

A Technology to Synthesize 360-Degree Video Based on Regular Dodecahedron in Virtual Environment Systems*

Petr Timokhin¹[0000-0002-0718-1436], Mikhail Mikhaylyuk²[0000-0002-7793-080X],
and Klim Panteley³[0000-0001-9281-2396]

Federal State Institution «Scientific Research Institute for System Analysis of the Russian Academy of Sciences», Moscow, Russia

¹webpismo@yahoo.de, ²mix@niisi.ras.ru, ³kpanteley@mail.ru

Abstract. The paper proposes a new technology of creating panoramic video with a 360-degree view based on virtual environment projection on regular dodecahedron. The key idea consists in constructing of inner dodecahedron surface (observed by the viewer) composed of virtual environment snapshots obtained by twelve identical virtual cameras. A method to calculate such cameras' projection and orientation parameters based on "golden rectangles" geometry as well as a method to calculate snapshots position around the observer ensuring synthesis of continuous 360-panorama are developed. The technology and the methods were implemented in software complex and tested on the task of virtual observing the Earth from space. The research findings can be applied in virtual environment systems, video simulators, scientific visualization, virtual laboratories, etc.

Keywords: Visualization, Virtual Environment, Regular Dodecahedron, Video 360, Panoramic Mapping, GPU.

1 Introduction

In modern human activities the importance of the visualization of virtual prototypes of real objects and phenomena is increasingly grown. In particular, it's especially required for training specialists in space and aviation industries, medicine, nuclear industry and other areas, where the mistake cost is extremely high [1-3]. The effectiveness of working with virtual prototypes is significantly increased if an observer experiences a feeling of being present in virtual environment. It is important that observer will have a possibility not only to sense such visual experience but also to reproduce it. For instance, this possibility allows to self-analyze the correctness of actions in virtual environment

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

* The publication is made within the state task on carrying out basic scientific researches (GP 14) on topic (project) "34.9. Virtual environment systems: technologies, methods and algorithms of mathematical modeling and visualization" (0065-2019-0012).

or demonstrate results to expert community. To implement such possibility, new technologies and formats of video recording the virtual environment are actively researched and developed. The examples are the technology of creating stereo presentations using autostereoscopic monitors [4] and the technology for deferred synthesis of 4K stereo clips [5]. In recent years, a technology of 360-degree video [6] has been developed rapidly, which point is omnidirectional capture of the environment (with covering imaginary sphere surrounding the observer). When playing such a video, the observer can change view direction and sense an effect of immersion into visualized environment.

Two major approaches to create 360-video can be highlighted. At the first approach spherical panorama is unfolded onto a plane directly using some cylindrical cartographic projection (equirectangular, Lambert equal-area) [7, 8]. At the second approach panorama is projected on the faces of some convex polyhedron (frustum, cube, rhombic dodecahedron) [9] which is unfolded onto a plane. Thanks to the equivalency of cube faces (there are no poles with expressed distortions as at the first approach) cubemap projection has been taken widely into use. However, projection detail degree in cube face centers is almost twice as smaller as near the edges, because of poor sphere approximation (just by 6 faces, 90 degrees each). To avoid blurring the panorama, the total cubemap resolution should be increased. That's leading to increase of the redundancy of video stream bitrate preventing playback smoothness.

Desire to reduce projection non-uniformity preserving its universality inspired us to use *regular dodecahedron* for virtual environment footage. Compared to the cube the dodecahedron has twice as much faces, therefore, its form is closer to a sphere. The dodecahedron is also the only Platonic solid in which all plane angles between edges at a vertex are obtuse (108 degrees). Owing to this the dodecahedron has more balanced (and less visible) projection non-uniformity than the Platonic solids with triangle faces (tetrahedron, octahedron, and icosahedron). The problem is that in virtual world (as in real) camera can make rectangular snapshots, and dodecahedron faces are regular pentagons as we know. Therefore, it poses a challenge of constructing virtual environment projection onto regular dodecahedron ensuring the visualization of continuous 360-panorama. In the paper an effective technology to solve this task is proposed, which is based on original scheme to locate virtual cameras and virtual environment snapshots obtained by them. The solution is implemented using the C++ language and OpenGL graphics library.

