SRI International

SRI's Digital Earth Project

August 1, 2002

Technical Note No. 560

Prepared By:	Yvan Leclerc,	Artificial Intelligence Center
	Martin Reddy,	Artificial Intelligence Center
	Michael Eriksen,	Artificial Intelligence Center
	John Brecht,	Center for Technology and Learning
	SRI International	
	David Colleen,	Planet 9 Studios

Approved by:

C. Raymond Perrault, Director
Artificial Intelligence Center

William S. Mark, Vice President
Information and Computing Sciences

SRI's Digital Earth Project

Y. Leclerc[†], *M. Reddy*[†], *M. Eriksen*[†], *J. Brecht*[‡], *and D. Colleen*[*]

[†]Artificial Intelligence Center, SRI International, Menlo Park, CA
[‡]Center for Technology and Learning, SRI International, Menlo Park, CA
[*]Planet 9 Studios, San Francisco, CA

INTRODUCTION

Objective

The goal of SRI International's Digital Earth project was to develop the infrastructure for an open, distributed, multiresolution, 3-D representation of the earth, into which massive quantities of georeferenced information can be embedded (Figure 1). Ex-Vice-President Al Gore referred to this vision as the "Digital Earth" (Gore, 1998). This required, in part, vast resources to build and maintain synthetic models of the earth, resources that only a worldwide distributed effort can achieve. It also required an internetwork capable of storing and rapidly indexing massive quantities of data by geographical location (color imagery of the planet at 1-cm resolution alone requires 1 petabyte of storage) and of delivering the data required for a given viewpoint at rates as high as 50 Mbits/s per person.

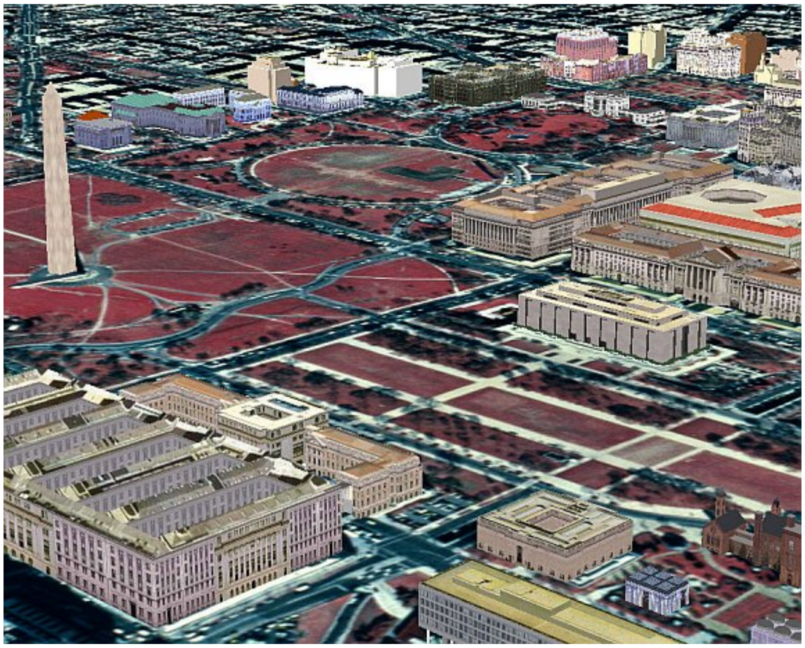


Figure 1. A high-resolution 3-D model of the National Mall area in Washington D.C., built as part of SRI International's Digital Earth project.

This work comprised a team from the Artificial Intelligence Center at SRI International (*www.ai.sri.com*), the Center for Technology and Learning also at SRI International (*www.ctl.sri.com*), and Planet 9 Studios (*www.planet9.com*). The project was funded by the DARPA Next Generation Internet (NGI) program under contract MDA972-99-C-0011 and ran from September 1999 to September 2002. A project website is maintained at *www.ai.sri.com/digital-earth*.

Approach

The project addressed two principal topics with respect to enabling a Digital Earth capability: 1) an architecture that enables the discovery of all information relevant to a user for a particular area of the planet, and 2) software clients that provide novel interfaces to navigate and present the data in this architecture. To demonstrate these capabilities, we also built compelling educational application scenarios and developed rich 3-D urban models and other data/metadata repositories.

In the first instance, we employed existing Web standards to harness resources and enable anyone to create, update, and access geographically indexed data anywhere in the world. Specifically, the Domain Name System (DNS) was used to develop a hierarchy of HTTP (and other) servers, where each server is named after, and is responsible for, a given geographical area (or cell) of the earth at a given level of detail. This architecture, which we refer to as the GeoWeb, is capable of storing and organizing the quadrillions of bytes of data required by the Digital Earth. This opens up the world of geographically indexed data in much the same way that search engines on the Web have opened up text-indexed data.

In addition to this DNS-based index, we developed a range of applications to enable the efficient and seamless navigation of this globally distributed database. The primary application was a radical enhancement of our cross-platform terrain visualization system, TerraVision. This high-performance system enables the user to browse massive, rich, distributed terrain; view embedded multimedia data; and collaborate with other users in the virtual 3-D space. As a part of this work, we leveraged and evolved our GeoVRML efforts to provide an open standards solution for the interchange and display of accurately georeferenced 3-D models and built large urban models, including sites within Washington, D.C., and New York. These technologies have all been demonstrated in concert operating over DARPA's testbed next-generation network, SuperNet.

Finally, we developed educational applications around these Digital Earth technologies, with a focus on wilderness education. This involved letting students explore an environment virtually before a field trip, and then to update the 3D model afterwards with results and observations to reinforce the experience.

TECHNOLOGY DEVELOPMENT

The GeoWeb

The Web has revolutionized the way documents are published, found, and viewed. This revolution has succeeded in part because Web protocols and publishing standards are open, and there exist freely available tools for publishing and reading Web documents. Just as important, powerful search engines let users find documents of interest almost instantaneously. Without these search engines, the Web would be almost useless.

Even though search engines can now find almost all Web-accessible documents about a given topic, it is currently difficult to find Web-accessible data about a given location because the location is typically not a part of the data themselves. Examples of such georeferenced data include aerial and satellite images, 3-D models of buildings and weather systems, and vacation pictures taken with GPS-enabled cameras. Today's search engine technology is not applicable to discovering these items, and the huge amount of georeferenced digital data that are available on the Web today is virtually inaccessible.

In an attempt to make georeferenced data accessible, several companies and organizations have created private or governmental databases that hold a small fraction of all georeferenced data. For example, companies such as MapQuest maintain private map databases with associated street addresses and links to businesses like restaurants and shops. These databases hold searchable metadata, a summary description of the data that includes geographic location, which can be searched either directly or via a clearinghouse. However, none of these organizations is either capable of, or willing to, create a database that would allow anybody in the world to publish and search georeferenced metadata. Indeed, this task is so large that no single, regional, organization could do it alone.

What is needed instead is a coordinated global infrastructure with participating organizations from around the world. We call this open-standards-based infrastructure the GeoWeb, which can be realized as a new top-level domain (e.g., .geo) that would enable anybody to publish and search for all metadata referring to a given area (Leclerc et al., 2001a; Leclerc et al., 2001b; Leclerc et al., 2000). The infrastructure is based on a hierarchy of servers whose domain names represent geographic areas. An example hierarchy, described below, is nominally of the form minutes.degrees.tendegrees.geo. (For convenience, we use “.geo” as the top-level domain name in all our examples of the GeoWeb hierarchy, although existing top-level domains could also be used, and in fact our prototype implementation uses “.dotgeo.net”.)

Using the GeoWeb hierarchy, Web sites or specialized applications let users specify a search profile (keywords and other pertinent information) and find all georeferenced data satisfying the profile in a given area. Because the data are georeferenced, they can be embedded in 2-D maps or 3-D models of the earth, letting the users navigate through the map or model, providing a much more natural means of finding data than traditional search engines.

The GeoWeb DNS Hierarchy of Servers

Metadata in the GeoWeb are distributed across servers for scalability. The servers have distinct Internet Domain Names that identify regions on the earth bounded by latitude and longitude. Such a region is called a *cell*, and the domain name that identifies it is called a *geographic domain name* (see Figure 2). Following are example applications of geographic domain names using the minutes.degrees.tendegrees.geo name schema. (This schema is used here for simplicity, although the schema currently favored is actually of the form tenminutes.degrees.tendegrees.ninetydegrees.geo.)

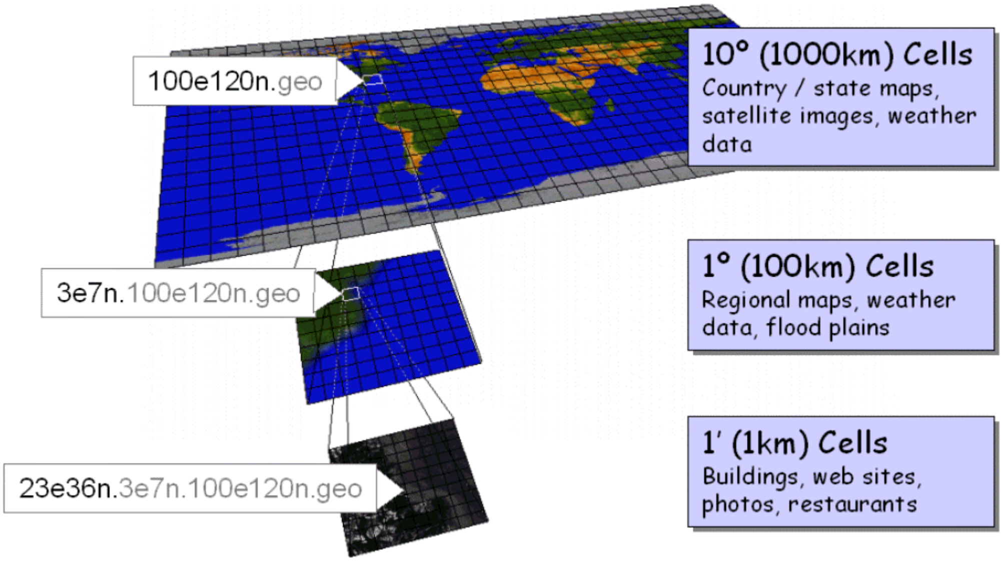

100e120n.geo
3e7n.100e120n.geo
23e36n.3e7n.100e120n.geo
10° (1000km) Cells
Country / state maps, satellite images, weather data
1° (100km) Cells
Regional maps, weather data, flood plains
1' (1km) Cells
Buildings, web sites, photos, restaurants

Figure 2. An illustration of the GeoWeb hierarchy showing example server names.

- The geographic domain name *20e30n.geo* identifies the 10-degree x 10-degree cell whose southwest corner is located at 20 degrees east, 30 degrees north.
- The geographic domain name *2e4n.10e50n.geo* identifies the 1-degree x 1-degree cell whose southwest corner is located at 12 degrees east, 54 degrees north.
- The geographic domain name *11e21n.3e7n.30e10n.geo* identifies the 1-minute x 1-minute cell whose southwest corner is located at 33 degrees, 11 minutes east and 17 degrees, 21 minutes north.

Metadata are placed in the hierarchy according to the number of cells that overlap the geographic coverage of the data to which they correspond (see Figure 3). Specifically, the level with the smallest cells is used so that no more than four cells overlap the geographic coverage. At this chosen level, the metadata are stored in all the overlapping cells.

This naming convention and metadata placement strategy allows clients to determine which host(s) to query for metadata referring to a given area, thereby distributing the load over many servers with no single point of failure or congestion. This distribution and query method also reduces the storage and bandwidth requirements of any given server, making it possible for thousands of high-speed visualization systems to be simultaneously searching for and retrieving metadata and data.

The use of DNS also allows metadata to be transparently moved to new hosts and subdomains as needed. For example, initially all metadata may be physically hosted on a single computer, with all .geo domain names aliased to that machine. But over time, as areas fill with metadata (such as large urban areas), their corresponding subdomain and metadata can be transparently transferred to new computers. These computers may be maintained by other organizations that may, when necessary, transfer data to new computers (and new organizations) deeper in the hierarchy.

