IMPLEMENTATION AND EVALUATION OF THE EFFECTIVENESS OF DYNAMIC OBJECT DETECTION TOOLS FOR MULTITHREADED AND DISTRIBUTED PROCESSING OF GRAPHICAL DATA

O. Ja. Kravets (1), I. A. Aksenov (2), I. V. Atlasov (3), P. A. Rahman (4), Yu. V. Redkin (5), O. Yu. Frantsisko (6), L. M. Bozhko (7)*

(1) Voronezh State Technical University, Voronezh; (2) Vladimir State University named after Alexander and Nikolay Stoletovs, Vladimir; (3) Moscow University of Ministry of Internal Affairs of Russian Federation named by V. Ja. Kikot, Moscow; (4) Ufa State Petroleum Technological University, Ufa; (5) Admiral Ushakov Maritime State University, Novorossiysk; (6) Kuban State Agrarian University named after I.T. Trubilin, Krasnodar; (7) Emperor Alexander I St. Petersburg State Transport University, St. Petersburg
Russian Federation

* Corresponding Author, e-mail: csit@bk.ru

Abstract: Modern monitoring and operational decision-making systems based on image series have become widely used for a wide variety of applications. At the stage of identification of graphic objects, it is established that the found dynamic object belongs to the class of objects of interest based on a comparative analysis of its contours with a given template. The article is the final in a series of articles devoted to theoretical, algorithmic and software components of the video analytics system. The description of the structural components of the video analytics system, the features of the implementation of the dynamic object detection module on video sequences, and the evaluation of the efficiency of the system is presented.

Key words: video analytics system, dynamic object detection, video sequences, implementation, machine graphics.

1. INTRODUCTION

Modern monitoring and operational decision-making systems based on image series have become widely used for a wide variety of applications. At the stage of identification of graphic objects, it is established that the found dynamic object belongs to the class of objects of interest based on a comparative analysis of its contours with a given template. The article is the final in a series of articles devoted to theoretical [1], algorithmic [2] and software components [3] of the video analytics system. The description of the structural components of the video analytics system, the features of the implementation of the dynamic object detection module on video sequences, and the evaluation of the efficiency of the system is presented.

The purpose of the article is an experimental evaluation of the effectiveness of dynamic object detection tools for multithreaded and distributed processing of graphical data.

The video analytics system shown in Figure 1 includes the following structural components: a set of video cameras, image capture and preprocessing module, dynamic object detection module, object recognition module, databases and knowledge bases, decision-making system, etc.

The author's description of the structural components of the video analytics system, the features of the implementation of the dynamic object detection module on video sequences, and the evaluation of the efficiency of the system is presented.

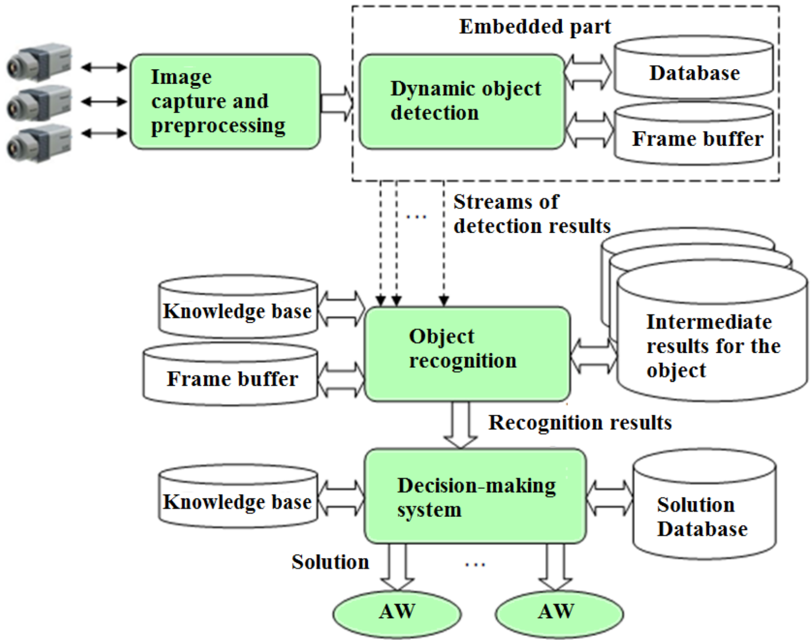

Figure 1. The video analytics system: AW - Automated workplace

2. INFORMATION ABOUT THE SOFTWARE PLATFORM

The software implementation of the embedded dynamic object detection module was created using the object-oriented C# programming language running on the platform.Net Framework 4.6, and Visual Studio 2017 development environments [4].

