Implementation of the Science on a Sphere Visualization System as a Web Application

Jurij Anžič†, Ciril Bohak*,†

University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, SI-1000 Ljubljana, Slovenia

Abstract

SOS is a visualization system originally developed by the U.S. National Oceanic and Atmospheric Administration National Oceanic and Atmospheric Administration (NOAA) to project dynamic global datasets onto a spherical display, enabling audiences to explore complex Earth system processes in an intuitive and immersive way. While highly effective in museums, science centers, and classrooms, the traditional Science on a Sphere (SoS) hardware installation requires dedicated infrastructure, limiting its accessibility and scalability. This paper presentsthe design and implementation of a web-based application that reproduces the core functionality of the Science on a Sphere (SoS) system in a browser environment. The application utilizes modern web technologies to render real-time spherical visualizations of global datasets. Users can load, manipulate, and interact with datasets such as atmospheric phenomena, ocean currents, or planetary imagery without the need for specialized hardware. The system also supports interactive controls for rotating, zooming, and overlaying multiple data layers, extending the pedagogical potential of Science on a Sphere (SoS) by enabling personal exploration on laptops, tablets, and mobile devices. By transitioning Science on a Sphere (SoS) from a physical installation to a lightweight, browser-based platform, the proposed solution broadens access to scientific visualizations, promotes remote and classroom learning, and ensures that Science on a Sphere (SoS) content can be integrated into modern online education ecosystems.

Keywords

Science on a Sphere (SOS), Web-based visualization, WebGL, WebGPU, Geospatial data, Educational technology

1. Introduction

Visualization plays a crucial role in communicating complex scientific phenomena to diverse audiences, ranging from experts to the general public. Large-scale datasets describing the Earth's atmosphere, oceans, and climate systems are inherently spatio-temporal and multidimensional, making them challenging to interpret without interactive and intuitive tools. To address this challenge, the National Oceanic and Atmospheric Administration (NOAA) developed the Science on a Sphere (SoS) platform, which projects dynamic datasets onto a spherical display, providing viewers with a global perspective of scientific processes. Since its introduction, SoS has been successfully deployed in museums, science centers, and classrooms worldwide, where it supports science communication, education, and public engagement.

Despite its success, the traditional SoS installation has several limitations. It requires specialized hardware, including high-resolution projectors and a physical sphere, as well as dedicated space and maintenance. These constraints hinder adoption in settings with limited resources and prevent learners from accessing SoS content outside of specialized venues. Furthermore, as educational practices increasingly move toward digital and remote learning environments, there is a growing need to make interactive scientific visualizations accessible on personal devices such as laptops, tablets, and smartphones.

Web technologies provide an opportunity to overcome these limitations. Advances in Web Graphics Library (WebGL), Web Graphics Processing Unit API (WebGPU), and modern JavaScript frameworks

Human-Computer Interaction Slovenia 2025, October 13, 2025, Koper, Slovenia

*Corresponding author.

†These authors contributed equally.

ig80333@student.uni-lj.si (J. Anžič); ciril.bohak@fri.uni-lj.si (C. Bohak)

https://lgm.fri.uni-lj.si/ciril (C. Bohak)

0000-0002-9015-2897 (C. Bohak)

https://doi.org/10.26493/978-961-293-559-7.4

have enabled real-time rendering of complex 3D graphics directly in a web browser, without the need for plugins or specialized hardware acceleration beyond consumer-grade devices. By leveraging these technologies, it is now possible to implement core SoS functionality as a browser-based application, making global scientific visualizations accessible to a much wider audience.

This paper presents the design and implementation of a web-based application that reproduces and extends the capabilities of the SoS system. The application enables users to interactively explore global datasets on a virtual sphere, including rotating, zooming, and layering multiple data sources. In contrast to the original hardware-bound installation, the web version emphasizes the user-reach, portability, and integration with digital learning platforms. Our goals are threefold: (i) to broaden the reach of SoS visualizations beyond dedicated installations, (ii) to enhance educational opportunities by enabling interactive engagement in remote and classroom settings, and (iii) to provide an open and extensible framework for future scientific visualization applications.

