

Three-dimensional terrain visualisation on the Internet using VRML, Java, and JavaScript

Dusan Dacic¹ and Dejan Rancic²

Abstract – This paper presents technologies applied to design of three-dimensional GIS systems on the Internet. VRML descriptive language is described as a suitable platform for system development and possibilities of procedural programming using Java and JavaScript are also presented. At conclusion, we present Web 3D Modeler, three-dimensional terrain visualization system on the Internet

Keywords – terrain visualization on the Internet, VRML, Java, JavaScript

I. INTRODUCTION

The complexity of modern Geographical Informational Systems (GIS), and tasks that they are meant to perform, demand integration of three-dimensional spatial information, semantics, and all the necessary tools to work with and analyze relationships between them. Currently, existing systems lack either three-dimensional analysis capabilities, as do the majority of commercial GIS systems, or are missing tools for analysis of information related to spatial objects, like CAD (Computer-Aided Design) software [1]. There are still only a few systems that have the capabilities to process 3D spatial queries that seem simple enough: “mark all landscape areas that are not covered by radar network”, or “mark all available apartments in Central Street having windows NOT looking at the newest discotheque”. For a while, it’s justifiable to develop systems with above described capabilities. Remote access to these systems over the Internet is a challenge yet to meet. There are three basic problems to be solved when designing this functionality, data structuring on the server side, ways to organize communication and queries between client and server side, and query result visualization on the client computer.

World Wide Web (WWW) is based on standard data formats, such as HTML and XML, JPEG and GIF image formats, and GML for 2D vector graphics images. These standards enable the use of “thin” client-server model in heterogeneous environment that the Web is.

The goal of this paper is to present technologies and standards that are setting direction for the development of three-dimensional GIS systems on the Internet, having very low development cost, multiplatform support, that are very easy to maintain and expand, and intrinsically enable easy integration into other systems.

Dusan Dacic is with the Electronic Faculty Nis, Beogradska 14, 18000 Nis, Serbia e-mail: dusandacic@elfak.ni.ac.yu

Dejan Rancic is with the Electronic Faculty Nis, Beogradska 14, 18000 Nis, Serbia e-mail: ranca@elfak.ni.ac.yu

The use of VRML standard in development of 3D client-server GIS systems is not an unknown idea, VRML exists since year 1994 [2], and it is no longer considered a new and immature standard, but until recently the widespread use of this standard was limited by narrow bandwidths, and also by the cost of powerful 3D graphics subsystems that could enable real-time rendering of VRML’s three-dimensional objects. With lower prices of graphics hardware, and constant increase of average bandwidth, we now have the possibility to construct 3D GIS systems for wide use, lowering the price for end-users, and thus increasing user population and justifying development effort.

II. VRML

VRML [2] stands for *Virtual Realty Modeling Language*. Similarly as HTML and XML are used to describe data with predefined fields, that describe structure and attributes, VRML has predefined nodes and statements that describe structure, appearance and interaction in 3D environment. Unlike existing 3D graphics languages and libraries, that are procedural, like OpenGL and DirectX, VRML is object-oriented descriptive language, and much more. VRML is neither typical Virtual Reality language, neither real modeling format. It is common that VR systems include complicated display systems and HUD displays, state-of-the art non-standard input devices such as gloves, movement tracking sensors, voice interaction systems, and VRML has standard support only for mouse interaction, and standard displays. For CAD modeling purposes, VRML has only minimal capabilities.

Many authors consider VRML to be the 3D analogue for HTML. This actually means that VRML has become a standard media for publishing 3D multimedia data on the Internet. VRML supports different multimedia formats for sound, video and animation.

By definition, VRML is file format for integration of 3D graphics and multimedia, an international ISO/IEC-14777 standard that offers efficient way to exchange 3D data. Many modern modeling and programming tools have support for import/export for VRML. Good examples are 3D Studio Max by Discreet, World ToolKit by Sense8 and Java 3D [3].

