PLOTGOOGLEMAPS: THE R-BASED WEB-MAPPING TOOL FOR THEMATIC SPATIAL DATA

Milan Kilibarda and Branislav Bajat
University of Belgrade, Faculty of Civil Engineering
Department of Geodesy and Geoinformatics, Belgrade, Serbia

Google Maps are increasingly used for communication throughout many map-based services and maps, embedded on third-party websites via the Google Maps API. The main objective of this study is to present a solution for an easy creation of an interactive web map, with a base map supplied by Google, where all map elements and additional functionalities are handled by just one line of code. The present solution for the automatic creation of a complete web map, based on the Google Maps API, is the R package plotGoogleMaps. This tool provides a new interactive plot device for handling the geographic data for web browsers. It also offers a complete map in the HTML format, which has become a regular medium for cartographic communication. The tool plotGoogleMaps is developed in the R software language and it is designed for the automatic creation of web maps, as a combination of users' data and Google Maps layers.

Milan Kilibarda
Kili@grf.bg.ac.rs

Branislav Bajat
bajat@grf.bg.ac.rs

1. Introduction

Although the Internet has been in existence, at least in some form, since the late 60s, only from the mid-90s, with the widespread use of the World Wide Web (Web), has the Internet become a foremost medium for cartographers [*Peterson* 2007]. The real breaking point in the usage of geographic information on the Web was in the year 2005. In June 2005, Google released the Google Maps Application Programming Interface (API), which allows a combination of geographic information from a variety of sources and formats. One of the most important capabilities of the API is the generation of mashup maps—the product of the combination of geographic data from one source with a map from another source [*Miller* 2006; *Gartner* 2007; *Haklay et al*. 2008]. The mashup maps are easy to create and implement in any web page, with no cost and without any technical specification and requirements whatsoever, resulting in an increased web mapping popularity. This progress, together with the popularity of web mapping, and its application, is clarified by Haklay [*Haklay et al.* 2008]:

> These rapid developments in web mapping and geographic information use are enabled and facilitated by global trends in the way that individuals and communities use the Internet and new technologies to create, develop, share and use information (including geographic information), through innovative, often collaborative, applications.

This change of direction of the Web philosophy, from communication media to contribution media, is named Web 2.0. The term and the concept, Web 2.0, was first coined and described by *Tim O'Reilly* [2005].

The collaborative nature of the Web 2.0 environment allows data production to be shared among many individuals [*Feick and Deparday*. 2010]. *Goodchild* [2007] described the term 'Web Mapping 2.0' as an important part of the Web 2.0 concept. Integration and visualization of different geographic information on base map (such as Google Maps/Earth, Virtual Earth, or Yahoo Map), is the core of 'Web Mapping 2.0'. The most significant part of Web Mapping 2.0 corresponds to Google Maps/Earth services. Google Maps/Earth is a ground-breaking software, in at least five categories: availability of application, high quality background maps, single coordinate system, web-based data sharing, popular interface and availability of API services [*Hengl* 2009].

Google Maps API has encouraged a considerable number of users, with intermediate and advanced programming knowledge, to build their own applications, using Google Maps data as the visualization interface [*Gibin et al*. 2008]. According to *Built With Trends* statistics [*Built with Web Technology Usage Statistics* 2011], the usage of Google Maps websites was over 1 billion, comprising mainly thematic cartography.

A thematic map displays the spatial pattern of social or physical phenomena, such as population density, life expectancy, or climate change. Thematic mapping has a long history in geography, and one important part of presenting thematic data is the provision of high quality base maps, which allows integration into the Google Map interface, through Google Maps API. The London Profiler [*Gibin et al*. 2008] presents geographic information as a series of choropleth maps on top of Google Maps. This example is an exception from most mashups, because they mostly display spatial point data (push pins).

The objective of this work is to provide a solution for the easy creation of an interactive web map, with a base map supplied by Google, where all map elements and additional functionalities are handled by just one line of code. The solution for the automatic creation of a complete web map based on the Google Maps API (the HTML file with CSS styling and Java Script functionality), is the R package *plotGoogleMaps* [*Kilibarda* 2010]. The package provides a solution to create and visualize vector and raster data and map features, choropleth maps, and proportional symbols. The growing popularity of the R language, particularly among academic and expert communities, and especially in the field of spatial data analyses, in addition to restrictions of existing solutions, was the driving force behind the development of the *plotGoogleMaps* tool. The goal of the presented work is to adopt and apply map design principles, in order to create mashups, and to focus on the minimization of coding and scripting. This would enable the creation of mashups based on Google Maps, without any Internet programming knowledge. Thus, the creation of web maps becomes a plot facility for R users. The web maps created by *plotGoogleMaps* package could be used as a temporary result of spatial visualization, generated on the local machine, or published on any web page. The next section contains a brief description of theoretical and technical background for the development of the *plotGoogleMaps* package, Web 2.0 and AJAX, Google Maps API, and R software environment. In addition to this section, there is a description of package functionalities, and of how these were programmed, with details concerning the source code, and instructions on how to get this package. The paper continues with examples of package applications, while the last section concludes the study.

2. Methods

2.1 Web 2.0 and AJAX

Best [2006] defined main characteristics of Web 2.0 in a few items, which are: a rich user experience, user participation, dynamic content, metadata, web standards and scalability. Examples of Web 2.0 applications embrace social networking sites, video sharing, wikis, blogs, etc. The Web 2.0 concept is based on the AJAX technology.

Asynchronous JavaScript and XML, or shorter, AJAX [*Garret* 2005; *Asleson* 2005] is the name given to a set of modern web application development technologies, previously known as dynamic HTML (DHTML) and remote scripting. The fact that this web application has a similar speed to the standard desktop application makes up the core of this technology. Traditional web pages, created in HTML, were static, nonflexible, and hardly adopted for a dynamic content. AJAX, on the other hand, as the base of Web 2.0 concept, was built for the dynamic, interactive, and efficient web pages, with high performance. Web pages without AJAX were slow; user interaction with the website required significant web server-side resources, so that if just one part of the web page was changed, the server needed to send a complete web page with the changed part back to the user. If the user drew a point on the web map, the server would send back the whole redrawn map with a new point. AJAX brings a new concept, so that the user interaction is left to a user's computer, which works only with a changeable part of the web page. Therefore, even if the user operates with one point only, it would not be necessary to redraw the whole map again. AJAX-based geographical applications significantly improve the usability of web mapping [*Skarlatidou and Haklay* 2006; *Haklay and Zafiri* 2007; *Haklay et al.* 2008; *Kilibarda et al.* 2010]. Apart from AJAX, APIs have also influenced web mapping strongly. API is a set of routines, protocols, and tools for building software applications. The most popular web mapping APIs are: Google Maps API, Yahoo! Maps API, Microsoft Virtual Earth API, and AOL MapQuest API [*Gartner* 2009].

2.2 Google Maps API

Google Maps API is a set of predefined JavaScript classes, designed for embedding the Google Maps site into an external website so that additional geographical data could be overlaid over a basic Google Map. This is possible to realize, even if the creator is not an expert in web programming, although basic knowledge in JavaScript programming language, XML, Ajax and XHTML is required. Google Maps API, compared to the relative complexity of Open Geospatial Consortium (OGC) standards, is much easier for implementation. Google Maps API provides mapping functionality and high-resolution background data, although map mashups implementation still requires some web programming knowledge. It enables a combination of geographic information from a variety of sources and formats. GIS data objects, such as vectors, points, polylines, polygons or raster, are represented in the mashups as Google Maps API JavaScript objects. For that reason, it is necessary to transform the GIS data into JavaScript objects, appropriate to Google Maps API. These objects could be cartographically represented, though with some limitation, especially with point symbols. Polylines and polygon's line representations could be defined with outline width, colour, transparency and fill colour, or transparency, for the polygon area. Transforming the GIS data into Google Maps API objects, and defining a cartographic representation for every single object, is time consuming, and rather difficult, especially for someone who has no experience in web programming. The developed R package, *plotGoogleMaps*, offers an easy creation of mashups as local files, or files ready to be published on the Web.