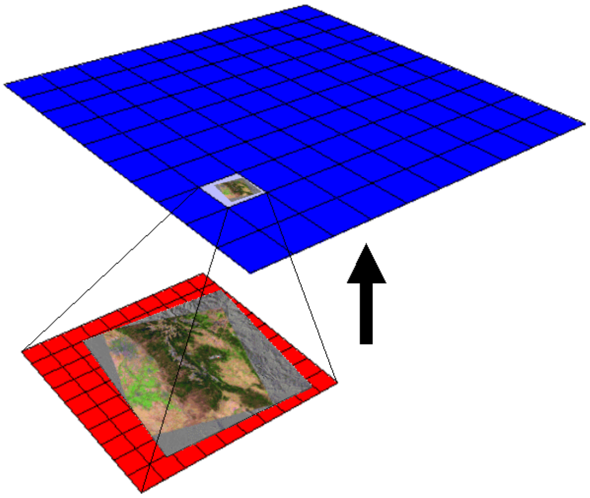


Figure 3. Demonstration of the procedure to locate the appropriate cell server(s) to store a new GeoWeb item.

GeoWeb Design Principles

The GeoWeb is a distributed searchable database of metadata. It comprises the metadata standard, the distributed searchable database in the form of a DNS hierarchy of servers, and an API library for publishing, searching, and validating/endorsing metadata in the servers. The GeoWeb is designed according to a few basic principles, similar in many ways to the Alexandria Digital Library (see *www.alexandria.ucsb.edu*):

1. **Naive transparency**. Mapping from categories to metadata content is syntactically and semantically trivial. It should not be a difficult task to put together a sophisticated query involving a combination of criteria for a search.

2. **Latitude/longitude and temporal extent**. Since the basic search criteria are latitude, longitude, elevation, and time, these must be explicit and required fields in the metadata.

3. **Efficient coverage**. In general, search fields should all be of relatively equal relevance, so that there is no overhead due to making something explicitly available for search that is rarely used.

4. **Abstract query formats**. One should explicitly avoid exposing any of the internal structure of the database in order to formulate queries. Moreover, one needs to allow for the possibility of sophisticated Boolean combination of simple queries in order to find exactly the desired set of metadata.

5. **Standard format**. It is desirable to adopt or at minimum support one of the existing standard metadata formats for external publishing (e.g., Dublin Core, FGDC, ISO/TC211).

6. **Multilingualism**. It is important to respect the need to publish and discover metadata in languages other than English. This demands support for both multilingual descriptions and cross-linguistic categorizations of objects and services (e.g., the Universal Standard Products and Services Classification (UNSPSC) system).

7. **Privacy and information security**. It is vital for many potential uses of the GeoWeb infrastructure to ensure that privacy can be maintained in the face of a publicly accessible searchable infrastructure. A general framework is needed for restricting access to links, disallowing "tracking" of dynamic metadata items and allowing third parties to validate or endorse metadata entries.

GeoWeb Metadata

As the concepts that lead to the GeoWeb have developed, it has become clear that we have a somewhat different view of what may be considered georeferenced information and thus envision a very different kind of metadata than is traditional. Traditional metadata (e.g., Dublin Core, FGDC, or ISO/TC211) usually describe a particular, single data item or service (e.g., a single Web site, Web page, aerial image, or photograph). The searchable metadata we have been developing are different from this in a fundamental way: we want to provide a semantic and geographical description of physical or conceptual objects that may have numerous different manifestations as data or services.

For example, a physical object such as a restaurant may have a metadata entry keyed to its location that contains a description and links for several distinct data elements (e.g., a home page, a menu, and a VRML model of the building) and services (e.g., a Voice over IP address or an online takeout order form). A user should be able to search for "a South Indian restaurant with an online menu and photographs within walking distance of my hotel." While it makes a great deal of sense to provide a semantic nexus for information at this level, this does not correspond to a traditional view of geographic metadata. Nevertheless, we have endeavored to build this level of capability on top of existing metadata work such as the ADL Collection Metadata (Hill et al., 1999).

It is worth noting, however, that the semantic break is not actually that great. It should be clear that traditional geographic metadata are actually encompassed by this broader conceptualization. A publisher could consider a single piece of data to be the appropriate conceptual object to be described by a metadata entry. It should be up to the metadata publisher to choose which level of object description is most appropriate for representing information within the index. Given that the choice to register a metadata entry is a very simple form of advertising, this flexibility can be considered to be one of the advantages of an opt-in registration strategy.

Our final metadata format relies upon the XML Resource Description Framework (RDF), which was chosen for its flexibility and ability to expand over time with the inclusion of new ontology schema. As such, we have produced an RDF Schema that represents all of the metadata entries for the GeoWeb. This schema is included for reference in the appendix of this report.

Dynamic GeoWeb Content

Many objects that might gain some advantage by being registered in the GeoWeb do not have fixed positions at all, and it is their current position (and potentially history and planned future of movement) that should be reflected in the GeoWeb. These dynamic objects require a different interface to maintain their positions in the GeoWeb hierarchy and their associated geographic domain names. One must design a facility whereby an intermediate directory server can manage the metadata for such objects based on a globally unique object identifier instead of geographic location. This intermediate server then translates updated locations to the appropriate geographic domain name and updates the metadata in the GeoWeb hierarchy, taking care to remove metadata entries when they dynamically move out of particular cells. These intermediate servers may be either public or private (depending on the need to advertise or hide the object-ID to metadata mapping) and may offer other associated services. By structuring the solution this way, all of the distribution and localized discovery advantages of the GeoWeb architecture are retained while allowing dynamic objects to efficiently update their locations.

For example, an aircraft equipped with GPS receivers and secure wireless Internet access could periodically contact the dynamic directory server to update its current location, velocity, and other attributes. This information would then be used to update the geographic coordinates (and other attributes) in the corresponding metadata record in the GeoWeb hierarchy and, when necessary, transfer the metadata record to a new cell. This dynamic metadata can then be used for many purposes, including air traffic analysis and providing global near-real-time maps of air traffic. Of course, the aircraft could in turn use the GeoWeb to download or access localized flight/airspace support information.

In the case of weather data, the dynamic directory server might be maintained by the U.S. National Weather Service itself, since its mandate includes the maintenance of real-time, public weather maps and storm information. As in the example above, the dynamic directory server would periodically update the corresponding metadata record and, when necessary, transfer it to a new cell.

Should the resource and network loads on these dynamic directory servers become too great for traditional single-point database methods, it would be relatively straightforward to take advantage of the strategy used by the location-based GeoWeb hierarchy and create a distributed DNS hierarchy to represent an object-identity index. For example, airplanes already have a registration ID that could be translated (and possibly encrypted for privacy) to a DNS name/id pair such as *us.air.geo/AC7321*. This server/id pair could then be used with exactly the same protocol as is used to update GeoWeb registry cells to update a dynamic directory server that then automatically updates the associated entry in the location-based GeoWeb hierarchy. In database terms, we can provide a second primary index into the global metadata database using exactly the same update protocols. These secondary indices could be either private or public, depending on data privacy concerns in the particular domains implemented.

Validation of Metadata and Data

To ensure the accuracy and validity of certain classes of data for certain purposes (such as elevation data used for city planning), the metadata record has a field for one or more optional validation certificates issued by validation organizations. The digitally signed certificate will certify that one or more elements of the metadata/data meet certain qualifications, such as accuracy or completeness, or compliance with local, national, or international standards or conventions. While essential for validation purposes, this facility also has other uses, such as for reviews. For example, an organization such as the American Automobile Association (AAA) could annotate hotel and restaurant listings with references to its own independent descriptions of the facilities and quality of service that these businesses provide. Clearly, this facility has further implications for security and privacy than those already outlined above.

GeoWeb Advantages

The GeoWeb infrastructure offers several advantages over today's state-of-the-art technology. It is specifically designed as an extremely scalable, open, and global geographical index to the Web. Below are a few of the key benefits of this technology.

- **Open opt-in scheme.** No one company or institution can hope to index all geographically related information for the whole planet. Therefore, the GeoWeb is an opt-in scheme where users can register their own data, be it information about their bed and breakfast, or pictures of their trip to the Big Island.

- **Integration of disparate data sources.** Instead of being restricted to one company's view of the world, the GeoWeb allows users to discover and integrate disparate sources of information for improved decision making.

- **Massively scalable architecture.** The entire GeoWeb index could be stored on one server, but as demands and database size grow, individual cell servers can be split off onto additional physical machines. To illustrate the inherent scalability, more than 60,000,000 1-minute servers are possible over the land regions of the earth.

- **No one-server bottleneck.** Most contemporary search and indexing interfaces require going through a single point-of-failure server. In the GeoWeb, the client works out which cell server to contact based on the geographic extent of the query.

- **Transparent load balancing.** Using the DNS hierarchy enables transparent load balancing by moving data to new servers and organizations as needed. The IP addresses change and the bandwidth improves, but the domain names remain the same.

- **Indexing of any content.** The GeoWeb can index any content, not just HTML. Movies, sound files, text, 3-D models, terrain, and so forth. can all be indexed. Content such as HTML with geographic meta tags allows an easy way to register that content in the GeoWeb.

- **Built-in security.** Security is imperative to ensure that private data remain private, while still providing easy access to public information. In addition, there is a scheme to allow appropriate organizations to validate and/or endorse data.

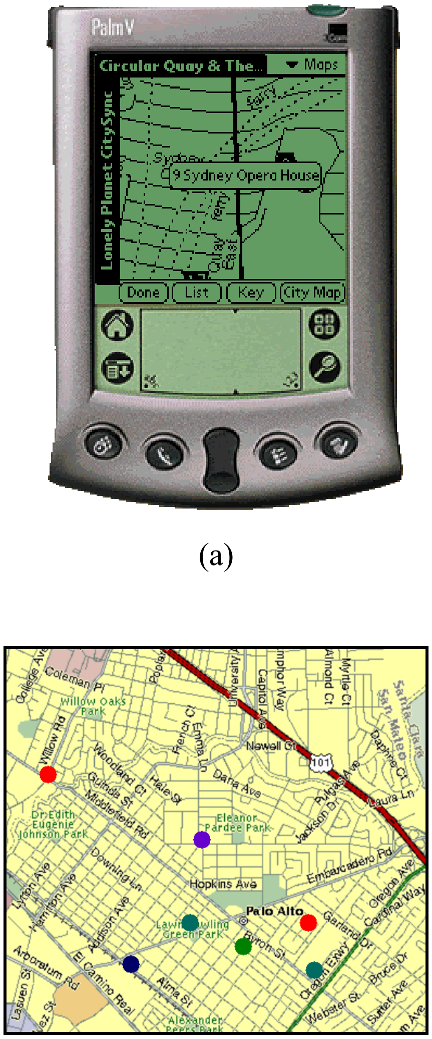
- **Support for high bandwidth demands.** Massive distribution of the geographic database enables many clients around the world to retrieve data simultaneously and at high speed.

- **Reduced server costs.** As the GeoWeb infrastructure is distributed over many cell servers, there is no need for a single massive server with huge storage requirements. The size and speed needed per server is therefore reduced, for example, divided by 10,000 or more for 10-minute servers.

- **Local control.** Companies, institutions, or local governments can manage the cell servers for their regions of the world, and also host the cell servers in their geographic regions.

GeoWeb Clients

Naturally, users will not normally browse the GeoWeb by typing in domain name addresses such as *11e21n.3e7n.30e10n.geo.* The benefit of the GeoWeb is that a client application or portal can trivially work out the server to contact for any given region of the earth. A range of possible GeoWeb clients can be envisioned. The following sections introduce a few of the interfaces that are proposed or are already under development.