For ease of embedding the dynamic object detection module into the video analysis system, the implementation of the algorithmic part is included in a specially developed library of classes, the description of which is presented below.

The detection module uses a set of third-party Open Source Computer Vision [5] and Emgu Computer Vision libraries to perform image analysis and processing.

The hardware and software requirements for the dynamic object detection module and the client application for its configuration are presented in Table 1.

Table 1. Hardware and software requirements

Component	Requirements
Operating system	Windows 7, 8, 10; Windows Server 2008 and higher
The main software components	Microsoft .Net Framework 4.0 and higher; Emgu CV SDK 3.2.0
Processor	2 cores or more with a frequency of 1.6 GHz or higher
Memory (RAM)	3 GB
Free hard disk space	at least 500 MB

The main program classes present in the developed class library and their interrelations are considered in Figure 2.

The following elements are presented in the class relationship diagram:

1. "Image" is a class consisting of an image in the form of a two-dimensional array of intensities, the date and time of its creation in UTC format as the number of milliseconds from 01.01.1970. The class in question implements the "SplitIntoBlocks()" method for converting a general array of intensities into a list of "Block" classes.

2. "Block" is a class designed to store a block of pixels and consisting of the coordinates of the beginning of the block and intensity values in the form of an integer array.

3. "Coordinates" is a class that defines pixel coordinates.

4. "KeyFrame" is a class designed to describe keyframes and includes an image identifier, the date and time of its creation, as well as a list of "Block" classes.

5. "IntermediateFrame" is a class designed to describe intermediate frames and consisting of the following fields: image ID, date and time of its creation, a set of links to previous blocks stored in other intermediate/keyframes in the form of a list of classes "LinkToTheBlock", a list of new blocks in the form of a list of classes "Block".

6. "LinkToTheBlock" - a class consisting of the following fields: coordinates of the current block on the current intermediate frame, coordinates of the storage location of the block, ID of the storage location to communicate with the frame in which the block is stored.