2 Constructing the projection onto regular dodecahedron

Consider the task of creating 360-video for the observer being movable in virtual environment. For each frame of the 360-video we will project virtual environment onto the imaginary dodecahedron surrounding the observer (see Fig. 1). In order to construct such projection we place twelve identical virtual cameras (C_0, \dots, C_{11}) with symmetrical frustums in observer position P_V , direct them into the faces (pentagons) centers of the imaginary dodecahedron and make snapshots of virtual environment. Herewith, the orientation of the imaginary dodecahedron is tightly bound to observer's view vector \mathbf{v} and «up» vector \mathbf{u} , as shown in Figure 1.

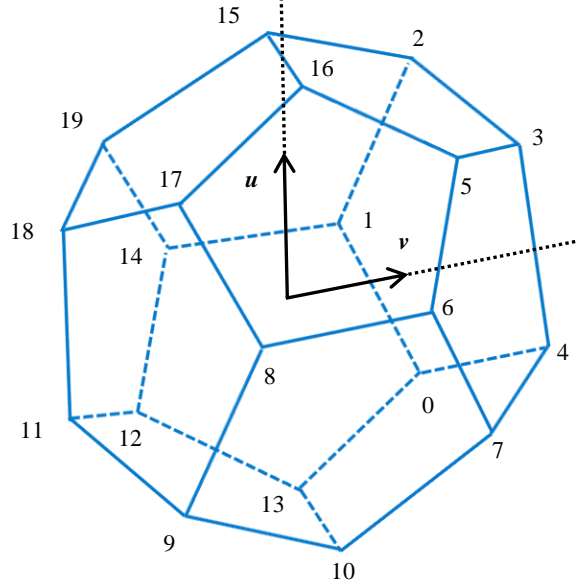


Fig. 1. Imaginary dodecahedron surrounding the observer

The key point of implementation of this scheme is calculating C_i camera's projection and orientation parameters providing the dodecahedron face (pentagon) to be inscribed into camera's rectangular viewport, namely: 1) calculating vertical FOV (field of view) angle γ and viewport *aspect* (the ratio of its width to its height), and 2) calculating the coordinates of view v_i and «up» u_i vectors of camera C_i in *World Coordinate System* (WCS). Let's consider these stages in more detail.

2.1 Calculation of the aspect and vertical FOV

As noted above, virtual environment projection onto a pentagon of imaginary dodecahedron is done by individual camera C_i . The camera has symmetrical frustum vertically and horizontally. This is necessary in order to scale increase (resulting from perspective projection) would be similar at all directions from pentagon center to its sides (see Fig. 2), and snapshots captured by adjacent cameras would be docked to each other in the same scale. For this reason, a camera with oblique frustum can't be used here, since in that case, the lower half of pentagon will be of larger scale than the upper one.

Figure 2 shows, that camera's C_i viewport sizes must be proportionate to the sizes of the rectangle $ABCD$. Its center O coincides the pentagon $MUSTQ$ center, $AB = CD = 2R_{penta}$, $BC = AD = d$, and AD is distanced from MQ by $h = R_{penta} - r_{penta}$, where R_{penta} , r_{penta} are radii of circumscribed and inscribed circle of the pentagon, and d is pentagon diagonal [10]:

$$R_{penta} = (MQ/2)\operatorname{cosec}(\pi/5), \quad r_{penta} = (MQ/2)\operatorname{ctg}(\pi/5), \quad d = 2MQ\sin(3\pi/10). \quad (1)$$

