UDC 004.932
https://doi.org/10.31891/CSIT-2022-1-6

SERHIY BALOVSYAK, SVITLANA VOROPAIEVA,
VALENTYNA HORDITSA, KHRYSTYNA ODAISKA, YULIYA TANASYUK
Yuriy Fedkovych Chernivtsi National University

SOFTWARE AND HARDWARE FOR DETERMINING GAUSSIAN NOISE LEVEL IN IMAGES

Accurate determination of the noise level in digital images is required to obtain their maximum signal-to-noise ratio, which is a necessary condition for the effective performance of the following stages of image processing: visualization, segmentation, recognition, etc. The task of calculating the Gaussian noise level is quite common, because such noise appears in most experimental images taken by video cameras. However, existing high-speed noise detection methods have a significant error, and the fairly accurate LLROI method has a low speed. The LLROI method is based on Low-frequency filtering of the noise component and Low-frequency filtering when selecting the Region Of Interest (ROI). Therefore, it is proposed to determine the level of Gaussian noise by the exact LLROI method and to increase its speed by appropriate hardware and software. Based on the LLROI method, a program in the MATLAB system was created, the structure and Simulink-model of a computer system for determining of Gaussian noise level on digital images were synthesized. Hardware implementation of image filtering units is made by FPGA Artix-7, which allowed us to increase the speed of the system. The results of calculating the Gaussian noise level for test images by the LLROI method using the developed hardware and software proved the errors not to exceed those provided by analogous methods.

The scientific novelty of the paper is to improve the LLROI method, namely to refine the threshold coefficient, which reduces the errors of calculating the noise level, even for images with clear contours and pronounced textures.

The practical significance of the developed tools is that they can be used to build high-speed computer systems (or subsystems) designed to increase the signal-to-noise ratio on digital images.

Key words: *Gaussian noise, Gaussian filter, image filtering, convolution, Matlab, Simulink model, FPGA Artix-7.*

СЕРГІЙ БАЛОВСЯК, СВІТЛАНА ВОРОПАЄВА,
ВАЛЕНТИНА ГОРДІЦА, ХРИСТИНА ОДАЙСЬКА, ЮЛІЯ ТАНАСЮК
Чернівецький національний університет імені Юрія Федьковича

ПРОГРАМНО-АПАРАТНІ ЗАСОБИ ВИЗНАЧЕННЯ РІВНЯ ГАУСОВОГО ШУМУ НА ЗОБРАЖЕННЯХ

Точне визначення рівня шуму на цифрових зображеннях потрібне для отримання їх максимального співвідношення сигнал/шум, що є необхідною умовою для ефективного виконання наступних етапів оброблення зображень: візуалізації, сегментації, розпізнавання, підготовки до якісного фотодруку та ін. Завдання обчислення рівня гаусового шуму є досить поширеним, оскільки на експериментальних зображеннях, отриманих за допомогою відеокамер, у більшості випадків присутній саме такий шум.

Проте, існуючі швидкодіючі методи визначення рівня шуму мають значну похибку, а досить точний метод LLROI (Low-pass & Low-pass filtration & Region Of Interest) має низьку швидкодію. Метод LLROI засновано на низькочастотній фільтрації шумової складової та низькочастотній фільтрації при виділенні ділянки інтересу зображення ROI (Region Of Interest). Характерною особливістю методу LLROI є врахування статистичних характеристик гаусового шуму при виділенні ділянок інтересу на зображеннях, на яких переважає такий шум. Тому в роботі запропоновано визначати рівень гаусового шуму точним методом LLROI, а його швидкодію підвищити за допомогою відповідних апаратно-програмних засобів.

На основі методу LLROI створено програму в системі MATLAB, синтезовано структуру та Simulink-модель комп'ютерної системи визначення рівня гаусового шуму на цифрових зображеннях, розроблено структурні схеми блоків системи. Апаратна реалізація блоків фільтрації зображень виконана засобами програмованої користувачем вентильної матриці (FPGA) Artix-7, що дозволило на порядок підвищити швидкодію роботи системи та виконувати оброблення не тільки окремих (статичних) зображень, але й кадрів відеопотоку в режимі реального часу. Результати обчислення рівня гаусового шуму для тестових зображень методом LLROI за допомогою розроблених апаратно-програмних засобів показати, що отримані похибки не перевищують похибок методів-аналогів.

Наукова новизна роботи полягає в удосконаленні методу LLROI, а саме в уточненні коефіцієнту порогу, що дозволяє зменшити похибки обчислення рівня шуму навіть для зображень з чіткими контурами та яскраво вираженими текстурами.

Практична значимість розроблених засобів полягає в тому, що на їх основі можлива побудова швидкодіючих комп'ютерних систем (або підсистем), призначених для підвищення співвідношення сигнал/шум на цифрових зображеннях.

Ключові слова: *гаусовий шум, фільтр Гауса, фільтрація зображень, згортка, Matlab, Simulink-модель, FPGA Artix-7.*

Introduction

Most experimental digital images have a certain level of Gaussian noise, which is described by its standard deviation (SD) σ_N [1-4]. Significant levels of Gaussian noise are observed in images obtained with video cameras in low light. An additional factor that causes noise is the size reduction of the photosensitive matrix elements of video cameras, as there is a tendency to increase the number of pixels of photosensitive matrices while maintaining the physical size of the matrices. Therefore, when filtering noise in the image, it is important to have an accurate estimate of the noise level, which allows maximizing the signal-to-noise ratio on the filtered image. This is a necessary condition for the effective implementation of the following stages of digital image processing: visualization, segmentation, recognition, preparation for high-quality photo printing, etc.

