

Web-oriented visualization of auroral oval geophysical parameters

A.V. Vorobev^{1,A,B}, V.A. Pilipenko^{2,B,C}, A.G. Reshetnikov^{3,A}, G.R. Vorobeva^{4,A}, M.D. Belov^{5,A}

^A Ufa State Aviation Technical University

^B Geophysical Center of the RAS

^C Institute of Physics of the Earth of the RAS

¹ ORCID: 0000-0002-9680-5609, geomagnet@list.ru

² ORCID: 0000-0003-3056-7465, pilipenko_va@mail.ru

³ ORCID: 0000-0002-9022-3085, btr180@gmail.com

⁴ ORCID: 0000-0001-7878-9724, gulnara.vorobeva@gmail.com

⁵ ORCID: 0000-0001-9861-7604, mishkabelka251@gmail.com

Abstract

Analysis of well-known services for monitoring geophysical parameters in the auroral oval area has revealed a number of disadvantages, which are typical and, in principle, repeating from implementation to implementation. They are the inability to dynamically scale and add auxiliary layers; the small number of displayed parameters; the lack of data on the current state of space weather and basic tools for spatial analysis of visualized parameters, etc. Thus, the paper is concerned with an approach (architecture and its software implementation) to create a web-based system utilizing virtual globe technology for visualizing geophysical parameters in auroral zone (www.aurora-forecast.ru). The visualized data are the intensity (probability of observing) of auroras obtained in accordance with the OVATION model and the potentials of electric and magnetic fields calculated according to the Weimer model. It is assumed that the considered approach to the visualization of geophysical parameters in the auroral oval area will increase the efficiency of work related to the study of the auroral oval dynamics and ultimately help to identify new knowledge about the topology of magnetosphere and its changes, for example, during geomagnetic storms and substorms.

Keywords: aurora, auroral oval, geomagnetic variations, space weather.

1. Introduction

It is known [1], that the auroral ovals are high-latitude regions of the planet northern and southern hemispheres. They have the shape of an oval belt, in the center of which there are the corresponding magnetic poles of the Earth. The geographic boundaries of these belts are determined by the possibility of observing the auroras within their limits. Here the auroras are the glow of the upper layers of the atmosphere, arising under the influence of the fluxes of solar electrons and protons penetrating into it.

So, in calm periods, the diameter of the auroral ovals is about 3000 km, while on the day and night side the border of the auroral zone is 10-16° and 20-23° from the magnetic pole, respectively. Since the Earth's magnetic poles are ~ 11.5° away from the geographic poles, the polar auroras are most often observed in the range of northern and southern latitudes of 67-70°. However, during the solar activity, the auroral oval expands and polar auroras can be observed 20-25° south or north (for the northern or southern hemisphere, respectively) from the boundaries of their usual manifestation.

Until recently, it was believed that the auroras in the northern and southern hemispheres are strictly symmetrical. However, the simultaneous observation of aurora borealis from the outer

space from the north and south poles in October 2002 demonstrated, that the northern and southern auroras are significantly different from each other [2].

Currently, the scientists developed several mathematical models to predict the value of various geophysical parameters in the region of the auroral oval (electric and magnetic potential, integrated excitation power of the upper atmosphere, etc.). These models usually use such input data as the solar wind and interplanetary magnetic field parameters recorded in real time by the ACE satellite, and since 2016 also by the DSCOVR satellite. (Both satellites are at Lagrange point L1 on the Earth-Sun line).

For example, the empirical model OVATION (Oval Variation, Assessment, Tracking, Intensity, and Online Nowcasting), developed by Patrick Newell and his colleagues from the Laboratory of Applied Physics at Johns Hopkins University, USA in 2009 [3], assumes a linear direct proportional relationship between the intensity of the auroras and the probability of their observation with the naked eye. In the future, according to the UVI (Ultraviolet imager) instrument complex installed on the NASA Polar satellite, this position was experimentally confirmed.

The NOAA web service (<https://www.swpc.noaa.gov/products/aurora-30-minute-forecast>), which is based on the OVATION-prime model, is used by the National Oceanic and Atmospheric Administration of the USA [4] for short-term prediction of the auroras intensity and provides visualization of the probability of atmospheric glow in the region of the auroral oval (Fig. 1). Today the service is probably one of the most famous software products of this kind. In addition to meteorological services and specialized research organizations, the visualization results (referring directly to the NOAA resource, or as separate applications copying image data) are used by travel agencies around the world, attracting millions of tourists to high-latitude regions of the world to observe the auroras in the most favorable for this periods.

