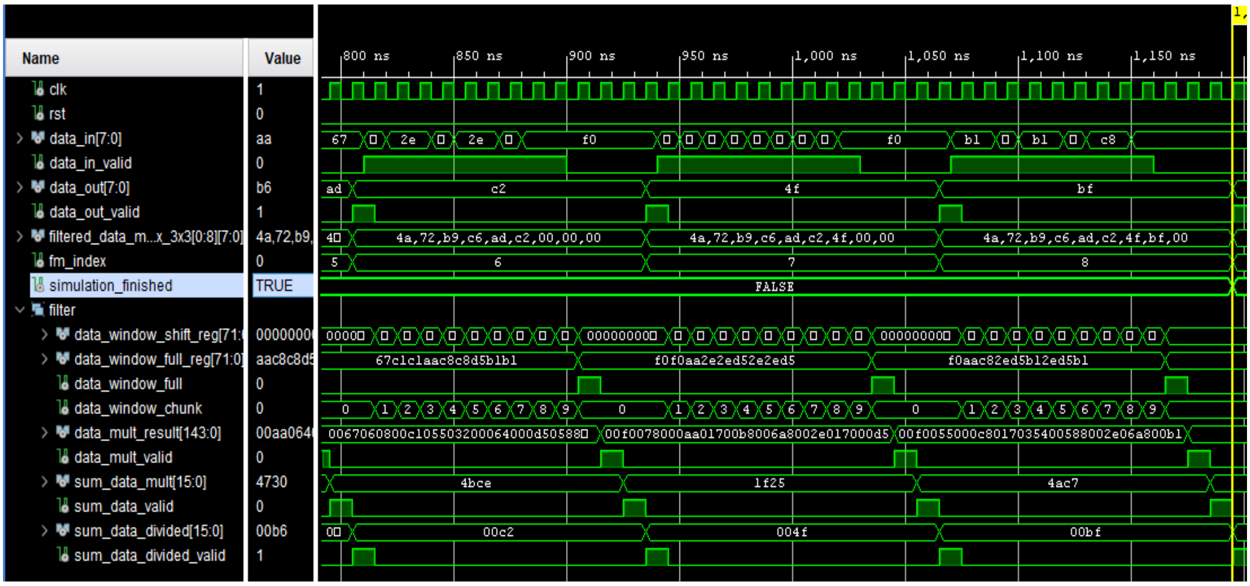


Fig. 2. Verification process for the scheme with 3×3 matrices

Table 1 – **Cost estimation for the synthesis of the filter for different variants of the Gaussian matrix**

Filter core Gaussian matrix	SLICE Units	LUT (Logic)	FF (Register)	Crit. Path Delay, ns
${}^{3}G^{int}$	70	127	240	6.207
${}^{3.3}G^{p2}$	47	86	147	6.097
${}^{3.6}G^{p2}$	96	236	248	9.820
${}^{5}G^{int}$	231	399	668	9.562
${}^{5.3}G^{p2}$	181	313	631	11.833
${}^{5.6}G^{p2}$	176	303	634	10.028
${}^{7}G^{int}$	360	899	1040	12.365
${}^{7.3}G^{p2}$	275	454	1114	13.317
${}^{7.6}G^{p2}$	324	501	1213	12.280
${}^{9}G^{int}$	737	1918	2220	13.375
${}^{9.3}G^{p2}$	610	1026	1703	12.753
${}^{9.6}G^{p2}$	579	1024	1990	12.602

The synthesis results indicate that in almost all cases, only the built-in resources of logic and configurable block registers were utilized, and no special FPGA resources (LUT as Memory, Block RAM, DSP Blocks) were engaged. Comparing the consumed resources, it is evident that filters based on Gaussian matrices with power-of-two coefficients are slightly more resource efficient compared to filters constructed with matrices having arbitrary integer coefficients. It can be concluded that as the size of the processed data window increases, along with the corresponding enlargement of matrices used in the filter, the resource savings become even more significant.

