

Oleksandr I. Tymochko¹, Volodymyr Larin¹, Maksym Kolmykov¹,
Oleksander O. Timochko², Vladislava Pavlenko³

¹ Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, Ukraine

² Kreditech Holding, Hamburg, Germany

³ V. N. Karazin Kharkiv National University, Kharkiv, Ukraine

RESEARCH OF IMAGES FILTRATION METHODS IN COMPUTER SYSTEMS

Abstract. It is known that human eyes are less sensitive to color, than to their brightness. In the RGB color space, all three components are considered equally important, and they are usually stored with the same resolution. However, you can display a color image more efficiently, separating the brightness from color information and presenting it with a higher resolution than color. RGB space is well suited for computer graphics, because it uses these three components for color formation. However, RGB space is not very effective when it comes to real images. The fact is that to save the color of an image, you need to know and store all three components of the RGB, and if one of them is missing, it will greatly distort the visual image representation. Also, when processing images in RGB space, it is not always convenient to perform any pixel conversion, because, in this case, it will be necessary to list all three values of the RGB component and write back. This greatly reduces the performance of various image processing algorithms. For these and other reasons, many video standards use brightness and two signals that carry information about the red and blue components of the signal, as a color model other than RGB. The most famous among such spaces is YCbCr.

Keywords: image; component; algorithm; standard; computer systems.

Introduction

Problem statement. The rapid development of information technologies led to the use of unidirectional specific images transmission in computer systems. According to that, in order to improve the quality of computer systems in public institutions, active implementation of such systems is taking place [1-3]. The main difficulty in working with video are large volumes of transmitted information and sensitivity to delays in the video information transmission. Therefore, in order to eliminate the maximum redundancy amount in the formation of the video sequence, 3 types of frames are used: I, P and B which form a frame group [4-7]. For a typical low complexity video sequence, the weight of each P-frame in the stream is approximately three times smaller than the I-frame weight. However, taking into account the number of P-frames in the group, they make the main contribution to the total video data amount. Therefore, the possibility of upgrading coding methods for P-frames is considered on preliminary blocks' type identification with the subsequent formation of block code structures.

It has been analysed the images filtration methods in the computer systems and its drawbacks has found during processing video data.

Research publications. A lot of researchers have investigated the images filtration methods. A comparison of international standards for lossless still image compression was made in [8-11] and in this case they have thoroughly investigated the compression ratio of all the well-known compression methods available at that time. However, the comparison in [12-14] does not deal with analysis of the latest computer systems standard. Another issue is that consideration is only given to still images.

In [15], the compression ratio and execution time of the compression methods was investigated. In addition, they investigated the efficiency of compression methods

based only on textual data and still images. Unmoving images are very different from the dynamic images in a differential-represented frame scenario.

Another study on the comparison of compression methods was conducted in [16]. They applied many compression standards and some compression programs to different dynamic images. They have compared both the compression ratio and the execution time of the compression techniques. They have pointed out that the compression performance depends on the type of images and the implication of this is that these results cannot be directly applied to dynamic images in a differential-represented frame scenario because of the different types of obstacles.

To ensure the timely delivery of video information resources, it is necessary to take into account the high-speed capabilities of communication channels. To do this, data compression algorithms are used.

The research aims and objectives. The aim of the study is to investigate of images filtration methods in computer systems for reducing time of information processing taking into account the limitations of data processing.

Research bases

Previous Images Filtration. Selection of the bright component. It is known that a color image requires at least three numbers per pixel to accurately convey its color. The method chosen to represent the brightness and color is called the color space [17].

There are three most common color models:

- RGB (used in computer graphics);
- YCbCr (used in video systems);
- CMYK (used in the color press).

All color spaces can be listed from the RGB space that can be obtained from the camera or scanner. The red, green, and blue are the main components of the colors represent three dimensions of this space, as shown in Fig. 1.

