

WebGL-based Geodata Visualization for Policy Support and Decision Making

Jonas Lukasczyk¹ and Ariane Middel² and Hans Hagen¹

¹TU Kaiserslautern & Computer Graphics and HCI Group, Germany

²Arizona State University, Global Institute of Sustainability

Abstract

Accessible data visualizations are crucial to reaching a wide audience and supporting the decision making processes. With increasing support of WebGL and the new HTML5 standard, the Internet becomes an interesting platform for visualization and data analysis. In contrast to standalone applications, WebGL-based visualizations are easily accessible, cross-platform compatible, and do not require additional software installation. In this work, we propose a WebGL-based visualization that combines various web technologies to interactively explore, analyze, and communicate environmental geodata. As an application case study, we chose an energy consumption dataset for Maricopa County, Arizona, USA. Our study highlights the power of web visualizations and how they can support decision making in collaborative environments.

Categories and Subject Descriptors (according to ACM CCS): H.2.8 [Information Systems]: Database Applications—Spatial databases and GIS H.3.5 [Information Systems]: Online Information Services—Web-based services I.6.6 [Computing Methodologies]: Simulation and Modeling—Simulation Output Analysis

1. Introduction

Visualization tools for data exploration and information extraction greatly enhance the understanding of complex issues and support decision making. To disseminate findings to a wider audience, easy access to visualizations is crucial and often cannot be achieved through standalone applications due to the lack of cross-platform compatibility, complications with distribution, and requirements of installing additional software. In recent years, researchers have recognized these problems and supported a trend towards web based visualizations [SB10] [CLKD09] [WIM*14]. Web applications using the Web Graphics Library (WebGL) have many advantages compared to standalone applications. These applications are easily accessible, as they are Internet based. The resulting visualizations can be retrieved instantaneously without installing additional software, greatly accelerating group development and communication. Furthermore, WebGL-based applications are cross-platform compatible, i.e., the resulting applications are able to run on desktops, notebooks, mobile phones, and tablets. Moreover, the projects asm.js and Emscripten enable developers to compile their C/C++ code directly to JavaScript to a limited extent. WebGL was standardized by the Khronos

Group [Khr10] and in recent years has gained increasing web browser support. Although OpenGL is more powerful than WebGL with regard to its performance and shader support, it is likely that WebGL will be improved with missing geometry and tessellation shaders becoming available in future releases. In addition, a wide variety of existing frameworks and communities will enhance web visualization development even further over time.

Application areas such as urban planning, advertisement, and other fields of public interest can benefit from the use of WebGL. Current examples of application include museums presenting photorealistic replicas on the web [SRWK11] and the automotive industry offering interactive car configurations online [MO12]. Researchers can also easily share results with their peers, collaborators, and stakeholders using this technology.

In this application case study, we developed a WebGL-based geovisualization to interactively and dynamically display environmental data. Georeferenced discrete data are rendered on top of a map using 3D color-coded bars, bezier curves, and particles (Figure 1). As a sample dataset, we used energy data from an ongoing research project that investigates the relationship between urban form and energy use.



Figure 1: Screenshot of the WebGL geovisualization case study. The application consists of two main windows: the 3D rendering (left) and the table visualization (right).

2. Related Work

The benefits of WebGL-applications have already been recognized by many research groups who utilize WebGL for their visualization tasks. Schwartz et al. [SRWK11] used WebGL to make photorealistic visualizations of museum exhibits available to a wider audience. Another example is the work of Resch et al. [RWW14] who visualized 4D marine geodata with WebGL. The potential of WebGL for data visualization is also showcased on the “Chrome-Experiments” [Goo14c] website where Google researchers publish, share, and collect remarkable WebGL-applications. Notable examples are the WebGL Globe [Tea14], which displays global geodata, the Zygote Body demo [Zyg10], an interactive human anatomy visualization, and the ocean wave simulation developed by David Li [Li13].

