



**HAL**  
open science

## 3D Volume Visualization of Environmental Data in the Web

Eric Braun, Clemens Döpmeier, Stefan Mirbach, Ulrich Lang

► **To cite this version:**

Eric Braun, Clemens Döpmeier, Stefan Mirbach, Ulrich Lang. 3D Volume Visualization of Environmental Data in the Web. 12th International Symposium on Environmental Software Systems (ISESS), May 2017, Zadar, Croatia. pp.457-469, 10.1007/978-3-319-89935-0\_38 . hal-01852629

**HAL Id: hal-01852629**

**<https://inria.hal.science/hal-01852629>**

Submitted on 2 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# 3D Volume Visualization of Environmental Data in the Web

Eric Braun<sup>1</sup>, Clemens Döpmeier<sup>1</sup>, Stefan Mirbach<sup>2</sup>, Ulrich Lang<sup>2</sup>

<sup>1</sup>Karlsruhe Institute of Technology, Karlsruhe, Germany  
eric.braun2@kit.edu | clemens.duepmeier@kit.edu

<sup>2</sup>Ingenieurgesellschaft Kobus and Partner GmbH, Stuttgart, Germany  
mirbach@kobus-partner.com | lang@kobus-partner.com

**Abstract.** The environmental community has an increasing need for visualizations because of the rapidly growing amount of data gathered from various sources e.g. sensors, users and apps. This paper presents a complex visualization for the Web that can be used to get insight into 3D volume data. Volumes like air, lakes and seas are often visualized using 2D slices or one dimensional diagrams that display measured values in a specific point of the volume. To get new insight the 3D volume has to be visualized in 3D. A technique called ray marching, which is known in the computer graphics field in combination with modern web technologies, can be used to create such visualizations for the Web. In addition to the new visualization, this paper also presents the usage of such complex software in a visualization framework created in the same research team. This framework hides the complexity behind user friendly web interfaces that allows one to configure the 3D volume visualization without any programming skills.

**Keywords:** Environmental information systems, 3D volume visualization, WebGL, Lake Constance

## 1 Introduction

The demand for dynamic visualizations of data in the Web grows with the popularity of mobile platforms and many solutions, especially for the visualization of charts and maps, already exist. The WebIS group at the KIT (Karlsruhe Institute of Technology) works on a modern microservice based visualization framework (FlexVis) that is modular, highly configurable and allows the utilization of state-of-the-art technology for implementing reusable and easy to use interactive visualization components which are embeddable into web pages. A new research focus refers to implementing reusable 3D volume visualization components for FlexVis to display complex multidimensional environmental data in the Web. In cooperation with kup (Ingenieurgesellschaft Kobus and Partner GmbH) the development of such 3D data visualizations for displaying the propagation of chemical substances (e.g. sodium chloride, phosphate...) in Lake Constance (“Bodensee”) is explored. Within the existing Web based information system BodenseeOnline one and two dimensional data visualizations of the substance

flow in Lake Constance are already implemented by kup for the LUBW (State Office for the Environment, Measurements and Nature Conservation of the Federal State of Baden-Württemberg). However, since substance flows in a lake essentially depend on three dimensions the new 3D visualization already proves that 3D volume visualizations can provide more information to users of Web based applications like BodenseeOnline.

In this paper, the concept of the 3D volume visualization component will be described in the context of the BodenseeOnline application as one use case. Also, basic concepts of the overall FlexVis framework for data visualization [1] and the integration of the 3D visualization component into this framework will be described. The paper will show that users of web based information systems can benefit from more complex multidimensional visualizations which can be implemented with modern web technologies. Additionally, the paper will show that the usage of the FlexVis framework provides many advantages for the implementation of complex web based information systems.