To assess performance, the duration of signal propagation along the longest combinational critical path in the implemented design can be considered. Static timing analysis showed that for all schemes using a 3×3 matrix, as well as for the scheme based on the ${}^{5}G^{int}$ matrix, the signal propagation delay along the critical path does not exceed 10 ns. This allows for the operation of filters with a clock frequency of 100 MHz. However, in schemes using filters based on matrices ${}^{5.3}G^{p2}$ and ${}^{5.6}G^{p2}$, the delay exceeds 10 ns, requiring a reduction in clock frequency. Testing the possibility of operating the technological implementation at a working frequency of 80 MHz (period of 12.5 ns) yielded positive results. For filters based on matrices of size 7×7 and 9×9, the signal propagation delays exceed 12.3 ns, which necessitates reducing the operating clock frequency. All considered variants of filters based on matrices of these dimensions can successfully operate at a frequency of 50 MHz (clock period of 20 ns).

The graphical representation of relative cost estimation for schemes based on various Gaussian matrices is shown in Fig. 3. The Fig. 4 shows an approximate view of the technological placement of the implemented filter scheme on the FPGA chip.

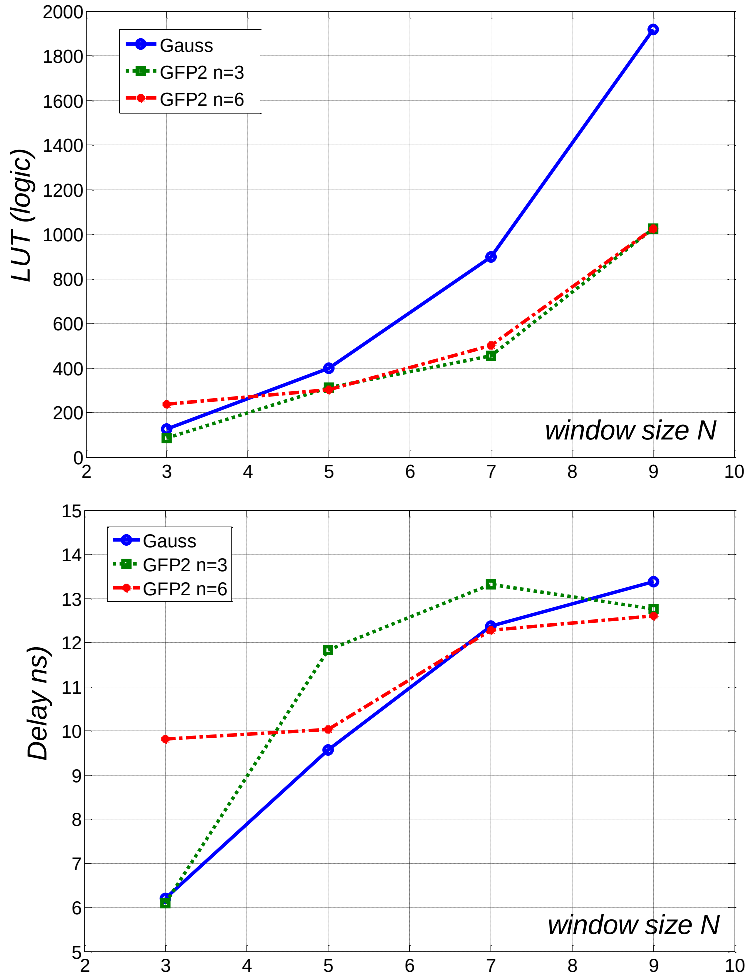
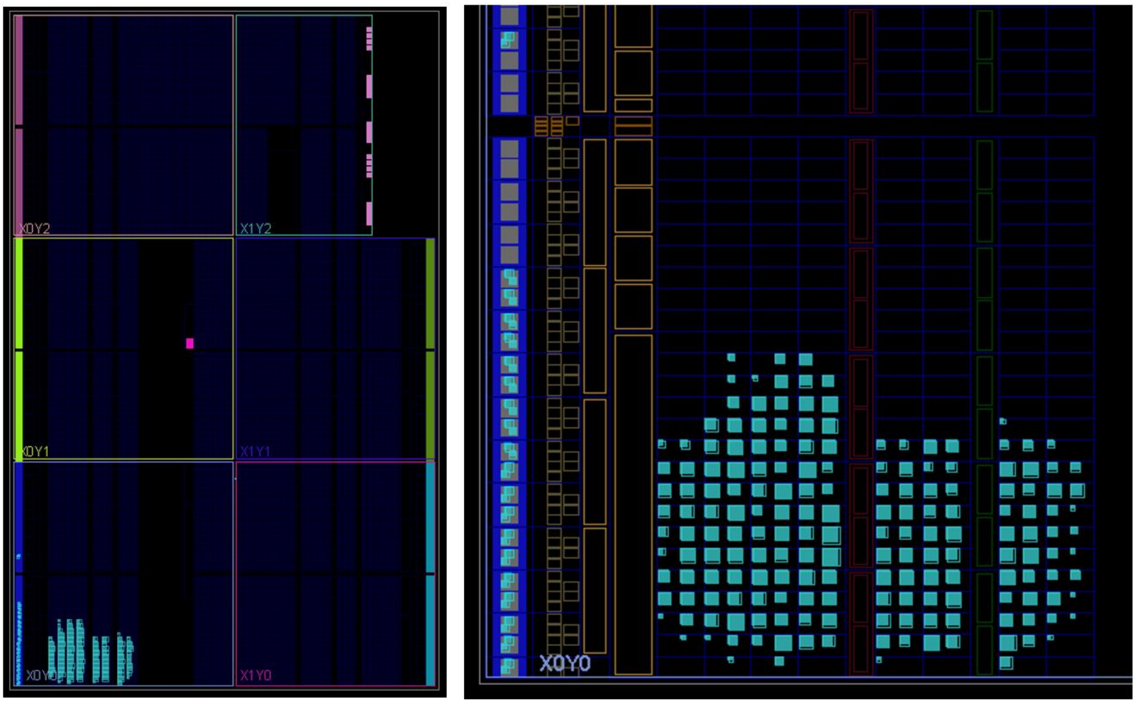


