A PRACTICAL APPROACH TO CREATE ISARITHMIC MAP LAYERS THAT CAN BE USED WITH WEB MAPPING SERVICES

Osman S. Kırtıloğlu, İ.Öztuğ Bildirici

Dr. Osman Sami Kırtıloğlu
Selcuk University
Engineering Faculty, Geomatics Engineering, Division of Cartography, Konya/Turkey
+90332 223 21 20 - okirtiloglu@selcuk.edu.tr

Prof. Dr. İ. Öztuğ Bildirici
Selcuk University
Engineering Faculty, Geomatics Engineering, Division of Cartography, Konya/Turkey
+90332 223 19 38 - bildirici@selcuk.edu.tr

Abstract
This study summarizes a practical approach to create an isarithmic map layer that can be used with online web mapping services such as Google Maps, Bing Maps, MapQuest, Open Street Map and so on. Isarithmic maps are types of thematic maps that represent a continuous field using isolines (lines of equal value) and isopleths (regions of similar value). Isarithmic maps are created by interpolating a set of isolines between sample points of known values. Also there are some basic solutions to create custom thematic maps with web mapping services – like choropleth maps – there is no clear solution to create isarithmic map layers and visualize them on an online map. To find a solution to this problem, we focused on JSON data interchange format for data layer and Multi-quadric Radial Basis Function Interpolation method for interpolating set of isolines. An alternative to XML, JavaScript Object Notation or JSON is a popular data-interchange that is now commonly used by web-mapping tools and web-based APIs. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. There are different types of formalized JSON syntax – such as GeoJSON for storing geodata such as points as longitude/latitude coordinates and shapes and other spatial features and TopoJSON that improves upon GeoJSON by processing GIS data with geospatial topology – for storing different types of optimized data. The main goal of this study is to develop a program with C programming language which starts with segmenting the area of interest (study area border with closed polyline) into cells that's number is determined by the user by taking into account the point in polygon test. Then interpolating sample points and assign interpolated values to each cell (centroid) by using multi-quadric interpolation method. As final step, the program writes all parameters according to GeoJSON format. The output file then can be used as a data layer for many web mapping services.

Keywords*: Web Mapping, Isarithmic Maps, Isopleth Maps, JSON, Thematic Maps, Statistical Maps*

INTRODUCTION

International Cartographic Association (ICA, 2003) defines a map as:

> "*A symbolized image of geographical reality, representing selected features or characteristics, resulting from creative effort of its author's execution of choices, and designed for use when spatial relationships are of primary relevance*".

Over the past two decades, developments in Internet and technology have brought the concept of web-mapping. Web mapping is the process of implementing and visualizing maps on the World Wide Web (WWW or commonly just Web). Since the birth of the Web in 1989 (Tim Berners-Lee, 2016), invented at CERN for the exchange of research documents and shortly after the emergence of the first map server Xerox PARC Map Viewer (Putz, 1994), web-based maps (or web maps) have raised widespread use and have evolved to Rich Internet Applications (RIA) at the present. Web maps are often used to display overlays to bring information to users via the Internet. These overlays may contain information from other web sources (i.e. RSS feeds) and/or users own external data. Getting an overlay onto a web map varies depending upon the mapping API (Application Programming Interface). In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software and applications

(Wikipedia, 2016). An API may be for a web-based system, operating system, or database system, and it provides facilities to develop applications for that system using a given programming language. Of course the way of delivering web maps substantially changed in 2005 when Google introduced its tile-based mapping system based on AJAX that facilitated interactive zooming and panning (Peterson 2012). Just 6 months after the release of the Google Maps, Google has launched Google Maps API which makes it possible to embed a Google Map on any website or web page free of charge and provides overlaying information from other web sources or users own external data. Other online map providers like Microsoft Bing Map, MapQuest, Yahoo! Maps, OpenStreetMap and ESRI include an API. Almost all of these services support adding geometric primitives like point (marker), polyline and polygons as an overlay and creating info windows, putting image overlays on top of the map and external data layer support for common types as KML, JSON or GeoRSS. Some providers like Google, also have advanced tools such as Fusion Tables for data storage, management and overlay as a layer on the map.

