

Vladyslav BILOZERSKYI, Kostyantyn DERGACHOV, Leonid KRASNOV,
Anatolii ZYMOVIN, Anatolii POPOV

National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine

NEW METHOD FOR VIDEO STREAM BRIGHTNESS STABILIZATION: ALGORITHMS AND PERFORMANCE EVALUATION

Subject of study. In this paper, for the first time, an original method for estimating the change in the brightness of video data under the influence of changes in the lighting conditions of the scene and external noise is proposed. Algorithms for stabilizing the brightness of video data are also proposed. An objective assessment of the quality of video data pre-processed is given. The purpose of the research is to create a methodology for analyzing the variability of video data parameters under the influence of negative factors and to develop effective algorithms for stabilizing the parameters of the received video stream. The reliability of the method is tested using real video recordings pictured through various conditions. **Objectives:** To determine the most universal, resistant to external influences, and informative indicator necessary for an objective assessment of the quality of video data under various shooting conditions and scene lighting features; develop and programmatically implement algorithms for stabilizing video parameters based on modern programming tools. **Research methods.** Statistical analysis and pre-processing of video stream parameters as a random spatio-temporal process, algorithms for processing video data by digital filtering, and adaptive stabilization of video stream parameters. **Research results.** It has been proposed and experimentally proven that the optimal indicator of video stream quality is the average frame brightness (AFB). An algorithm for spatiotemporal processing of video data is proposed that generates a sequence of AFB values from the original video stream. The paper also proposes digital algorithms for filtering and stabilizing the brightness of a video stream and investigates the effectiveness of their application. **Conclusions.** The scientific novelty of the results obtained lies in a new method for analyzing and evaluating the parameters of video surveillance data and algorithms for filtering and stabilizing the brightness of the video stream. The performance of the proposed algorithms has been tested on real data. The algorithms are implemented in the Python software environment using the functions of the OpenCV library.

Keywords: video stream; average frame brightness; video brightness trend; trend digital filtering; brightness stabilization algorithms.

1. Introduction

An important aspect of solving the fundamental problem of pattern recognition is image recognition, better known as computer vision. This is a fast-growing segment of artificial intelligence that involves the analysis and interpretation of digital images and videos using deep learning algorithms and models. Computer vision is a basic tool for identifying and locating objects in an image or video, used for facial recognition, optical character recognition, and traffic sign recognition, and many other useful applications. However, the effectiveness of computer vision algorithms is significantly limited due to a number of negative factors and external influences. Factors such as poor shooting angles, noise, imperfect scene lighting, camera shake, etc. negatively affect the quality of images and video data. This causes visual discomfort when viewing images and videos and significantly degrades the quality of vision systems. Note that among many negative phenomena, the dominant

influence on the deterioration of video data quality is exerted by the uncontrolled moving of the video camera [10 – 12] and insufficient or large variability of scene lighting [17 – 20]. Unfortunately, many of the results of recent years and the developed algorithms for improving video quality have turned out to be unproductive. This encourages the authors to continue research in this direction. We believe that the development of new methods for pre-processing video data to compensate for negative factors affecting the quality of image and video data processing is invariably relevant and useful.

2. Related works

The development of technologies for detecting and recognizing objects with the use of technical vision systems based on neural networks is proceeding in two main directions. The first way is the development of the architecture of neural networks, methods for training them and managing available resources, followed by

creation of specialized libraries. The second way is the use of primary data preparatory processing to compensate for negative external influences. This results in a much more complete and efficient use of neural network resources.

The best technique for considering pattern recognition problems is the use of CNNs (Convolutional Neural Networks). This is the main image recognition tool. There are several CNN applications: deep convolutional neural network (DCNN), region-CNN, convolutional neural network (FCNN). One of the first thoughts on the new architecture of convolutional neural networks [1] was published in 1998. In 2012, the AlexNet convolutional neural network was designed [2, 3]. It successfully classifies million images of thousand different categories [4]. This architecture is used for deep learning in computer vision tasks.

The OpenCV library realizes hierarchical generalization and resource management in supporting neural networks deep learning. OpenCV version 3.1 includes a DNN (Deep Neural Network) module for direct link with neural networks pre-trained using popular frameworks. Libraries to form modern networks have been created. These are the Caffe [5] and TensorFlow [6] platforms, the scientific computing platform Torch/Pytorch [7], the Keras library [8]. In OpenCV 3.3, the status of the deep learning module has been upgraded from the `opencv_contrib` repository status to the status of the main repository [9].

Presently, there is an increased interest in data pre-processing aimed to enhance the efficiency of pattern recognition using neural networks. Examples of thematic publications are works [10, 11]. These present video stream preprocessing algorithms for industrial control and a preprocessing method, which employs a new HVS model suitable for video compression.

A severe problem when viewing and analyzing video data are interferences caused by uncontrolled camera movements during recording. This degradation manifests itself as random fluctuation of frames and is accompanied by a blurring effect. In [12], an overview of the video stabilization problem is presented, considering the practical features and basic mathematical approaches of video stabilization methods. Particular attention is paid to assessing the quality of stabilization.

The ever-relevant problem of complete reference visual assessment of image quality with noise reduction was studied in [13], where special attention was paid to images with low contrast and noise-like texture.

The work [14] shows how images can be subjected to lossy compression in such a way that the distortions introduced are not visible. To get this, one can use two modern visual quality metrics – MSSIM and PSNR-HVS-M. These proposed lossy compression

techniques can be successfully used in pre-processing image and video data for various applications.

The articles [15, 16] consider the concept of improving the quality of the operation of vision systems through a set of image pre-processing algorithms at the discretion of the user. Algorithms for compensation of external negative influences (unfavorable geometrical factor, poor lighting conditions during video recording, noise effects, etc.) have been created.

The article [17] focuses on the development of a diagnostic system to assist film crews in calibrating stereoscopic cameras and analyzing 3D depth in real-time on a film set. The system intends GPU acceleration with using algorithms that do not rely on knowledge of camera parameters, making it a valuable tool for ensuring the quality of 3D imagery in cinematography.

The paper [18] describes a preprocessing method based on contrast enhancement that can improve object recognition in computer vision tasks involving complex scene lighting conditions. The proposed procedure involves several stages and effectively improves object recognition results in scenarios where lighting poses challenges.