Fig. 3. Graphical representation of relative cost estimation according to Table 1:
Gauss – Gaussian filter implementation using approximated integer coefficients,
GFP2 – Gaussian filters $^{N,n}G^{p2}$ with powers-of-two coefficients, n – maximum exponent value



a *b*

Fig. 4. Technological placement of the filter scheme on the FPGA chip: *a* – overall view of the chip (the scheme occupies part of the resources in sector X0Y0, in the bottom left corner), *b* – enlarged image of the portion occupied by the scheme

Experimental results

To estimate the efficiency of the developed algorithms, a series of computational experiments were performed, and the results are presented in Figures 4 and 5. An elevation map of the terrain surface was chosen as the original image (higher areas were assigned a lighter color, lower ones a darker color).

The original image was subjected to the influence of white noise with a normal distribution, having a zero mean and a variance $D = 0.06$. Subsequently, noise filtering was performed using both traditional Gaussian filters with various apertures and Gaussian filters with coefficients rounded to powers of two.

Visual analysis of the images showed that processing with a Gaussian smoothing filter allows for a significant reduction in additive noise but leads to the smoothing of fine details in the images, especially with an increase of span N in the filtering window. Smoothing with a GFP2 filter with a central element of $2^6 = 64$ provides nearly identical results to traditional Gaussian filtering. When using a GFP2 filter with a central element of $2^3 = 8$, slightly less smoothing and a greater number of artifacts can be observed.

