XML-BASED INTERACTIVE 3D CAMPUS MAP

JULIAN H. KANG, JIN G. PARK
Texas A&M University, 3137 TAMU, College Station, TX 77843-3137
juliankang@tamu.edu, jpark@neo.tamu.edu

AND

BYEONG-CHEOL LHO
Sangji University, 660 Woosan-dong, Wonju, Kangwon-do, Korea
bclho@mail.sangji.ac.kr

Abstract. This paper presents the development of a prototype XML-based 3D campus map using the 3D VML library. Many universities in the U.S. use two-dimensional (2D) raster image to provide the campus map along with additional building information on their Web site. Research shows that three-dimensional (3D) expression of the 3D objects helps human beings understand the spatial relationship between the objects. Some universities use 3D campus maps to help visitors more intuitively access the building information. However, these 3D campus maps are usually created using raster images. The users cannot change the view point in the 3D campus map for better understanding of the arrangement of the campus. If the users can navigate around in the 3D campus map, they may be able to locate the building of their interest more intuitively. This paper introduces emerging Web technologies that deliver 3D vector graphics on the Web browser over the internet, and the algorithm of the prototype XML-based 3D campus map. Some advantages of using VML in delivering the interactive 3D campus map are also discussed.

1. Introduction

Many universities in the U.S. use two-dimensional (2D) raster maps on their Web site to help visitors locate the buildings of their interest. Sometimes the image of a certain building in the map is linked with the Web page that provides the additional building information. The 2D image map is frequently used for providing the campus arrangement via the Web site because 1) the 2D image map may be the easiest way of delivering the geographic

arrangement of the campus over the Internet; 2) many end-users are expected to understand the relationship between the buildings by reading the conceptual 2D map. Indeed, abstract 2D drawings sometimes express multi level information in one sheet efficiently. Tufte (1990) demonstrated that a graphic timetable for Java railroad line, drawn in 1937, effectively expressed six different types of information on one sheet using station title and inclined lines. However he also demonstrated how it would be difficult and confusing to understand the 3D world via the 2D expression using Galileo's 2D observation of sunspots' movement. It was not easy for Galileo to prove that the sun itself is rotating based on his 2D observation. Tufte reasoned that the spatial relationship among 3D objects can be more intuitively explained using 3D expression. It is therefore reasonable to expect that the 3D campus map would help visitors locate the buildings of their interest more effectively.

Some universities in the U.S., including Texas A&M University (http://www.tamu.edu/map/building/detail/ACAD.html), use the 3D campus map in order to take advantage of the merit of 3D expression. Rich information that is carried via 3D expression is expected to help the visitors grasp the arrangement of the campus efficiently. However, most 3D campus maps used in these universities do not allow the visitors to navigate around because they use a birds-eye-view type raster image for the 3D campus map. If the visitors wanted to know how they can access the building of their interest from the far side of campus from the view point, they would have to rearrange the relationship between the buildings in their mind. We can infer from this situation that the birds-eye-view type 3D campus map sometimes may not provide the visitors with intuitive help for identifying the access road for a certain building.

In order to resolve the limitation of the birds-eye-view type 3D campus map, some universities use a Virtual Reality Modeling Language (VRML) model. Since the VRML model allows the visitors to navigate around in a 3D environment and familiarize with the arrangement of the campus, it is considered an effective supplementary tool for the 2D campus map. James Cook University in Australia demonstrates a 3D VRML model of JCU Douglas Campus at http://www.tesag.jcu.edu.au/jcu/virtual/virtualcampus.html, in which the visitors can walk around and explore the campus. The views of individual buildings in this VRML model are linked to the information pages for that building, in which the visitors can see the photos and description of the building. Students at University of Tasmania in Australia also created a VRML model of Sandy Bay campus in which the landscape was modeled using realistic contour data (http://www.comp.utas.edu.au/app/KXA351/2000/uni3d.jsp). This model enables the visitors to take tours from one facility to another by simply selecting the two facilities of interest.