The article [19] is dedicated to the use of deep learning techniques, specifically SRGAN and DAIN, to improve the quality of real-time video transmission while alleviating the problem of system overload, which is particularly relevant in the context of non-face-to-face work and the development of streaming services. The proposed approach involves selectively applying these techniques to different regions of the image to achieve the desired improvements in image quality and system performance.

The research work [20] focuses on addressing the bandwidth management challenges in multi-camera video streaming for deep learning video analytics. DeepStream offers solutions such as the ROI-Det algorithm, content-aware bandwidth optimization, and an Elastic Transmission Mechanism to optimize utility and bandwidth efficiency. Evaluation results on real-world datasets demonstrate the effectiveness of these solutions.

The conducted analysis clearly demonstrates the relevance of the research carried out in our work, which will bring undoubted benefits in increasing the efficiency of computer vision systems.

3. Work objective

The following tasks were set for the research:

- to analyze changes in initial data in video surveillance systems under the influence of negative external influences;
- to develop a universal, stable, and informative quality criterion that allows objective assessment of the

state of video data under various shooting conditions and the nature of scene illumination;

- based on the analysis results, create effective algorithms for stabilizing the video stream brightness indicator;
- evaluate the change in video data quality as a result of preprocessing;
- write program code for algorithms in Python using the functions of the OpenCV library;
- verify the reliability of the results obtained using real video recordings.

4. Analysis of video data parameters

When solving various problems of detecting and recognizing objects using video surveillance, all types of disturbing influences make a definite negative contribution to the quality of the original video data. The most significant factor in the performance degradation of video systems is unfavorable scene illumination conditions. This makes it difficult to use binarization algorithms with clipping at the brightness threshold, algorithms for determining the boundaries of objects, etc. when processing images of individual frames. In addition, the overall level of illumination also changes over time. Changes in the brightness of video data can be considered as a non-stationary random process containing fast (high-frequency noise) components and slowly changing components due to changes in the overall level of illumination.

When analyzing video recordings or real time video, it is necessary to use a universal, stable and informative criterion of performance (quality), which yields an objective estimation of the video data state across various shooting conditions and the nature of the scene illumination. The authors believe that the most appropriate and almost the only stable quality indicator is the average frame brightness, AFB. It is customary to encode video pixel brightness in the `uint8` numerical format. Hence, the AFB value estimates lie in the range [0...255]. No less important are the indicators of the change in the average brightness of the entire video sequence, which is strictly periodic. Recall that the video recording speed standard is characterized by the number of frames per second (fps). In most cases fps = 25, 30 or 60 is used.

According to the authors, in order to obtain an objective assessment of changes in video brightness, it is advisable to trace the dependence of the average brightness of AFB frames on time and calculate the statistical characteristics of this random process (trends in average values over the entire analysis interval or in individual areas, dispersion of brightness deviation from the average value, AFB value distribution histograms).

Videos are usually presented in an orthogonal three-component RGB color space. This is convenient

for viewing, but the R, G, B frame components individually do not contain complete information about the image pixels brightness. It is customary to overcome this limitation by transforming images from the RGB color space to the HSV space. Recall that HSV is a color model based on three characteristics: color tone (hue, H), saturation (S), and brightness (value, V). These options reflect the following:

- Hue - color tone (in degrees). Generally, it varies within 0 - 360°, but sometimes it is given in the range 0 to 100 (per cent) or 0 ... 1. In Windows, the entire color spectrum is divided into 240 colors (this can be observed in the MS Paint palette editor); that is, a Hue range is squeezed to the 0 - 239 area here (color 240 is missing as it would duplicate 0);

- Saturation. This varies within 0 - 100 or 0 - 1. The larger this figure, the "purer" the color, which is why the parameter is often called color purity. The closer this parameter is to zero, the closer the color is to neutral gray;

- Value (or Brightness) - component magnitude, also given in 0 - 100 or 0 - 1 range.

The transition from the RGB to HSV space makes it easy to evaluate the level of the frame average brightness (using the component V):

$$AFB = \frac{1}{MN} \sum_{n=1}^N \sum_{m=1}^M V(n, m), \quad (1)$$

where $V(n, m)$ is an element of the two-dimensional array representing the brightness of pixels in the frame picture of $M \times N$ size. Note that the calculation of AFB is a rather complicated procedure since the number of arithmetic operations increases dramatically with an increase in the resolution of video frames. As a rule, modern camcorders get high resolution. For example, a standard frame in Full HD format has resolution 1920 x 1080 and it involves about 2 million pixels; and a frame in the Ultra HD of the 3840 x 2160 standard contains about 8.3 million pixels by now. This may give rise to serious performance limitation not only at the stage of pre-processing video data but also when solving various applied problems. How this issue can be overcome will be shown below.

Using the resources of the Python programming language and the OpenCV library, we analyzed the investigated parameters of video data. We do not supply complete code pieces for brevity. We are focusing on the procedure of transition from RGB space to HSV space and the program function that determines the AFB value. Image conversion from the BGR format (see comment on characters writing order below) to HSV format is performed using the function `cv2.cvtColor(img, cv2.COLOR_BGR2HSV)`.

The code fragment that was used to estimate the AFB value is shown in Fig. 1.

```

51 # Average frame brightness calculation
52 def brightness_avg(image_hsv):
53     # Getting separate channels
54     h, s, v = cv2.split(image_hsv)
55     # Calculate the average frame brightness as a result
56     return np.mean(v)

```

Fig. 1. AFB calculation function

For analysis completion, it is necessary to investigate the dependence of the average brightness of AFB frames on time and calculate the statistical characteristics of this random process (the average value and the spread over the entire analysis interval or individual sections of the video sequence). A universal algorithm for processing video data is proposed for this purpose, which allows generating an AFB trend from the source video. The structure of the algorithm is shown in Fig. 2. The algorithm model for calculating and analyzing AFB is shown in Fig. 2.