Related works

Currently, the basic methods used to assess the noise level on images are as follows:

- Method of histogram analysis of brightness distribution for the selected area of the image (Region Of Interest – ROI), where noise is mainly present [2].
- Filter-based approaches or a smoothing-based method [5-6], in which the image with noise is processed by high-frequency and low-frequency filters, and the noise level is calculated based on the filtered image.
- Patch-based or block-based methods [7], in which the set of homogeneous blocks (areas) is highlighted on the images, and the noise level is calculated by the principal component analysis (PCA).
- Statistical approaches [7] are based on image filtering and analysis of the value of excess for the distribution of image intensity depending on the noise level.
- Methods based on Fourier spectrum analyses that take into account the difference between the spectra of the useful signal and noise [2].
- Methods based on the wavelet transforms [2, 6] which involves the analysis of wavelet coefficients with a certain threshold.
- The LLROI (Low-pass & Low-pass filtration & Region Of Interest) method [8], which is based on low-frequency filtering of the noise component and low-frequency filtering when selecting the Region Of Interest (ROI) of the image.

However, the considered methods of determining the noise level (except LLROI) typically provide high speed, rather than high accuracy. At the same time, the high-precision LLROI noise calculation method has a rather low speed, which limits its scope of use.

Therefore, the **purpose** of this study is the development of the hardware and software for implementation of high-precision LLROI method of calculating the noise level, which would also provide high speed.

Determination of Gaussian noise level on the image by low-frequency filtering

The low-pass filtering method (or LLROI method) [8] is to calculate the Gaussian noise level on a digital image f_n (grayscale), which is obtained by video camera and transmitted to a computer system in the form of a rectangular matrix $f_n = (f_n(i, k))$, where $i = 1,..., M$, $k = 1,..., N$, M – image height (in pixels), N – image width. When being read from video camera the color image is transformed in shades of gray. The noise in numerous/various experimental images is rather accurately described with the Additive White Gaussian Noise (AWGN) model, so this paper considers image noise in the AWGN model or simplified in the Gaussian noise model [2].

The probability density distribution for Gaussian noise is described by the formula [2]

$$p_{DF}(z) = \frac{1}{\sqrt{2\pi}\sigma_N} exp\left(\frac{-(z-z_C)^2}{2\sigma_N^2}\right). \quad (1)$$

where z is the brightness of the image, z_C is the mathematical expectation of the distribution, σ_N is the standard deviation of noise, which in the case of AWGN is the noise level (theoretical).

In the LLROI method, the selection of the noise component of the image f_n is performed using low-pass filtering. Spatial low-pass filtering [2, 4] consists in the convolution of the image $f_n = (f_n(i, k))$ with the kernel (window) of the low-pass filter $w = (w(m, n))$ with size $M_w \times N_w$ elements by the formula

$$g(i,k) = \sum_{m=1}^{Mw} \sum_{n=1}^{Nw} f_n(i-m+m_c, k-n+m_c) \cdot w(m,n), \quad (2)$$

where $g = (g(i, k))$ is filtered image; $i = 1,..., M$, $k = 1,..., N$;
M_{w2}, N_{w2} are whole parts of half the size of the filter kernel;
$m_c = (M_{w2} + 1)$ is the center of the filter kernel in height;
$n_c = (N_{w2} + 1)$ is the center of the filter kernel in width.

The sum of the elements of the low-frequency kernel w is equal to 1. The operation of convolution the image f_n with the kernel w is simply written in the form $g = f_n * w$.

The two-dimensional Gaussian function (Gaussian filter) is used as the kernel of the filter w

$$w(m,n) = \frac{1}{\sigma_w\sqrt{2\pi}} exp\left(\frac{-((m-m_c)^2+(n-n_c)^2)}{2\sigma_w^2}\right), \quad (3)$$

where $\sigma_w = 1.75$ is SD of the Gaussian distribution; $m = 1,..., M_w$, $n = 1,..., N_w$;
m_c and n_c are the coordinates of the filter kernel center in height and width.

The dimensions of the filter kernel w are chosen taking into account the rule of 3σ for the normal distribution.

The value of the noise component f_h of the image is obtained by subtracting the smoothed image g from the initial f_n by the formula

$$f_h = f_n - g. \quad (4)$$

The local increase of the amplitude f_h corresponds to the contours and textures of the image noise component, so hence such local areas should be excluded from the ROI area to reduce the error of calculating the SD of noise. To detect areas of contours and textures, the image f_d is calculated first as the absolute value of f_h , while an average image of the noise level f_{ds} results from the convolution of the image f_d with the kernel of the Gaussian filter with SD of σ_w.

$$f_{dc} = f_d * w. \quad (5)$$

The pixel of the image f_h belongs to the ROI area only if the f_{dc} value of the corresponding pixel does not exceed the set threshold T_h. The affiliation of the pixels of the f_h image to the ROI area is written in the f_{ROI} matrix and determined using the T_h threshold of the ROI area according to this rule

$$f_{ROI}(i,k) = \begin{cases} 1, & f_{dc}(i,k) \le T_h \\ 0, & f_{dc}(i,k) > T_h \end{cases}, \quad (6)$$

where $i = 1,..., M$, $k= 1,..., N$; f_{dc} is the average noise level of an image.