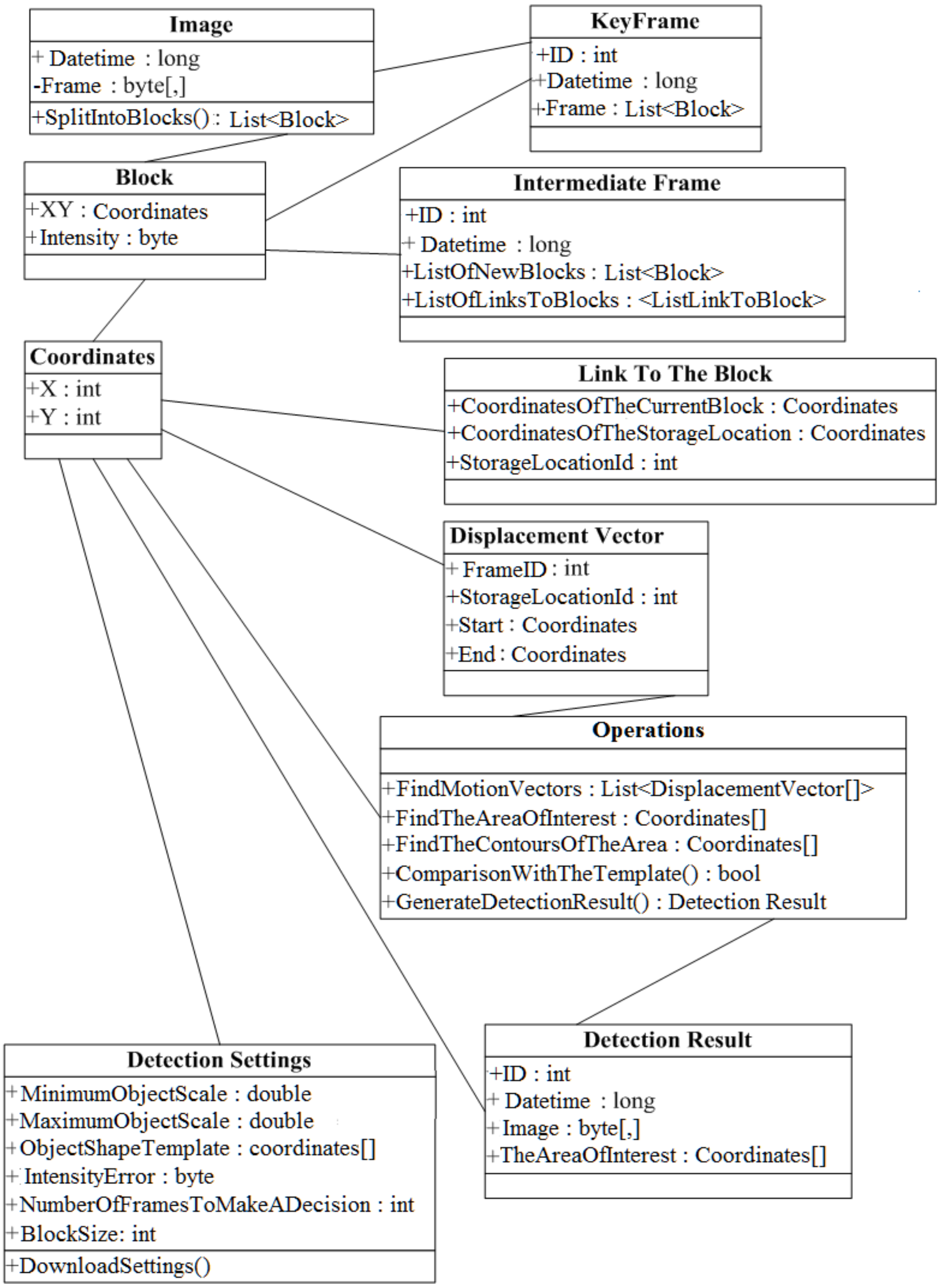

Figure 2. Diagram of the relationship of classes of the dynamic object detection module

7. "DisplacementVector" is a class designed to describe the motion vector of a block on video sequences and consisting of the following components: ID of the intermediate or keyframe where the pixel block is stored, coordinates of the beginning of the vector, coordinates of the end of the vector.

8. "DetectionSettings" - a class for storing the settings of the dynamic object detection module and including the following information: template of the object of interest, in the format of an array of pixel coordinates; the minimum and maximum scale of the found object in the images relative to the template; the intensity error, which is the threshold of the inter-pixel difference, at which the pixels are considered different; the size of one side of a square block of pixels; the number of frames for making a decision, which means the frequency of issuing detection results.

9. "DetectionResult" is a class designed to describe the result of detecting a dynamic object and containing the following information: the image on which the decision was made, coordinates of the detection zone, object ID, date and time.

3. SOFTWARE FOR CONFIGURING THE EMBEDDED DYNAMIC OBJECT DETECTION MODULE

The developed client software allows you to create the necessary detection settings for their further use in the video analytics system as input data of the built-in dynamic object detection module. The input information of the client application is the detection settings, if any, as well as the video sequence of dynamic objects to check the effectiveness of motion recognition. Figure 3 shows the main window of the client application, on which there are two buttons "Settings" and "Video Download", as well as a component for displaying video sequences and recognition results of dynamic objects.

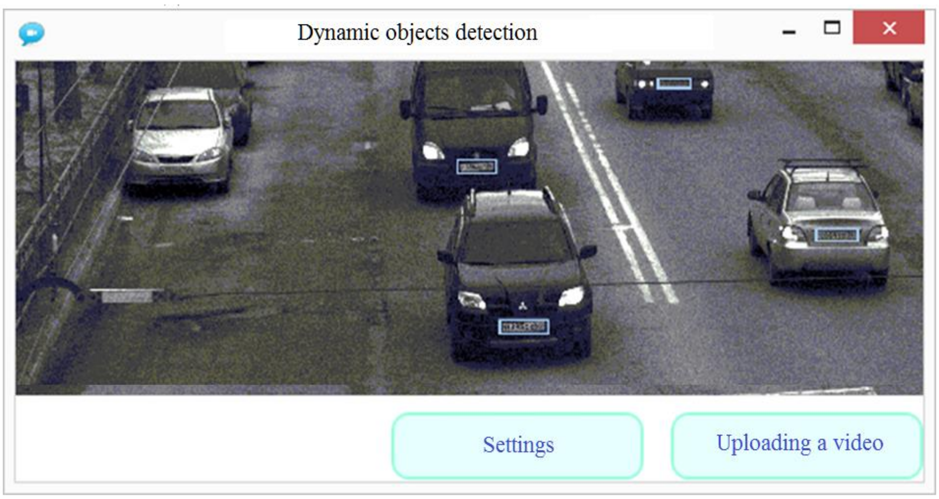

Figure 3. The main window of the client application

To be able to select a video sequence, a component has been added to the form of the main window of the application "Detecting dynamic objects" - the "Video Upload" button. Clicking on the button opens a dialog box for selecting a video file. To create or change the settings of the dynamic object detection module, a component has been added - the "Settings" button. When you click on the button, the form shown in Figure 4 opens.