## **2 BodenseeOnline**

Lake Constance is a transboundary lake situated partly in Germany, Switzerland and Austria and it is an important drinking water reservoir. More than 4 million people get their water at home from the lake without considerable treatment. Beside this, the lake is also used for a fishery, recreational purposes and tourism. For improved monitoring of potential risks which can be caused by emissions into the lake (e.g. from industrialized areas or the intensive use of agriculture around the lake), the environmental decision support and information system BodenseeOnline was developed in an interdisciplinary research project from 2005 to 2008 by five project partners including kup. Using three-dimensional numerical models, the distribution of hydrophysical and biochemical parameters in the lake is simulated on a daily basis using up to date data from measurement stations around the lake. Additionally, results from weather forecasts are used to provide a forecast for the next 78 hours. Measurement data from a wide variety of meteorological and hydrological data are gathered and stored in a database. The results from the simulations as well as the measurement data are visualized on a website and represent an important basis for decision making in case of accidents, disturbances or extreme events in or around the lake for local water authorities, disaster management and drinking water supply companies. The system is currently monitored, maintained and developed further by kup.

Currently, the visualization of the three-dimensional model data is based on an interactive map application which allows the display of time series, profiles and longitudinal as well as transverse sections of data at arbitrary, user-defined positions in the lake. Additionally, the simulated movement of the flow field of substances can be visualized using animated vectors. The propagation of substances in the flow field can be calculated using a particle tracking algorithm. These 1D and 2D visualizations offer extensive possibilities for data analysis and are sufficient for most use cases, like the evolution of water quality parameters at water extraction sites, the vertical distri-

bution of critical parameters such as dissolved oxygen and the movement of substances or objects at the water surface. However, in case of the 3D propagation of local plumes of substances the current methods of 1D and 2D visualization are of limited suitability to capture the dispersion in all three directions. Typical examples are the dispersal of wastewater plumes or the inflow of sediments during flood events, which can form complex patterns of propagation in different depths of water in the lake. Figure 1 shows an example of the latter by visualizing the suspended sediment concentrations at 140 m water depth in the lake during a flood event in the river Rhine. Not unlike an avalanche, the denser riverine water forms a turbidity current, flowing along the lake bottom into deeper regions. The limitations of a 2D visualization in capturing the propagation and extent of the current in the lake are clearly visible. In these cases, a 3D visualization could offer a much more convenient and intuitive analysis of the plume propagation on a qualitative and, maybe with certain limitations, on a quantitative basis.

The users to be reached by this application are mainly:

- sailors who are interested in the wind speed at the surface of the lake
- fishermen who use information of wind speed and the height of waves for their advantage
- divers who are looking for water flow and water temperature information in several depths
- bathers who are also interested in the water temperature (mainly at the surface)
- experts who want to track more complex information about the lake (e.g. suspended sediment concentration)

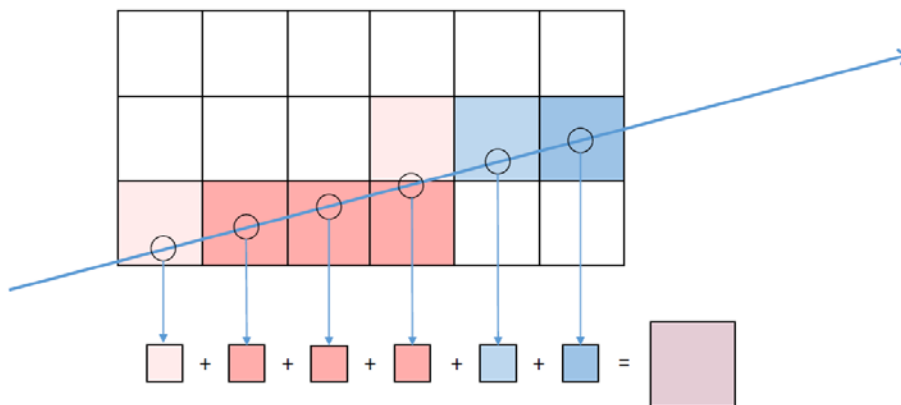
With the current application, these users are provided with a limited view on the data because of the missing 3D visualizations. However, especially divers and expert users are interested in more detailed data since they want to see a full 3D picture of the lake. The next chapters describe enhanced types of visualization for those users.