The samples recorded by the drone were used as video files for processing. They were shot from high altitude in high resolution 3840 x 2160 at a standard frame rate of 30 fps. Unfortunately, the limited space of the publication allows only one example to be given. A characteristic frame of these records is shown in Fig. 3 and is provided with a list of basic parameters (format, frame size in pixels, frame rate, number of frames processed). In the following consideration, it is more convenient to evaluate AFB characteristics on a time scale. Great variability in the features of these videos can be seen even with a quick analysis. The point is the different degrees of illumination of the scene and the different proximity and speed of objects in the shooting area. The results of calculating the changes in the average brightness of the picture over all recorded frames make it possible to objectively evaluate the features of video data. These are given in Fig. 3 too. Along with graphs of changes in AFB, normalized histograms of the distribution of AFB values in a certain brightness range are of practical interest. They clearly show the degree of brightness variance relative to the average values.

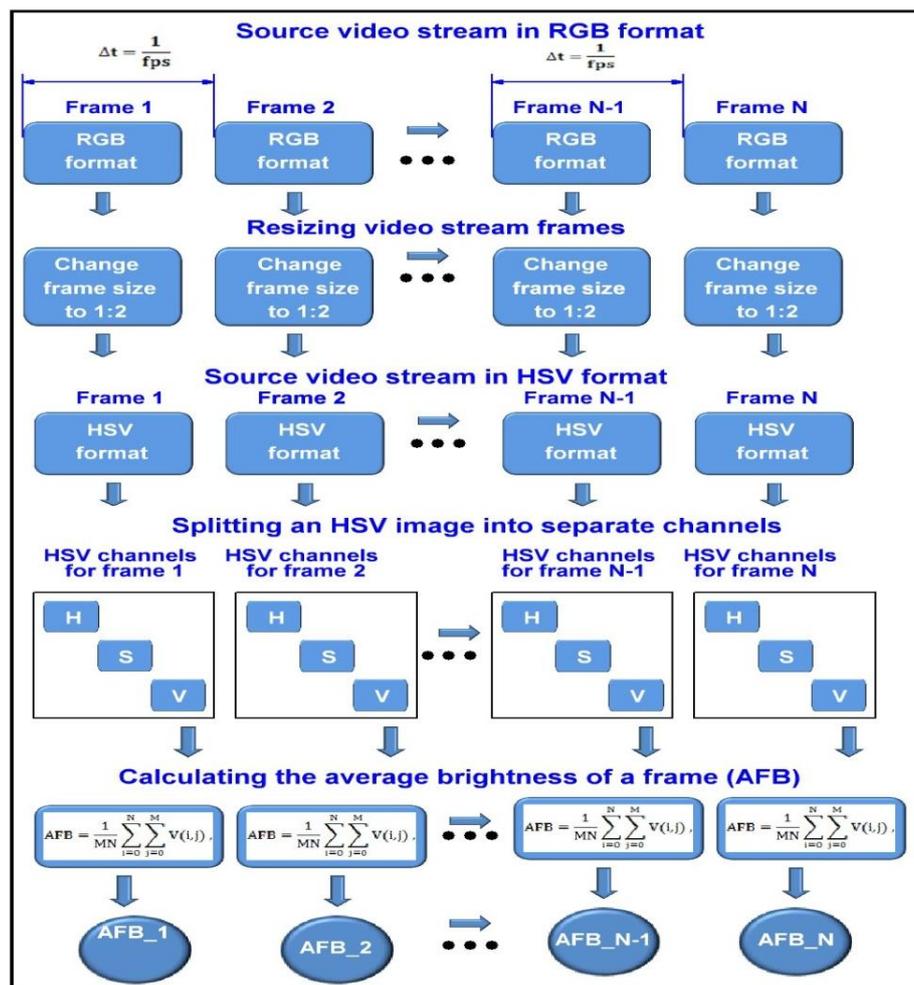


Fig. 2. AFB analysis chart

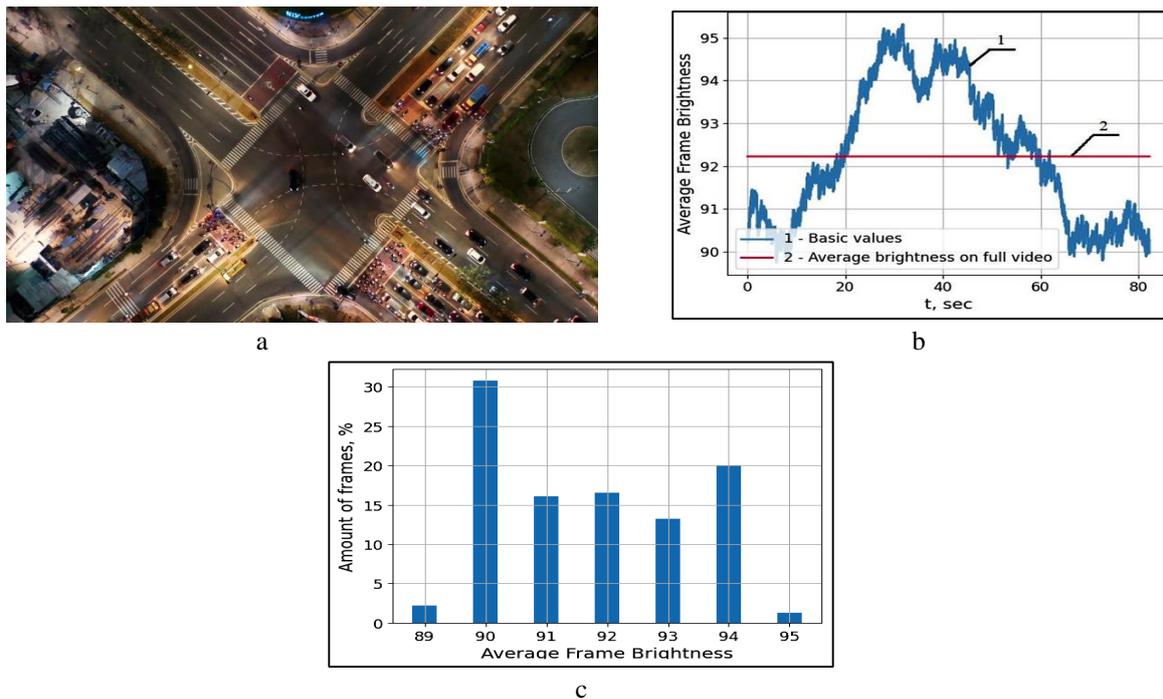


Fig. 3. A sample shot and results of the video stream analysis:

- a – video stream frame, frame format: (width x height) = 3840.0 x 2160.0, frames count = 2466.0, fps = 30.0;
- b – AFB trend for the entire video; c – histogram of average frame brightness

To resize an image in Python you can use the `cv2.resize()` function from OpenCV and choose the appropriate interpolation method. Depending on the method chosen (in the case study, it is `cv2.INTER_AREA`), each step uses an appropriate algorithm to determine which pixel gets its current value based on neighboring pixels and which resizing scale is involved. The program code for changing the frame size is shown in Fig. 4.

```
new_width = int(frame.shape[1] * size_multiplier)
new_height = int(frame.shape[0] * size_multiplier)
dim = (new_width, new_height)
frame = cv2.resize(frame, dim, cv2.INTER AREA)
```

Fig. 4. The excerpt code to resize video stream frames

Reducing the size of high-resolution video frames is purely technological operation. It is much more important to ensure that the calculation of the AFB parameter on resized frames does not lead to significant errors. To test this, we examined AFB trends in different video streams (an example snapshot was shown in Fig. 3). We handled frames of the source stream (3840 x 2160), half-resized frames (1920 x 1080), and frames scaled down by a factor of 4 (960 x 540). The evaluation results are shown in Fig. 5, a. The difference between the three curves in the diagrams is small enough to be assessed visually. Compared to the reference trend (resulted from the stream of high-resolution), the other two trends (frames of 1920 x 1080 and 960 x 540 size) have

noticeable distinctions in the AFB index, as can be seen in Fig. 5, b.

The graphs clearly show that the estimation scatter is no more than 0.1 elementary unit of the brightness scale. In the uint8 data format, the difference in estimates is negligible and is no more than 0.05% of the brightness span.

5. Filtering AFB trends

When analyzing the brightness of the video stream (see Fig. 3), it is easy to see that these are graphs of a random process with low- and high-frequency components. The low frequency component reflects the slowly changing lighting conditions of the scene and carries useful information. High frequency components behave like noise. They reflect rapid and spontaneous changes in the video scenario brightness. These HF components can be considered interference that must be eliminated. For example, using a rejection procedure (low-pass filter).

The authors used one of the simplest and most efficient digital low-pass filtering algorithms – an averaging filter with a sliding window. It acts like a buffer that stores the latest AFB averaging data in an amount determined by the product of the filter window size (in sec) and the frame rate (fps). For example, the number of AFB samples stored in the averaging buffer would be 150 at a 5 sec window and frame rate = 30. At each

filtering stage, the data is averaged according to the formula:

$$AFB_{filtr} = \frac{1}{W} \sum_{w=1}^W AFB_w, \quad (2)$$

where AFB_w is a one-dimensional array of numbers composed of AFB values filled the filter window W . At each new filtering step, the buffer is shifted, i.e., it captures a new number AFB_{w+1} , while removing the first number AFB_1 . Such an averaging routine is known as a "sliding window".

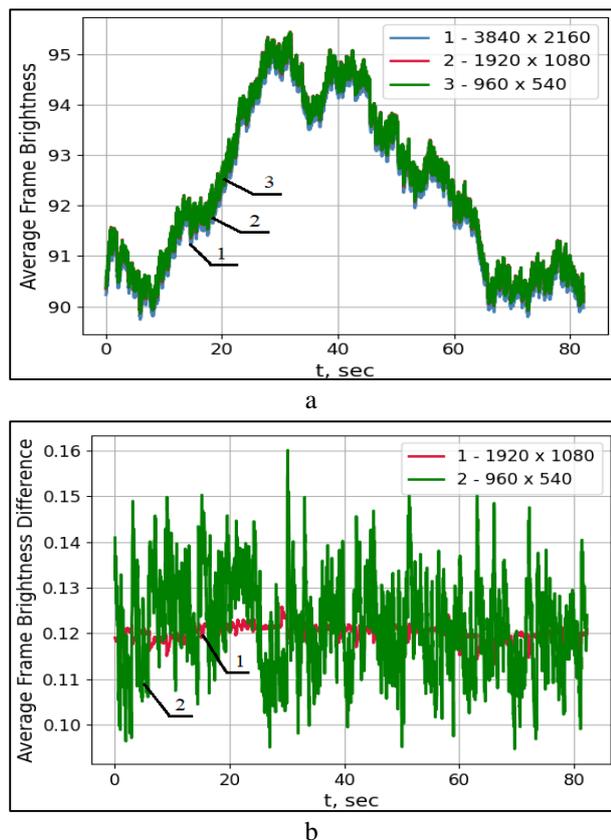


Fig. 5. Difference in the trends of AFB under different frame size

The results of filtering AFB trends are shown in Fig. 6. To evaluate the filtering efficiency, various durations of the smoothing window W were chosen – 1, 5, 10, 20 seconds. For each video record, the AFB trends before and after filtering were plotted together (Fig. 6, a). It is obvious that for a small smoothing window ($T = 1$ s) the filter efficiency is low. It has a wide bandwidth and does not deform the shape of the AFB trend, although it is not able to remove some of the noise. With a larger filter's time constant ($T = 5$), the filtering quality improves (Fig. 6, b) – the noise component is completely suppressed, and the influence of mid-frequency noise is reduced. This option is displayed as a

separate graph against the background of the original AFB curve.