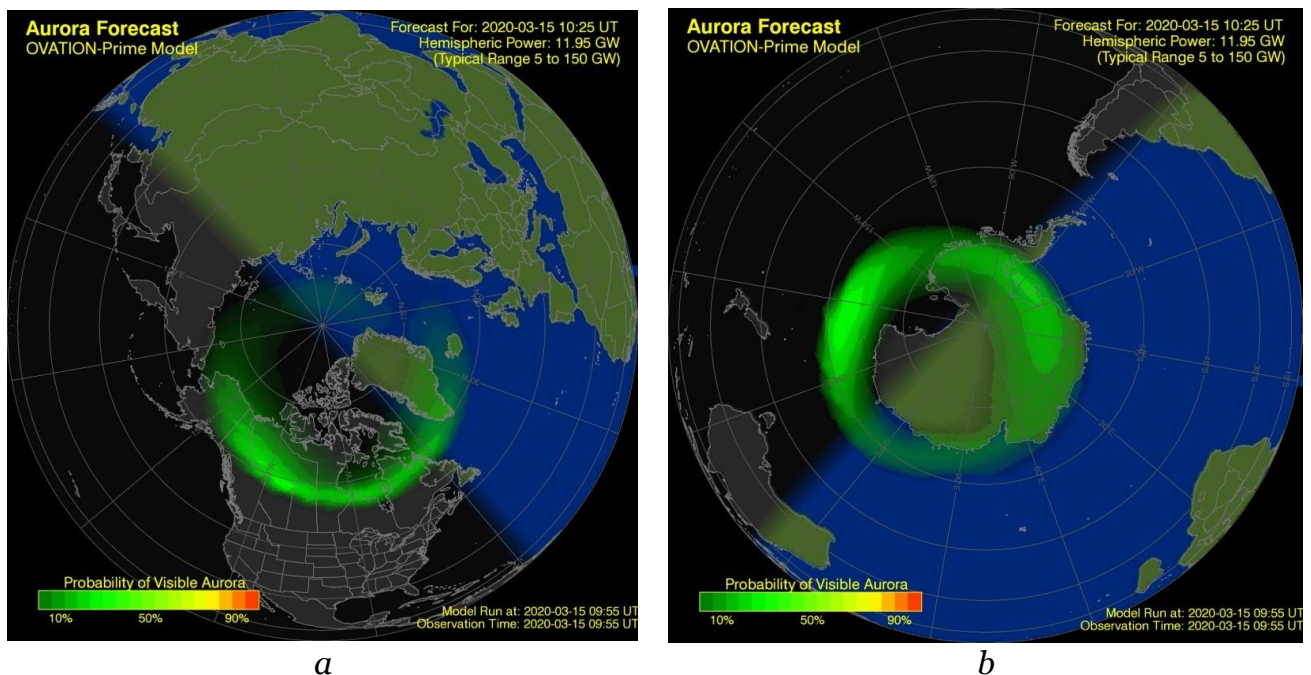


Fig. 1. An example of visualization of a short-term forecast of the probability of visibility of auroras by the NOAA service for the northern (a) and southern (b) hemispheres of the Earth

In addition to the NOAA product, there are less-known web services focused on regional monitoring of fragments of auroral oval. So, for example, Fig. 2a and Fig. 2b respectively show the screen shots of the services developed by the University of Alaska at Fairbanks, USA (<https://www.gi.alaska.edu/monitors/aurora-forecast>) and the Iceland Weather Service (<https://en.vedur.is/weather/forecasts/aurora/>).

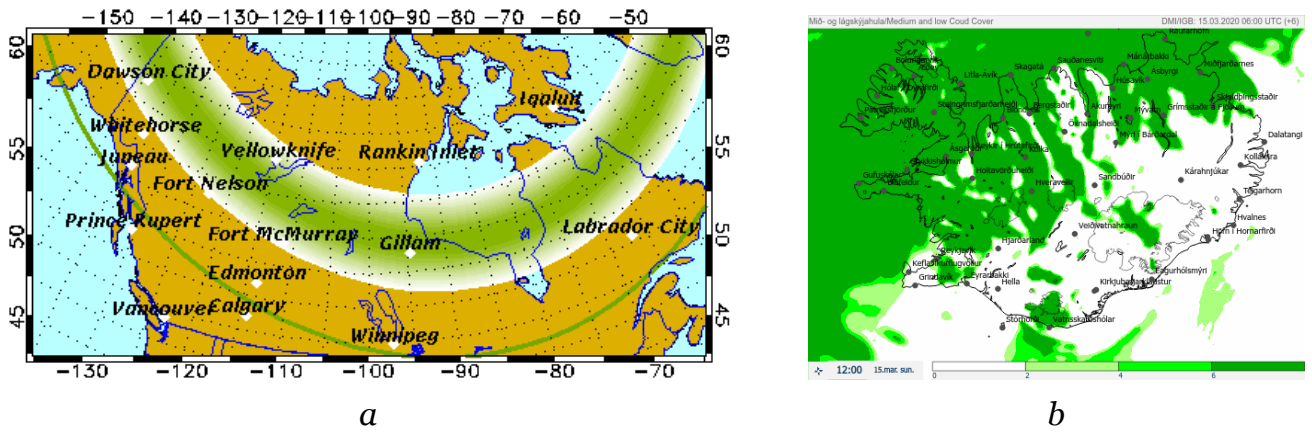


Fig. 2. An example of visualization of the forecast of auroras for the regions of North America (a) and Iceland (b)