(a)

(b)

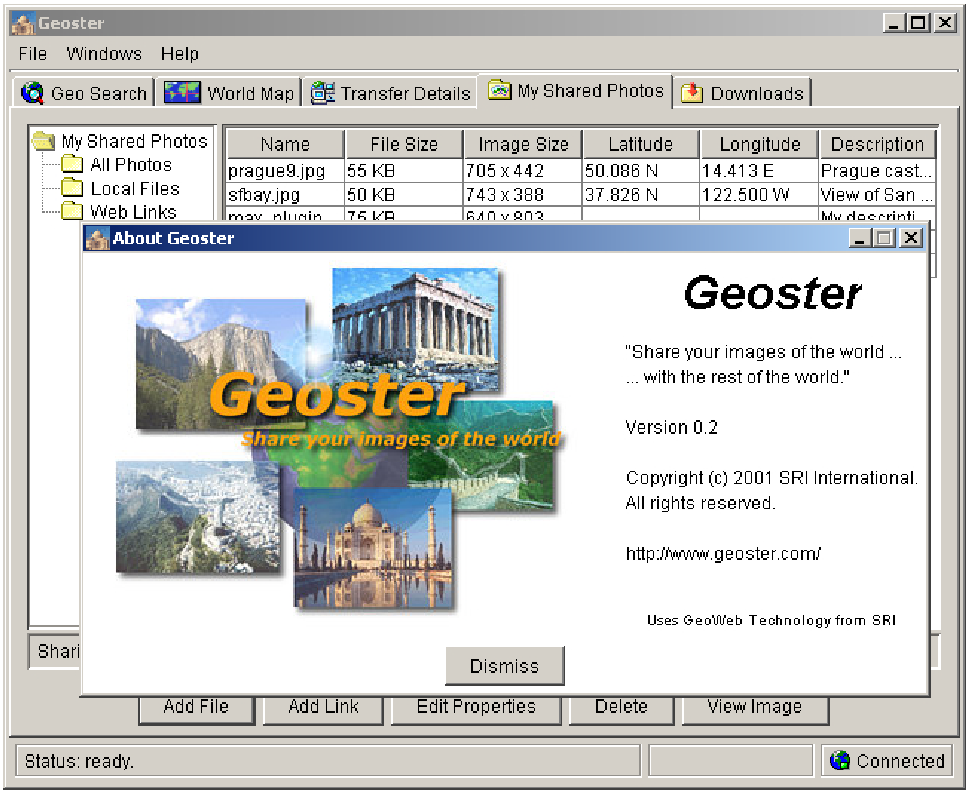
Figure 4. Potential GeoWeb clients, (a) a handheld interface, (b) a map-based interface with hyperlinked icons.

Text-based: The most basic kind of interface would be similar to the types of search engines that we use today. However, in addition to entering a search criterion, the user would also enter a location for the search. This could be a latitude/longitude coordinate or, more intuitively, a street address or a feature name that can be translated to a geodetic coordinate by using a gazetteer service. The results could be presented as a simple list of links (ordered by distance), as is the case for most search engines today. This type of interface has the advantage that it scales down well to current handheld devices such as a PalmPilot^{TM} or iPad^{TM}, as shown in Figure 4(a).

Map-based: There are already various mapping sites on the Web that will generate a 2-D map for a region of interest, for example, MapBlast!, Yahoo! Maps, MapQuest, and Maps.com. It is easy to imagine a GeoWeb client that lets users search over a particular geographic region and return the results as a map image with various icons overlaid (see Figure 4(b)). These icons could be hyperlinked so that clicking over them would take the user to more information about that item, for example, clicking over the icon for a restaurant might lead to its menu or online reservation page.

TerraVision: TerraVision^{TM} is a distributed, interactive terrain visualization system developed by SRI International. It allows users to navigate, in real time, through a 3-D graphical representation of a real landscape created from elevation data and aerial images of that landscape. TerraVision is described in detail later in this chapter. It is designed as a GeoWeb client so that, as the user flies around the world, the system is constantly querying the GeoWeb index to discover all relevant information about the place being viewed. This information is then streamed automatically to the user's display for browsing, thus providing a novel visual and geographic interface to browsing the Web.

Geoster: The popularity of file sharing protocols and applications has been amply demonstrated with the enormous usage of services like Napster and Gnutella. SRI developed a prototype file sharing application along the lines of Napster, but using the GeoWeb as the underlying indexing and search infrastructure. Geoster (pronounced *jester*) is an application built for sharing photos over the Web using the GeoWeb (see Figure 5). It lets users index photos from around the world and share these with the rest of the world. This would allow users to do searches such as, "show me all photos that have been taken of the Great Wall of China". Geoster is written in Java and hence available across multiple platforms. We believe that this type of client could produce a large interest in geographic search and generate a substantial amount of GeoWeb registrations.



Name	File Size	Image Size	Latitude	Longitude	Description
prague9.jpg	55 KB	705 x 442	50.086 N	14.413 E	Prague cast...
sfbay.jpg	50 KB	743 x 388	37.826 N	122.500 W	View of San ...
max_plugin	75 KB	640 x 803			My descripti

Geoster

"Share your images of the world ...
... with the rest of the world."

Version 0.2

Copyright (c) 2001 SRI International.
All rights reserved.

http://www.geoster.com/

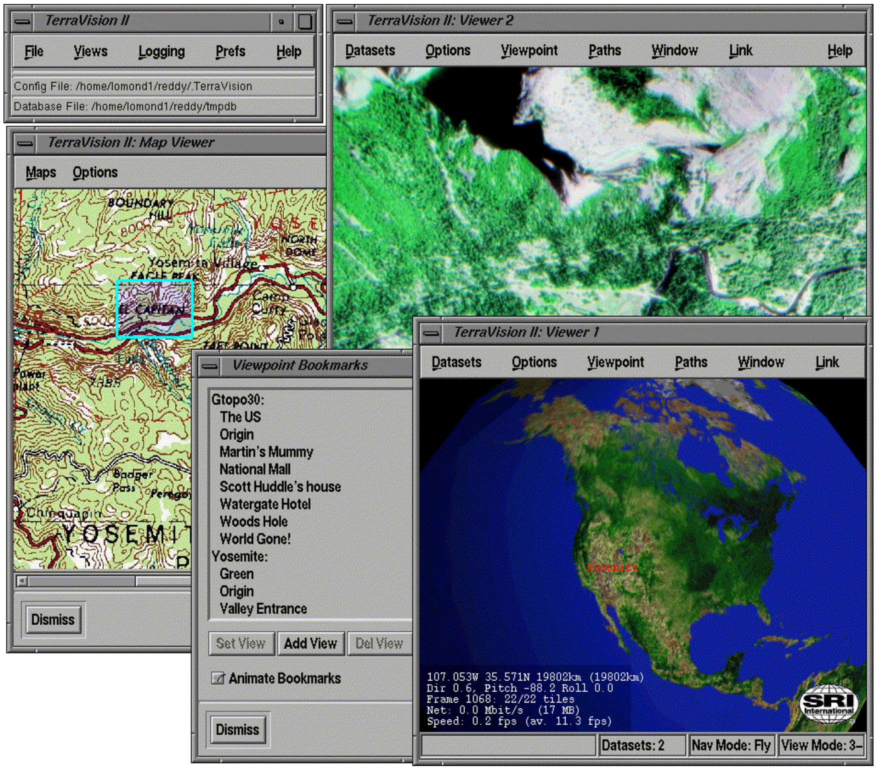
Uses GeoWeb Technology from SRI

Status: ready.

Figure 5. A sample GeoWeb client: the Geoster photo-sharing client.

TerraVision

TerraVision^{TM}, a system for interactively browsing 3-D representations of large geographic areas (Leclerc and Lau, 1994; Reddy et al., 1999), was developed in response to shortcomings in other systems that can browse terrain data only from local disk resources, and hence are limited by users' ability to store and maintain entire datasets locally. TerraVision can therefore retrieve and merge massive volumes of remotely located data, including aerial and satellite imagery, topography, weather data, buildings, and other cultural features (see Figure 6). These data can be terabytes in size, distributed over multiple servers across the Web, and can be automatically discovered using the GeoWeb technology described earlier in this chapter.



Config File: /home/lomond1/reddy/.TerraVision
Database File: /home/lomond1/reddy/tmpdb

```
107.053W 35.571N 19802km (19802km)
Dir 0.6, Pitch -88.2 Roll 0.0
Frame 1068: 22/22 tiles
Net: 0.0 Mbit/s  (17 MB)
Speed: 0.2 fps (av. 11.3 fps)
```

Datasets: 2 | Nav Mode: Fly | View Mode: 3–

Figure 6. Screenshot of the TerraVision system.

User Scenario

The user starts TerraVision and sees a 3-D view of the earth. As the user travels around the globe, browsing for a particular area, TerraVision automatically fetches data distributed across many sites throughout the Internet by querying the GeoWeb. The user flies down to ground level and is able to find all nearby restaurants, places of interest, and even the local garage sales to be held on that day.

Performance, scalability, and portability make TerraVision (and its associated data) a useful tool for numerous applications, such as military personnel performing mission planning and battle damage assessment, emergency teams fighting a forest fire or organizing disaster relief efforts, environmental workers evaluating a floor, or other time-critical conditions.

Technology Overview

To interact with massive, remotely located repositories in real time, TerraVision employs a tiled, multiresolution data representation (described in detail below). This involves segmenting the original data into rectangular tiles over a range of resolutions, where each tile contains the same number of pixels or elevation data. By employing customized caching, culling, and data-fetching optimizations, the number of polygons and texture maps required for rendering remains approximately constant, independent of dataset size and the viewpoint. Tiles are requested for an area using a coarse-to-fine progression so that TerraVision always has low-resolution data for the area of interest. TerraVision supports the open standard GeoVRML format for representing building, weather, and other 3-D entities. TerraVision handles multiple types of imagery, allowing the user to select or blend between different datasets, for example, full aerial and weather earth models where parts of the surface will have imagery down to 1-m resolution. The user can navigate the terrain data with standard software on a personal computer connected to the Internet. A user can employ a graphics workstation connected to a fast network with high-speed disk servers to quickly navigate around a large area, but can also access this same data from a laptop machine over a wireless link when working in the field

TerraVision Features

- **Open Source**. Upon completion of the Digital Earth project, SRI International decided to release the entire source tree for TerraVision as Open Source in an effort to contribute to the community. This code base is therefore available to interested parties to learn from and to extend.
- **Distributed data.** TerraVision browses data that is distributed over a wide-area network, such as the Web, as well as locally installed data. TerraVision was specifically designed to cope with the inherent unpredictability of accessing data over a network.
- **Massive, scalable datasets.** TerraVision views massive datasets, in the order of terabytes. It achieves this by employing powerful optimization algorithms including view frustum culling, terrain and imagery level of detail, horizon culling, caching, and prediction.
- **Entire earth visualization.** TerraVision handles datasets in a variety of geographic coordinate systems, for example, latitude/longitude, UTM, and LVCS, and can transform these on the fly to a round-earth, or geocentric, representation.
- **Multiple datasets.** TerraVision views multiple datasets at once. For example, a user can have a 1-km resolution globe model, with a 25-m model of the San Francisco Bay Area, and a 1-m inset for Palo Alto.
- **Multiple viewers.** TerraVision allows the user to open multiple viewer windows to look at the same dataset from different perspectives at the same time, or different combinations of sets. The user can even control the view of one viewer through another.
- **Gazetteer support**. We have added support for querying several remote gazetteer Web services to translate a street address or place name into a

geographic location to which TerraVision can then fly directly. We use the Alexandria Digital Library Gazetteer at UCSB for this purpose.

- **GeoVRML model overlays.** 3-D models can be overlaid on the terrain to provide support for cultural features, such as buildings and roads, and atmospheric simulations, such as wind vectors and clear air turbulence models. We use GeoVRML to represent all models with a precise geographic location.

- **OGC WMS support.** The OpenGIS Consortium (OGC) developed the Web Mapping Server (WMS) interface as a standard way to serve and integrate maps over the Web. TerraVision can access data from any OGC-compliant WMS, and we include an OGC-compliant WMS Apache module to serve TerraVision and GeoVRML models.

- **Flight paths.** The user can set up predefined flight paths by marking viewpoints and then telling TerraVision to fly a path connecting those viewpoints. The user can vary the velocity, loop the path, close the loop, and select linear or spline interpolation.

- **Viewpoint bookmarks.** Any particular viewpoint, for example the user's own house, can be bookmarked. TerraVision will remember that location and camera orientation so that the user can fly back to it at a later time.