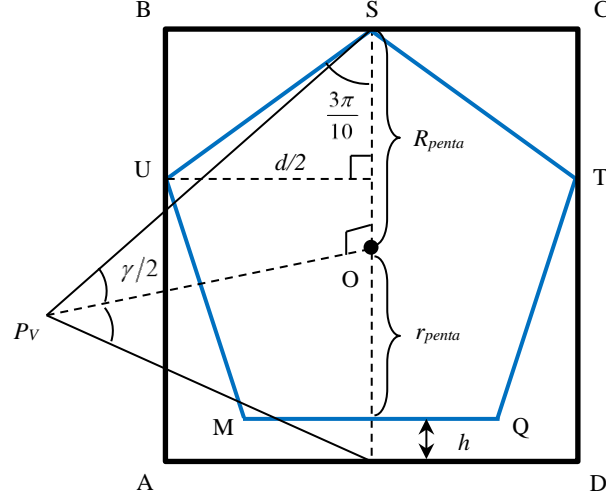


Fig. 2. Placing the pentagon into the viewport of the camera C_i

Then *aspect* value of camera's C_i viewport can be expressed as

$$\text{aspect} = \frac{BC}{AB} = \frac{d}{2R_{penta}} = \frac{2\sin(3\pi/10)}{\text{cosec}(\pi/5)} \approx 0.951. \quad (2)$$

To find camera's C_i angle γ , consider the right triangle P_vOS . Its side P_vO is the radius of inscribed sphere of the dodecahedron [10], and is calculated as

$$P_vO = (MQ/2)\text{ctg}(\pi/5)\text{tg}(\theta/2), \quad (3)$$

where $\theta = 2\arcsin(\cos(\pi/3)/\sin(\pi/5)) \approx 116.565^\circ$ is dihedral angle of regular dodecahedron. From triangle P_vOS the expression for $\text{tg}(\gamma/2)$ can be written:

$$\text{tg}(\gamma/2) = \frac{R_{penta}}{P_vO} = \frac{\text{cosec}(\pi/5)}{\text{ctg}(\pi/5)\text{tg}(\theta/2)} = \frac{1}{\cos(\pi/5)\text{tg}(\theta/2)} = \sec(\pi/5)\text{ctg}(\theta/2), \quad (4)$$

from where we get the expression for the required angle γ :

$$\gamma = 2\arctg(\sec(\pi/5)\text{ctg}(\theta/2)) \approx 74,8^\circ. \quad (5)$$

Obtained *aspect* and γ values are set for each C_i camera using *gluPerspective* operator of OpenGL library.

2.2 Calculating the coordinates of view and «up» vectors of camera C_i

After we have set projection parameters (common to all cameras C_0, \dots, C_{11}) calculating the coordinates of v_i view and u_i «up» vectors for each C_i camera is required. These vectors direct the camera into its face of imaginary dodecahedron, as shown on Fig. 2.

At first, the correspondence between cameras C_0, \dots, C_{11} and dodecahedron faces is set. We enumerate the dodecahedron vertices as shown in Figure 1 and write the sequence H of pentagons through them: 0 - {4, 7, 10, 13, 0}, 1 - {0, 1, 2, 3, 4}, 2 - {4, 3, 5, 6, 7}, 3 - {7, 6, 8, 9, 10}, 4 - {10, 9, 11, 12, 13}, 5 - {13, 12, 14, 1, 0}, 6 - {12, 11, 18, 19, 14}, 7 - {1, 14, 19, 15, 2}, 8 - {3, 2, 15, 16, 5}, 9 - {6, 5, 16, 17, 8}, 10 - {9, 8, 17, 18, 11}, 11 - {18, 17, 16, 15, 19}. In order to calculate v_i and u_i vectors we'll need the coordinates of dodecahedron faces centers. For this, the calculation of dodecahedron vertices coordinates in WCS system will be discussed in more depth.

In this work, dodecahedron vertices coordinates calculating is done using “golden rectangles” [11] – three mutually perpendicular rectangles which size related to φ number (see Fig. 3). The number φ is “golden ratio” of two values, in which the greater number relates to the smaller, as well as their sum relates to the greater of them [12]:

$$\varphi = (1 + \sqrt{5}) / 2 = 2 \sin(3\pi/10) = 2 \cos(\pi/5). \quad (6)$$

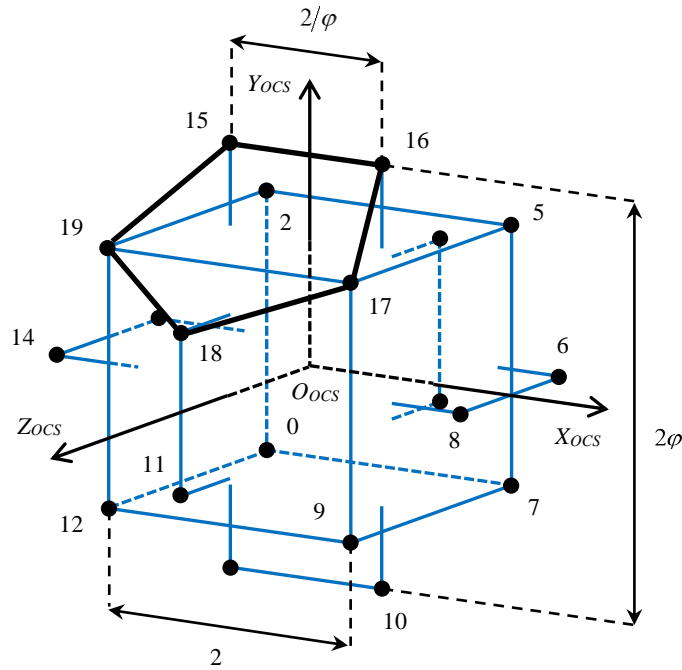


Fig. 3. The relation between dodecahedron vertices and “golden rectangles”

Define the 8 dodecahedron vertices to be vertices of inscribed cube of size 2, and remaining 12 - as “golden rectangles” vertices. Denote by P_0, \dots, P_{19} the vertices coordinates in dodecahedron’s *Object Coordinate System (OCS)* $O_{ocs}X_{ocs}Y_{ocs}Z_{ocs}$. The origin O_{ocs} of the coordinate system is located in the dodecahedron center, the axis $O_{ocs}X_{ocs}$ passes through the middle of dodecahedron edge {6, 8}, the axis $O_{ocs}Y_{ocs}$ - through the middle of the edge {15, 16}, and the axis $O_{ocs}Z_{ocs}$ - the middle of the edge {11, 18}. Based on Figure 3 we draw up the Table 1 of coordinates P_0, \dots, P_{19} .

Table 1. The coordinates of the dodecahedron vertices in *OCS* system

Coordinates	Values			
P_0, P_2, P_5, P_7	(-1 -1 -1)	(-1 1 -1)	(1 1 -1)	(1 -1 -1)
$P_9, P_{12}, P_{17}, P_{19}$	(1 -1 1)	(-1 -1 1)	(1 1 1)	(-1 1 1)
P_1, P_6, P_8, P_{14}	(- φ 0 -1/ φ)	(φ 0 -1/ φ)	(φ 0 1/ φ)	(- φ 0 1/ φ)
P_3, P_4, P_{11}, P_{18}	(0 1/ φ - φ)	(0 -1/ φ - φ)	(0 -1/ φ φ)	(0 1/ φ φ)
$P_{10}, P_{13}, P_{15}, P_{16}$	(1/ φ - φ 0)	(-1/ φ - φ 0)	(-1/ φ φ 0)	(1/ φ φ 0)

Denote by $\{v_{i,0}, v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ vertices numbers of some i th pentagon from H sequence (see the Section 2.2 beginning). The center coordinates O_i of the i th pentagon are calculated as average of its vertices coordinates taken from the Table 1:

$$O_i = \frac{1}{5} \sum_{j \in \{v_{i,0}, \dots, v_{i,4}\}} P_j. \quad (7)$$

From the expression (7) the coordinates of view vector $\mathbf{v}_{ocs,i}$ and «up» vector $\mathbf{u}_{ocs,i}$ of camera C_i in *OCS* system are easy obtained:

$$\mathbf{v}_{ocs,i} = O_i, \quad \mathbf{u}_{ocs,i} = P_{v_{i,2}} - O_i, \quad (8)$$

and desired coordinates of \mathbf{v}_i and \mathbf{u}_i vectors in *WCS* system can be calculated as

$$\mathbf{v}_i = M_v^{-1} \mathbf{v}_{ocs,i}, \quad \mathbf{u}_i = M_v^{-1} \mathbf{u}_{ocs,i}, \quad (9)$$

