Converting raster images to XML and SVG

The potential of XML - encoded images and SVG image files in Geomatics

**Keywords:** SVG, XSLT, GML, Geomatics, Raster images

**Byron Antoniou**  
Survey Engineer. Captain HMGS  
National Technical University of Athens  
H. Polytechniou 9, 157 80 Zographou Campus  
Athens, Greece Phone: +30-210-7722730 Fax: +30-210-7722734  
rs02626@mail.ntua.gr

_Biography_

Byron Antoniou is an Officer in the Greek Army and since 1998 serves in the Hellenic Military Geographical Service (HMGS). He graduated from the Department of Rural and Survey Engineering (National Technical University of Athens) in July 2004, being the top graduate in his department. Currently his interests concentrate in the exploitation of GML and SVG technologies for cartographic procedures.

**Lysandros Tsoulos**  
Cartography Laboratory, Faculty of Rural and Surveying Engineering  
National Technical University of Athens  
H. Polytechniou 9, 157 80 Zographou Campus  
Athens, Greece Phone: +30-210-7722730 Fax: +30-210-7722734  
lysandro@central.ntua.gr  
http://www.survey.ntua.gr

_Biography_

Prof. Lysandros Tsoulos graduated from the Department of Rural and Surveying Engineering (University of Thessalonica) in 1972. He received his Dr. Eng. degree from NTUA in 1989. His dissertation entitled _The Electronic Chart and its Use in Navigation_ contributed to the standardization of the content and functionality of electronic charts on board the ships. He worked for the Hellenic Navy Hydrographic
Service for 16 years as director of the directorates of Cartography and the Computing Center.

Currently he is an Associate Professor at the Department of Rural and Surveying Engineering, Vice-Chairman of the Department and Director of the NTUA Geomatics Center. He has 60 publications in national/international journals and conference proceedings on Digital Mapping and Geographical Information Systems. He has also participated in a number of projects like: Impact II (Environmental and Social Impact of Air Quality), STRIDE, MEDSPA (Management of Ecosystems), WERATLAS (Atlas of wave energy resource in Europe), GEOMED (Geographical Mediation System), Eurostat/GISCO (Cartography and Map Design) and STATLAS (Statistical Atlas of the European Union). He is member in a number of National and International Scientific Committees.

Abstract

The advent of XML-based technologies has surpassed even the expectations of the most imaginative users. New, revolutionary ideas are emerging for the storage, sharing and display of data. In addition, new formats have been created for almost all kinds of data, applications and knowledge domains. A considerable - and still growing - number of specifications have already been issued from international organizations (W3C and OGC) in order to provide every possible assistance to the user community, to meet their needs and promote their efforts.

Watching this frenzy trend of transforming everything into XML-based structures, one thing seems really odd: raster images. Languages like HTML and SVG do not seem to bother about having such an unlike structural feature in their environment. Instead, both of them provide means (i.e., <img> or <image> elements) for the incorporation of raster images in text based files, usually with an inline reference. From a more objective point of view, somebody would have thought of raster formats as an obstacle. The existing specifications instead of dealing with this problem, bypassed it. Users followed this idea and continue to treat raster formats more or less like a requisite tool for their work. The blunt truth is that there is nothing common between XML-based structures and raster images. Raster image encoding is neither text based nor human readable. It cannot be parsed, checked for validity or well forming. Moreover, the raster image content has
almost no flexibility (apart from resizing) in an XML-based environment, since pixel values are well locked inside the raster formats.

This paper deals with the conversion of raster images into XML and SVG files. Each one of these two perspectives has a different objective. Converting an image from raster format into an XML file enables the user to get the information residing in the image in a text format, in accordance with international standards. The user can then select, read and manipulate every single part of the XML file or the file as a whole, in a number of ways in accordance with the application at hand. This is the starting point for classification, statistical processing, filtering or the implementation of other algorithms with the use of XML technology. Issues like the storage and sharing of image information may acquire a new meaning. Converting an image from raster to SVG is even more exciting. An SVG image file enjoys all the above-mentioned advantages and in addition the user can see the effect of every change imposed on it. This leads to the last issue that this paper deals with, Geomatics. Nowadays, Geomatics is a sector that depends on images to such an extend that one could tell that images are the most valuable source for collecting geographic data. Feature extraction from a raster image is a very common task for geographical organizations around the world. Through the combination of image information and SVG code instead of vector data, digitization could produce scalable vector data (i.e., <point>, <line>, <polyline>, <polygon> etc. elements) or GML (Geography Markup Language) encoded data.