The initial value of SD σ_h of noise is calculated for the whole image f_h. After calculating the ROI area, the values of σ_h are determined only taking into account those pixels of f_h that belong to the ROI. The iterative process of calculating ROI areas terminates if the difference between the previous and next values of σ_h becomes less than the $\Delta_{\sigma h}$ constant, e.g. $\Delta_{\sigma h} = 0.004$.

Taking into account the statistical characteristics of Gaussian noise, the T_h threshold [8], which determines the affiliation of the image pixel to the ROI (6), is calculated by the obtained formula

$$T_h = z_C + k_{\sigma Th} \cdot \sigma_{fdc} = \sqrt{\frac{2}{\pi}}\sigma_h + k_{\sigma Th} \cdot (0.162\sigma_h), \quad (7)$$

where z_C is the mathematical expectation of f_{dc}; σ_{fdc} is SD of f_{dc}; $k_{\sigma Th} = 1.22$ – threshold coefficient.

The experimental noise level σ_{NE} of the image f_n is calculated by the formula:

$$\sigma_{NE} = (\sigma_h)^{k_{\sigma h}}, \quad (8)$$

where $k_{\sigma h} = 1.018$ is the nonlinearity coefficient of σ_h.

In the LLROI method described above (1-8), the numerical value of the threshold coefficient $k_{\sigma Th}$ (7) was obtained by analysing a relatively small number of test images (6 samples). Therefore, in this paper, to reduce the error in calculating the experimental noise level σ_{NE} (8), the threshold coefficient $k_{\sigma Th}$ ($k_{\sigma Th} > 0$) was refined. One-parameter non-smooth optimization of $k_{\sigma Th}$ was performed by the iterative method of coordinate descent with increment of 0.01. As a target function of M_{SE} ($k_{\sigma Th}$) we used the standard error of the experimental noise SD σ_{NE} calculation for the series of q_I = 25 test images [9-10] (test set), to each of which Gaussian noise with theoretical SD σ_N was added programmatically.

$$M_{SE}(k_{\sigma Th}) = \frac{1}{q_N \cdot qI} \sum_{n_n}^{q_N} \sum_{n_i}^{qI} (\sigma_{NE}(n_i, n_n, k_{\sigma Th}) - \sigma_N(n_n))^2 , \quad (9)$$

where n_i is the image number; n_n is noise level σ_N number; $q_N = 20$ is the number of values σ_N; the values of σ_N ranging from 0.01 to 0.20 with steps of 0.01.

The largest error of calculating the noise level occurs for images with clear contours and proniunced textures, thus such images make a significant part of the test set. In addition, after the software addition of Gaussian noise (in the MATLAB system) to the test images, such images were saved in graphical files, rather than processed directly in the MATLAB system. The obtained digital images are saved in graphical files with 256 brightness levels, considering the quantization of their brightness. This resulted in the optimal value of the threshold coefficient $k_{\sigma Th} = 1.68$, which enabled us to specify the formula (7) for the threshold T_h as follows

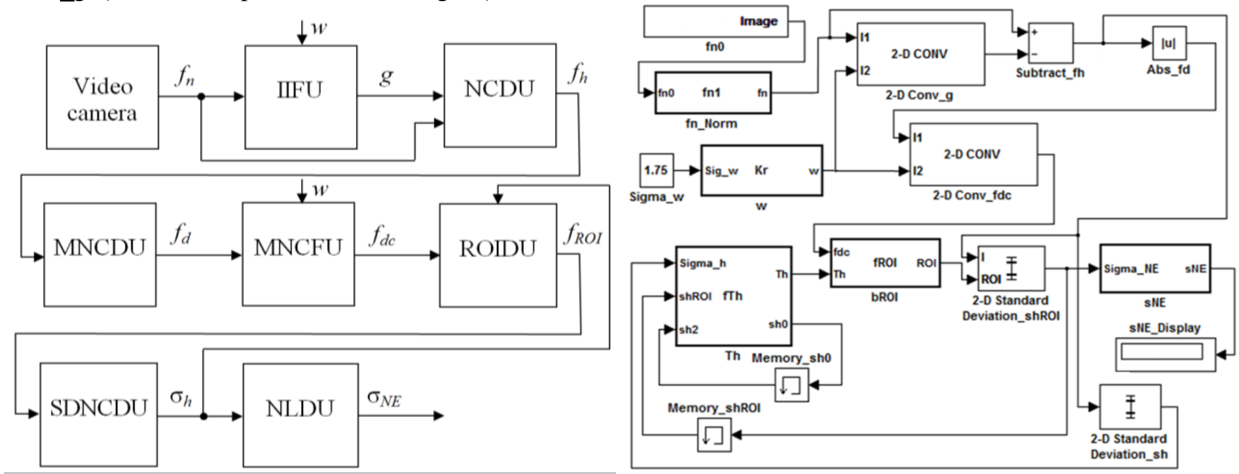
$$T_h = 0.798\ \sigma_h + 1.68 \cdot 0.162\ \sigma_h = 1.070\ \sigma_h. \quad (10)$$

Thus, the value of the threshold T_h (10) allows one to more accurately outline the ROI areas with predominant noise, even for images with textures.

