Scientific Visualization, 2025, volume 17, number 2, pages 137 - 146, DOI: 10.26583/sv.17.2.10

Simulation and Visualization of Optical Distortions for Virtual Surveillance Devices

A.V. Maltsev[1,A]

Scientific Research Institute for System Analysis of the National Research Centre“Kurchatov Institute”

[1] ORCID: 0000-0003-1776-814X, avmaltcev@mail.ru

Abstract

The paper describes methods and approaches for imitation of optical effects when rendering images from models of various surveillance device in a three-dimensional virtual environment. This allows for the possibility expansion of such modeling by ensuring coverage of a wide range of real devices as prototypes for the virtual analogues being created. An original approach to two-stage image synthesis with radial distortion implementation is proposed, as well as methods and algorithms for simulation of lateral and longitudinal chromatic aberrations. Developed solutions are based on the use of modern multi-core graphics processing units and shader rendering technology. They provide visualization of video streams from simulated surveillance devices in real time. Software modules were created based on developed methods and approaches. Approbation of them was carried out in virtual environment system VirSim [1] and showed adequacy and effectiveness of proposed solutions when using in virtual environment systems and training complexes.

Keywords: visualization, virtual environment, simulation, surveillance devices, lenses, radial distortion, chromatic aberration.

1. Introduction

At present, elements of a virtual environment, generated by means of a computer, and training complexes built on their basis are used in various areas of human activity [2-8]. This approach involves the complete or partial replacement of real-life objects with their three-dimensional virtual models created in such modeling systems as 3ds Max [9], Maya [10], Blender [11], etc. In the context of training operators to control complex technical means, such replacement has a number of advantages. Among them, there are the absence of the need to purchase expensive equipment for training purposes and, accordingly, the impossibility of its breakdown in the event of erroneous actions during training, as well as relatively easy scalability for a different number of trainees.

One of the important elements of virtual environment are surveillance devices [12] that simulate real devices placed on various fixed (walls, pillars, columns) and moving objects (cars, robots, aircraft, etc.). Video signal received from them provides the operator with the ability to view the surrounding environment, make decisions and visually monitor the results of some process control, such as the movement of objects. The effectiveness of human training using a synthesized virtual environment, as well as the correctness of the skills imparted, largely depend on the similarity of images from virtual surveillance devices and those that he could see using real devices.

Surveillance devices are technically complex equipment consisting of a large number of electronic, mechanical and optical elements. Each of them can introduce different types of effects and distortions into the resulting image. This may be caused by the characteristics and quality of used elements, their operating scheme as a single system, external interference,

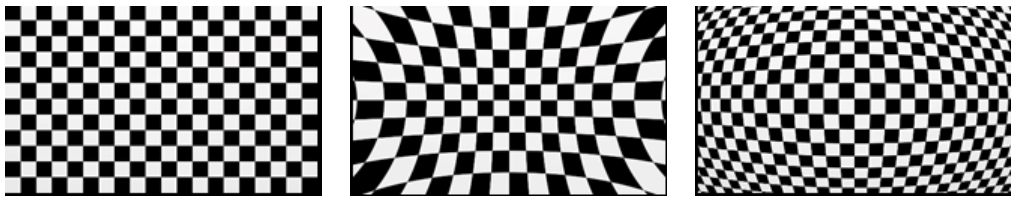
equipment failure and a number of other factors. Therefore, in order to develop the correct skills of the trainees for technical equipment control, by means of training systems based on three-dimensional virtual environment, it is advisable to simulate effects and distortions on visualized images from virtual surveillance devices, which are inherent to their real prototypes.