There are a lot of alternatives to create a web map that fulfills basic mapping functions and almost all of these are free of charge except some usage limits. Even the majority of them allow creating powerful heatmaps automatically. But thematic mapping capabilities with web mapping services somewhat limited. For example users can make a choropleth map easily using Google Maps API and Google Fusion Tables service with their tabular data. But there is no clear solution about making isarithmic maps which are created by interpolating a set of isolines between sample points of known values. When web map makers want to produce such thematic maps they should produce these thematic layers by third party software and export them as raster images then add to base map as a ground overlay. Otherwise they should develop their own solutions. In this research a practical approach is suggested to create isarithmic map layers that can be used with web mapping services. The following sections address the developed solution. In this context, dataset used for testing the developed algorithm, interpolation method for deriving values between sample points and creating a layer to assign these values and visualize them are discussed in detail.

MATERIAL AND METHOD

Sample Dataset

The sample dataset of this research consist of two main groups. The first dataset can be named as area of interest (AOI). It identifies the boundary of the subject area for thematic mapping. Konya Closed Basin (KCB) border has been used for this purpose which is located at the center of the Turkey's Anatolian peninsula. It has a large area and the boundary data includes approximately 1400 line segments and it is ideal for testing our approach in terms of performance and usability. Second dataset is a set of sample points, includes 12 meteorological stations used to derive isolines of precipitation from meteorological stations located in the KCB. Figure 1 illustrates the KCB boundary and meteorological stations.

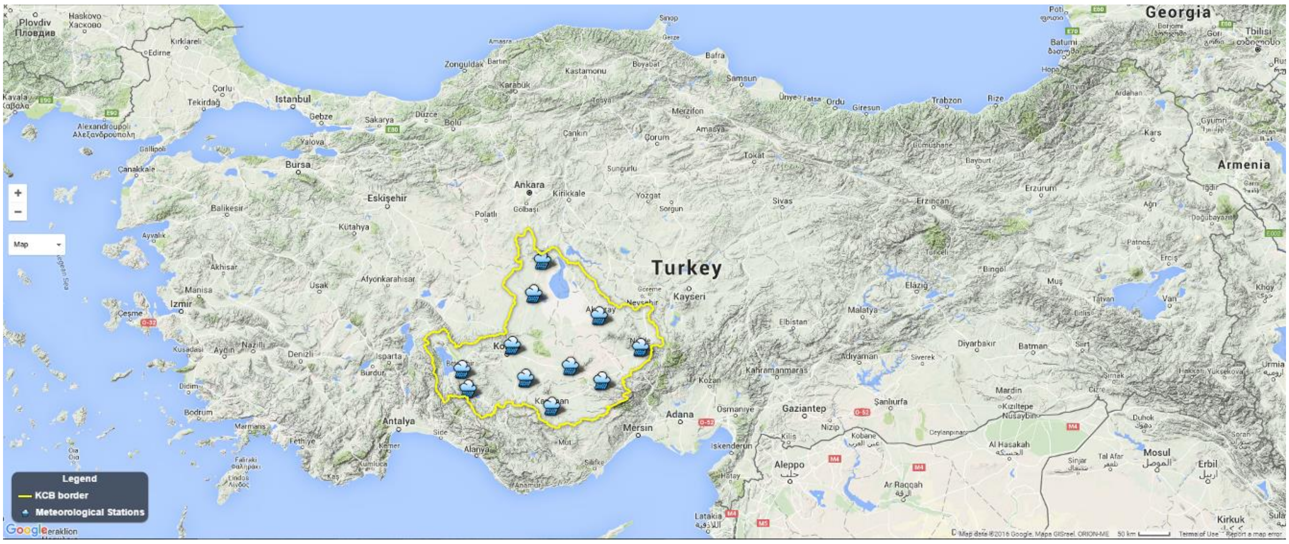

Figure 1. Sample dataset consists of KCB boundary and meteorological stations located in it.