The remainder of this paper is structured as follows. Section 2 reviews related work on spherical visualization systems, educational applications of scientific visualization, and web-based 3D rendering technologies. Section 4 describes the methodology used in implementing the web-based SoS system. Section 5 presents the outcomes of the implementation and performance evaluation. Section 6 discusses the strengths, limitations, and implications of this approach. Finally, Section 7 concludes with a summary of findings and future directions.

2. Related Work

2.1. Digital Earth and Virtual Globes

The concept of a "Digital Earth" has been a driving vision for global-scale visualization systems, emphasizing the integration of geospatial data into accessible, interactive platforms [1]. Virtual globes have become a central tool in this vision, supporting both scientific analysis and public communication of geospatial phenomena [2]. Among them [3], Google Earth[1] has had a transformative impact over the last two decades, widely adopted in research, education, and public outreach. Its accessibility and integration of diverse data sources have demonstrated the potential of virtual globes to democratize access to geospatial information.

2.2. Web-based Virtual Globes and Thematic Mapping

The evolution from desktop to web-based platforms has significantly expanded the reach of virtual globe technologies. Recent work [4] highlights how multiple-view comparisons of geospatial datasets can be effectively implemented in browser environments. Cesium[2], an open-source web-based virtual globe [5], has become a popular framework for thematic mapping and visualization, supporting applications across science, urban planning, and education. National Aeronautics and Space Administration (NASA) has also contributed important web-accessible visualization platforms such as Global Imagery Browse Services (GIBS) [6] and Worldview [7], providing global imagery and near-real-time environmental data to a broad user base. In addition, Google Earth's migration to WebAssembly[3] demonstrates how large-scale applications can be efficiently brought to the browser [8], while educational initiatives such as Earth Voyager [9] extend the pedagogical potential of these systems.

2.3. 3D Web Visualization Platforms

The integration of geospatial visualization with web-based 3D graphics has benefited from advances in web technologies such as HTML5, WebGL, and WebGPU. Early explorations [10] demonstrated the feasibility of web-based 3D analysis and visualization of spatial data using HTML5 and WebGL. More

[1]https://earth.google.com/
[2]https://cesium.com
[3]https://webassembly.org

recent studies [11] have leveraged game engines to deliver web-based 3D visualization of urban big data, reflecting a convergence of gaming and geospatial domains. A broader review [12] highlights the adaptation of technologies from the gaming industry into geospatial visualization platforms, emphasizing performance and interactivity. Emerging rendering engines such as RenderCore [13] extend this trend by harnessing WebGPU for efficient visualization of scientific datasets, demonstrating the potential of next-generation web technologies for scalable and high-performance rendering.

2.4. Geospatial Standards and Coordinate Systems

The effectiveness of global visualization systems depends on robust geodetic standards. The World Geodetic System 1984 (World Geodetic System 1984 (WGS84)[4]) provides the foundation for global positioning, georeferencing, and dataset integration across platforms. Such standards ensure interoperability between different virtual globe implementations, enabling applications to align diverse datasets consistently on a spherical model of the Earth.

Prior work has established a rich ecosystem of virtual globes, web-based visualization frameworks, and rendering engines, making global-scale data more accessible. However, existing systems often require significant computational resources, rely on proprietary platforms, or lack seamless integration into lightweight educational contexts. Our work contributes to this space by adapting the principles of SoS into a browser-based application that combines accessibility, interactivity, and extensibility for educational and scientific visualization.

