

Spherical Visualization Database Prototype for Education and Scientific Research: Designing and Managing

Sergey Presnyakov¹ [0000-0001-8710-5494], Grigory Boyarshinov¹ [0000-0003-0062-4350],
Anastasia Odintsova¹ [0000-0002-5037-4778], Alena Rybkina^{1,2} [0000-0002-9223-5936]

¹Geophysical Center of the Russian Academy of Sciences,
119296, 3 Molodezhnaya st., Moscow Russia

²The Schmidt Institute of Physics of the Earth of the Russian Academy of Sciences (IPE RAS),
Moscow, Russia

{s.presnyakov|g.boyarshinov|a.odintsova|a.rybkina}@gcras.ru

Abstract. The article describes methods for spherical visualizing of data as global geophysical, environmental, atmospheric processes on the surface of planets and different kinds of processes on the surface of spherical bodies. Such data can be demonstrated through virtual globes and spherical screens. Volumetric visualization significantly increases the degree of visibility, comprehension and assimilation of the demonstrated content and may be used in the scientific and educational process.

The main goal of this research is to develop database requirements for spherical visualization. The requirements are based on the modern educational and science research representation approaches. The following tasks have been formed and accomplished: a strict classification of data; a convenient way to interact at the stage of adding data to the database and analyzing; their representativeness; introduction of additional data classifications; the possibility of individual user navigation and a high degree of interactivity. For the proposed model, the basic functions of the database management system are described. Requirements to a basic hierarchical data model were substantiated. The main object of hierarchical data model is a spherical Slide representing a separate topic of particular discipline. The superstructure is used above the hierarchical model in the form of an individual route map representing a directory with links to the main database.

Keywords: Spherical Visualization, Scientific Visualization, Database, Hierarchical model, Spherical Screens, Virtual Globes, Learning methods.

1 Introduction. Interactive Spherical Visualizations

Over the past few decades massive amounts of data were produced and their interdisciplinary application relevance has increased. This led to the development of new methods and approaches in working with data: collection, storage and analysis. Of all sorts of data, geospatial data were among the first to be presented graphically. They were used for navigation in the form of simple symbols and maps.

The value of two-dimensional images is difficult to overestimate. However, flat static maps are not always representative enough. For example, in any 2D map projection, there are distortions of lengths, angles, areas, shapes. Thus, creation and implementation of innovative methods for data visualization are highly important.

Methods of interactive Earth 3D visualization or in general visualizations of physical and mathematical models of spherical bodies (interactive spherical visualizations, ISV) are actively developing. Interactivity is understood as a high degree of interplay through Human-Machine Interface (HMI). At the same time interactivity should be ensured with equal success for different user groups and regardless of the type of content.

Following ISV variants were considered by the authors: virtual globes and spherical screens.

1.1 Virtual Globes

The growing interest in three-dimensional visualization GIS applications lead to their active development and implementation. Among the mapping applications with wide functionality for the presentation and analytical processing of geodata on a virtual globe are Google Earth, ArcGIS, Mapbox, etc.

Most of these applications run in the online environments and their main purpose is to simulate a real (or as close as possible to a real) picture of the world based on high-resolution raster space and aerial photographs for various data (terrain, infrastructure, etc.), as well as vector layers (see Fig. 1 [1]).

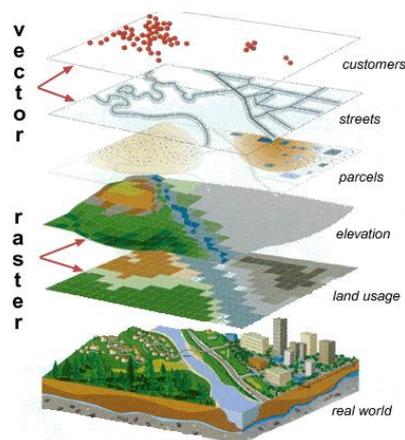


Fig. 1. Raster and vector data layers

The user is provided with functionality, which can include scaling and searching for locations, laying out routes, measuring distances, etc. Creating services with a wide range of visualized data, a convenient interface, and the ability to run online from different devices has significantly increased the requirements for such applications. The main part of such services is the development of the topography (geospatial objects) visualization.