Active research is being conducted in the area of developing approaches for simulation of video surveillance equipment in virtual environment systems, including methods for visualization of images received from them, taking into account physical characteristics and possible distortions of different natures. Thus, a model of virtual camera with a thin lens is described in [13]. It uses such physical parameters as focal length and aperture number. The authors of paper [14] propose a solution for distributed implementation of the depth of field effect on images from virtual surveillance devices using multi-core graphics processing unit (GPU). Articles [15, 16] consider the problem of rendering three-dimensional scenes in extended dynamic range of brightness (HDR) with subsequent conversion to a range that can be displayed by output devices without HDR technology support. The paper [17] describes methods and algorithms for real-time visualization of virtual environment with imitation of external influence on the camera, consisting of rain precipitation on the glass of its lens. Approaches and methods for implementing such distortions as white noise, synchronization breakdown, illumination and overexposure are presented in [18].

This paper proposes original methods and algorithms for virtual prototype simulation of real surveillance devices in three-dimensional scenes with imitation of video signal distortions associated with the design features of optical elements and expressed in the effect of distortion and two types of chromatic aberration. They are based on the use of distributed computing using multi-core GPU. The novelty of the developed solutions is original approach to barrel distortion implementation based on correction of virtual camera's field of view, ensuring full and effective synthesis of frame image, as well as the author's method for simulation of chromatic aberrations on graphics processor in real time.

2. Radial distortion

One of the most common types of image distortion in surveillance devices is radial distortion, which occurs in optical systems of lenses. This is expressed in a change of the linear magnification coefficient when displayed objects move away from the optical axis. As a result, the geometric similarity between real object and its image is violated. The distortion effect turns straight lines of objects that do not intersect the optical axis into arcs. If they are concave relative to the center of the image, then pincushion distortion occurs, and it is barrel distortion if convex. Figure 1 clearly demonstrates both of these cases for image of surface with a checkerboard texture.



a) b) c)

Fig. 1. Original image (a), pincushion (b) and barrel (c) distortions.

In this paper, radial distortion implementation for images obtained from virtual surveillance devices is based on the division model. This model was first proposed by Reimar Lenz in 1987, but attracted more attention after the research [19]. It is formulated as follows:

$$L(r) = \frac{1}{1 + \sum_{i=1}^{n} k_i r^{2i}} \tag{1}$$

where k_i – are radial distortion coefficients, $n \in \mathbb{N}$ – is the number of terms in the sum, r – is the distance from a point of distorted image to its center. To achieve a realistic result, it is sufficient to limit n to 2, i.e. use only parameters k_1 and k_2. Let the origins of coordinate systems U_0V_0 and UV be respectively located at the center of undistorted and distorted images, the axes U_0 and U are directed horizontally to the right, the axes V_0 and V are directed vertically upwards, and the coordinate values are specified on the interval [-0.5, 0.5]. Then, within the framework of model (1), an arbitrary point P with coordinates (u, v) on the distorted image corresponds to a point P_o with coordinates (u_0, v_0) on original image, which is defined as

$$P_o = L(r) \cdot P = \frac{P}{1 + k_1 r^2 + k_2 r^4}, r = \sqrt{u^2 + v^2} \,. \quad (2)$$

To simulate radial distortion, we use an approach based on post-processing technology of rendered frame. It includes two stages. At the first of them, image from virtual model of surveillance device is visualized without distortion. Rendering result is written to special off-screen buffer (FBO) with the same resolution as the current framebuffer. At the second stage, post-processing shader performs distributed shading of framebuffer itself on the GPU, using created FBO as the input texture T_0, which stores undistorted image.

The essence of the shader's work is as follows. Let the frame size be W x H pixels, and each pixel to be shaded has integer coordinates $x \in [0, W-1]$ and $y \in [0, H-1]$ in a coordinate system XY with the origin at the lower left corner of the frame and the axes pointing left (X) and up (Y). To convert x and y to the above-mentioned coordinate system UV, it is necessary to scale the axes and shift the origin of the system XY:

$$u = \frac{x}{W} - 0.5 \,, \; v = \frac{y}{H} - 0.5 \,. \quad (3)$$

Next, the shader uses equation (2) to compute coordinates u_0 and v_0 on undistorted image corresponding to u and v, and transforms them into coordinates

$$s_0 = u_0 + \; 0.5, t_0 = v_0 + \; 0.5$$

with values from the interval [0.0, 1.0] to access texture T_0 and obtain the desired color of pixel (x, y) being processed.