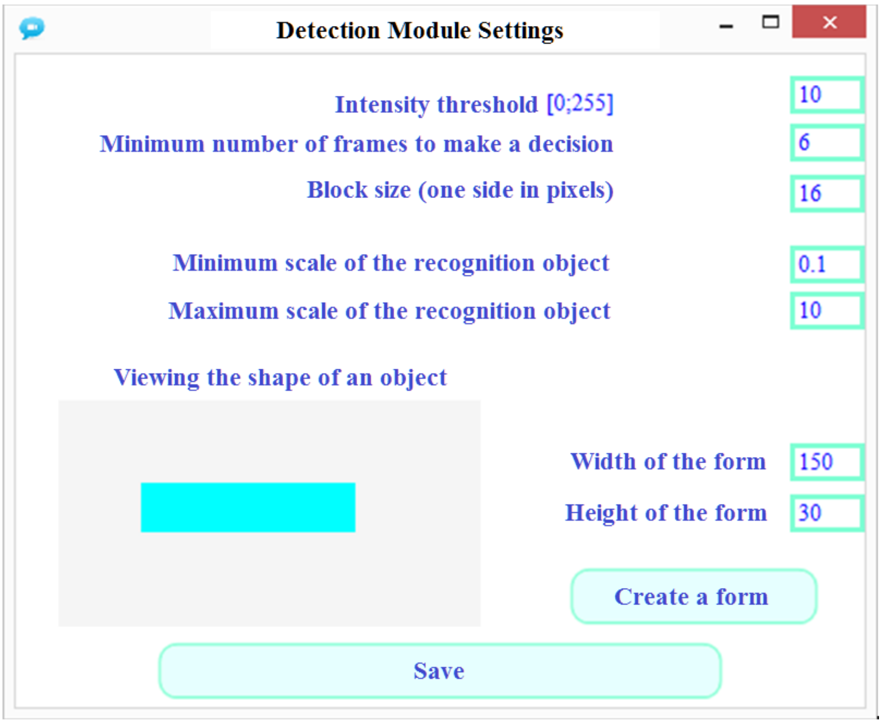

Figure 4. Detection module settings

The program form "Creating an object shape template" is designed to create a template for the shape of an object of interest and includes components for drawing the shape of an object, as well as the following buttons:

1. "Rectangle" for drawing rectangular shape,
2. "Circle" for drawing a round shape,
3. "Arbitrary shape" for drawing a shape formed from a set of points.
4. "Save" to save the drawn shape.

After closing the form in question, the transition to the "Detection Module Settings" form occurs with a changed pattern of the object template displayed on the "Object Form View" component.

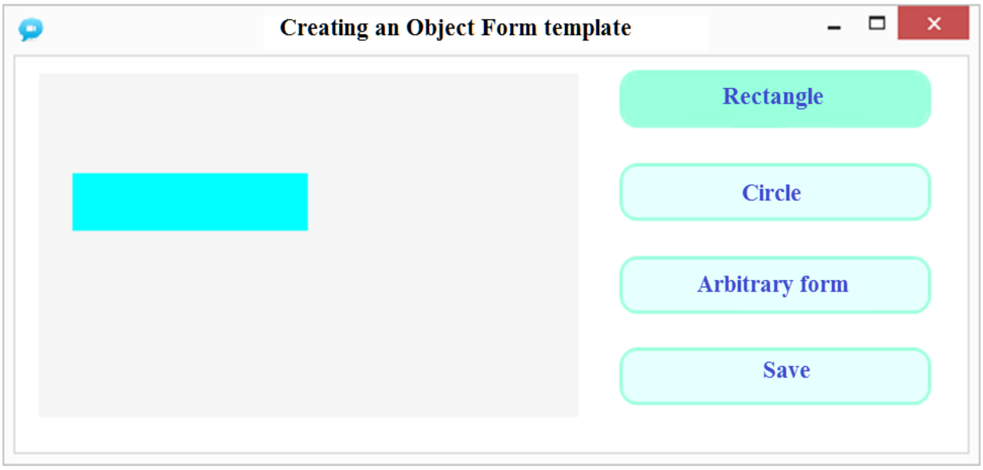

Fig. 5. Creating an object form template

To check the correctness of the saved settings, the selected video file is analyzed and the detection results are displayed in the form of selected objects on the video sequence.