Minimum Boundary Rectangle and Point in Polygon Test

An Isarithmic maps is a type of thematic map that represents a continuous field using isolines (lines of equal value) and isopleths (regions of similar value) (Kraak and Ormeling, 2011). Dividing the area of interest into seamless grid cells, may be a good way to obtain a continuous field from a point or multi-line segment dataset. In this case, size of cells determines resolution of the field. Our approach uses an algorithm that developed with C programming language and requires the dataset in KML (Keyhole Markup Language) format. The first process is defining the Minimum Boundary Rectangle (MBR) of the AOI. After defining the MBR from minimum and maximum coordinates of the dataset, second process draws squares (grid cells) in MBR that's size are determined by user. The challenge is to eliminate the cells occurring outside of the AOI. Figure 2 illustrates MBR and cells occurring inner and outside of the AOI.

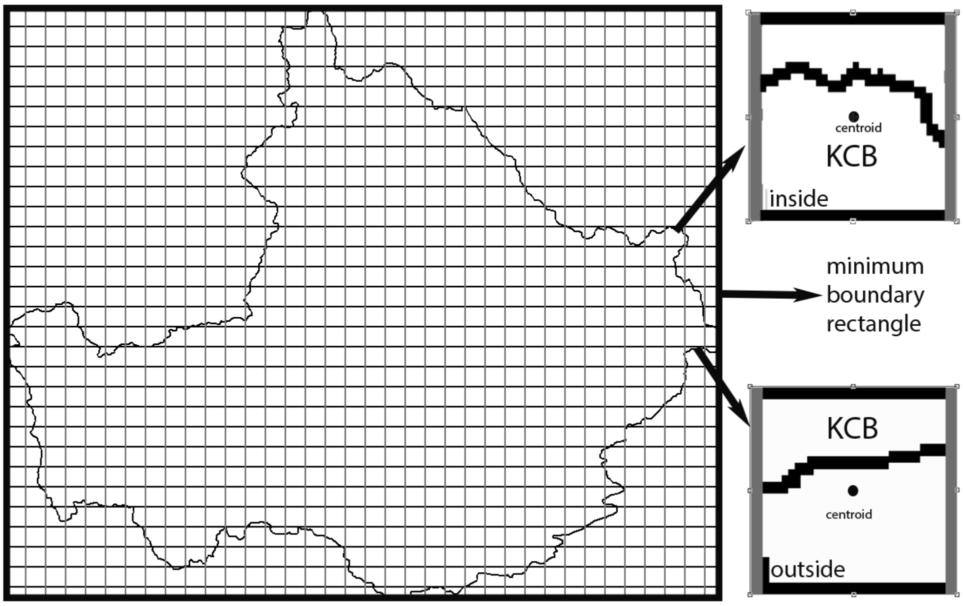

Figure 2. Minimum Boundary Rectangle of KCB and sample inner and outer cells.