3. Science on a Sphere

SoS is a visualization system developed by NOAA to enhance public understanding of Earth system science through large-scale, immersive displays. The system projects dynamic datasets onto a physical sphere suspended in space, creating the illusion of a rotating planet (see fig. 1). This format allows audiences to intuitively perceive global phenomena such as atmospheric circulation, ocean currents, climate variability, and planetary exploration data. Since its introduction, SoS has been deployed in museums, science centers, and educational institutions worldwide, where it has become a widely used tool for science communication and environmental education.

The strength of the original SoS lies in its ability to combine scientific accuracy with engaging visualization. Data sources include satellite imagery, real-time environmental observations, and model simulations curated by NOAA and partner organizations. Audiences interact with the globe through guided presentations, enabling educators to contextualize complex scientific processes on a global scale. However, the system's reliance on specialized hardware—high-resolution projectors, spherical displays, and dedicated installation space—has limited its availability to institutions with sufficient resources. These constraints motivated the development of alternative implementations, including the web-based version presented in this work, which seeks to broaden access to SoS content by leveraging modern web technologies.

The dataset preparation pipeline involves several steps as illustrated in fig. 2. First, the base layers are defined (usually from NASA image data), next, the educators and partners define the data and scenarios they would like to visualize on the sphere, which translates into a new SoS dataset. In the following step, these datasets are reviewed by NOAA, where the maintenance is also defined, and finally, these datasets are included in the SoS Dataset Catalog. The system is also available as a desktop application SoS Explorer[5].

[4]https://gisgeography.com/wgs84-world-geodetic-system/
[5]https://sos.noaa.gov/sos-explorer/

Figure 1: The setup of the original SoS system projecting the content onto a spherical projection globe using 4 projectors.

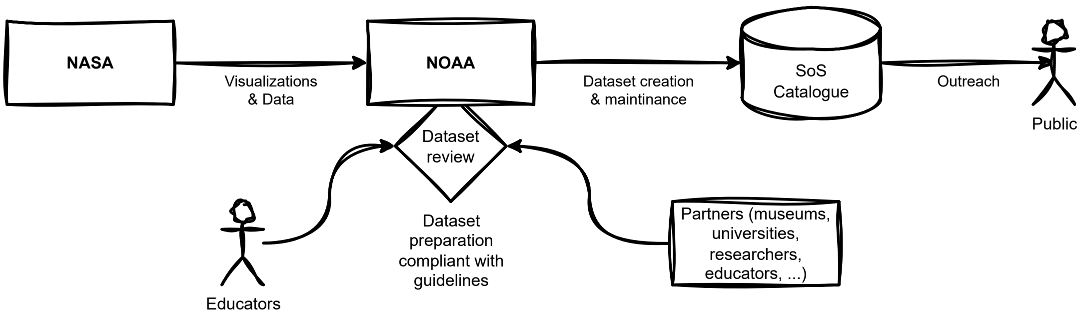


Figure 2: The outline of the dataset preparation pipeline.

4. Methodology

The development of the web-based SoS system followed a modular architecture designed to replicate the core functionalities of the original installation (see fig. 3) while ensuring scalability, portability, and accessibility across devices. The methodology is organized into four main components: data acquisition and preprocessing, rendering pipeline, user interaction design, and system integration.

4.1. Data Acquisition and Preprocessing

The web-based system relies on publicly available geospatial datasets provided by organizations such as NOAA and NASA. Data sources include satellite imagery, atmospheric and oceanographic data, and planetary datasets accessible via services such as NASA's GIBS [6] and Worldview [7]. Acquired datasets are stored in widely used formats (e.g., GeoTIFF, NetCDF, PNG image tiles). Where necessary, preprocessing ensures consistency in resolution, projection, and temporal alignment. Georeferencing is handled using the WGS84 coordinate system[6], allowing datasets to be mapped accurately onto a spherical model.

