FractVRML: A New Proposition to Extend X3D with Fractal Geometry

Javad Sadeghi, Mohsen Sharifi
Computer Engineering Department,
Iran University of Science and
Technology, Tehran, Iran
{javad | msharifi}@iust.ac.ir

ABSTRACT

Fractal geometry is a branch of mathematics that has found applications in different sciences. Natural phenomena that are not specifiable with thousands of mathematical formulae can be modeled easily with fractals. Virtual reality uses interactive simulation of real world to simulate a sense of presence in a real or imaginable environment. Because of increasing demand for virtual worlds, and the lack of a fast way in visualization of natural phenomena, we have studied the applications of fractal geometry in virtual reality and its 3D modeling language (VRML). We found that large natural worlds modeled in VRML can take a very long time to download and render.

Web3D Consortium has introduced X3D as the next generation of VRML. This paper, proposes a new profile, nicknamed *FractVRML*, for easily modeling natural phenomena with fractals. This profile adds more than 16 new nodes to X3D using extensibility mechanism of X3D. It also adds support for complex coordinates using quaternions. In addition to high resolution, natural scenes designed with FractVRML have lower volume. FractVRML provides more natural representation of objects, as well as making noticeable improvements in the speed of loading and navigation of scenes in the web.

Keywords

Fractal, 3D Visualization, Virtual Reality, VRML, GeoVRML, X3D.

Archive of SID

1. INTRODUCTION

Have you ever seen a cloud or a tree? If positive, then you have seen a fractal. Fractals are all around us in the natural world. Plants, clouds, mountains, rivers, lightning, trees and ferns are some examples of natural fractals.

What exactly is a fractal? A fractal is a rough or fragmented geometric shape that can be subdivided in parts, each of which is (at least approximately) a reduced-size copy of the whole. Fractals are generally self-similar and independent of scale [1]. Benoit B. Mandelbrot [2] gives a mathematical definition of a fractal as a set for which the Hausdorff-Besicovich dimension strictly exceeds the topological dimension.

Topological dimension is the “normal” idea of dimension and Hausdorff-Besicovich dimension can be calculated by taking the limit of the quotient of the log change in object size and the log change in measurement scale, as the measurement scale approaches zero [3].

Fractals are geometric objects that have found many applications in different sciences [4] such as physics, chemistry, geography, aerology, agriculture, etc. as well as in the engineering of data and information in information technology field.

WWW services have been in use in the internet since 1991, and there have been increasing demands for more and better quality services. For example, people have required more natural and interactive services. In 1994, a community of computer graphics professionals has responded to this demand by extending the web’s text-centric feature set to include interactive and animated 3D virtual worlds. The vehicle for their work has been VRML, a *Virtual Reality Modeling Language*.

As a text language, VRML lets you to quickly build virtual worlds incorporating 3D shapes, light sources, fog, animation, and even sound effects. Each virtual world is described by one or more VRML world files named with a *.wrl* filename extension (short for “*world*”) and delivered across the web with the “*model/world*” Multipurpose Internet Mail Extensions (MIME) type. To display a VRML world from the web or off your hard disk, you need a VRML browser; typically configured as a plug-in for a web browser [5].

The Web3D Consortium is the only non-profit organization with a mandate to develop and promote open standards to enable 3D web and broadcast applications. X3D (“*Extensible 3D*”) is the next-generation open standard for 3D on the web. It is an extensible standard that can easily be supported by

www.SID.ir

content tools, proprietary browsers, and other 3D applications, both for importing and exporting. It replaces VRML, but also provides compatibility with existing VRML content and browsers. X3D is VRML97 broken down into *components*, with a mechanism to add new components to extend beyond VRML97 functionality. A *profile* is a collection of components covering several different areas of functionality. X3D includes: an *interchange profile* for exchanging X3D content among authoring and publishing systems; an *interactive profile* to support the delivery of lightweight interactive animations; an *extensibility profile* to enable the development of add-on components and robust applications; and a *VRML97 profile* to ensure interoperability between X3D and VRML97 content [7].

Motivated by the increasing demand for virtual worlds and the extensibility of the X3D, we have started our studies on the benefits of using fractals in VRML and X3D.

In this paper we discuss the application of fractals in virtual reality. An *X3D profile,* nicknamed *FractVRML*, is then proposed to meet the requirements for visualizing natural fractals easily. Given the strengths of fractal geometry in modeling natural phenomena [4], we are quite confident that the deployment of fractal formulas in the generation of virtual worlds provides a more natural view of the worlds' phenomena, and takes less space on the computer for this purpose.