VRML Consortia is a community gathered around defining an open standard for VRML. This consortium defines VRML as an “Open standard for three-dimensional multimedia on the Internet”. Open standard means that VRML is publicly available, like Java.

To view VRML content, one must use plug-ins for Internet browsers, or stand-alone software that enable loading and rendering of VRML files. Most used plug-ins are Cortona VRML Client by Parallel Graphics [4], Cosmo Player by

Computer Associates [5], and Blaxxun Contact by Blaxxun interactive [6] (Fig 1.)

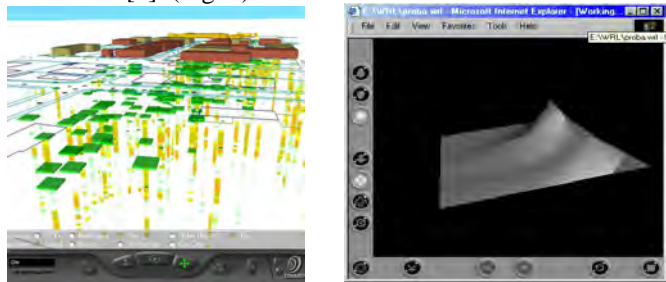
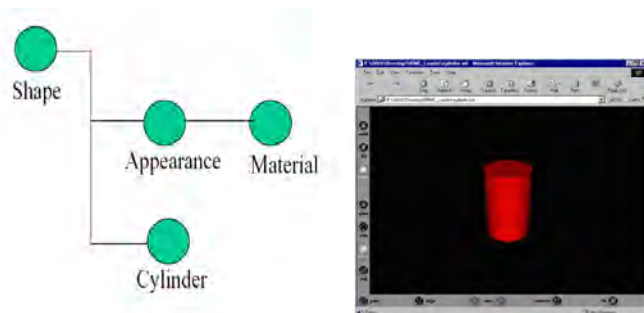


Fig. 1. VRML browsers

III. VRML PROGRAMMING

VRML is very useful for developing service for interactive three-dimensional visualization on the Internet. VRML describes 3D object by using scene graph structure. Scene graph is hierarchical tree-like structure that describes the entire 3D scene, including geometrical object representation, object attributes and relations between objects [2]. Scene graph makes it easy to build complex objects composed of simple objects. Separate models and scenes could be easily combined in order to build complex virtual environment. Each object is presented by a node or a sub-tree, and it can be only one entity or composition of spatially or logically related entities. Each object in the scene has its own unique name. Primitive objects can be enhanced by additional attributes as material, texture or some other attribute, by using attribute-defining nodes. Attributes defined in one node are applied to all children of that node.

An example of object attribute definition is given in Fig 2. The code models the scene graph.



```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 1 0 0 # R G B -comment
    }
  }
  geometry Cylinder {
    height 3
  }
}
```

Fig. 2. VRML code example

The given example shows that the graph scene is defined by object instantiation. *Shape* node contains a reference to an *Appearance* object that defines material from which the virtual object is built.

A VRML node can be compared with object or class in OO (Object Oriented) languages like C++ or Java. Objects are generated after the browser parses the VRML file. VRML has 54 nodes that cover almost every aspect of 3D modeling. In order to model terrain, *ElevationGrid* node is used together with *QuadLOD* node to model LOD functionality (Level Of Detail) as shown in Figs. 3 and 4.

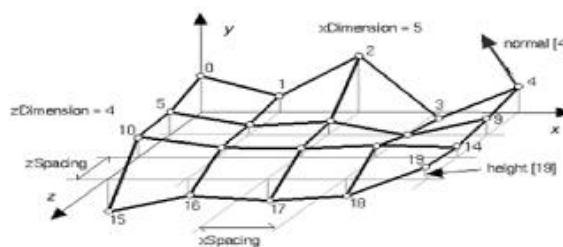


Fig. 3. *ElevationGrid* node model

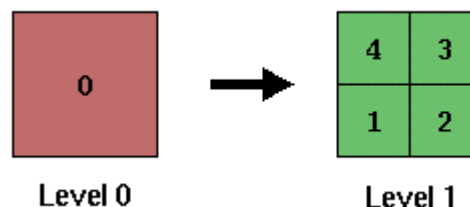


Fig. 4. *QuadLOD* node model

QuadLOD node is derived node type developed in order to enable implementation of level-of-detail algorithm. LOD algorithm enables the viewer to transfer and render only the detail that can be seen on the raster display, depending on the virtual viewpoint. One node can only contain data for two levels of detail, but using the quad-tree-like structure multiple level of detail can be achieved.