Fig. 1. 2D slice at 140m water depth visualizing suspended sediment concentration

### 3 3D Volume Visualization

One technique to visualize 3D volumes (e.g. the propagation of substances in a lake) is known from the medical area and is called volume visualization. It is used to visualize MRI scans, for example. However, there are many more use cases since a volume visualization only requires a 3D volume containing spatial information, e.g. density information. This could be the anatomy of a human body (for MRI scans) or the density of different substances that are present in a lake. The rendering of such visualizations is commonly accelerated using a graphics adapter.

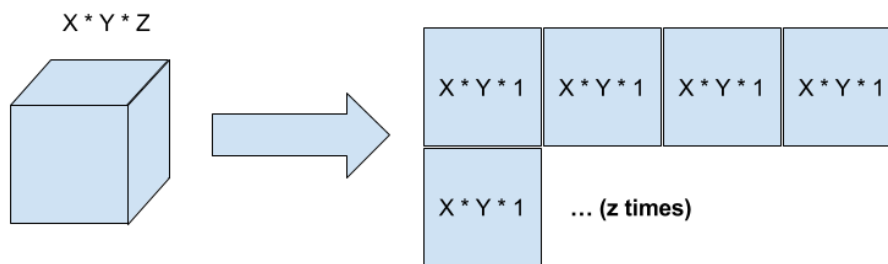


**Fig. 2.** Ray marching and alpha blending

To visualize 3D information on a 2D screen, a ray marching algorithm can be used. The idea of ray marching is that for each pixel a ray is cast through the volume, starting at the camera. The values hit by this ray are captured as a specific step size along the ray. These different values can be translated into a color and the different colors can be accumulated by the graphics card using alpha blending, which is the combination of multiple colors with a specific transparency. Figure 2 shows this technique. The step size in Figure 2 is approximately equal to the length of one square. Therefore, the ray collects the colors of six squares and adds them up to receive a resulting color.

This can be implemented by drawing a cube in the scene which has the same size as the data volume itself. Afterwards, this cube can be shaded with custom vertex and fragment shaders to visualize the volume. The performance depends on the size of the volume and the resolution with which the ray samples the volume (i.e. step size).

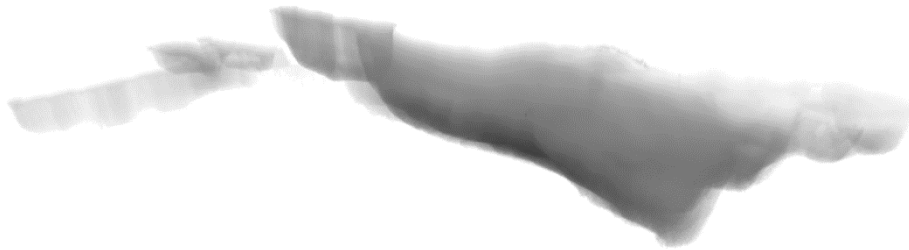
Volume visualizations on desktop computers are typically implemented by using the OpenGL API. As for the Web, the WebGL API, which is supported by modern browsers on a decent computer, can be used. It allows the browser to directly render scenes on the graphics card. The current WebGL version is 1.0 and it implements the OpenGL ES 2.0 [2] features. This OpenGL version was released in March 2007 and it lacks various features of modern graphics adapters. One very important feature for volume visualization is 3D texture support which is unfortunately not included in WebGL 1.0. Therefore, the implementation of such visualizations for the Web has to simulate 3D textures with 2D textures which costs a lot of performance. Figure 3 displays how a 3D texture can be unpacked into a 2D texture. The 3D texture with the size of  $X * Y * Z$  is converted into  $Z$  different slices of 2D textures with the size of  $X * Y$ . This workaround adds a lot of complexity to the fragment shader but it is necessary until WebGL 2.0 is released. More details about the implementation of volume visualization in WebGL 2.0 can be found in the bachelor thesis by Michael Becher [3].