where M_v is the upper-left 3x3 submatrix of the matrix V transforming coordinates from *WCS* system to left-handed view coordinate system (*View Coordinate System, VCS*) [13]. In our work *VCS* system is tightly bound to *OCS* system of the dodecahedron: the origin O_{vcs} coincides the origin O_{ocs} and dodecahedron center, the axis $O_{vcs}Y_{vcs}$ coincides in direction with the axis $O_{ocs}Y_{ocs}$ and the vector \mathbf{u} , the axis $O_{vcs}Z_{vcs}$ - with the axis $-O_{ocs}Z_{ocs}$ and the vector \mathbf{v} , and the axis $O_{vcs}X_{vcs}$ - with the axis $O_{ocs}X_{ocs}$.

We set calculated \mathbf{v}_i and \mathbf{u}_i vectors for each C_i camera using OpenGL library operator *gluLookAt*.

3 Reconstructing the panorama from dodecahedron snapshots

Consider the task of playback 360-video based on the projection onto regular dodecahedron. Every frame of such video contains 12 snapshots of virtual environment, corresponding to twelve dodecahedron faces. In order to reconstruct 360-panorama, we will create separate virtual scene, place an observer in the center, and arrange snapshots (textured rectangles) around the observer, as demonstrated in Figure 4.

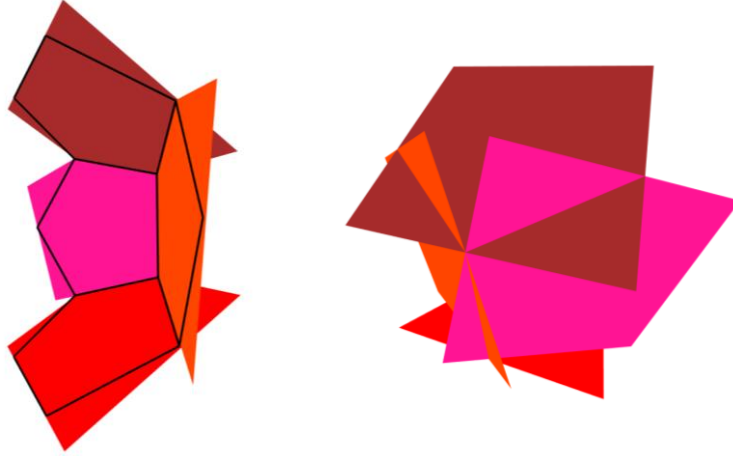


Fig. 4. The principle of docking dodecahedron snapshots

Snapshots' rectangles are tightly bound with our virtual scene WCS' system in the same way, as OCS system is bound with dodecahedron on Figure 3, i.e. axis $O_{wcs'}X_{wcs'}$ passes through the middle of the edge formed by 3 and 9 snapshots cross-section (numbers correspond to pentagons numbers), and etc.

Consider calculating the coordinates of the i th rectangle vertices in WCS' system. Denote by A'_i, B'_i, C'_i, D'_i its vertices coordinates and by $M'_i, U'_i, S'_i, T'_i, Q'_i, O'_i$ - vertices and center coordinates of the pentagon placed in $A'_iB'_iC'_iD'_i$ (see Fig. 5). As seen from the Figure 5, we don't use the addition h (introduced in Section 2.1) at visualization, since h is required only to obtain correct perspective projection while taking a snapshot. Note, that vertices and center coordinates of the pentagon are also calculated according to Table 1 and equation (7).

Let's introduce unit vectors $e_1 = M'_iQ'_i/|M'_iQ'_i|$ and $e_2 = O'_iS'_i/|O'_iS'_i|$. Then for desired vertices coordinates of $A'_iB'_iC'_iD'_i$ the following expressions can be written:

$$B'_i = S'_i - \frac{d'}{2}e_1, C'_i = S'_i + \frac{d'}{2}e_1, A'_i = B'_i - (R'_{penta} + r'_{penta})e_2, D'_i = C'_i - (R'_{penta} + r'_{penta})e_2, \quad (10)$$

where d' , R'_{penta} and r'_{penta} are the diagonal and the radiuses of circumscribed and inscribed circle of the pentagon, calculated similarly to d , R_{penta} and r_{penta} from (1).