Section 2 gives an outline of relevant attempts. Sections 3 and 4 describe the architecture and design of our proposed X3D profile, respectively. Finally, section 5 presents an evaluation of this profile and outlines directions for improvement.

2. PREVIOUS WORKS

For the past two years, we have been working on VRML to see how it can support natural visualizations. We found that VRML has some limitations for this kind of support; so we were directed to X3D. Our previous researches were centered on:

- The methods of visualizing fractals in VRML, and
- Extending VRML standard for better support of fractals.

This section briefly describes some previous attempts to produce "*VRML Fractals*" alongside their pros and cons. It then describes the technique of adding extensions to VRML, and briefs the *GeoVRML* standard (a successful extension to VRML) that we have used its structure in our extension to X3D.

2.1 Visualizing fractals in VRML

Computer graphics has greatly aided the investigation of the dynamics of iterative functions. The most common fractal forms we are familiar with today are two-dimensional. Several reasons support this claim. For example, many interesting fractals are subsets of, and are naturally visualized by plots on the complex plane [8]. Standard 2D frame buffer techniques have provided sufficient visual information about the structures, since most of our research has concentrated on the dynamics of complex variables. A matrix of color values can specify a 2D computer image, but 3D objects are harder to describe. However, recent investigations into higher-dimensional dynamical systems have shown the need for 3D visualization tools that will give researchers a better understanding of these objects. Furthermore, 3D representations are best appreciated when the point of view can be changed, and the computation of fractal sets is usually expensive enough without having to add the cost of real-time 3D rendering [9].

Although there is no defined way to visualize objects within seamless details on the web, two solutions for visualization of fractals in VRML has been proposed:

- Visualizing an estimation of the fractal, and
- Representing different levels of detail, depending on the user's position.

The first method creates shapes that should be called '*pseudofractals*' rather than "fractals" because they look like fractals from a distance, but up close they clearly have a finite number of elements. For example, Leemon Baird [10] has visualized some kinds of these fractals using JavaScript and in limited iterations of recursive *PROTO* nodes. Figure 1 shows one of these shapes in IE browser using Cortona.

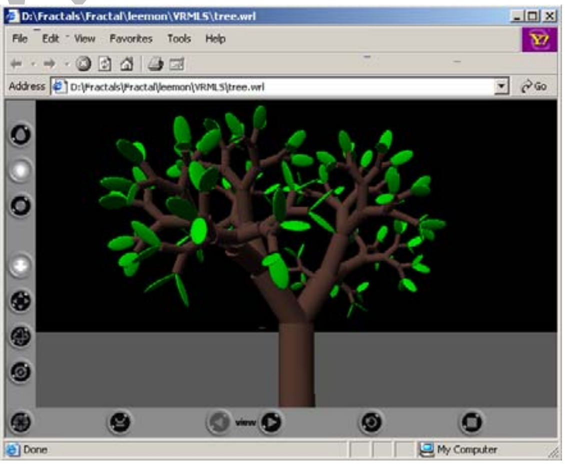

Figure 1: A pseudo fractal

The second method can create *'true fractals'* but they are rendered very slowly. True fractals have infinite number of components, though some may not be visible until you get very close to them. A good approach to increase the speed of navigation of these scenes is using LOD node to show different alternative details, depending on the viewer's position. An example of efforts on true VRML fractals is San Diego Super Computer Center's plant fractals project that uses L-systems in generation of fractals. Figure 2 shows the 3D Mandelbrot set that has been created with java. True fractals can be written with an applet (class files could be embedded in *script* nodes). Due to JIT compiler, java applets

Archive of SID

are more than 20 times faster than JavaScript. We need a 3D engine to visualize 3D objects. Some of content creators prefer to use Java3D API specification instead of VRML so there is a competition between Java3D and VRML until a 3D engine is included in browsers. This is now in progress and we will refer to it later in the paper when discussing Xj3D.