3D models of surface and subsurface objects are visualized in relation to the coordinates of the virtual globe as additional functionality. Not all objects need to be modeled and placed metrically accurately. Many of them are necessary only to create the effect of visual realism and can be modeled with varying degrees of conventionality [2].

The thematic coverage of virtual globes is limited to Earth and planetary data; however, the subject can be significantly expanded. The software environment provides the ability to create and add user generated content, but it is not always user friendly: the presence of highly specialized terms, ability to understand algorithms, etc. Dealing with spatial elements is often limited. Not every system supports working with spherical screens as well.

1.2 Spherical screens

The other way of ISV using is a hardware-software complex (HSC) with a spherical screen. In comparison with the traditional globe the layer on the surface could be chosen by the user. The plane projection of planets on a spherical screen automatically becomes three-dimensional (see Fig. 2). It allows visualizing the global Earth processes as they look from space including additional data, graphics, descriptions, etc., which makes the process of our planet exploration more effective [3].



Fig. 2. HSC “ORBUS”, developed by GC RAS

The disadvantage of a spherical screen is the two-dimensionality of the projection surface itself, which distorts the display of three-dimensional objects on it. Augment Reality (AR) significantly expands the capabilities of the spherical screen. AR is a more advanced method of modeling Extended Reality (XR). The spherical screen onto which high-definition and high-contrast maps are projected is an object of the material environment, visible without additional devices like smartphones. AR cameras use a system of small pictures (AR markers [4]) on a spherical screen or the whole complex as an

AR marker. According to this approach the requirements for real-time rendering on a mobile device are significantly reduced. It is not necessary to render a spherical object with a high-resolution texture or a sequence of textures (video animation) which means that there is no need to transfer large files via the Internet.

2 Spherical visualization for science and education

3D visualization methods are widely used in the scientific and educational process due to significantly enhancing the visibility, understanding and assimilation of the demonstrated content.

Among the features of 3D visualization can be distinguished the following targets: large information capacity, improved perception of information and interactivity.

Information capacity is paramount for large multidimensional data sets. 3D objects provide information through shapes, colors, textures, positions, sizes, orientations, etc. The only limiting factor for 3D visualization is the user's perception.

Spatial representation of data is close to people's perceiving of the world. 3D environment allows the user to choose a better viewpoint for more efficient understanding of the observed data, thereby improving perception of information [5].

Interactivity implies user interaction with content: moving and rotating objects of interest, selecting an object among others, navigation in space, etc. via wide types of additional interface tools.

2.1 Usage for learning purposes

The implementation of new technologies in education is rapidly developing and is found both at the elementary school and at the higher education institutions. It was established in 1989 [6] that 50% of brain neurons are involved in vision. Moreover, it is believed that 3D visualization stimulates more neurons and involves most of the brain in the problem-solving process. Imitation of spatial reality allows the student to learn information faster and develop spatial thinking.

There are various learning approaches:

- extensive, covering many topics and directions, and intensive, focused on one topic and studying it as detailed as possible [7];
- linear, studying the subject in strict sequence in the prescribed volume, and non-linear, selecting individual educational routes of the student [8, 9];
- traditional, corresponding to the "class-lesson" system, when a range of educational disciplines is divided into sections with phased transfer, and modular, in which various topics are grouped into blocks, and objects are studied from the point of view of different disciplines [10];
- passive, in which the main acting figure is the teacher, and students act as passive listeners; active, in which the student can determine the direction, volume, training time; interactive, innovative, in which the student learns new material in the process of solving tasks [11].

An important task in modern education is to increase the learning efficiency of the student. Many educational institutions introduce adjustments related to the replacement or modernization of methods, techniques, forms of organization of educational activities. For example, among the perspective are modular, interactive and non-linear approaches.