Software and hardware for determining the noise level

Based on the considered LLROI method (1-10), which is designed to calculate the Gaussian noise level σ_{NE}, the appropriate software and hardware have been developed, in particular, the software "GaussNoise18" in the Matlab system [8, 11]. Based on the LLROI method, the structure of a computer system (CS) for determining the noise level on images using low-frequency filtering was also synthesized (Fig. 1a). The source of the initial image f_n for the computer system is a digital video camera, and the output is noise level σ_{NE}. According to the structure of the noise detection computer system, the Initial Image Filtering Unit (IIFU) performs convolution (2) of the image f_n with the low-pass filter kernel w (3), while the Noise Component Determining Unit (NCDU) calculates f_h by subtracting the matrices f_n and g (4). In the Modulus of Noise Component Determining Unit (MNCDU) the elements of the matrix f_d are obtained as the modules of the elements of the matrix f_h. The Modulus of Noise Component Filtering Unit (MNCFU) performs image convolution f_d with the kernel of the low-pass filter w (5). In the Region Of Interest Determining Unit (ROIDU) on the basis of the matrix f_{ds} and the threshold T_h (10) the image of sites of interest f_{ROI} (6) is obtained. The Standard Deviation of Noise Component Determining Unit (SDNCDU) calculates the value of SD σ_h of noise component based on f_h, taking into account the area of ROI. The noise level σ_{NE} (8) is calculated in the Noise Level Determining Unit (NLDU) through SD σ_h.

Based on the structure of the computer system to determine the level of Gaussian noise (Fig. 1a) its Simulink model (Fig. 1b) has been developed in MATLAB using Simulink and Blockset (Video and Image Processing). In the developed model, the initial image is read in the block "fn0", the convolution operations are implemented by blocks "2-D Conv", and the operation of calculating SD of the image is performed by blocks "2D Standard Deviation". The initial image f_{n0} is read from the graphical file, while when the block "From Multimedia File" is used as a signal source a series of images is read as frames of the video stream. The sequence of image processing in the proposed model is as follows. The block "fn_Norm" (with the MATLAB code) is designed to normalize the brightness of the original image f_{n0} in the range from 0 to 1, resulting in a normalized image f_n. The core of the low-pass filter w is calculated in block "w" (with the MATLAB code) based on the SD of filter "Sigma_w". The operation of convolution the image f_n with the kernel of the filter w (resulting in a filtered image g) is performed in the block "2-D Conv_g" (which corresponds to IIFU in Fig. 1a).



a) b)

Fig. 1. Structure (a) and Simulink model (b) of a computer system for determining the noise level in images using low-frequency filtering

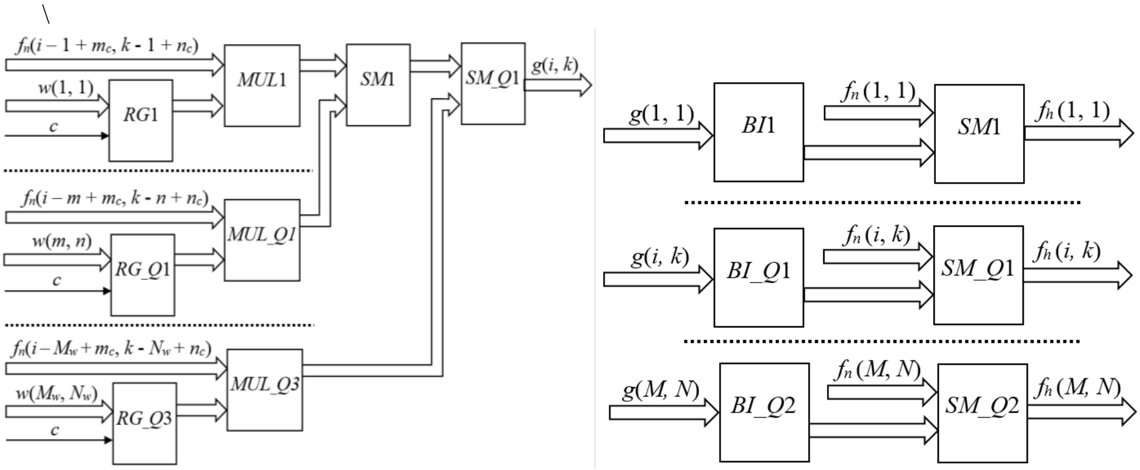
The noise component f_h is calculated as the difference of the g and f_n matrices in the block "Substract_fh" (which corresponds to NCDU). In the block "Abs_fd" (the analog of MNCDU) the modulus of the noise component f_d is determined. Viewer modules can be added to the model to visualize images during modeling. The operation of the image f_d, convolution resulting in the image f_{dc}, is performed in the block "2-D Conv_fdc" (the analog of MNCDU). The initial value of SD Sigma_h of noise component f_h is calculated in the block "2-D Standard Deviation_sh" (the analog of SDNCDU). The threshold T_h of the ROI area is determined in the block "Th" taking into account Sigma_h, and based on the threshold T_h and the image f_{dc} in the block "bROI" the values of the ROI area are calculated (blocks "Th" and "bROI" implement ROIDU in Fig. 1a). The standard deviation shROI of noise component f_h (taking into account the region of interest ROI) is calculated in the block "2-D Standard Deviation_shROI" (the analog of SDNCDU). The obtained SD shROI is specified in the cycle using the block "Th", in which the output sh0 denotes SD shROI for the previous iteration of the cycle. The initial values of the shROI (-1) and sh2 (1) signals are set in the "Memory_shROI" and "Memory_sh0" memory blocks, respectively. The ROI refinement is completed if in the block "Th" the difference between the values of shROI (new) and sh2 (previous) becomes less than the threshold $\Delta_{\sigma h}$ ($\Delta_{\sigma h} = 0.004$). The output signal σ_{NE} is calculated in the block "sNE" (the analog of NLDU). The simulation results (Gaussian noise level σ_{NE}), obtained using the developed Simulink model (Fig. 1b), are consistent with the results of data processing in MATLAB using the program "GaussNoise18".