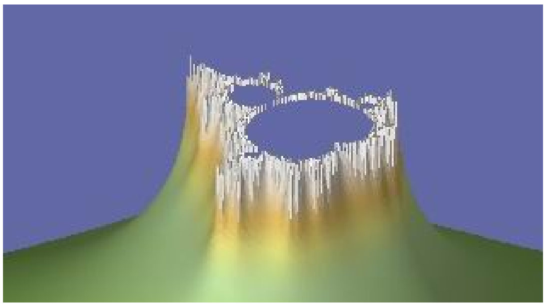

Figure 2: 3D Mandelbrot fractal

2.2 Extensions to VRML

In the previous section we described how one can extend VRML by authoring custom interpolators and sensors in java or JavaScript within a *script* node. Also a VRML *PROTO* declares a new node type that encapsulates an arbitrary set of other VRML nodes, scripts, and animation circuits. Using a PROTO, one can extend the language, encapsulating and hiding details in much the same way that classes are used in programming languages like java [5].

We wanted to develop a suitable platform that, in addition to the ease and accurate visualization of fractals, could improve some weaknesses of VRML standard using fractal geometry; especially in supporting complex coordinate system. Our research was also aimed at adding some new nodes to VRML, so we searched for a successful extension to VRML and found GeoVRML standard [12].

2.2.1 GeoVRML standard

GeoVRML is an extension to the VRML standard that provides geoscientists with a rich suite of enabling capabilities that cannot be found elsewhere as a web browser plug-in. That is the ability to model dynamic 3D geographic data that can be distributed over the web and interactively visualized using a standard browser configuration. GeoVRML includes ten new extensions, or nodes, that sit on top of VRML97. These nodes are defined using VRML’s EXTERNPROTO extensibility features. In addition, it addresses issues such as: different geographic coordinate systems, data fusion, high precision, dataset scalability, metadata linking and animation support and provides some tools for geographic applications [11]. All these facilities provide geoscientists with an excellent medium to present complex 3D geographic data in a dynamic, interactive, and web accessible format.

GeoVRML working group was formed on 27 Feb 1998 in Web3D Consortium and has a mailing list where discussions and developments are posted. Currently, this list consists of over 200 members. GeoVRML is an open standard and its specification is published openly with a source-level sample implementation [12]. We have used this standard structure in our work because there are similarities between FractVRML and GeoVRML and it is a successful extension that now is added to X3D as a profile (the goal that we have set for ourselves).

2.3 X3D features for extensibility

X3D is a software system for defining interactive, animated 3D graphics integrated with other rich media such as hypertext, audio and video. It is intended to be a universal interchange format for integrated 3D graphics and multimedia. X3D improves upon VRML with new features, advanced application programmer interfaces, additional data encoding formats, stricter conformance, and componentized architecture that allows for a modular approach to supporting the standard [7].

VRML has only the EXTERNPROTO mechanism for extensibility, but no real mechanism for creating groups of functionality extensions. X3D's component, level, and profile mechanisms allow for this. Groups of features are encapsulated in what are called "components". A component is usually specific for one particular area of functionality. Any X3D component may designate a level of service by using a numbering scheme in which higher-numbered levels denote increasing qualities of service.

A profile is a grouping of components covering several different areas of functionality. A profile can even contain the functionality of several profiles. We briefly introduced the X3D profiles in the introduction section of the paper, so we will avoid its reiteration here. A profile is a subset of the X3D specification designed to meet a particular application, platform or market need. Each profile represents a set of features that are commonly used together to meet such a need. A profile description may specify restrictions on certain features in order to simplify implementation or reduce working set, such as imposing array size limitations or designing unsupported fields within a node. Profiles are defined as a set of components and levels of each component as well as the minimum support criteria for all of the objects contained within that set. Browsers may be considered conformant to a particular profile even if they do not implement the entire X3D specification.

3. PROPOSED EXTENSIONS TO X3D

In our search for an appropriate way of visualizing fractals in virtual reality, X3D proved to be a more suitable environment for our proposition, than VRML. So it was selected for our purpose.

Before describing our proposed extension to X3D, let us first outline some points about 3D computer graphics and computational geometry, which are required for the generation of 3D fractals.

Archive of SID

3.1 3D fractals [1,8]

Computer graphics programmers and designers are used to deal with a Cartesian (x, y, z) coordinate system for representing all 3D models. However, we need to use complex coordinates for representation of fractals. A common way to represent 3D fractals is to compute Julia sets with quaternions instead of complex numbers. The resulting Julia set is four dimensional. By taking a slice through the 4D Julia set (e.g. by fixing one of the coordinates), a 3D object is obtained. This object can then be displayed using computer graphics techniques such as ray tracing.