Using the example of the settings and recognition of the detection module, a rectangular template corresponding to the license plate of cars was created. The minimum scale of the object of interest (number plate) in the image was 15*3 pixels, and the maximum was 1500* 300 pixels. The video sequence consisted of images with a resolution of 2048*1024. All four moving car license plates were found (Figure 3) in the video.

4. PERFORMANCE EVALUATION

To evaluate the effectiveness, a set of annotated ImageNet VID images was used, designed for processing and testing computer vision methods. The numerical experiment was carried out on 3750 video fragments from a training set with a resolution of 480*360 pixels recorded at a speed of 25 frames/sec.

Figure 6 shows graphs of the ratio of the number of processed frames per second and the accuracy of the algorithms for groups of objects. The frame frequency is located along the abscissa axis. Along the ordinate axis is a characteristic of average accuracy.

The study was conducted in two groups of 10 classes of objects in each. The first group included simple objects: a brick, a bag, a ball, a license plate, a wheel, etc. In the second group, objects with a complex structure were considered: a human face, a car including headlights, wheels, license plate, bumper, hood, etc. The detection methods selected for comparison are advanced modern developments presented at recent conferences on image processing, such as The IEEE/CVF Computer Vision and Pattern Recognition Conference [6-9]. The experiment showed that for objects of the 1st group, the proposed algorithm for detecting dynamic objects allows achieving the highest possible accuracy compared to other algorithms considered throughout the entire observation period.

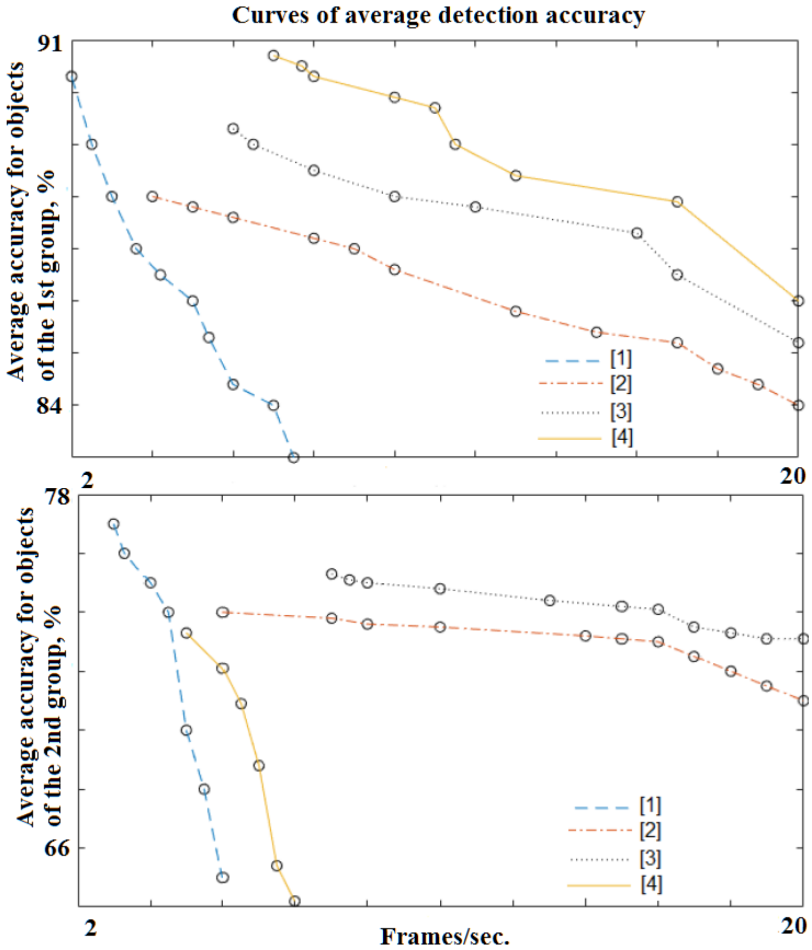

Figure 6. Graphs of average detection accuracy for objects of the 1st and 2nd groups: [1] - Dense aggregation of objects; [2] - Permitted distribution of objects; [3] - Temporary adaptive planning of key frames; [4] - Developed algorithm

As for the objects of the 2nd group, it can be concluded that the detection of objects with a complex structure is ineffective, which indicates a limitation of the scope of the proposed algorithm. Thus, the developed software of the dynamic object detection module, which is distinguished by its integration into the video analysis system, allows improving the accuracy and speed of recognition of objects with a simple structure in real time by 11%.