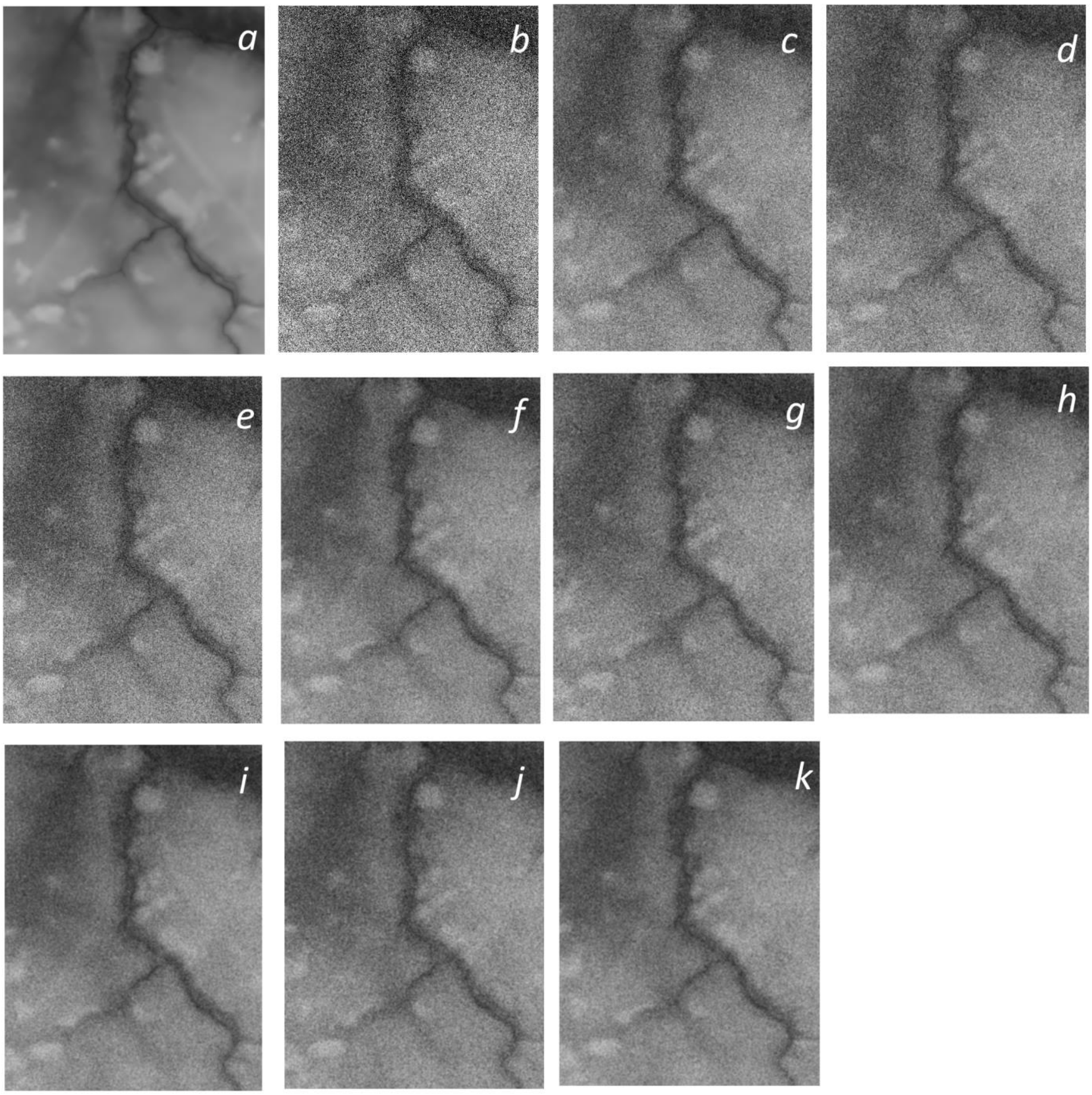


Fig 4. Terrain surface filtering results using Gaussian filters.
a – original image; *b* – noised image; c – Gaussian filter 5×5; *d* – GFP2 2^3 5×5;
e – GFP2 2^6 5×5; *f* – Gaussian filter 7×7; *g* – GFP2 2^3 7×7; *h* – GFP2 2^6 7×7;
i – Gaussian filter 9×9; *j* – GFP2 2^3 9×9; *k* – GFP2 2^6 9×9

The degree of noise suppression by both traditional and proposed filters was also quantitatively estimated.

The Peak Signal-to-Noise Ratio (PSNR) is commonly used for quantitative estimation. To determine PSNR, the Mean Squared Error (MSE) is calculated beforehand:

$$MSE = \frac{1}{K \cdot L} \cdot \sum_{i=1}^{K} \sum_{j=1}^{L} \left(\tilde{I}_{ij} - I_{ij}\right)^2,$$

where I_{ij} – original image of size $K \times L$ pixels, $\tilde{I}_{ij}$ – noisy image.

Then $PSNR = 10\log_{10}\frac{\max{(I_{ij})^2}}{MSE}$.

Sometimes, Mean Absolute Error (MAE) and the Structural Similarity Index Measure (SSIM) are also used as quality criteria for filtering.

Since in the article different types of filters are considered, it is advisable to compare the signal-to-noise ratio (or signal-to-noise ratio improvement) before and after filtering and use Signal-to-Noise Ratio Improvement Factor (SNRIF), as discussed, for example, in [29].

$$SNIRF = 10\log_{10}\frac{\sum_{i=1}^{K}\sum_{j=1}^{L}\left(\tilde{I}_{ij} - I_{ij}\right)^2}{\sum_{i=1}^{K}\sum_{j=1}^{L}\left(\hat{I}_{ij} - I_{ij}\right)^2},$$

where I_{ij} and $\tilde{I}_{ij}$ – original and noisy image subsequently, $\hat{I}_{ij}$ – filtered image.