Instead of quaternions, one can of course use other functions. For instance, one could use a map with more than one parameter, which would generate a higher-dimensional fractal. Another way of generating 3D fractals is to use 3D iterated function systems (IFS). These are analogous to 2D IFS, except that they generate points in a 3D space.

A third way of generating 3D fractals is to take a 2D fractal such as the Mandelbrot set, and convert the pixel values to heights to generate a 3D Mandelbrot mountain (Figure 2). This 3D object can then be rendered with normal computer graphics techniques.

In our research, we have used quaternions in visualization of fractals because of their powerful features. A quaternion value q is a four-tuple consisting of one real part and three imaginaries:

$$q = q_l + q_i \boldsymbol{i} + q_j \boldsymbol{j} + q_k \boldsymbol{k}, \quad (1)$$

where $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are imaginary units,

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = -1, \quad (2)$$

Algebraic operations can be defined in the quaternions by treating the quaternion values as polynomials of three variables i, j, k. For example, the coefficients of the sum of two quaternion values may be found by adding their corresponding coefficients.

Quaternion multiplication is also similar to polynomial multiplication but with the special cases

$$\boldsymbol{ij} = \boldsymbol{k};\ \boldsymbol{jk} = \boldsymbol{i};\ \boldsymbol{ki} = \boldsymbol{j}, \quad (3)$$

and

$$\boldsymbol{ji} = -\boldsymbol{k};\ \boldsymbol{kj} = -\boldsymbol{i};\ \boldsymbol{ik} = -\boldsymbol{j}, \quad (4)$$

revealing an unfortunate side effect of the quaternions: *non-commutative multiplication*.

3.2 Proposed Architecture

All FractVRML nodes that contain complex coordinate have a *fractSystem* field. This field determines the method of visualization in complex plane. For example two common methods of projection of fractals are described below.

Escape-Time Landscapes [8]: This type of 3D projection simply adds a third dimension (height) to the complex plane. The height at each point is determined by how slowly iterations of the function at that point escape to infinity. This value is approximated by counting the number of the first iteration of a point whose modulus is greater than 2, a common technique.

Since it would be impossible to test every point in an area of the complex plane, the escape times of each member of a grid of evenly spaced points are tested. Surfaces in between grid points are approximated by triangular facets with the grid points as vertices.

Roving Riemann Spheres [8]: The Riemann Sphere is a unit sphere placed on the complex plane, such that its south pole touches the origin. Its north pole is the infinity point, and any point on the sphere except the infinity point can be mapped to a unique point on the complex plane: the point on the complex plane which intersects the line passing through the infinity point and the other point on the sphere. One could imagine projecting an image in the complex plane onto a unit sphere by mapping each point to the corresponding point on a Riemann sphere. Such a projection would yield a "fish-eye" view of the complex plane.

The Roving Riemann sphere projection generalizes the previous projection in two ways: both the position of the South Pole on the complex plane and the radius of the sphere are allowed to vary. By reducing the radius of the sphere and moving the sphere so that the South Pole rests on a different point on the complex plane, the projection can zoom in to an area of interest.

As in the previous projection method, all points in an area cannot be tested. Instead, the surface of a sphere is approximated with triangular facets and the escape times at the vertices are tested.

X3D uses Cartesian coordinate for representing 3D scenes. So FractVRML nodes, that use complex coordinate, must convert their coordinate to Cartesian for representation on a browser. This is done using *FractOrigin* node.

3.2.1 FractOrigin

This node supplies support of complex coordinates with quaternion. When it is used, it translates complex coordinates to Cartesian coordinates. By calculating the difference between each coordinate and the FractOrigin, an offset is produced that can be used for faithful rendering.

3.2.2 FractCoordinate

This node is used to keep a list of complex coordinates that can be used in standard X3D geometry nodes such as *IndexedFaceSet*, *IndexedLineSet*, or *PointSet*, to specify coordinates.

3.2.3 FractLOD

This node automatically manages loading of higher resolution data as the user approaches to an object and unloads it after user passes it. Here, using quadtree, we have created a hierarchical structure

Archive of SID

that when the viewer enters the specified range, geometry is replaced with the contents of the four children files, and when the user leaves this range, geometry is unloaded from memory.

3.2.4 *FractPositionInterpolator*

This node provides an interpolator capability where key values are specified in complex coordinates and the interpolation is performed within the specified spatial reference frame.

3.2.5 *FractTouchSensor*

This node tracks the location and state of a pointing device and detects when the user points at geometry contained by the parent group of the FractTouchSensor.

