International Conference on Applied Internet and Information Technologies, 2016 DOI:10.20544/AIIT2016.23

Visualization of flood data using HTML5 technologies

Edis Latifoski[1], Zoran Kotevski[2] and Ilija Hristoski[3]

[1,2] Faculty of information and communication technologies, Partizanska bb, 7000 Bitola, Macedonia
[1]e.latifoski@gmail.com, [2]zoran.kotevski@fikt.edu.mk
[3] Faculty of economics, Marksova 133, Prilep, Macedonia
ilija.hristoski@uklo.edu.mk

Abstract. Human civilization always used to settle near water sources, such as rivers, lakes and seas. But, beside the vast benefits that these regions offer, they sometimes present a huge threat of flooding the inhabited areas, when even the human life could be endangered. Thus, a systems that would offer an early information about the flood state of an endangered region is crucial for all the inhabitants, so they can base their quick actions on real and accurate data and not only on certain speculations. This paper presents a low cost solution that offers precise information about a flooded (or high risk) region using current web technologies. The system presents mapped data visualizations that show the position, the water level and the time of the last condition of multiple marked map points representing the flood sensors placement.

Keywords: Flood visualization, HTML5, WebGL, Cesium map engine, Flood security planning.

1. Introduction

Inhabited areas situated near water surfaces (rivers, lakes and seas) sometime are endangered by floods that can occur due to various environmental factors. In such areas, where beside the significant negative economic impact the human life can be threatened as well, it is crucial that a system that offers early and accurate flood information exists. Security planning for limitation of flood consequences includes many aspects of engineering and analyses, as defined in [1], such as: surveillance of previous and current water levels of the endangered region, statistical and hydrological model analyses, mapping of the flooded regions including the water levels and historic water level data, engineering, design and construction of flood proof buildings, flood state monitoring, weather forecasting and in-time notification of the population. Concerning the possibilities for forecasting and early warning for such disasters on European land, the Joint Research Centre of the European Commission was assigned with the task to develop EFAS (European Flood Awareness System) [2]. As stated in [3,4], the main goal of EFAS is to increase preparedness for floods in trans-national European river basins by providing national hydro-meteorological services with medium-range deterministic and probabilistic flood forecasting information, from 3 to 10 days in advance. We must admit that this effort is of immense importance, but nevertheless, there are times when floods simply cannot be prevented. When these unwanted situations happen the next logical step is to provide the population with immediate and accurate information about the flood state in the region, for which one of the best means is to map the flood water levels. Many systems for flood mapping already exist. For example, Jerad D. in [5] use LiDAR (Light Detection And Ranging) technology to create flood-inundation maps for selected stream gage sites in the North Carolina Tar River basin, where the system exhibits vertical accuracy of 20 cm. Majority of other systems for flood mappings use SAR (Synthetic Aperture Radar) technologies, such as in [6-8]. All these systems present excellent results, but the main drawback of these systems is that they are quite expensive, which was our main motivation to explore a feasible and less expensive solution.

This research deals with the visualizations and mappings of a flooded region and water levels using current HTML5 technologies. Since this research only deals with visualizations and represents only one segment of a larger system, the collection of data using certain flood sensors is assumed. We also assume that these sensors are distributed on multiple locations over the observed region, and in defined time intervals they send flood data in a specific format including location (longitude, latitude), water level and the exact time of the flood state. These data are written in a database wherefrom the visualization module collect's them and updates the map with current flood information. In this manner, the system serves real-time data for the current condition of the flooded region that can be accessed via web interface and used by the population to plan their necessary activities.

The rest of the paper is organized as follows. Section two describes the technologies that are used to realize the task of flood data visualizations. Section three presents the developed system and its functions, while concluding remarks are presented in section four.

2. Web framework for flood state visualizations

As a basic tool to deliver the practical model for flood state visualizations we used CESIUM [9]. CESIUM is built on several HTML5 technologies, among which the most important is WebGL (JavaScript API for rendering interactive 3D computer graphics and 2D graphics within any compatible web browser without the use of plug-ins). We should also bear in mind that these new web standards and technologies are still evolving and some web browsers, even though quickly adaptive, may exhibit certain issues or require the latest updates in order to work properly. CESIUM is a package that is carefully built by an experienced team, statistically analyzed, well documented and publicly tested of over 90% of the code. Cesium was founded by Analytical Graphics, Inc. (AGI) in 2011 as a cross-platform virtual globe for dynamic-data visualization in the space and defense industries. Since then, Cesium has grown into a 3D globe, serving industries from geospatial oil and gas, to agriculture, real estate, entertainment, and sports. AGI leads Cesium's open-source development with support from a growing contributor community.