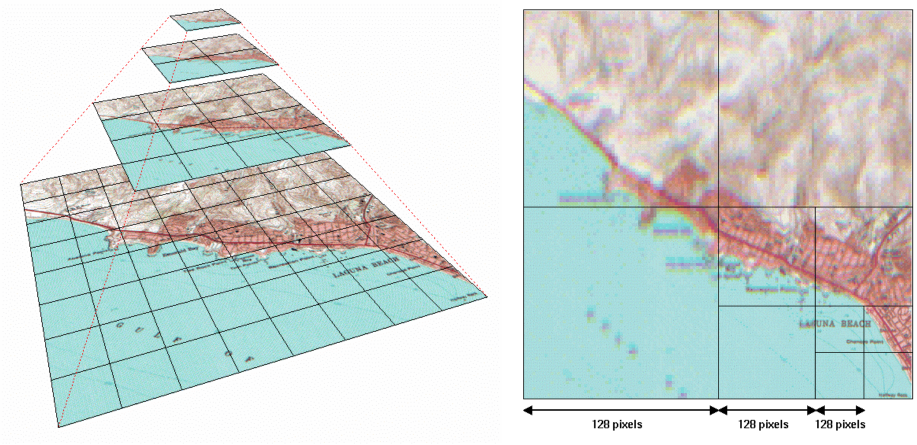
- **Heads-up display.** A simple HUD is available to provide information such as viewer location (in latitude/longitude), orientation, number of tiles displayed, data burst rate, and frame rate.

- **Documentation.** A Web-based user's guide is available to help new users familiarize themselves with the TerraVision system. TerraVision offers Help menus that bring up appropriate sections of this guide in users' Web browsers.

- **Cross platform**. TerraVision is available for SGI IRIX (6.3 and up), Linux, and Windows 98/ME/NT/2000. It is available as a full version with Tcl/Tk interface, a Netscape plug-in, and an ActiveX component for embedding in Microsoft applications such as Internet Explorer and PowerPoint.

Managing Multiresolution Grids

Level of detail (LOD) is the technique of changing a model's complexity based upon some selection criteria, such as distance from the viewpoint or projected screen size (Luebke et al., 2002). The basic premise for these two selection criteria is that any distant detail that projects to less than a single pixel on the screen will not generally be visible. To implement this, we need a mechanism to simplify the geometry and imagery for a dataset. Several existing polygon simplification algorithms work well for terrain. However, many of these are view-independent techniques that force the same degree of simplification across the entire terrain. These are not appropriate for our application because switching to the highest resolution would still involve loading every point of the original dataset. Instead, we require a view-independent technique where the degree of simplification can be varied with respect to the current viewpoint. This is often done using a hierarchical data structure, such as a quad-tree. A further requirement is that the LOD algorithm must not require access to the entire high-resolution version of the dataset. This is necessary because we do not want to be limited to viewing only datasets that can fit onto the user's local storage system. These requirements suggest that a tiled, pyramid representation is best suited to our needs.



(a) (b)

Figure 7. An example image pyramid. (a) Illustrates four different resolutions of an original image, where each level of the pyramid is segmented into 128- x 128-pixel tiles. (b) Shows how this structure can be used to alter the resolution of an image in different regions.

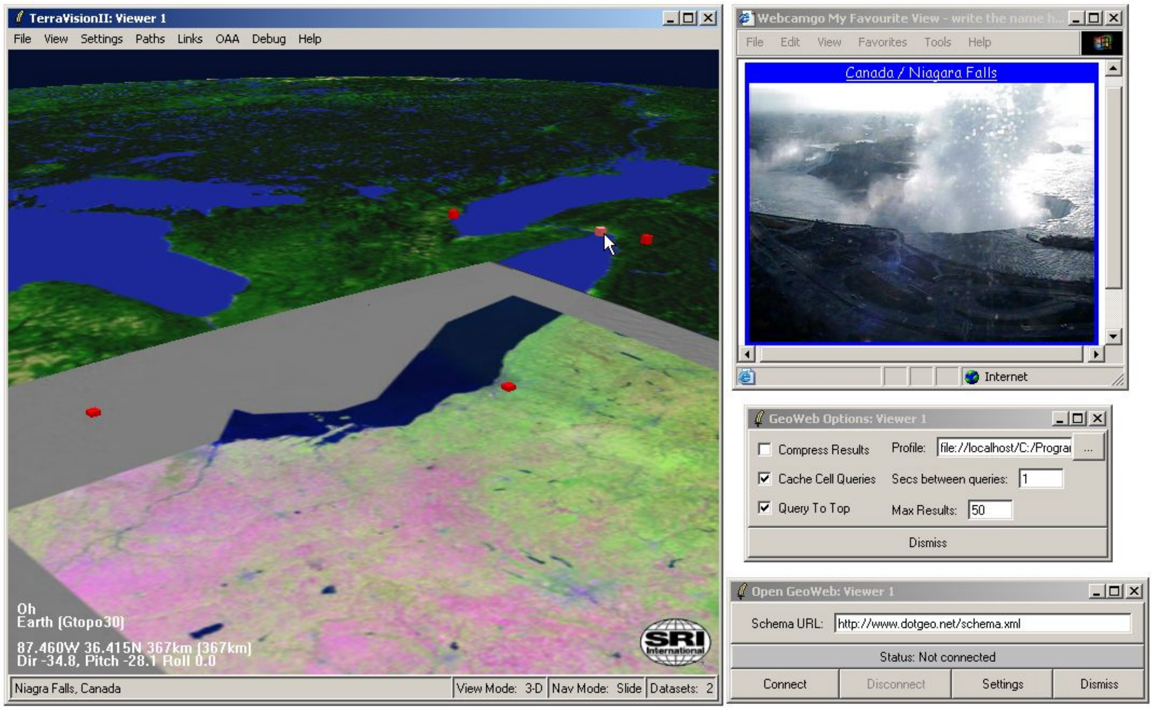
A pyramid is simply a multiresolution hierarchy for a dataset (DeFloriani, 1989). For example, if an original image is 1024 x 1024 pixels at 1-m resolution, then an example pyramid might contain this original image along with down-sampled versions at resolutions of 512 x 512 pixels (2-m), 256 x 256 pixels (4-m), 128x 128 pixels (8-m), and so on. Each image of the pyramid is then segmented into rectangular tiles, where all tiles have the same pixel dimensions, for example, 128 x 128 pixels as in Figure 7(a). A tile at one level of the pyramid will therefore map onto four tiles on the immediately higher resolution level. That is, the tiles at the higher resolution level cover half the geographical area of the former.

Using this representation, it is possible to recursively resolve certain regions of a dataset in more detail than other regions. For example, in Figure 7(b) we have displayed the lower-right corner in high resolution with the surrounding regions displayed in progressively lower resolution. Assuming a tile size of 128 x 128 pixels, this example requires downloading and rendering only 491 KB (10 tiles) instead of the entire 3.1 MB high-resolution image. If the user is located at the bottom-right corner, then we gain the effect that distant imagery is rendered at lower resolution than near imagery, that is, we have implemented distance-based LOD.

For simplicity, we have only considered pyramids of images so far. However, the same techniques can be applied to elevation grids, as well as other types of terrain data. By employing a tiled pyramid representation for the geometry as well as the imagery, TerraVision is able to optimize the amount of data transferred over the network, the number of polygons in the scene, and the amount of memory required for texture maps. The effect of this technique is that we need to fetch and display data only for the region that the user can see, and only at a sufficient resolution for the user's viewpoint. This solution scales well to arbitrarily large datasets because it effectively attempts to keep the polygon count constant for any viewpoint.

GeoWeb Integration

We have extended TerraVision to become a GeoWeb client. This means that, once connect to the GeoWeb, TerraVision will constantly query the index for any relevant data in the current view.



Oh
Earth (Gtopo30)
87.460W 36.415N 367km (367km)
Dir -34.8, Pitch -28.1 Roll 0.0

Niagra Falls, Canada | View Mode: 3-D | Nav Mode: Slide | Datasets: 2

□ Compress Results Profile: file://localhost/C:/Progra
☑ Cache Cell Queries Secs between queries: 1
☑ Query To Top Max Results: 50

Schema URL: http://www.dotgeo.net/schema.xml
Status: Not connected

Figure 8. Illustrating TerraVision acting as a GeoWeb client. The GeoWeb has discovered two visible terrain models (the earth and Ohio) as well as Web camera locations, which are displayed as red icons by TerraVision. The user has clicked on the icon over Niagra Falls and received a recent image for that area (top right).

If any metadata are returned by the GeoWeb, TerraVision will attempt to display this in the most appropriate fashion. For example, if the metadata describes a terrain model, TerraVision will attempt to add this to its 3-D model of the globe and begin streaming the terrain data over the Web. If the metadata describe a GeoVRML model, then this will be streamed over the Web and overlaid on the terrain appropriately. If the metadata describe another resource, such as a Web camera or MPEG video, then TerraVision may display an icon over the terrain and allow the user to click on this icon to open the media.

We demonstrated our TerraVision and GeoWeb integration to DARPA in early 2002. This involved setting up a node on DARPA's next-generation network SuperNet with a 70 GB disk containing terrain datasets. TerraVision would then contact our prototype GeoWeb server at SRI over the regular Internet. For the purposes of the demonstration, we built a database of Web camera locations and URLs. Then as a user flew over the earth, new terrain models and icons to Web cameras for those areas were automatically discovered and streamed to the display (see Figure 8).

GeoVRML

Dealing with 2-D maps and simple feature overlays has long been a common task for traditional Geographic Information Systems (GISs). However, enabling interactive, accurate, and dynamic 3-D visualizations of geospatial data that can be disseminated over the Web is only now becoming a reality. This is an important component of our Digital Earth vision because it allows rich 3-D models to be accurately represented on the surface of the planet. The GeoVRML working group of the Web3D Consortium recently announced an extension to the

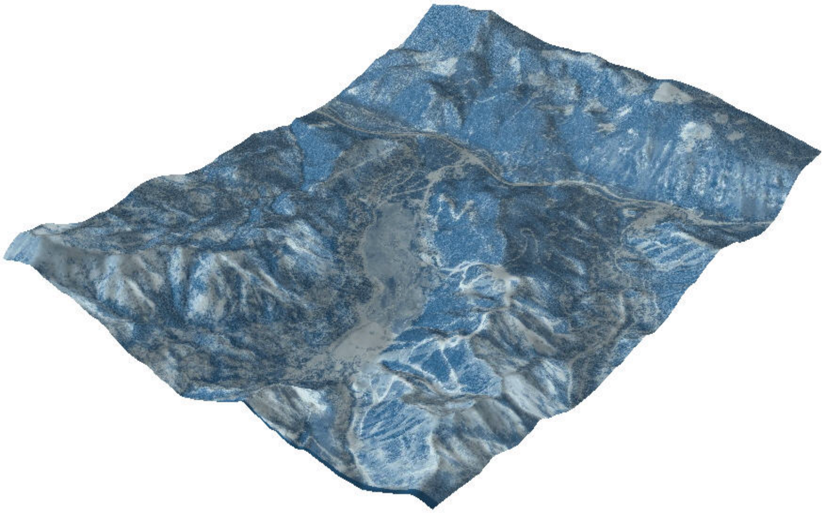


Figure 9. A GeoVRML model of Squaw Valley, California, output by ESRI's ArcInfo 8.1 product (with 3D Analyst Extension). Data provided courtesy of ESRI.

ISO standard Virtual Reality Modeling Language (VRML, 1997) to enable just these capabilities. This extension is called GeoVRML 1.0 (Reddy et al., 2001; Reddy et al., 2000a; Reddy et al., 1999b). GeoVRML is an official recommended practice of the Web3D Consortium and is included in an upcoming amendment to the VRML97 ISO standard.

GeoVRML 1.0 content can be browsed interactively in any standard VRML97 browser, a number of which exist as freely available plug-ins for common Web browsers such as Netscape Communicator and Microsoft Internet Explorer. Tools have been written to simplify the authoring and conversion of geospatial data to GeoVRML. In addition, GeoVRML is an open standard: its specification is published openly and a source-level sample implementation is provided. Here we concentrate on a few of the commercially available tools that support the GeoVRML format, and also describe some of the capabilities that this solution provides geoscientists for the purpose of integrating their geographic data directly into a 3-D computer graphics scene graph.