3.2.6 *FractViewpoint*

This node allows the specification of a viewpoint in a complex coordinate. It can be used wherever a viewpoint node can be used. It can be combined with viewpoint node too.

3.2.7 *FractElevationGrid*

Many data visualization techniques produce data that can be rendered as terrain maps. The *ElevationGrid* node of VRML can be used for this purpose. The FractElevationGrid node specifies a uniform grid of elevation values within a complex spatial reference frame.

A hierarchical view of these basic nodes is shown in the form of a UML class diagram in Figure 3.

4. FRACTVRML SPECIFICATION

In the past section, we briefly introduced a *complex component* that fulfills our needs in the preparation of a suitable platform. The next part of our profile consists of some fractal nodes that will generate natural fractals. The most famous fact in fractal science is that: “complete structure of natural systems, is mostly reflected in each part of it” and the second fact is that: “a system is self-similar when similar energies act in many levels of scale” [14]. Given these facts, we have proposed a set of fractal nodes for visualization of fractals in X3D.

4.1 Fractal nodes

Complex polynomials are used to draw the well known images of Julia and Mandelbrot sets. The first researches on iteration of complex polynomials were carried out by Julia (1918) and Fatou (1919-1920) [3]. Let us consider the simple polynomial z^2+c

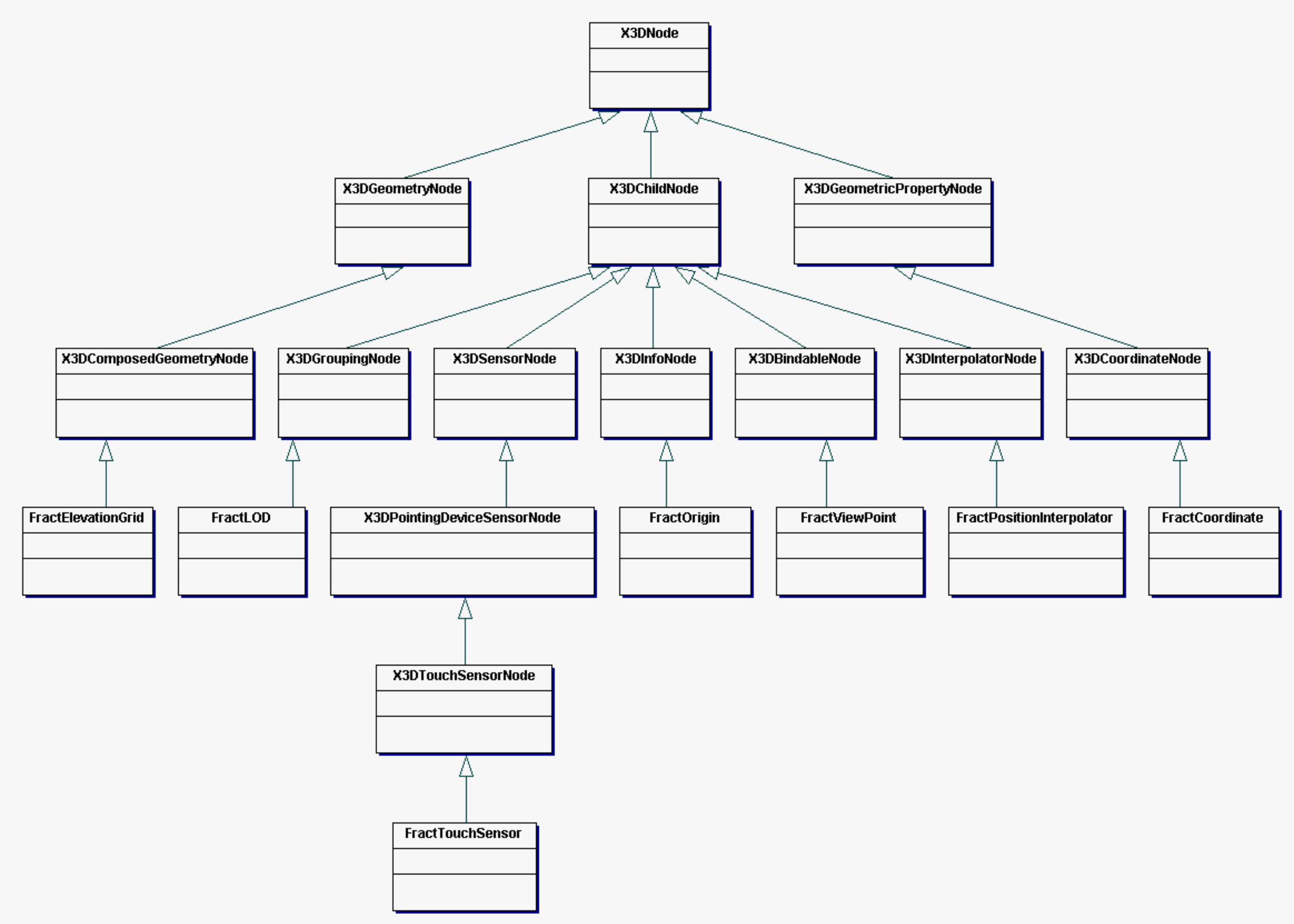

Figure 3: Relationship between nodes in the proposed architecture

www.SID.ir

where c is an arbitrary constant and z is a variable. Moreover, z and c are complex numbers. Now we write $z_{(n+1)}=z^2_{(n)}+c$ and start, for example, with z=0. This means that the first value of the polynomial (at the first iteration, i.e. when n=1) is c. Now, for the second iteration (n=2), the value of z is c; the result of the first iteration and the polynomial has now value c^2+c. The polynomial will be iterated endlessly, giving to z, at