The decision of a cell whether in AOI or not, depends on the position of centroid of a cell in our approach. In computational geometry, the point-in-polygon (PIP) test asks whether a given point in the plane lies inside, outside, or on the boundary of a polygon. One simple way of finding whether the point is inside or outside a simple polygon is to test how many times a ray, starting from the point and going in any fixed direction, intersects the edges of the polygon. If the point is on the outside of the polygon the ray will intersect its edge an even number of times. If the point is on the inside of the polygon then it will intersect the edge an odd number of times. Unfortunately, this method won't work if the point is on the edge of the polygon (Shimrat, 1962). Regardless of how complex the shape of the polygon, this simple test gives accurate results. This process is performed by determining the intersection count of two line segments in two-dimensional coordinate space. The test performed here is just for investigating the number of intersection. Therefore the coordinate of intersection point is not required and has not been calculated. The complexity and the number of line segments may increase the number of calculations. For example, KCB boundary has 1400 line segments and that means 1400 calculation of intersection for each line segment should be performed. To avoid this and accelerate the process, the algorithm benefits from a programming technique called "box test". Box test is a logical test that determines probability of intersection of two line segments or overlapping of two bounding boxes. Figure 3(d) illustrates sample boxes and Figure 3(b) and (c) illustrates intersecting and non-intersecting boxes respectively. Box coordinates can be obtained from the maximum and minimum coordinate values of the relevant object. If both equation (1) and (2) given below are provided it can be said that intersection is not possible, otherwise intersection is possible. By this simple test redundant calculations can be avoided and process can be accelerated (Cromley, 1989).

$$\max(y_i, y_j) < \min(y_k, y_l) \quad (1)$$

$$\max(x_i, x_j) < \min(x_k, x_l) \quad (2)$$

To find whether a centroid of a cell is inside or outside KCB boundary polygon, first we have defined a point outside of the KCB box (in Figure 3(e) defined as "i") by using;

$$x_i = \min(x) - (\max(x) - \min(x)) \quad (3)$$

$$y_i = \min(y) - (\max(y) - \min(y)) \quad (4)$$

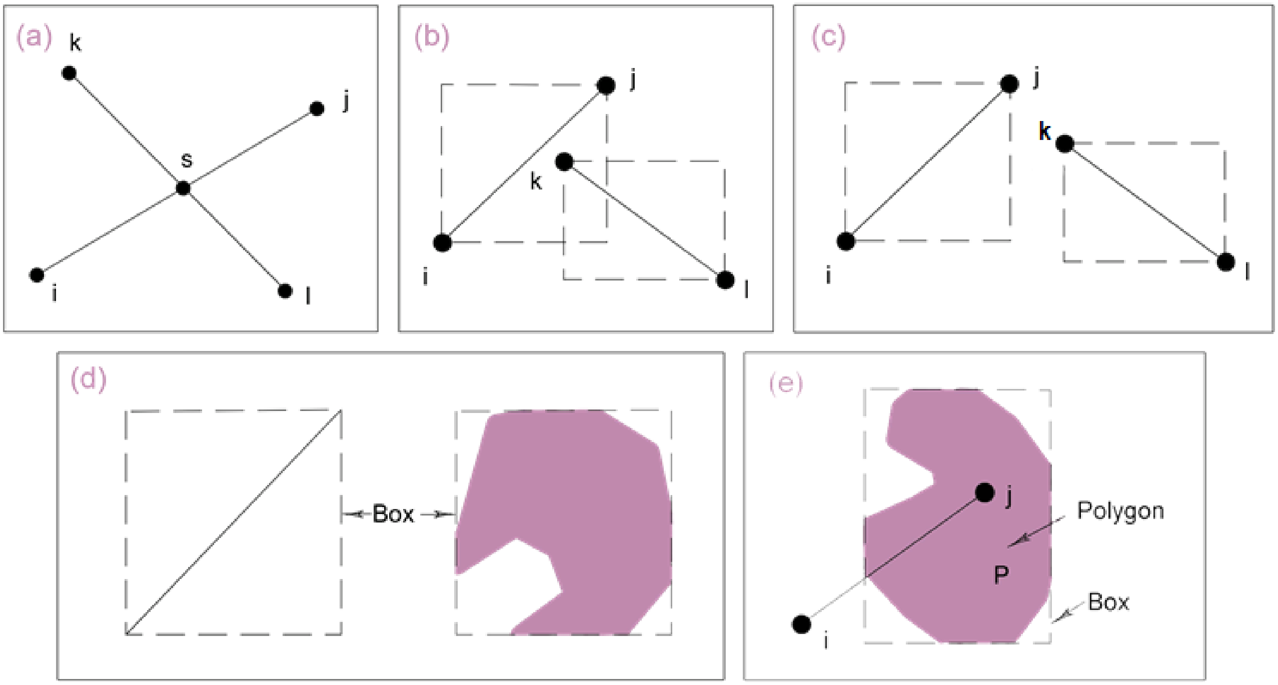

Figure 3. (a) intersection of two line segments (b) boxes are intersecting (c)boxes are not intersecting (d) boxes of a line segment and a polygon (e) an outer point of box i; a point in the box j.