LOD algorithm should be able [1]:

1. To form a terrain view with varying detail level, based on viewpoint location and terrain configuration, meaning that areas with small or none height variation can be represented with fewer polygon count than areas with high level of variations, such as canyons, cliffs, and mountains.

2. To solve appearance of *cracks* in terrain rendering that appear due to resolution mismatch of neighboring quadratic areas at different levels of detail

3. To enable smooth and seamless transition between terrain tiles that present the same area but at different level of detail.

To achieve such functionality, it's not enough to use only VRML descriptive language; therefore, we looked for possibility of introducing procedural described behavior. VRML standard enables procedures to be defined through the use of External Authoring Interface (EAI) system, basically a framework of Java base classes that are to be sub-classed or

used directly, and when linked dynamically with VRML browser, used as a behavioral description of VRML objects and scenes. To access these classes, one could use Java or JavaScript programming languages.

VRML node used to link Java or JavaScript with VRML world is *Script* node. This node has input and output ports like almost every node in VRML. These ports or *events* are used to enable communication between VRML objects: it's possible, for example, for proximity sensor to switch the light on in a virtual environment when the virtual viewpoint reaches the certain threshold. The same logic is used to incorporate a *Script* node into virtual environment. Script has no body, but it can be used to receive events, pass them to Java code for processing, and pass generated events back to the virtual world.

Java code can also be used to access *Factory* engine of the VRML browser that has the ability to introduce new objects into the VRML scene.

JavaScript can be used for simple world scripting, and for processing user input from an environment external to a VRML world, such as a web page that has VRML world embedded in it. The best thing about JavaScript is that it can be coded directly into the VRML scene or HTML environment that surrounds the VRML.

In Fig. 5 is shown a snippet of JavaScript code that is used to model a simple behavior together with a VRML code that links it with a scene.

```

DEF MoveObject Script {
  eventIn SFFloat set_timeFraction
  eventOut SFVec3f location
  url "javascript:
  function set_timeFraction (time_fraction, tm)
  {
    location[0] = 0.0;
    location[1] = 4.9*(1-time_fraction);
    location[2] = 0.0;
  }"
}
DEF Timer TimeSensor {
  loop TRUE
}
ROUTE Timer.fraction_changed TO
MoveObject.set_timeFraction
ROUTE MoveObject.location_changed TO
MovingObject.set_translation

```

Fig. 5. Script sample

Java stands out as a complicated but powerful medium for programming VRML objects behavior. It enables a powerful programming environment, and executes a lot faster than a JavaScript code. Java has advanced network communication abilities, and can be used efficiently to model user interface into a VRML world, that intrinsically has no interface but one that enables us to move in virtual space.

IV. GINIS WEB 3D MODELER

GINIS Web 3D Modeler is a Web application developed at *Geographic Informational Systems and Computer Graphics Laboratory* with *Electronic Faculty* in Nis. It is an experimental application for testing the possibilities of presenting virtual world on the Internet containing GIS informative capabilities.