**Fig. 3.** Transformation of a 3D texture into a 2D texture

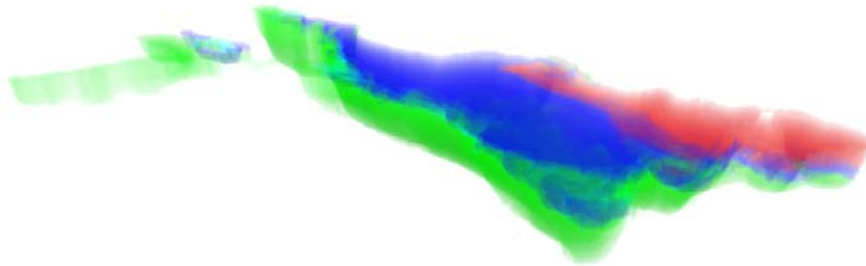
## 4 Lake Constance Volume Visualization

Volume visualization as described in the previous chapter can now be applied to a volume that represents a specific substance present in Lake Constance, e.g. the density of sodium chloride (salt) in the water. The challenge is to determine a good mapping from values to specific colors. If the visualization only uses a grayscale color mode, the values for each step will be normalized between 0 and 1 and the resulting color  $c$  in the RGB color scheme would be  $\text{rgb}(c, c, c)$ . The color scheme can be improved by the creation of a non-linear scaling which matches the human perception more closely. A screenshot of this color mapping for salt in Lake Constance is depicted in figure 4.



**Fig. 4.** Densities of sodium chloride in Lake Constance using the grayscale mode

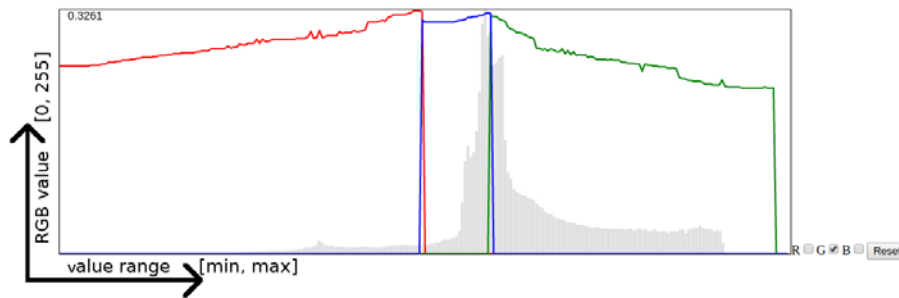
A second approach is to use different color ranges for different density ranges. As shown in figure 5, the lake is divided into 3 colors. The area with the highest salinity is shown in green, medium salinity is colored in blue, and low salinity is colored in red. The comparison between both screenshots shows that different colors provide much more options to show the distribution in the volume.



**Fig. 5.** Different densities of salt in Lake Constance

This second approach allows the user to perceive more details from the visualization, especially if the values are not evenly distributed between the extremes or if there exist clusters of values. Figure 6 shows the color settings of the previous screenshot. The gray lines show the distribution of the different values between the minimal value (left border of the rectangle) and the maximal value (right border). The distribu-

tion has one cluster which was broken down into three different colors to visualize the different parts of the distribution in a clear way. The height of the color line indicates the brightness of the color.



**Fig. 6.** Color mapping and distribution of values

Figure 7 depicts how the volume visualization can be combined with an interactive user interface to build a complex interactive 3D volume visualization component. The color mapping can be dynamically changed and the user sees the changes immediately in the visualization. Furthermore, the user can rotate, zoom and pan the camera of the visualization to look at the volume from different perspectives. The black box in the right upper corner can be used to change parameters that the graphics adapter uses to create the visualization. Such settings might not be included in the interactive user interface of final versions of the volume visualization component because they add too much complexity for the user. However, they can be exposed as configuration parameters to web page creators.