Graphs were constructed showing the dependence of the averaged over 20 realizations of noise suppression coefficient of GFP2 on the noise level.

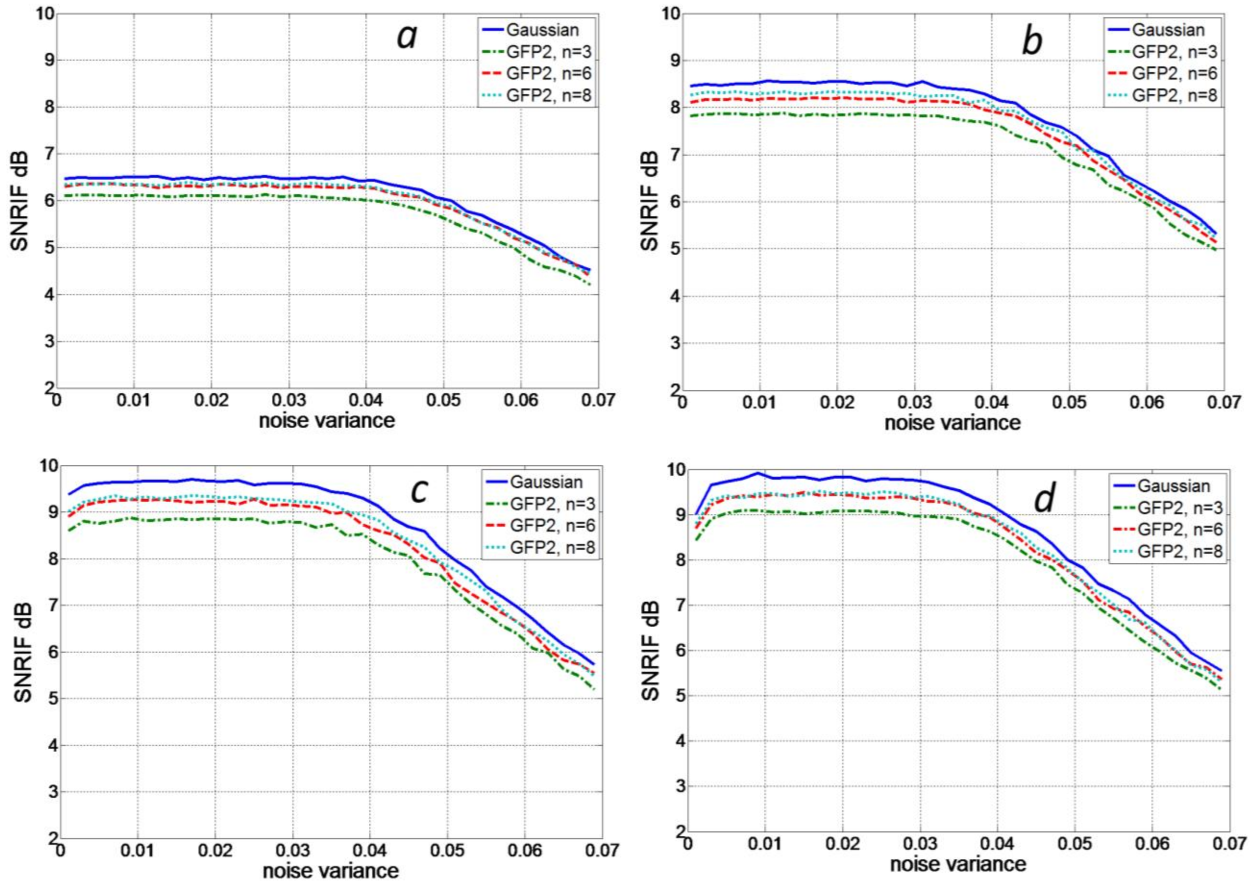


Fig 5. Performance comparison of Gaussian filters and GFP2:
a – window 5×5; *b* – window 7×7; c – window 9×9; *d* – window 11×11

The analysis of figures and graphs showed that the proposed filters in terms of filtration quality practically do not differ from traditional Gaussian filters (by no more than 0.5 dB). It is impractical to increase the maximum power of two beyond 6. Also, increasing the window size beyond 9 is impractical. It should be noted that when processing images with different-sized details, it may be more effective to use filters with different apertures and powers of two.

At the same time, as demonstrated earlier, the use of smoothing filters with coefficients in the form of powers of two significantly improves performance and reduces hardware complexity in FPGA implementations.