Given the above, we decided to use the Internet as our target platform and explored 3D geovisualization frameworks. At first we reviewed already existing work based on features of Google Maps [Goo14b], Google Earth [Goo14a], and Cesium [Ana14]. Shrestha et al. [SB10] show that Google Maps can easily be used to visualize geodata and has advantages over commercial Geographic Information Systems (GIS). However, Google Maps currently only supports 3D visualizations via the Flash and Android API. Flash is an outdated technology, and the Android API is limited to mobile target platforms. Isenburg and Shewchuk [IS09] used Google Earth to visualize big LIDAR data sets in 3D, and Chen et al. [CLKD09] propose and illustrate general solu-

tions to visualizing Earth science data via Google Earth. The same authors also utilized Google Earth to display NASA campaign mission vertical profiles datasets [CLK09]. For our study, Google Earth was excluded, since it requires an additional plug-in. Cesium [Ana14], a JavaScript library for creating 3D globes and 2D maps that uses WebGL, is a powerful geovisualization platform due to its geocoding and multi-level tile mapping capabilities. Cesium also supports custom geometries, time, and animation. However, Cesium is not flexible enough for our purposes, although it is a powerful framework for global geodata applications. Our visualization would not benefit from the main feature of Cesium, the multi-level tile mapping capability, since our geodata are of limited spatial extent, e.g., city-blocks, counties, or states. Moreover, in order to be in more control of the rendering and create a specific visual style with Cesium, numerous modifications would be necessary. For example, to show an area of interest with Cesium, the camera position has to be checked in every rendering loop and adjustments have to be performed accordingly. Therefore, we decided to develop our own visualization framework in WebGL.

3. Case Study: EnergyVis

To introduce our visualization framework and demonstrate its application, we chose an energy data set from an ongoing urban planning research project. The energy budget of a city, and therefore, the amount of greenhouse gas emissions, is significantly influenced by a city’s land use and urban form,

i.e., the arrangement and type of built structures. Our visualization facilitates the analysis of the relationship between urban form and energy intensity of a city.

3.1. Energy Data

The geographic extent of our data set is Maricopa County in the state of Arizona, USA. Energy data are associated with travel analysis zones (TAZ), which are small geographic areas roughly corresponding to U.S. Census tract polygons. The energy budget for each TAZ is provided in *Joules* or *CO₂* equivalent in metric tons and can be split into an operational and embedded component. The operational energy refers to energy that is consumed during the in-use phase of the urban environment, e.g., energy used for travel or air-conditioning. Embedded energy refers to the amount of energy that is consumed throughout the whole life-cycle of an urban structure. It includes energy for construction, maintenance, and end-of-life disposal. The data set can further be divided into energy related to built structures and energy related to transportation. The built environment includes single family homes, multi-family homes, commercial structures, and road infrastructure. Transportation energy refers to energy consumed for home-based auto trips for shopping or work. Both the life cycle energy embedded in the built environment and the operational energy use are annualized to provide a basis for comparison.

3.2. Web Interface

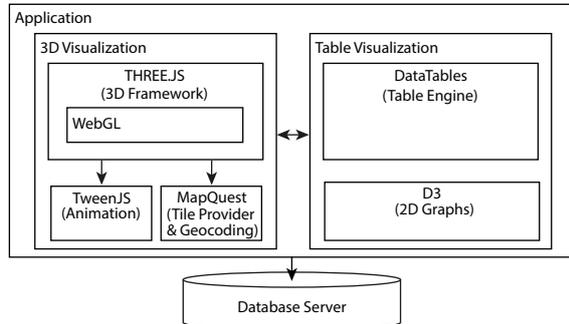


Figure 2: Overview of the software architecture

The visualization framework proposed in this work combines various web technologies, such as D3 [BOH11], DataTables [Jar14], and THREE.JS [Cab14] to interactively display geodata on a map using a web browser. An overview of the implemented technologies is given in Figure 2. The user interface consists of two windows, i.e., the 3D rendering and the table visualization (Figure 1). Inside the 3D rendering, all WebGL functionality is encapsulated in the THREE.JS library. THREE.JS is a powerful framework that enables developers to create and visualize complex 3D

scenes without limitation to a specific context. The geometries, materials, rendering, and element selections inside the 3D scene are implemented using THREE.JS classes.