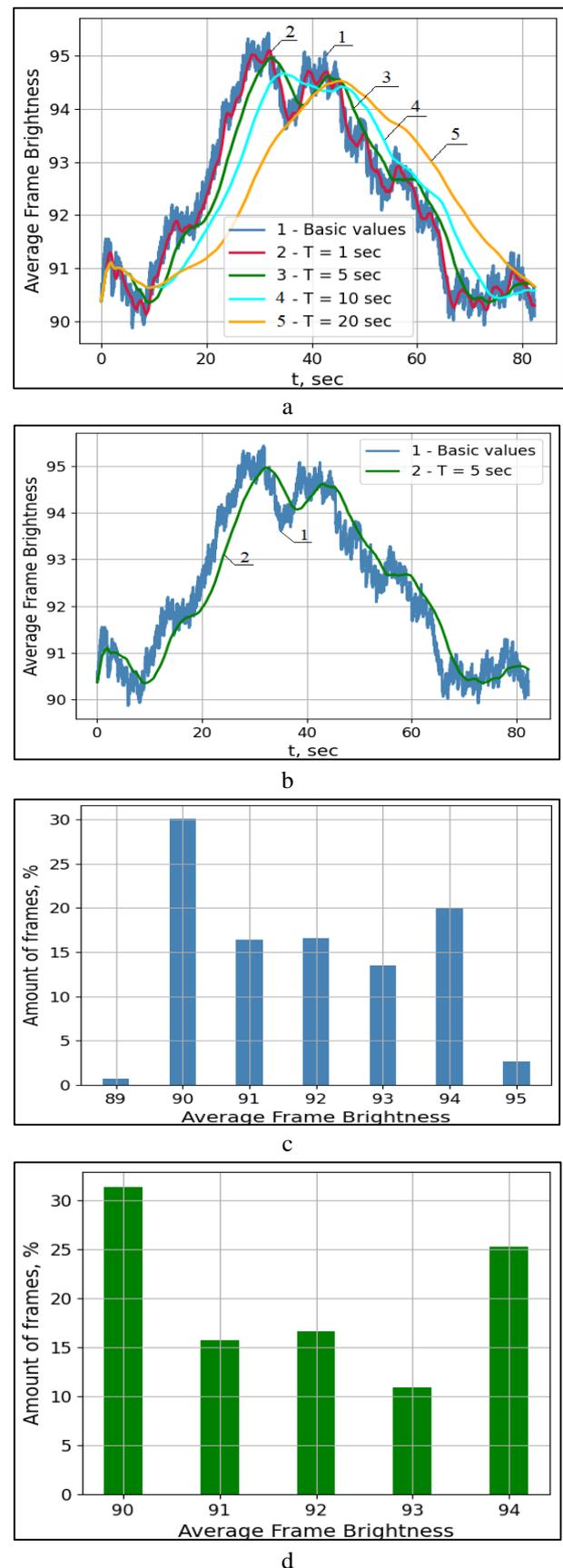


Fig. 6. Results of data filtering

Equally informative are the results of comparing the histogram of the AFB distribution before and after filtering with a time constant $T = 5$ s (blue histogram for the initial distribution and green histogram for the distribution after filtering).

The histograms (Fig. 6, c and Fig. 6, d) clearly show that the nature of the distribution of AFB has not changed, but the spread of its values relative to the average after filtering has noticeably increased. Both before and after filtering, the accuracy of representing the brightness averaging data is determined by rounding errors that do not exceed half a division of the gray scale. In the uint8 data format, this rounding error is negligible and does not exceed 0.4%.

6. Video stream brightness stabilization

To stabilize the brightness of video stream frames, the following algorithm was proposed:

- as a reference, AFB_{filtr_i} are used; these are the average brightness values at frames processed with a low-pass filter;
- for each frame of the video stream, the difference between the values of the average brightness AFB_i and the values AFB_{filtr_i} is determined, taking into account the sign;
- based on the obtained increments ΔAFB_i , a procedure for linear pixel correction is constructed according to the rule:

$$\begin{aligned} \Delta AFB_i &= AFB_i - AFB_{filtr_i}; \\ \text{if } \Delta AFB_i \geq 0 \text{ then} \\ AFB_{i\text{correction}} &= AFB_i - \Delta AFB_i; \quad (3) \\ \text{if } \Delta AFB_i < 0, \text{ then} \\ AFB_{i\text{correction}} &= AFB_i + \Delta AFB_i; \end{aligned}$$

- during correction, the brightness of pixels in the frame is limited and kept in the range $[0 \dots 255]$.

The stabilization results are shown in Fig. 7.

Fig. 8 shows the distribution histograms for pixel brightness in the frame before and after stabilization. It can be seen that the treated frame (see Fig. 7) after brightness stabilization becomes slightly brighter than the original one. At the same time, the histogram of this frame does not change its shape, but only shifts towards greater brightness.

From the results of this consideration, a very important conclusion should be drawn, namely: the proposed procedure for stabilizing the brightness of the video stream is linear. This does not result in color imbalance in video frames as happens when using well-known nonlinear procedures for normalizing frame brightness histograms (equalizers).

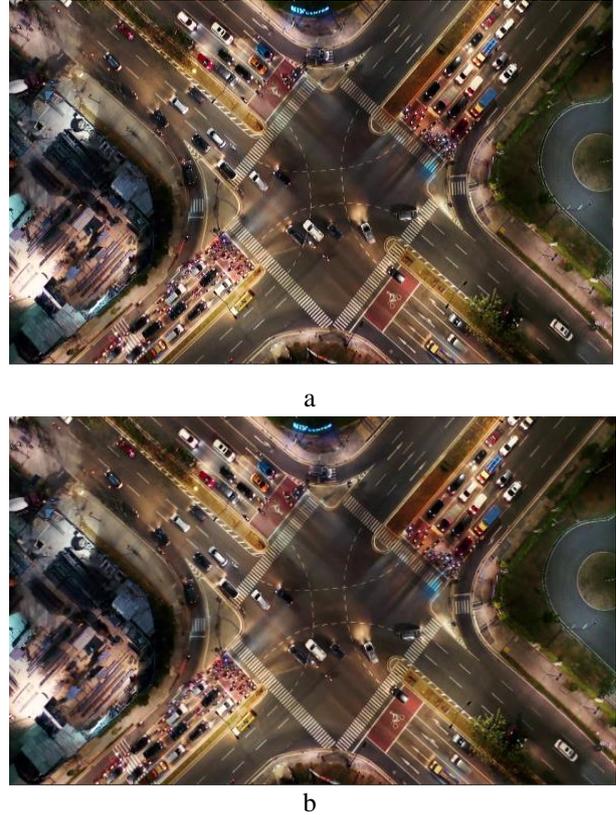


Fig.7. Results of brightness stabilization: a –before correction; b – after correction

