

DESIGN AND IMPLEMENTATION OF A WEB-BASED APPLICATION FOR THE VISUALIZATION OF LARGE SCALE PHOTOGRAMMETRIC AND CARTOGRAPHIC DATA

Maria Spanaki^a, Lysandros Tsoulos^b

^a Surveying Eng. MSc Geomatics – PhD Student, ^b Assoc. Professor NTUA
Cartography Laboratory, Faculty of Rural and Surveying Engineering
National Technical University of Athens
9 H. Polytechniou, 157 80 Zographou Campus, Athens, Greece
spanaki@mail.ntua.gr, lysandro@central.ntua.gr

KEY WORDS: Photogrammetry, Cartography, Web-based, Standards, Open systems

ABSTRACT:

The objective of this paper is to show the way multiple source data like aerial photos, maps and descriptive information are combined and amalgamated to form an Internet or Intranet application. The visual outcome is a large scale, precise and detailed image/map displaying the interiors of buildings in a layered fashion, enabling the user to interact with the objects shown on it. Scalable Vector Graphics - SVG is an XML language for the encoding/representation of two dimensional vector data, especially designed for integration with other web standards. SVG provides those basic structures, which are needed for map generation and overcomes the disadvantages inherent to bitmap images displaying spatial data. In the domain of spatial data, SVG constitutes a new powerful standard for visualization and has a great potential along with other XML encoding languages like Geographic Markup Language - GML, Extensible Stylesheet Language Transformation – XSLT and Cascading Style Sheets – CSS for the development of Web-based applications utilizing open source software and standards. This paper elaborates on the way SVG data is generated from existing cartographic and photogrammetric sources and the use of scripts in order to provide the user with interactive and animation capabilities, resulting to a comprehensive and user friendly map in the Internet. More specifically, photogrammetric data is translated into a neutral format and subsequently to SVG graphics with the utilization of tools that implement this conversion that is carried out separately for the point, line and polygon data, which are subsequently combined into a uniform SVG image. The Graphics User Interface specifically developed for this application, enables the user to select a building from the specified area [i.e. a University campus], to virtually “enter” into the interior of the building and to visualize structural and other information relative with the use of the office space [libraries, laboratories etc] along with the corresponding data for the phone number, IP address etc.

1. INTRODUCTION

Photogrammetry and Remote Sensing are two major scientific domains enabling the collection, analysis and distribution of huge amounts of geospatial information. Such information in the form of aerial photos, satellite images, orthophotomaps, and topographic maps is essential to be easily exploited in a web based application environment. Since XML technologies are gaining wide acceptance and are incorporated as standards in software products, a need for utilizing photogrammetric and remote sensing data in a new fashioned way is evident.

This paper elaborates on a case study concerning the way photogrammetric data can be easily transformed and combined with other source data to build a web application based on the Scalable Vector Graphics - SVG format. The aim of the study is the development of a map application on the web - covering the area of a university campus - that enables the user to interact with floor layered information, which contains construction details just as shown in a vector file format like dwg or dgn. The basic geometric constructs like lines, points and polygons are extracted from aerial views using fundamental photogrammetric methods and contain information about the outline of campus' buildings in real world coordinates. The next phase refers to the transformation of all other existing drawing data at the floor level in a building to the same coordinate system. The transformation of vector file formats to SVG graphics is implemented using open source software tools available on the Internet. Since SVG has no compatibility problems with other web standards and raster file formats, the development of the final application is carried out in a very fast and efficient way

providing the end user with high graphic quality results along with interactive capabilities. Emphasis is given to the exploitation of capabilities that SVG provides and detailed references are made to specific elements, attributes and design ideas implemented behind what user sees.