The experience with using the mentioned and some other similar services allowed to reveal a number of characteristic disadvantages, which repeat from implementation to implementation. They are the inability to dynamically scale and add additional layers; a small number of displayed parameters; lack of data on the current state of space weather and basic tools for spatial analysis of visualized parameters.

So, the development and modernization of web services that provide effective monitoring and visualization of geophysical parameters in the auroral oval area, as well as allowing for their concomitant analysis by GIS methods, is still an urgent scientific and technical task, the solution of which will obviously help to better understand physics of various kinds of processes both in auroral and adjacent areas.

2. Synthesis and preparation of source data

Here the following parameters in the auroral oval region are considered as visualization objects: the probability of observing the glow of the upper atmosphere by the naked eye, the electric and magnetic field potentials in the region of the northern auroral belt, and the geometric centers of the northern and southern auroral belts determined by the coordinates of the southern and northern magnetic poles in accordance with the expressions (1). So, for example, the coordinates of the north magnetic pole, and hence the center of the southern auroral ovals on January 1, 2020 will be 80.59° S, 72.68° W:

$$\tan(X_0) = \frac{g_1^0}{\sqrt{(g_1^1)^2 + (h_1^1)^2}}; \tan(Y_0) = \frac{h_1^1}{g_1^1}, \quad (1)$$

where X_0 and Y_0 are the latitude and longitude of the geomagnetic dipole north pole, respectively; g_1^1 and h_1^1 are the spherical harmonic coefficients (IGRF coefficients) for a given magnetic era (<https://www.ngdc.noaa.gov/IAGA/vmod/coeffs/igrf13coeffs.txt>).

The probability of observing the glow of the upper atmosphere with the naked eye, i.e. of auroras, is determined (like in the NOAA service) in accordance with the OVATION-prime forecasting model, which implements a short-term (30 min) forecast of auroras. The delay value of 30 min corresponds to a solar wind speed of ~ 800 km / s, however, in fact, the delay time varies from less than 30 minutes to an hour or more, depending on the average solar wind speed.

The data are represented in ASCII format as an array of 1024×512 values, which corresponds to the longitude values from 0 to 360° in increments of 0.32846715° and the latitude values from -90 to 90° in increments of 0.3515625° , respectively (<https://services.swpc.noaa.gov/text/aurora-nowcast-map.txt>). The array values are ranged from 0 to 100 , where 0 corresponds to the minimum probability of observing auroras, and 100 to the maximum. Data is updated every 5 minutes.

To determine the boundaries of the auroral oval, as well as to approximate geomagnetic disturbances caused by ionospheric Hall currents, to calculate the flux of the Poynting or Joule

heating vectors in the ionosphere, it is necessary to calculate the parameters of the electric and magnetic potential [5-6]. In this work, the electric and magnetic potential in the auroral zone is calculated according to the empirical Weimer model [7]. The model was implemented as a separate server script in Python 3, with the prospect of its use for web-based visualization under the Django framework. As input parameters in the model, as well as in the previous case, the solar wind and interplanetary magnetic field parameters recorded by the DSCOVR satellite are used. Also according to the model, when determining potential values, spherical harmonic functions are used only in a narrow region of high latitudes. At lower latitudes, potentials are calculated by several functions of the longitude of the Fourier series with a discrete step in latitude. This data is presented as separate files physically separated from the main script.

3. Comparative analysis and choice of visualization tools

The geophysical parameters of the auroral zone as an object of visualization are a structured set of spatial and attribute data, processing and graphical interpretation of which, obviously, is advisable to be implemented through web-based GIS technology. For example, the experience of building web GIS for visualization and analytical control of the parameters of geomagnetic field and its variations [8–9], showed the viability of these software tools for solving such problems. According to the approach of representing geospatial data, existing GIS methods in general can be divided into classic flat maps and virtual globes [10]. In this case, given the specifics of the high-latitude location of the visualized objects (i.e., auroral ovals, auroral zones and their parameters), the obvious advantage of virtual globes is the quality of their visual perception, preservation of the geometric similarity of the contours, the ratio of the surface of the Earth and the absence of cartographic distortions of projections, peculiar to flat maps, especially in the polar regions.