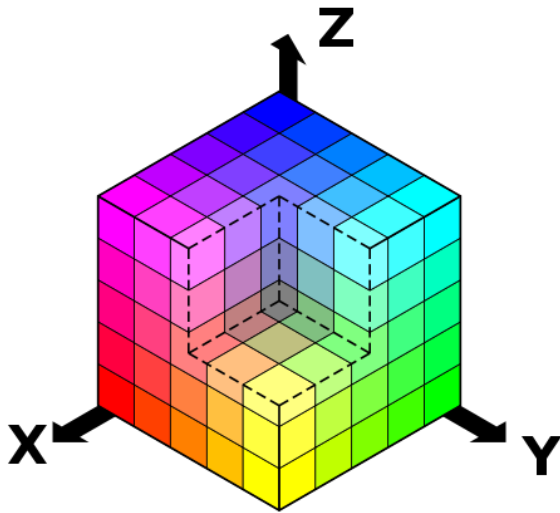


Fig. 1. RGB-colored model presented as a cube

RGB space is well suited for computer graphics, because it uses these three components for color formation [18]. However, RGB space is not very effective when it comes to real images. The fact is that to save the color of an image, you need to know and store all three components of the RGB, and if one of them is missing, it will greatly distort the visual image representation. Also, when processing images in RGB space, it is not always convenient to perform any pixel conversion, because, in this case, it will be necessary to list all three values of the RGB component and write back. This greatly reduces the performance of various image processing algorithms. For these and other reasons, many video standards use brightness and two signals that carry information about the red and blue components of the signal, as a color model other than RGB. The most famous among such spaces is YCbCr [19].

It is known that human eyes are less sensitive to color, than to their brightness. In the RGB color space, all three components are considered equally important, and they are usually stored with the same resolution. However, you can display a color image more efficiently, separating the brightness from color information and presenting it with a higher resolution than color.

The letter Y in such a color space denotes the component of brightness, which is calculated, as a weighted averaging of components R, G and B under the following formula:

$$Y = k_r R + k_g G + k_b B, \quad (1)$$

here is k - denotes the corresponding weight factor.

According to the recommendation of the ITU R BT.601 [20] the following ratios are proposed $k_r = 0,229$: and $k_b = 0,114$. The multiplier k_g is derived from the relation $k_r + k_g + k_b = 1$. Using these values, it is obtained a wide spread formula:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,144 \cdot B. \quad (2)$$

In order to process images in this work, the bright component of the YCbCr color space is used. Accordingly, in order to reach this, by using the formula

2 in the software Mathcad 15. Noises in images no system provides the perfect image quality for the object under research. Images during the process of forming their systems (photographic, holographic, and television) are usually exposed to various occasional interference or noise. A fundamental problem in the field of image processing is the effective noise removal, while preserving the image's parts which are important for further recognition [21, 22]. The complexity of the solution to this problem depends much on the noise's nature. In contrast to the deterministic distortions described by the functional transformations of the original image, additive models, pulse and multiplicative noise are used to describer and influences.

The most common type of interference is the random additive noise, which is independent from the signal. The additive noise model is used when the signal at the system's output or at some transformation stage. It can be considered as the sum of the useful signal and some random signal. The additive noise model describes well the effect of the film graininess. The fluctuation noise in the radioengineering systems, quantization noise in analog-to-digital converters, and the like.

Additive Gaussian noise is characterized by adding to each pixel images of values with normal distribution and with zero mean values. Such noise, usually, occurs during the digital image formation. Basic images information is contours of objects.

Classic linear filters can effectively eliminate, but the degree of small parts blurriness the image may exceed the permissible values.

Pulsed noise is characterized by the replacement of pixels' part in the image, with the values of a fixed or random variable. The automated digital image quality assessment carried out using the metric of objective image quality – Peak Signal to Noise Ratio (PSNR):

$$PSNR = 10 \cdot \log \left(255 / \sqrt{\sum_{i=1}^N (X_i - Y_i)^2 / N} \right), \quad (3)$$

here are: X_i – pixel of the image with which it is compared; Y_i – pixel of the image being compared; N – the number of image pixels.

In the image, such interferences look like isolated contrast points, an example of output and distorted impulse noise of an image with a density of 5% is shown in Fig. 2 and 3 respectively. Peak signal to noise ratio in this case: $PSNR = 9,213$. Pulse noise is typical of devices for inputting images from a television camera, image transfer systems via radio channels, as well as for digital imaging and transfer systems.

The use of linear filtration in this case is ineffective - each of the input pulses gives feedback in the form of a pulse filter's characteristic, and their aggregate contributes to the noise spread in the entire area of the image. To remove impulse noise, a special class of nonlinear filters based on rank statistics is used. The common idea of such filters is to detect the position of the pulse and replace it with the estimated value, while maintaining other pixels of the image unchanged.



Fig. 2. Output Image



Fig. 3. Image distorted by impulse noise

Median image filtering. A successful solution to solve an impulse noise is to use median filtration proposed by John Tuke [23] for the analysis of economic processes. It should be noticed that median filtration is a heuristic processing method, its algorithm is not a mathematical solution to a strictly formulated problem. Therefore, the researchers pay much attention to the analysis of the image effectiveness processing on its basis and comparison with other methods.