GeoVRML Capabilities

The following is a high-level description of the capabilities specifically addressed by GeoVRML 1.0.

1. **Geographic coordinate systems.** Most 3-D graphics systems, including VRML, use a simple Cartesian (x, y, z) coordinate system to model all data. GeoVRML provides the ability to also specify locations using geodetic or projected coordinate systems such as those commonly used in the geosciences. Specifically, GeoVRML 1.0 has support for geodetic (latitude/longitude), Universal Transverse Mercator (UTM), and geocentric coordinate systems. In addition, 21 ellipsoids and 1 geoid are supported. This support is built upon the SEDRIS Spatial Reference Model (see *www.sedris.org*).

2. **Data fusion.** GeoVRML handles data from disparate servers across the Web, generated from different sources, at different resolutions, and specified in different coordinate systems. These are all fused into a single global context for visualization. For example, the user can overlay a Global Positioning System (GPS) track of latitude/longitude coordinates over a UTM-rectified terrain model.

3. **High precision.** VRML97 provides only single-precision floating-point values. This is insufficient to represent data on a planetary scale down to around meter resolution or beyond. GeoVRML provides solutions to extending this precision to sub millimeter by employing the specification of local coordinate systems.

4. **Dataset scalability.** Terrain models often involve large elevation grids or image files. GeoVRML provides various scalability features to manage the

streaming of large, multiresolution models, thus facilitating real-time access and display of arbitrarily large terrain models.

5. **Metadata linking.** GeoVRML provides the ability to specify a generic subset of metadata describing geographic objects, including a way to link to a full metadata description.
6. **Animation support.** Key frame animations within the supported geographic coordinate systems cab be performed so that animations are defined with respect to key points on the surface of the planet.

GeoVRML Implementation

Essentially, GeoVRML 1.0 consists of 10 new extensions, or nodes, that sit on top of VRML97, that is, GeoVRML includes all of VRML97 as a subset. These nodes are defined using VRML's EXTERNPROTO extensibility features. For the purposes of the initial sample implementation, these nodes have been implemented using Java class files that are embedded within the new nodes, although we have already noted that at least one commercial VRML browser has now implemented these nodes natively. The following sections detail a small selection of the nodes provided by GeoVRML 1.0 (Reddy et al., 2000b).

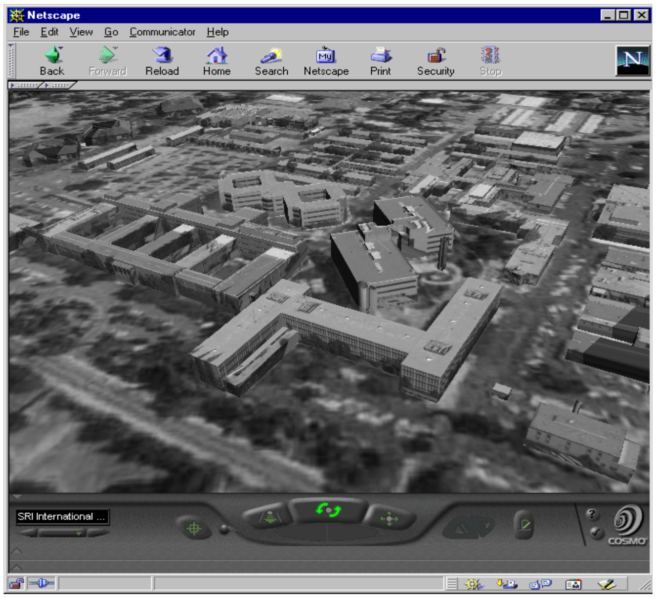


Figure 10. A texture-mapped site model of the SRI International campus that has been georeferenced onto an underlying terrain model. Note the accurate alignment of the latitude/longitude-specified buildings with the underlying 1 m resolution USGS DOQ satellite imagery in UTM coordinates.

GeoCoordinate: The GeoCoordinate node enables the specification of coordinates by using geographic coordinate systems. This node can be used within standard VRML geometry nodes such as IndexedFaceSet, IndexedLineSet, or PointSet, enabling the modeler to specify coordinates in a system such as UTM. For example, a GPS will normally output location as a latitude/longitude coordinate. With the GeoCoordinate node, we can insert these coordinates directly into a VRML file and have them integrated with any other geospatial data.

GeoElevationGrid: The GeoElevationGrid node provides the capability to define a grid of height values offset from the ellipsoid or geoid used to model the planet. It supports the specification of height fields in latitude/longitude or UTM coordinate systems. VRML97 already provides an ElevationGrid node; however, in this node all values are offset from a single flat plane. This is acceptable if the area being modeled is small in extent, for example, less than 1 km^2, but for larger areas the curvature of the earth becomes significant. Figure 10 illustrates an example showing a GeoElevationGrid.

GeoLocation: The GeoLocation node lets the user georeference an arbitrary VRML model, that is, locate it at a specific point on the earth. It also orients the model correctly, depending upon its position on the earth, so that +Y is aligned with gravitational up, +Z points true north, and +X points east. This ensures that a model built using the standard VRML right-handed coordinate system will be placed on the earth so that its base is aligned with the surface of the planet. Figure 10 illustrates the capabilities of the GeoLocation node by georeferencing models of individual buildings of the SRI International main campus to an underlying terrain model of Menlo Park, California.

GeoLOD: The GeoLOD node provides the capability to browse multiresolution, tiled terrain data that are streamed over the Web. It automatically manages the progressive loading of higher-resolution data as the user approaches the terrain, and also unloads terrain data that the user has flown past. These are essential memory management and scalability operations for browsing massive terrain datasets. For example, Figure 11 illustrates a large multiresolution dataset that is built using the GeoLOD node. We illustrate the capability to fly down through several levels of detail while higher-resolution data are streamed over the Internet to the user's display.

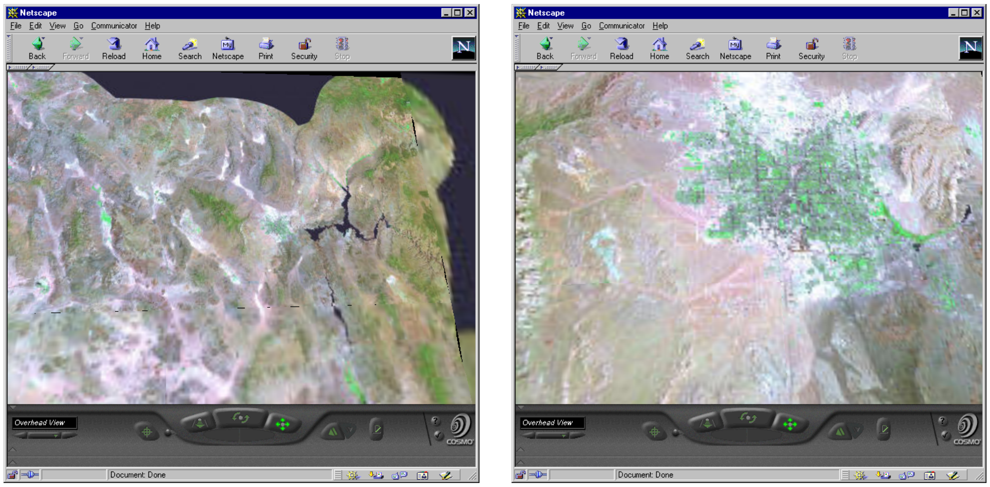

Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Overhead View
Document: Done
Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Overhead View
Document: Done

Figure 11. An example showing the scalability features provided by the GeoLOD node. The two images show a user's descent into a terrain model of the Mojave desert in southern California. The source imagery and elevation data for this terrain model total 1.3 GB.

GeoPositionInterpolator: One of the many strengths of VRML is its ability to model dynamic systems. GeoVRML therefore incorporates the capability to animate models using geographic coordinates. This is implemented through the GeoPositionInterpolator node, which functions in much the same way as the standard VRML97 PositionInterpolator node, except that the key frame values can be specified using geographic coordinates. For example, if a GeoPositionInterpolator is created and given two geodetic coordinates, (122.413 W, 37.7483 N, 10000) and (116.388 E, 39.906 N, 10000), and the output is routed to a VRML model of a Boeing 777, then this aircraft would fly over the surface of the planet, from San Francisco to Beijing, at a constant altitude of 10,000 m. Using this capability, a vehicle could be animated based upon the list of latitude/longitude coordinates from a prerecorded GPS track, or from a live feed of Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs).

GeoVRML Tools

Several companies have developed tools that support the GeoVRML file format. These include modeling tools that can export to GeoVRML, conversion tools to take standard mapping products and produce GeoVRML representations of these, and visualization technologies that allow the user to browse GeoVRML content. These tools include the following; more examples can be found from the GeoVRML home page (*www.geovrml.org*).

ArcInfo / ArcView: With version 8.1 of the 3-D Analyst extension for the ArcView and ArcInfo products, released in April of 2001, ESRI added GeoVRML support to their range of capabilities. These products now allow export of GIS data to the GeoVRML file format for viewing over the Web. Figure 9 was created with ArcInfo 8.1 by combining an orthographic image (retouched to have a blue color) with a Digital Elevation Model (DEM) of the Squaw Valley ski resort, near Lake Tahoe. The ESRI home page is *www.esri.com*.

ShapeViz: Bashir Research produces a tool that allows the user to view Shape files. This file format was created by ESRI as a way to interchange geometry data and is supported by ESRI's range of GIS products. ShapeViz can convert Shape files to VRML for interactive visualization over the Web. With version 1.2 of ShapeViz, GeoVRML export support was added, along with the ability to specify the coordinate system of the Shape file (as this information is not ordinarily included in the Shape file). Figure 12 shows an example Shape file converted to GeoVRML using the ShapeViz tool. The Bashir Research home page is *www.my3d.com*.

Cortona: ParallelGraphics produces a popular VRML browser caller Cortona, which is provided as a plug-in for Netscape and Internet Explorer. With version 3 of the Cortona browser, ParallelGraphics added a native implementation of the GeoVRML extensions. The implementation for these nodes can be automatically downloaded and installed when GeoVRML nodes are first used. The original GeoVRML implementation is written in Java so that it can run within any VRML browser that supports Java as a scripting language. With a native implementation, the Cortona browser offers more integrated and efficient support for this geospatial format. Figure 12 shows the converted Shape file being viewed with the Cortona browser. The Cortona browser can be found at *www.parallelgraphics.com*.

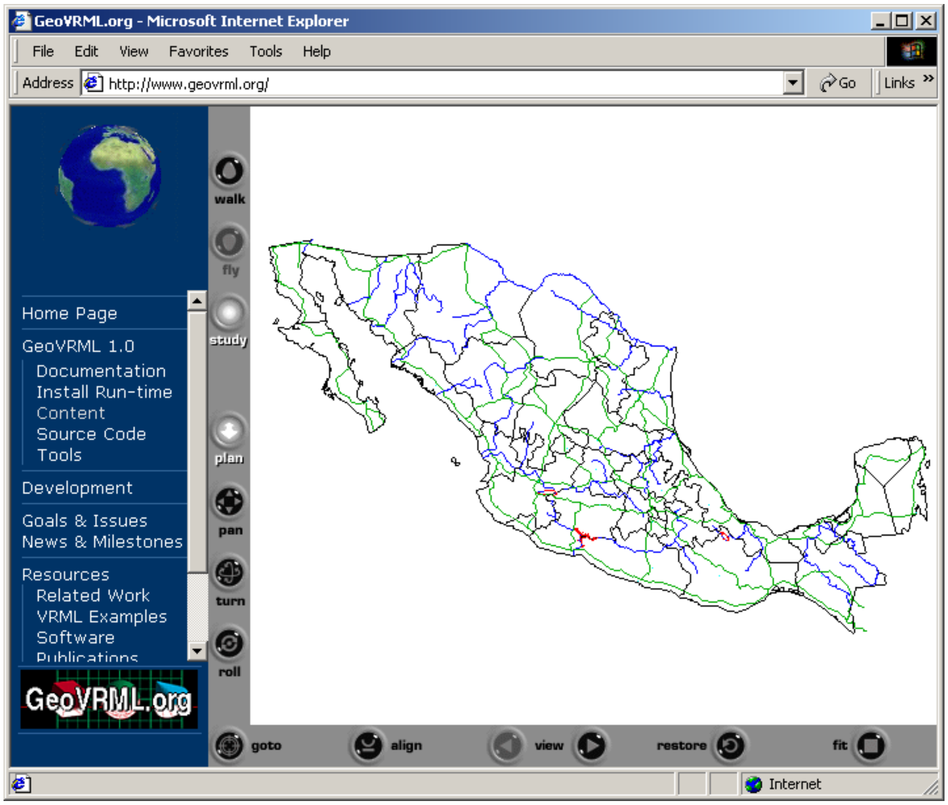


Figure 12. A GeoVRML translation of a Shape file produced by the ShapeViz utility from Bashir Research. This shows a number of layers for Mexico, including state boundaries, rivers, roads, lakes, and cities. This is displayed in ParallelGraphics' Cortona 3.1 browser. Model provided courtesy of Bashir Research.