Table 1 represents results of comparing of the characteristics and some of the possibilities of using modern software libraries that allow to embed a virtual globe into a web application.

A significant criterion for choosing a tool that provides visualization according to the problem to be solved is the speed of image rendering, which practically determines the level of detail and realism of both the virtual globe itself and the loaded layers that reflect the spatial distribution of the analyzed parameters. Thus, taking into account the geographical boundaries of the auroral zone, it makes sense to use small-scale (from 1: 2,000,000 to 1: 10,000,000) cartographic substrates. At this scale, spatial visualization of the analyzed objects can be realized by means of a ranked color indication of the parameters, which can be applied both to point-type objects and to polygonal associations.

Comparative analysis and experience with the APIs (Table 1) showed that to solve the problem (without presenting requirements for the interface, tools, input data, visualization quality, compatibility and the possibility of further development), any of the libraries mentioned can be used. However, given the advanced features of 2D / 3D visualization of the Earth's surface, the use of various formats for rendering layers by data arrays and, importantly, support for the Python language, in this work, we decided to use the ArcGIS API.

Table 1 - Programming technologies for virtual globes

Name URL	Programming Languages Support			Visualization Mode	Open Source	Web Access	Hardware acceleration
	JavaScript	Python	Java				
ArcGIS API https://developers.arcgis.com/	+	+	+	2D/3D	-	+	+
Cesium https://cesium.com/	+	-	-	2D/2.5D/3D	+	+	+
NASA World Wind https://worldwind.arc.nasa.gov/	+	-	+	3D	+	-	+
Miniature Earth	+	-	-	3D	-	+	+

https://miniature.earth/							
WebGLEarth https://www.webgl-earth.com	+	-	-	3D	+	+	+
Gio.js https://giojs.org/	+	-	-	3D	+	+	+
WhirlyGlobe https://mousebird.github.io/WhirlyGlobe/	+	-	-	2D/3D	-	+	+
GLOBE Elasticsearch API https://www.globe.gov/ru/globe-data/globe-api	+	-	-	3D	-	+	+
OpenGlobus API https://www.open-globus.org/	+	-	-	3D	+	+	+

4. Architecture and programming implementation of the visualization system

The proposed visualization system is based on a client-server architecture typical for web applications, implemented using the MVC (Model-View-Controller) design pattern and the corresponding component separation of application data, user interface, and control logic. The advantages of the approach include the structuredness of the program code, the possibility of its reuse and, as a result, reducing the complexity of the web application. Currently, the architectural pattern is supported by many web development tools, however, the implementation of the proposed solutions is based on the Django framework, which is a framework with many built-in high-level capabilities and a standardized structure of the developed applications.

Also the framework includes mechanisms to prevent common attacks such as XSS (Cross-Site Scripting) and CSRF (Cross Site Request Forgery), uses ORM (Object -Relational Mapping) and can withstand highly loaded applications due to caching and load balancing. Django implements the principles of RAD (Rapid Application Development) of the software development technological process organization. But most importantly, Django uses Python as a programming language, which expands its functionality for complex analytics of big data, its processing and visualization. All of the above distinguishes Django from other approaches to web development and determines its choice as a tool for implementing a system of visualization of geophysical parameters in the auroral oval area.

It is important to note that the concept of MVC is used in Django in a slightly modified form: the framework implements an architectural pattern designated as MVT (Model-View-Template) and is a modification of MVC.

Client-server interaction within the framework of the indicated pattern is implemented as follows (Fig. 3). The control logic of the application is defined at the level of views, which, in turn, are placed in the Views.py file and are set by a number of functions with an input argument of type HttpRequest and a return value of the form HttpResponse. Each view is attached to the project by a corresponding link. Client-server interaction within the framework of the indicated pattern is carried out as follows. Each view is attached to the project by a corresponding link in the URL.py file (a component known as the URL manager) (Fig. 3). On receipt of a user request, the controller through the URL.py file determines which server resource should be used to generate the response, and redirects the request parameters to the corresponding representation.

In Django, the Views.py file performs the role of a controller that processes an incoming request and generates a response, the form of which for the user is determined by the Template. The template contains static HTML and dynamic data, the substitution of which into the result is described using the appropriate program instructions. Template calls are made directly from the function in the view using the render method of the django.shortcuts package, which exe-

cutes the specified template and returns an instance of the HttpResponse object with the received content.

All the mentioned components (the controller, views, templates, URL dispatcher) make up the Django application. The interaction of the Django application directly with the web server is carried out according to the WSGI (Web Server Gateway Interface) standard. The web server executes the code based on the received HTTP request, retrieves its parameters and passes them along with the callback function (if available) to the web application. The web application processes the request as described above and returns the result back to the web server.