Described solution is well suited for the implementation of pincushion distortion, in which $k_1 > 0$ and value of function $L(r) < 1$. In this case, original image is stretched and visible space is "pushed" outside the frame. For each point P inside distorted image, there is a corresponding point P_o in texture T_0. With barrel distortion ($k_1 < 0$, $L(r) > 1$), the opposite effect occurs. Original image is pulled toward the optical axis, and areas of space that were previously outside the frame appear at its edges. The problem is that there is no information about these areas in original image and texture T_0, which means the shader cannot compute their color.

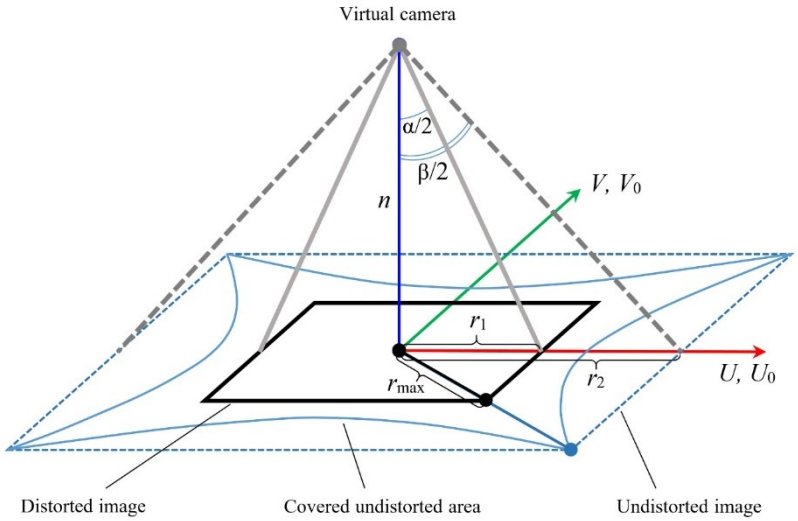


Fig. 2. The field of view correction for virtual camera.

To solve this problem, we propose an original approach based on correction of virtual camera's field of view (FOV) at the stage of undistorted image visualization, if parameter k_1 has a negative value, with subsequent coordinate scaling in the system UV at the post-processing stage. Let's consider it in more detail. Let α be current angle of horizontal FOV for used virtual camera, β be desired angle at which rendered undistorted image will contain the necessary information to execute our post-processing shader, and n be the distance to the near clipping plane of the camera. Figure 2 shows the combination of distorted and undistorted images relative to their coordinate systems UV and U_0V_0, as well as horizontal FOVs of virtual camera with angles α and β. They half-angle tangents are equal

$$\operatorname{tg}\frac{\alpha}{2}=\frac{r_1}{n},\ \operatorname{tg}\frac{\beta}{2}=\frac{r_2}{n}=\frac{k\cdot r_1}{n},$$

where r_1, r_2 are distances along axes U and U_0 from image centers to their vertical boundaries, k is unknown scaling factor. To compute k, let us recall that with barrel distortion a value of k_1 is negative. When moving from distorted image to undistorted one, radial stretching occurs according to equation 2. The distance r from an arbitrary point to the center increases by $L(r)$ times. The maximum stretching occurs at points (-0.5, -0.5), (-0.5, 0.5), (0.5, 0.5), (0.5, -0.5), corresponding to the frame's corners and located at the distance $r_{\max}$ from its center (Figure 2). Thus, in order for undistorted frame to contain all the points needed to form the distorted one, the factor k must be computed as

$$k=L\left(r_{max}\right), r_{max}\approx\ 0.7071.$$

Then required angle β of the camera's horizontal FOV for the stage of undistorted image visualization is determined by the following equations:

$$\operatorname{tg}\frac{\beta}{2}=L\left(r_{\max}\right)\cdot\operatorname{tg}\frac{\alpha}{2},$$

$$\beta=2\cdot\operatorname{arctg}\left(L\left(r_{\max}\right)\cdot\operatorname{tg}\frac{\alpha}{2}\right).$$