Each centroid of cells tested by the point “i” and evaluated by the number of intersection. First step is investigating if the two line segments are parallel or not. Because, if they are parallel to each other it is clear that intersection is not in question. The parallelism between two line segments can be checked by following method:

If as shown in Figure 3(a) starting and ending points of first line segment defined as i(x_i, y_i) and j(x_j, y_j) and second line segment k(x_k, y_k) and l(x_l, y_l) and “s(x_s, y_s)” is the intersection point, then;

$$d = (x_l - x_k)(y_i - y_j) - (x_i - x_j)(y_l - y_k) \quad (5)$$

d=0 means that two line segments are parallel and there is no intersection. If d≠0 then the parameters p_1 and p_2 are calculated;

$$p_1 = \frac{(x_i - x_j)(y_k - y_i) - (x_k - x_i)(y_i - y_j)}{d} \quad (6)$$

$$p_2 = \frac{(x_k - x_i)(y_l - y_k) - (x_l - x_k)(y_k - y_i)}{d} \quad (7)$$

After p_1 and p_2 parameters are calculated, $0 > p_1 > 1$ and $0 > p_2 > 1$ conditions should be discussed. If both conditions are provided, it can be said that line segments are intersecting, otherwise there is no intersection. This process is applied to each line segment and by the number of intersections, cells located out of the polygon are eliminated. After applying point-in-polygon test the appearance of 2’x2’ grid cells are shown in Figure 4(a).

Interpolation of Sample Points

In this research Multiquadric Radial Basis Function is used to assign values to each grid cell, interpolated from sample points. The multiquadric function which is first published by R.L. Hardy (1971), determines a surface from sample points. The unknown values for an any (x, y) pair can be calculated from sample points as;

$$z=\sum_{i=1}^{n} C_i\sqrt{(x-x_i)^2+(y-y_i)^2+\Delta^2} \quad (8)$$

where "z" is the unknown value of an any (x, y) pair. Δ is the parameter that defines the smoothness of the surface. There are a lot of discussions about the value of Δ. For suggestions and applications about this subject see Hardy (1990). In our approach we have used the following equation for Δ.

$$\Delta^2=\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left[(x_i-x_j)^2+(y_i-y_j)^2\right]}{n(n-1)} \quad (9)$$

Once unknown values are assigned to each cell, colorizing cells based on the given value is now possible and as simple as writing a few javascript code depending on which mapping API is used. Following section discusses the appropriate file formats and loading these files on a web map with mapping API's. Figure 4(b) illustrates the generated isarithmic map layer by using multiquadric function.