Up to now the code written in Visual Basic performs the conversion from raster images to XML and SVG files. The ongoing efforts concentrate on:

- Photogrammetric applications
- Image processing
- Digitization and export to GML.

**Table of Contents**

1. Problem Statement
2. The conversion of raster images to XML code
3. Converting raster images to SVG code
4. The benefits to Geomatics
1. Problem Statement

It is a fact that raster images and the knowledge domain of Geomatics have a very close relationship. That is because the use of raster datasets is the most common way to gather geographic information. Raster images carry huge amounts of information. This information is stored in a tabular way and each structural element of this table (pixel - picture element) usually preserves three values for red, green and blue (in three-band true color images) in a range between 0-255. Most Web users see this table (the raster image) as a pretty scene and nothing more. For a Geomatics application, though, there is more than that into an image. In an aerial photo or a satellite image, for example, we have recording of the reflection of sun-beams upon the surface of earth, and the values stored in each pixel represent the reflection that took place at a specific fragment of the surface covered from the aerial photo. Taking into consideration that each material reflects sun beams (which consist of visible light, infrared, ultraviolet radiation etc.) in its own way, it can easily be understood that the recorded values in a raster image can lead us to useful conclusions. In addition, there is a very precise photogrametric process which can register an image to real world coordinates. That means that each pixel of a raster image conveys both qualitative and locational/geometric information. In other words, when it comes to Geomatics, separate pixel values and geometric information is what really matters in raster images.

**Figure 1** shows oblique normal color (a) and color infrared (b) aerial photographs. The football field has artificial turf with low near-infrared reflectance. That is why in normal color photo the whole region appears with green color and in the color infrared photo the real vegetation is red and the artificial black. (The photographs were taken from the book: *Remote Sensing and image interpretation*, Lillesand and Kiefer)
This paper aims at the description of the approach behind the conversion of raster datasets in XML and SVG.

2. The conversion of raster images to XML code

The best way to derive pixel information from a raster image, is to segregate pixel values in their components (usually red, green, and blue). This is not a tough process using Visual Basic (or other programming languages). Pixel values are separated using the standard VB functions, which first get hexadecimal values and then convert them to decimal ones. Figure 2 shows the VB code for segregating pixel values of a raster image and assign them in three discrete variables:
For xi = 0 To width
    For xj = 0 To height
        pixel& = Picture1.Point(xi, xj)
        red = pixel& Mod 256
        green = ((pixel& And &HFF00FF00) / 256&)
        blue = (pixel& And &HFF0000) / 65536
    Next xj
Next xi

**Figure 2: The VB code for the segregation of pixel values**

In order to reconstruct the information that a raster image holds in terms of XML encoding, we must put each value in a different element as shown in **Figure 3** and the sample XML file in **Figure 4**:

```
<?xml version="1.0" encoding="UTF-8"?>
<image xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="E:\SVGenie_ver2\SchemaFromVB.xsd">
    <pixel id="0" row="0" column="0">
        <r>201</r>
        <g>171</g>
        <b>81</b>
    </pixel>
</image>
```