[6]https://earth-info.nga.mil/?dir=wgs84&action=wgs84

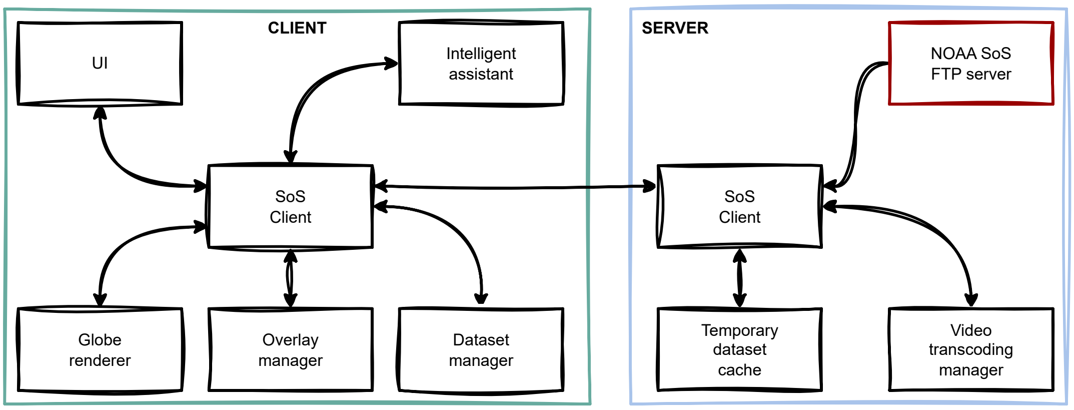


Figure 3: The architecture of our system is composed of client and server, outlining the functional components.

4.2. Visualization Pipeline

For the visualization engine, we use Babylon.js[7], enabling efficient rendering of spherical projections directly in the browser without external plugins. Textures representing global datasets are projected onto a 3D sphere mesh, which serves as the virtual analog of the physical SOS globe. Tile-based rendering strategies are employed for large datasets to support progressive loading and minimize memory consumption, following approaches used in web-based virtual globes [4, 5].

4.3. User Interaction Design

A key design goal was to provide intuitive interaction paradigms that mirror the physical experience of rotating and exploring a spherical display and experience in SoS Explorer. Users can rotate and zoom the globe using standard mouse or touch gestures. Additional interactive features include:

- Layer management for overlaying multiple datasets (e.g., atmospheric circulation over sea surface temperature).
- Temporal controls for exploring time-series datasets, including animation playback.
- Thematic filters, which allow users to highlight or adjust specific data ranges (e.g., temperature thresholds).

Interactive elements are implemented using lightweight JavaScript frameworks, ensuring responsiveness across desktops, tablets, and mobile devices. The US is presented in fig. 4

4.4. System Integration and Deployment

The application was designed as a client-centric web platform requiring no server-side rendering, reducing infrastructure demands and ensuring scalability. Datasets are accessed dynamically through APIs or preprocessed into static tiles hosted on a web server. This architecture minimizes latency and enables offline deployment in controlled environments such as classrooms or exhibitions. The system is fully compatible with modern browsers supporting WebGL and WebGPU, ensuring broad accessibility without additional installation steps.

4.5. Intelligent Assistant

The intelligent assistant module is implemented as a lightweight, local language model that runs directly in the browser using the Web-LLM[8] library. Specifically, it employs the model `Llama-3.2-3B-Instruct-q4f32_1-MLC`, optimized for on-device inference. The assistant provides concise descriptions

[7] https://www.babylonjs.com
[8] https://webllm.mlc.ai

Figure 4: The UI of our system, closely mimicking the UI in SoS Explorer.

of datasets and supports geographic orientation by extracting coordinates from its responses, enabling the camera to automatically fly to the corresponding location on the globe.

4.5.1. System Instructions

Before any interaction, the assistant is initialized with a role definition as a Geographic Information System (GIS) helper operating in the WGS84 geodetic reference system. Responses are typically limited to two to four sentences, without filler or meta-commentary. When possible, the assistant concludes with a line in the format `COORDS: <lat>, <lon>` (degrees). All exchanges, both user prompts and assistant replies, are stored in a conversation history, allowing for contextual continuity in dialogue.