ISV is extremely necessary to study and explain many phenomena and processes related to the natural sciences: geography, astronomy, physics, meteorology, etc. A presentation mode for demonstrating images can be set on a spherical screen as a result of which passive linear traditional learning is implemented. The user can interact with ISV by selecting the interesting themes for him realizing an active non-linear approach. Users can learn as many subtopics in the theme as wants to realize a combination of extensity and intensity. The interactive approach is implemented by various responses depending on the user's actions.

2.2 Usage for research and popularization purposes

One of the modern tools for the effective analysis of various scientific data are computer visualization or scientific visualization which is widely used in theoretical and experimental issues.

The main goal of scientific visualization is to make the invisible subjects visible. By invisibility we mean different abstract and real objects. A real object may be invisible if it is 'very large' (for example, the Galaxy) or 'very small' (for example, micro- and nanostructures of the real world). An abstract object (for example, a function of many variables) may be invisible by its intangible nature [12].

The visualization of data obtained during scientific research differs from the others in a number of features. A graphic illustration of scientific data should be given with maximum accuracy and reliability for a qualitative assessment and understanding by scientists. The method of interacting with data should be simple and convenient, maintaining the ability to analyze data with enhanced specific functionality: changing the level of details, certain data stratification for conducting analytical studies, etc. Given the above, the choice of visualization techniques and tools should be accurately selected and may differ from others.

High-quality ISVs are a powerful tool for modeling and systematic analysis of scientific data revealing new facts and patterns. Interactivity, namely changing angles, combining substrates, controlling playback parameters of video animation substrates, working with spatial elements, etc., play a fundamental role in this case.

Recently, among the scientific community around the world the popularization of scientific results and scientific data has been especially important. Scientific popularization means the process of spreading scientific knowledge in an acceptable form for a wide range of people.

However, the knowledge gained in the course of research can often be accessed only by scientists themselves and at the same time be extremely difficult for the perception of non-specialists. To enhance communication between society and science and to demonstrate scientific results the role of such online and offline mediators as museums, exhibitions, science festivals, etc., is fundamentally important.

Researchers in the natural sciences can easily present their scientific results via interactive exhibition stands equipped with modern demonstration devices. The conceptual use of ISV in stands visualizing planetary processes and data is undeniable.

3 Database Management System (DBMS) Requirements

Further requirements for database objects and database model will be described, the functions necessary for data management and implemented variants will be selected.

3.1 Objects

The main object of the database is 'Slide' - usually a unique unit with a single parent element. A Slide is a separate topic of a particular discipline, a digital model of a process on a spherical surface. The Slide is stored as a folder with files.

The 'SlideGroup' object contains a list of child Slides or other SlideGroups. A SlideGroup in this case is a mapping of a discipline, section or subsection into a database. In the SlideGroup you can set the order of the arrangement of the child elements, thus realizing the traditional linear learning route. Each SlideGroup has its own directory, where there is a file containing information about the directories of the child Slides.

A schematic representation of the objects is shown in Figure 3.

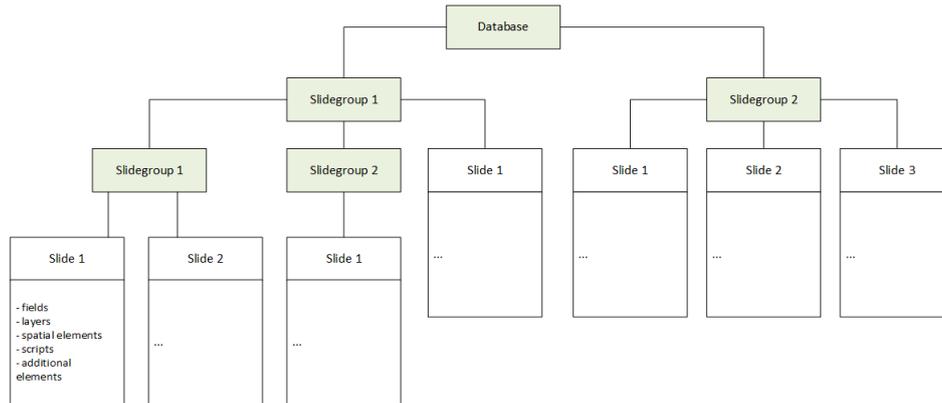


Fig. 3. Database objects

3.2 Structure and composition of a Slide

A Slide is a complex structure that combines several parameters:

- fields describing the main characteristics of the Slide;
- layers in the form of images projected onto a spherical surface;
- spatial objects located at specified surface coordinates;

- scripts that describe the dynamics of processes;
- additional materials such as icons, legend, text description.