2. Motivation

VRML is a good open format to deliver the 3D vector graphics on the Web browser. However, we discovered several limitations in the VRML-based 3D campus map. First of all, the VRML-based 3D campus map is only displayed on the VRML plug-in software such as Cortona or Cosmo Player. Manipulating the 3D campus map may be limited by the functions of the VRML browser. Secondly, VRML can not be incorporated with HTML because VRML is not a markup language and the VRML code is saved in a separate file. The developers may not be able to enhance the user interaction of the VRML-based 3D campus map using other Internet technologies such as JavaScript in order to allow the visitors to do more than just browsing the VRML model of the campus.

The computer industry made some advancement recently in delivering vector graphics on the Web browser using eXtensible Markup Language (XML). XML is a simple and very flexible text format that is designed to meet the challenges of large-scale electronic publishing (W3C, 2003). In 1998, a technical specification of Vector Markup Language (VML) was submitted to World Wide Web Consortium (W3C) by Microsoft, AutoCAD, and other companies. VML is an application of XML and it defines a format for delivering a text-based vector graphics on the Web browser over the Internet. The VML code can be incorporated with the HTML code, and it runs on the Microsoft Internet Explorer without any plug-ins software installed (W3C, 1998). In 2001, the W3C announced Scalable Vector Graphics (SVG), which is another specification for describing 2D graphics in XML (W3C, 2001). SVG is known a better, cleaner, and more complete standard than VML, however the users need to install a plug-in viewer to see the vector graphics on the Web browser. Like VML, it is reported that the developers can create rich interactive graphics by combining SVG with existing Web technologies such as JavaScript, DOM (Document Object Model), Java, or Visual Basic.

Although VML and SVG seem to be good tools for delivering vector graphics and additional information to the end users over the Internet, they are designed only for displaying 2D vector graphics on the Web browser. It was Gareth Richards at gersolution.com who introduced a 3D VML library that displays 3D vector graphics using VML for the first time (Richards, 2000). The 3D VML library is essentially a collection of JavaScript codes that creates a perspective view of the 3D model using 2D vector graphics based on the viewer's location. Complicate matrix manipulation, therefore, is an unavoidable process. Lutz Tautenhahn (2002) later introduced a SVG-VML-3D, which was essentially similar to the 3D VML library, for illustrating 3D vector graphics on the Web browser. Examples introduced by Richards inspired us to develop a Web application that could do more than

just showing the 3D VRML model on the Web browser. We expected that the 3D VML library should enable us to try a new way of delivering the 3D campus map over the Internet.

3. Development

3.1. XML DATA ISLAND

The 3D VML library defines 3D polygons in the XML data island. The XML data island is an "island" of data that is embedded inside HTML file (Microsoft, 2003). It can be accessed via the standard Document Object Model (DOM) Application Programming Interface (API). The syntax of the XML data island used in the 3D VML library is introduced in Figure 1.

```
<XML>
  <SCALE> numeric value of scale </SCALE>
  <POINTS>
    <P>
      <X> x coordinates of the point 1 </X>
      <Y> y coordinates of the point 1 </Y>
      <Z> z coordinates of the point 1 </Z>
    </P>
    <P>
      <X> x coordinates of the point 2 </X>
      <Y> y coordinates of the point 2 </Y>
      <Z> z coordinates of the point 2 </Z>
    </P>
    repeat for all points used in the 3D model
  </POINTS>
  <FACES>
    <F id= identification name of the face>
      <V> the first point used for fabricating the face </V>
      <V> the second point used for fabricating the face </V>
      <V> the third point used for fabricating the face </V>
      repeat for all points used for fabricating the face
    </F>
    repeat for all faces used for building the 3D model
  </FACES>
</XML>
```

Figure 1. Syntax of the XML data island

The XML data island contains three root level sections: 1) SCALE, which defines the scale of the polygon set; 2) POINTS, which defines the points

that make up the polygon set; and 3) FACES, which defines a face of the polygon. F tag has the unique "id" that identifies the face of the 3D object. The content of the F tag contains a list of faces separated by the V tag (Richards, 2001).

Although the XML data island seemed to be simple to use, we expected that it would be confusing to create the large XML data island manually. We decided to develop our own database structure that would store the 3D geometry of the campus building more intuitively, and convert the database model into the XML data island using a computer application. The structure of the database we designed is shown in Figure 2.