When applying a median filter, each image pixel is sequentially processed. For median filtration, a two-dimensional window (filter aperture) is used, usually has a central symmetry, with its center located at the current filtration point. The dimensions of the aperture are among the parameters that are optimized in the process of analyzing the algorithm efficiency. Image pixels, that appear within the window, form a working sample of the current step.

However, as discussed above, median filtering smoothens the image borders to a lesser degree than any linear filtering. The mechanism of this phenomenon is very simple and is as follows. Assume that the filter aperture is near the boundary separating the light and image's dark areas, with its center located in the dark area. Then, most likely, the work sample will contain more elements with small brightness values, and, consequently, the median will be among those elements of the work sample that match this area of the image. The situation changes to the opposite, if the aperture center is shifted to the region of higher brightness. But this means the presence of sensitivity in the median filter to brightness variations.

Filling the values operation of the median filter for the pixel value of the elementary object will correspond to the following expression:

$$y_k = x_{i,j}, \begin{cases} i = k - \lfloor k/m \rfloor \cdot m, j = \lfloor k/m \rfloor, \\ k = 0..(2m+1)^2 - 1 \end{cases} \quad (4)$$

here are: $x_{i,j}$ - the value of the pixel of the original image with coordinates i and j ; y_k - a set of pixels' values included in the function's structure of the median filter; k - pixel values' index of the median filter; m - the radius of the median filter.

Fig. 3 gives an explanation of the performed medial filtering operations, and shows the location of the median filter elements' values for radius values. In this case, the final, resulting value of the median filter will be obtained after sorting the found items by their value.

$$\begin{cases} \{y'_k\} = \{y'_0 \geq y'_1 \geq y'_2 \geq \dots \geq y'_{(2m+1)^2-1}\}; \\ \{y''_k\} = \{y'_0 \leq y'_1 \leq y'_2 \leq \dots \leq y'_{(2m+1)^2-1}\}, \end{cases} \quad (5)$$

here are: $\{y'_k\}$ - a declinely sorted collection of the values of the median filter's function; y'_0 - the pixel's first value in the sorted set structure of all values of the median filter's function; y'_1 - the pixel's second value in the sorted set structure of all values of the median filter's function; $(2m+1)^2-1$ - the maximum number of median filter's elements for a rectangular window; m - the median filter's radius.

The value $y'_{((2m+1)^2-1)/2}$ will be the median value of the median filter. Therefore, it will be the solution of processing one pixel. In order to find the following values, the window function shifts to the right and when the image border reaches down one pixel down and moves from the original zero position on the horizontal axis until the image boundary is reached. In the end, an array containing the found values of the median filter will be formed. In this work, the median filtering is implemented using the software package Mathcad 15, the functional diagram of the algorithm is shown in Fig. 4.

The example below describes the use of a rectangular median filter matrix, but applying a rectangular window has errors in finding the correct mean value. To minimize this error, use the window in the form of a "snowflake", which is given by the formula:

$$y_k = x_{i,j}, \begin{cases} |i| = |j|, |i| \leq m; \\ |i| \leq m, j = 0; \\ |j| \leq m, i = 0. \end{cases} \quad (6)$$

here is

$$\begin{cases} |i| = |j|, |i| \leq m; \\ |i| \leq m, j = 0; \\ |j| \leq m, i = 0. \end{cases} -$$

the condition for selecting the elements of the median filter for the window in the "snowflakes" form;

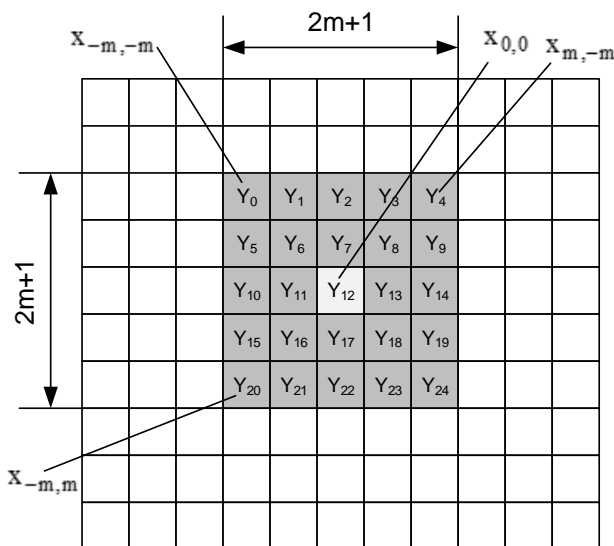


Fig. 4. Median filtering rectangular window

The explanation for median filtering for windows in the "snow flakes" for mand in the form of a circle are shown in Fig. 5 and Fig 6 respectively.