In our case study, the geovisualization consists of map tiles for spatial context, bars that represent operational and embedded energy of built structures, and a particle system to visualize travel energy data. The height and color of the bars represent user-selected cumulative energy data for each TAZ. Travel data from each TAZ to any other TAZ are visualized using bezier curves and particles traversing the curves. Here, the curve height and number of particles encode the amount of energy consumed for transportation. The map tiles are data from OpenStreetMap [Coa14], and after retrieving the tiles from a MapQuest server [DC14], the images are processed by a costume THREE.JS shader, which converts them into grayscale images. MapQuest is also used for geocoding, enabling users of our energy visualization to search for data at specific latitude/longitude coordinates or addresses. For animation, we used the TweenJS library [Mam14], which facilitates transitions between numerical values of elements inside the THREE.JS scene. We used TweenJS explicitly to animate the bar-height transition, camera movement, and color blending.

If the user selects elements inside the 3D scene, detailed information about those elements is displayed in the table visualization. The table lists various energy categories (columns) and energy values associated with selected TAZs (rows). Each table row features a donut chart that illustrates the composition of cumulative energy values for the corresponding TAZ. The donut charts are created using D3, a library providing data visualizations via HTML, SVG, and CSS. The complete table functionality, such as sorting and row expansion, is controlled by the DataTables library. This library enhances and augments interactive controls of the HTML table element.

3.3. Energy Visualization in the Decision Theater

The Decision Theater at Arizona State University is a 740 m^2 visualization facility that was designed for collaborative and interactive decision-making support. A 260 degrees 7-screen panoramic setup, called “the drum”, serves as the data display to visualize solutions to complex sustainability problems. The Decision Theater actively engages researchers, government agencies, community planners, and policy-makers. To bring our visualization into the Decision Theater (Figure 3), we adapted the GUI layout to the resolution of the drum projectors by editing the style sheets of our website. We also optimized the visualization to match the resolution of dual-monitor systems and tablets. Our visualization framework enables stakeholders at the Decision Theater to analyze and discuss the energy intensity of various urban areas in Maricopa County in a collaborative setting. While interacting with the geovisualization on the panoramic screens, participants can also access the visualization on their own tablets and laptops. By extending our



Figure 3: Demo of our energy visualization in the Decision Theater at Arizona State University

application to process input from these devices, it is possible to visualize user input on the main screen, thus enabling multi-user interaction.

4. Conclusion and Future Work

Our visualization framework demonstrates how the combination of various web-technologies results in powerful visualizations that can be used to disseminate information to a wide audience. We also highlighted the benefits of cross-platform compatibility and portability of those applications. Through an application case study using energy data from urban planning research, we showed the potential of WebGL for the creation of interactive geovisualizations that can inform decision making processes.

Future work includes conducting a user study in the Decision Theater to assess the usability of our application and evaluate our visualization. We will also explore how multi-user interaction can be integrated into our approach. Due to the successful outcomes of this work, WebGL will be our primary target platform for further projects. We anticipate that WebGL will become more powerful and WebGL-based websites will be the future of scientific visualization.

Acknowledgements

This research was supported by the National Science Foundation (Grant 1031690, Urban Form and Energy Use Explored through Dynamic Networked Infrastructure Model; Grant SES-0951366, Decision Center for a Desert City (DCDC) II: Urban Climate Adaptation) and the German Science Foundation DFG (IRTG 1131 at University of Kaiserslautern, Germany). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[Ana14] ANALYTICAL GRAPHICS, INC.: Cesium. <http://cesiumjs.org/>, 2014. Last Accessed: April 28, 2014. 2