Block diagrams of the main units of the CS to determine the noise level have been developed (Fig. 1a). The block diagram of the Initial Image Filtering Unit (IIFU) describes the operation of convolution of the image f_n with the filter core w (Fig. 2a). The values of each of the $M_w \times N_w$ elements w are written to the corresponding register RG. Multiplying the values of f_n with the values of the corresponding elements of the kernel w is performed in the operating units of multiplication MUL. The addition of the obtained multiplier is performed in the combination adders SM, and at the output of the last adder SM_Q1 the pixel brightness of the filtered image g is formed. Convolution of images can be implemented in hardware using "multiplier-adder" (MADD) operations, which are characterized by high speed.

The block diagram of the unit for determining the noise component of the NCDU (Fig. 2b) describes the operation of calculating the noise component f_h by subtracting the matrix g from the matrix f_n. The block diagram contains blocks of inverters $\hat{A}^2 1 \ldots BI\_Q2$, where $Q2 = M \times N$, and the same number of combination adders SM. The outputs of the blocks $\hat{A}^2$ form the inverse brightness values of the corresponding pixel of the image g, which are fed to the second information input of the adders SM, and the first information input of the adders SM receives the brightness values of the pixels f_n. The value of $f_h = (f_n - g)$ is formed at the output of the adders SM operating in the subtraction mode. Subtraction in the adders $SM1 \ldots SM\_Q2$ occurs in the complementary code, so its transfer inputs receive the level of logical "1".

The Modulus of Noise Component Determining Unit (MNCDU) describes the operation of determining the modulus f_d of the noise component f_h (Fig. 2b).

\



a) **b)**

Fig. 2. Block diagrams: a) Initial Image Filtering Unit (IIFU), c is a synchronization signal; b) Noise Component Determining Unit (NCDU)

MNCDU is constructed similarly to NCDU (Fig. 2b), but in MNCDU the sign of f_h is considered. Therefore, the output of adders SM, operating in subtraction mode, the values $f_d = f_h = (f_n - g)$ (if $f_h \geq 0$) or the value $f_d = |f_h| = |f_n - g|$ (if $f_h < 0$) are formed. The sign of f_h is determined by the signal level at the transfer output in the adders SM.

The block diagram of the Modulus of Noise Component Determining Unit (MNCDU), which performs the convolution of the image f_d with the kernel of the low-pass filter w, is built similarly to the block diagram of the IIFU (Fig. 2a).

The block diagram of the Region Of Interest Determining Unit (ROIDU) (Fig. 3a) describes the operation of calculating the f_{ROI} matrix based on f_{ds} and taking into account the threshold T_h. In the operational multiplication unit $MUL1$, the threshold T_h (10) is calculated. At the output of the inverter $\hat{A}^2 1$ the inverse values of T_h are formed, which are fed to the second information input of the adders SM, and the first information input SM receives the values of f_{ds}. At the output of the adders SM, that operating in the subtraction mode, the value of f_{ROI} is formed, which is equal to 0 (if $f_{dc} > T_h$) or 1 (if $f_{dc} \leq T_h$). The sign of $(f_{dc} - T_h)$ is determined by the signal level at the output of the transfer in the adders SM.

The block diagram of the Standard Deviation of Noise Component Determining Unit (SDNCDU) describes the operation of calculating the value of SD σ_h of noise component based on f_h and taking into account the image of the region of interest f_{ROI} (Fig. 3b). The values of the elements of the noise component f_h (read from the registers $RG1$-RG_Q2) are squared in the multiplication blocks $MUL1 - MUL\_Q2$, and then multiplied by the corresponding value of f_{ROI}. The addition of the obtained multipliers is performed in the combination adders SM, where the output of the adder SM_Q1 is formed by the sum of the f_{ROI} values. In the divisor $DIV1$ is the division of the output of the adder SM_Q3 to the output SM_Q1, and in the Block calculation of the square Root ($BR1$) the SD σ_h is calculated.