A feature of the 3D visualization component currently missing is some kind of orientation help. The data in our example have geographical meaning and users that know the structure of the Lake Constance should be able to see this structure at first glance. A possible enhancement is an option for the user to enable marks that show the nearby cities or villages. With additional markers, for example Constance and Lindau which are two cities near the lake, the user can immediately recognize familiar points of orientation. A second method which could help with spatial orientation is to add a compass of some sort which shows markers for north, south, east and west. These changes can easily be added to the current prototype and this will be the next step to improve the software. Additionally, in order to help the user select appropriate mappings, future user studies can be conducted to create a more intuitive and user friendly interface.

The prototype presented in this chapter can be integrated in the FlexVis framework for data visualization as an additional visualization type. The procedure for achieving this integration will be discussed in the next section.



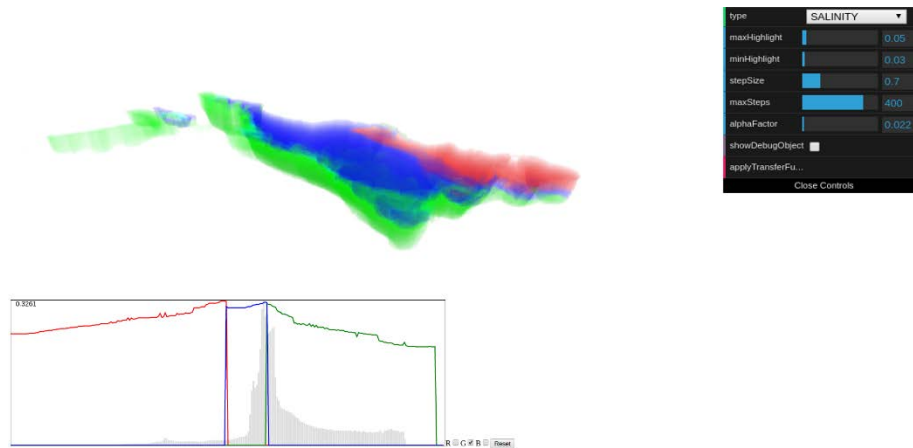


Fig. 7. Complete user interface

## 5 Integration of the Visualization in FlexVis

The FlexVis framework for data visualization in the Web can be used to provide web page creators the opportunity to add highly complex visualization components (such as the volume visualization) to web pages without the need to write any code. Instead, the web page creator can add the volume visualization component to the web page by configuration only. Such a configuration allows the user to change the features that the visualization provides, the behavior of the different features, the style of the visualization, as well as the selection of data sources which determine the data behind the volume visualization.

The data model of the FlexVis framework stores configuration information needed for a visualization in three separate services: a template, a data source and a visualization instance service. The most important configuration information for integrating the new 3D component into FlexVis is the definition of a new visualization type template for the new 3D visualization type. It contains a URL to an HTML5 web component that contains the implementation of the new component. This link allows the FlexVis framework to load the implementation of the 3D component into the web browser. The template also contains formal descriptions of configuration parameters which are used in the source code and will be customizable by a web page creator while embedding the component. Once defined, a template can be instantiated indefinitely on arbitrary web pages with different parameters each time. Finally, the visualization template also describes what parameters have to be defined by a web page creator when instantiating a visualization so that the framework can load data into the component e.g. from external data sources. In its simplest form, the location of data can be defined as simple REST URL that returns a JSON file when the components get instantiated.