GeoVRML Standards Positioning

GeoVRML 1.0 is currently an accepted recommended practice of the Web3D Consortium. In addition to this, the GeoVRML nodes have been submitted to ISO as part of an amendment to the VRML97 specification. The intention is that these capabilities will eventually become an optionally implementable part of the current VRML ISO specification, and also an additional profile in any future revisions of the specification.

The GeoVRML work is of major relevance to several exciting initiatives that are currently evolving. For example, this work provides the technical foundation to build the vision of a highly accurate and large-scale model of the earth into which we can embed massive quantities of georeferenced data; a vision that we refer to here as the Digital Earth (Leclerc et al., 1999; Reddy et al., 1999a; Reddy et al., 1999b). Related to this topic, the OpenGIS Consortium (OGC) is evolving its Web Mapping Testbed (WMT) technology as a means to revolutionize the use of geospatial data on the Web (de La Beaujardière, 2002). Its efforts to date have focused largely on 2-D presentations and also on the cataloging of geographic data. As such, the GeoVRML work is well positioned to integrate with the OGC efforts and add Web-based 3-D visualization capabilities to this initiative; for example, SRI has produced an OGC-compliant Web Map Server (WMS) that can produce results in GeoVRML format as well as the more common GIF and JPEG image formats.

In addition, the GeoVRML working group is contributing to the evolving X3D development. X3D is the next-generation VRML specification and supports extensibility through profiles that can be plugged into a core browser (Daly and Williams, 2002; Bietler et al., 2002). The GeoVRML nodes have already been added as a Geospatial profile for X3D, and a document type definition (DTD) for this profile has been developed. For further information, see the X3D Web page at *www.web3d.org/x3d.html*.

Urban Modeling

As part of SRI's Digital Earth effort, we have built a variety of large 3-D urban models. This work was performed by Planet 9 Studios, which was tasked with

1. Developing a production approach, using commercial off-the-shelf (COTS) tools, to build accurate digital cities
2. Creating a white paper on how to build digital cities
3. Building reference 3-D datasets of New York and Washington D.C.
4. Consulting with SRI's Digital Earth team on infrastructure issues from the perspective of a production studio

Production Approach

SRI's Digital Earth framework for 3-D models (later called GeoVRML) enables the referencing of 3-D urban datasets to an earth ellipsoid. This provides higher accuracy to the real world but also offers many challenges, as all current consumer tools are based on a "flat earth" premise. There are tools used within the military and mapping industries that allow construction of data on an ellipsoid, but they are far from mainstream. It was decided that either geo-referencing plug-ins needed to be developed for mainstream tools or the infrastructure needed to be flexible enough to accommodate data created in different authoring systems. In the end, the GeoVRML extensions were written to allow "flat earth" datasets to be referenced by centroid as well as for other datasets to be referenced by either latitude/longitude or UTM corner points. The most common COTS 3-D authoring tool is 3-D Studio Max, which was used for the creation of the reference datasets. Authoring techniques are described in the "Building Digital Cities" white paper.

Building Digital Cities

In the early days of 3-D computer graphics, most critical developments came from military simulation needs. In recent years, the game industry has become the leading developer of cutting edge technology and the generic desktop PC has become the display medium of choice. The goal of the "Building Digital Cities" white paper was to examine the techniques used by the game industry to produce high-performance, low-bandwidth simulation while maintaining good accuracy and data registration. It was discovered that little has been written about these techniques and that nothing has been created to describe the level of resolution and detail of urban datasets. The paper includes basic proposed standards to allow classification of various types of urban data.

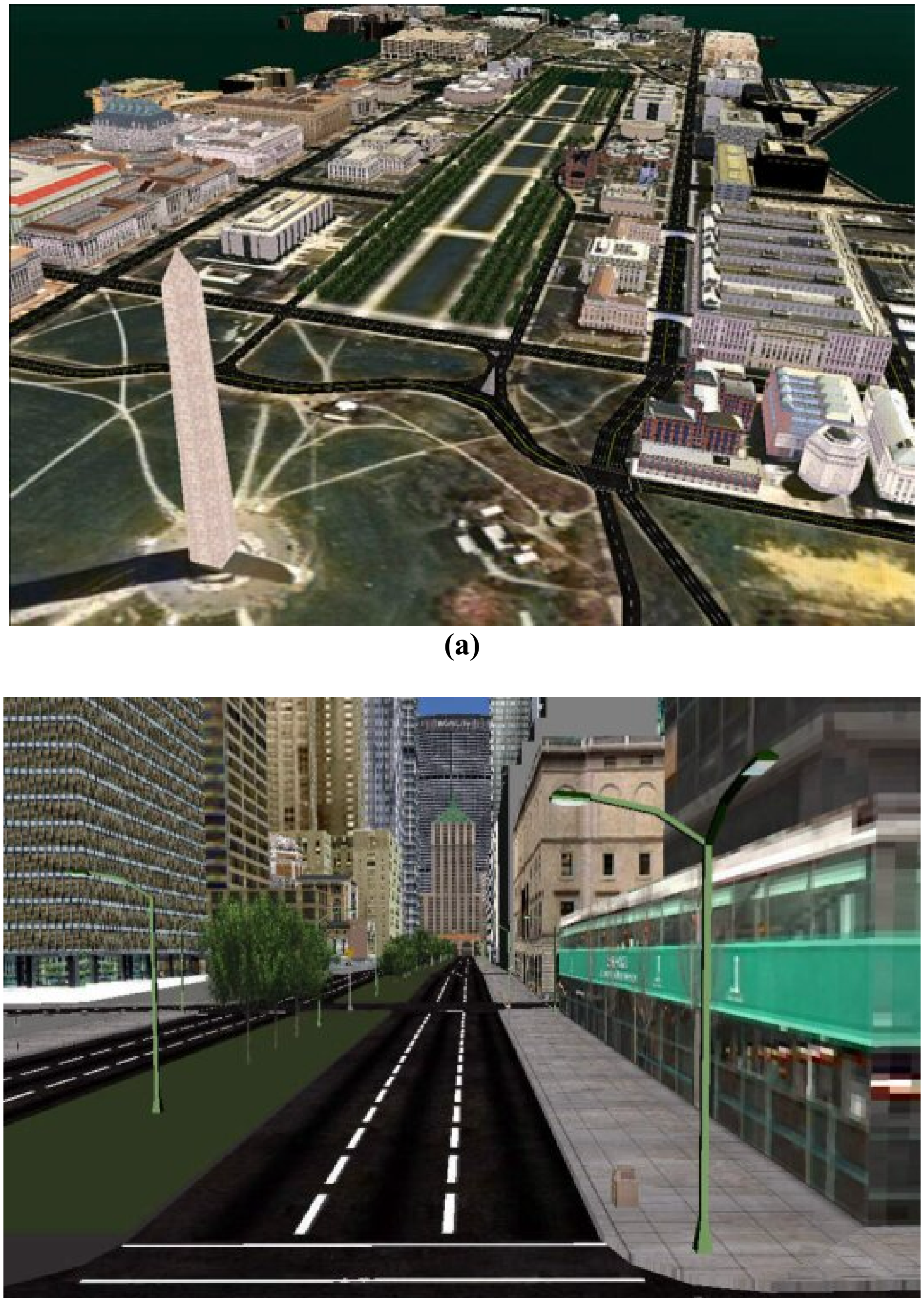
Reference Datasets

It was decided to build reference datasets (Midtown Manhattan and Washington D.C., see Figure 13) of two types of dense urban data

- To show what dataset size and resolution is practical to be built and displayed via the Internet
- To show how efficient data are built using game industry techniques
- To act as a performance benchmark for viewing environments particularly as they develop enhancements to better handle large datasets
- To showcase GeoVRML functionality

Infrastructure Consultation

One reason that SRI asked Planet 9 Studios to participate in the Digital Earth project was to gain an industry perspective on the issues to be encountered and addressed. Planet 9 Studios contributed, in regular team meetings and e-mail exchanges, with views and suggestions at all levels of development during the 3-year project. The focus of Planet 9 Studios on practical and commercially driven issues helped to provide a good grounding to the overall project.



(a)

(b)

Figure 13. Two large 3-D urban models built by Planet 9 Studios as part of the Digital Earth project. (a) Covers the National Mall of Washington, D.C., with around 113,000 polygons and 645 textures, and (b) covers the area of New York City between Madison Avenue to Lexington Avenue and 41st to 54th streets, with 115,000 polygons and 222 textures.

Educational Application

SRI International is working to create Digital Earth applications specifically crafted for education. Using the TerraVision browser, students seamlessly access data sets of satellite imagery and aerial photography to swoop into the earth as a whole and then focus in on a single building. In our educational application, students "fly around" to find an area of focus. As the users fly closer to the earth, higher-resolution data sets are brought in, giving an impression of visual continuity. Once a region of interest has been chosen, the student can analyze it using an allied visualization and data analysis environment based on Java VisAD components.

In addition, educational applications of TerraVision provide opportunities for students to author within the space, analyze what they find there, and, ideally, explore natural phenomena via simulations. Authoring within the space can be done by placing VRML objects; adding points, lines, and polygons (e.g., where they would like to hike or a study site for future work); and adding annotations via hypertext links to text, images, animations, and sounds. Analysis requires the ability to graph variables, compute new data sets, derive aggregate values, and calculate relationships. Simulations require an underlying model of phenomena that the student manipulates, often by perturbing some initial set of conditions and letting the simulation run for a set amount of time. The deployment of education applications can occur in an academic setting (e.g., high school classrooms), for virtual tourism (e.g., Internet-based exploration), or for travelers en route (e.g., visitor information kiosk).

The advanced data discovery and presentation capabilities offered by the Digital Earth could prove beneficial to application scenarios such as

- **Virtual tourism**. Tourists could use the Digital Earth to plan a holiday or experience the attractions at a remote site before actually visiting the place.
- **Crisis management**. Emergency teams could use dynamic high-resolution 3-D terrain representation tools to help oversee, plan, and coordinate disaster relief efforts, for example, for forest fire fighters and earthquake relief workers.
- **Historical visualization**. Users could navigate through the fourth dimension to view a depiction of a place over time. For example, to see how the Ghost town of Rhyolite, Nevada, changed over the years since its original planning in 1905.
- **Sense of place and scale**. The multiresolution capabilities of the Digital Earth could be used to relate local phenomena to global phenomena. For example, where does a town's water come from? Or, where does its sewage go?

- **Wilderness Education**. Rich 3-D models for remote sites could be used to improve education about some of our county's natural resources such as national parks and wilderness areas.

To demonstrate the utility of our Digital Earth technologies, we developed an educational application focus centered on the wilderness education scenario described above. This effort was driven by members of the Center for Technology and Learning (CTL) at SRI International.

Wilderness Education

We chose Yosemite National Park, California, as the site for our educational outreach because of its prestige, diversity of resources, and large number of visitors. We worked with the Yosemite National Park Service, U.S. Geological Survey (USGS) researchers, and educators in the park to provide a Virtual Yosemite Community where users could use Digital Earth to interact with and learn about the park. The format of this educational program took on three stages, described below, that helped to maximize the experience gained by students and also to reinforce what they had learned during a field trip to the location.