Next, obtained angle β is set for the corresponding virtual camera, and the scene is rendered to FBO buffer (texture T_0), which resolution coincides with the frame size, namely W x H.

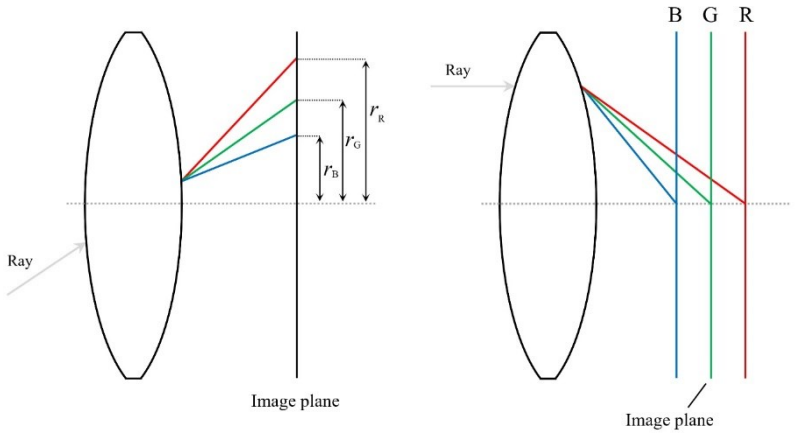
Taking into account the performed increase of the camera's FOV, as well as the absence of a change in actual sizes (in pixels) of FBO buffer and texture T_0 containing undistorted image, at the post-processing stage it is necessary to scale coordinates of point P_o obtained using equation (2). Such scaling should have a factor equal to $1/k$. Then

$$P_0=\frac{L(r)}{L\left(r_{\max}\right)}\cdot P,\ L(r)=\frac{1}{1+k_1r^2+k_2r^4},\ r=\sqrt{u^2+v^2}\ .$$

3. Chromatic aberration

Another common type of distortion in surveillance devices is chromatic aberration. The reason for its appearance is that light waves of different lengths are refracted differently in the objective lenses and, as a result, are focused at various points. This is expressed as color halos and fringing in the image, leading to a loss of clarity. There are two types of chromatic aberrations: transverse (lateral) and longitudinal (axial). They have different characteristics, but can happen simultaneously.

Transverse chromatic aberration occurs when light waves of different lengths are focused at various points in the focal (image) plane. Left side of Figure 3 (a) demonstrates this case.



a) b)

Fig. 3. Transverse (a) and longitudinal (b) chromatic aberrations.

Parameters r_R, r_G and r_B denote the distances from the points corresponding to waves of red, green and blue spectrum to the optical axis. In this case, image scale is not the same for different color channels of the obtained frame. This type of distortion does not depend on used aperture number and focal length. Longitudinal chromatic aberration occurs when different wavelengths of light are focused at various distances from the lens (Figure 3, b). Since many cameras focus using green color channel, the resulting image will show the majority of defocus in red and blue channels. Typically, this distortion becomes less noticeable when stopping down to f/2.8-f/4.