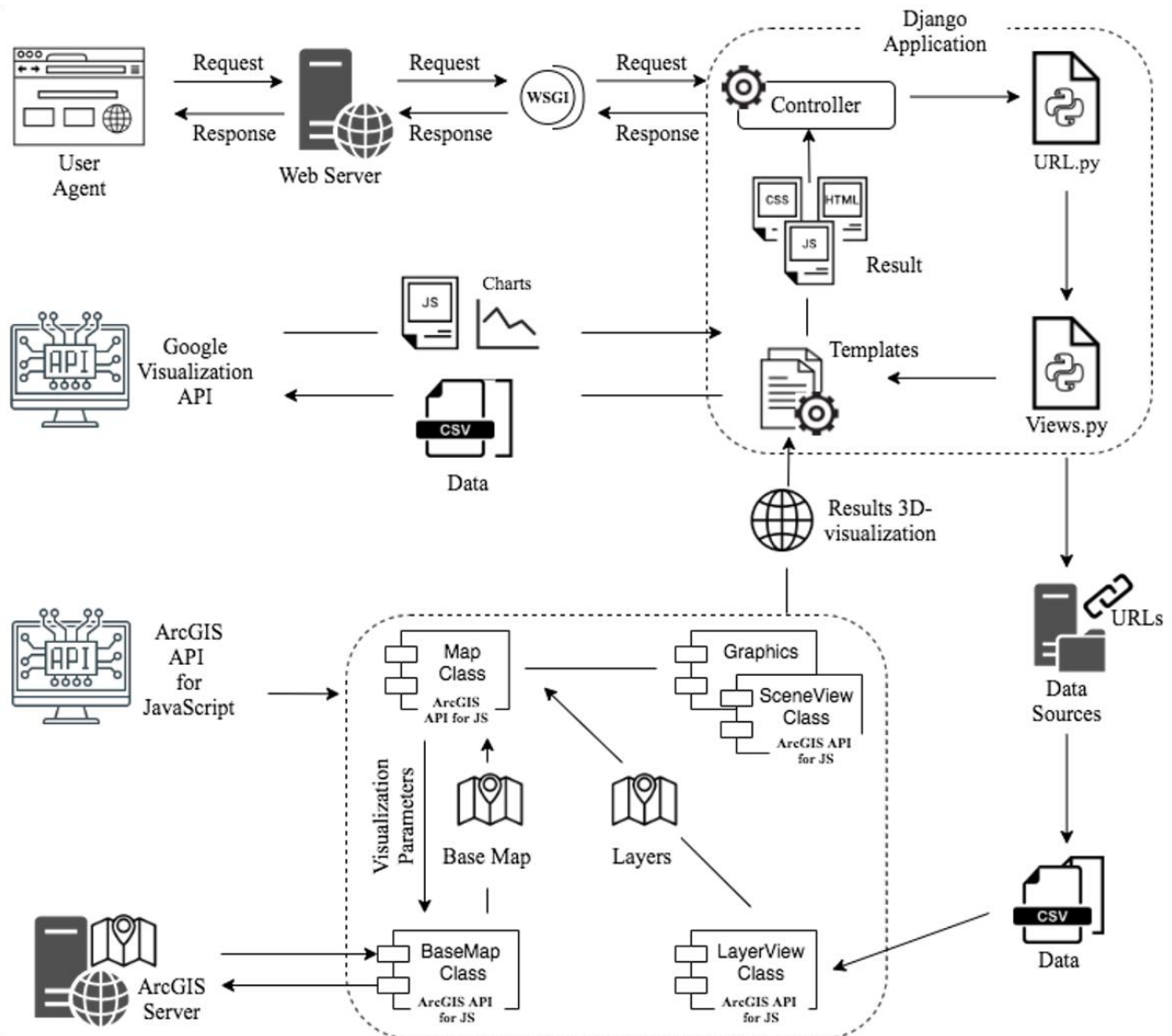


Fig. 3. Architecture of system for geospatial visualization of geophysical parameters in the auroral oval area

In processing and visualization of spatially distributed values of various geophysical parameters of the auroral zone, CSV and JSON data from third-party sources are used. At the same time, in order to avoid conflicts associated with the fact that most browsers follow the same-origin policy, access to remote data sources is carried out only on the server side. The corresponding scripts are executed at the beginning of the user session with a Django application and involves sending a request to external data sources with the transmission of the parameters received in the user message. So server python script sequentially accesses NOAA data in the area of the auroral oval (url: <https://services.swpc.noaa.gov/text/aurora-nowcast-map.txt>, url: <https://services.swpc.noaa.gov/text/aurora-nowcast-map.txt>), interplanetary magnetic field (url: https://services.swpc.noaa.gov/products/solar-wind/mag_-6-hour.json)

and solar wind(url: <https://services.swpc.noaa.gov/products/solar-wind/plasma-6-hour.json>). The results of the script are transferred to the appropriate template from the Django application and sent by the web server to the client side as a response for subsequent rendering by the user agent (browser). Thus, work with sources of visualized data is carried out on the principle of federalization without data alienation, followed by local storage of file sets on a web server.