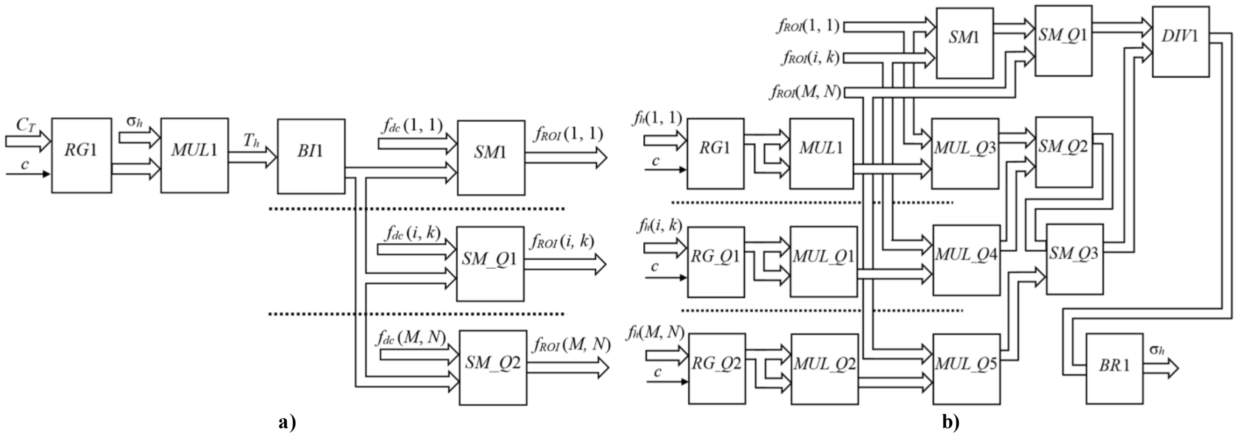


Fig. 3. Block diagrams: a) Region Of Interest Determining Unit (ROIDU), coefficient C_T = 1.070; b) Standard Deviation of Noise Component Determining Unit (SDNCDU), *c* is the synchronization signal

In the Noise Level Determining Unit (NLDU) the noise level σ_{NE} is calculated on the basis of SD σ_h through the power function (8).

The hardware implementation of the main units of the developed CS (Fig. 1) is made by means of Field-Programmable Gate Array (FPGA) Artix-7 (XC7A200T-1SBG484C) from Xilinx, which is designed for image processing [12]. Based on the developed Simulink model (Fig. 1b) it is possible to generate VHDL or Verilog code for FPGA (when using Simulink units supported by HDL encoder). However, the hardware implementation of the CS units is not based on Simulink models but is performed through modifying the existing open project "Gauss-filter-FPGA-for-video-processing" [13] for FPGA Artix-7 image processing in Verilog, since such a project takes into account the features of the FPGA when processing video streams.

FPGA programming [14] was performed by Vivado Design Suite 2019.2 [12] via Micro USB (JTAG) connector. FPGA implemented an Initial Image Filtering Unit (IIFU) (Fig. 1a) using the project "Gauss-filter-FPGA-for-video-processing" [13]. The hardware implements the image filtering unit, because software image filtering requires a lot of CPU time. The input signal from the USB video camera was fed through the USB host connector. The output signal (filtered images) was output through the HDMI (High Definition Multimedia Interface) connector. When implementing IIFU by means of FPGA, the actual filtering is performed only in the image filtering module, and other modules provide complex interaction with ports, RAM, peripherals, etc. The Artix-7 FPGA contains 33,650 logic blocks (each consisting of four 6-input look-up tables (LUT), 8 triggers and 3 programmable multiplexers (MUX)), 215 K logic cells, 740 DSP (digital signal processors) blocks. For Artix-7, the amount of RAM is about 13 Mbit, the clock frequency is 450 MHz. Such FPGA parameters are sufficient for the implementation of the blocks developed computer system of digital image processing.

The modification of the «Gauss-filter-FPGA-for-video-processing» project [13] mainly consisted of in calculating the filter kernel parameters *w* (3) according to the LLROI method. The calculated filter kernel (SD σ_w = 1.75) was described in Verilog, where the elements *w* /GAUSS_KERNEL/ used integers 24-bit numbers, with the sum of values of *w* was normalized to 2^{24}:

```
GAUSS_KERNEL <= {
24'd50347, 24'd113894, 24'd185873, 24'd218838, 24'd185873, 24'd113894, 24'd50347,
24'd113894, 24'd257649, 24'd420479, 24'd495050, 24'd420479, 24'd257649, 24'd113894
24'd185873, 24'd420479, 24'd686214, 24'd807914, 24'd686214, 24'd420479, 24'd185873
24'd218838, 24'd495050, 24'd807914, 24'd951199, 24'd807914, 24'd495050, 24'd218838
24'd185873, 24'd420479, 24'd686214, 24'd807914, 24'd686214, 24'd420479, 24'd185873
24'd113894, 24'd257649, 24'd420479, 24'd495050, 24'd420479, 24'd257649, 24'd113894
24'd50347, 24'd113894, 24'd185873, 24'd218838, 24'd185873, 24'd113894, 24'd50347};
```

Testing the developed tools for the noise level definition

The developed hardware and software tools for calculating experimental SD σ_{NE} of noise (based on the LLROI method), tested in a series of test images (Fig. 4, Fig. 5) with software added Gaussian noise, for which the value of SD σ_N was in the range from 0.01 to 0.20.