The approach to simulation of chromatic aberrations proposed in this paper, as in the case of radial distortion, includes two stages. The first is to render a virtual scene without aberrations to the texture T_0 using FBO buffer. At the second stage, distributed post-processing of the obtained image is performed using special fragment shader. Next, we consider the basic principles of implementing this shader in more detail. Let the size of the image being formed be W x H (in pixels), and each pixel to be shaded has coordinates $x \in [0, W\text{-}1]$ and $y \in [0, H\text{-}1]$ in the system XY located in the lower left corner of the frame. As mentioned earlier, with transverse chromatic aberration, each of color channels R, G and B of the image obtained by a real camera has some own scale relative to dimensions of surveillance device's sensor (electronic matrix). Therefore, to compute a color of a pixel (x, y), it is necessary to find the corresponding points in R, G and B channels of the texture T_0. Note that all color channels of T_0 initially have the same image scale. Thus, to solve the task, we will not compute coefficients of transition from original texture coordinates to the scaled ones, but vice versa, from scaled coordinates to the original ones.

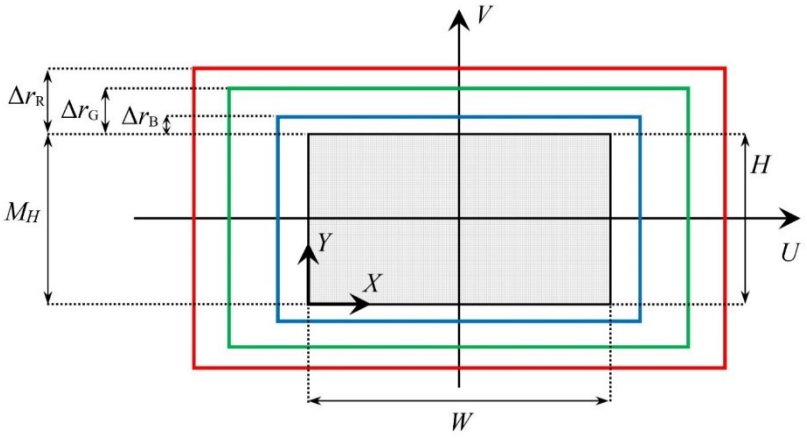


Fig. 4. Example of image scale for R, G and B color channels relative to the sensor size when transverse chromatic aberration occurs.

Let us denote the width and height of modeled surveillance device's sensor as M_W and M_H, respectively, and the distances between the conditional upper boundaries of scaled color channels R, G, B and the upper boundary of this sensor as Δr_R, Δr_G, Δr_B, specified in millimeters (Figure 4). Then scaling factors k_R, k_G, k_B for red, green and blue channels relative to the given matrix size are computed using following equations:

$$k_R = \frac{M_H + 2 \cdot \Delta r_R}{M_H},\ k_G = \frac{M_H + 2 \cdot \Delta r_G}{M_H},\ k_B = \frac{M_H + 2 \cdot \Delta r_B}{M_H}.$$

Resizing of the image in color channels is radial. Therefore, the shader must first transform the given x and y coordinates of the pixel to the system UV with the origin at the frame center described in the previous section. Coordinates u and v are computed by means of equations (3). The point (u, v) defines a position in scaled channels. To move to the points in original undistorted image stored in the texture T_0, we need to apply coefficients that are the inverse of k_R, k_G, k_B and shift the origin of coordinate system to the lower left corner. Then texture coordinates for red, blue and green channels of T_0 will be determined as

$$(s_R, t_R) = \left(k_R^{-1}u + 0.5,\ k_R^{-1}v + 0.5\right),$$
$$(s_G, t_G) = \left(k_G^{-1}u + 0.5,\ k_G^{-1}v + 0.5\right),$$
$$(s_B, t_B) = \left(k_B^{-1}u + 0.5,\ k_B^{-1}v + 0.5\right).$$

By sampling a color from each channel of the texture T_0 separately with use of the obtained coordinates, the shader forms color of the pixel (x, y) under transverse chromatic aberration conditions.

To take into account the influence of longitudinal chromatic aberration, it is necessary to apply a Gaussian filter when sampling from R and B color channels, which will simulate the defocusing of these channels. The effect power depends on the filter size. We will remind that G channel retains sharpness, since in most devices it is used to focus the image.