In the generation of the resulting HTML code by the web application templates, the functionality of two external APIs was used. The first one is the Google Visualization API (<http://www.google.com/jsapi>, the visualization module), which serves as the basis for the formation and rendering on the client side of graphs characterizing the change in the parameters of the interplanetary magnetic field and the solar wind according to data from third-party sources. At the initial stage, a separate plug-in module of the Google loader (loader.js) creates a gateway for connecting to graphic components for generating client-side charting code. Then, directly in the template, the load method of the google.chart class instance connects to the application the necessary external modules (in this case, for example, 'corechart', 'controls'), to the input of which the data array for visualization is supplied. Finally, after setting up the graphic components and specifying the HTML element to bind the generated graphic content to it, the draw method draws the corresponding graphics on the application page using the callback function.

Another third-party API used in the application is responsible for visualizing spatial data characterizing the distribution of geophysical parameters in the auroral zone with reference to geographical coordinates. The ArcGIS API for JavaScript is used for this purpose (<https://developers.arcgis.com/javascript/>). Work with spatial data begins with the initialization of an instance of the Map class, designed to visualize a variety of map layers. It is used as a base one and is formed by the BaseMap class, which accesses a remote ArcGIS server, receives from it a cartographic substrate of the type specified in the request and passes it to a previously created instance of the Map class (within the framework of the indicated web application, the ArcGIS map of the type acts as a cartographic layer "Streets-night-vector").

The visualization special feature of the data used in the work is a large number of spatial points (524288 points), each of which is a separate instance of the Point class and, as a result, a separate DOM object for the browser. The DOM object, for security purposes, provides for the limitation of rendered in the user window the number of DOM objects, which manifests itself at an extremely low speed of rendering points at a level close to half a thousand and the impossibility of rendering (with the number of points exceeding ~ 500 objects). In this regard, the rendering of spatial points is carried out by grouping them into a cartographic layer with setting up a color map for a range of visualized component points.

To solve the problem, the customizable cartographic layers that provide visualization of the auroral zone parameter values spatial distribution are formed as instances of the LayerViewClass class. As an input parameter, the constructor of the designated class accepts a CSV file containing the data to be displayed on the map. At the output, a cartographic layer is formed and tied to a previously created copy of the cartographic substrate.

At the final stage, the cartographic base with attached layers is converted into an instance of the virtual globe by initializing an instance of the SceneViewClass class, which also accepts layers and binding to the map instance as input. The result of applying these layers within the framework of a Django application template is a stream of data sent to the client side as a response and containing HTML code, stylesheets and scripts for execution and rendering in the browser.

5. The discussion of the results

The interface of the developed system for web-based visualization of geophysical parameters in the area of the auroral oval is shown in Fig. 4. Here, in general terms, the space is logically divided into 8 functional areas: A - current date and time UT (Universal Time); B is an explanation of the color indication; C - current parameters of the solar wind and interplanetary mag-

netic field; D - virtual globe; E - visualization mode menu (midnight, northern or southern hemisphere, territory of the Russian Federation or free view); F - panel displaying the current cursor coordinates and scale; G - panel displaying the current cursor coordinates and scale; H - basic tools for spatial analysis; I - graphs of changes in the solar wind and interplanetary magnetic field over the past 6 hours. The green and blue lines correspond to the zero meridian and the meridian passing through a point with a local time of 00:00 (midnight meridian).