4.5.2. Coordinate Extraction

The preprocessing script first attempts to parse explicit coordinates in the format `COORDS: lat, lon`. If no such line is present, the text is normalized (diacritics and punctuation removed), and the regular expressions and a synonym dictionary are used to identify referenced countries. The system then retrieves the geographic center of the country from the local index. Additionally, the system stores suggested coordinates for selected datasets; if the model outputs "COORDS: Unknown", the response is supplemented with dataset-specific central coordinates, which are then used for the camera jump.

4.5.3. Integration with the Globe

After generating a response, the system attempts to resolve coordinates. If successful, a "Fly to ..." button is appended below the assistant's reply. When clicked, this button moves the camera to the computed location, directly linking the textual explanation with the spatial visualization. Whenever a user selects a dataset, the assistant automatically provides a short summary along with coordinates, ensuring that the globe view aligns with the described content.

4.5.4. Worker Optimization

To improve responsiveness and stability, the assistant runs in a separate worker thread. The worker initializes a single persistent model instance, while the main thread handles the UI and globe rendering. Streaming generation delivers text fragments incrementally as they are produced, significantly reducing perceived latency. On the main thread, fragments are combined and displayed within the rendering loop, ensuring smooth integration with rendering.

4.6. UI Integration

The intelligent assistant is integrated into the system within a new panel, displaying the content generated by the local LLM. Two responses are visible on the right-hand side of the UI shown in fig. 4.

5. Results

The web-based SoS system was preliminarily evaluated with respect to visualization quality, system performance across devices, and educational applicability. We report the outcomes of rendering tests, interaction responsiveness, and demonstrations in classroom settings.

5.1. Visualization Outcomes

The system successfully reproduced the core functionality of the original SoS installation. Global datasets such as atmospheric circulation, ocean surface temperature, and planetary imagery were projected onto the virtual globe with high visual fidelity. Tile-based rendering allowed smooth exploration of large datasets without noticeable delays, while layer management enabled users to combine and compare multiple datasets interactively. Figures 5a to 5c illustrate examples of rendered datasets, including Earth observation imagery and thematic overlays. Figure 5a shows the age of different parts of the sea floor together with the colormap, fig. 5b shows an animated dataset of flights within one day, and fig. 5c shows an animated dataset of carbon emissions.

5.2. Visualization Evaluation

Performance tests were conducted on a laptop with an integrated GPU—Lenovo ThinkPad X1 Gen 13 and a Razer Blade Stealth 13, both running Windows with Chrome—and a smartphone—Google Pixel 6a running Android 15 with Chrome. The application runs smoothly (above 60 FPS) on all devices, with the main shortcoming being the lower loading of larger (video) datasets. Since the UI was originally developed for a desktop screen, it is not suitable for comfortable use on a smartphone. All other functionalities work well.