Conclusions

1. Gaussian smoothing filters effectively suppress additive white noise in images, including those of the terrain surface, both natural and artificially synthesized.

2. The developed algorithm allows for approximating the coefficients of the Gaussian filter with powers of two for a given window size and maximum power of two. The relative mean squared deviation of the synthesized filter from the original did not exceed 0.18.

3. Rounding the filter coefficients to powers of two significantly simplifies FPGA implementation of the filter and reduces hardware costs by approximately 30-50% to traditional Gaussian filters.

4. Digital modeling of the proposed filters confirmed their effectiveness for processing images of the terrain surface.

The filters provide an improvement in the signal-to-noise ratio within the range of 5-10 dB (depending on the noise level, window size, and maximum power of two) and exhibit a filtration quality within 0.5 dB compared to traditional Gaussian filters.

5. Further analysis of Gaussian filters with coefficients in the form of powers of two is advisable to choose optimal filter parameters based on the noise level, details' sizes in the image, and implementation characteristics.

REFERENCES

1. Smelik, R. M., Tutenel, T., de Kraker, K. J. and Bidarra R. (2010), "Declarative Terrain Modeling for Military Training Games", *International Journal of Computer Games Technology 2010*, doi: https://doi.org/10.1155/2010/360458
2. Merlac, V., Smatkov, S., Kuchuk, N. and Nechausov, A. (2018), "Resourses Distribution Method of University e-learning on the Hypercovergent platform", *Conf. Proc. of 2018 IEEE 9th Int. Conf. on Dependable Systems, Service and Technologies. DESSERT'2018*, Kyiv, May 24-27, 2018, pp. 136–140, doi: http://dx.doi.org/10.1109/DESSERT.2018.8409114
3. Genevaux, J.-D., Galin, E., Guérin, E., Peytavie, A. and Benes, B. (2013), "Terrain Generation Using Procedural Models Based on Hydrology", *ACM Transactions on Graphics*, vol 32(4), pp. 1–13, doi: https://doi.org/10.1145/2461912.2461996
4. Kachanov, P. and Zuev, A. (2014), "Automated synthesis of landscape surface model based on satellite images", *Eastern-European Journal of Enterprise Technologies*, vol. 6(3), pp. 10–15. doi: https://doi.org/10.15587/1729-4061.2014.30423
5. Zhou, H., Sun, J., Turk, G. and Rehg, J. M. (2007), "Terrain synthesis from digital elevation models", *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 834–848, doi: https://doi.org/10.1109/TVCG.2007.1027
6. Zuev, A., Karaman, D. and Yevsina, N. (2023), "Procedural Generation of Vegetation Cover for Simulators of Special Purpose Vehicles", *2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek)*, Kharkiv, Ukraine, 02-06 October 2023, pp. 606–611, doi: https://doi.org/10.1109/KhPIWeek61412.2023.10312930
7. Shmatko, O., Kolomiitsev, O., Rekova, N., Kuchuk, N. and Matvieiev, O. (2023), "Designing and evaluating dl-model for vulnerability detection in smart contracts", *Advanced Information Systems*, Vol. *7*(4), pp. 41–51, doi: https://doi.org/10.20998/2522-9052.2023.4.05
8. Gupta, D., Rathi, R. and Gupta S. (2018), "An Efficient Approach of Filtering for Noises on Images", *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, doi: https://doi.org/10.1109/CCAA.2018.8777553
9. Kuchuk, H., Kovalenko, A., Ibrahim, B.F. and Ruban, I. (2019), "Adaptive compression method for video information", International Journal of Advanced Trends in Computer Science and Engineering, 8(1), pp. 66–69, doi: http://dx.doi.org/10.30534/ijatcse/2019/1181.22019
10. Tania, S. and Rowaida R. (2016), "A Comparative Study of Various Image Filtering Techniques for Removing Various Noisy Pixels in Aerial Image", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 3, pp. 113–124, doi: https://doi.org/10.14257/ijsip.2016.9.3.10
11. Kamran, Ahmad, Jawad, Khan and Muhammad Salah Ud Din, Iqbal (2019), "A comparative study of Different Denoising Techniques in Digital Image Processing", *2019 8th International Conference on Modeling Simulation and Applied Optimization* (ICMSAO), doi: https://doi.org/10.1109/ICMSAO.2019.8880389