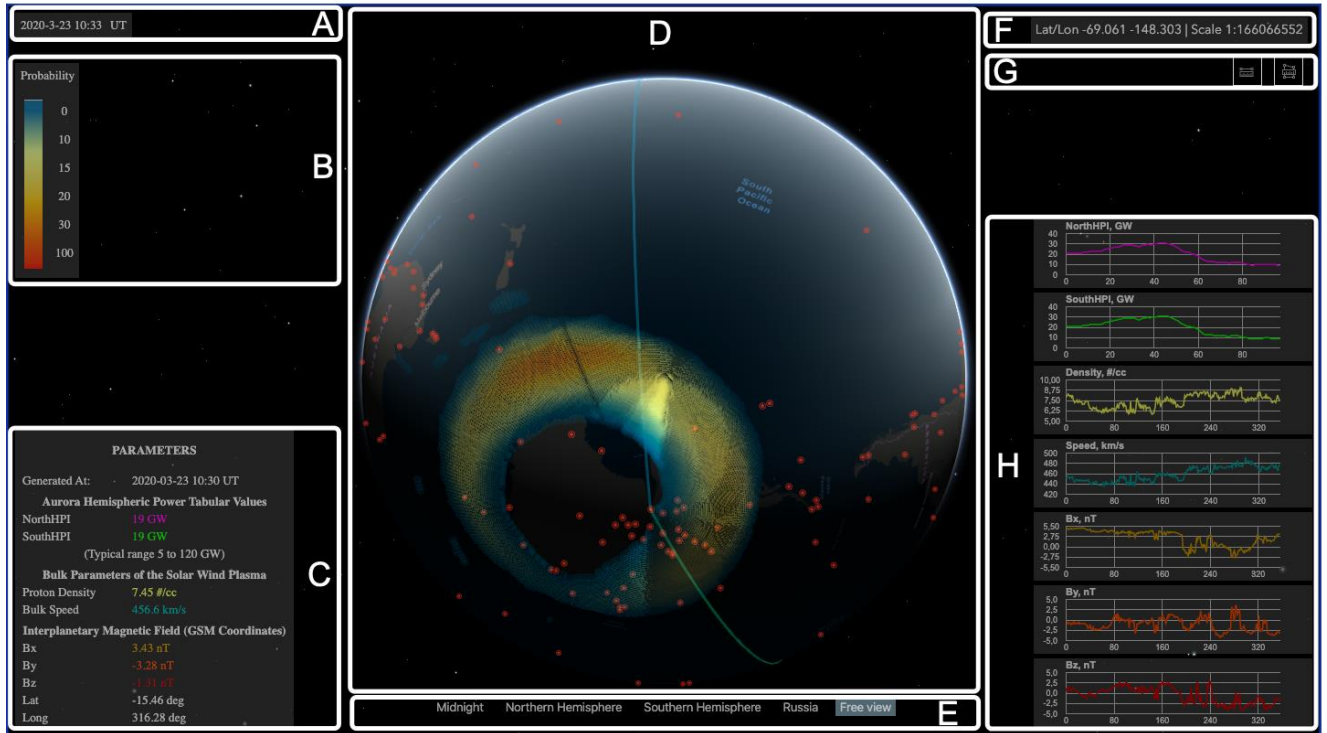


Fig. 4. Web application interface for visualization of geophysical parameters in the area of auroral oval

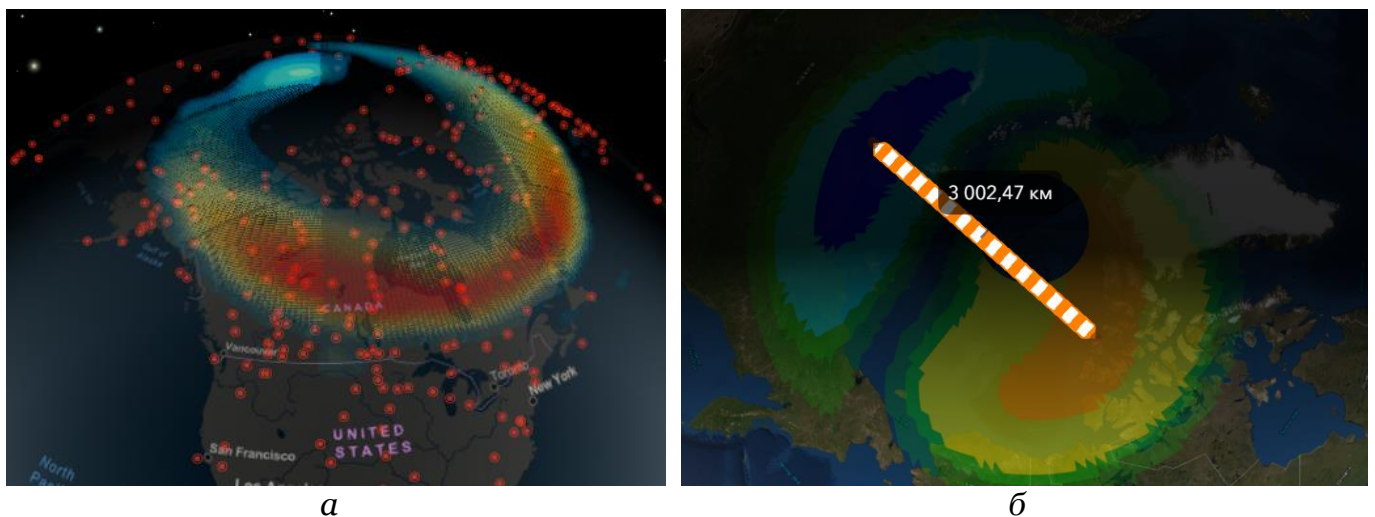


Fig. 5. Screenshots of the developed web service for visualizing geophysical parameters in the auroral oval area (<http://aurora-forecast.ru/>):
a - the intensity of the glow of the auroral oval;
b - electric field potential