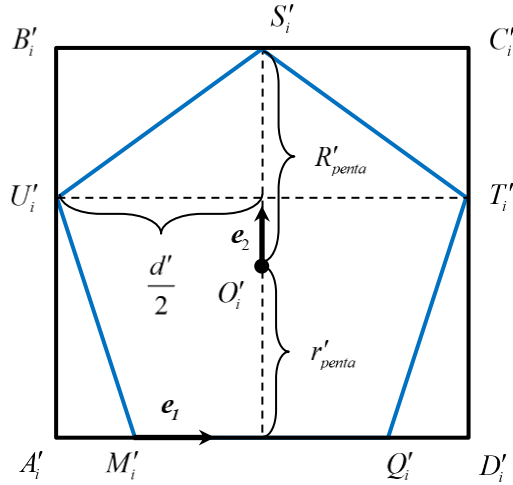


Fig. 5. Calculating the coordinates of snapshot vertices

To reconstruct the 360-panorama, we calculate vertices coordinates for all the twelve rectangles using formulas from (10) and texture them with snapshots. Figure 6 shows linking dodecahedron faces to the textures-snapshots of the virtual environment, written in a frame of 360-video (see Fig. 7).

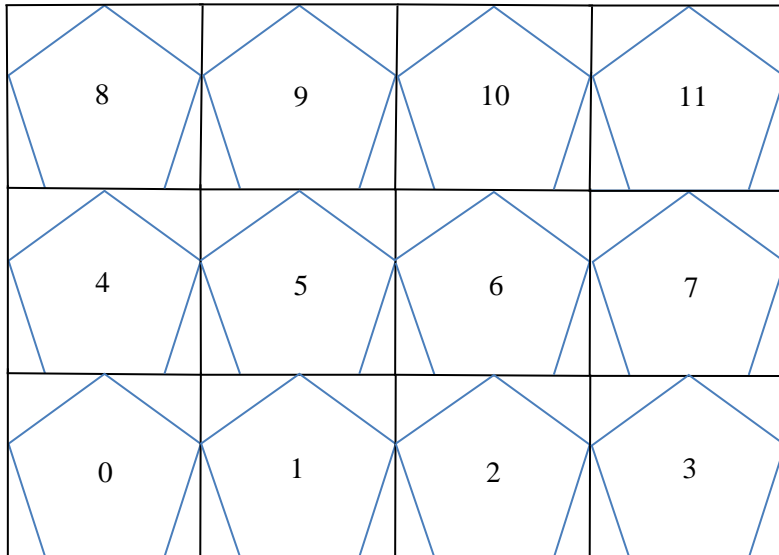


Fig. 6. The arrangement of dodecahedron face textures in a frame of 360-video

4 Results

Developed technology was implemented in software complex for 360-video synthesis based on virtual environment projection onto regular dodecahedron. The complex is written in C++ using graphics library OpenGL and a set of FFmpeg libraries for coding/decoding digital videos. The proposed solution includes a module for capturing 360-frames, to be embedded in visualization system, and a player for created 360-video.

The solution was tested in visualization system of virtual experiment for Earth observation from the International space station (ISS) [14]. Using the capture module, a video of the flight along the section of the ISS 3rd day orbit track was created. The flight passed through Balkhash lake (from 73 to 78 degrees eastern longitude). Video capturing was carried out at the frequency of 25 frames/s based on video compression standard H.264 and MP4 container. Dodecahedron snapshots had sizes of 896x848 pixels, and 360-video frames – size of 3584x2544 pixels (are multiples of 16 according to H.264). A frame example from the synthesized 360-video is demonstrated in Figure 7. During a 360-video playback viewer can “turn their head” up/down and right/left and zoom in/out the image. The Figure 8 illustrates the visualization of virtual Earth panorama reconstructed at several viewing directions from 360-frame shown in Figure 7.