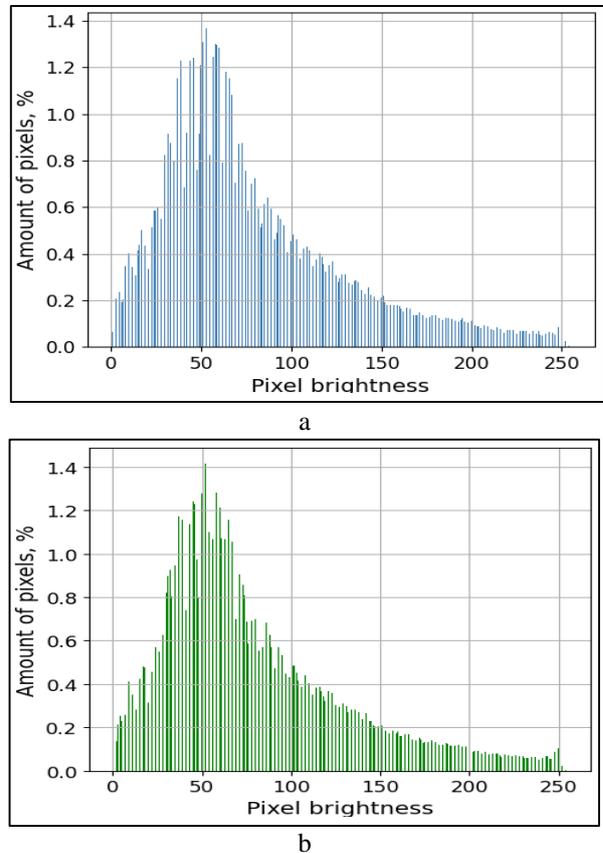


Fig. 8. Results of video stream brightness stabilization: a –before correction; b – after correction

We also note the high accuracy of introducing corrections – as in the case of filtering the AFB trend, the error is no more than half a division of the gray scale.

7. Results of experimental studies

A study was conducted to test the performance quality of video data pre-processing. In most practical applications, quality is understood as a measure of the proximity of two images – the converted and the original. For comparison purposes, criteria of visual perception and objective criteria based on weighing the difference between pairs of pixels in two images were used in the work. Typically, peak signal-to-noise ratio (PSNR) is used as an indicator of similarity. It is defined as:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}, \quad (4)$$

where the mean square error MSE is given by

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(m, n) - C(m, n))^2. \quad (5)$$

This metric is used to analyze processing quality in most applications. Nevertheless, the authors believe that it is more expedient to use in the case study a linear measure of correspondence, since the proposed procedure for stabilizing frames brightness is linear. This indicator is called the mean absolute error (MAE) and is calculated using the formula:

$$\text{MAE} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |I(m, n) - C(m, n)|. \quad (6)$$

Here and below, $I(m, n)$ and $C(m, n)$ denote the original and corrected frame images, respectively.

The results of processing the video sequence are shown in Fig. 9 as AFB trends before/after correction and as MAE distribution histograms for all video frames. The histograms of the MAE error scale are calibrated in the range $[0 \dots 255]$.

An important integral performance indicator for the proposed procedure of stabilizing the brightness of a video stream is the factor of correction, which characterizes quantity of the video frames corrected with respect to the total number of frames. Thus, the correction factor CF is computed as the ratio of the number of corrected frames in the record to the total number of recorded frames. The CF indicator is further evaluated in %.

Fig. 10 shows two studied examples for the distribution of MAE estimates; estimates were obtained after

filtering the video records using time constants $T = 10$ s and 20 s. Analysis of the data received indicates that:

- the MAE value in these examples is relatively small and does not exceed two units on the brightness scale;
- in most samples of the tested video sequence, certain intervals can be observed, in which the frames were not subject to brightness correction;
- the CF values in the examples given range from 19% to 45%.

Note that an increase in the values of the correction factor CF indicates the correct choice of the AFB trend filtering parameters (the width of the filter window is optimally selected).

Based on the research conducted, the principal conclusion can be drawn in respect of obtaining high-quality stabilization of the video stream brightness: it is advisable to flexibly configure video data preprocessing parameters and govern filtering characteristics in accordance with the recording features and variability of scene lighting conditions.

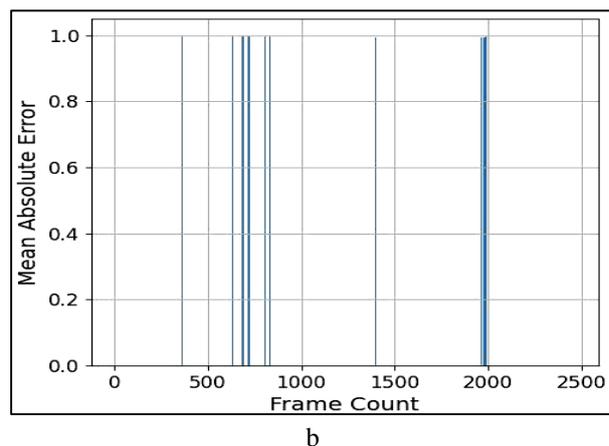
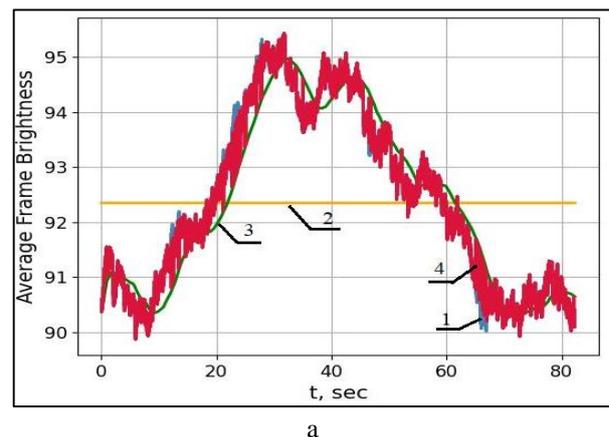


Fig. 9. Assessing the quality of brightness stabilization: a – 1. basic values, 2. average brightness on full video, 3. average brightness with dynamic window, 4. average brightness after correction; b – T filt. = 5 s, $CF = 3.37\%$