For a better visualization there is a terminator, which is used to show the dividing line of the day and night. However, we note that on the day side, even despite the rather high intensity during visualization, the auroras are most likely not to be visible. However, one hour before

sunrise or one hour after sunset, with the proper intensity of the aurora, one can already observe with them the naked eye.

6. Conclusion

The following results were obtained:

– Despite the current lack of interactive web services that provide visualization of geophysical parameters in the auroral zone, it was shown that the construction of such information systems, supported with basic GIS methods, is not only possible, but also provides significant advantages over the well-known today approaches.

– Due to the geospatial specifics of the distribution of the analyzed data (narrow high-latitude belts) because of distortion of the geometric similarity of the contours, the ratio of the Earth surface area and cartographic distortions of the projections, the traditional flat cartographic substrates are impractical to use, and when implementing the system, it is worth choosing virtual globes. The exception is flat projections of the northern and southern hemispheres, however, in addition to the fact that this type of map is quite exotic and is not supported by most of the available software libraries, it is also inferior to the three-dimensional Earth model in the quality of visual perception of information.

– Among a number of libraries known today that allow (using virtual globe technology) to implement visualization of geophysical parameters in the auroral oval, according to the authors, the choice should be made in favor of the ArcGIS API, although NASA World Wind and Cesium also have significant potential for solving such problems.

– Based on the architecture developed and described earlier, a prototype system has been created, which can be found at: ([url: http://aurora-forecast.ru/](http://aurora-forecast.ru/)). At the same time, work on increasing the functionality and efficiency of visualization of the application continues.

Thus, as already noted, the development and modernization of approaches to develop web-oriented services that provide effective monitoring and visualization of geophysical parameters in the auroral oval area, as well as allowing for their concomitant analysis by GIS methods, is quite complicated today, not having an unambiguous and / or final solution to the problem. In this case, it is assumed that the development and evolution of such information system will increase the efficiency of work related to the study of the dynamics of the auroral oval [11], and ultimately it will help to reveal new knowledge regarding the topology of the magnetosphere and its changes, for example, during geomagnetic storms and substorms.

Acknowledgement

The research is supported by RSF No.17-77-20034 and RFBR No. 20-07-00011-a.

References

1. Mizun Yu.G. Auroras. Moscow: Nauka, 1983. 136 p [in Russian].
2. https://www.nasa.gov/vision/earth/lookingatearth/dueling_auroras.html
3. Newell P.T., Liou K., Zhang Y., et al. OVATION Prime- 2013: Extension of auroral precipitation model to higher disturbance levels // Space Weather. 2014. V. 12, N 6. P. 368– 379. doi: 10.1002/2014sw001056.
4. <https://www.swpc.noaa.gov/products/aurora-30-minute-forecast>
5. Weimer, D. R. (2005), Improved ionospheric electrodynamic models and application to calculating Joule heating rates, J. Geophys. Res., 110, A05306, doi:10.1029/2004JA010884
6. Weimer, D. R., C. R. Clauer, M. J. Engebretson, T. L. Hansen, H. Gleisner, I. Mann, and K. Yumoto (2010), Statistical maps of geomagnetic perturbations as a function of the interplanetary magnetic field, J. Geophys. Res., 115, A10320, doi:10.1029/2010JA015540.
7. Weimer, D. R. (2013), An empirical model of ground-level geomagnetic perturbations, Space Weather, 11, 107–120, doi:10.1002/swe.20030

8. Vorobev A.V., Vorobeva G.R. Geoinformation system for amplitude-frequency analysis of geomagnetic variations and space weather observation data //Computer Optics. 2017. No. 41(6). P. 963-972.
9. Vorobev A.V., Vorobeva G.R. Web-oriented 2D/3D-visualization of geomagnetic field and its variations parameters // Scientific Visualization. 2017. Vol. 9, No 2. P. 94-101.
10. Bobkov A.E. Virtual globe: history and modernity / A.E. Bobkov, A.V. Leonov // Scientific visualization. - 2017. - No. 2. - P. 49-63.
11. Lunyushkin, S.B. Diagnostics of the boundaries of the auroral oval based on the magnetogram inversion technique / S.B. Lunyushkin, Yu.V. Pensky // Solar-terrestrial physics. 2019. Vol.5, No. 2. P. 97–113.