each time, the value of the polynomial computed in the previous iteration. The above mentioned way is a sample iterative method of creation of fractals. We have used similar formulas for creation of new nodes that support natural fractals. Figure 4 shows the relation between these nodes, which are detailed in the following.

4.1.1 FractMountain

Usually by a method such as taking a triangle, dividing it into 3 sub-triangles, and perturbing the center point, a fractal mountain is generated. This process is then repeated on the sub-triangles. This results in a 2D table of heights, which can then be rendered as a 3D image. This is referred to as *midpoint displacement* [13].

4.1.2 FractRiver

Mandelbrot introduced *squig curves* as “a model of a river’s course, patterned after the well known pictures in geology or geography that show the successive stages of a river that burrows into a valley, defining its course with increasing precision”. The production predecessor is a triangle with the edges labeled entry, exit, and neutral. The entry edge represents the set of possible sites where the curve may enter this triangle, and the exit edge represents the set of possible sites where the curve may leave it. The neutral edge is not intersected by the curve. The squig curve can be constructed in a manner analogous to the deterministic implementation of the midpoint displacement method [13].

4.1.3 FractIsland

Fractal island generation is a combination of fractal mountain and fractal river generation techniques. Here in each iteration, the water way and shape of mountain are determined. By changing the *seaLevel* field, we can control height of sea water for generation of different islands

4.1.4 FractCloud

Plasma clouds are similar to fractal mountains. Instead of a 2D table of heights, the result is a 2D table of intensities. They are formed by repeatedly subdividing squares. Because of chaotic feature of clouds, zooming on them will discover more complex structures. Cloud structures affect weather