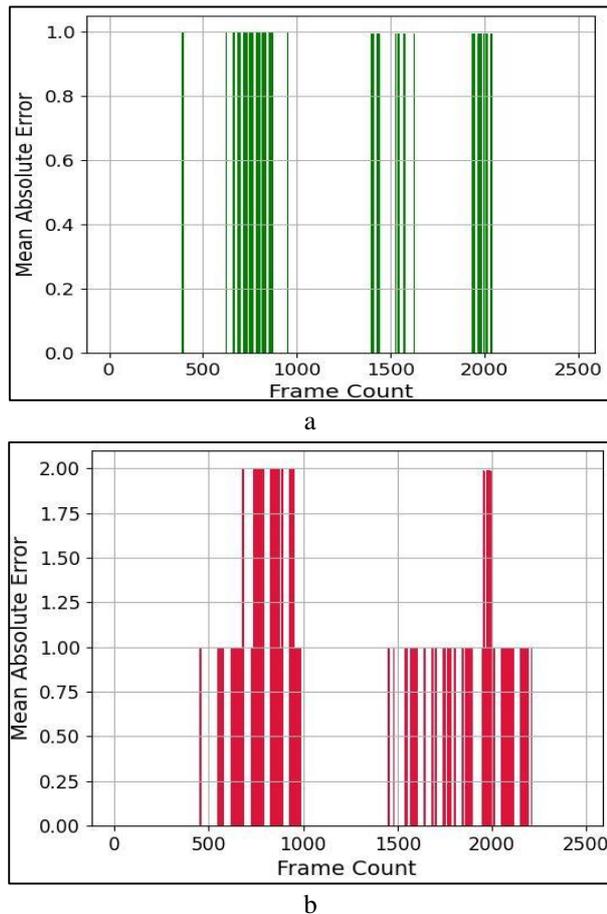


Fig. 10. Errors correction distribution for the entire video stream:
 a – T filt. = 10 s, CF= 19.71 %;
 b – T filt. = 20 s, CF= 44.89 %

8. Discussion

At the final stage of the work, a detailed discussion of the prospects for further research into the problem of stabilizing the brightness of video data was carried out. The obvious advantage of using linear transformation algorithms was surely confirmed. The linearity of conversion allows you to preserve the color balance of the original video data, increase accuracy and simplify the calculation procedure.

It is advisable to additionally perform an in-depth comparative statistical analysis of the proposed algorithm and known nonlinear algorithms associated with stabilizing the brightness and increasing the contrast of video data. Over this, the brightness stabilization procedure should be considered as a generalized conversion of video stream frames. For performance quality evaluation, one should use not only the standard deviation MSE, but normalized cross-correlation coefficients and indices of structural similarity of the original and transformed frames as well.

The authors believe it useful to conduct further research into the capabilities of real time brightness stabilization algorithms for high-resolution video. While bringing this, one should focus not only on high-performance computers, but also on microcomputers (for example, Raspberry Pi). This will facilitate the emergence of useful applications for mobile computer vision systems.

9. Conclusion

Preliminary processing of video data improves the quality of computer vision systems; the latter are used to solve a wide range of pattern recognition problems that require artificial intelligence technologies based on neural networks. Accordingly, the research conducted is relevant and useful. Recommendations for implementing video image brightness stabilization algorithms are universal in nature and could be used when designing new and upgrading existing systems. The work has an applied focus. The proposed algorithms are implemented as program modules in the Python programming language with the use of OpenCV library resources. The performance characteristics of different methods for stabilizing the brightness of a video stream has been experimentally tested.

Authors' contributions

Vladyslav Bilozerskyi – selection and optimization of modern programming methods and information resources for the implementation of promising algorithms for stabilizing video stream parameters in computer vision systems; launch and testing of the developed software; and conduct experiments.

Kostyantyn Dergachov – analysis of information resources related to improving the quality of video data in computer vision systems, setting the task on researching methods and algorithms for stabilizing the brightness of a video stream, formulating requirements for algorithms implementation.

Leonid Krasnov – development of a methodology for analyzing algorithms for stabilizing video stream parameters using modern methods of digital image filtering as applied to high-resolution and ultra-high-resolution video images, creating and applying algorithms for stabilizing video parameters, and testing the proposed algorithms.

Anatolii Zymovii – formulation of the main conclusions on the results of the study and recommendations for their practical use, editing of the article, translation of source materials from Ukrainian into English.

Anatolii Popov – analysis of the results and evaluation of the experimental studies effectiveness.

All authors read and consented to publication of the manuscript version.

References

1. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324. DOI: 10.1109/5.726791.
2. Information about the ImageNet library [Electronic resource]. Available at: <https://en.wikipedia.org/wiki/ImageNet> (accessed 17.04.2023).
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, 2012, vol. 25, pp. 1097–1105.
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017, vol. 60, no. 6, pp. 84–90. DOI:10.1145/3065386.
5. Software Caffe [Electronic resource]. Available at: <http://caffe.berkeleyvision.org> (accessed 17.04.2023).
6. TensorFlow library [Electronic resource]. Available at: <https://www.tensorflow.org> (accessed 17.04.2023).
7. Scientific computing platform Torch [Electronic resource]. Available at: <http://torch.ch> (accessed 17.04.2023).
8. Keras library [Electronic resource]. Available at: <https://keras.io> (accessed 17.04.2023).
9. OpenCV DNN library [Electronic resource]. Available at: <https://github.com/opencv/opencv/tree/master/samples/dnn> (accessed 17.04.2023).
10. Skadins, A., Ivanovs, M., Rava, R., & Nesenbergs, K. Edge pre-processing of traffic surveillance video for bandwidth and privacy optimization in smart cities. *2020 17th Biennial Baltic Electronics Conference (BEC)*, Tallinn, Estonia, 2020, pp. 1-6. DOI: 10.1109/BEC49624.2020.9276799.
11. Zaharia, P., Bigioi, P., & Corcoran, P. M. Hybrid video-frame pre-processing architecture for HD-video. *2011 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2011, pp. 89-90. DOI: 10.1109/ICCE.2011.5722927.
12. Guilluy, W., Oudre, L., & Beghdadi A. Video stabilization: overview, challenges and perspectives. *Signal Processing: Image Communication*, 2021, vol. 90, article no. 116015. DOI: 10.1016/j.image.2020.116015.
13. Egiazarian, K., Ponomarenko, M., Lukin, V., & Ieremeiev, O. Statistical Evaluation of Visual Quality Metrics for Image Denoising. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 6752-6756. DOI: 10.1109/ICASSP.2018.8462294.
14. Lukin, V. V., Zriakhov, M. S., Ponomarenko, N. N., Krivenko, S. S., & Zhenjiang, M. Lossy compression of images without visible distortions and its application. *IEEE 10th international conference on signal processing proceedings*, Beijing, China, 2010, pp. 698-701. DOI: 10.1109/ICOSP.2010.5655751.
15. Dergachov, K., Krasnov, L., Bilozerskyi, V., & Zymovin, A. Data pre-processing to increase the quality of optical text recognition systems. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2021, no. 4, pp. 183-198. DOI: 10.32620/reks.2021.4.15.
16. Bilozerskyi, V., Dergachov, K., & Krasnov, L. Analiz i poperednya obrobka videodanykh dlya pidvyshchennya yakosti roboty system tekhnichnoho zoru [Analysis and pre-processing of video data to improve the quality of computer vision systems]. *Problemy keruвання ta informatyky – Problems of control and informatics*, 2023, vol. 68, no. 2, pp. 50–66. DOI: 10.34229/1028-0979-2023-2-4. (In Ukrainian).
17. Perek, P., Mielczarek, A., & Makowski, D. High-Performance Image Acquisition and Processing for Stereoscopic Diagnostic Systems with the Application of Graphical Processing Units. *Sensors*, 2022, vol. 22(2), article no. 471. DOI: 10.3390/s22020471.
18. Wang, T-s., Kim, G. T., Kim, M., & Jang, J. Contrast Enhancement-Based Preprocessing Process to Improve Deep Learning Object Task Performance and Results. *Applied Sciences*, 2023, vol. 13(19), article no. 10760. DOI: 10.3390/app131910760.
19. Cho, S-Y., Kim, D.-Y., Oh, S-Y., & Sohn, C-B. Reducing System Load of Effective Video Using a Network Model. *Applied Sciences*, 2021, vol. 11(20), article no. 9665. DOI: 10.3390/app11209665.
20. Guo, H., Tian, B., Yang, Z., Chen, B., Zhou, Q., Liu, S., Nahrstedt, K., & Danilov, C. DeepStream: Bandwidth Efficient Multi-Camera Video Streaming for Deep Learning Analytics. *arXiv – CS – Networking and Internet Architecture*, 2023, doi: 10.48550/arXiv.2306.15129.