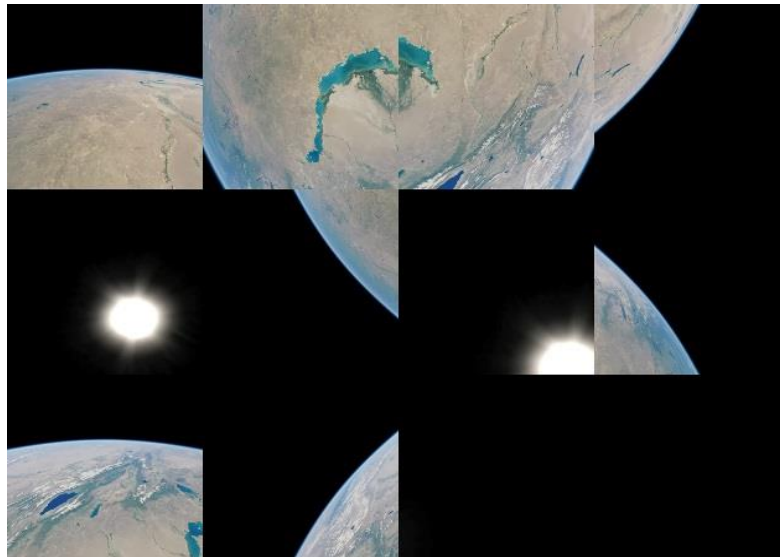


Fig. 7. A frame of synthesized 360-video based on regular dodecahedron

We also compared the quality of the projection of the virtual environment onto a regular dodecahedron and onto a cube. We evaluated changing the number of pixels that could be occupied by the projection of a test segment onto face texture for the dodecahedron and the cube. The segment occupied the largest number p_{max} of pixels in the region of face vertex, and the smallest p_{min} - in face center. In the case of dodecahedron, the p_{min}/p_{max} ratio was about 0.6, while for a cube it was only about 0.3 (ideally,

for a sphere, this ratio is 1). Thus, the use of a projection onto a dodecahedron can reduce the loss of detaility by up to 2 times compared to cubemap projection. Meanwhile, video stream bitrate depends on, how efficiently dodecahedron faces are arranged in the frame. While experimenting with software complex a packing scheme for dodecahedron faces was developed. This packing allows the number of unused frame parts to be extremely reduced (almost twice) without any quality loss. Implementation of such packing algorithm is planned in extended research of developed technology.

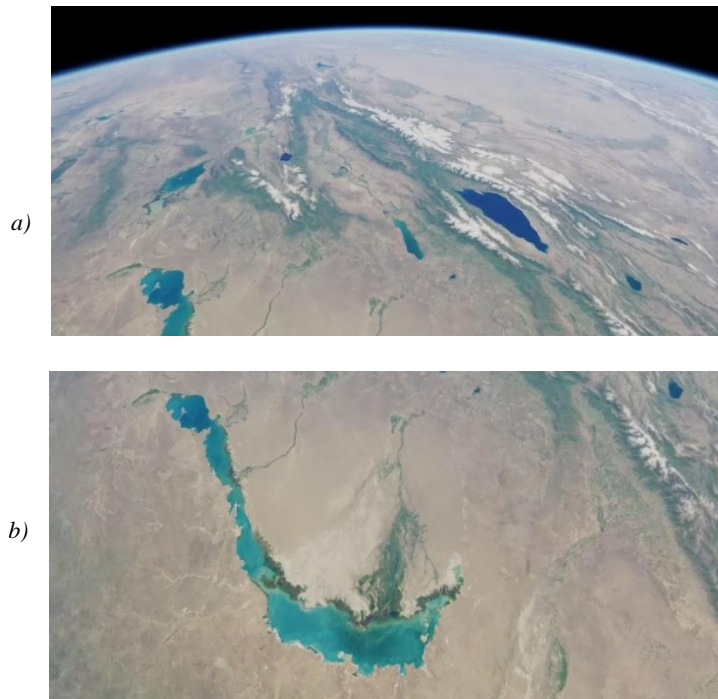


Fig. 8. Virtual Earth panorama at viewing directions: a) forward; b) nadir

5 Conclusion

This work relates to the direction of research and development of new technologies and methods for the synthesis of 360-video in virtual environment systems, based on the projection of the environment on the faces of convex polyhedra. Despite the great popularity of the cubemap projection, it has several disadvantages that affect the quality and redundancy of the video stream bitrate. In this paper, we propose a new technology based on the projection of virtual environment on the faces of a regular dodecahedron. The technology includes projection construction stage resulting in 12 rectangular snapshots of the virtual environment, and the stage of 360-panorama reconstruction from these images. To implement the first stage, methods to calculate the projection and the