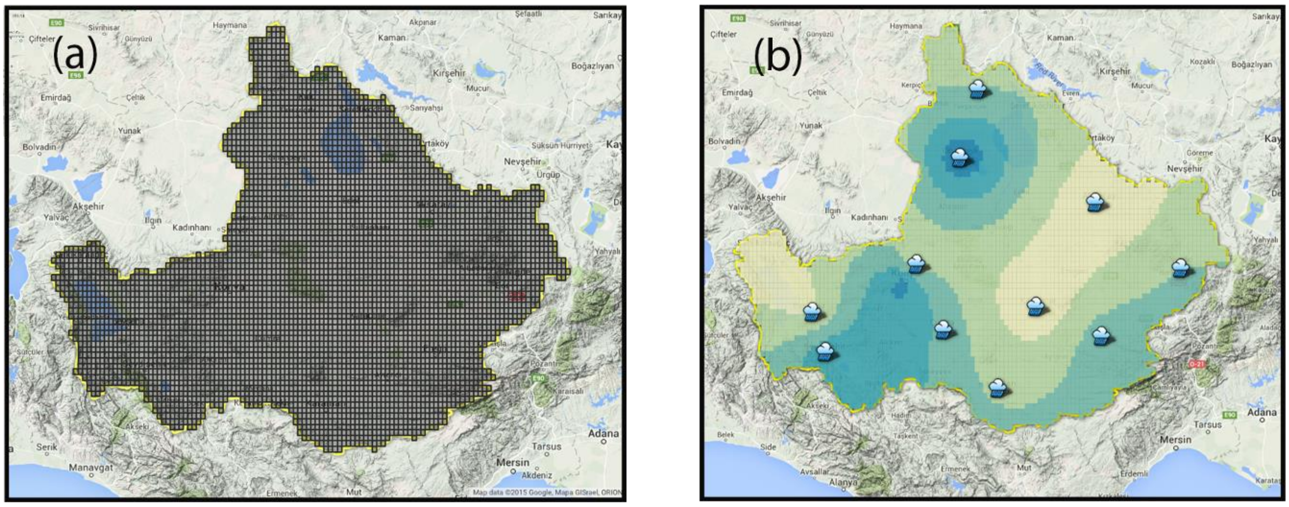

Figure 4. (a) Generated grid cells after point-in-polygon test (b) Generated isarithmic map layer.

The Ultimate File

There are a lot of web mapping APIs and supported data formats for adding external data as a layer varies depend on which mapping API is being used. The most common formats for geo-referenced data are GML, KML, GeoRSS and GeoJSON. Except GeoJSON, all of them are XML based markup languages. Almost all of mapping API's support KML and GeoJSON format. Because KML became an international standard of the Open Geospatial Consortium (OGC) in 2008 (OGC, 2008) and both KML and GeoJSON are supported by numerous mapping and GIS software packages, including OpenLayers, Leaflet, MapServer, Geoforge software, GeoServer, GeoDjango, GDAL, Safe Software FME, and CartoDB. It is also possible to use GeoJSON with PostGIS and Mapnik, both of which handle the format via the GDAL OGR conversion library. Bing Maps, Here Maps and Google Maps also support KML and GeoJSON in their API services. A notable offspring of GeoJSON is TopoJSON, an extension of GeoJSON that encodes geospatial topology and that typically provides smaller file sizes. Rather than representing geometries discretely, geometries in TopoJSON files are stitched together from shared line segments called arcs. Arcs are sequences of points, while line strings and polygons are defined as sequences of arcs. Each arc is defined only once, but can be referenced several times by different shapes, thus reducing redundancy and decreasing the file size. In addition, TopoJSON facilitates applications that use topology, such as topology-preserving shape simplification, automatic map coloring, and cartograms.

The developed algorithm in our approach can export generated cells with interpolated values as KML, GeoJSON and TopoJSON. Either using KML or GeoJSON, cell size is directly effects the size of file. Big data size takes longer loading times Table 1 shows cell numbers and size of files depend on different selected cell sizes for GeoJSON format and Table 2 shows file sizes of different cell sizes and different file formats.

Table 1. Cell numbers and size of files depend on different selected cell sizes.

Cell size	Cell number	File size	File format
30” x 30”	73576 cells	22.9MB	GeoJSON
1’ x 1’	18398 cells	5.7MB	GeoJSON
1.5’ x 1.5’	8167 cells	2.5MB	GeoJSON
2’ x 2’	4594 cells	1.4MB	GeoJSON

Table 2. Cell numbers and size of files depend on different selected cell sizes.