All these parameters should be described in a single file in an easily readable and simple format for parsing. The Lua format is suitable in this case, the tables of which can be edited by the user even without special editors and there are many libraries for parsing it using various programming languages. In addition, Lua is a scripting language, scripts can be implemented on it, and OOP is implemented using metatables [13].

The main field of the Slide is 'Name', by which this Slide can be searched.

An example of a Slide description is shown in Fig. 4.

```

MainLayer = {
  Type = "ImageSequence";
  Url = "!resources/06 Earth";
}

Layers = {
  "MainLayer"
}

Slide = {
  NameEN = "The Earth";
  NameRU = "Земля";
  Duration = 40;
  AnimationDuration = 40;
  RotationSpeed = 0;
}

```

Fig. 4. Slide description fragment

A layer is a projection of the surface of a spherical object. The most common projection for displaying processes on the Earth's surface is a cylindrical equidistant projection. Since the main feature of ISV is its focus on global processes, first of all, it provides the basic ability to work with large scales and store layers in a single HD image. Coordination requirements become less accurate, eliminating the necessity for multiple scale levels of layers. These factors simplify the production of content for the database.

Spatial elements on a spherical surface are analogs of discrete objects. They can be placed with help of primitives like points, lines, polygons, meshes, image icons or text.

All unique information about an element is described via attributes. Attributes are information not related to positioning that exists in the form of a 'Name - Value' pair.

Scripts are used to configure user interaction with the Slide. Script allows setting a change on the spherical screen over time, making the Slide dynamic and interactive.

Figure 5 shows an example of Lua code that sets the event of changing the coordinate of an object in 40 seconds after the Slide starts and the event of showing/concealing the element when clicked at a certain point (Figure 5).

```
Object = {  
  Position = {  
    value = Vector2(0,0),  
    event = TimerEvent(40, Vector2(10,0))  
  }  
  Visible = {  
    value = true,  
    event = ClickEvent(Vector2(10,0))  
  }  
}  
  
function Object.ClickEventFunction()  
  Object.Visible.Value = not Object.Visible.Value  
  return Object.Visible.Value  
end
```

Fig. 5. A fragment of pseudo-code script

Thus, text output, movement of objects, display of prompts and etc. can be set.

Each script describes various user interaction options specific to a particular data set.

3.3 Hierarchical model

The objects described above are well organized via ISV database. One of the most important requirements for the ISV database is the simple logic of its structure and the convenience of interacting with data thus the hierarchical model for organizing data can be chosen as a basis.

Hierarchical database model allows to organize data in the traditional linear form of training when it is grouped by discipline. Each discipline has its own topics and sub-topics. The number of branching levels can be unlimited.

Each object in the hierarchical model must have references to the parent and child elements.

Navigation across the entire database structure is simplified in the case of a hierarchical model which makes it easier for administrators to add content.

For described purposes document-oriented databases such as CouchDB [14], MongoDB [15] can be used. Collections implement modules, the link mechanism is implemented through keys. Individual structure is created based on existing thematics with help of training routes on the client side. Those educational approaches needed to add a link mechanism which will be discussed in section 4.2.

3.4 Slide Management System functions

Regardless of choice DBMS a set of basic functions additionally required for the full operation of the Slides. Prototypes of such functions described below.

For searching Slide by words from the search query function findSlideByKeyWord needs.