12. Volovyk, A. (2023), "Synthesis of quasi-optimal fast filters by the least square criterion", *Advanced Information Systems*, vol. 7, no. 2, pp. 21–27, 2023, doi: https://doi.org/10.20998/2522-9052.2023.2.04
13. Nattapong, Jundang and Surachai, Ongkittikul (2018), "Preserving Edge from Gaussian Filter by Edge Template", *2018 International Electrical Engineering Congress* (iEECON), doi: https://doi.org/10.1109/IEECON.2018.8712301
14. Zuev, A., Ivashko, D. and Lunin, D. (2023), "Accelerated Algorithm for Ridge and Valley Detection", *2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek)*, 02-06 October 2023, Kharkiv, Ukraine, p. 550–553. doi: https://doi.org/10.1109/KhPIWeek61412.2023.10312827
15. Magnier, B., Montesinos, P. and Diep, D. (2011), "Ridges and Valleys Detection in Images using Difference of Rotating Half Smoothing Filters", *ACIVS 2011: Advanced Concepts for Intelligent Vision Systems*, pp 261–272, doi: https://doi.org/10.1007/978-3-642-23687-7_24
16. Monteiro, M., Seidel, I. and Guntzel, J. L. (2023), "A Design Space Exploration of Power-efficient Gaussian Filter Architectures using Logical Optimization and Approximated Adders", *Journal of Integrated Circuits and Systems*, vol. 18, no. 02, pp. 1–12, doi: https://doi.org/10.29292/jics.v18i2.702
17. Khorbotly, S. and Hassan, F. (2011), "A modified approximation of 2D Gaussian smoothing filters for fixed-point platforms", *2011 IEEE 43rd Southeastern Symposium on System Theory*, pp. 154–159, doi: https://doi.org/10.1109/SSST.2011.5753797
18. Svyrydov, A., Kuchuk, H. and Tsiapa, O. (2018), "Improving efficienty of image recognition process: Approach and case study", Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018, pp. 593–597, doi: https://doi.org/10.1109/DESSERT.2018.8409201
19. Spagnolo, F., Corsonello, P., Frustaci, F. and Perri S. (2022), "Design of Approximate Bilateral Filters for Image Denoising on FPGAs", *IEEE Access*, vol. 11, pp.1990–2000, doi: https://doi.org/10.1109/ACCESS.2022.3233921
20. Pires, B. R., Singh, K. and Moura José M. F. (2011), "Approximating image filters with box filters", *2011 18th IEEE International Conference on Image Processing*, doi: https://doi.org/10.1109/ICIP.2011.6116693
21. Takahiro, Yano and Yoshimitsu, Kuroki (2016), "Fast implementation of Gaussian filter by parallel processing of binominal filter", *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, doi: https://doi.org/10.1109/ISPACS.2016.7824738
22. Kuchuk, G.A., Akimova, Yu.A., Klimenko, L.A. (2000), "Method of optimal allocation of relational tables", *Engineering Simulation*, 2000, 17(5), pp. 681–689.
23. Taha, Basheer, Riyadh Z., Mahmood and Enaam, Altaher (2021), "Efficient Implementation of Smoothing Filters Using FPGA", *2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic" (IEC)*, pp. 73–77, doi: https://doi.org/10.1109/IEC52205.2021.9476098
24. Di Carlo, S., Prinetto, P., Rolfo, D. and Trotta, P. (2013), "AIDI: An adaptive image denoising FPGA-based IP-core for real-time applications", *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Torino, IT, 24-27 June, pp. 99–106, doi: https://doi.org/10.1109/AHS.2013.6604232

25. Ahmed Ben Atitallah (2022), “An FPGA Design for Real-Time Image Denoising”, *Computer Systems Science and Engineering*, April 2022, vol. 43(2), pp. 803–816, doi: https://doi.org/10.32604/csse.2022.024393
26. Akkad, G., Ayoubi, R. and Abche, A. (2018), “Constant Time Hardware Architecture for a Gaussian Smoothing Filter”, *2018 International Conference on Signal Processing and Information Security* (ICSPIS), pp. 1–4, doi: https://doi.org/10.1109/CSPIS.2018.8642751
27. Cabello, F., Leon, J. and lana, Y. (2015), “Rangel Arthur. Implementation of a Fixed-Point 2D Gaussian Filter for Image Processing based on FPGA”, *2015 Signal Processing: Algorithms, Architectures, Arrangements, and Applications* (SPA), pp. 28–33, doi: https://doi.org/10.1109/SPA.2015.7365108
28. Dong, J., Hu, C., Chen, J., Hao, H. and Ma, Yu. (20110, “Edge Preserving Filtering by Combining Filters for Magnetic Resonance Image”, *2011 4th International Congress on Image and Signal Processing*, pp. 676–681, doi: https://doi.org/10.1109/CISP.2011.6100276