Received 03.07.2023, Accepted 20.09.2023

НОВИЙ МЕТОД СТАБІЛІЗАЦІЇ ЯСКРАВОСТІ ВІДЕОПОТОКУ: АЛГОРИТМИ ТА ОЦІНКА ЕФЕКТИВНОСТІ

Владислав Білозерський, Костянтин Дергачов, Леонід Краснов,
Анатолій Зимовин, Анатолій Попов

Предмет дослідження. У цій статті вперше запропоновано оригінальний метод оцінки зміни яскравості відеоданих під впливом зміни умов освітленості сцени та зовнішніх шумів. Також запропоновано алгоритми стабілізації яскравості відеоданих. Дано об'єктивну оцінку якості відеоданих за результатами попередньої

обробки. **Метою роботи** є створення методології аналізу мінливості параметрів відеоданих під впливом негативних факторів та розробка ефективних алгоритмів стабілізації параметрів отриманого відеопотоку; дається ретельна перевірка достовірності методу на реальних відеозаписах, здійснених у різних умовах. **Завдання:** Визначити найбільш універсальний, стійкий до зовнішніх впливів та інформативний показник, необхідний для об'єктивної оцінки якості відеоданих за різних умов зйомки та особливостей освітлення сцени; розробити та програмно реалізувати алгоритми стабілізації параметрів відео на основі сучасних засобів програмування. **Методи дослідження.** Статистичний аналіз та попередня обробка параметрів відеопотоку як випадкового просторово-часового процесу, алгоритми обробки відеоданих шляхом цифрової фільтрації та адаптивної стабілізації параметрів відеопотоку. **Результати досліджень.** Запропоновано та експериментально доведено, що оптимальним показником якості відеопотоку є середня кадрова яскравість (AFB). Запропоновано алгоритм просторово-часової обробки відеоданих, який генерує послідовність значень AFB з вихідного відеопотоку. У роботі також запропоновано цифрові алгоритми фільтрації та стабілізації яскравості відеопотоку та досліджено ефективність їх застосування. **Висновки.** Наукова новизна отриманих результатів полягає в новому методі аналізу та оцінки параметрів даних відеоспостереження, алгоритмах фільтрації та стабілізації яскравості відеопотоку. Продуктивність запропонованих алгоритмів перевірено на реальних даних. Алгоритми реалізовані в програмному середовищі Python з використанням функцій бібліотеки OpenCV.

Ключові слова: відеопотік; середня яскравість кадру; тренд яскравості відео; цифрова фільтрація трендів; алгоритми стабілізації яскравості.

Білозерський Владислав Олександрович – асп. каф. систем управління літальних апаратів, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Дергачов Костянтин Юрійович – канд. техн. наук, старш. наук. співроб., зав. каф. систем управління літальних апаратів, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Краснов Леонід Олександрович – канд. техн. наук, старш. наук. співроб., доц. каф. систем управління літальних апаратів, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Зимовін Анатолій Якович – канд. техн. наук, проф., проф. каф. систем управління літальних апаратів, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Попов Анатолій Владиславович – д-р техн. наук, доц., проф. каф. аерокосмічних радіоелектронних систем, Національний аерокосмічний університет ім. М. Є. Жуковського "Харківський авіаційний інститут", Харків, Україна.

Vladyslav Bilozerskyi – PhD student at the Department «Aircraft Control Systems», National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: v.o.bilozerskyi@khai.edu, ORCID: 0000-0002-5503-3163.

Kostyantyn Dergachov – Candidate of Technical Science, Senior Researcher, Head of the Department of «Aircraft Control Systems», National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: k.dergachov@khai.edu, ORCID: 0000-0002-6939-3100.

Leonid Krasnov – Candidate of Technical Science, Senior Researcher, Assistant Professor of the Department of «Aircraft Control Systems», National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: leonid.krasnov.1947@gmail.com, ORCID: 0000-0003-2607-8423.

Anatolii Zymovin – Candidate of Technical Science, Professor of the Department of «Aircraft Control Systems», National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: zim301g@gmail.com, ORCID: 0000-0001-8580-2317.

Anatoliy Popov – Doctor of Technical Science, Assistant Professor, Professor of the Aerospace Radio-electronic Systems Department, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine, e-mail: a.v.popov@khai.edu, ORCID: 0000-0003-0715-3870.