- [BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D3: data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309. doi:<http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.185>. 3
- [Cab14] CABELLO R.: Three.js - javascript 3d library. <http://threejs.org/>, 2014. Last Accessed: April 28, 2014. 3
- [CLK09] CHEN A., LEPTOUKHM G., KEMPLER S.: Visualization of NASA Campaign Mission Vertical Profiles using Google Earth. *Geoinformatics, 2009 17th International Conference on*, IEEE Computer Society, pp. 1–5. doi:<http://dx.doi.org/10.1109/GEOINFORMATICS.2009.5293558>. 2
- [CLKD09] CHEN A., LEPTOUKH G., KEMPLER S., DI L.: Visualization of Earth Science Data Using Google Earth. NASA. URL: http://www.isprs.org/proceedings/xxxviii/4-w10/papers/VCGVA2009_01704_Chen.pdf. 1, 2
- [Coal14] COAST S.: Open Street Map. <http://www.openstreetmap.org/>, 2014. Last Accessed: April 28, 2014. 3
- [DC14] DONNELLEY R., COMPANY S.: MapQuest. <http://www.mapquest.com/>, 2014. Last Accessed: April 28, 2014. 3
- [Goo14a] GOOGLE: Google Earth. <http://www.google.com/earth/>, 2014. Last Accessed: April 28, 2014. 2
- [Goo14b] GOOGLE: Google Maps. <https://maps.google.com/>, 2014. Last Accessed: April 28, 2014. 2
- [Goo14c] GOOGLE DATA ARTS TEAM: Chrome Experiments. <http://www.chromeexperiments.com/>, 2014. Last Accessed: April 28, 2014. 2
- [IS09] ISENBURG M., SHEWCHUK J.: Visualizing LIDAR in Google Earth. *Geoinformatics, 2009 17th International Conference on*, IEEE Computer Society, pp. 1–4. doi:<http://dx.doi.org/10.1109/GEOINFORMATICS.2009.5293402>. 2
- [Jar14] JARDINE A.: DataTables. <http://datatables.net>, 2014. Last Accessed: April 28, 2014. 3
- [Khr10] KHRONOS GROUP: WebGL Specification 2.0. <http://www.khronos.org/webgl/>, 2010. Last Accessed: April 28, 2014. 1
- [Li13] LI D.: Ocean wave simulation, 2013. Last Accessed: April 28, 2014. URL: <http://david.li/waves/>. 2
- [Mam14] MAMZELLE: Tween.js - A Javascript library for tweening and animating HTML5 and Javascript properties. <http://www.createjs.com/>, 2014. Last Accessed: April 28, 2014. 3
- [MO12] MORENO I., OTETEA D.: Chrome Experiments - car visualizer. <http://carvisualizer.plus360degrees.com/threejs/>, 2012. Last Accessed: April 28, 2014. 1
- [RWW14] RESCH B., WOHLFAHRT R., WOSNIOK C.: Web-based 4d visualization of marine geo-data using WebGL. *Cartography and Geographic Information Science* 41, 3 (2014), 235–247. URL: <http://dx.doi.org/10.1080/15230406.2014.901901>, <http://dx.doi.org/10.1080/15230406.2014.901901>, doi:10.1080/15230406.2014.901901. 2
- [SB10] SHRESTHA S., BEHR F.-J.: Data Visualization Using Google Maps: the Hard Way and the Easy Way. In *Proceedings of Applied Geoinformatics for Society and Environment (AGSE)* (2010), pp. 125–131. URL: <http://www.applied-geoinformatics.org/>

index.php/agse/agse2010/paper/view/174/109.
1, 2

- [SRWK11] SCHWARTZ C., RUITERS R., WEINMANN M., KLEIN R.: WebGL-based streaming and presentation framework for bidirectional texture functions. In *The 12th International Symposium on Virtual Reality, Archeology and Cultural Heritage VAST 2011* (Oct. 2011), Eurographics Association, Eurographics Association, pp. 113–120. Best Paper Award. URL: <http://diglib.eg.org/EG/DL/WS/VAST/VAST11/113-120.pdf>, doi:10.2312/VAST/VAST11/113-120. 1, 2
- [Tea14] TEAM G. D. A.: Chrome experiments: The WebGL globe. <http://www.chromeexperiments.com/globe/>, 2014. Last Accessed: April 28, 2014. 2
- [WIM*14] WANG F., IBARRA J., MUHAMMAD A., LONGLEY P. A., MACIEJEWSKI R.: What's In a Name? Data Linkage, Demography and Visual Analytics. EuroVis Workshop on Visual Analytics. URL: <http://rmaciejewski.faculty.asu.edu/papers/2014/Names.pdf>. 1
- [Zyg10] ZYGOTE: Zygote Body. <http://www.zygotebody.com/>, 2010. Last Accessed: April 28, 2014. 2