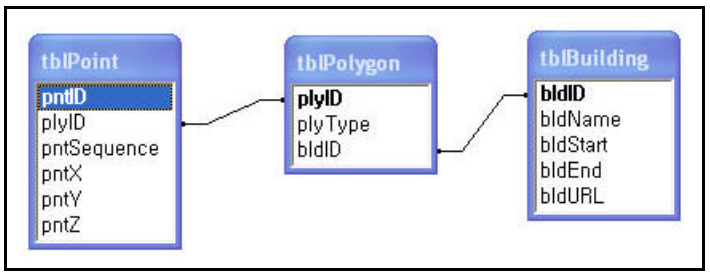


Figure 2. Database structure for the 3D campus map

The database consists of three tables: 1) tblPoint, which defines the points that make up the polygon set; 2) tblPolygon, which defines the polygons that make up the buildings; and 3) tblBuilding, which defines the buildings. These tables have one-to-many relationships with each other and provide a convenient environment for defining the points, polygons, and buildings as shown in Figure 3. Once the users click the "+" icon of a certain entity in the table, the view of Microsoft Access is expanded and allow the user to enter data for that entity.

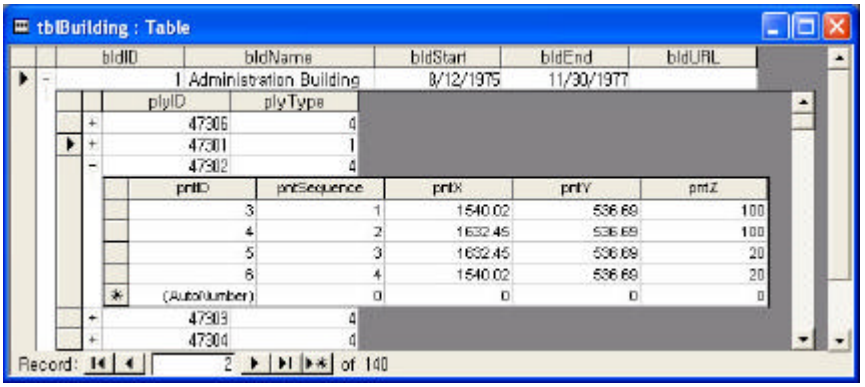
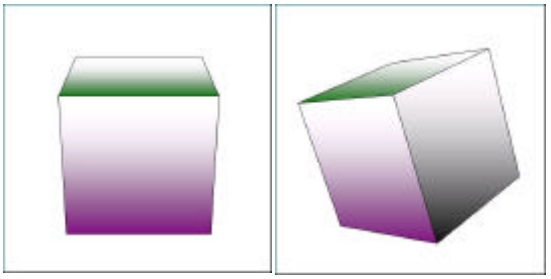


Figure 3. Data entry for the 3D campus map

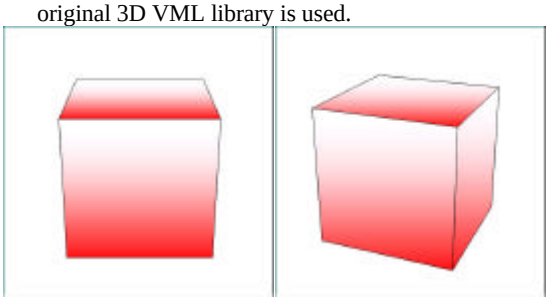
In order to covert the geometry in the database into the XML data island automatically, we developed an interactive Web application using Active Server Pages (ASP). This application retrieves the 3D geometry of the building from the database in the Web server, and creates an XML file in the same server using the format introduced in Figure 1.

3.2. MODIFICATION OF 3D VML LIBRARY

The 3D VML library rotates the model on the global axis by default. Rotating the 3D model on the global axis may not always work for browsing the building because sometimes the 3D model tends to lean if it is rotated on the global z axis as shown in Figure 4 (a). In order to have the buildings to stand up vertically no matter what the end users do, we needed to have the 3D model to be rotated on the local z axis as shown in Figure 4 (b). We modified the rotating function of the 3D VML library so that the 3D model is rotating on the local z axis.



(a) The box is rotated on the global axis when the original 3D VML library is used.



(b) The box is rotated on the local axis after the 3D VML is modified.