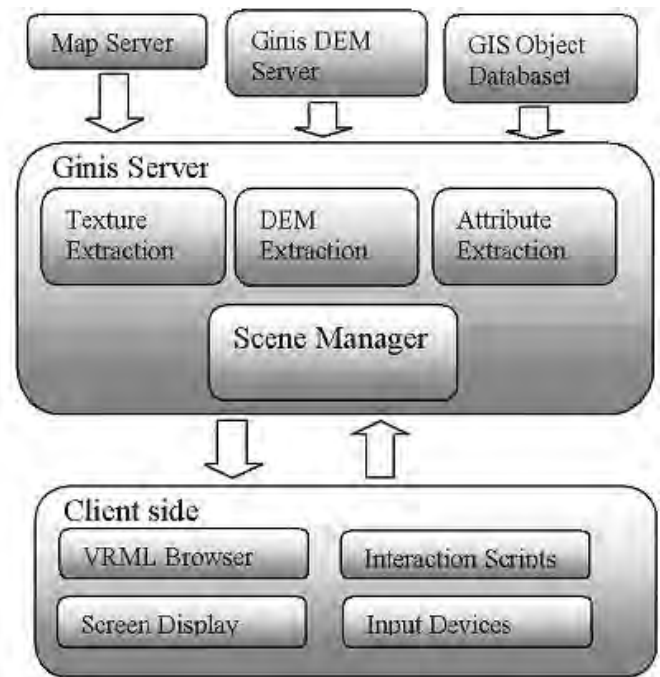


Fig. 6. GINIS Web 3D Modeler

Conceptual model of the system is shown on Fig 6. By using VRML methodology, all of the actual 3D graphics rendering process is done solely on the client side. Interaction with the user is done inside the virtual environment, or through the scripts embedded in the web page that has access inside the virtual world. To access virtual environment, user selects geographic area of a special interest, and gets directed to a page with 3D rendering (Figs. 7 and 8)

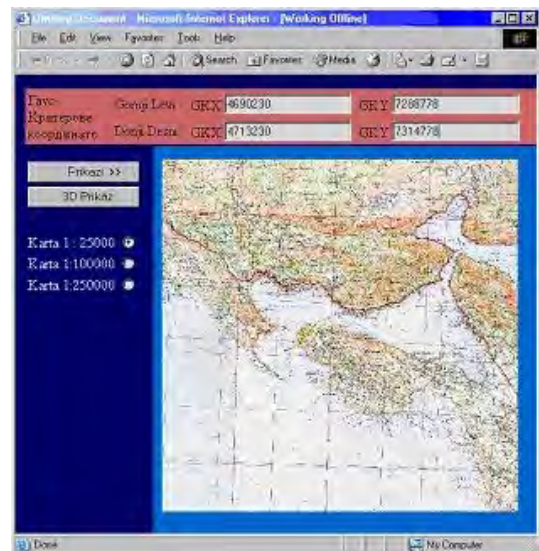


Fig. 7. Selection of special interest area

When the user approaches some part of the terrain, the tile used for that area is being replaced with a tile that has a higher resolution per measure unit (Figs. 8 and 9).

V. CONCLUSION

In the paper, we tried to explain the ways of developing a system that has all the benefits of the two worlds: organized geographical and georeferenced information capability of a full-featured GIS system, and an easy-to-use three-dimensional presentation of GIS information. There are numerous applications of such system. System like this can be used for a public-place automatic-information desk, or with a recent increase in small mobile devices performance, it could be easily executed on any mobile device that has a ported version of VRML browser. Cortona has developed a Pocket Cortona Client for PocketPC devices [4], and it is expected for the other companies in the business to do the same.

Java and JavaScript programming languages enable us to expand existing capabilities of VRML, and are powerful tools for design of user interaction system and performance optimization algorithms.

Ginis Web 3D Modeler is a step toward designing a truly powerful interface to existing GIS datasets, available to larger population of users.

VRML is an open standard, but recently, X3D as an XML-compatible standard has gained its position in a WWW community [7]. X3D is by all concepts similar to VRML, and also has same basic nodes with same structuring rules, and it is an easy task to convert existing VRML-based systems in a way to support X3D, only disadvantage of X3D is the slow response of VRML browser manufacturers. Implementation of X3D should be the first step in developing this system as a full-featured Web service.