2.3 R environment

As stated in the *Introduction of the R Language Definition* on-line manual [*R-project* 2011], R is a system for statistical computation and graphics, which provides, among other things, programming facilities, high-level graphics, interfaces to other languages, and debugging facilities. R implements a language similar to the S language, originally developed by John Chambers [*Becker and Chambers* 1984]. The main difference is in the license statement, because in contrast to the S language, R is free and open source software under the terms of the GNU General Public License. The syntax of the R language is analogous to the C programming language. However, a fully functional interpreter permits the creation of functions and calculations within an environment defined by a command line window, or graphical user interface [*Grunsky* 2002]. R is organized as a collection of packages designated for specific tasks. There is also a set of developed packages, which could be especially interesting for geoscientists.

The R package system has been one of the key factors in the overall success of the R project [*R Development Core Team* 2008]. R contains the base system which enables statistical computation, linear algebra computation, graphics creation, and similar. A package is a related set of functions, help, and data files that have been bundled up together. Packages in R are similar to modules in Perl [*Perl* 2011], libraries in C/C++ [*C programming* 2011], and classes in Java [*Oracle* 2011]. It is not necessary to install the specific packages, if it is not a part of the user's computing and analyzing interests.

The existing solution of using a Google Maps image, as a background for plotting spatial data, is the *RgoogleMaps* R package, based on Google Static Maps API. The Google Static Maps API [*Google code* 2011] allows the embedding of a Google Maps image in the user's webpage, without requiring the JavaScript, or any dynamic page loading. This package provides static maps, without the interaction with attribute data pan or zoom control, and with a constrained quality of the Google background map, offered by Google Static Maps API. The maximum zoom level, provided by Google Static Maps API, concurs with the max size limitation of 640 x 640 pixels.

The other package with similar functionalities, which provides an interface between the R and the Google Visualisation API, is called the *googleVis* [*Gesmann and de Castillo* 2011]. The Google Visualisation API offers interactive charts that can be embedded into web pages. The *googleVis* package contains options to produce maps mashups based on Google Maps API. The input data for the package is the data frame with marked columns, related to location information. This is a typical package used for handling spatial data and their visualization.

3. The Developed Software Package—plotGoogleMaps

The newly developed R package, *plotGoogleMaps*, based on AJAX and Google Maps API service, produces HTML file map mashups (web maps), with Google Map high-resolution background data, and additional data layers.

The package contains two components for spatial data handling, *sp* and *rgdal* R packages. These sets of developed packages are especially interesting for geoscientists. The R developers have written the R package *sp* to extend R with classes and methods for spatial data [*Pebesma and Bivand* 2005]. Classes specify a structure and define how spatial data are organized and stored. Methods are instances of functions, specialized for a particular data class [*Bivand at al* 2008]. The package *rgdal* provides functionalities for the import and the export of the most popular formats of GIS data in R. This package uses functions of the Geospatial Data Abstraction Library to read and write the GIS data, with options for handling a coordinate referent system (CRS). With *rgdal* package, users can optionally define, or own a CRS, or inherit it from the input data, as well as perform data transformations between different CRS's, by using PROJ4 library [*OSGeo* 2011]. The imported GIS data are converted to *sp* objects, used for handling the vector and raster data in R [*Pebesma and Bivand* 2005]. These functionalities enable the use of a very large amount of the GIS data formats, as input for the *plotGoogleMaps*. Input data, *sp* objects with defined CRS, is the only mandatory argument in the *plotGoogleMaps* functions. Hence, different GIS formats of input data are read in R, and afterwards based on a predefined visualization method; those data are mapped as web map (Figure 1).

The *plotGoogleMaps* contains functionalities from PROJ4 library, which performs coordinate transformations from source CRS to WGS84 CRS, used for spatial data handling by Google Maps. Google Maps API allows for additional spatial data handling, in the form of XML, KML, and GeoRSS, but some visualization functionalities, as well as interaction with attribute data in the form of Google Maps API *InfoWindow* object and similar, are difficult to be controlled. Another solution is to use the data in the form of predefined JavaScript classes of vector data primitives; point, line and polygon data and raster overlay. This approach is also implemented in *plotGoogleMaps*. It means that every single primitive is separated from the spatial object, and its geometry is translated into a JavaScript object. Attribute data for every single feature is converted into a JavaScript *InfoWindow* object; its activation is available by clicking on the related feature on the produced web map. Additional visualization options supported by Google Maps API objects, such as outline width, colour, and transparency, can be specified in *plotGoogleMaps* functions. Visualization of mandatory parameters is easy to set, by using optional arguments in *plotGoogleMaps* functions. Therefore, *plotGoogleMaps* writes object arguments in every JavaScript object in the final HTML file. Google Maps API provides the majority of Google Maps utilities, such as pan, zoom, background layer control, and scale bar. Map utilities are controlled by optional arguments in *plotGoogleMaps* functions. Similarly, map width, background colour, layer name, legend name, default background map, etc., can be set by using optional arguments in the *plotGoogleMaps* function. Some advanced utilities for interactive controls of additional layers are pro-

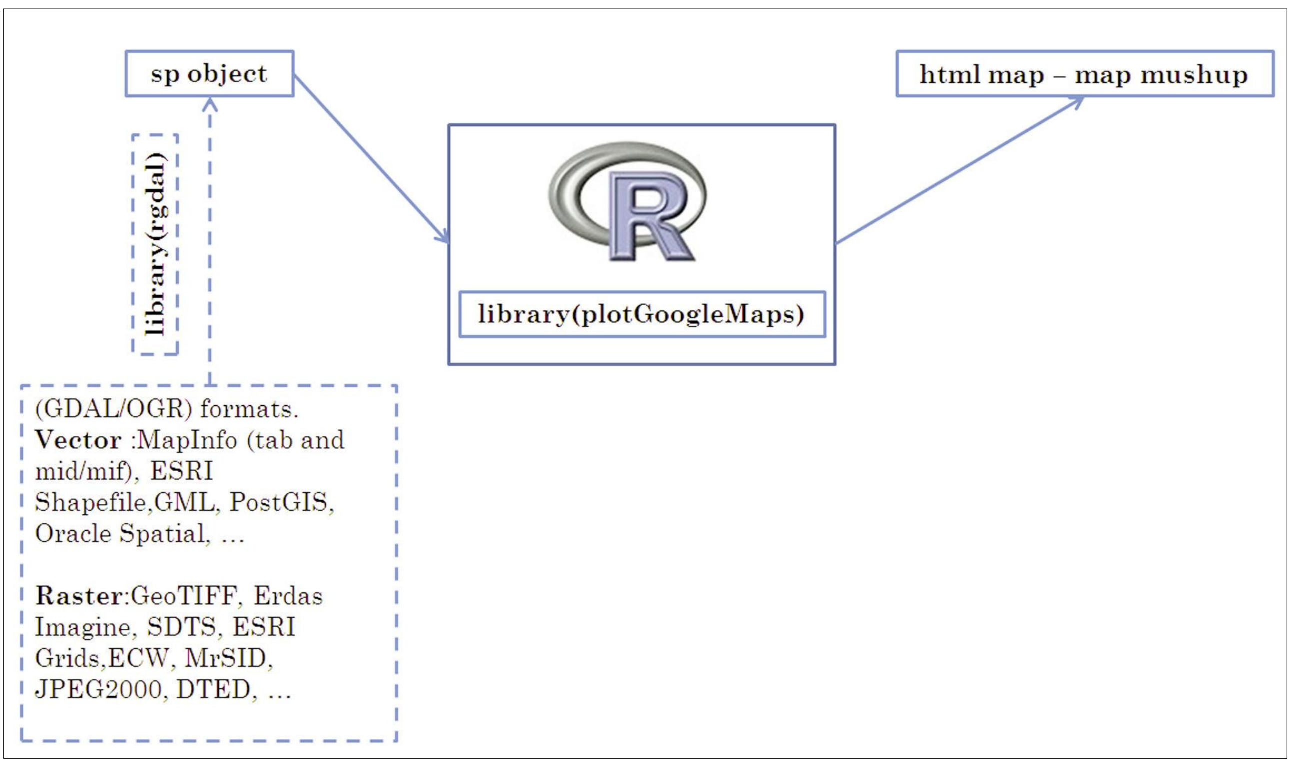

Figure 1: Simplified workflow for web map production by using *plotGoogleMaps*.

vided by default in *plotGoogleMaps*: layer appearance, line width, transparency, and legend colours display control. Spatial data, with visualization parameters and utilities, are written in the HTML file with JavaScript and CSS elements. Thus, in some languages, *plotGoogleMaps* with only one function and with few arguments, may produce many lines of codes (Figure 2- right bottom window).