2. TRANSFORMATION OF VECTOR MAP FILES TO SVG

Since SVG was introduced a few years ago, considerable efforts have been made in the direction of providing users with format conversion techniques, tools and software programmes for the transformation from vector proprietary formats to the non-proprietary SVG format. There are many such tools available on the Internet giving users the opportunity to convert vector data used in GIS applications to SVG web graphics. The major part of GIS data is stored in proprietary vector formats like shape files or cad files, so the need for tools that convert these files to SVG should be quickly fulfilled (Carto.net, 2003). One of these tools - utilized in this case study - is the CAD2SVG converter by Savage software. The CAD2SVG converter enables the conversion of DWG and DXF files to the SVG format. This can be applied to single files or directories. The same conversion can be also applied to DGN files, through the DGN2SVG converter. In both cases, using the option of compressing the SVG files to svgz format, the converter compresses the output which typically results in a file savings of 80%-90% over the original file size. In addition, when “smart origin” is selected, the converters automatically normalize the origin of the file, so

that it is best displayed in the Adobe SVG viewer. If there is a need for the origin to be maintained, for example if there is a need for a mapping application and the user wishes to query the location of objects, the original coordinates can be maintained (Savage Software, 2004).

In some cases, certain files contain more than just polygons, lines or points. Block data and attribute data is valuable information. The converters turn that data into organized XML, encapsulated in metadata tags. Then, with the use of standard JavaScript techniques the data can be displayed. If there is no metadata in the input document, there will be no metadata tags in the output converted file.

2.1 An overview of Scalable Vector Graphics

Scalable Vector Graphics (SVG) is a new graphics file format and Web development language based on XML. SVG enables Web developers and designers to create dynamically high-quality graphics from real-time data with precise structural and visual control. With this powerful technology, SVG developers can create a new generation of Web applications based on data-driven, interactive, and personalized graphics. SVG looks and feels very familiar, thanks to its roots in XML (W3C SVG, 2003).

SVG is text based, therefore encoding techniques can be learned by leveraging the work of others. This drastically reduces the overall learning curve. The JavaScript language and Document Object Model (DOM) are familiar to those using DHTML. Developers who use JSP, PHP, and ASP for HTML or text content today will be able to create graphics in the same way. Nowadays, there are software tools generating and handling SVG graphics. Tricky constructs like rollovers and interactive menus are easily created through such software programmes like Adobe Illustrator. No compatibility problems occur while working with SVG because is text based and works seamlessly with current Web technologies like HTML, Cascading Style Sheets (CSS), Extensible Style Sheet Language, SMIL, ASP, JSP, and JavaScript and formats like GIF, JPEG, PNG. By combining SVG with the existing Web technologies developers can create extremely rich interactive graphics. Any SVG graphic element can modify or control any other SVG or HTML element or object. Sophisticated user interfaces or even full-blown online applications can be created with SVG. And because SVG is text based, the text inside graphics can be translated for other geographies quickly (Adobe Systems, 2004).

Graphics created in SVG can be scaled without loss of quality across various platforms and devices. SVG can be used on the Web, in print and even on portable devices while retaining full quality. Another notable aspect while using SVG, is the reduction of maintenance cost. For example, a navigation button that normally requires a minimum of two raster files can be replaced by a single SVG file, rollover states and behaviours are specified via easily scriptable attributes such as colour, shape, size, text, or opacity. SVG can also reduce server loads by allowing client platforms to perform the graphics rendering. If the client platform has limited processing resources (i.e. PDAs and cell phones), the server can pre-render and optimize content before delivery. In both cases, the source content is the same. The overall goal is to have a single source file, which transforms gracefully in a wide variety of situations. Client-side rendering can also dramatically improve the user experience. For example, zooming in on an SVG-enabled map is extremely fast and can instantly provide additional details such as streets names, building addresses, and topographic information.

SVG has complete Unicode character support to display text in many languages, vertically, horizontally and bi-directionally. A

content developer may easily embed an unusual font, enabling text to be rendered as intended without making assumptions about the user's available fonts. SVG works effectively with style sheets to control presentation elements and attributes separating page aesthetics from content. Cascading Style Sheets (CSS) can be used not only for font characteristics (size, family and colour) but also for properties of other SVG graphic elements. SVG content can define colours using either an ICC (International Colour Consortium) profile or an RGB specification.

3. APPLICATION CONCEPT AND ARCHITECTURE

The basic concept for the implementation of this work follows two axes. One is to use new xml graphic technologies within a web environment combining traditional file formats and raster images and the other is to prove that this system architecture is easily implemented and gives the end user efficient web application results.