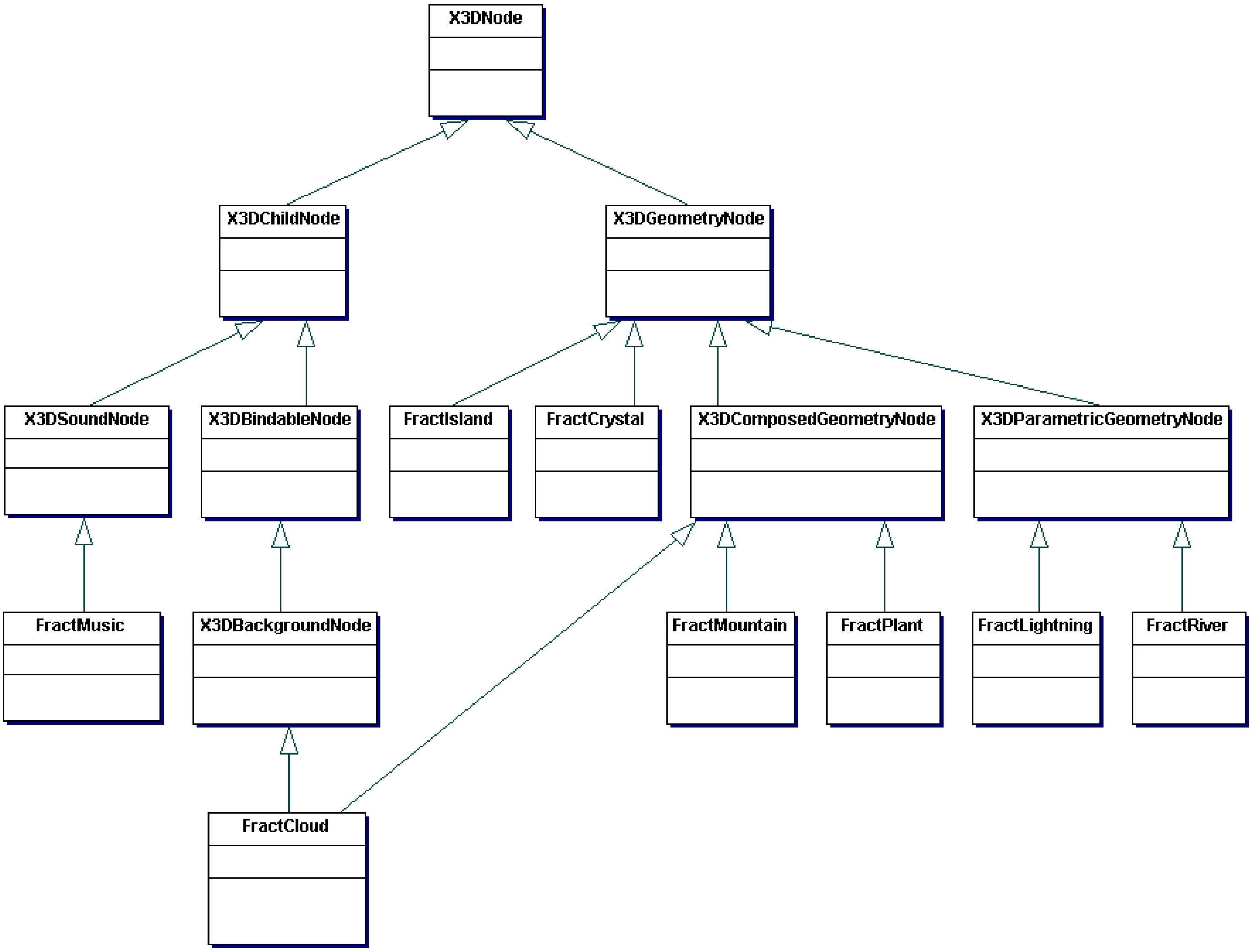

Figure 4: Relationship between fractal nodes in the proposed architecture

www.SID.ir

conditions on the earth, so providing a good representation of them in X3D will benefit us in aero logical studies.

4.1.5 FractPlant

We can create tree like shapes easily using fractal geometry, because plants' shapes are based on simple recursive units. Using suitable L-system algorithms we can produce simple L-system creation rules that create an especial class of plants. Ferns are one of the most ordinary and interesting forms created with IFS. Using varieties of fractal trees and fractal ferns we can generate fractal jungles.

4.1.6 FractLightning

This node generates natural fractal lightning. The power density generated by lightning does not have the smooth characteristics expected from a dipole model of lightning, but it has the structured form expected from a horizontal fractal antenna. It is important to realize that a lightning discharge must be horizontal, as in intra-cloud lightning, to project the energy upwards into the lower ionosphere. A vertical discharge, as in cloud-to-ground lightning, will radiate its energy horizontally as a vertical antenna. The tortuous, i.e. fractal, discharge path increases the effective dipole moment, since now the path length along the discharge is longer than the Euclidean distance. Therefore, a fractal discharge is expected to increase the radiated power density at local spots, hence reducing the required current and charge threshold required to produce sprites.

4.1.7 FractCrystal

This node generates fractal crystals. In solid state physics, generation of different models of crystals that are near to each other will help to predict their final state during their growth. Because of the chaotic nature of movement of molecules, they will be modeled best with fractals; e.g. brownian motion.

4.1.8 FractMusic

This node generates fractal music. This kind of music consequents fractal frequency of nature and has been found in rivers' sound and nature. We hear 1/f noise that is called flicker noise or pink noise everywhere. Also we can include other spatial sounds in *source* field of this node.

4.2 User interface

Architecture of X3D extensions has close relation with browsers and components that are accepted to new profiles will include in next browser profile. There are some X3D browsers but we chose Xj3D for our work. Xj3D is an open-source project that is dedicated to implement a collection of Java code for viewing and incorporating VRML content into applications of all forms [15]. One part of our work is to add support of new nodes we proposed to Xj3D standard. Xj3D project will remain compliant with the current VRML97 standard, and will evolve through X3D capability profiles to support the full VRML200x specification.

X3D-Edit is an authoring tool for Extensible 3D (X3D) Graphics scenes developed using IBM's *Xeena*, an XML-based tool-building application that enables simple error-free editing, authoring and validation of X3D or VRML scene-graph files. We have used this tool in our work too.

5. EVALUATION

In this paper we first introduced different aspects of applications of fractals in VRML and then introduced a new architecture for supporting fractal geometry in X3D. In summary, the following advantages and merits of FractVRML profile are in order:

- Support for complex coordinate system,
- 3D visualization of fractals, and
- Using XML tagset

The architecture of the FractVRML is designed so that the design of each internal section is completely separate from other sections. It consequently will separate implementation of each section and we have more freedom in choosing the best method of implementing each section. Our design is based on powerful mathematical methods of modeling and visualization of curves and objects.

Because of built-in support for natural phenomena, scenes designed with FractVRML have better resolution and more natural view, and take less space in comparison with other existing natural virtual worlds.

In discussions with some applied science experts they expressed their requirements for a high quality web-based and interactive system for their researches; the requirements which have directed our proposed design.

6. CONCLUSION

Fractals are an independent mental and art revolution. Fractals describe many real-world objects, such as clouds, mountains, coastlines, roots, branches of trees, blood vessels, and lungs of animals; perhaps the structure of the universe itself. They can also describe quite well natural phenomena such as percolation of liquids in porous materials (like soils), distribution in the time of river floods, earthquakes, brownian motion, etc.

We have presented a new profile, called *FractVRML*, for easily and efficiently modeling natural phenomena with fractals, as an extension to X3D. The set of capabilities proposed in our profile is both unique and novel, and it has three good features, namely, support for complex coordinate systems, 3D visualization of fractals, and use of XML tagset.

Related further research can be directed in the following areas:

- Implementation of the proposed profile and tracking of its standardization process.
- Application of FractVRML to GeoVRML (such as coastlines and weather prediction).

Archive of SID

- Design of a collaborative and distributed environment in support of real-time activities.
- Modeling of **Internet3D** with FractVRML.
- Support of XGL (an XML-based file format for representation of 3D information based upon the OpenGL rendering library) in FractVRML.

7. REFERENCES

[1] Noser H.R. et al.: *Physics-Enhanced L-Systems*. WSCG '01, 2001

[2] Mandelbrot, B. B. *The Fractal Geometry of Nature*, 2nd ed. Freeman, San Francisco, 1982.

[3] H. Peitgen, H. Jurgens, D. Saupe. *Fractals for the Classroom*, Springer Verlag, 1991.

[4] Barnsley, M. F. *Fractals Everywhere. Academic Press*, New York, 1988.

[5] David R. Nadeau. *Building Virtual Worlds with VRML*, IEEE Transaction on Computer Graphics and Applications, pp. 18-29, March/April 1999.

[6] The Virtual Reality Modeling Language. International Standard ISO/IEC 14772-1:1997. 1997.

[7] ISO/IEC 19775-2:200x, Extensible 3D (X3D). http://www.web3d.org/fs_specifications.htm

[8] Christian Cumbaa. *3-D Visualization of Julia Sets and other Fractals*, University of Waterloo, 1997.

[9] John C. Hart, Daniel J. Sandin, Louis H. Kauffman. *Ray Tracing Deterministic 3-D Fractals*, ACM Transaction on Computer Graphics, Volume 23, Number 3, July 1989, pp. 289-296.

[10] Leemon Baird’s Home Page. http://www.leemon.com , 2003.

[11] M. Reddy, L. Iverson, and Y. Leclerc. *GeoVRML 1.0: Adding Geographic Support to VRML*. GeoInformatics, Volume 3, September 2000.

[12] GeoVRML Working Group. http://www.geovrml.org , 2003.

[13] Przemyslaw Prusinkiewicz and Mark Hammel. *A Fractal Model of Mountains with Rivers*, Proceeding of Graphics Interface 93, pages 174-180, May 1993.

[14] Dick Oliver, Daniel Hoviss. *Fractal Graphics for Windows*, SAMS Publishing, 1994.

[15] Xj3D Open Source Toolkit. http://www.xj3d.org, 2003.

Archive of SID