ACKNOWLEDGEMENT

The research was partially supported by the project "Geographical Information System Designed to Improve the Local Municipal Function based on Internet/WWW Technologies", funded by Ministry of Science, Technology and Development, Republic of Serbia and Municipality of Niš, Contract No. IT.1.23.0249A.

REFERENCES

- [1] Zlatanova, S., A. Rahman, M. Pilouk, "3D GIS: current status and perspectives", Proceedings of ISPRS, 8-12 July, Ottawa, Canada, 2002
- [2] Rikk Carey and Gavin Bell, "The annotated VRML 97 reference". URL: <http://www.best.com/rikk/Book>, 1999
- [3] Coors, V. and V. Jung, "Using VRML as an Interface to the 3D Data Warehouse", Proceedings of VRML'98, New York
- [4] PARALLELGRAPHICS,2003 url:<http://www.parallelgraphics.com/products/>
- [5] COSMOSOFTVARE,2003 url:<http://www.cosmosoftware.com>
- [6] BLAXXUN,2004 url: <http://www.blaxxun.com>
- [7] WEB3D,2004: <http://www.web3d.org/x3d/>

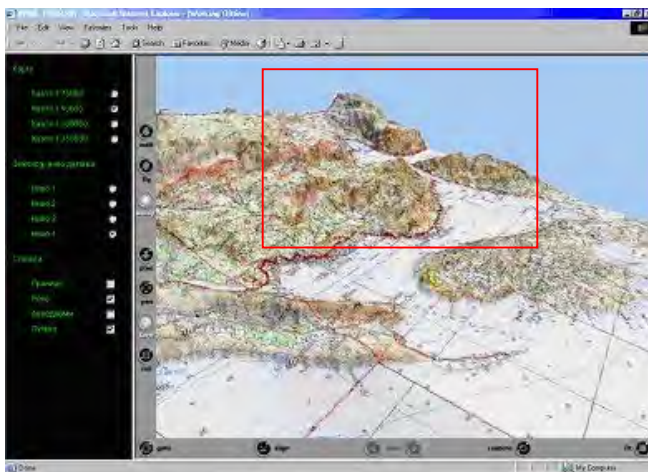


Fig. 8. Terrain viewed from a distance

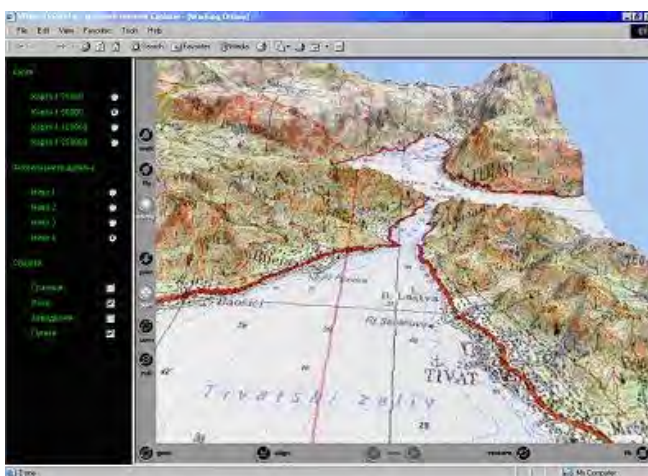


Fig. 9. Close-up look with a different LOD

Table I shows average measured frame rates with LOD functionality on and off, and also with static or dynamic viewpoint. Measured terrain sample had 60000 height points, and was textured with 1024x1024 24-bit bitmap. Test configuration was Athlon 1GHz CPU, 512 MB DDR RAM, GeForce 2 Graphics card, Windows 2000 OS, MS Internet Explorer 6.0, Cortona VRML browser.

It can be observed that we gained an average of 10 frames per second, which can be considered a small gain by static camera, but makes almost 50 percent gain with dynamic camera, and together with optimized bandwidth requests justifies the introduction of LOD algorithms.

TABLE I
MEASURED AVERAGE FRAME RATES

	Static camera	Dynamic camera
LOD switched OFF	80	23
LOD switched ON	88	34