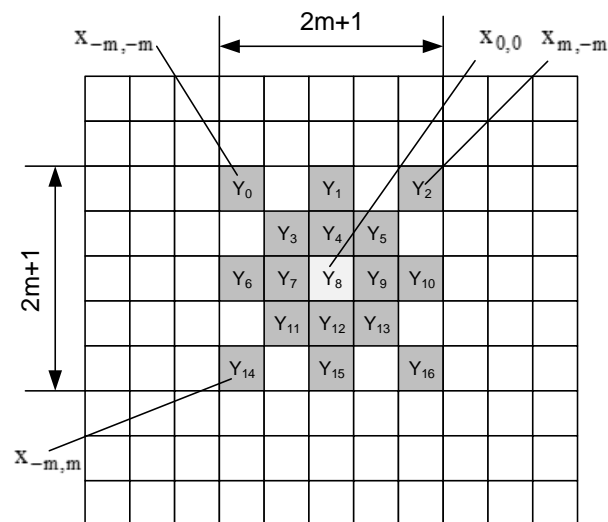


Fig. 5. Median filtering in the form of a "snow flake"

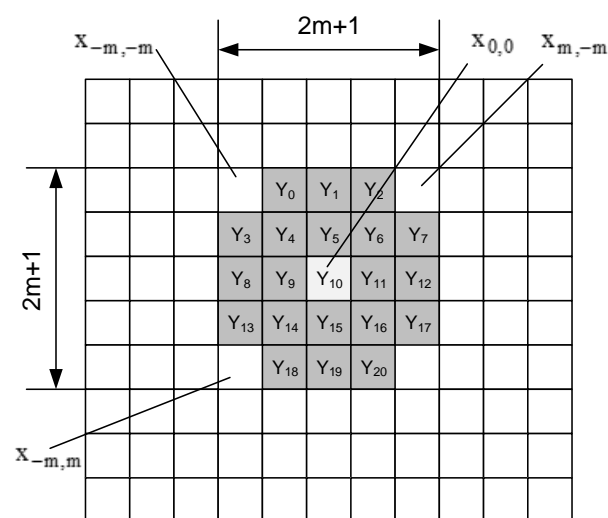


Fig. 6. Median filtering in a circle-shaped window

Each of these windows gives good results in analyzing individual image, so in order to a verify the result site suggested to use the average value obtained a result of separate three operations.

Examples of this algorithm's work for an image having an average saturation with small details given in Fig. 7, at various values of the size and type of filter aperture.



Fig. 7. Image distorted by impulse noise

Gauss filter. Often, all images have a lot of small details that greatly impede image processing, and may also result in some algorithms that are not working correctly, such as contour selection. Therefore, there is a need for pre-processing images that reduces the amount of small parts. Usually Gauss filter [18] is used to solve such problems. The Gauss filter is based on the Gaussian functions of one and two variables:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}}; \quad (7)$$

$$G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Here are: σ – standard deviation of normal distribution, which specifies the "blurring degree" of the processed image; x, y – distance from the starting point (pixel) to the point for which the value of the function is calculated vertically and horizontally, respectively. In the general case, Gauss distribution has the form (Fig. 8).

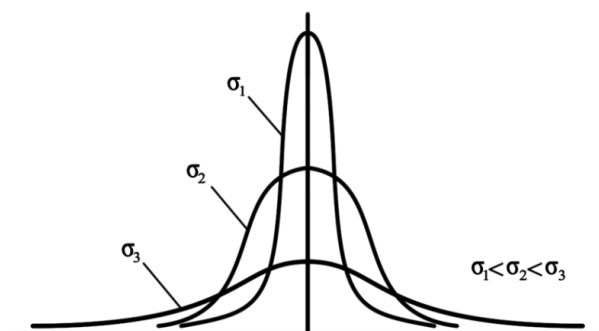


Fig. 8. The Gaussian distribution for different values

Thus, based on the Gauss function, a convolution matrix is constructed, for which the average weighted value of the neighboring pixels is calculated for each pixel of the image:

$$F_{m,n} = \sum_{i=-k}^k \sum_{j=-k}^k r_{m+i,n+j} G(i,j), \quad (8)$$

here is k – the dimension of the matrix of convolution.