Figure 4. Modification of the 3D VML library for rotation

Once the XML data island is created, the following tag is used to 1) pick up the geometry defined in the XML data, and 2) generate the object that belongs to the Web browser.

```
<XML id=3dcampus onreadystatechange=fnStartInit();
        src="3Dcampus.xml">
</XML>
```

The 3D VML library takes this XML object to fabricate several polygons and create a perspective view of the 3D campus model using these polygons. The 3D VML library then displays the perspective view on the Web browser using the VML syntax. Overall process of displaying the 3D model on the Web browser is illustrated in Figure 5.

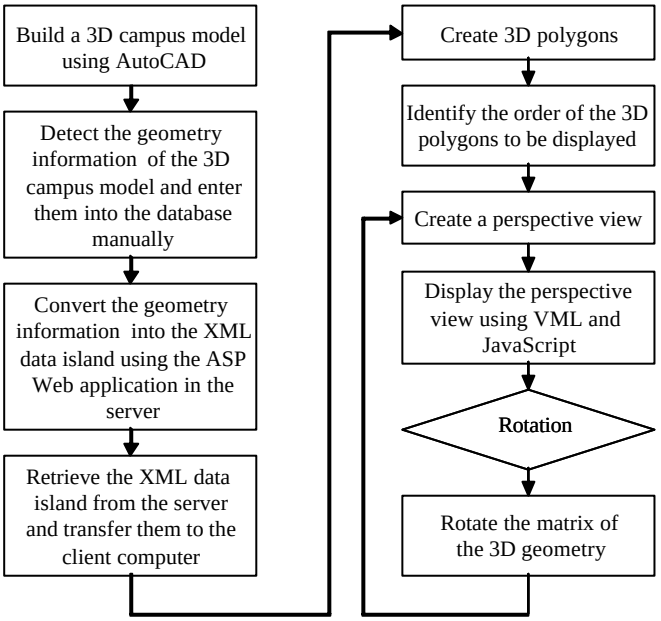


Figure 5. Algorithm of the 3D campus map

4. Application

We first built a 3D model of the campus using AutoCAD, and collected the coordinates of each and every vertex that builds up the campus buildings. After we entered the geometry data into the database, we convert it into the XML data island as shown in Figure 6 using the Web application we developed in this research.

```
<XML>
<SCALE>500</SCALE>
<POINTS>
<P><X>1623.27</X><Y>20</Y><Z>-583.53</Z></P>
<P><X>1623.27</X><Y>100</Y><Z>-583.53</Z></P>
<P><X>1632.45</X><Y>100</Y><Z>-583.53</Z></P>

... ... ... ...

<p><X>257.51</X><Y>80</Y><Z>-689.71</Z></P>
<P><X>195.05</X><Y>80</Y><Z>-689.71</Z></P>
<P><X>195.05</X><Y>0</Y><Z>-689.71</Z></P>
</POINTS>
<FACES>
<F id='47306'><V>0</V><V>1</V><V>2</V><V>3</V></F>
<F id='47301'><V>4</V><V>8</V><V>11</V><V>7</V></F>
<F id='47301'><V>5</V><V>9</V><V>8</V><V>4</V></F>

... ... ... ...

<f id='46859'><V>670</V><V>671</V><V>672</V><V>673</V></F>
<F id='46860'><V>674</V><V>675</V><V>676</V><V>677</V></F>
<F id='46861'><V>678</V><V>679</V><V>680</V><V>681</V></F>
</FACES>
</XML>
```

Figure 6. XML data island for the 3D campus map

Once the XML data island is created, the Web page that was developed in this research 1) downloaded the XML data island into the client computer; 2) created the 3D polygons using the XML data island; 3) created a perspective view; and 4) displayed it on the Web browser. Figure 7 illustrates the Web site that displays the 3D campus map of Texas A&M University. As shown in this figure, the 3D campus map provides functions for flying around, panning, and zooming the model.

5. Discussion

The elapsed time for displaying the vector graphics on the Web browser was not fast enough to attract the end users. It took about 1 minute 30 seconds to load the 3D campus map shown in Figure 6 on the Microsoft Internet Explorer 6.0 that was running on the Personal Computer with a 652 MHz Crusoe Process TM5800 and 240MB of RAM. Once the 3D campus map is loaded on the Web browser, it took about 5 seconds for each movement or rotation in the model.