FlexVis 1.0 Data Sources Templates Instances

Name  
Volume Visualization


Description

Component Url  
http://localhost:flexvis/components/volume\_visualization/main.html

Tag  
volume-visualization

Parameters

| Name                                       | Data Type | Default Value |
|--|-----------|---------------|
| <input type="checkbox"/> showSettingsPanel | bool      | true          |
| <input type="checkbox"/> stepSize          | float     | 0.7           |
| <input type="checkbox"/> maxSteps          | integer   | 400           |
| <input type="checkbox"/> showColorPanel    | bool      | true          |
| <input type="checkbox"/> controlCamera     | bool      | true          |
| <input type="checkbox"/> alphaFactor       | float     | 0.03          |



**Fig. 8.** Web user interface to configure a template

After a visualization template is defined for the 3D component, one or more instances of the template can be created which contain, beside a link to the template, concrete parameter configurations including the parametrization for data loading for this particular instance. It is important to note that the instance configuration should connect the right data source to the template, e.g. the data model of the data returned by the data source should match the data model used in the web component implementation. If this requirement is fulfilled, the web component should be able to load the data based on the given configuration. The instance can be further customized according to the other configuration parameters that the template features. For displaying a visualization instance on a web page, only the tag of the FlexVis wrapper component with a data parameter specifying the instance id of the visualization instance has to be specified. The wrapper component itself will then load the code of the component and configure it accordingly to the instance parameter configuration.

To illustrate these steps the next figures will show how a new visualization template and visualization instance can be configured. Figure 8 shows the configuration of the template for the new 3D visualization component. The component URL points to the main file that defines the web component for the volume visualization. This file has to be accessible by the frontend browser. The tag defines the name of the web component which is important to embed the visualization into a web page. Finally, the parameters section contains different configuration parameters that are used in the source code. In our example, the configuration allows the settings panel and the color

panel to be shown or hidden, the camera control can be enabled and three further parameters control the rendering behavior of the visualization (*stepSize*, *maxSteps*, *alphaFactor*).

The screenshot shows the FlexVis 1.0 web user interface. At the top is a blue navigation bar with the text "FlexVis 1.0" and links for "Data Sources", "Templates", and "Instances". Below the navigation bar, the interface is divided into several sections:

- Name:** A text input field containing "Lake Constance Volume Visualization".
- Volume Visualization:** A dropdown menu currently showing "Volume Visualization".
- Data Source Instances:** A table with columns "Name", "Data Source", and "Parameters". It contains one entry: "lakeCons" with "lake volume" as the data source and "CONFIG PARAMETERS" as parameters. There is a checkbox to the left of the entry and an edit icon below it.
- Template Parameters:** A table with columns "Name" and "Value". It lists several parameters: "stepSize" (0.7), "alphaFactor" (0.03), "showSettingsPanel" (false), "controlCamera" (true), "maxSteps" (400), and "showColorPanel" (false).
- Data Mappings:** A table with columns "Data Source", "Data Slot Id", and "Data Set Id". It is currently empty.
- Preview:** A 3D visualization of a lake, colored with a gradient from green to blue to red, representing different data values.
- Buttons:** An "APPLY" button is located at the bottom left of the configuration area.

**Fig. 9.** Web user interface to configure a visualization instance

The data source for the volume visualization is a URL that points to a volumetric data set to be loaded. The next section discusses the format of this data set. The resulting visualization instance (Figure 9) can be customized according to the template parameters. The instance can be embedded in any web page by adding the FlexVis web component and specifying the ID of the created visualization instance.

The main advantage of the FlexVis framework is that only the creation of the template requires some programming knowledge but the rest of the parts, e.g. the data source and the visualization instance, can be created and edited by configuration only. This section illustrates that even quite complex visualizations can be integrated in the framework and do not require any programming knowledge once the appropriate template is registered in the framework.