4. Results

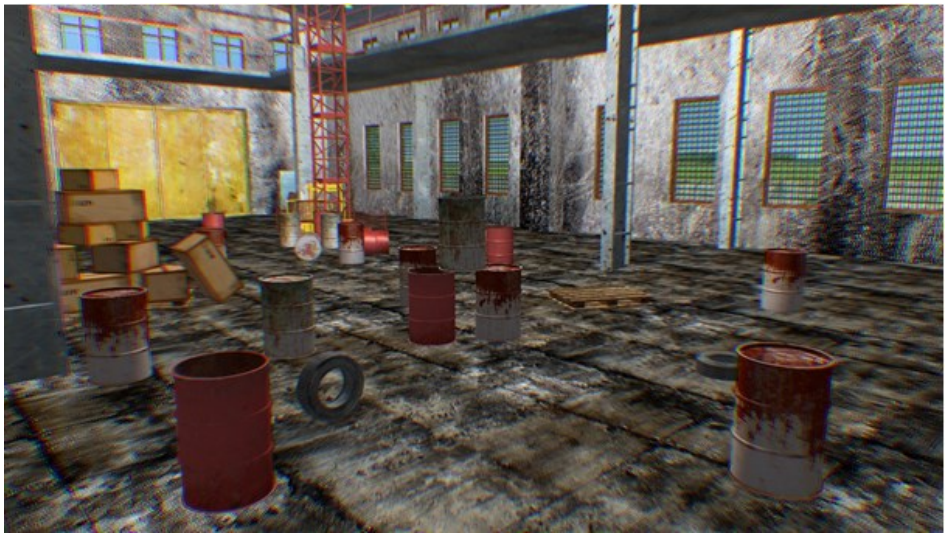
Proposed methods and approaches for optical distortion simulation of a video signal received from virtual surveillance devices were implemented in virtual environment system VirSim [1] developed at the NRC "Kurchatov institute" - SRISA. Testing of these solutions was carried out using a scene of the virtual polygon, the visualization result of which (without optical distortions) is presented in Figure 5. For this, a camera with adjustable physical parameters was placed in the scene, including radial distortion and chromatic aberration coefficients, as well as the focal length and aperture value of the lens. Performed tests involved both independent application of each type of distortion and their combination for the virtual camera mentioned above.

Fig. 5. Visualization for a scene of virtual polygon without optical distortions.

a)

b)

Fig. 6. Visualization of the scene with simulation of barrel (a) and pincushion (b) distortions.



a)

b)

Fig. 7. Visualization of the scene with simulation of transverse (a) and longitudinal (b) chromatic aberration.

5. Conclusions

This paper presents original distributed methods for implementing radial distortion and chromatic aberrations in images of three-dimensional scenes obtained using virtual cameras. They allow expanding the range of surveillance devices modeled in virtual environment. Results obtained in the paper can be used in software development for training complexes and virtual environment systems.

6. Acknowledgements

The publication is made within the state task of Scientific Research Institute for System Analysis of the National Research Centre “Kurchatov Institute” on topic No. FNEF-2024-0002 “Mathematical modeling of multiscale dynamic processes and virtual environment systems”.

References

1. Mikhaylyuk M.V., Maltsev A.V., Timokhin P.Ju., Strashnov E.V., Krjuchkov B.I., Usov V.M. Sistema virtual'nogo okruzhenija VirSim dlja imitacionno-trenazhernyh kompleksov podgotovki kosmonavtov [The VirSim virtual environment system for the simulation com-

plexes of cosmonaut training] // Pilotiruemye polety v kosmos, 2020, Vol. 4, No. 37, pp. 72-95 [in Russian].

2. Maltsev A.V. Integration of Physical Reality Objects with Their 3D Models Visualized in Virtual Environment Systems // Scientific Visualization, 2024, Vol. 16, No. 2, pp. 97-105.