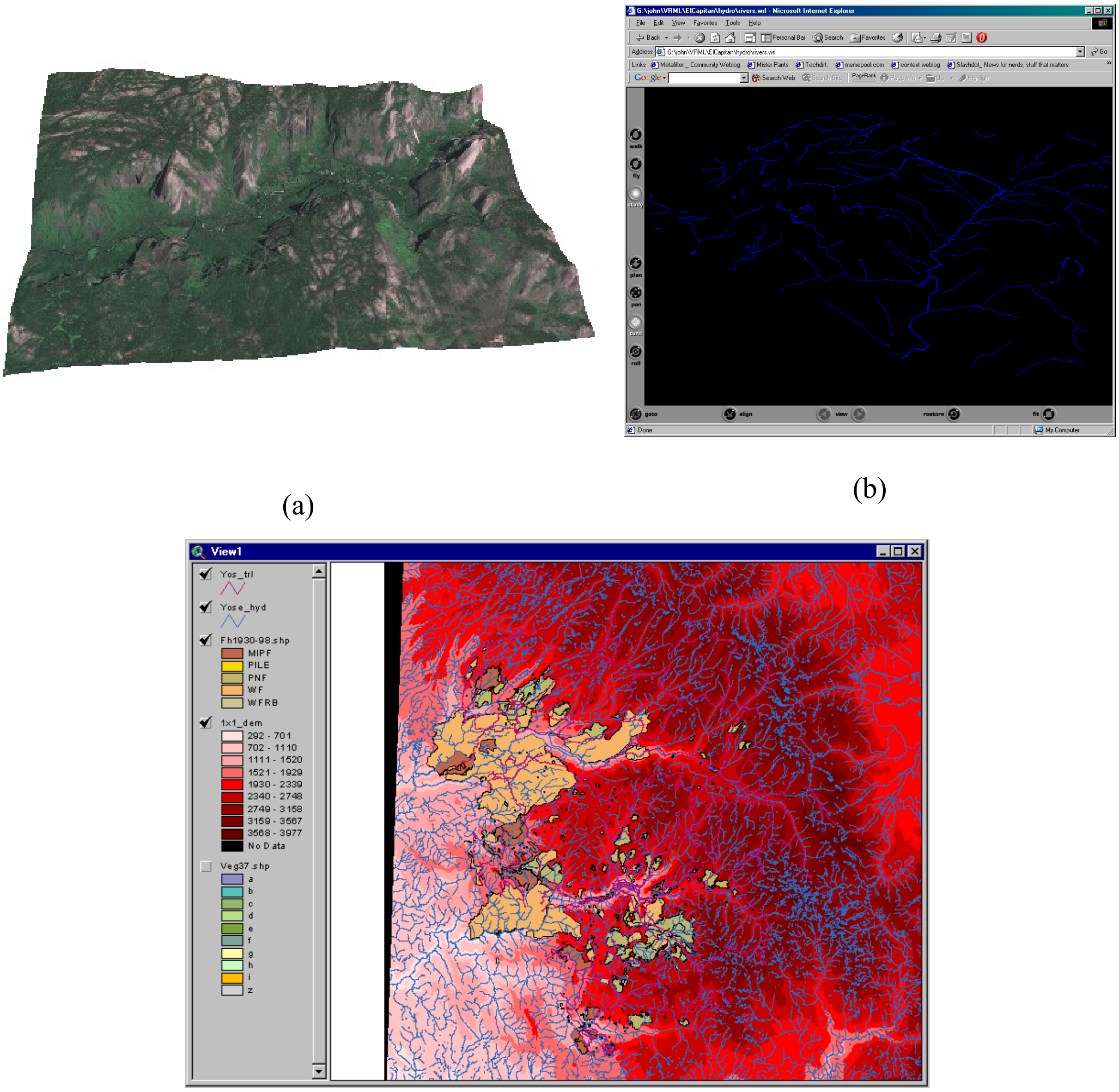
1. **Pre-trip**. Prior to a visit to a wilderness area, students can tangibly get the "lay of the land" via the virtual experience of the terrain delivered by TerraVision. They can also access information in place describing the area, such as hydrology data, cultural information, and media. These data can be used to plan the trip and any investigations to be performed during the visit.
2. **Trip**. While visiting the wilderness area, students make observations (scientific observations, media recordings, and personal observations) and use GPS units to georeference these observations.
3. **Post-trip**. After the visit, students can publish their observations to the Digital Earth by using a Geoster-like tool. They can then revisit the virtual depiction of the area they visited both to see their own observations in place, and to reassess the depiction conveyed by the Digital Earth relative to their actual experience of the place. From here on, the Digital Earth can be used as a platform for dialog about the place both among students who made the trip together and between various student groups that visit the place over time.

WildLink

We selected WildLink as the ideal wilderness education organization with which to work for our purposes. WildLink consists of students, teachers, and park rangers who are already technology aware and intent on using the Web and GIS techniques. In particular, WildLink has a focus to reach students with backgrounds that are normally underrepresented in the Park's vistorship. In effect, WildLink became a GLOBE school collecting environmental data in Yosemite National Park. We worked with this organization to develop our educational scenario and provide an ideal test of the Digital Earth as a tool to convey a sense of place and to inspire students to learn about a wilderness site in a compelling and exciting manner.

Data Collection

The foundation data for the Virtual Yosemite model consisted of the following items (see Figure 14). These were collected from a range of sources as well as some data being created and/or processed in-house.



(a)

(b)

(c)

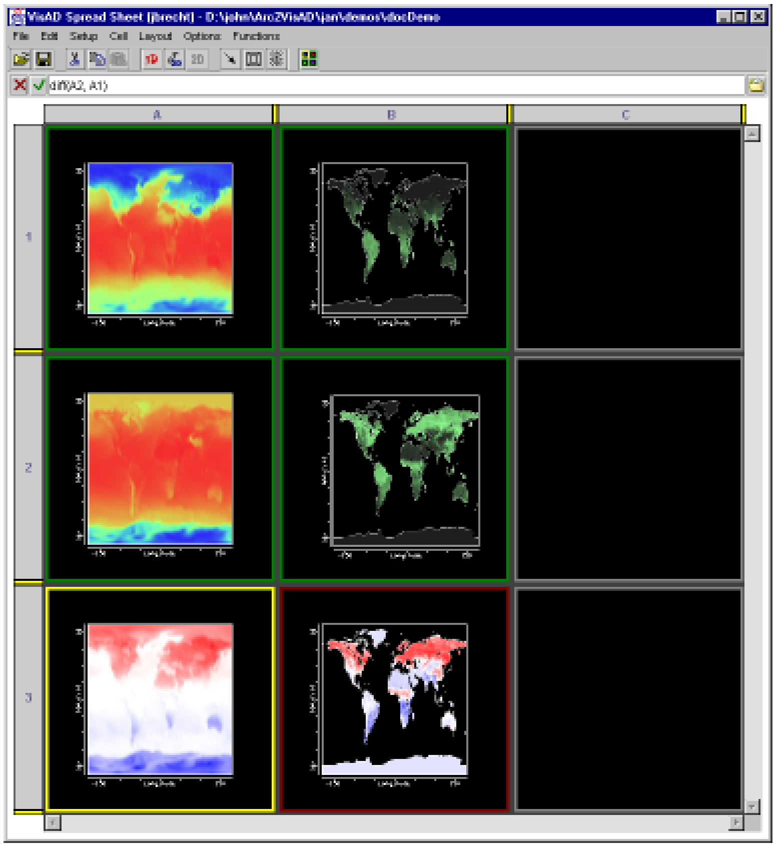
Figure 14. Examples of the foundation data used in the Virtual Yosemite educational scenario. (a) Satellite imagery, (b) river vector data, and (c) GIS data layers.

- **Imagery**: 15-m Landsat image of Yosemite National Park and environs, 1-m aerial imagery of selected areas
- **GIS data**: 30-m DEM, historical forest fire regions, geology, hydrology, vegetation, and cultural layers
- **The WildLink study area**: the area extended outside of the Park slightly, so polygon layers from Yosemite National Park had to be merged with layers from Sierra National Forest.
- **GeoVRML**: for display in TerraVision, GIS data layers were converted into GeoVRML objects.

Tool Development

We developed custom to support the educational needs of the project. Two of the major such tools were GeoVRML-DB and TerraAnalysis.

- **GeoVRML-DB**. This is a system for querying a database of georeferenced point values and representing the results of the query as dynamically generated GeoVRML objects. These objects can then be displayed, such as in our TerraVision browser over an appropriate terrain model. For example, a query could be made to the database to return temperature samples over a specified area for a certain period of time and have these results displayed in TerraVision. This has the benefit of encouraging investigative studies of the data through the use of intuitive and interactive 3-D interfaces. The system was tested and demonstrated to run in the Microsoft Internet Explorer Web browser, querying a remote MySQL database. The results could be displayed in place using the Cortona VRML browser (or other commercial systems) as an alternative to the TerraVision system.
- **TerraAnalysis**. To provide analysis functionality for our educational application, we developed TerraAnalysis. TerraAnalysis is a 5-D visualization and analysis environment for gridded, georeferenced datasets such as climate data (see Figure 15). TerraAnalysis was built using the VisAD Java component library and uses a spreadsheet as an organizational and computational metaphor. Each cell in a TerraAnalysis spreadsheet contains a complete data set, as opposed to a single data point, and all cells in a single sheet will contain data corresponding to the same geographic region. Data in any cell are easily mapped to three spatial dimensions, color, and time. TerraAnalysis provides convenience functions for common mappings such as relief maps or contour maps. Spreadsheets are familiar to most students and thus TerraAnalysis builds on existing skills to quickly ramp students up to complicated computational notions such as regional averages or differences. See *vgc.sri.com/projects/DigEarth/TerraAnalysis/*.


VisAD Spread Sheet [jbrecht] - D:\john\Arc2VisAD\jan\demos\docDemo
File
Edit
Setup
Cell
Layout
Options
Functions
A
B
C
1
2
3

diff(A2, A1)

Figure 15. The TerraAnalysis visualization and analysis environment.

TECHNOLOGY TRANSFER

The .geo Effort

On October 2, 2000, SRI International submitted an application to the Internet Corporation for Assigned Names and Numbers (ICANN) for a new top-level domain name (TLD) called ".geo." The goal of this proposal was to position our GeoWeb architecture as the primary means to discover information on the Web based upon location; in effect, to become the latitude and longitude of the Internet's virtual world. We argued that .geo would make the Internet more efficient, both technically and as a tool for human communication and commerce. It also would open new global and local opportunities for science, commerce, education, and enterprises – even in the developing world, where such civic and commercial activity has in the past been technologically or economically difficult or impossible.

In total, ICANN received 47 applications for new TLDs. To help focus these applications, ICANN issued a list of criteria for assessing TLD proposals. ICANN specified the following relevant factors:

1. The need to maintain the Internet's stability
2. The extent to which selection of the proposal would lead to an effective "proof of concept" concerning the introduction of top-level domains in the future
3. The enhancement of competition for registration services
4. The enhancement of the utility of the DNS
5. The extent to which the proposal would meet previously unmet types of needs
6. The extent to which the proposal would enhance the diversity of the DNS and of registration services generally
7. The evaluation of delegation of policy-formulation functions for special-purpose TLDs to appropriate organizations
8. Appropriate protections of rights of others in connection with the operation of the TLD
9. The completeness of the proposals submitted and the extent to which they demonstrate realistic business, financial, technical, and operational plans and sound analysis of market needs

SRI's .geo effort engendered a large degree of positive support and interest, including written support from the United Nations Conference on Trade and Development (UNCTAD). SRI produced a press release to bolster the .geo application and our bid was covered in a range of journals and national newspapers. ICANN's intellectual property group, IPC/DNSO, reviewed all 47 TLD applications and gave .geo the highest rating. Furthermore, ICANN hired technical reviewers to access the merits of each application. They ultimately recommended a single specific TLD, .geo, and made the following comments, among others:

> *The novelty of the proposal makes it stand out from the vast majority of all other proposals.*
>
> *.geo proposal will help to "humanize the web."*
>
> *This is a revolutionary proposal that suggests an entirely new style of usage for the Internet and its naming system. The use of the name system to point to locations where data is stored based on geographical information is extremely novel.*

The final decision was made by ICANN at its open board meeting in Marina del Rey, California, on November 16, 2000. The board made an initial selection of nine TLDs, which included the .geo application. However, after some further discussions, the board removed the SRI .geo and Sarnoff .i applications, eventually resting on seven new TLDs: .aero, .biz, .coop, .info, .museum, .name, and .pro. It was felt that SRI's .geo application would be in a strong position for a second round of TLD applications. No such second round has been announced to date and it is not clear when or whether this will occur. A website describing SRI's .geo TLD bid is maintained at *www.dotgeo.org*.