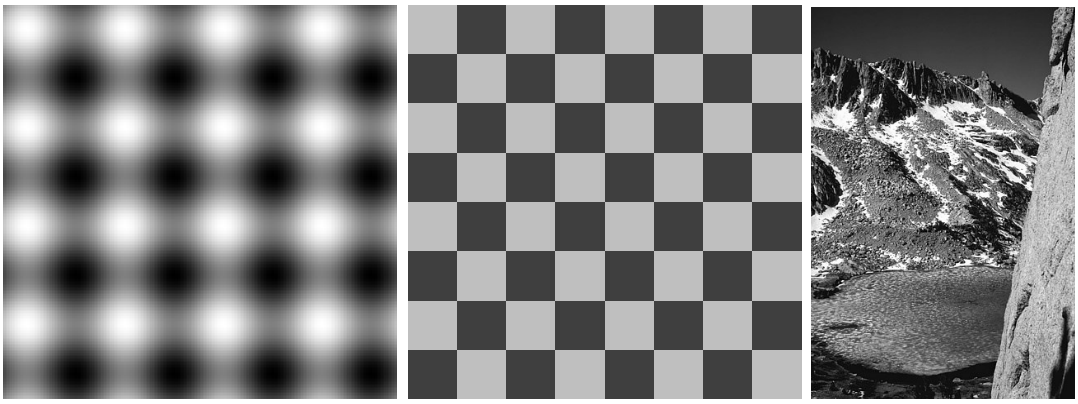
Initially, the calculation of the noise level was performed using the "GaussNoise18" developed program in the MATLAB system. For each theoretical noise level σ_N the Root Mean Square Error (RMSE) is obtained, which is calculated between the values of σ_{NE} and σ_N (for all images)

$$R_{MSE}(\sigma_N) = \sqrt{\frac{1}{q_I}\sum_{n_i=1}^{qI}(\sigma_{NE}(n_i) - \sigma_N)^2}, \quad (11)$$

where n_i is the image number; q_I is the number of images.

a) b) c)

Fig. 4. Test images *f*: a) image "face" with a small amount of details (size 300 × 300 pixels); b) image "cameraman" with an average number of details (256 × 256 pixels); c) image "crowd" with a large number of details (512 × 512 pixels) [15]



a) b) c)

Fig. 5. Test images *f*: a) image "sin2_64g" with a smooth change in brightness (size 256 × 256 pixels); b) image "ch64g" with clear outlines (256 × 256 pixels); c) mountain image with textures (321 × 481 pixels) [9]

The root mean square error R_{MSES} for all images and all q_N values of the theoretical noise level σ_N is calculated by the formula

$$R_{MSES} = \sqrt{\frac{1}{q_N}\sum_{n_n=1}^{q_N}(R_{MSE}(n_n))^2}, \tag{12}$$

where n_n is the number of the theoretical noise level σ_N.

The results of determining the experimental SD σ_{NE} of Gaussian noise for images with small (Fig. 4a), medium (Fig. 4b) and large (Fig. 4c) number of details in all cases showed high accuracy of calculating the noise level: RMSES error is in the range 0.0015-0.0017 (Table 1). Compared with the results of processing the considered images (Fig. 4), the accuracy of calculating the noise level for images "sin2_64g", "ch64g" and "mountain" (Fig. 5) is about twice lower (Table 1), due to the presence of contour images and textures. However, even for images with textures (Fig. 5c), the error of R_{MSES} (12) is insignificant (RMSES = 0.0026) and lower than the errors of analog methods [5-7]. This is due to the fact that the noise level was calculated only for homogeneous regions of interest (ROI), where there is mainly noise.

Table 1

Results of calculation of the experimental Gaussian noise level σ_{NE} for test images (Fig. 4, Fig. 5) with the theoretical noise level σ_N by the LLROI method

σ_N, 10^{-2}	σ_{NE}, 10^{-2}					
	face	cameraman	crowd	sin2_64g	ch64g	mountain
1	1.0786	1.1385	1.0963	0.9626	0.9255	0.9872
2	1.9840	2.0065	1.6361	1.9119	1.8822	1.9279
3	2.9341	2.9209	2.9157	2.8556	2.8180	2.8826
4	3.8830	3.8694	3.9256	3.8292	3.7386	3.7882
5	4.8581	3.8694	4.9802	4.7805	4.6267	4.7213
6	5.8391	5.8127	6.0489	5.7427	5.5906	5.7723
7	6.8160	6.7728	7.0686	6.7381	6.5372	6.6827
8	7.7786	6.7728	8.1548	7.7318	7.4367	7.5526
9	8.7848	8.8142	9.1676	8.6429	8.4441	8.5650
10	9.7928	9.7621	10.2235	9.7102	9.2096	9.7842
R_{MSES},10^{-2}	0.1505	0.1694	0.1588	0.2189	0.4154	0.2618

The calculation of the noise level on the test images was also performed using a modified program "GaussNoise18" and FPGA Artix-7, where the FPGA implemented the filtering unit of the original image, and other stages of processing were performed programmatically. FPGAs allow to filter video streams (512 × 512 pixels) at 24 frames per second, which is an order of magnitude faster than image processing with Matlab. Hardware implementation of all stages of the LLROI method using the Artix-7 FPGA will potentially further increase the speed of noise calculation.

Conclusions

1. The LLROI (Low-pass & Low-pass filtration & Region Of Interest) method, based on low-pass filtering of the noise component and low-frequency filtering when selecting the region of interest (ROI) of the image is described.