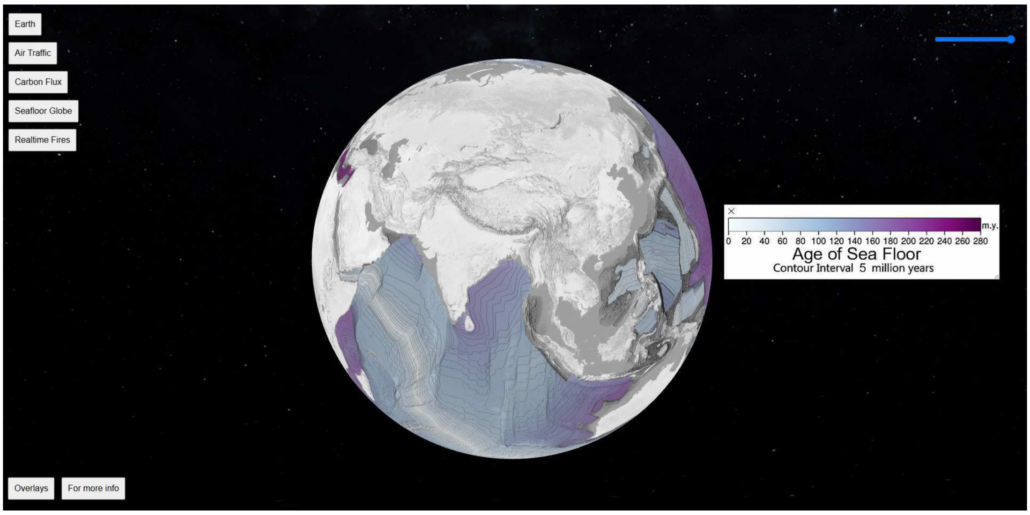
5.3. User Interaction Responsiveness

Interaction responsiveness was assessed for globe rotation, zooming, and layer management. On all platforms, globe manipulation through touch and mouse gestures was immediate, with no perceptible latency. Temporal controls for time-series datasets supported animation playback sufficient for exploring phenomena such as seasonal variation in sea ice extent. It would be beneficial to also implement some on-client loading and caching of the video datasets to avoid pauses due to loading. The overlays could be improved by exchanging them for a vector or multi-resolution representation, which would allow users to zoom in closer to the surface. The same is true for the tiled globe rendering, which only supports a single level of detail.

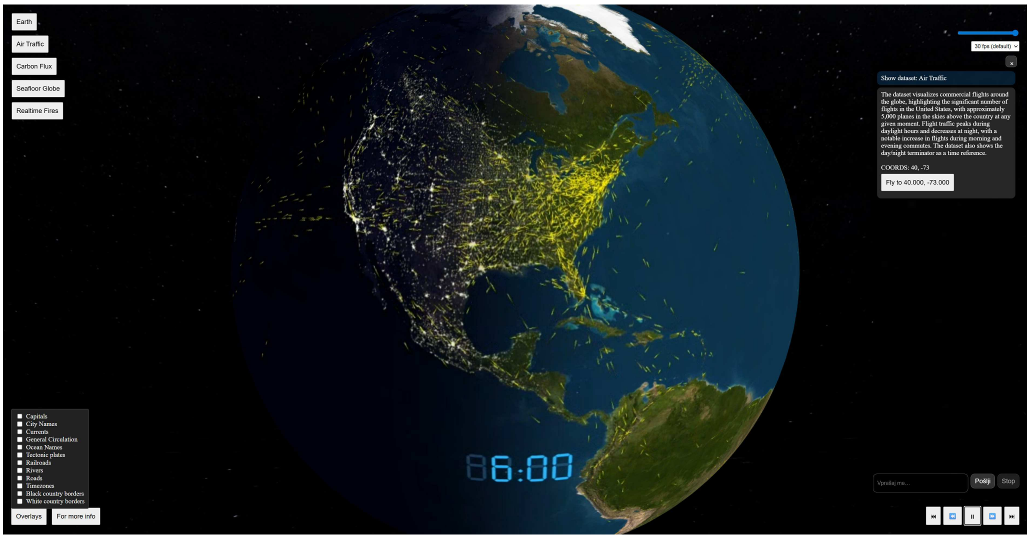
A more thorough system evaluation is planned once more features are implemented.