As a result of applying the Gaussian filter, all distorted pixels (pixels, which brightness is very different from the brightness of the neighboring pixels) take an average value, noise is suppressed, and the contours of the objects are emphasized, which is very useful in recognizing images on digital images.

In this work, the Gauss filter is implemented by using the software package Mathcad 15. The value of the parameters is selected under condition [24-26]: the range of observation should be chosen to be the least. For each case, the execution time is calculated.

Binaryization of the image. Important stage of image processing and analysis is segmentation. In other words, this is the division of an image into an area for which a certain homogeneity criterion is performed. For example, the allocation of regions, which has approximately the same brightness. The concept of an image area is used to define a coherent group of image elements that have a certain common feature (property).

One of the basic and easy ways is segmentation using the threshold. The threshold is a feature (property) that helps to divide the desired signal into classes. The operation of the threshold division is to match the brightness value of each image's pixel with a given threshold value.

Operation of the threshold division, which eventually gives a binary image, is called binary operation. The purpose of the binarization operation is to drastically reduce the amount of information contained in the image. In the binarization process, the original halftone image having a certain number of brightness levels is converted into a black and white image whose pixels have only two values - 0 and 255.

The need of elimination a large number of errors in the binarization process, led to a large number of binary methods. They are divided into two groups, which are based on the construction of the threshold surface:

- global binarization methods;
- local binarization methods.

The threshold surface is a dimension matrix $M \times N$ corresponding to the size of the original image. Each matrix cell defines the threshold of binary brightness for the corresponding pixel in the original image. In the global binarization methods, the threshold surface is a plane with a constant threshold brightness value. In case of local binaryization methods, threshold brightness values change from point to image point, and is calculated based on some local features in the pixel vicinity or group of pixels.

The global binarization methods:

- binarization with the upper threshold: the simplest method. In this case, if the pixel's brightness, below the given threshold, is converted to black (0), the rest is white (255):

$$f'(m,n) = \begin{cases} 0, & f(m,n) \leq P; \\ 255, & f(m,n) > P. \end{cases} \quad (9)$$

- binarization with two thresholds: if the pixel's value is above the upper limit - it takes the maximum value (255), if the pixel is lowered (0), the pixels with values falling within the range of the thresholds take the fixed mean (127), which, usually, specified at the following stages [27-29]:

$$f'(m,n) = \begin{cases} 0, & f(m,n) \leq P_1; \\ 127, & P_1 < f(m,n) < P_2; \\ 255, & f(m,n) \geq P_2. \end{cases} \quad (10)$$

- Berns' binarization method. A simple square aperture with an odd number of pixels runs in a loop over all pixels of the original image. At each step are Min and Max for this aperture. The average value is found $P = (Min + Max)/2$. If the current pixel is larger than P - it becomes white (255), otherwise black (0).

Local binarization methods.

- Otts'binaryization method. This method uses a distribution histogram of the pixels' brightness values of the of the image. A histogram is constructed and based on values $h_i = n_i/N$, here is N - the total number of pixels in the image, n is the number of pixels with and brightness level. The brightness range is divided into two classes using the threshold brightness value k - the integer value from 0 to L . Each class corresponds to the relative frequencies w_0 w_1 :

$$\begin{aligned} w_0(k) &= \sum_{i=1}^k h_i; \\ w_1(k) &= \sum_{i=k+1}^L h_i = 1 - w_0(k); \\ \mu_0(k) &= \sum_{i=1}^k \frac{i \cdot h_i}{w_0}; \\ \mu_1(k) &= \sum_{i=k+1}^L \frac{i \cdot h_i}{w_1}. \end{aligned} \quad (11)$$

The method, which is proposed in this work, is an improved Burns' method. Burns' method has a great time to perform, so the following algorithm is proposed for improvement:

- the image is divided into blocks of some size (optional);
- for each block the threshold is determined;
- comparing the pixel value of the corresponding block with - the threshold value corresponding to this block.

Conclusions

1. For best results it is expedient to use for each image type your template size, as well as its type.
2. For the first type of image, it is better to choose a filter size of 5 pixels.
3. For all image types, regardless of the filter's size, the best result can be obtained by using a "snowflake" pattern.
4. An increase in the size of the filter is not feasible because it leads to an increase in the algorithm's running time and a "tampering" of the image.