## 6 Data Management for 3D Volumes

The 3D volumes of Lake Constance used in the prototype for this paper are provided by a backend service that stores data in NetCDF files [4]. The exchange format between the service and the visualization is a JSON object that includes a few parameters and the data itself as a three dimensional array. The client visualization converts the array into a JavaScript Typed Array. Figure 10 shows an overview of the used JSON format. The data are serialized as a flattened three dimensional array. The dimensions and interval properties are metadata that describe the size of the data and the range that the values describing the measured substance lie in.

```
{
  "dimensions": {
    "width": 673,
    "height": 230,
    "depth": 57
  },
  "interval": [0, 0.02]
  "data": ...
}
```

**Fig. 10.** JSON object for the phosphate data set

Data management can still be improved in different areas:

- The data sent to the frontend can be reduced in size by varying the resolution of the data if the user zooms out of the scene and vice versa. This can be achieved by pre-computing the volume in different resolutions and selecting the appropriate resolution depending on the zoom level in the frontend. The disadvantage of this change is the higher storage need which scales with the amount of different precomputed resolutions.
- A second way to reduce the size of the data sent to the frontend is to transmit only parts of the whole volume depending on the region the user wants to see. In this case the server has to precompute at least a second resolution that has more detail. This resolution will only be send partially depending on the selection the user makes.
- The backend and both previous improvements can be further improved by storing data in an appropriate database or data structure. This decreases the access time to retrieve the desired data and speeds up queries that only need a part of a volume (as described in the previous improvement). Additionally, the data will be preprocessed before inserting it into the database or data structure to eliminate any processing at runtime. This preprocessing can include handling of missing data and normalizing data.

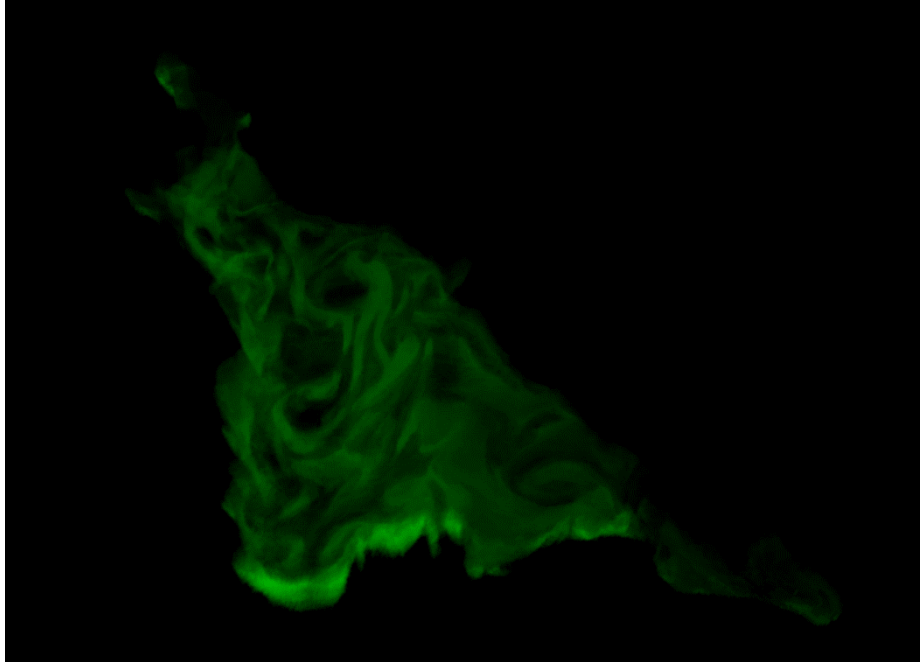
These features are not yet included in the prototype but can be extended in the future without changing core concepts of the architecture.

## 7 Evaluation

After a detailed description of the different aspects of the volume visualization concept, this section evaluates the web application. The main advantage of the new form of visualization is that BodenseeOnline users who are interested in 3D data sets can profit from new insights on the data. Especially divers and expert users, as mentioned in the second section of this article, can use the full advantage of 3D information. Divers can have a look at the water temperature for all depths at the desired diving location. They can immediately see the differences in temperature in a specific volume they are considering to dive in. This is an improvement in contrast to the 1D/2D visualization of the existing web application. Furthermore, expert users benefit from new 3D visualizations of different substances in the water. Figure 11 shows that a simple 3D visualization of the temperature reveals not only information about the temperature itself but also information about the flow of the water which is correlated with its temperature.