6. Discussion

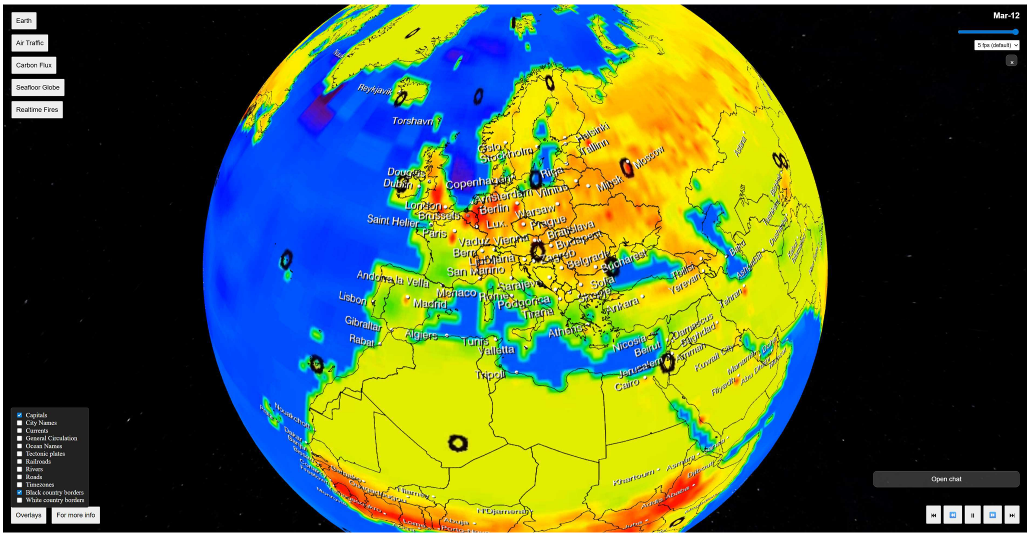
The results demonstrate that a web-based implementation of SoS is both technically feasible and pedagogically valuable. By replicating the essential visualization capabilities of the original hardware-bound system in a browser environment, the platform broadens access to global scientific datasets and enables their use in diverse educational contexts.



(a) Sea floor visualization with color map legend.



(b) Global flights dataset visualization.



(c) Carbon emissions dataset visualization.

Figure 5: Examples of datasets visualized in the web-based SoS application.

A primary strength of the system lies in its broad reach and portability. Unlike traditional SoS installations, which require dedicated hardware and physical space, the web-based version runs directly in modern browsers across desktops, tablets, and smartphones. This cross-platform compatibility allows students, educators, and researchers to engage with SoS datasets regardless of their location or available infrastructure. Performance benchmarks confirm that the system maintains smooth interactivity even on mobile devices, demonstrating the maturity of web technologies for real-time 3D rendering.

Another advantage is extensibility. The modular design allows the integration of new datasets and interaction modes with minimal overhead. For example, APIs such as NASA's GIBS [6] or educational datasets from platforms like Google Earth Voyager [9] could be incorporated seamlessly. This flexibility ensures the system can evolve alongside emerging data sources and pedagogical needs.

6.1. Limitations

Despite these strengths, several limitations must be acknowledged. The browser environment imposes constraints on memory management and parallel processing, limiting scalability for very large spatio-temporal datasets. Additionally, the system currently provides visualization and basic exploratory tools but lacks advanced analytical functionality (e.g., quantitative measurements, cross-sectional analysis), which may limit its use in research-focused scenarios. Finally, user evaluations were limited to informal interviews with a few users; structured usability studies with broader audiences are necessary to validate their educational impact.

6.2. Future Work

Future developments will focus on several key areas. First, the adoption of next-generation rendering engines, such as RenderCore [13], which may significantly improve performance and support more complex visualizations. Second, expanding the set of interaction modes—such as annotation tools, storytelling features, or integration with virtual and augmented reality—could further enhance engagement and learning outcomes. Third, large-scale user studies across different educational settings will be conducted to systematically evaluate the system's effectiveness in supporting comprehension of global processes. In addition, consultations and interviews with educators and domain experts are planned to better understand pedagogical needs and identify potential improvements for classroom use.

6.3. Implications

The transition of SoS from a hardware-dependent installation to a lightweight web application underscores the potential of modern web technologies to broaden access to scientific visualization. By aligning with established educational initiatives in digital Earth and virtual globe research [1, 2, 3], the system contributes to the ongoing trend of leveraging digital platforms for science communication and education. While the implementation demonstrates technical feasibility and a promising framework for accessible, interactive visualization, its educational and pedagogical value warrants further investigation through more comprehensive user studies and longitudinal evaluations.