REFERENCES

1. Yevseiev, S., Ahmed Abdalla, Osiievskiy, S., Larin, V. and Lytvynenko, M. (2020), "Development of an advanced method of video information resource compression in navigation and traffic control systems", *EUREKA: Physics and Engineering*, No. 5, pp. 31–42, DOI: <https://doi.org/10.21303/2461-4262.2020.001405>.
2. Mashtalir, S., Mikhnova, O. and Stolbovyi, M. (2018), "Sequence Matching for Content-Based Video Retrieval", *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP*, art. no. 8478597, pp. 549–553, DOI: <https://doi.org/10.1109/DSMP.2018.8478597>.
3. Ruban, I., Smelyakov, K., Vitalii, M., Dmitry, P. and Bolohova, N. (2018), "Method of neural network recognition of ground-based air objects", *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT*, pp. 589–592.
4. Sumtsov, D., Osiievskiy, S. and Lebediev, V. (2018), "Development of a method for the experimental estimation of multimedia data flow rate in a computer network", *Eastern-European Journal of Enterprise Technologies*, Vol. 2, Is. 2-92, pp. 56–64, DOI: <https://doi.org/10.15587/1729-4061.2018.128045>.
5. Pavlenko, M., Kolmykov, M., Tymochko, O., Khmelevskiy, S. and Larin, V. (2020), "Conceptual Basis of Cascading Differential Masking Technology", *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, DOI: <https://doi.org/10.1109/deSSERT50317.2020.9125024>.
6. Hubbard, T. and Bor, R. (2016), *Aviation Mental Health: Psychological Implications for Air Transportation*, Routledge, London, 376 p.
7. Tyurin, V., Martyniuk, O., Mirnenko, V., Open'ko, P. and Korenivska, I. (2019), "General Approach to Counter Unmanned Aerial Vehicles", *2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, DOI: <https://doi.org/10.1109/apuavd47061.2019.8943859>.
8. Mistry, D., Modi, P., Deokule, K., Patel, A., Patki, H. and Abuzagheh, O. (2016), "Network traffic measurement and analysis", *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, DOI: <https://doi.org/10.1109/LISAT.2016.7494141>.
9. Tkachov, V.M., Tokariyev V.V., Radchenko V.O. and Lebediev V.O. (2017), "The Problem of Big Data Transmission in the Mobile "Multi-Copter – Sensor Network" System", *Control, Navigation and Communication Systems*, Is. 2, pp. 154–157.
10. Mistry, D., Modi, P., Deokule, K., Patel, A., Patki, H. and Abuzagheh, O. (2016), "Network traffic measurement and analysis", *2016 IEEE Long Island Systems (LISAT)*, DOI: <https://doi.org/10.1109/LISAT.2016.7494141>.
11. Kharchenko N., Tristan, A. and Kharchenko N. (2014), "The Problem Aspect of Control of Bit Speed of the Video Stream in Telecommunication Networks", *International Conference TCSET'2014 [Modern problems of radio engineering, telecommunications, and computer science]*, Lviv-Slavske, Ukraine, February 25 – March 1, 2014, pp. 533–534.
12. Donets, V., Kuchuk, N. and Shmatkov, S. (2018), "Development of software of e-learning information system synthesis modeling process", *Advanced Information Systems*, Vol. 2, No 2, pp. 117–121, DOI: <https://doi.org/10.20998/2522-9052.2018.2.20>.
13. Wang, S., Zhang, X., Liu, X., Zhang, J., Ma, S. and Gao, W. (2017), "Utility-Driven Adaptive Preprocessing for Screen Content Video Compression", *IEEE Transactions on Multimedia*, 19 (3), art. no. 7736114, pp. 660–667.
14. Pavlenko, M., Timochko, A., Korolyuk, N. and Gusak, M. (2014), "Hybrid model of knowledge for situation recognition in airspace", *Automatic Control and Computer Sciences*, Vol. 48, Is. 5, pp. 257–263.
15. Taylor, S.E. (2015), *Health psychology*, McGraw-Hill Education, New York, 430 p.
16. Popovskiy, V.V., Saburova, S.O., Oliynik, V.F., Losev, Yu.I. and Ageev, D.V. (2006), *Matematichni osnovi teoriiy telekomunikatsiy nih system*, SMIT, Kharkiv, 564 p.
17. Pavlenko, M.A. (2012), "Metody i procedury otbora operatorov ASU pri ispol'zovanii intellektual'nyh sistem podderzhki prinyatiya reshenij", *Zbirnik naukovih prac HUPS*, No. 4 (33), pp. 171–177.
18. Kharchenko, V. and Mukhina, M. (2014), "Correlation-extreme visual navigation of unmanned aircraft systems based on speed-up robust features", *Aviation*, Vol. 18, Is. 2, pp. 80–85, DOI: <https://doi.org/10.3846/16487788.2014.926645>.
19. Brown, T.A. (2015), *Confirmatory factor analysis for applied research*, Guilford Press, New York, 462 p.
20. Gonzales, R.C. and Woods, R.E. (2002), *Digital image processing*, Prentice Inc. Upper Saddle River, New Jersey, 779 p.
21. Ericsson, K.A., Charness, N., Feltovich, P.J. and Hoffman, R.R. (2018), *The Cambridge handbook of expertise and expert performance*, Cambridge University Press, New York, 918 p.
22. Wickens, C.D. (2015), *Engineering psychology and human performance*, Psychology Press, New York, 544 p.
23. Li, L. (2015), "The UAV intelligent inspection of transmission lines", *Proceedings of the 2015 International Conference on Advances in Mechanical Engineering and Industrial Informatics*, DOI: <https://doi.org/10.2991/ameii-15.2015.285>.
24. Carter M.W. and Price, C.C. (2017), *Operations research: a practical introduction*, Boca Raton, CRC Press, FL, 416 p.
25. Fletcher, R. (2017), *Practical methods of optimization*, John Wiley & Sons, New York, 456 p.
26. Larin, V., Yeromina, N., Petrov, S., Tantsiura, A. and Iasechko, M. (2018), "Formation of reference images and decision function in radiometric correlation-extremal navigation systems", *Eastern-European Journal of Enterprise Technologies*, Vol. 4, Is. 9, pp. 27–35, DOI: <https://doi.org/10.15587/1729-4061.2018.139723>.
27. Qassim, H., Verma, A. and Feinzimer, D. (2018), "Compressed residual-VGG16 CNN model for big data places image recognition", *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, DOI: <https://doi.org/10.1109/ccwc.2018.8301729>.
28. Merlac, V., Smatkov, S., Kuchuk, N. and Nechausov A. (2018), "Resources Distribution Method of University e-learning on the Hyperconvergent platform", *Conf. Proc. of 2018 IEEE 9th International Conference on Dependable Systems, Service and Technologies. DESSERT'2018*, Ukraine, Kyiv, May 24–27, pp. 136–140, DOI: <http://dx.doi.org/10.1109/DESSERT.2018.8409114>.
29. Larin, V., Yerema, D. and Bolotska, Y. (2019), "The reasoning of necessity enhancing video privacy in conditions of providing the quality of the video information service provided in virtual infocommunication systems", *Sistemi ozbroennya i viyskova tehnika*, Vol. 2(35), pp. 158–162.

