Next Steps for X3D Geospatial Specification

Chris Thorne
Ping Interactive Broadband
October 21 2007

Aim of document

This document reviews current issues with the geospatial specification in relation to GeoOrigin and offers improvements for future versions. This document is offered in partial fulfillment of the requirements of X3DEarth NPS contract N00244-07-P-1192.

We begin with a brief description of the geospatial nodes to provide necessary background for the remaining discussion. The discussion focuses mainly on deficiencies in the specification and use of GeoOrigin in relation to other geospatial nodes.

Purpose of geospatial nodes

As illustrated in Figure 1, the X3D geospatial nodes are used in geospatial applications which can be viewed as a *simulation pipeline* of successive stages from application modelling to final video output. Simulation pipelines share a common precision funnel constraint: as processing moves from the early stages to the final video output stages the numeric precision decreases from high precision, e.g. 64bit double precision, to low 32bit single precision or less. The X3D geospatial nodes perform two main functions: projection of double precision geographical application coordinates into cartesian coordinates for video output; and enhancement of the accuracy of the visualisation which is seen as video output. The nodes themselves will not be enumerated or described here, but we will focus on nodes of current concern, such as GeoOrigin. For detailed information please refer to the X3D specification [1].

Geospatial nodes perform visualisation accuracy enhancement to addresses a fundamental problem with large scale simulation such as full scale modelling of the earth: coordinates cannot be represented with enough bits of precision at the video output stage to maintain human scale fidelity when navigating from space to ground or from the center of the earth to the surface. The attainable accuracy is generally poorer than one meter resolution. The video output stage, often called a graphics pipeline or image generator, is generally restricted to a precision of IEEE single precision floating point, i.e. 32 bits, or lower.

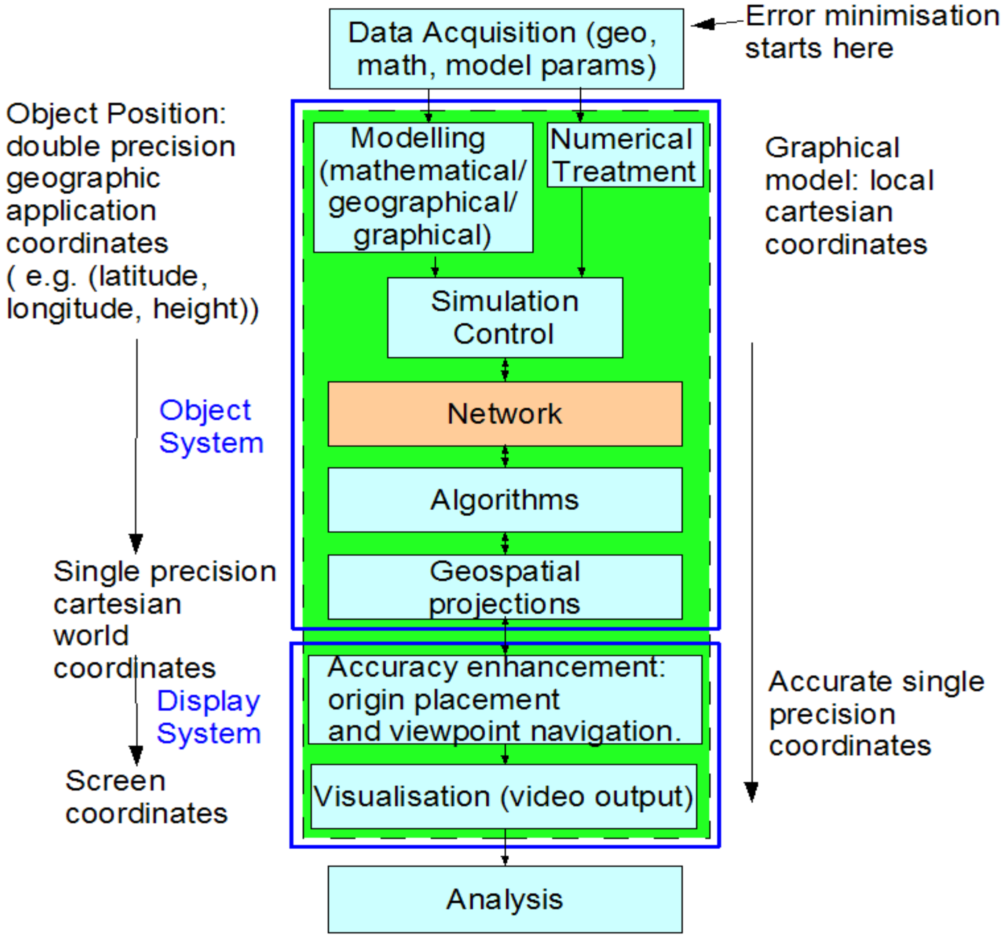

Figure 1: Stages of a geospatial simulation pipeline: cartographic input, geospatial and mathematical models, calculations and projections, accuracy operations and visualiasation. X3D geospatial nodes fit in the middle stages and provide geospatial projections and accuracy enhancement. This application of origin centric techniques to improve accuracy and scale is described in my Cyberworlds 06 paper [4].

Properties of floating point space

In order to understand how geospatial nodes improve accuracy it is necessary to understand some fundamental properties of floating point space. The accuracy limitation of graphics hardware is exacerbated by a property of the floating point representation used for coordinates: the resolution of floating point space is not uniform. Resolution decreases with distance from the origin, as illustrated in Figure 2. Although the nonuniform property of floating point representation is a major weakness at large distances from the origin, it can paradoxically be a major strength near the origin where resolution is so precise is it sub-millimeter. In Figure 2, the gaps between points are only representative to show their nonuniform distribution: if all representable points were plotted the gaps would not be visible so near the origin. Figure 3 illustrates floating point space in 3 dimensions. Representable coordinates are shown as red dots. At a large distance from the origin, over a million meters, the gaps between single precision floating point coordinates are so large they can be clearly seen. Near the origin, the gaps are so small that the points would appear as a solid volume.

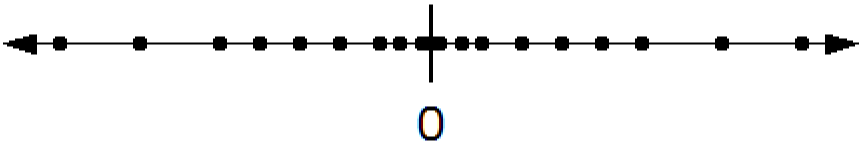

Figure 2: An illustration of how the representable floating point numbers are spaced with increasing gaps the further they are from the origin. The gaps between points are only representative to show their nonuniform distribution: if all representable points were plotted the gaps would not be visible

The high fidelity centered around the origin can be exploited, turning the nonuniform resolution into a strength, by taking an unconventional approach: by moving the space around the viewer instead of the viewer through the space. The geospatial nodes do this by moving the coordinate space origin to an area of interest prior to moving the viewpoint there.

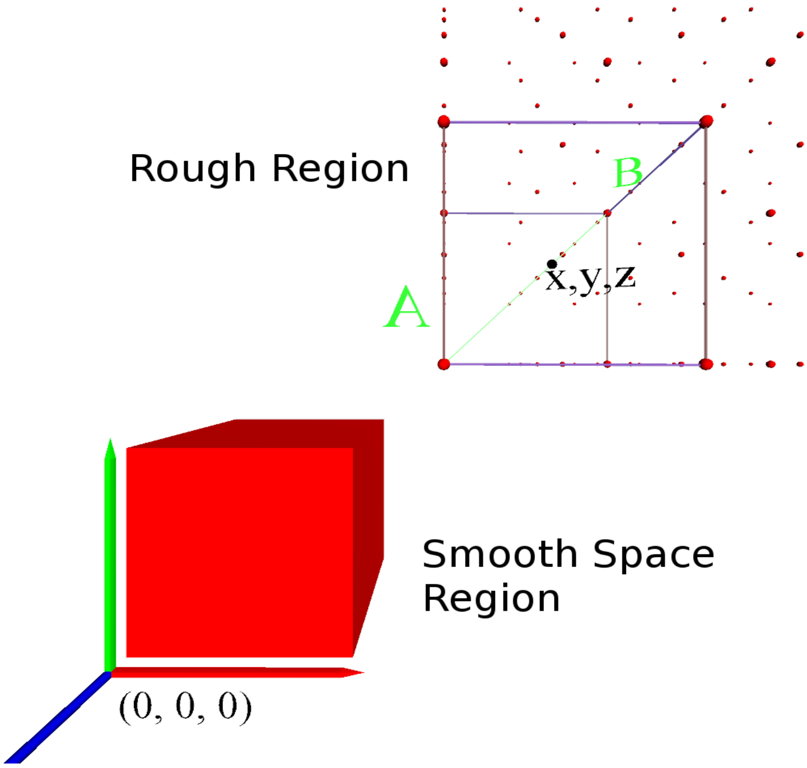

Figure 3: Variable resolution of floating point space. Real world coordinate (x,y,z) is quantised in a virtual world into one of the representable coordinates (red dots). The fidelity of virtual space is so high near the origin that the discrete red dots would appear as a continuous block. Spatial jitter is more evident in the rough region – away from origin.

How geospatial nodes transform coordinates while maintaining high fidelity

To explain how geospatial nodes work to improve accuracy we describe the operation of two nodes: the GeoViewpoint and the GeoOrigin. GeoViewpoint contains double precision coordinates, in some geographic reference system, such as (latitude, longitude, height), to move the viewpoint to. The GeoOrigin contains double precision coordinates representing where the origin of the local cartesian coordinate frame in the latter stages of the pipeline is to be.

A GeoViewpoint contains a geoOrigin field that references a GeoOrigin node. This field represents the center of an area of interest. It also contains a viewpoint position that is nearby or at its geoOrigin. Both geoOrigin and position are double precision application coordinates. The GeoViewpoint projects its position coordinate to a cartesian double precision coordinate. The geoOrigin coordinate is also projected to cartesian and subtracted from the GeoViewpoint cartesian position, resulting in a new position that is close to the local cartesian origin. Since the new cartesian viewpoint position is close to the origin, the floating point values are small and thus have the high accuracy of coordinates close to the origin illustrated in Figures 2 and 3. The GeoViewpoint cartesian coordinates are then converted to single precision floating point to be used by the lower precision graphics stages. Even after this conversion, because they are small values close to the origin, they still retain a high accuracy despite being in single precision, because even single precision shares the variable resolution property of all IEEE floating point coordinates.

In summary, setting a GeoOrigin does not alter the spatial relationship between objects. The spatial relationship between objects and viewpoints also stays the same. What does change is the fidelity of the space in which an object is rendered and calculations are performed. Moving the GeoOrigin to the area of interest improves the fidelity of the local cartesian space in that region and consequently improves the accuracy of calculations and rendering there.

The process described for GeoViewpoint also occurs similarly for other geospatial nodes, i.e. for all geospatial objects. In this way an entire large scale simulation, such as a full scale earth, can be visualised accurately, with smooth interactive fidelity, on single precision graphics hardware.

The current problems with GeoOrigin

Each geospatial node references a GeoOrigin via its geoOrign field. Therefore, there is a current need to have multiple GeoOrigin nodes in a scene when there are multiple X3D files containing GeoViewpoints or other nodes that must reference a GeoOrigin. For example, terrain consisting of a quadtree 5 levels deep will have some 127 tiles. If each contained a GeoOrigin node to be referenced by other nodes, it would lead to a huge proliferation of GeoOrigins.

At any one point in time, the video graphics output stage can only display information relative to one local coordinate frame. That is, only one cartesian origin is in use at any one time for the visual output to each user. Hence only one GeoOrigin can be in use at a time. Therefore it would make sense to have one GeoOrigin for the whole scene which is updated as the user navigates.

In the GeoVRML specification [5], from which the X3D geospatial specification was derived, “USE” of a unique GeoOrigin was stipulated: “N.B. you cannot define multiple local coordinate systems in a single scene. Therefore, it is recommended that only one GeoOrigin node is defined within a single VRML world, then all subsequent geoOrigin fields can USE this GeoOrigin node”. Also, in an earlier version of the X3D specification [6]: “As originally stated in the GeoVRML rfc (http://www.ai.sri.com/geovrml/rfc1.shtml - under Proposal: Level Two, about 2/3 into the document): " The Level 1 specification above is probably perfectly adequate *as long as only one GeoOrigin is defined within a single VRML world*. "”

With the current X3D specification, it is difficult to conceive of a way in which a unique GeoOrigin node could be used by a scene consisting of multiple files containing geospatial nodes. It may be possible to access a single GeoOrigin via an IMPORT although, because it “may not be instanced via the USE statement” so some ROUTE would be required to set the field. As you can see, the current possibilities for use of a unique GeoOrigin in X3D are inconvenient and cumbersome, if indeed they will work at all in an existing Browser implementation.

Even if it was possible within the current specification to use a single GeoOrigin for the entire scene, what about moving to other areas of interest? The GeoOrigin geoCoords field is an [in,out] field, which means it can be modified. So, one GeoOrigin in the main scene file could have its geoCoords dynamically modified as one moved to different areas around the globe. This, however, does not solve the difficulty or impossibility (within the current specification) of using a single GeoOrigin in multiple files.

A current compromise is to use a single GeoOrigin value throughout a world scene. This approach means that when rendering the globe with the requirement for a fixed value GeoOrigin coordinate, one does not know where the detailed terrain will be placed. This creates other problems:

1. It defeats the purpose of GeoOrigin because you cannot locate it where the detailed terrain will be (unless you pre-force the globe's origins to be there).
2. It requires hand editing of the detailed terrain to match the GeoOrigins in it to that of the globe (or vice versa).
3. You cannot achieve high accuracy when moving from one part of the earth's surface to another, or from space to the surface.

There are two other problems with the current X3D geospatial specification:

1. Once a GeoViewpoint places the user view at a position with high fidelity near the GeoOrigin, there is nothing within the geospatial specification preventing the user from freely navigating (e.g. in Walk, Fly mode) to a great distance from the origin where the resolution will be so poor that jittery motion and other *spatial jitter* [3] effects begin to occur.
2. There is also nothing within the specification preventing the GeoViewpoint position being the other side of the globe from the GeoOrigin, which will again cause severe loss of accuracy.

How to solve the current problems with GeoOrigin

Solution 1: remove need for multiple GeoOrigins or excessive IMPORT/ROUTES

In order to solve the problem of using a scene GeoOrigin in multiple files, a more elegant solution than what is possible within the current specification must be found. The X3D specification needs to define the requirement that Browsers that implement the geospatial component must apply the GeoOrigin, wherever it is defined in the scene, to all geospatial nodes. All geospatial nodes will implicitly use this GeoOrigin. It will therefore not be necessary to have a geoOrigin field in other geospatial nodes and the field can be removed from the node specifications.

It is necessary that a Browser behaves correctly to having a single GeoOrigin in a main scene file. Text should therefore be included in the node definition to the effect that if the geoCoords field is updated, the Browser will move the local cartesian coordinate frame origin correspondingly.

This solution does not address the problem of free navigation causing an increase of spatial error as the distance from the local coordinate frame origin increases. There is also nothing within the specification preventing the GeoViewpoint position being the other side of the globe from the GeoOrigin, which will again cause severe loss of accuracy.

Solution 2: The ultimate solution – remove GeoOrigin

Ultimately, the only real requirement is that the Browser meets the accuracy and scalability requirements of X3DEarth. That is, it needs to ensures that a local origin is located at or near to where the user navigates, yielding similar accuracy improvements as described above for the implementation of GeoViewpoint but in a foolproof way. Therefore, the X3D specification only needs to stipulate that Browsers implementing the geospatial component must meet an accuracy metric: maintaining a specified accuracy when navigating from space to surface and from one place on the surface to another, or underground.

It may be worth retaining some facility for the content developer to provide a hint to the Browser where the highest fidelity is required. This would allow the Browser to ensure its accuracy optimisations best serve the X3D application. The geoOrigin field in GeoViewpoint could be used for this purpose: it could become a n indication of the center an area of interest to which the GeoViewpoint is associated.

Meeting such and accuracy metric can be achieved by implement a form of moving origin (documented in [2,3,4]) such that the geospatial nodes will deliver sufficient accuracy to allow navigation from space to ground, without loss of *avatar interactive fidelity.* That is, there must be enough fidelity at all times for smooth, precise interaction at the avatar scale. Avatar interactive fidelity will need to be defined in the specification to an agreed metric, such as: *coordinates must always be accurate to a millimeter*. Precise math for calculating spatial error is given in [2]. It will then be up to the Browser to implement one of the many approaches to moving the origin in order to maintain the accuracy metric. My draft thesis [2] provides complete coverage of existing and my proposed best practices to maintaining high fidelity throughout a large scale simulation space. Browser implementers are free to choose from the existing approaches reviewed in the thesis or the origin centric approach developed in the thesis itself. Examples are: a continuous floating origin [2, 3], a periodic shifting of origin (MS Flight Simulator, Halo, Terravision), piecewise shifting (worldwind) or something else. It is only the accuracy metric that X3D should specify.

Benefits of proposed solutions

The benefits of the two proposed solutions are:

1. They solve the current problem of proliferation of GeoOrigin nodes.
2. They reduce the complexity of the content by removing the need for the content developer to specify multiple GeoOrigin and IMPORT/ROUTE information just to manage accuracy.
3. The second solution removes the onus for managing much of the complexity of maintaining accuracy from the content developer and places it on the Browser implementer.
4. For the second solution, Browser implementers have the freedom to implement the solution that best fits their requirements and design as long as the accuracy metric is maintained.
5. The specification is simplified through the reduction of fields/nodes while still achieving the goal of enabling space to face scalability and maintaining suitable accuracy at the same time.
6. The second solution addresses the problem of free navigation causing an increase of spatial error as the distance from the local coordinate frame origin increases because the Browser can implement either a continuous floating origin [2,3] or on-the-fly or periodic shifting of the origin[2,3] to prevent spatial error from increasing beyond a prescribed limit.

Note that the above solutions do not change the need for content developers to manage the hither/yon clip planes in the content. Clip plane management can also be managed automatically by a Browser but is currently modifiable by the content developer via the visibilityLimit and avatarSize fields of the NavigationInfo node.

References

1. http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/.
2. Draft Thesis: Origin-Centric Techniques for Optimising Scalability and the Fidelity of Motion, Interaction and Rendering, http://www.planet-earth.org/phd/ThorneThesis.pdf
3. THORNE, C. 2005. Using a Floating Origin to Improve Fidelity and Performance of Large, Distributed Virtual Worlds, *Proceedings of the International Conference on Cyberworlds 2005*, Nanyang Technical University, Singapore, 23-25 November, pp. 263-270. http://planet-earth.org/cw05/FloatingOrigin.pdf
4. THORNE, C. 2006. Error Minimizing Pipeline for Hi-Fidelity, Scalable Geospatial Simulation, *Proceedings of the International Conference on Cyberworlds 2006, Lausanne, Switzerland, Nov. ember 28-29,* pp 81-88.

http://planet-earth.org/cw06/thorne-CW06.pdf
5. http://www.ai.sri.com/geovrml/1.1/doc/nodes.html#GeoOrigin
6. http://www.igraphics.com/Standards/ISO_IEC_FDIS_Interim_19775_1_2008_2007_09_26/Part01/components/geodata.html#high-precisioncoords