The map mushup (Figure 2- right bottom window) is produced by the R command:

> plotGoogleMaps(meuse,filename='myMap.htm')

The command contains two arguments: the first is the set of spatial data, named *Meuse*, containing 155 points, with 12 soil properties in the form of attributes, while the second is an optional argument, a file name of the output map mashup. This function contains many optional arguments. For example, the description of the function arguments may be provided by using the "help" function in R, or found on the package web page [*Kilibarda* 2010], in the reference manual. The same web page contains the package source; the source code is open and available under the GPL license. Technical details about the used solutions in the package could be obtained from the source code.

Generally, the idea of package implementation is denoted as the automatic production of HTML map mashups, by using spatial data objects. The R command used in the previous example produces an HTML file from the Meuse data. The output map also contains CSS elements, and JavaScript scripts. The control of CSS elements is available by using optional function arguments, in order to set the dimensions of a map area. Since the optional arguments are not set in this example, then the HTML file contains default CSS styling settings, with the map canvas area covering 80% of the Web page, while the remaining 20% is reserved for the layer control options. The rest of the produced HTML is JavaScript that contains Google Maps API and a set of JavaScript functionalities related to layer control options. The Meuse data was transformed from native CRS to WGS 84, and every single point was

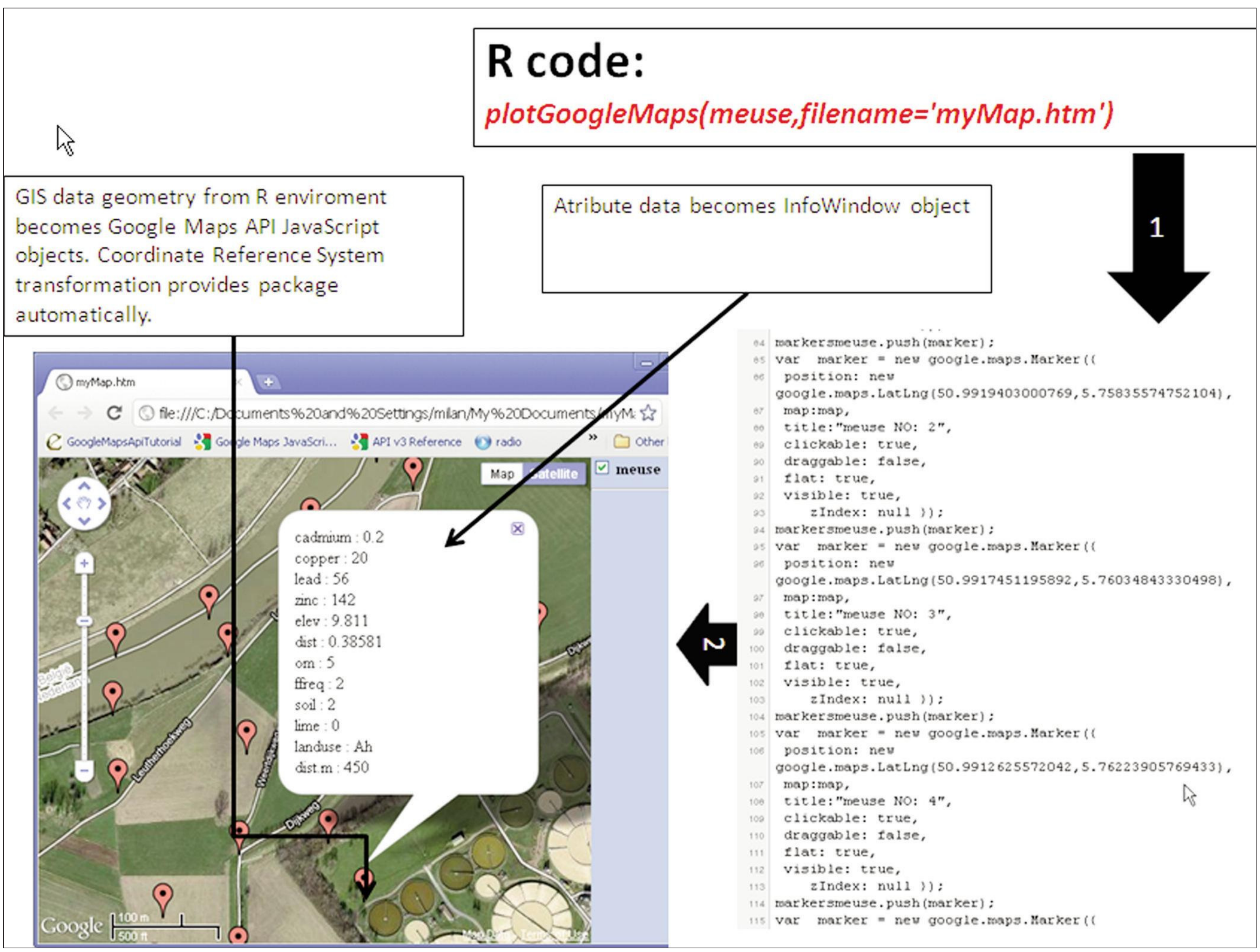

Figure 2: Plotting vector point data. Just one line of R code substitutes many lines of HTML with JavaScript and CSS code.

translated to Google Maps Marker object (i.e. JavaScript object), used in Google Maps API. The base map is set, by default, to be a Google hybrid map, where the initial zoom and central points of the base map depend on the Meuse bounding box. The attribute data for every single point is converted separately to *google.maps.InfoWindow* objects, with an associated *listener* function to handle the click event on the marker.

The next section contains six examples focusing on different applications of the plotGoogleMaps package, while the complete code with all data and the R script is available from the coauthor's website[1].

4. Implementation and Applications

The functionalities of the package, used for the production of web maps of thematic and spatial data, are presented through the following case studies. The applications illustrated in this paper pertain to different studies concerning spatial data analysis. Some of them are related to the visualization of pedologic and geologic soil survey data, and subsequent results obtained through their analysis. One case study is related to the mapping census data of Serbia. The applications of *plotGoogleMaps* in engineering fields are depicted by flood risk hazard mapping, and geodetic control network design mapping. The majority of these case studies are located in the Republic of Serbia.

4.1 Visualization of Meuse Data Set

The first case study represents Meuse, the Netherlands data set, which is a training data set used frequently in a lot of scientific papers and researches related to geostatistical spatial data analysis. This data set, collected in the flood plain of the river Meuse, in the area around Meers and Maasband (Limburg, the Netherlands) (N=50°58′16″ E=05°44′39″), during a fieldwork in the year 1990. It contains locations and topsoil heavy metal concentrations of the 155 observed locations, together with soil properties and distances to the river. The Meuse data set is exhaustively depicted by *Rikken and Van Rijn* [1993], and *Burrough and McDonell* [1998].

The spatial prediction of zinc, based on sampled values from the Meuse data set, was conducted using the ordinary kriging method [*Isaaks and Srivastava* 1989]. The results were obtained by using *gstat* R package [*Pebesma* 2004]. The conceptual data model of zinc variation in the soil is illustrated as a field [*Burrough and McDonell* 1998] and the grid repesenting concentration of zinc in the soil is given in standard R graphic interface (Figure 3. left). The same result is visualized by *plotGoogleMaps* command, where the obtained grid is overlaid on Google satellite base map. The obtained result provides a clear cartographic representation of the modelled spatial variable, in relation to other spatial content. The *PlotGoogleMaps* package provides other additional functionalities related to grid representations, such as transparency and map legend with hypsometric scale (Figure 3. right).

The map mushup (Figure 2- right) is produced by the R commands:

```
pal<-colorRampPalette(c("green","orange","brown"),space="Lab")
m411<-plotGoogleMaps (meuse_zinc, zcol=1, colPalette=pal(20))
```

The first command, *colorRampPalette*, represents the functions that interpolate a set of given colours, to create a new colour palette of the output map.

The grid with two attributes, the prediction of zinc concentration and variance in every grid cell, is set as the mandatory argument in the *plotGoogleMaps* function. The R *sp* classes provide the possibility for a single grid/raster to comprise several different attributes. The second optional argument denotes which attribute from the spatial object is being visualized; predicted concentration, in this case, will be the first grid argument. Finally, the last argument defines colours for grid representations, the colour palette containing 20 colours, from green to orange/brown, which automatically means that the plotGoogleMaps categorizes grid data in 20 intervals and uses colours like hypsometric tints. The output map mashup is stored in the R working directory.

4.2 Census Data of Serbia

The interactive web map of Serbian census population data changed from the year 1991 to 2002 [*Statistical Office of the Republic of Serbia* 2003], where the blue colour represents population loss, whereas the brown the growth in population, created only by the *plotGoogleMaps* function (Figure 4). The additional interactive control utilities: layer appearance, transparency line width, and legend colours display, are on the right side of the browser window. The attribute data become apparent in a list box, after clicking on the feature of interest. The colour coding system for map design in R is supported by *RColorBrewer* package. It provides palettes for drawing appealing maps, shaded

[1] *http://www.grf.bg.ac.rs/~bajat/geomatica_script_and_data.zip*

according to a variable. This package was derived from ColorBrewer website [*Brewer et al*. 2003].

4.3 Soil Survey in East Serbia

Quantitative point symbols, such as proportional symbols of varying sizes, used to symbolize totals at a point, are available in the *plotGoogleMaps*. The most frequently used shapes are circles, but squares and triangles are also possible solutions offered by the *plotGoogleMaps* function. Multivariate mapping depicts more than one variable in the same view, related in some way. The package can produce a single map, which combines choropleth representation and proportional symbols of two related variables. Figure 5 shows the results of a detailed survey carried out in order to produce a soil map intended for detailed design of drainage and irrigation networks, as well as for

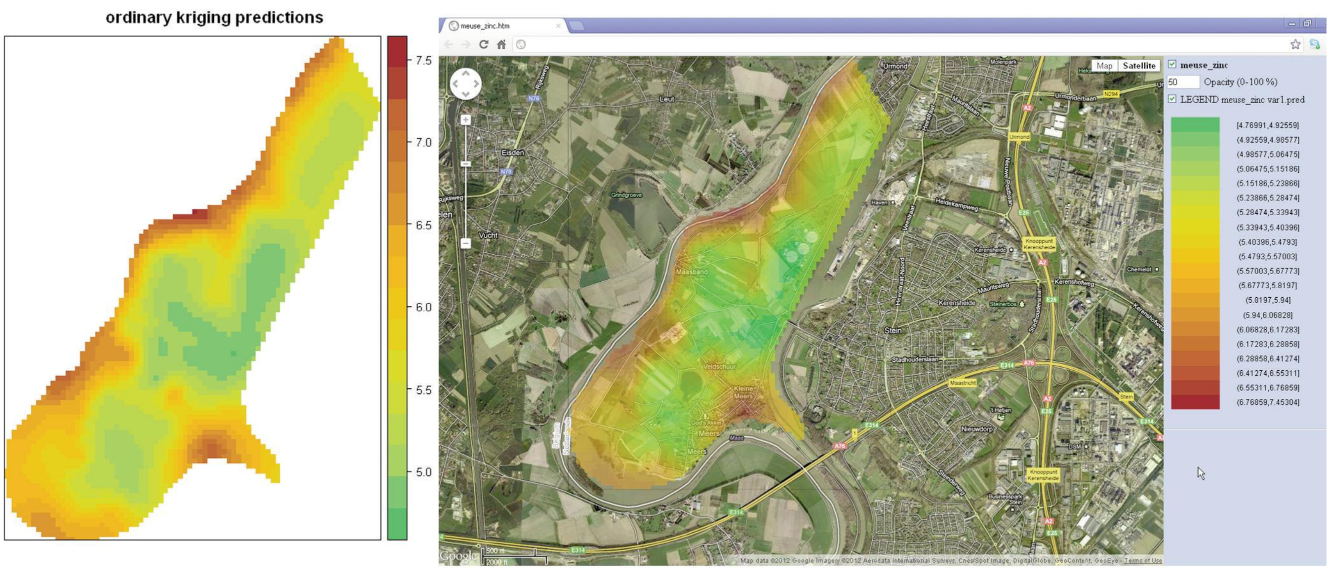

Figure 3: The same result plotted in R graphic interface (left) and by *plotGoogleMaps* (right).

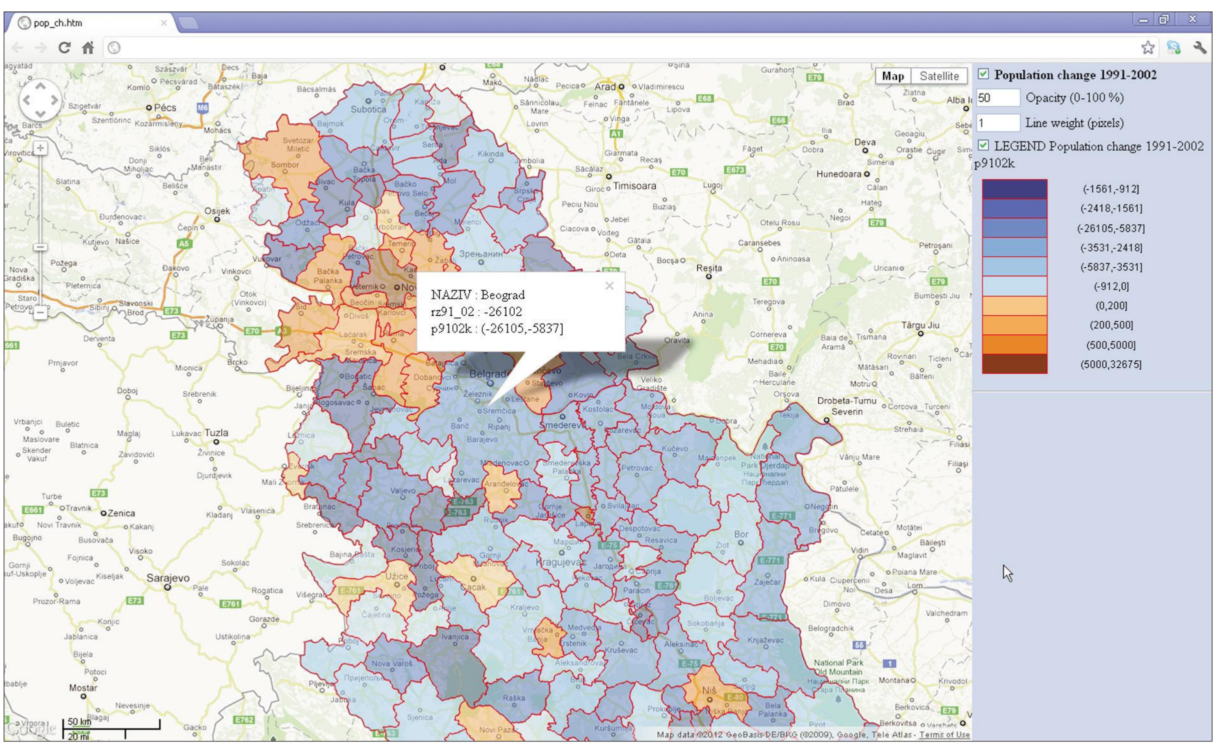

Figure 4: Choropleth web map showing changes in population in Serbia from 1991 to 2002 produced by *plotGoogleMaps* package.

establishing new vineyards in the Negotin municipality, in eastern Serbia close to the Romanian and Bulgarian borders (N 44°12′, E 22°28′). The total area covered by the survey approximates 1272 ha, with altitudes between 106 m and 290 m above the sea level, meaning that, morphologically, it belongs to a moderate hilly area. The land use predominantly consists of croplands and grasslands (meadows), subdivided into numerous individual lots. The mapping of soil units was based on field and laboratory observations of chemical, physical, biological, and mineralogical properties of horizons, geological properties of parent rock material, and geomorphological features of landscape. Delineated soil types polygons are classified according to the World Reference Base (WRB) for Soil Resources [*FAO* 2006]. In Figure 3, soil type polygons, depicted with different colours, are overlaid with proportional symbols presenting the concentration of humus (soil organic matter) within each polygon (Figure 5). Layers might be active at the same time, or just one may be on.

4.4 Geological Survey in the Municipality of New Belgrade

Cartograms, like pie chart maps, are not yet available in the package *plotGoogleMaps*. Figure 6 shows one part of the results obtained by comprehensive geological survey in New Belgrade's (N=44°48′, E=20°24′) municipality area (territory about eight square kilometers), in which the zones are marked as alluvial sediments in the fundamental geological map of Belgrade, the capital of Serbia. Afterwards, the analysis of the values of geomechanical properties, and the statistical elaboration of all parameters based on 150 to 300 samples (grain size, effective particle size, bulk density, water content, plasticity limits, porosity, void ratio, saturated density, permeability value, shear strength parameters, compressibility of soil, penetration resistance, etc.) was done [*Radić* 2007]. Silty-sandy clay covers almost the whole investigated surface and subsurface areas of New Belgrade, with average thickness between 2.5 and 5.0 m in depth, and in some instances up to 7.0 m. The function *segmentGoogleMaps* can produce a map like the one presented in Figure 6.

Presenting the results in this manner has multiple advantages over the conventional use of the existing geotechnical documentation, at various stages of planning and construction design. Instead of scrolling through extensive existing text documents, an expert's job comes down to the unified and visually presented results of the conducted measurements. Based on the visual presentation of quantitative and qualitative data, it is now easier to make evaluations of the general characteristics of a terrain, for the planning, designing, and constructing purposes.

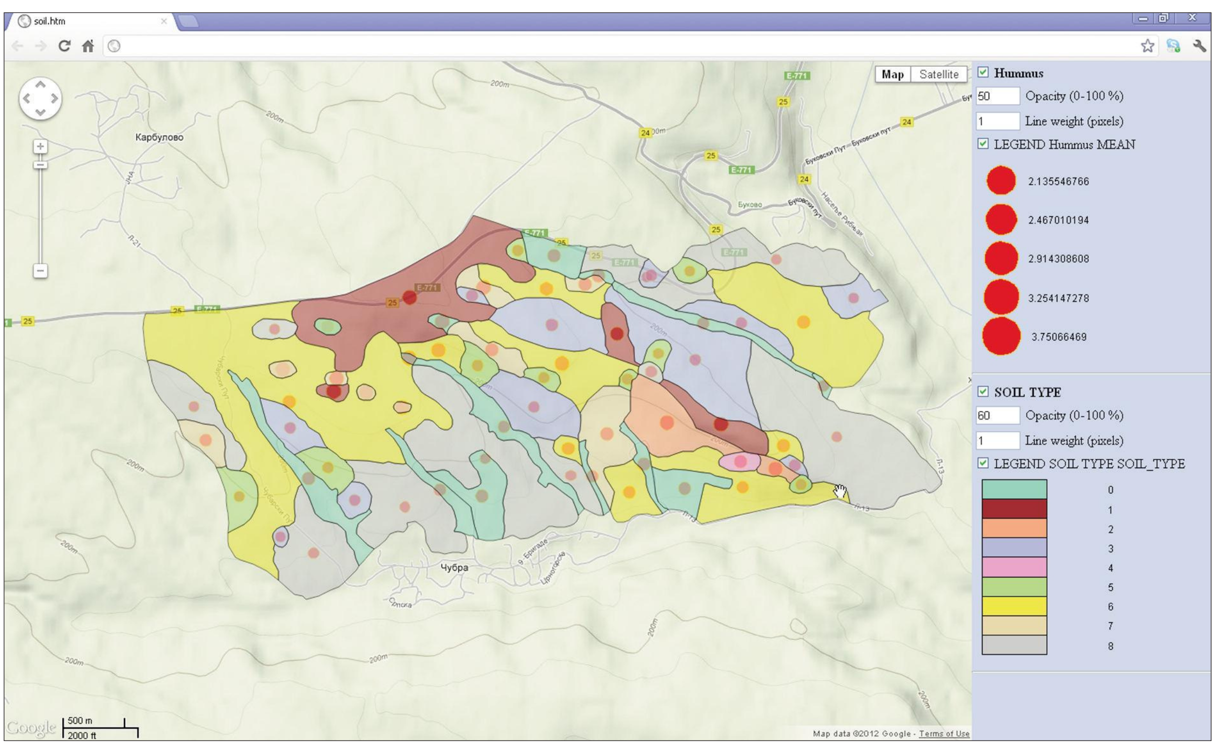

Figure 5: Proportional symbols of humus concentration overplayed across polygon layer representing soil types.

It is also easy to evaluate the number of necessary surveys and proper selection of samples, for further geological exploration of an area. Moreover, new data collected from field and laboratory tests can be easily added to the existing measurements and thus be easy to update.

4.5 Flood Mapping in West Serbia

In addition to mapping the results obtained by various spatial analyses, *plotGoogleMaps* can be used as a tool for gathering the necessary information for the subsequent spatial data processing. For the purposes of the assessment of population vulnerability to a one-hundred-year flood in the valley of the Lim River in Prijepolje, western Serbia (N=43°23′ E=19°38′), *plotGoogleMaps* was used to identify the population density classes, necessary for the dasymetric population distribution modelling [*Mennis* 2003; *Krunić et al.* 2011].

The polygon representing the one-hundred-year flood was generated using the HEC-RAS software [*Dyhouse et al*. 2003; HEC 2008], for the 1D hydraulic modelling. The HEC-RAS model is the successor of HEC-2 program, that has been used for the last 30 years in the USA and in the whole world, as one of the standard programs for hydraulic calculations. The Input Digital Terrain Model was provided as the sub-product throughout CARDS PROJECT "Digital Orthophoto Production in the Republic of Serbia" Project, [*RGZ* 2011] with 5 m grid size and an average height accuracy of ± 40 cm for urban areas. The enhanced accuracy (± 10 cm) of DTM, covering the DTM city zone of Prijepolje, was obtained directly from a photogrammetric stereo model. The obtained DTM accuracy completely fulfills proposed standards for flood risk mapping [*National Research Council* 2009]. Supplementary river bed cross-section geometric data were provided throughout the bathymetric survey. In order to recognize a hazard prone area, flood polygon was mapped over the polygons of census designated places. CDPs Zalug and Prijepolje, with the total of 1047 and 15031 inhabitants respectively, were identified as vulnerable areas. Through the application of plotGoogleMaps, the one-hundred-year flood polygon overlayed over land covered classes (CLC) [*CORINE* 2000], and indicating artificial areas (contінous and disconinous urban fabric, roads network, etc), was exported on the GoogleMaps (Figure 7). After the inspection of the covered area of the GoogleMaps satellite image, three different classes of population density (low, middle and high), as well as uninhabited polygons were identified in order to modify the inhabited polygons. The proportions of 5%, 15% and 80% were assigned, respectively to each class of inhabited polygons to calculate the average population density within each polygon. The percentage of participation is certainly determined on a case-by-case basis, depending on the analyst's arbi-

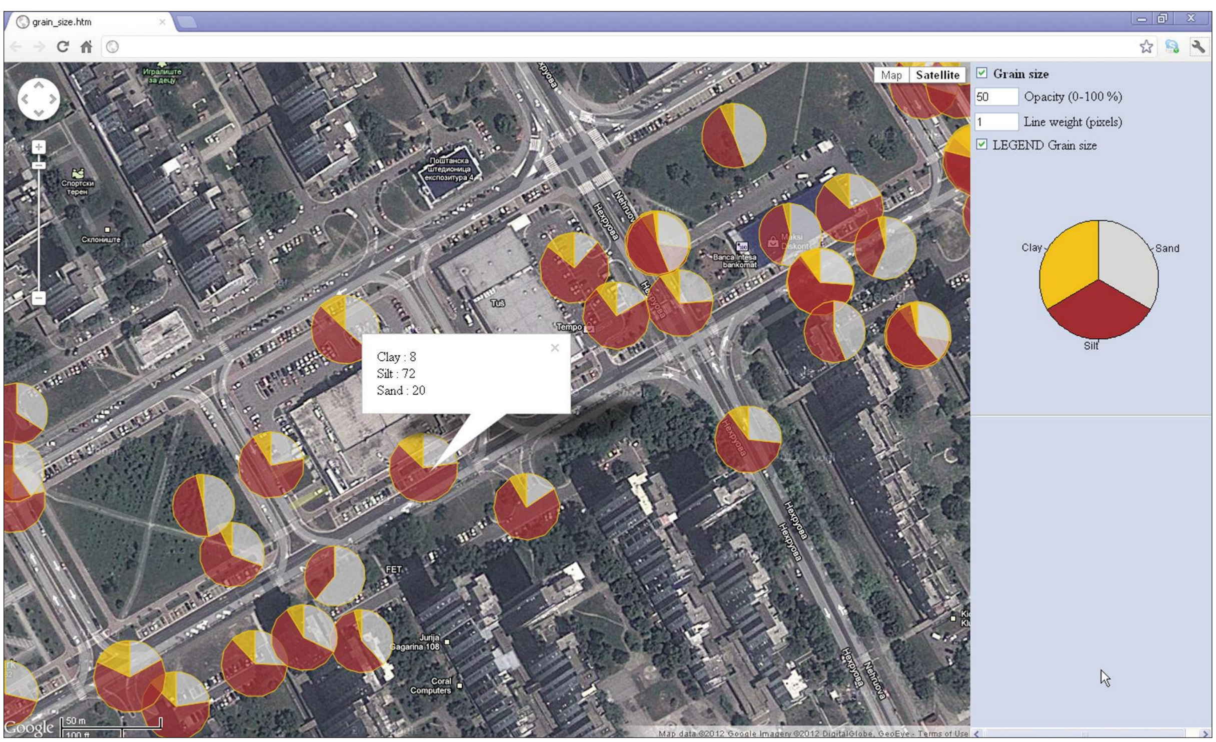

Figure 6: Google Map based on satellite imagery, with grain size diagrams (pie chart maps) of geological sampling, produced by *plotGoogleMaps*.

tration. The CLC polygons, modified in this way, were subsequently used for the disaggregation of the population census data by the Three-Class Dasymetric Method [*Eicher and Brewer* 2001], in order to generate reliable population distribution over the engaged area. The total count of the vulnerable population was estimated by simply overlaying the one-hundred-year flood polygon over the population density polygons (PDP), within the ArcGIS software environment. New vulnerable density population polygons obtained as the result of the intersection of initial PDPs and flood polygon were generated. The number of inhabitants within these polygons is a product of their population density and the covered area. In this case, the sum of all polygons produces 1441 people.

4.6 Geodetic Control Network Design Mapping

The implementation of *plotGoogleMaps* could be of particular inertest to the geodetic control network's design process. The geodetic control network design involves identification of appropriate network geometry (spatial arrangement of stations) and a plan of network observations (the number, arrangement, and the precision of observations). In the conventional geodetic network design, topographic maps or aerial photos are often used for the determination of the possible station positions.

The experimental site located in Belgrade, 9 km from the city center to the South, was chosen to show a possible application of the *plotGoogleMaps* package in the geodetic network design. The network design was simulated for the purpose of the drifting and building of the Straževica tunnel. This tunnel is 745 meters long, and represents a part of the projected Belgrade's outer beltway. The designed control network provides an outer geodetic framework for the tunnel digging and settling out. The latitude and longitude of the simulated network are approximately N=44°43.3′, E =20°27.0′.

The completed observational plan, together with all possible combinations of distance and direction observations, was exported to the HTML file by using *plotGoogleMaps*. Final observational plan is the input for the least squares adjustment process in the control network design (package presently being developed in R language environment by the authors of this study). A stochastic model, necessary for the creation of weighted matrix of observations, was built in R by defining the measurement equipment accuracy standards for the directions and distances.

R functionalities enable the transformation of the design input coordinates and measurements simulation, derived from Google Maps (WGS84 CRS) to the national CRS or any other predefined coordinate system (local system of building site). It is possible to transform geometry to the desired CRS specified by PROJ4 notation.

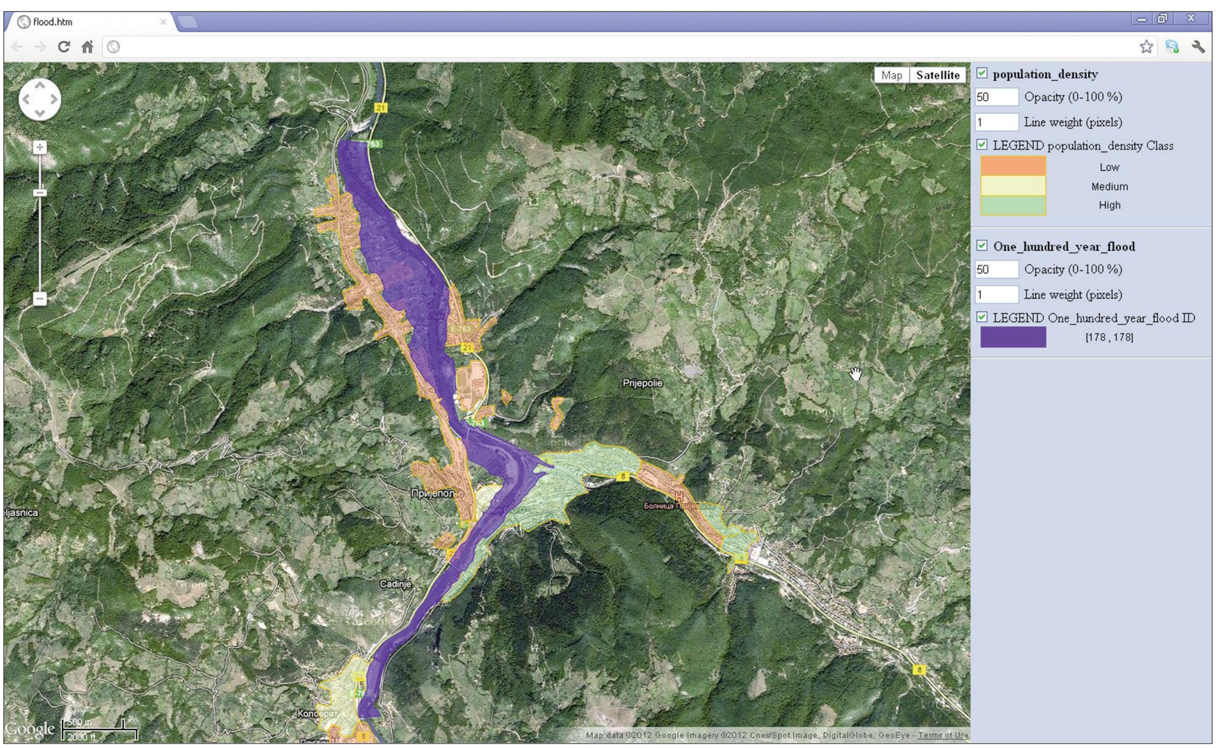

Figure 7: The one hundred year flood polygon intersected with polygons depicting three density population classes (HTML GoogleMaps view).

The least square adjustment computation [*Ghilani and Wolf* 2006] has been applied to the final observational plan with the equipment specification. Quality measures for the points and measurements have been derived from the adjustment computation:

- Coordinate standard deviation and error ellipse geometry parameters were computed for the unfixed points in the network;
- Internal reliability and marginally detected errors were computed for measurements.

Numerical quality assessment for the points has been incorporated into the point coordinates, and then converted to *SpatialPointsDataFrame*, *sp* class in R. This class contains four slots, i.e. data containers that fully determine the data in a spatial manner. *Project4string* slot defines CRS, bbox—boundary box, cords—all coordinates of points, cords.nrs—records the column positions in the data file, that the coordinates were taken from (as a precaution when a spatial class is derived from a data frame, since the data frame is the most popular data format in R). In this case, the points' attributes, the numerical quality assessment of the points, are stored in the remaining *SpatialPointsDataFrame* slot named data. *SpatialPointsDataFrame* could be exported to many GIS file formats, by using R package *rgdal*. Spatial class, *SpatialLinesDataFrame* and *SpatialPolygonsDataFrame*, also used in this project, have similar properties.

Observations with quality parameters are adapted to the *SpatialLinesDataFrame*. Error ellipses given in a desired scale, with other quality point measures, were created as attributes by a function developed in the R language [*Kilibarda and Pejović* 2011] as the *SpatialPolygonsDataFrame* class. Thus, all the results of the adjustment may be converted to many GIS formats in the desired CRS. Besides the visualization of the geometry, numerical results of the adjustment were stored in HTML. Numerical results are available by simple clicking on the points, ellipses and lines in *Google Maps*.

The geodetic network geometry with some additional functionality, like hyper-linking image of geodetic benchmark pillar, is depicted in Figure 8.

The package that is currently under development in the R language environment, which is going to be dedicated to the adjustment of geodetic control networks, will use *plotGoogleMaps* as integrated module for graphic outputs.

5. Discussion and Conclusions

The *plotGoogleMaps* is a free and open source software solution for the simple creation of rich interactive web maps. In this case, *plotGoogleMaps* uses the web browser as a plotting device instead of the default R graphic device. Therefore, it offers more advantages compared to classical R plotting device environment, i.e. high quality of background Google layers which make better abstractions of geographical reality, spatial data exploration functionality, and map interactivity (navigation control, pan, zoom, attribute info windows, etc).

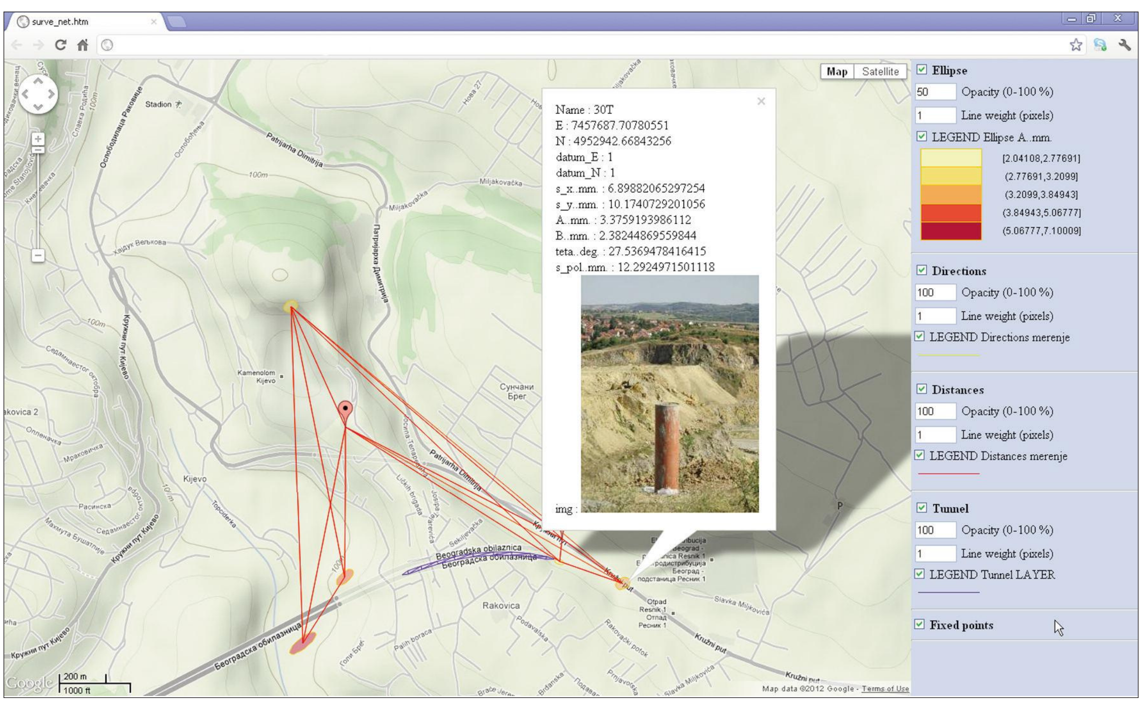

Figure 8: Results of the network adjustment in *Google Maps*, produced by *plotGoogleMaps* package.

This package promotes the creation of interactive maps in user friendly environments where a map is stored in the HTML format. Map sharing with non GIS map users is easier, it is not necessary to use the GIS software and only a web browser is needed, while the map remains interactive, and simpler than professional software. Also, R users can use this package instead of standard plot functions, because it provides a faster preview of the mapped data in relation to geographic reality provided by Google Maps.

Google Maps API is not suitable for storing large amounts of data and consequently *plotGoogleMaps* has the same constraint. One of the possible alternative solutions might be the combination of server side software (e.g. Geoserver) and client side software (e.g. Openlayers).

The future improvements of the package will include cartographic techniques for the visualization of quantitative data. The proportional symbols technique has already been developed in the package. Chart maps, bar and pie charts, over features which symbolize multiple attributes have not been implemented yet, except for the pie charts provided by *segmentGoogleMaps* command. All these features and other methods for achieving specific thematic mapping tasks have to be embedded in the new version of the *plotGoogleMaps* package. Another challenge will be developing a spatio-temporal representation, which is important for the dynamic GIS modelling.

Acknowledgments

This study is fully supported by the research projects No. III 47014: The role and implementation of the national spatial plan and regional development in the renewal of strategic research, thinking and governance in Serbia“, and TR 36035: Spatial, ecological, energy and social aspects of settlements' development and climate changes—interrelationships“, funded by the Ministry of Education and Science of the Republic of Serbia.

References

Asleson, R. and N. Schutta. 2005. *Foundations of Ajax*. Apress, Berkeley, CA. 273 p.

Becker, R.A. and J.M. Chambers. 1984. S: *An Interactive Environment for Data Analysis and Graphics*. Wadsworth, Belmont, CA, 550 p.

Best, D. 2006. Web 2.0 Next Big Thing or Next Big Internet Bubble? Lecture Web Information Systems, Eindhoven: Technische Universiteit Eindhoven.

Bivand, R. S., E.J. Pebesma, and V. Gomez-Rubio. 2008. *Applied Spatial Data Analysis with R*. Springer, New York, 378 p.

Brewer, C. A., G.W. Hachard and M.A. Harrower. 2003. ColorBrewer in Print: A Catalog of Color Schemes for Maps, *Cartography and Geographic Information Science* 30(1). pp. 5-32.

BuiltWith Web Technology Usage Statistics website: *http://trends.builtwith.com/websitelist/Google-Maps* Accessed September 21, 2011.

Burrough, P. A. and R.A. McDonnell. 1998. *Principles of Geographic Information Systems*. New York, Oxford University Press Inc. 333 p.

CORINE. 2000. Land cover, *http://www.eea.europa.eu/data-and-maps/data/corine-land-cover-2000-clc2000-seamless-vector-database*.

Cprogramming website: *http://www.cprogramming.com/* Accessed November 10, 2011

Dyhouse, G., J. Hatchett and J. Benn. 2003. Floodplain Modeling using HEC-RAS. Haestad Methods Inc., USA.

Eicher, C. L. and C.A. Brewer. 2001. Dasymetric mapping and areal interpolation: Implementation and evaluation. *Cartography and Geographic Information Science* 28, pp. 125-138.

FAO. 2006. World reference base for soil resources, by FAO-UNESCO-ISRIC. World Soil Resources Report No. 103, Rome.

Feick, R. and V. Deparday. 2010. Evaluating selected visualisation methods for exploring VGI. *Geomatica* 64(4), pp. 427-437.

Gartner, G. 2009. Applying Web Mapping 2.0 to Cartographic Heritage, *e-Perimetron*, pp. 234-239.

Garret, J. 2005. Ajax: A new approach to Web Applications. *http://www.adaptivepath.com/publications/essays/archives/000385.php*

Gesmann, M. and D. de Castillo. 2011. Using the Google Visualisation API with R. The R Journal, 3(2), pp. 40-44 *http://journal.r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf*

Gibin, M., A. Singleton, R. Milton, P. Mateos, and P. Longley. 2008. An Exploratory Cartographic Visualisation of London hrough the Google Maps API. *Applied Spatial Analysis and Policy*, Springer Netherlands, pp. 85–97.

Ghilani, D. and P. Wolf. 2006. *Adjustment computations—Spatial Data Analysis.* John Wiley&Sons. New Jersey, 611 p.

Goodchild, M. F. 2007. Citizens as voluntary sensors: spatial data infrastructure in the world of Web 2.0. *International Journal of Spatial Data Infrastructures Research* 2, pp. 24–32.

Grunsky, E.C. 2002. R: a data analysis and statistical programming environment– an emerging tool for the geosciences. *Computers & Geosciences* 28, pp. 1219–1222

Haklay, M., and A.F. Zafiri. 2007. Usability engineering for GIS – Learning from a snapshot. Presented at the 23rd International Cartographic Congress, Moscow, Russia, 4–10 August 2007.

Haklay, M., A. Singleton and C. Parker. 2008. Web Mapping 2.0: The Neogeography of the GeoWeb. *Geography Compass*. 2(6), pp. 2011–2039.

HEC (Hydrologic Engineering Center). 2008. HEC-RAS River Analysis System: Hydraulic Reference Manual.

Hydrologic Engineering Center, Davis, CA. Availabe at: *http://www.hec.usace.army.mil/software/hec-ras/documents/HEC-RAS_4.0_Reference_Manual.pdf*

Hengl, T. 2009. *A Practical Guide to Geostatistical Mapping, 2nd Edt*. University of Amsterdam, *www.lulu.com*, pp. 291.

Isaaks, E. and R. Srivastava. 1989. *An Introduction to Applied Geostatistics*. Oxford University Press, New York, 561 p.

Kilibarda, M. 2010. *plotGoogleMaps*: Plot HTML output with Google Maps API and your own data. R package version 1.0. *http://CRAN.Rproject.org/package=plotGoogleMaps*

Kilibarda, M. and M. Pejović. 2011. Application of open source/free software (R + Google Earth) in designing 2D geodetic control network, Proceedings of International scientific conference and 14th meeting of Serbian Surveyors "Professional practice and education in geodesy and related fields" 24-26 June 2011, Kladovo, Serbia. pp.216-225.

Kilibarda, M., D. Protić and I. Nestorov. 2010. Application of Google Maps API service for creating web map of information retrieved from CORINE land cover databases. *Bulletin of the Serbian Geographical Society/Glasnik Srpskog geografskog društva*, 90 (4), pp. 103-114.

Krunić, N., B. Bajat, M. Kilibarda and D. Tošić. 2011. Modelling the Spatial Distribution of Vojvodina's Population by Using Dasymetric Method. *Spatium* 24, pp. 45-50.

Loecher, M. 2011. RgoogleMaps: Overlays on Google map tiles in R. R package version 1.1.9.6. *http://CRAN.R-project.org/package=RgoogleMaps*

Mennis, J. 2003. Generating surface models of population using dasymetric mapping, *The Professional Geographer*, 55 (1), pp. 31–42.

Miller, C. M. 2006. A beast in the field: the Google Maps mashup as GIS/2. *Cartographica*. 41 (3), pp. 187–199.

National Research Council (U.S.) 2009. Mapping the Zone: Improving Flood Map Accuracy. Washington, D.C.: National Academies Press. *http://www.nap.edu/catalog.php?record_id=12573*

OSGeo website: h*ttp://trac.osgeo.org/proj/* Accessed October 10 2011.

Oracle website: *http://www.oracle.com/technetwork/java/javase/downloads/index.html* Accessed November 10 2011.

O'Reilly, T. 2005. What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software. *http://oreilly.com/web2/archive/what-is-web-20.html* (accessed 19.10.2011)

Pebesma, E. J. 2004. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences* 30(7), pp. 683–691.

Pebesma, E. J. and R.S. Bivand. 2005. Classes and methods for spatial data in R. *R News*, 5(2), pp. 9–13.

Perl website: *http://www.perl.org/* Accessed November 10th 2011.

Peterson, M. 2007. *The Internet and Multimedia Cartography*, Second Edition, Springer Berlin Heidelberg, pp. 35-50.

R-project website: *http://cran.r-project.org/doc/manuals/R-lang.html* Accessed October 1st 2011.

Radić, Z. 2007. Geotechnical models off alluvial sediments in New Belgrade, PhD thesis, Belgrade. 136 p.

RGZ website: *http://www.rgz.gov.rs/template1a.asp?PageName=2011_06_24_01 &LanguageID=3* Accessed October 10, 2011.

Rikken, M. G. J. and R.P.G. Van Rijn. 1993. Soil pollution with heavy metals - An inquiry into spatial variation, cost of mapping and the risk evaluation of copper, cadmium, lead and zinc in the floodplains of the Meuse west of Stein, the Netherlands. Technical report. Dept. of Physical Geography, Utrecht University. Doctoraalveldwerkverslag.

R Development Core Team. 2008. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Statistical Office of the Republic of Serbia. 2003. Knjige popisa stanovništva, domaćinstva i stanova u 2002. godini. Beograd: Republički zavod za statistiku Srbije (in Serbian).

Skarlatidou, A., and M. Haklay. 2006. Public web mapping: preliminary usability evaluation. Presented at GIS Research UK 2005, Nottingham, UK, 5–7 Apr.

Authors

Milan Kilibarda received his M.S.c degree in Geodesy from the University of Belgrade, Faculty of Civil Engineering, Department of Geodesy and Geoinformatics. He is currently a Ph.D. student in the same department and does research in space-time interpolation of daily meteorological variables for global land areas. His research area is automated geostatistical interpolation algorithms using open source software tools (R+OSGeo) and web mapping tools. Some of his contributions in the field of geostatistical modelling are mapping population change and modelling of spatial distribution of populations in Serbia. He is currently involved with the Interoperable Smart City services through an Open Platform for urban Ecosystems, innovation project. He is a teaching assistant in the field of cartography.

Branislav Bajat is an Associate Professor in the Faculty of Civil Engineering, Department of Geodesy and Geoinformatics at the University of Belgrade. His main research areas are the application of novel spatial and spatio-temporal statistical methodologies in the geosciences and environmental disciplines. His current research is based on the application of data mining techniques to spatial data and the development of data driven models for environmental modelling. ❏