Cell size	KML	GeoJSON	TopoJSON
30”x30”	45.3MB	22.9MB	7.7MB
1’x1’	11.3MB	5.7MB	1.9MB
1.5’x1.5’	5MB	2.5MB	1.1MB
2’x2’	2.8MB	1.4MB	823Kb

CONCLUSIONS

This study summarizes a practical approach to create an isarithmic map layer that can be used with online web mapping services. Also there are some basic solutions to create custom thematic maps with web mapping services – like choropleth maps – there is no clear solution to create isarithmic map layers and visualize them on an online map. To find a solution to this problem, we focused on most common supported file formats by web mapping services and generating an isarithmic map layer with this format. Our approach first divides the area of interest into grid cells then eliminates the unnecessary cells and assigns a value to each cell derived from sample points. Final step is to export the results as an appropriate file format. In this research TopoJSON gives the best performance for our case and generates a grid file consist of 2’x2’ cells with a size of less than 1MB and 30”x30” cells with a size of 7.7MB. Once generating the output file, users can use this file with any mapping API that supports these types of files. But note that the output file does not contain any styling information. It includes cells and their derived values. As a future work we are planning to create a user interface of our algorithm.

REFERENCES

Berners-Lee, T., 2016. Frequently Asked Questions-Start of the Web. https://www.w3.org/People/Berners-Lee/FAQ.html [26.04.2016]

Cromley, R., 1989. Digital cartography. Prentice-Hall.

Hardy, R.L., 1971. Multiquadric equations of topography and other irregular surfaces. Journal of geophysical research, 76(8), pp.1905-1915.

Hardy, R.L., 1990. Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968–1988. Computers & Mathematics with Applications, 19(8), pp.163-208.

ICA, 2003. A Strategic Plan for the International Cartographic Association. http://icaci.org/files/documents/reference_docs/ICA_Strategic_Plan_2003-2011.pdf [26.04.2016]

Kraak, M.J. and Ormeling, F., 2011. Cartography: Visualization of Spatial Data. Guilford Press.

Open Geospatial Consortium, 2008. OGC® Approves KML as Open Standard. http://www.opengeospatial.org/pressroom/pressreleases/857

Peterson, M. P., 2012. Online mapping with APIs, In: Online Maps with APIs and WebServices, Eds: Springer, p. 3-12.

Putz, S., 1994. Interactive information services using World-Wide Web hypertext, Computer Networks and ISDN Systems, 27 (2), 273-280.

Shimrat, M., 1962. Algorithm 112: position of point relative to polygon. Communications of the ACM, 5(8), p.434.

Wikipedia Contrubitors, 2016. The definition of Application Programming Interface. https://en.wikipedia.org/wiki/Application_programming_interface [26.04.2016]

BIOGRAPHY

I. Oztug Bildirici was born in 1965 in Konya/Turkey and graduated from Selcuk University/Konya in 1986 as a Surveying Engineer. He trained Cartography at Polytechnic University of Berlin from 1991 to 1992. Having completed his MSc-Study at Selcuk University in 1994, he started his PhD at Istanbul Technical University. He worked as a Research Assistant at Selcuk University from 1990 to 1995. He worked in the same position at ITU from 1995 to 2000. In 1997, he was awarded a German DAAD Scholarship. He did research about generalization of buildings and roads at the Institute of Cartography and Geoinformatics (ICG) of the Hanover University under the supervision of Prof.Dr. D. Grünreich. He completed his PhD in 2000 at ITU. He was awarded a DAAD scholarship for 3 months in 2003, again. He did research about multiple representation databases at ICG of Hanover University. He has been working at Selcuk University as a professor on cartography since 2001. He is the head of the geomatics department and the cartography division.

Osman Kirtiloglu received the B.S. and M.Sc. degrees in Geomatics from Selcuk University, Konya, Turkey, in 2006 and 2009, respectively and currently working toward the Ph.D. degree with Selcuk University, Geomatics Department . He is a Research Scientist with the Department of Geomatics, Division of Cartography, Selcuk University, Konya, Turkey. His research interests include Web-Cartography, information visualization and visual analytics.