For creating, editing and deleting Slide functions createNewSlide, editSlide, moveSlide and deleteSlide need accordingly. These functions are available only to the administrators, all actions are recorded in the log. A graphic or console dialog mode is

set after calling first and second functions and Slide is (re)created with the initial data in accordance with section 3.2. If all steps of the dialog mode are successfully completed, data integrity is guaranteed. When new media files are uploaded, a script for checking their validity is automatically executed by calling the commands of some cross-platform utilities for working with media, for example, ImageMagick [16] and FFmpeg [17]:

```
magick identify file.png
ffmpeg -v error -i file.mp4 -f null - 2>error.log
```

Also, to reduce the network client traffic for raster graphics and video, duplicates are created in the WebP [18] and WebM [19] formats, respectively, which provide compression without loss of quality. An example of duplicating media files by calling commands of the same utilities that are described above:

```
magick convert input.jpg output.webp
ffmpeg -i input.mp4 output.webm
```

For retrieving Slide and SlideGroup metadata such as title and icon functions `readSlideTitle` and `readSlideGroupTitle` need. For retrieving other Slide fields described in section 3.2 function `readSlideFields` needs. And for extracting Slide files and single file functions `readSlide` and `readFile` need accordingly.

Assumed arguments and results all functions are shown in Table 1.

Table 1. A set of basic functions of data management system

Function name	Arguments (type)	Result (type)	Short description
<i>findSlideByKeyWord</i>	(str) <i>keyWordSlide</i>	(group str) <i>dirSlide</i>	Find Slide by key word
<i>createNewSlide</i>	(str) <i>dirSlide</i>	(str) <i>dirSlide</i>	Create new Slide
<i>editSlide</i>	(str) <i>dirSlide</i>	(bool) <i>result</i>	Edit Slide
<i>moveSlide</i>	(str) <i>oldDir</i> , (str) <i>newDir</i> , (uint) <i>newPos</i>	(bool) <i>result</i>	Move Slide to a new directory
<i>deleteSlide</i>	(str) <i>dirSlide</i>	(bool) <i>result</i>	Delete Slide
<i>readSlide</i>	(str) <i>dirSlide</i>	(group binary files)	Read Slide as an initial data set
<i>readSlideTitle</i>	(str) <i>dirSlide</i>	(str + binary file)	Read Slide title and icon
<i>readSlideGroupTitle</i>	(str) <i>dirGroup</i>	(str + binary file)	Read Slidegroup title and icon
<i>readSlideFields</i>	(str) <i>dirSlide</i>	(group str) <i>dirFile</i>	Read Slidefields
<i>readFile</i>	(str) <i>dirFile</i>	(binary file)	Read file

4 User interaction system

Large volumes of initial data structured and systematized according to the principles described in section 3 are stored on a file server. The user gets access to the Slides through the web server with help of Application Programming Interface, API (Figure 6). The functionality of the web server and network client implementing the convenient work of the user is considered below.

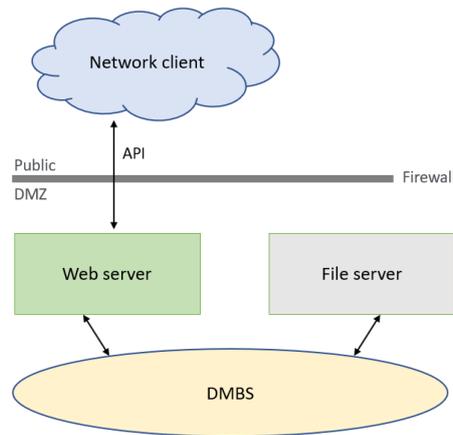


Fig. 6. ISV System Architecture

4.1 Web server functionality

Since the DMBS is located in an isolated zone a web server is used to process external network user requests. Such a server is entrusted with full control over the DMBS: regulation of the load and request queues, prevention of duplicate operations, collisions, etc.

The web server identifies the network client by IP address and API key, giving it access to API functions. A call to API function is equivalent to calling the same name function of the slide management system with parameters transfer according to certain control rules. Referring to the function described in Section 4 means its calling and parameter passing through API.

Data transfer transparency between the network client and the web server is controlled by network exchange functions, for example, XHR [20].

4.2 Network Client Functionality

The user gets access to a personal Slide collection which in a particular case is a subset of the entire database Slide collection and can implement individual data organization logic and add its own data to the Slides. These processes as well as some aspects of working with a spherical screen are described below.