orientation parameters of virtual cameras are proposed, by which 12 snapshots are synthesized. To implement the second stage, an original scheme of snapshots arrangement (docking) around the observer is proposed, which provides synthesis of continuous 360-panorama, as well as a method to calculate the positions of such snapshots. The developed technology and methods were implemented in the software complex and tested in the visualization system of virtual experiments to observe the Earth from the ISS. Testing confirmed the adequacy of the results obtained to the task, and their applicability in the areas of virtual environment systems, video simulators, virtual laboratories, etc.

References

1. Zhdanov, D.D., Galaktionov, V.A., Voloboy A.G., Zhdanov A.D., Garbul' A.A., Potemin I.S., Sokolov V.G.: Photorealistic Rendering of Images Formed by Augmented Reality Optical Systems. *Programming and Computer Software* 44(4), 213–224 (2018).
2. Barladian, B.K., Shapiro, L.Z., Mallachiev, K.A., Khoroshilov, A.V., Solodelov, Yu.A., Voloboy, A.G., Galaktionov, V.A., Koverminskii, I.V.: Visualization Component for the Aircraft Real-Time Operating System JetOS. *Programming and Computer Software* 46(3), 167–175 (2020).
3. Mikhaylyuk, M.V., Torgashev, M.A.: The System of Visualization “GLView” for Simulators and Virtual Environment Systems. In: 25th International Conference on Computer Graphics and Vision (GraphiCon 2015), pp. 96–101. Protvino, Russia (2015).
4. Andreev, S.V., Bondarev, A.E., Galaktionov, V.A., Bondareva, N.A.: The problems of stereo animations construction on modern stereo devices. *Scientific Visualization* 10(4), 40–52 (2018).
5. Timokhin, P.Yu., Mikhaylyuk, M.V., Vozhegov, E.M., Panteley, K.D.: Technology and methods for deferred synthesis of 4K stereo clips for complex dynamic virtual scenes. *Trudy ISP RAN/Proc. ISP RAS* 31(4), 61–72 (2019).
6. Corbillon, X., Simon, G., Devlic, A., Chakareski, J.: Viewport-Adaptive Navigable 360-Degree Video Delivery. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–7. IEEE, Paris (2017).
7. Jiang, H., He, G., Yu, W., Wang, Z., Li, Y.: Scalable Video Coding Based on the User’s View for Real-Time Virtual Reality Applications. In: Zeng, B. et al. (eds.) *Advances in Multimedia Information Processing – PCM 2017*, LNCS 10736, pp. 766–775. Springer (2018).
8. Yu, M., Lakshman, H., Girod, B.: A Framework to Evaluate Omnidirectional Video Coding Schemes. In: 2015 IEEE International Symposium on Mixed and Augmented Reality, pp. 31–36. IEEE, Fukuoka (2015).
9. Fu, C.-W., Wan, L., Wong, T.-T., Leung, C.-S.: The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video. *IEEE Transactions on Multimedia* 11(4), 634–644 (2009).
10. Zwillinger, D.: *CRC Standard Mathematical Tables and Formulas*. 33rd edition. CRC Press, Boca Raton (2018).
11. Vasilieva, V.N.: Golden Section and Golden Rectangles When Building Icosahedron, Dodecahedron and Archimedean Solids Based On Them. *Geometry & Graphics* 7(2), 47–55 (2019).
12. Akhtaruzzaman M., Shafie A.A.: Geometrical Substantiation of Phi, the Golden Ratio and the Baroque of Nature, Architecture, Design and Engineering. *International Journal of Arts* 1(1), 1–22 (2011).
13. Hill F.S. *Computer Graphics Using OpenGL*. Second Edition. Prentice Hall (2001).
14. Mikhaylyuk, M.V., Timokhin, P.Y., Maltsev, A.V.: A method of Earth terrain tessellation on the GPU for space simulators. *Programming and Computer Software* 43(4), 243–249 (2017).