A second advantage of the visualization component is the customizability of colors. Users can define their own color mapping to customize the visualization to their needs. A user with color blindness for example can select different colors in order to get better contrast. Average users however do obviously not want to customize the colors in such a complex manner that the current implementation allows them to. Therefore, such users can be provided with some predefined color settings that are easy to understand. An expert on the other hand might want to operate on the full color mapping interface.

The evaluation shows that the requirements for the 3D visualization are met and that new insights on the volume data can be acquired. In order to further enhance the visualization, user studies can be conducted that lead to more specific requirements especially for the color mapping interface.



**Fig. 11.** Visualization of a small interval of temperature values in Lake Constance

## **8 Conclusion and Outlook**

A volume visualization constitutes a complex example of a visualization in the Web. However, its implementation for BodenseeOnline shows that even complex visualizations can be included as easy to use interactive web components into the FlexVis framework to provide a simple configurable interface for web page creators who want to add such visualizations to their web page but do not have the skills to develop a visualization on their own. The different configuration options mentioned in the previous section show that the volume visualization component can help the web page creator to visualize any kind of 3D volumes (not only substance flows in Lake Constance). This can be helpful in many environmental fields. For example, the tracking of substances in water and in air can be visualized in a new way providing 3D insight into the data that cannot be achieved with one or two dimensional visualizations. The flexibility that the visualization framework provides allows the creation of easy to use visualizations for the general public but also rich visualizations with many features for expert users. The only downside of volume visualization is that the client device, which renders the visualization, requires access to an appropriate GPU chip and that the backend has to manage the volumetric data accordingly. However, the number of devices that have no GPU chips has decreased substantially over the past few years. Even most mobile devices have a graphics card that can render visual-

izations presented in this paper. Furthermore, most of the browsers nowadays support WebGL [5].

In the future, volume visualization can be used in many different areas for the visualization of environmental data. As a short term goal, other water bodies can be visualized. As a long term goal, the visualization can be used to show the propagation and the density of pollutants in the air. The scenario is similar to the previous one but the amount of data can be much bigger since lakes or rivers can have much smaller volumes than a volume in the air. Thus, visualizations that cover the air of a whole city can raise new data management challenges and demand big data solutions and smart services for data management.

A further future enhancement could lie in the animation of a volume visualization. Such an animation can display the change of the volume over time. Again, this feature raises new data management challenges because the amount of data increases tremendously. Such an animation also raises new visualization challenges because the animation has to be visualized which means that the graphics card has to compute an image for every frame, including those frames that lie between two volumes that were captured consecutively.

Finally, the user interface consisting of the settings panel, color panel and main visualization can be enhanced in many ways. A more space-saving design can help the user to have more space on the screen for the actual visualization and a common theme for both panels gives the application an enhanced appearance.

## References

1. Braun, Eric et al.: “Generic Web Framework for Environmental Data Visualization”; Advances and New Trends in Environmental Informatics, EnviroInfo 2016, Berlin, Germany, September 14-16, 2016; pp.289-299; DOI: 10.1007/978-3-319-44711-7\_23
2. Khronos Group: “OpenGL ES 2.X”; [https://www.khronos.org/opengles/2\\_X/](https://www.khronos.org/opengles/2_X/); visited September, 26th 2016
3. Becher, Michael: “Interactive volume visualization with WebGL”; Bachelor Thesis, <https://elib.uni-stuttgart.de/handle/11682/2973>, 2012
4. Unidata: “NetCDF”; <http://www.unidata.ucar.edu/software/netcdf/>; visited September, 26th 2016
5. “Can I use”: “WebGL - 3D Canvas graphics”; <http://caniuse.com/#feat=webgl>; visited September, 26th 2016