Work with local copies. To avoid repetitive network traffic, an authorized network client can make a local copy of the personal Slide collection available to him. Each Slide accessible to the user is retrieved by calling the *readSlide* function.

To keep the user's Slide collection up to date, it needs regular synchronization. The network client periodically scans the log for the actions of administrators to add, modify or delete Slides. The corresponding operations are performed with the initial Slide files on the client side if such events are recorded.

Individual routes, modules, links. As mentioned above to expand the capabilities of purely hierarchical databases it is necessary to implement a link mechanism. The link is the full path from the root element of the database to the specified Slide and is stored for the client in the table as an entry with a unique key. Access to the link is carried out by a unique key which is necessary for changing the path to the Slide.

The module is a database object Slidegroup combining various topics with listed link keys instead of direct paths.

Individual routes are introduced by adding links to the Slide. A user studying an object of a certain subject can independently switch to areas of interest through additional links.

To work with the mechanism of links the additional hierarchy is set. An additional root directory (route map) is created and can be configured individually by the user. The catalog is a file of Slidegroups associated with various topics and modules. Slidegroups of the secondary directory refer to Slides in the main database. An empty folder is created for each Slide in the additional directory. To implement individual routes, a file is added to the Slide folder of the additional directory - a list of links to Slides of related topics. Files can be added to the additional directory. For example, a text description of the Slide, which will replace the main database files at boot time. An example of the main database is shown in Fig. 7 an additional directory shown on Fig. 8.

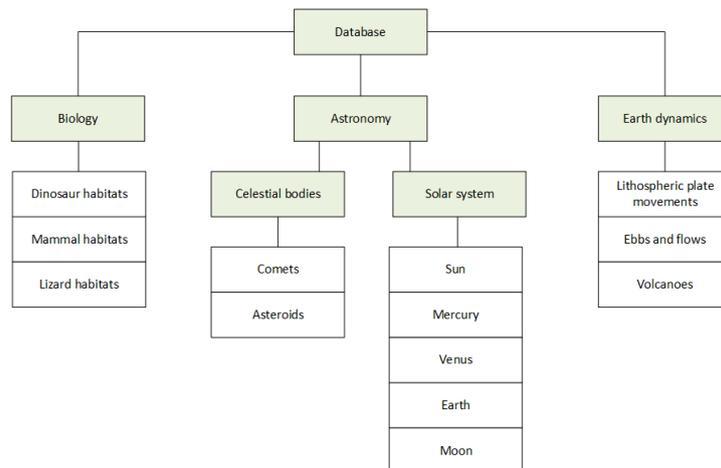


Fig. 7. Main database

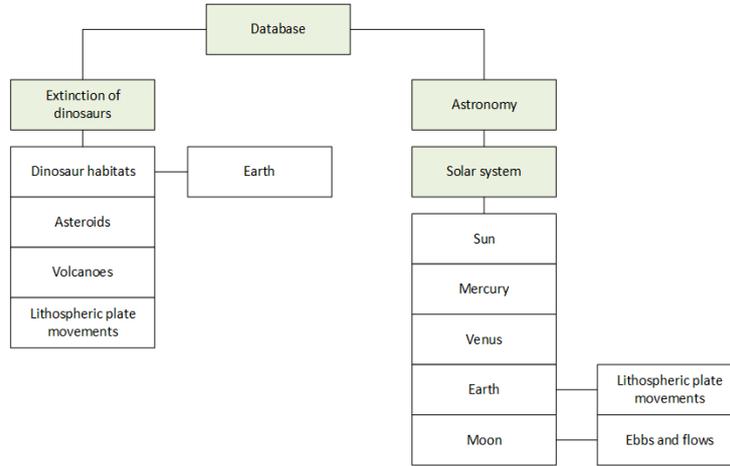


Fig. 8. Additional catalog (route map)

Algorithms to work with route maps. To work with route maps (additional catalog), the implementation of the basic functions described in table 2 is necessary.