3. Hülsmann F., Mattar N., Fröhlich J., Wachsmuth I. Simulating Wind and Warmth in Virtual Reality: Conception, Realization and Evaluation for a CAVE Environment // Journal of Virtual Reality and Broadcasting, 2014, Vol. 11, No. 10, pp. 1-21.

4. Bruguera M.B., Ilk V., Ruber S., Ewald R. Use of virtual reality for astronaut training in future space missions – spacecraft piloting for the Lunar Orbital Platform – Gateway (LOP-G) // 70th International Astronautics Congress, Washington D.C., 2019.

5. Maltsev A.V., Strashnov E.V., Mikhaylyuk M.V. Methods and technologies of cosmonaut rescue simulation in virtual environment systems // Scientific Visualization, 2021, Vol. 13, No. 4, pp. 52-65.

6. Mikhaylyuk M.V., Timokhin P.Yu. Memory-effective methods and algorithms of shader visualization of digital core material model // Scientific Visualization, 2019, Vol. 11, No. 5, pp. 1-11.

7. Pezent E., Macklin A., Yau J.M., Colonnese N., O'Malley M.K. Multisensory Pseudo-Haptics for Rendering Manual Interactions with Virtual Objects // Advanced Intelligent Systems, 2023, Vol. 5, pp. 1-13.

8. Garcia A.D., Schlueter J., Paddock E. Training astronauts using hardware-in-the-loop simulations and virtual reality // AIAA SciTech Forum, Orlando, FL, 2020.

9. 3ds Max, https://www.autodesk.com/products/3ds-max/overview.

10. Maya, https://www.autodesk.com/products/maya/overview.

11. Blender, https://www.blender.org/get-involved/documentation/.

12. Maltsev A.V. Computer simulation of video surveillance complexes in virtual environment systems // Scientific Visualization, 2022, Vol. 14, No. 2, pp. 88-97.

13. Brian A. Barsky, Daniel R. Horn, Stanley A. Klein, Jeffrey A. Pang, Meng Yu. Camera models and optical systems used in computer graphics: part I, object-based techniques // In Proceedings of the 2003 international conference on Computational science and its applications, 2003, pp. 246-255.

14. Maltsev A.V., Torgashev M.A. Raspredelennoe modelirovanie glubiny rezkosti pri vizualizatsii trekhmernykh stsen na GPU [Distributed simulation of DOF when rendering virtual scenes on GPU] // Trudy NIISI RAN, 2019, Vol. 9, No. 5, pp. 100-104 [in Russian].

15. Timokhin P.Yu., Torgashev M.A. Vizualizatsiia kosmicheskikh stsen v rasshirennom diapazone v rezhime realnogo vremeni [Real-Time Visualization of Space Scenes in High Dynamic Range] // Informatsionnye tekhnologii, 2014, No. 12, pp. 53-60 [in Russian].

16. M.V. Mikhaylyuk, P.Y. Timokhin, and M.A. Torgashev. The Method of Real-Time Implementation of Tone Mapping and Bloom Effect // Programming and Computer Software, 2015, Vol. 41, No. 5, pp. 289–294.

17. Maltsev A.V. Sintez izobrazhenii trekhmernykh stsen s modelirovaniem kapel dozhdia na obieektive virtualnoi kamery [Image Synthesis of 3D Scenes with Simulation of Raindrops on Virtual Camera's Lens] // Trudy NIISI RAN, 2021, Vol. 11, No. 3, pp. 36-40 [in Russian].

18. Maltsev A.V. Metody modelirovaniia svetovykh effektov i iskazhenii videosignala v virtualnykh sredstvakh nabliudeniia [Methods for simulation of light effects and video signal distortions in virtual surveillance devices] // Informatsionnye tekhnologii i vychislitelnye sistemy, 2019, No. 2, pp. 54-61 [in Russian].

19. Fitzgibbon A.W. Simultaneous linear estimation of multiple view geometry and lens distortion // Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2001, pp. 125-132.