Received (Надійшла) 15.12.2020

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Тимочко Олександр Іванович – доктор технічних наук, професор, професор кафедри, Харківський національний університет Повітряних Сил імені Івана Кожедуба, Харків, Україна;

Oleksandr Tymochko – Doctor of Technical Sciences, Professor, Department Professor, Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, Ukraine;

e-mail: tymochko.alex@gmail.com; ORCID ID: <http://orcid.org/0000-0002-4154-7876>.

Ларін Володимир Валерійович – кандидат технічних наук, доцент кафедри математичного та програмного забезпечення АСУ, Харківський національний університет Повітряних Сил імені Івана Кожедуба, Харків, Україна;

Volodymyr Larin – Candidate of Technical Sciences, Associate Professor of Mathematical and Software ACS Department, Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, Ukraine;

e-mail: Volodymyr.Larin@gmail.com; ORCID ID: <http://orcid.org/0000-0003-0771-2660>.

Колмиков Максим Миколайович – кандидат технічних наук, старший науковий співробітник, докторант, Харківський національний університет Повітряних Сил ім. І. Кожедуба, Харків, Україна;

Maksym Kolmykov – Candidate of Technical Sciences, Senior Researcher, doctoral student, Air Force Science center, Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, Ukraine;

e-mail: MaksymKolmykov@gmail.com; ORCID ID: <https://orcid.org/0000-0003-1972-3543>.