Received (Надійшла) 29.02.2023
Accepted for publication (Прийнята до друку) 24.04.2023

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Івашко Андрій Володимирович – кандидат технічних наук, доцент, доцент кафедри автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Andrey Ivashko – Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of automation and control in technical systems, National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine.
e-mail: andrii.ivashko@khpi.edu.ua; ORCID ID: https://orcid.org/0000-0002-4012-1697;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57220201505.

Зуєв Андрій Олександрович – кандидат технічних наук, доцент, завідувач кафедри автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Andrey Zuev – Candidate of Technical Sciences, Associate Professor, Head of the Department of automation and control in technical systems, National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine.
e-mail: andrii.zuiev@khpi.edu.ua; ORCID ID: http://orcid.org/0000-0001-8206-4304;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57207707035.

Караман Дмитро Григорович – старший викладач кафедри автоматики та управління в технічних системах, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна;
Dmytro Karaman –Senior Lecturer of the Department of automation and control in technical systems, National Technical University «Kharkiv Polytechnic Institute», Kharkiv, Ukraine.
e-mail: dmytro.karaman@khpi.edu.ua; ORCID ID: http://orcid.org/0000-0002-7252-3172;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57202228956.

Міха Мошкон – доктор філософії, доцент, факультет комп’ютерних та інформаційних наук, Університет Любляни, Любляна, Словенія;
Miha Moškon – PhD, associate professor, Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia.
e-mail: miha.moskon@fri.uni-lj.si; ORCID ID: http://orcid.org/0000-0003-4600-1730;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=24470299300.

Реалізація зглаждуючого фільтру Гауса з коефіцієнтами ступенів двох на основі FPGA

А. В. Івашко, А. О. Зуєв, Д. Г. Караман, М. Мошкон

Анотація. **Метою дослідження** є розробка методів синтезу гаусівського фільтра, які гарантують спрощену апаратну та програмну реалізацію, зокрема фільтрів з коефіцієнтами, що є степенями двійки. Такі фільтри можуть забезпечувати ефективне приглушення шумів на зображеннях, включаючи ландшафтні карти, як природні, так і синтетично створені. Дослідження також включає аналіз методів реалізації на FPGA, порівняння апаратної складності результатів їх роботи, продуктивності та приглушення шумів у порівнянні з традиційними гаусівськими фільтрами. Представлений алгоритм округлення коефіцієнтів фільтра до степенів двійки, що забезпечує оптимальне наближення побудованого фільтра до оригіналу, разом з прикладами розроблених фільтрів. Розглянуті теми включають практичну реалізацію на основі FPGA Xilinx Artix-7. Обговорюються структури фільтрів, методи тестування, результати симуляції та верифікація схеми. Надані приклади технологічного розміщення реалізованої схеми на чіпі FPGA. Проведені порівняльні оцінки ресурсів та продуктивності FPGA для запропонованих та традиційних гаусівських фільтрів. Представлені цифрові моделі фільтрів та оцінки зменшення шумів для зашумлених зображень поверхні місцевості. Розроблений алгоритм забезпечує наближення коефіцієнтів гаусівського фільтра у вигляді чисел ступеню двійки для заданого розміру вікна та максимальної кількості бітів з відносною похибкою не більше 0,18. Реалізація запропонованих фільтрів на FPGA призводить до зменшення витрат на апаратне забезпечення з порівняною продуктивністю. Комп'ютерне моделювання показує, що як традиційні, так і запропоновані гаусівські фільтри ефективно приглушують адитивний білий шум на зображеннях. Запропоновані фільтри покращують співвідношення сигнал/шум на 5-10 дБ та практично відповідають якості фільтрації традиційних гаусівських фільтрів.

Ключові слова: фільтр Гауса; фільтрація шуму; коефіцієнти ступенів двійки; апаратна реалізація; оцінка якості та продуктивності; імітаційний навчальний комплекс; комп'ютерно-генероване середовище; удосконалення процедурної генерації ландшафту.