Table 2. Functions required to work with route maps

Function name	Arguments (type)	Result (type)	Short description
<i>getLinkPath</i>	(uint) key	(group dirSlide str)	Get link path
<i>changeLink</i>	(uint) key (str) newDir	(bool) result	Change link
<i>updateDir</i>	(str) newFolder	(str) localDir	Store the current position in the catalog
<i>localDataLoad</i>	(str) localDir (group str) SlideData	(group SlideNewData str)	Replace loaded Slide parameters (including file paths) with specified local data

changeLink function is used after synchronization with the main database to make changes to directories for Slidegroups used in links.

The following is the processing algorithm for selecting a Slidegroup by the user:

1. the user selects the desired module or topic;
2. in the additional directory, a list of key links is read from the corresponding file of Slidegroup;
3. getLinkPath function returns an array of paths corresponding to a set of keys;
4. readSlideTitle requests are sent to the main database;
5. according to the received data the user is given a list of names/Slide icons and Slidegroups for further navigation;
6. the current position in the additional directory is updated.

The algorithm for processing the selection of a Slide by the user:

1. the user selects a Slide;
2. in the additional directory the link key is read from the corresponding file of the Slidegroup;
3. getLinkPath function returns the path corresponding to the link key;
4. readSlideFields request is sent to the main database, returning a list of parameters and paths to the Slide files;
5. loadData function is executed to update the Slide parameters;
6. a file is read with additional Slide links specified by keys;
7. getLinkPath function returns an array of paths corresponding to a set of keys;
8. readSlideTitle requests are sent to the main database;
9. according to the data received, the user is given a list of names/Slide icons for further navigation;
10. files, parameters, additional Slide links are transferred to the user;
11. the current position in the additional directory is updated.

The described functions and algorithms implement the basics of working with route maps.

Visualization on a spherical screen and interaction with it. Visualization on the client side is most often performed through the platform-independent software interface OpenGL [21] or its web version of WebGL [22]. The main element of the 3D scene is the sphere with the textures. The geometry of the sphere does not change; its size is single. The position of additional spatial coordinates is more convenient to set in a spherical coordinate system.

The process of projecting a scene onto a spherical screen is described in [3].

Rotation of the spherical Slide by the user is realized by changing the position of the virtual camera (six virtual cameras in the case of a spherical screen) the point of view of which is always directed to the center of the sphere.

Selection of objects is implemented through the ray-tracing algorithm for ray from the center of the sphere and to the point under the cursor (user touch point) [23].

Zoom on a spherical screen can be implemented through the tool 'Loupe'. On a spherical screen a circle is constructed with the center at the point selected by the user. An image from an additional camera is projected inside the circle, aimed at the center of the sphere and mounted on a straight line passing through the center of the sphere and the point selected by the user. The scale is set by the viewing angle of this camera.

Related text and graphic information associated with the Slide can be displayed on standard flat-panel monitors installed nearby or through the application on a mobile device with synchronization via a direct network connection via WiFi or Bluetooth [24].

To work with augmented reality in a mobile application markers and 3D models must be loaded on the user's device. The necessary data is loaded according to the Slide shown. Switching of the Slide makes the data to be deleted from the user's device.

5 Conclusion

This article examined the demonstration of spherical data on virtual globes and spherical screens. The main difference is in the methods of representing 3D data: on spherical screens it is necessary to work with augmented reality, while in virtual globes the whole scene is visualized in one window.

For scientific and educational purposes, it is necessary to conduct a unified classification of data, which is achieved through a hierarchical data model. Images can be used as substrates in the simplest, equally spaced cylindrical projection, without scale levels, which simplifies the creation of content and allows filling the database not only with geospatial data. Modern learning approaches were considered. As reflected in 2.1 an interactive modular non-linear approach was chosen as the basis. The user chooses the topics of learning and interacts with spherical slides according to the implemented scenarios while solving different tasks. Data from different disciplines related to the same objects are grouped into modules. Implementation of this approach as a superstructure to the hierarchical model in the form of route maps, which are directories with links to the slides of the main database is proposed.

A prototype of the client-server architecture for the ISV system is proposed. The main module of it is the database management system. It organizes access at the level of management of the Slide units (as well as its initial files) and the Slidegroups. Relying on its functionality web server processes network client API requests.