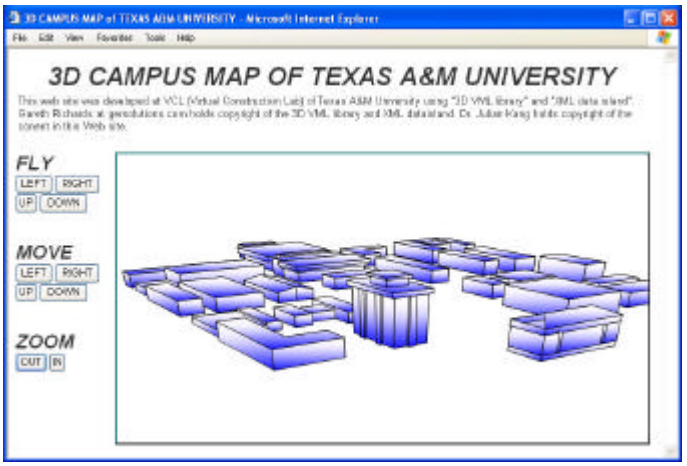


Figure 6. XML data island for the 3D campus map

The slow speed may be due to the JavaScript code used for handling the 3D vector graphics. When JavaScript converts the 3D vector information into 2D perspective view based on the viewer's location, it identifies surfaces that are hidden by other surfaces from a certain view point and displays them by the right order. In order to make this 3D campus map attractive to the end users, the process of converting the 3D vector graphics into the 2D perspective view should be implemented by a faster tool. Java applet is a compiled computer application that can communicate with JavaScript. It runs on top of Java Virtual Machine (JVM), and usually runs faster than the JavaScript code. Therefore, it may be possible for us to bring the 3D campus map on the Web browser and navigate around the map faster if we use a Java applet for converting the 3D graphics to 2D perspective view.

The prototype Web site introduced in this paper does not provide any links for inquiring additional building information. The XML data island that Gareth Richards suggested for the 3D VML library did not have any room but containing the geometry of the campus buildings and roads. We did not attempt to modify the syntax of the XML data island in order to include additional building information, because there was a possibility of getting 3D campus information directly from the database without going around the XML data island. We plan to test whether it would be feasible in our next development.

6. Conclusion

The prototype XML-based 3D campus map demonstrated a possibility of using VML for illustrating 3D vector graphics on the Web browser. It also

demonstrated that VML can be cooperatively used along with JavaScript and plain HTML, which implies a possibility of combining VML with Active Server Pages (ASP) and creating more dynamic environment in the 3D campus map. However, the prototype XML-based 3D campus map also demonstrated that the speed of loading the map as well as navigating around the buildings is not fast enough to attract the end users. Considering our literature review on techniques for combining JavaScript and Java applet, it is expected that speed problem will be fixed in the near future. In conclusion, although the prototype XML-based 3D campus map seems to have a long way to go toward the ideal 3D campus map, combining VML, ASP, JavaScript, and Java applet may eventually enable us to accomplish the goal.

Acknowledgements

The development of the prototype XML-based 3D campus map was carried out as part of a wider research project sponsored by the College Research and Interdisciplinary Council (CRIC) of the College of Architecture at Texas A&M University on 'An Empirical Study of the Merits of 4D Visualization in Web-based Collaborative Construction Scheduling'.

References

Lutz Tautenhahn: 2002, *SVG-VML-3D 1.0*, http://home.t-online.de/home/lutz.tautenhahn/svgvml3d.

Microsoft: 2003, *XML Data Islands*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk30/htm/xmconxmldataislands.asp.

Richards, G.: 2000, *3D Interactive VML*, http://www.gersolutions.com/vml.

Tufte, E.R.: 1990, *Envisioning Information*, Graphics Press, Cheshire.

W3C: 2003, *Extensible Markup Language (XML)*, http://www.w3.org/XML.

W3C: 1998, *Vector Markup Language (VML)*, http://www.w3.org/TR/NOTE-VML.

W3C: 2001, *Scalable Vector Graphics (SVG) 1.0 Specification*, http://www.w3.org/TR/SVG.