5. CONCLUSION

1. An embedded software module for detecting dynamic objects has been developed that implements the previously proposed algorithm for the effective organization of interaction of the software parts of the video analytics system in real time.

2. The architecture of multithreaded and distributed software processing of graphical data is proposed, which allows organizing the efficient operation of the system in real time. An improvement in the quality of recognition of objects with a simple structure in motion has been experimentally shown.

3. The software system can be used as part of a control complex in transient modes of operation of a computer network [10].

REFERENCES

[1] Ding G., Guo B.L. Motion vector estimation using line-square search block matching algorithm for video sequences. *EURASIP Journal on Advances in Signal Processing*, no.11, 2004, pp. 1750-1756.

[2] Tian N., Ling B.W.-K., Wu C., Teo K.L. Determination of fractional pixel location via an optimal group delay estimation approach. *Signal Image and Video Processing*, vol. 13, no. 2, 2019, DOI: 10.1007/s11760-019-01497-2.

[3] Gorshkov A.V., Kravets O.Ja., Aksenov I.A., Redkin Yu.V., Atlasov I.V. Architecture of a video analytics system using parallel processing. *International Journal on Information Technologies and Security*, vol. 14, no. 4, 2022, pp. 3-12.

[4] Fraser S.R.G. *Overview of the .NET Framework*. Apress, 2009, 1048 pp. https://doi.org/10.1007/978-1-4302-1054-2_1.

[5] Kaehler A., Bradski G. *Learning OpenCV 3*. O'Reilly, 2017, 1022 pp.

[6] *The IEEE / CVF Computer Vision and Pattern Recognition Conference 2019*, Long Beach, California, USA, 2019. https://cvpr2019.thecvf.com/.

[7] *The IEEE / CVF Computer Vision and Pattern Recognition Conference 2020*, Virtually, 2020. https://cvpr2020.thecvf.com/.

[8] *The IEEE / CVF Computer Vision and Pattern Recognition Conference 2021, Virtually, 2021*. https://cvpr2021.thecvf.com/.

[9] *The IEEE / CVF Computer Vision and Pattern Recognition Conference 2022*, New Orleans, Louisiana, USA, 2022. https://cvpr2022.thecvf.com/.

[10] Kravets O.Ja., Atlasov I.V., Aksenov I.A., Molchan A.S., Frantsisko O.Yu., Rahman P.A. Increasing efficiency of routing in transient modes of computer network operation. *International Journal on Information Technologies and Security*, vol. 13, no. 2, 2021, pp. 3-14.

Information about the authors:

Oleg Jakovlevich Kravets – Dr. Sci (IT), professor of Voronezh State Technical University, areas of scientific research - system analysis, optimization, simulation of complex objects

Ilia Antonovich Aksenov – PhD, associate professor of Vladimir State University named after Alexander and Nikolay Stoletovs, areas of scientific research - application of information technologies in economic systems

Igor Victorovich Atlasov – Dr. Sci (Phys.-Math.), professor of Moscow University of Ministry of Internal Affairs of Russian Federation named by V. Ja. Kikot, areas of scientific research - system analysis, optimization, simulation of complex objects

Pavel Azizurovich Rahman – PhD, associate professor of Ufa State Petroleum Technological University, areas of scientific research - information technologies, computer networks, reliability modelling and engineering

Yuriy Viktorovich Redkin – PhD, associate professor of Admiral Ushakov Maritime State University, Novorossiysk, areas of scientific research - wireless networks, digital signal processing, control and data processing systems

Olga Yurievna Frantsisko – PhD, Associate professor of Kuban State Agrarian University named after I.T. Trubilin, Krasnodar, areas of scientific research - economic and mathematical modeling, scenario approach, processing industries of agricultural enterprises, information technology

Lesya Mikhailovna Bozhko – Dr. Sci (Economy), professor of Emperor Alexander I St. Petersburg State Transport University, St. Petersburg, areas of scientific research – change management, market research, strategic management, project management

Manuscript received on 25 November 2022