Тимочко Олександр Олександрович – кандидат технічних наук, інженер з тестування програмних продуктів фірми «Kreditech», Гамбург, Німеччина;

Oleksander Timochko – Expert Quality Assurance Engineer of Kreditech Holding, Hamburg, Germany;

e-mail: alexander.timochko@gmail.com; ORCID ID: <http://orcid.org/0000-0003-0424-0426>.

Павленко Владислава Анатоліївна – студентка Харківського національного університету імені В. Н. Каразіна, Україна;

Vladislava Pavlenko – Student of V. N. Karazin Kharkiv National University, Kharkiv, Ukraine;

e-mail: Vladislava.Pavlenko@gmail.com; ORCID ID: <https://orcid.org/0000-0003-0976-0252>.

Дослідження методів фільтрації зображень в комп'ютерних системах

О. І. Тимочко, В. В. Ларін, М. М. Колмиков, О. О. Тимочко, В. М. Павленко

Анотація. У статті вказується, що людські очі менш чутливі до кольору, ніж до їх яскравості. У кольоровому просторі RGB усі три компоненти вважаються однаково важливими, і зазвичай вони зберігаються з однаковою роздільною здатністю. Однак можливо ефективніше відображати кольорове зображення, відокремлюючи яскравість від кольорової інформації та представляючи його з вищою роздільною здатністю. Простір RGB добре підходить для комп'ютерної графіки, оскільки він використовує ці три компоненти для формування кольорів. Однак простір RGB не дуже ефективний, коли справа стосується реальних зображень. Справа в тому, що для збереження кольору зображення потрібно знати та зберігати всі три компоненти RGB, і якщо один із них відсутній, це сильно спотворить візуальне зображення. Крім того, при обробці зображень у просторі RGB не завжди зручно виконувати будь-яке перетворення пікселів, оскільки в цьому випадку буде потрібно перерахувати всі три значення компонента RGB і зробити зворотний запис. Це значно знижує продуктивність різних алгоритмів обробки зображень. З цих та інших причин багато стандартів відео використовують яскравість та два сигнали, що передають інформацію про червоний та синій компоненти сигналу, як кольорову модель, відмінну від RGB. Найвідоміший серед таких просторів – YCbCr. Було виявлено, що для досягнення найкращих результатів доцільно використовувати для кожного типу зображення розмір шаблону, а також його тип. Для зображення першого типу краще вибрати фільтр розміром 5 пікселів. Для всіх типів зображень, незалежно від розміру фільтра, найкращий результат можна отримати, використовуючи шаблон "сніжинка". Збільшення розміру фільтра неможливо, оскільки це призводить до збільшення часу роботи алгоритму та "фальсифікації" зображення.

Ключові слова: зображення; компонент; алгоритм; стандарт; комп'ютерні системи.

Исследование методов фильтрации изображений в компьютерных системах

А. И. Тимочко, В. В. Ларин, М. М. Колмыков, А. А. Тимочко, В. М. Павленко

Аннотация. В статье указывается, что человеческие глаза менее чувствительны к цвету, чем к их яркости. В цветовом пространстве RGB все три компонента считаются одинаково важными, и обычно они хранятся с одинаковой разрешающей способностью. Однако возможно эффективно отражать цветное изображение, отделяя яркость от цветовой информации и представляя его с более высоким разрешением. Пространство RGB хорошо подходит для компьютерной графики, поскольку он использует эти три компонента для формирования цветов. Однако пространство RGB не очень эффективен, когда дело касается реальных изображений. Дело в том, что для сохранения цвета изображения нужно знать и хранить все три компонента RGB, и если один из них отсутствует, это сильно исказит визуальное изображение. Кроме того, при обработке изображений в пространстве RGB не всегда удобно выполнять любое преобразование точек, поскольку в этом случае потребуется перечислить все три значения компонента RGB и сделать обратную запись. Это значительно снижает производительность различных алгоритмов обработки изображений. По этим и другим причинам многие стандарты видео используют яркость и два сигнала, передающие информацию о красном и синем компонентах сигнала, как цветную модель, отличную от RGB. Самым известным среди таких пространств есть YCbCr. Было обнаружено, что для достижения лучших результатов целесообразно использовать для каждого типа изображения размер шаблона, а также его тип. Для изображения первого типа лучше выбрать фильтр размером 5 пикселей. Для всех типов изображений, независимо от размера фильтра, лучший результат можно получить, используя шаблон "снежинка". Увеличение размера фильтра невозможно, поскольку это приводит к увеличению времени работы алгоритма и "фальсификации" изображения.

Ключевые слова: изображения компонент; алгоритм; стандарт; компьютерные системы.