Technologies and formats that are used and implemented in all steps as shown in figure 1 are: html and SVG for web page representations, jpg and SVG formats for graphic images and JavaScript for user's interactions.

Initially the user selects a building as shown in a general campus map. Selection is made by clicking the SVG element that covers the building's surface. There is an on/off option for the raster aerial photo that covers campus area which is the background image of the first web page. In the next step the user selects the floor by clicking on graphic SVG elements. The background in this case is again a raster image showing building's façade. This alteration from campus ground plan to building's façade is necessary in order to enable the user to select the desirable floor.

In the next window there are two SVG images, one showing construct designed details and the second (hidden) gives to the user the option to point on every area that is highlighted when the mouse cursor is over it. The selection of an area is made on a different layer that contains only the outline of each room. This approach was selected in order to assist to the easy editing of the descriptive information of every room in a short coded SVG file.

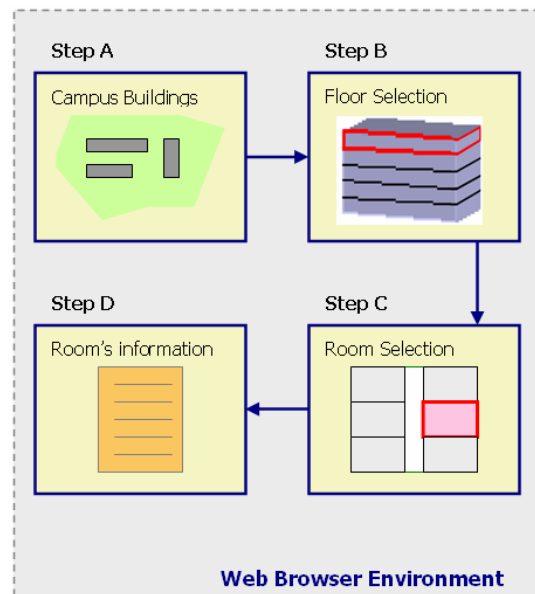


Figure 1: Application's concept and architecture

When the user selects a specific room, information is displayed in a separate web page. This page can be either an html or SVG file.

3.1 Data structure and file sizes

As mentioned earlier the, transformation of dwg files to SVG format was achieved with the use of CAD2SVGmodule. The source dwg file contains lines, dashed lines, polylines, hatches and polygons. Due to the fact that these structures are not identical in SVG, the converter uses mainly the <path> element to represent features' geometry. SVG provides the general <path> element, which when used can create a huge variety of graphical objects.

This general transformation of the various data constructs to one repeating used <path> element causes the output SVG file size to be five times larger than the one of the source. This can become really a problem in a web environment and for this reason SVGz compression is applied resulting to a zipped file 50% smaller than the original dwg.

4. APPLICATION PROGRAMMING AND IMPLEMENTATION

In this section, apart from the original application screenshots, there are explanations about the implementation of several hot spots in SVG coded images.

At step B the user selects the floor layer he wants to virtually visit, from a screen that shows the exterior of the building, by pointing on the desirable level, which is actually an SVG image (Figure 2). SVG images can be embedded in html pages using either html <embed> or <object> element as shown in the code above.

```
<embed src="NestedSVG.svg" width="500" height="400" type="image/svg+xml">
```

```
<object data="file:///c:/ NestedSVG.svg" width="500" height="400" type="image/svg+xml">
```



Figure 2: Raster building image with SVG rectangle elements designed for the floor selection.

The picture that the user gets at step C is a ground plan of the selected floor in SVG graphic format. This image can be zoomed in and out, panned using the popup menu of the Adobe viewer by right clicking inside the SVG image. While moving the mouse cursor over the rooms of the building, some areas such as rooms, public areas etc. are highlighted (figure 3).