7. Conclusion

This paper presented the design and implementation of a web-based version of the SoS visualization system. By leveraging modern web technologies, the platform successfully reproduces the core functionalities of the original hardware installation, including spherical projection of global datasets, interactive manipulation, and multi-layer overlays. Performance benchmarks across laptops and smartphones confirmed that the system delivers responsive and visually accurate experiences on widely available devices, demonstrating the feasibility of extending SoS beyond dedicated exhibition spaces.

The transition of SoS to a browser environment has some additional educational implications. It broadens access to global datasets, allowing learners to explore atmospheric, oceanographic, and planetary phenomena in classrooms, at home, or in remote learning contexts. Preliminary demonstrations suggest that the system enhances engagement, fosters spatial understanding, and supports interdisciplinary learning by linking diverse datasets within an intuitive spherical model.

Looking ahead, future work will focus on expanding interaction features, integrating additional datasets, and evaluating the platform in large-scale educational settings. Advances in rendering technologies, along with the integration of storytelling and annotation tools, will further increase the system's interactivity and pedagogical value. Overall, the web-based SoS application highlights the potential of lightweight, scalable visualization systems to democratize access to scientific knowledge and strengthen digital education ecosystems.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT (GPT-5, OpenAI) for grammar and spelling checks, and sentence rephrasing. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] M. F. Goodchild, The future of digital earth, Proceedings of the National Academy of Sciences 109 (2012) 11088–11094. doi:10.1073/pnas.1202383109.

[2] T. Blaschke, et al., Virtual globes: Serving science and society, Information 3 (2012) 372–390. doi:10.3390/info3030372.

[3] S. Liang, et al., Applications and impacts of google earth: A decadal review (2006–2016), ISPRS Journal of Photogrammetry and Remote Sensing (2018). doi:10.1016/j.isprsjprs.2018.08.019.

[4] L. Zhu, et al., Multiple-view geospatial comparison using web-based virtual globes, ISPRS Journal of Photogrammetry and Remote Sensing (2019). Article on web-based virtual globes.

[5] M. Gede, Thematic mapping with cesium, in: ICCGIS 2016, 2016. URL: https://cartography-gis.com/docsbca/iccgis2016/ICCGIS2016-29.pdf.

[6] Global imagery browse services (gibs) — api documentation, https://nasa-gibs.github.io/gibs-api-docs/, 2024.

[7] Nasa worldview — tool overview, https://www.earthdata.nasa.gov/data/tools/worldview, 2025.

[8] J. Mears, How we're bringing google earth to the web (webassembly case study) (2019).

[9] Explore earth – google earth voyager, 2025. URL: https://www.google.com/earth/education/explore-earth/.

[10] K. Chaturvedi, Web based 3D analysis and visualization using HTML5 and WebGL, Master's thesis, University of Twente, Faculty of Geo-information Science and Earth Observation (ITC), Enschede, The Netherlands, 2014. URL: https://hindi.iirs.gov.in/iirs/sites/default/files/StudentThesis/MSc_Thesis_KanishkChaturvedi.pdf.

[11] M. Zheng, H. Zhu, Y. Huang, Web-based 3d visualization of urban big data using game engine: A case study of nanjing, china, in: Proceedings of the 26th International Conference on Geoinformatics, 2018.

[12] A. Heller, M. Gede, From the gaming industry to geospatial: A review of 3d web visualization platforms, in: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, volume 43, 2020, pp. 111–118.

[13] Bohak, Ciril, Kovalskyi, Dmytro, Linev, Sergey, Mrak Tadel, Alja, Strban, Sebastien, Tadel, Matevž, Yagil, Avi, Rendercore – a new webgpu-based rendering engine for root-eve, EPJ Web of Conf. 295 (2024) 03035. URL: https://doi.org/10.1051/epjconf/202429503035. doi:10.1051/epjconf/202429503035.