**Figure 3: The segregation of pixel information**
Figure 4: XML-encoded pixel values

The above XML file conforms with the schema shown in Figure 5:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created by Antoniou Byron with XMLSpy v2004 rel. 3
(http://www.xmlspy.com) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="image">
    <xs:annotation>
      <xs:documentation>Schema for XML encoded pixel values</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:schema>
```
Figure 5: The XML Schema for encoding pixel values
Both XML-encoded images and the corresponding XML Schema can be generated by SVGenie, a program written in Visual Basic which reads raster images and transforms them into XML code. The same program converts raster images to SVG encoding.

### 3. Converting raster images to SVG code

The conversion of raster images in SVG-encoded image files is carried out as follows: The tabular structure of raster format is reproduced with an array of `<rect>` elements. The size of each `<rect>` element is set to 1px. The rendering of each rectangle is the same as the equivalent pixel in the raster image. Hence, the most crucial point is the existence of a logical link between the exact place of the pixel in the raster image and the coordinates in the canvas of the SVG image. Given that each position of an array can be defined uniquely using i,j (row, column) coordinates, this problem is easily solved. The above procedure ensures that both qualitative (i.e. pixel values of all the bands used in the image) and geometric information are conveyed without mistakes. Geometric information relates to relative position of each rectangle inside the array and to the absolute position of each rectangle in real world coordinates. The last step can be defined accurately with the use of SVG elements that control `viewport space` and `user space` in a SVG file.

Another important issue is the way that rendering information should be stored in an SVG image file. It is already known that there are three possible ways to embody styling instructions to an SVG file using CSS:

- External CSS style sheets referenced from the current
- Internal CSS style sheets (i.e., style sheets embedded within the current document, such as within an SVG style element)
- Inline style (i.e., CSS property declarations within a style attribute on a particular SVG element)

All the above methods can be used for the styling of the rect elements. It is recommended, though, the avoidance of the last method since it leads to extremely large files. An optimized method should be followed in order to avoid multiple entries of the same color.

**Figure 6** shows the SVG code for creating a white cross in black background and **Figure 7** shows the result:

```xml
<?xml version="1.0" encoding="UTF-8"?>
```
<svg>
  <defs>
    <rect id="p" width="1px" height="1px"/>
    <style type="text/css">
      #a {
        fill:rgb(0,0,0);
      }
      #b {
        fill:rgb(255,255,255);
      }
    </style>
  </defs>
  <g>
    <use id="a" xlink:href="#p" x="0px" y="0px"/>
    <use id="a" xlink:href="#p" x="0px" y="1px"/>
    <use id="b" xlink:href="#p" x="0px" y="2px"/>
    <use id="a" xlink:href="#p" x="0px" y="3px"/>
    <use id="a" xlink:href="#p" x="0px" y="4px"/>
    <use id="a" xlink:href="#p" x="1px" y="0px"/>
    <use id="a" xlink:href="#p" x="1px" y="1px"/>
    <use id="b" xlink:href="#p" x="1px" y="2px"/>
    <use id="a" xlink:href="#p" x="1px" y="3px"/>
    <use id="a" xlink:href="#p" x="1px" y="4px"/>
    <use id="b" xlink:href="#p" x="2px" y="0px"/>
    <use id="b" xlink:href="#p" x="2px" y="1px"/>
    <use id="b" xlink:href="#p" x="2px" y="2px"/>
    <use id="b" xlink:href="#p" x="2px" y="3px"/>
    <use id="b" xlink:href="#p" x="2px" y="4px"/>
    <use id="a" xlink:href="#p" x="3px" y="0px"/>
    <use id="a" xlink:href="#p" x="3px" y="1px"/>
    <use id="b" xlink:href="#p" x="3px" y="2px"/>
    <use id="a" xlink:href="#p" x="3px" y="3px"/>
    <use id="a" xlink:href="#p" x="3px" y="4px"/>
    <use id="a" xlink:href="#p" x="4px" y="0px"/>
    <use id="a" xlink:href="#p" x="4px" y="1px"/>
  </g>
</svg>
4. The benefits to Geomatics

Since pixel values are available in XML encoding the way is open for the implementation of classification algorithms based on these values. Up to now these algorithms were part of proprietary software that had full control over the format used to store raster images. The advent of open standards enables users to store raster files in XML code and apply any algorithm on the real data. The most common way to do this is utilizing XSLT. XSLT is an XML-based specification supporting the transformation of XML files from one form to another. XSLT has all the necessary programming tools needed (conditional statements, loop functions etc) to carry out the manipulation of the XML-encoded images. A factor that contributes to this is the simple and straightforward structure of the XML file that describes the raster image and the corresponding XML Schema (as shown in Figure 5 and Figure 6). The XSLT style sheet could, for example, include template rules that examine whether each pixel value belongs to a certain range of values (which means that the reflection of sun beams comes from a specific material) and if so apply templates to change the pixel values (classification). Figure 8 shows the XSLT stylesheet needed to transform a XML encoded image with black and white pixel values to another XML encoded image with green and white pixel values:

```xml
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" media-type="image/svg+xml"
indent="yes"/>
  <xsl:template match="/">
    <image name="test.jpg">
      <xsl:for-each select="eikona/pixel">
        <xsl:variable name="pid" select="@id"/>
      </xsl:for-each>
    </image>
</xsl:template>
</xsl:stylesheet>
```
Figure 8: XML stylesheet for XML encoded images classification

Another step in the image classification process could be the implementation of region growing algorithms, which could lead (again with the use of XSLT) to the creation of SVG or GML data.

Apart from storing three-band images, XML could as well serve the storage of more sophisticated raster data. Such data could be multi-spectral aerial photos or even satellite images. This kind of images consist of a large number of bands; which record the sun-beams reflection of a very small segment of spectrum. Of course such XML encoded images will need another, more elaborated, XML Schema.

Figure 9 shows the steps followed in order to crate SVG and GML data from raster images:
Similar to XML encoded images; SVG image files have the advantage of the segregated pixel values along with the ability to display the result of every change imposed on them. The use of XSLT in this case could be the equivalent of applying either image processing filters or geo-reference transformations. There is a large number of image processing tasks (i.e. creation of ortho images) that can be accomplished with the use of XSLT over well formed files that have as a discrete entity the pixel value of every band off a raster dataset.

The output of image processing using XSLT could be used as input for another set of algorithms supporting the creation of vector data (as it usually happens in Geomatics applications). The execution of raster to vector algorithms could lead to SVG elements which can be transformed (with the use of XSLT) to GML data. Another perspective could be the on-screen digitization (with the use of ECMAScript) of SVG elements followed by the transformation to GML data.

Finally, even the use of raw raster datasets could be used in order to digitize geographic features on screen as SVG elements and then transform them into GML data.

**Figure 10** shows in full extent the potential of XML- encoded images and SVG image files:
5. Drawbacks

The above described process has a number of drawbacks, the most important being the large size of the produced files. The analytical recording of each pixel value results to the creation of files that are extremely large compared to the initial ones. On the other hand, though, the use of compressed XML and SVG files, sets that problem in an realistic level given the fact that the files created are approximately 2,5 times larger than the original raster datasets.

Another drawback is the delay in the rendering of the SVG image files. Any SVG viewer should read a CSS instruction for every <rect> element in the file. This influences considerably the time needed for rendering the whole image.

6. Conclusions

The transformation of raster images into XML encoding, apart from changing the way files are stored, contributes substantially to the way users interact with real data. The transformation from Binary to Unicode format is easily implemented and besides the graphics, the result is enriched with qualitative information, which can be further
exploited by the user. This is very important in the domain of Geomatics where the knowledge of both geometry and attribute information is indispensable for further processing and utilization of spatial data, previously encoded in a raster format.

The conversion of raster images in XML and SVG code constitute the initiative for an in-depth research on the exploitation of open standards in Geomatics. The importance and the tremendous amount of information stored in raster datasets along with the capabilities of XML technologies (XML, XSLT, GML and SVG) in Geomatics, set a very promising environment for both developers and users.

Bibliography

[ECMA]

[SVG]
Scalable Vector Graphics (SVG) 1.0 Specification, J. Ferraiolo, editor, W3C Recommendation, 4 September 2001. Available at (http://www.w3.org/TR/SVG)

[XPath]
XPath W3C Recommendation, http://www.w3.org/TR/xpath

[XSLT]
XSLT W3C Recommendation, http://www.w3.org/TR/xslt