CESIUM is created with the following main goals and capabilities: visualizations of geospatial dynamic data, high resolution terrain visualizations and variety of supported map layer imagery and sources, sush as WMS (Web Map Service), TMS (Tile Map Service), WTMS (Web Map Tile Service), Bing Maps, Mapbox, Google Earth Enterprise, OpenStreetMap, ArcGIS MapServer, standard image files, and custom tiling schemes.

Each layer can be alpha-blended with the layers below it, and its brightness, contrast, gamma, hue, and saturation can be dynamically changed. It supports industry standard vector formats, such as KML, GeoJSON, and TopoJSON. CESIUM uses data-driven time-dynamic scenes using CZML (Cesium Language) and supports drawing and styling of a wide range of geometries (polylines, billboards, labels, points, polygons, rectangles, circles, ellipses, boxes, spheres, ellipsoids, cylinders, corridors, polyline volumes, and walls). Atmospheric drawing is realized using techniques for fog, sun, sun lighting, moon, stars, and water. It includes camera navigation with mouse and touch handlers for rotation, zoom, pan with inertia, flights, free look, and terrain collision detection. It also supports batching, culling, and JavaScript and GPU optimizations for performance, as well as precision handling for large view distances (avoiding z-fighting) and large world coordinates (avoiding jitter). CESIUM features a 3D globe, 2D map, and Columbus view (2.5D) with the same API.

The task of visualization in this research is realized using the KML data format.

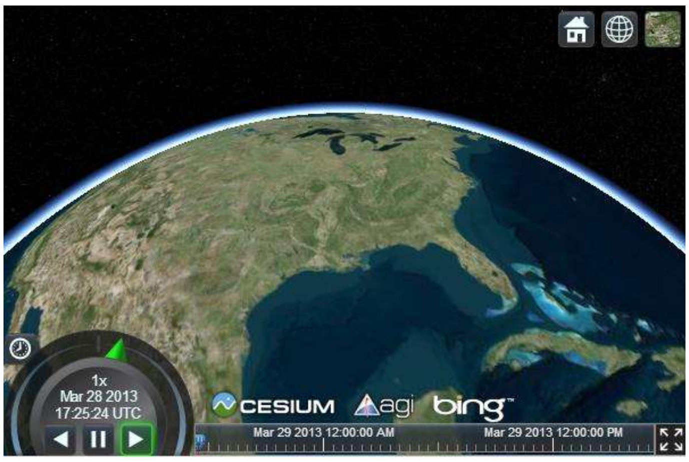


Fig. 1. Sample image from a CESIUM interface

2.1. KML (Keyhole Markup Language)

KML is a XML notation for representation of geographic data and visualizations that are integrated into Internet-based, two-dimensional maps and three-dimensional earth views [10]. KML is created by Keyhole Inc, a company that was officially acquired by Google in 2004. Google Earth is the first application in which KML files can be accessed and edited. Later in 2008 KML data format was officially accepted as an international standard by the OGC (Open Geospatial Consortium).

The KML data format defines variety of properties (places, images, polygons, 3D models, textual descriptions etc.) for map display in many geographic software packages that understand and implement KML. Each point in KML always contains longitude and latitude and other data that can provide specific visual appearance. KML files can be used in compressed KMZ format as well. The KML file can also use 3D geography coordinates (longitude, latitude and altitude), with negative values for west, south and below mean sea level if the altitude data is available. The longitude and latitude components (decimal degrees) are formatted as defined by the World Geodetic System of 1984 (WGS84). The vertical component (altitude) is measured in meters from the WGS84 EGM96 Geoid vertical datum. If altitude is omitted from a coordinate string, e.g. (21.21647, 41.89763) then the default value of 0 (approximately sea level) is assumed for the altitude component, i.e. (21.03647, 41.89763, 0).

A formal definition of the coordinate reference system (encoded as GML) used by KML is contained in the OGC KML 2.2 Specification. This definition references well-known EPSG (European Petroleum Survey Group) CRS components.

3. A model for visualization of flood data

The model that we have developed in this research places real-time data from the MySQL server in a KML data format, which is then visualized on the map using CESIUM. The data inside KML are structurally ordered and contain map point name, geographic coordinates, time of the last change and visual display of the flooded point which is three dimensionally elevated over the mapped region. During the research and development of this system other possibilities for displaying the flood points were also investigated. Some of those possibilities include displaying the data using geometry polygons. Using polygons was better to present the flooded region, but the main drawback was that these polygons weren't suitable for displaying the elevation of the water surface. Other investigated visualization idea was to use lines that would connect the many flood sensor point on the map and to display the full flooded region, but we encountered some problems visualizing this data and for now we concentrated purely on this more simplistic model with the presentation of a certain points by images.

Displaying data by the use of discrete points have shown to be among the simplest and fully functional solutions that encompass all the required data for these flood state visualizations. A single point is represented with its geographic coordinates, including elevation, and can have time attributes. These map points represent the flood sensors that are distributed over the flood endangered region, usually near rivers, lakes or seaside.
The user interface of the developed application is the standard view of the CESIUM frame, which offers several options for viewing the flood data. The options include Globe view, two dimensional top view and 3D perspective view. It is also possible to choose from different mapping imagery. In the system developed in this research we use Globe view and Bing mappings. At the bottom of the interface we placed a time strip from which the user can choose the time at she/he wants to see the flood condition over a predefined time period.

Several short commands are used for the CESIUM frame to load the required functions that are used for graphic mapping. The startup code is displayed in figure 2 which starts with the declaration of variables for function callings required for map display. After the definition of the basic view there is a requirement for definition of a KML data reader. This is presented in figure 3. In the code of figure 3 the KML file is loaded from its location, and the clock multiplier is set to 30 which means that the changes in the water level will be presented quicker than real-time. Further we can hypothesize that the data from the sensors is read once in five minutes by default, but the time frame for which we want to observe the flood visualizations can be interactively defined, as well as the position of the time frame, such as earlier in time.

```
function startup(Cesium) {
var viewer = new Cesium.Viewer('cesiumContainer');
var options = {
camera :viewer.scene.camera,
canvas :viewer.scene.canvas };
```

Fig. 2. Declaration of a function for CESIUM startup

```
onselect : function() {
viewer.dataSources.add(Cesium.KmlDataSource.load('KML_Data.
kml'      ,options)).then(function(dataSource){
viewer.clock.multiplier = 30;
viewer.clock.shouldAnimate = true; }); }); }
```

Fig. 3. Definition of a KML reader

The KML file is generated with a PHP script that makes a connection with the database [11]. The entire structure of the file is stored in a variable. Figure 4 presents a diagram of the whole process from the sensor data generation to the final phase of visualization, and in figure 5, a map that contains visualizations of the lake side in Struga region, R. Macedonia is presented.

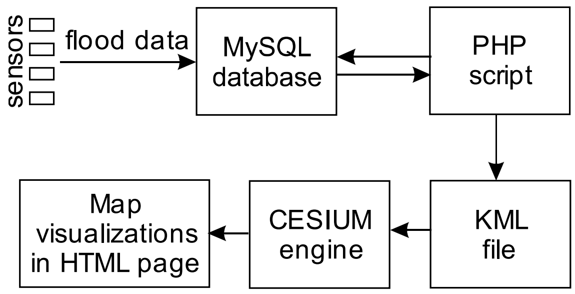


Fig. 4 Diagram of the visualization process

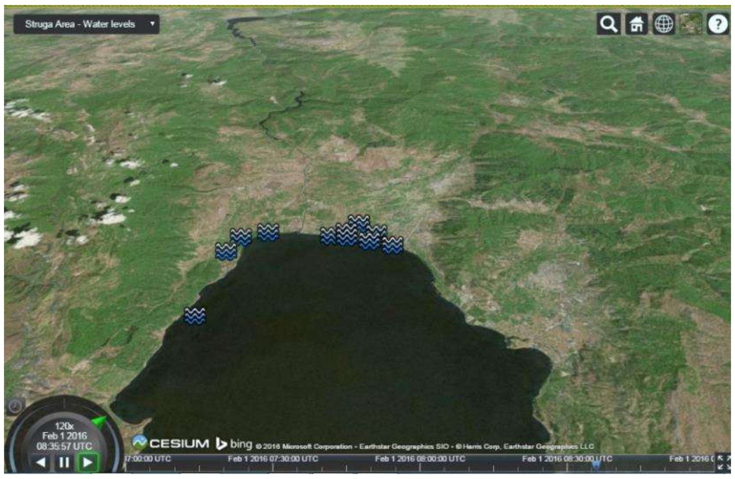


Fig. 5 Top view visualizations of the flooded region

The observation of the sensor points can be experienced from different angles and views, which enhances the user interaction and the visual experience. The angle view is presented in figure 6.

The structure of the KML data format enables representation of multiple points on the same geographic coordinates, where they can be displayed in different time frames. The next code presented in figure 7 creates a point which will be visible, describes that point coordinates along with the water level and contains the time frame for visibility of that point. In figure 8 the point is visually displayed.

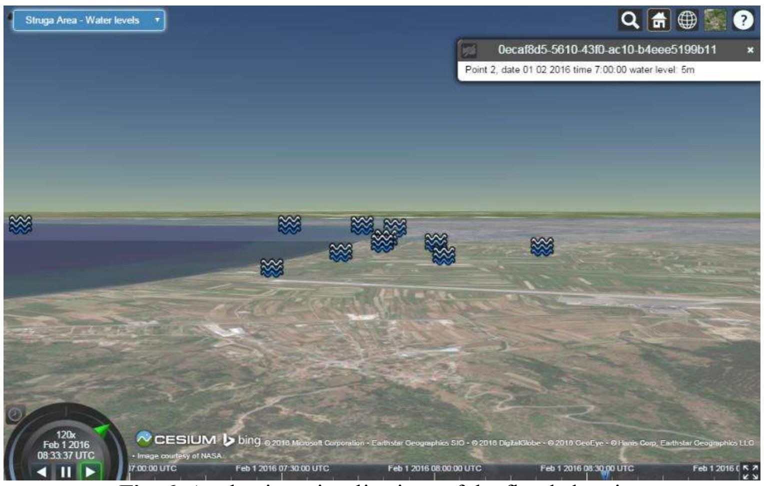
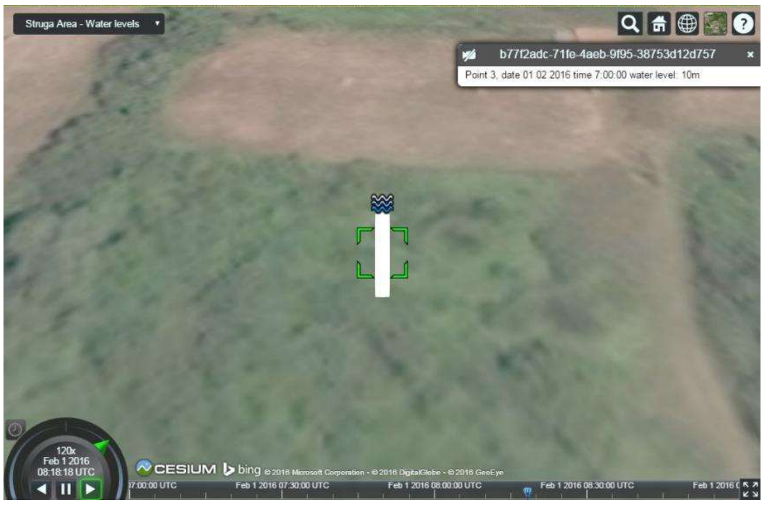

0ecaf8d5-5610-43f0-ac10-b4eee5199b11
Point 2, date 01 02 2016 time 7:00:00 water level: 5m

Fig. 6. Angle view visualizations of the flooded region

```
<Placemark>
<visibility>1</visibility>
<description>Point 1, date 01 02 2016 time 7:00:00 water level:
5m</description>
<styleUrl>#water</styleUrl><Point><extrude>1</extrude><altitudeMo
de>relativeToGround</altitudeMode><coordinates>20.697,41.172,5</c
oordinates></Point>
      <TimeSpan><begin>20160201T07:00:00Z</begin><end>2016-02-
01T07:59:59Z</end></TimeSpan>
</Placemark>
```

Fig. 7. Code for presenting a certain point in a time frame

The main drawback of the KML data format is that the structure of the data is written in a way that the point data is related to a single symbol (image) and cannot be dynamically changed when the flood level value is updated. Therefore the symbol of the flooded point remains the same and the only quick visual indication of the water level at certain time would be if a number representing water level in meters is added. The code in figure 9 represents the image used to display the flooded point in the region, which in this case is a PNG image.



b77f2adc-71fe-4aeb-9f95-38753d12d757
Point 3, date 01.02.2016 time 7:00:00 water level: 10m

Fig. 8. Visualization of the point defined in the code in figure 7

```
<Style id="water">
<LabelStyle>
       <scale>1.5</scale>
</LabelStyle>
<IconStyle>
<Icon>
<href>http://maps.google.com/mapfiles/kml/shapes/water.png</href>
</Icon>
</IconStyle>
<LineStyle><width>10</width></LineStyle>
</Style>
```

Fig. 9. Code for definition of the water image

In the next figure 10, water level image along with the number representing the water depth is presented.

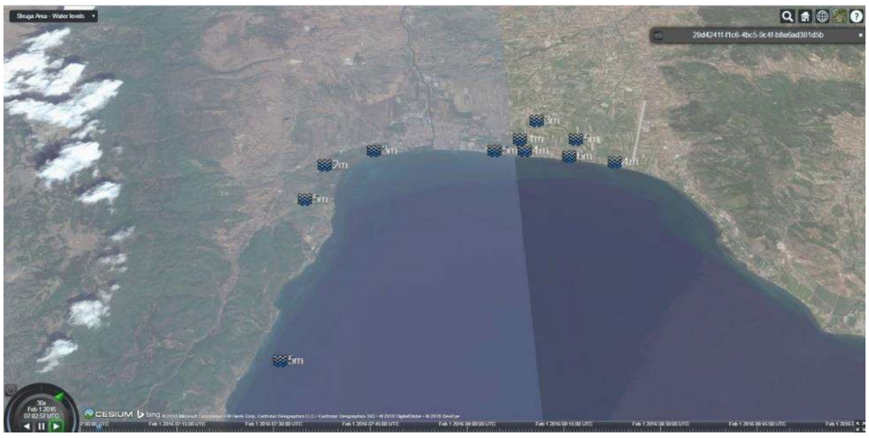


Fig. 10. Visualizations of water levels along with the water depth

4. Conclusion and future work

In this research we presented a model for visualizations of water levels in a certain flooded region. The flood data is assumed to be gathered by flood sensors distributed over the endangered region and in five minute intervals they send flood information to a web server where PHP script writes that data in a MySQL database. The latest data from the database is then extracted by another PHP script which created KML data file and writes it on the web server. The CESIUM module that we have created reads the KML file that contains the complete flood data and updates the map creating current visual information. For the purposes of constant observations of the flood advancement the system offers selection of time frames, as well as historical statistical analyses due to the fact that all the historical information is recorded in the database. Further research will be concentrated on placement of real sensors that would feed the database with real flood data, as well as on a development of a module that would offer early e-mail and/or SMS notifications to the registered users. The final phase of the project would include distribution of the system to include multiple regions that are potentially threatened by flood disasters.

5. References

1. Eychaner, J.H., “Lessons from a 500-year record of flood elevations”, Association of State Floodplain Managers, Technical Report 7, 2015
2. https://www.efas.eu/
3. J. Thielen , J. Bartholmes , M.-H. Ramos and A. de Roo, “The European Flood Alert System – Part 1: Concept and development”, Journal of Hydrology and Earth System Sciences, Vol. 13, pp125-140, 2009
4. J. C. Bartholmes, J. Thielen , M. H. Ramos and S. Gentilini, “The european flood alert system EFAS – Part 2: Statistical skill assessment of probabilistic and deterministic operational forecasts”, ”, Journal of Hydrology and Earth System Sciences, Vol. 13, pp141-153, 2009
5. Jerad D. Bales, Chad R. Wagner, Kirsten C. Tighe, and Silvia Terziotti, “LiDAR-Derived Flood-Inundation Maps for Real-Time Flood-Mapping Applications, Tar River Basin, North Carolina”, Scientific Investigations Report 2007-5032, Geological survey (U.S.) 2007
6. R. Oberstadler, H. Honsch and D. Huth, “Assessment of the mapping capabilities of ERS-1 SAR data for flood mapping: a case study in Germany”, Journal of Hydrological Processes, Vol. 11/10, pp. 1415–1425, August 1997
7. P. Matgen, R. Hostache, G. Schumann, L. Pfister, L. Hoffmann, H.H.G. Savenije, “Towards an automated SAR-based flood monitoring system: Lessons learned from two case studies”, Journal of Physics and Chemistry of the Earth, Parts A/B/C, Vol. 36/7–8, pp. 241–252, ELSEVIER, 2011
8. J.-B. Henry, P. Chastanet, K. Fellah and Y.-L. Desnos, “Envisat multi-polarized ASAR data for flood mapping”, International Journal of Remote Sensing, Vol. 27/10, pp. 1921-1929, Taylor & Francis, 2006
9. https://cesiumjs.org/features.html
10. https://developers.google.com/kml/documentation/kml_tut
11. https://developers.google.com/kml/articles/phpmysqlkml#step-3-using-php-to-output-kml