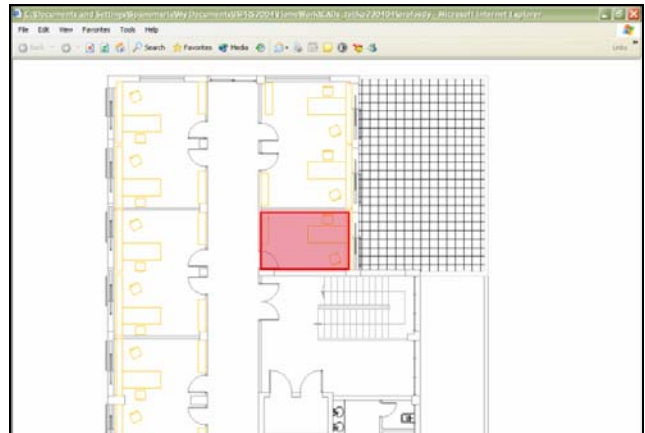


Figure 3: Plan of the selected floor where highlighted room is user's selection

By left clicking on a highlighted feature a new web page is loaded showing information about the selected area (Figure 4). Highlighting can be easily implemented using the "set" element from the SVG specification combined with "mouseover" and "mouseout" options or by java scripting.

```
<set begin="r42_1.mouseover" end="r42_1.mouseout" attributeName="fill" to="#CCCCFF"/>
```

SVG linking capabilities give the option this page to be rather an html page (Figure 3) or an SVG one.

Code <a xlink:href="Testpage2.html"> opens an html page.
Code <a xlink:href="Testpage1.svg"> opens an svg page.

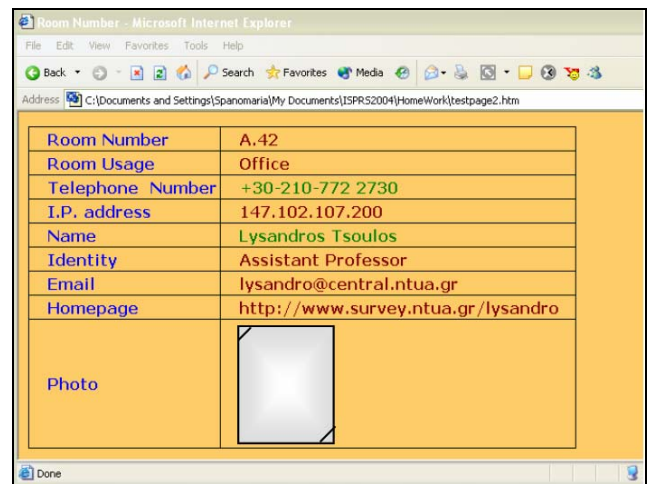


Figure 4: information shown after clicking on a ground's plan room in html web format.

SVG has the advantage of presenting text information in user's system language. For example, in case the web page that appears after users' selection on a highlighted area is in SVG format and if the proper SVG code is used, all descriptive information about rooms will be shown in Greek or English characters. This can be of course implemented in every language that can be part of the SVG code.

The following slice of SVG code shows that if user's system language is English the "Telephone Number" text will appear in the SVG web page, or if user's system language is set to Greek the "Αριθμός Τηλεφώνου" text will be shown accordingly.

```

<switch>
<text x="..." y="..." style="stroke:....; fill:....; font-size:20;"
systemLanguage="en">
Telephone Number
</text>
<text x="..." y="..." style="stroke:....; fill:....; font-size:20;" 0;"
systemLanguage="el">
Αριθμός Τηλεφώνου
</text>
</switch>

```

In case the web page that appears after users' selection on a highlighted area is in SVG format and there is a need for embedding a raster image, SVG uses the <image> element as follows:

```

<image x= "10" y="20" width="200px" height="150"
xlink:href="testImage.jpg">

```

5. DATA DISSEMINATION AND INTEROPERABILITY ISSUES

SVG is an XML language enabling users to "read" the information contained in SVG files. This information concerning both geometry and descriptive data, in some cases has to be used in other applications or disseminated among different users. The solution proposed is the use of Geography Markup Language - GML. GML is an XML standard developed by the Open GIS Consortium - OGC designed to support interoperability among different data models and feature representations by providing a common data model, a set of basic and sophisticated geometry tags to describe the spatial and non spatial features (OGC GML, 2003).

Based on XML technologies (XML and XML Namespaces), GML is extensible, supports definitions of profiles, is open and vendor independent. The value of using the GML is well understood when geographic data have to be transported to other recipients in heterogeneous environments. Based on the standard mark-up elements, the receiving party knows exactly what each data component means and how to extract it so that nothing gets lost or distorted in the transport and translation process. In order to achieve that, the receiving party gets apart from GML files the corresponding application schema. The application schema is the implementation of a schema language that models geographic information in GML and defines rules for application schemas. Also defines standard elements and types for use (Portele, 2004).

Whenever interoperability issues have to be taken into account, the standard procedure that has to be followed is the transformation of the source data files to GML format. This intermediate phase is considered necessary in order that potential users are capable to utilize the GML files without facing any interoperability problems (Figure 5). GML upholds the principle of separating content from presentation, thus a styling mechanism is required to display GML data. Potential graphical display formats apart from SVG are the Microsoft Vector Markup Language (VML) and the X3D (Lake, 2003).

6. BENEFITS

The huge amount of existing - large and small scale - data may benefit from the XML technology and more specifically from SVG. There are a number of converters enabling users to easily

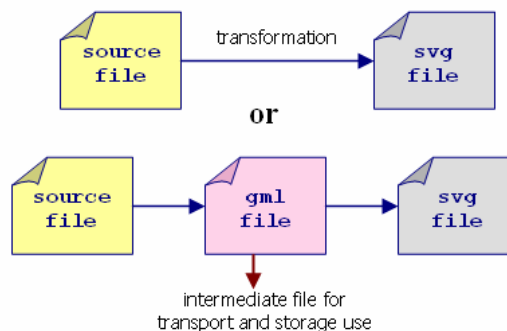


Figure 5: Alternative approaches to SVG result

publish CAD, PDF and other popular format projects online as SVG, the open-standard W3C recommendation for vector graphic viewing on the web. The flawless conversion process outputs compact SVG that can be viewed in the free viewers which offer printing, zoom, pan, copy, and many other features. The converters automate many of the formerly manual processes associated with deploying and managing the publishing of files on web sites. More specifically the conversion process and the resulting graphics have the following characteristics:

- Maintain the integrity of the original files
- Have the option to "invert" the colour scheme, which transforms the standard black background to the web standard white background and at the same time change any line colours that would appear to be "washed out" when viewed with a white background
- Save as SVGZ compressed files for delivery
- Convert single files or whole directories
- All the hard earned data that is in the block attributes is maintained, created at easily accessible metadata
- Viewers are free, unlike other proprietary viewers available on major platforms and OS's
- XML style of document structure means edits can be done in any text

7. CONCLUSIONS

The SVG format is emerging through the cooperative efforts of the World Wide Web Consortium (W3C) and its members. To Web developers, SVG will look and feel very familiar, thanks to its roots in XML. SVG is text based; therefore, coding techniques can be learned by leveraging the work or code of others. This drastically reduces the overall learning curve. The JavaScript language and Document Object Model (DOM) will be very familiar to people who use DHTML. Developers who use JSP, PHP, and ASP for HTML or text content today will now be able to create graphics in much the same way.

SVG is text based and works seamlessly with current Web technologies like HTML, GIF, JPEG, PNG, SMIL, ASP, JSP, and JavaScript providing to the user full compatibility.

Graphics created in SVG can be scaled without loss of quality across various platforms and devices. SVG can be used on the Web, in print and even on portable devices while retaining full quality.

Sophisticated user interfaces or even full-blown on-line applications can be created with SVG. And because SVG is text based, the text inside graphics can be translated for other geographies quickly, often collapsing localization efforts to days instead of weeks.

8. BIBLIOGRAPHY

Adobe Systems, 2004 , www.adobe.com SVGZone.

Carto.net, 2003, www.cartonet.net.

Lake R., 2003. What is a Map Style? Building on GML and SVG. In Proceedings of SVG Open Conference.

OGC GML, 2003 Geography Markup Language (GML) 3.0 Implementation Specification, OpenGIS Consortium.

Portele C., 2004. Interoperability Workshop and Tutorial, 7th AGILE conference on Geographic Information Science.

Savage Software, 2004, www.savagesoftware.com.

W3C SVG 2003, Scalable Vector Graphics (SVG) 1.1 Specification, World Wide Web (W3C) Recommendation.