2. Based on the LLROI method, "GaussNoise18" software for calculating the Gaussian noise level has been developed in the MATLAB system, and the structure and Simulink model of a computer system for determining the noise level in images have been synthesized.

3. Block diagrams of the main units of the computer system for determining the noise level have been developed. The hardware implementation of the filtering unit of the developed computer system was made by FPGA Artix-7 from Xilinx, which enabled increasing the speed of the system and processing not only individual images but also streaming videos.

4. The results of calculating the Gaussian noise level for test images by the LLROI method using the developed hardware and software showed that the errors obtained do not exceed those of analogous methods. Therefore, the developed tools can be used to build high-speed computer systems designed for quasi-optimal Gaussian noise filtering in images.

The scientific novelty of the work is to improve the LLROI method, namely to refine the threshold coefficient, which reduces the errors in calculating the noise level, even for images with clear contours and pronounced textures.

References

1. Bovik A.L. The Essential Guide to Image Processing. Amsterdam : Elsevier Inc., 2009. 853 p.
2. Gonzalez R., Woods R., Eddins L. Digital Image Processing using MATLAB. New Jersey, USA : Prentice Hall, 2004. 609 p.
3. Krigg S. Computer Vision Metrics. Survey, Taxonomy, and Analysis. Basel, Switzerland : Springer, 2014. 498 p.
4. Russ J.C. The Image Processing. Handbook. Abingdon-on-Thames, UK : Taylor & Francis Group, 2011. 853 p.
5. Liu X., Tanaka M., Okutomi M. Single-Image Noise Level Estimation for Blind Denoising. *IEEE Transactions on Image Processing*. 2013. Vol. 22. No. 12. P. 5226-5237.
6. Pyatykh S., Hesser J., Zheng L., Image noise level estimation by principal component analysis. *IEEE Transaction on Image Processing*, 2013. Vol. 22. No. 2. P. 687-699.
7. Zoran D., Weiss Y. Scale invariance and noise in natural images. *IEEE 12th International Conference on Computer Vision*. 2009. P. 2209-2216.
8. Balovsyak S. V., Odaiska Kh. S. Automatic Determination of the Gaussian Noise Level on Digital Images by High-Pass Filtering for Regions of Interest. *Cybernetics and Systems Analysis*. 2018. Vol. 54. No. 4. P. 662-670.
9. The Berkeley Segmentation Dataset and Benchmark. BSDS300. URL: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds. (Accessed on: May 20, 2021).
10. Fowlkes C., Martin D., Malik J. Local Figure/ Ground Cues are Valid for Natural Images. *Journal of Vision*. 2007. Vol. 7(8). No. 2. P. 1-9.
11. Balovsyak S.V., Odaiska Kh.S. Hardware and Software Complex for Automatic Level Estimation and Removal of Gaussian Noise in Images. *Advances in Intelligent Systems and Computing*. 2019. Vol. 754. P. 144-154.
12. Nexys Video Artix-7 FPGA: Trainer Board for Multimedia Applications. URL: https://store.digilentinc.com/nexys-video-artix-7-fpga-trainer-board-for-multimedia-applications. (Accessed on: May 20, 2021).
13. Gauss-filter-FPGA-for-video-processing. Pipeline-architecture-gauss-filter. URL: https://github.com/Wirilila/gauss-filter-FPGA-for-video-processing. (Accessed on: May 21, 2018).
14. Rusyn B.P., Lutsyk A.A., Kosarevych R.Ya. Modified architecture of lossless image compression based on FPGA for on-Board devices with linear CCD. *Journal of Automation and Information Sciences*. 2019. Vol. 50. No. 2. P. 41-49.
15. Image Processing Place. Image Databases. URL: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm. (Accessed on: May 4, 2021).

Serhiy Balovsyak **Баловсяк Сергій**	Doctor of Engineering Sciences, Associate Professor of Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Ukraine, e-mail: s.balovsyak@chnu.edu.ua. https://orcid.org/0000-0002-3253-9006 Scopus Author ID: 6506889690 ResearcherID: AAK-7602-2021, https://scholar.google.com.ua/citations?user=pNps2xMAAAAJ&hl=uk	доктор технічних наук, доцент кафедри комп’ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна.
Svitlana Voropaieva **Світлана Воропаєва**	PhD, Assistant of Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Ukraine, e-mail: s.voropaieva@chnu.edu.ua. https://orcid.org/0000-0001-5687-7491	кандидат технічних наук, асистент кафедри комп’ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна.

Khrystyna Odaiska **Валентина Горділа**	PhD, Assistant of Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Ukraine, e-mail: k.odaiska@chnu.edu.ua. https://orcid.org/0000-0002-3167-1195 Scopus Author ID: 57202220212	асистент кафедри комп'ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна.
Valentyna Horditsa **Христина Одайська**	Assistant professor at Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Ukraine, e-mail: v.horditsa@chnu.edu.ua. https://orcid.org/0000-0003-4548-2536 Scopus Author ID: 57200179749	кандидат технічних наук, асистент кафедри комп'ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна.
Yuliya Tanasyuk **Юлія Танасюк**	PhD, Associate Professor of Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Ukraine, e-mail: y.tanasyuk@chnu.edu.ua. https://orcid.org/0000-0001-8650-0521 Scopus Author ID: 6506296466	кандидат фізико-математичних наук, доцент кафедри комп'ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна.