

Creating Styles, Legends, and Charts for 3D Maps

A Mashup of D3.js, osgEarth, and the Chromium Embedded Framework

Raimund Schnürer, Remo Eichenberger, René Sieber
Institute of Cartography and Geoinformation, ETH Zurich

Introduction

The JavaScript library *D3* enables to visualize and animate data in an attractive way. This work shows how *D3* can be used to style maps as well as to create legends and charts in a real-time 3D environment. Charts are placed as SVG billboards in the virtual globe toolkit *osgEarth*, whereas legends are superimposed the virtual globe with the web browser engine *Chromium Embedded Framework (CEF)*.



Fig. 1 Tree species represented by extruded grid cells and various shades of green and brown

Styles for 3D Maps

Attributes of geographic features can be transformed into styles in *osgEarth* by JavaScript functions (Fig. 2). Thus, *D3* scales can be included to assign, to limit, or to interpolate data values. Modifications allow the chaining of two domains and multiple ranges, and the handling of missing and invalid input values. The result is less code, a better readability and a higher flexibility compared to other styles, like *SLD*.

```
var fill = attribute("tree")
    .unique_values("1 2 3 4 5 6 7 8 9 10 11 12")
    .fill("#156244 #089d08 #427a68 #9ffa72 #68c702 #36fdab " +
        "#badd14 #fafd01 #787d26 #cab315 #dd761a #784501").no_data("#aaaaaa");

function getStyle() {
    return style(
        fill, {
            'fill-opacity': 0.8,
            'altitude-clamping': "terrain",
            'extrusion-height': "1000m"
        }
    );
}

// Calling the function with this variable
feature = {
    'id': 412,
    'attributes': {
        'tree': 2
    },
    'geometry': ...
}

// Returns a CSS-like string for osgEarth
{fill: #089d08; fill-opacity: 0.8; altitude-clamping: terrain; extrusion-height: 1000m}
```

Fig. 2 Source code for styling a grid cell in the map above (Fig. 1)

Automatically generated Legends

Legends are derived from the styles by reusing wrapped *D3* scales. Metadata, e.g. value maps and translations, are added for legend captions, groups and labels. Embedded in a matrix structure, various legend types (Fig. 3) have been developed for choropleth, line, symbol, and raster maps. Changes in map styles are instantaneously adapted in the legend.

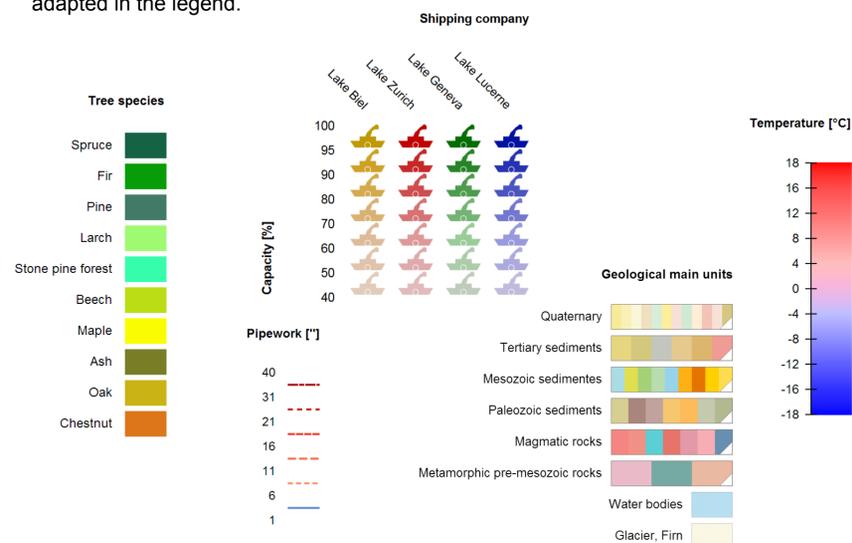


Fig. 3 Legend types for unique values, value ranges, combinations of those, value categories, and color gradients (L-R)

Top 6 Billboard Charts

Six common chart types (Fig. 4) have been implemented using *D3* shapes and scales. I.a. colors of chart segments can be specified and sizes can be calculated from attribute values. An SVG extension of *osgEarth*'s icon driver enables to insert charts as billboards, taking advantage of the automatic decluttering and occlusion culling. Charts are pre-rendered for a better performance.

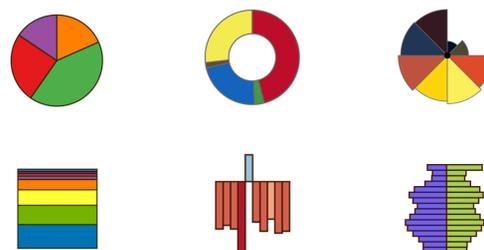


Fig. 4 Pie chart, ring chart, wing chart, divided area chart, bar chart, and divergent bar chart (L-RTB)

Conclusion

The combination of *D3*, *osgEarth*, and *CEF* facilitates the production of 3D maps including legend and charts (Fig. 6). Thus, statistical data can be presented in an interesting way to the public. The approach keeps the source code at a minimum but still maintains the flexibility of further extensions. The JavaScript code may be reused in other contexts (e.g. *WebGL*) and for other cartographic products.

Outlook

- Legends for comparing sizes
- Interactive feature selection by legend
- Improved tilting of charts
- Height-adaptive scaling of billboards
- 3D charts

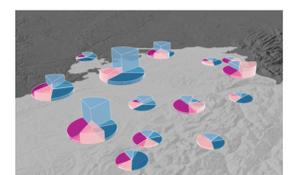


Fig. 5 Extruded pie charts

Acknowledgements

The authors want to express their gratitude to Pelican Mapping, the company behind *osgEarth*, to Mike Bostock, the creator of *D3*, and to all Chromium and Chromium Embedded Framework developers for providing and maintaining their open-source libraries.



Fig. 6 Ring charts displayed as billboards on a virtual globe depicting the electricity consumption in Europe