6 Further implementation

In the framework of the Project Authors underscore the significant role of ISV for educational and research proposes on several occasions. The article has demonstrated the number of approaches and algorithms that can be effective in the implementation and achievement of the principles and objectives.

The Author's experience could effectively assist to develop ISV software with due regard the guidelines clearly articulated in the article.

7 Acknowledgements

This work was conducted in the framework of budgetary funding of the Geophysical Center of RAS, adopted by the Ministry of Science and Higher Education of the Russian Federation.

This work employed facilities and data provided by the Shared Research Facility "Analytical Geomagnetic Data Center" of the Geophysical Center of RAS (<http://ckp.gcras.ru/>).

References

1. SEOS Homepage, <https://seos-project.eu/>. Last accessed 15 July 2020.
2. Bobkov, A., Leonov, A.: Virtual globe: history and the state of the art. *Scientific Visualization* 2(9), 49-63 (2017).
3. Rybkina, A., Bobkov, A., Nikiforov, O.: Hardware and software system for visualization of geophysical data on a spherical screen. *Scientific Visualization* 2(7), 38-49 (2015).
4. NyID Marker Homepage, <http://sixwish.jp/AR/Marker/idMarker/>. Last accessed 15 July 2020.
5. Li, D., Wang, M., Hu, Q., Hu F.: On three-dimensional visualization of geospatial information: graphics based or imagery based? *Annals of GIS* 5:2, pp. 75-84 (2009), DOI: 10.1080/19475680903250749.
6. Hubel, D. H., Freeman, W. H.: *Eye, Brain, and Vision*. 2-nd ed. Scientific American Library 22 (1995), ISBN:0716760096.
7. Russian textbook (in Russian) Homepage, <https://rosuchebnik.ru/material/vshir-ili-vglubekstensivnoe-i-intensivnoe-obrazovanie/>. Last accessed 15 July 2020
8. Dyachenko, V.: The collective structure of the educational process and its development (in Russian), p. 230. Moscow (2011).
9. Cheredov, I.: The system of forms of organization of training in the Soviet secondary school (in Russian), p. 290. Prosveschenie, Moscow (2012).
10. Shayuk, Yu., Comparative characteristics of classroom and modular learning systems (in Russian). *Modern geographic education: problems and development prospects (Materials of the All-Russian scientific-practical conference)*, pp. 291-295, Pero, Moscow.
11. Zagvyazinskij, V.: *Learning theory: modern interpretation (textbook for high schools)*, 3-rd ed., p. 192. Academy, Moscow (2006).
12. Suhenko, N.: The specifics of the popularization of science in Russia. *Bulletin of NSTU named after Alekseev Management in social systems. Communication technology*.
13. Ierusalimschy, R.: *Programming in Lua*. 3rd edition (2012). ISBN 9788590379850.
14. CouchDB Homepage <https://couchdb.apache.org/>. Last accessed 15 July 2020.
15. MongoDB Homepage, <https://www.mongodb.com/>. Last accessed 15 July 2020.
16. ImageMagick Homepage, <https://imagemagick.org/>. Last accessed 15 July 2020.
17. FFmpeg Homepage, <https://ffmpeg.org/>. Last accessed 15 July 2020.
18. WebM Homepage, <https://www.webmproject.org/>. Last accessed 15 July 2020.
19. WebP Homepage, <https://developers.google.com/speed/webp/>. Last accessed 15 July 2020.
20. XHR Documentation, <https://developer.chrome.com/xhr/>. Last accessed 15 July 2020.
21. Hearn D., Baker, P.: *Computer Graphics with OpenGL*. Prentice Hall 3-rd ed. 2003 ISBN 9780130153906
22. OpenGL ES for the Web Homepage, <https://www.khronos.org/webgl/>. Last accessed 15 July 2020.
23. VBomesh blog http://vbomesh.blogspot.com/2011/11/blog-post_08.html. Last accessed 15 July 2020.
24. Google developers Android <https://developer.android.com/guide/topics/connectivity/bluetooth>. Last accessed 15 July 2020.