Software Products

Published Software Packages

SRI's Digital Earth technologies encompass a range of software products, practically all of which we have made available to the general public at no cost, including several Open Source releases. We have done this in an effort to make our work more accessible and widely applicable, and also to build a community of users who can provide feedback and suggestions. The following software packages have been made available.

- **TerraVision**. Users can download the TerraVision binary, with GeoWeb support, for Windows platforms, Linux, and Silicon Graphics. The full C/C++ source code is also freely available. In addition to this material, we provide extensive documentation in the form of a User's Manual and FAQ, a mailing list for discussion of TerraVision issues, and access to all the terrain datasets that we have generated. The website for this software is *www.tvgeo.com*.
- **GeoVRML**. The GeoVRML effort was formed and spearheaded by SRI International, although now it encompasses a large international community. SRI wrote the majority of the GeoVRML Java implementation and this is hosted at SRI. Users can download the GeoVRML runtime environment and also the full Java source code. This includes the GeoTransform package that performs various fast geographic coordinate transformations. See *www.geovrml.org*.
- **tsmApi**. The Tile Set Manager Application Program Interface (tsmApi) is an Open Source library of high-level C functions for reading, writing, and processing the terrain data used by the TerraVision terrain visualization application, released under the Apache license. The tsmApi library can ingest standard mapping products and produce GeoVRML representations of these terrain models. Specific formats that are supported include USGS DEM and DOQ, DTED level 0, GeoTIFF, TIFF, GIF, JPEG, PPM, and PGM. For further details, refer to *www.tsmapi.com*.

- **DEM2GeoEG**. DEM2GeoEG is a program to convert USGS DEM data into a VRML .wrl file that uses the GeoVRML 1.0 GeoElevationGrid (see below). The GeoElevationGrid is a version of the standard ElevationGrid but it lets the user georeference the data to geographic coordinate systems such as UTM. One important benefit of this is that multiple GeoElevationGrids can be fused into a single scene and they will be correctly located with respect to each other. Figure 16 illustrates a USGS DEM converted to GeoVRML using this utility. The program and source code can be found from the GeoVRML home page at *www.geovrml.org*.

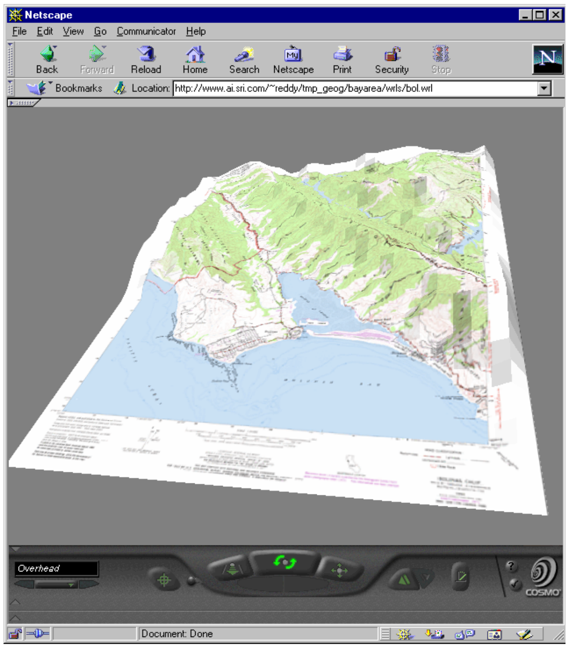

Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://www.ai.sri.com/~reddy/tmp_geog/bayarea/wrls/bol.wrl
Overhead
COSMO
Document: Done

Figure 16. A USGS DEM that has been converted to a GeoElevationGrid using the dem2geoeg utility, and then draped with a DRG image for the same area.

Meetings and Articles

During the project, the Digital Earth technologies presented in this report were demonstrated at several venues, including the following conferences and meetings:

- DARPA NGI Kick-Off Meeting for the NGI Program.
- DARPA NGI PI Meeting in Washington D.C., 2-4 October 2000. Installed a 70 GB disk of TerraVision datasets at ISI on the SuperNet. Configured and ran TerraVision over SuperNet.
- Optical Fiber Conference (OFC 2000) at Baltimore, 6-10 March 2000. Manned a TerraVision booth for 3 days as part of the DARPA NGI exhibit space.
- 5th International Conference on Virtual Systems and Multimedia (VSMM'99), Dundee, Scotland, 1-3 September. Presented paper on our early Digital Earth vision.
- Web3D RoundUp at SIGGRAPH 2000 in New Orleans, 23-28 July 2000. Gave a live quick-fire TerraVision demonstration for more than 2,000 conference attendees.
- GIScience 2000, Savannah, Georgia. October 28-31, 2000. Presented a paper on GeoWeb.

- Digital Earth Working Group meeting, Santa Barbara, 12-13 January 2000. Presented TerraVision and GeoVRML work.
- Discrete Global Grids (DGG 2000) conference in Santa Barbara, 25-28 March 2000. Presented GeoWeb technology.
- Web3D-VRML 2000 conference. Presented a paper and tutorial on GeoVRML (Best Paper Award).

In addition to these venues, SRI's Digital Earth technologies have been featured in a range of journals and articles. See the references section at the end of this report for details of all the publications that we have produced from this work. The following articles have been written by others, and include national press and television coverage.

- *New York Times*, November 16, 2000. "Agency to Vote on Web Domain Names" by Chris Gaither
- *MIT Technology Review*, January/February 2001. "Innovation: Where in the World?" by Alexandra Stikeman
- *New York Times*, April 5, 2001. "New Rules for Net Searches: Location, Location, Location" by Catherine Greenman
- TechTV Cable TV Channel, June 8 2001. "Download 3-D TerraVision Movies"
- Cable News Network, CNNDotCom, June 16, 2001. "High-tech Originators Unveil New Creations"
- *Scientific American*, October 2001, "A Wide Web of Worlds" by W. Wayt Gibbs

SUMMARY OF ACHIEVEMENTS

As a fundamental new service on top of the Web, the GeoWeb provides a unique solution for harnessing the power of the Internet. It is based on the way human beings perceive and comprehend their world: geospatially, in three dimensions, and over time. The GeoWeb enables Internet users to navigate, access, and visualize georeferenced data as they would in a physical world, but without the barriers imposed by space and time in the physical world (see *www.dotgeo.net*).

TerraVision provides a freely available and cross-platform capability to browse massive terrain datasets in 3-D. It can browse huge datasets, in the order of terabytes, and these data can be distributed over multiple servers across the Web. It is available as a full application, a Netscape plug-in, and an ActiveX component. The system supports the overlay of 3-D GeoVRML models and offers an OGC-compliant Web Mapping Server interface. The tools to generate TerraVision datasets are released as Open Source for anyone to download and augment (see *www.tvgeo.com*).

GeoVRML provides the geosciences field 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 3-D geographic data that can be distributed over the Web and interactively visualized using a standard browser configuration. With the presence of this standard, and the recent emergence of many low-cost 3-D accelerator cards for common desktop computers, the scene is now set to allow researchers, educators, and businesses to publish their rich, 3-D geospatial products to a wide audience over the Web (see *www.geovrml.org*).

ACKNOWLEDGMENTS

We would like to acknowledge the work of a number of people who have contributed to SRI's Digital Earth project over its lifetime. Lee Iverson, now at the University of British Columbia, contributed greatly to the project and was the lead developer of TerraVision for a number of years. Aaron Heller of SRI International provided his expertise in the areas of terrain and site modeling and created a large amount of data for us to use. Marie Bienkowski at SRI International provided valuable input and direction on educational applications. Michael Summers, Bob Jacobson, and Catherine Jaeger provided important market and commercial insights for location-based services centered on the GeoWeb concept. Finally, we would like to thank our program managers at DARPA, Mari Maeda and Gary Koob, for their support during this exciting work.

REFERENCES

A. Gore. "The Digital Earth: Understanding Our Planet in the 21st Century". Speech delivered at the California Science Center (CSC), Los Angeles, California, 31 January 1998.

M. Bietler, M. Bourges-Sevenier, D. Brutzman, P. Diefenbach, T. Parisi, M. Reddy and D. Silverglate (2002). "X3D Architecture and Profiles". Tutorial at the 7th International Conference on 3D Web Technology, 24-28 February, 2002.

L. Daly and J. Williams (2002). "Introduction to X3D". Tutorial at the 7th International Conference on 3D Web Technology, 24-28 February, 2002.

L. DeFloriani (1989). "A Pyramidal Data Structure for Triangle-based Surface Description". *IEEE Computer Graphics and Applications*, 9(2), pp. 67-78.

J. de La Beaujardière (Editor). "OpenGIS Web Map Server Interface Implementation Specification", 16 January 2002.
www.opengis.org/techno/implementation.htm

L. L. Hill, G. Janée, R. Dolin, J. Frew and M. Larsgaard (1999). "Collection Metadata Solutions for Digital Library Applications", *Journal of the American Society for Information Science*, 50(13): 1169-1181.

Y. Leclerc, M. Reddy, L. Iverson and M. Eriksen (2001a). "The GeoWeb (aka dot-geo) --- Indexing Data on the Internet by Location". Digital Earth 2001, New Brunswick, Canada, 24-28 June 2001.

Y. Leclerc, M. Reddy, L. Iverson and A. Heller (2001b). "The GeoWeb --- A New Paradigm for Finding Data on the Web". In *Proceedings of the International Cartographic Conference (ICC2001)*, Beijing, 6-10 August 2001.

Y. Leclerc, M. Reddy, L. Iverson, M. Eriksen and A. Heller (2000). "The .geo-web: A Scalable Index for the Digital Earth". In *Proceedings of GIScience 2000*, Savannah, GA. 28-31 October, 2000.

Y. Leclerc, M. Reddy, L. Iverson, and N. Bletter (1999). "Digital Earth: Building the New World". *In Proceedings of the 5th International Conference on Virtual Systems and Multimedia (VSMM'99)*, pp. 250-262. 1-3 September 1999. Dundee, Scotland.

Y. G. Leclerc and S Q Lau (1994). TerraVision: A Terrain Visualization System. Technical Report 540. SRI International. Menlo Park, CA. April 1994.

D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson and R. Huebner (2002). "Level of Detail for 3D Graphics", Morgan Kaufmann Publishers, San Francisco, CA.

M. Reddy, L. Iverson, Y. Leclerc and A. Heller (2001). "GeoVRML: Open Web-based 3D Cartography". In *Proceedings of the International Cartographic Conference (ICC2001)*, Beijing, 6-10 August 2001.

M. Reddy, L. Iverson and Y. G. Leclerc (2000a). "Under the Hood of GeoVRML 1.0". In *Proceedings of Web3D-VRML 2000: The Fifth Symposium on the Virtual Reality Modeling Language*. Monterey, CA. 21-24 February 2000, pp. 23-28.

M. Reddy, L. Iverson and Y. Leclerc (2000b). "GeoVRML 1.0: Adding Geographic Support to VRML". *GeoInformatics*, Volume 3, September 2000.

M. Reddy, Y. G. Leclerc, L. Iverson, N. Bletter and K. Vidimce (1999a). "Modeling the Digital Earth in VRML". In *Proceedings of SPIE - The International Society for Optical Engineering*. Volume 3905, pp. 113-121.

M. Reddy, L. Iverson, and Y. G. Leclerc (1999b). "Enabling Geographic Support in Virtual Reality Modeling with GeoVRML", *Cartography and Geographic Information Science*. 26(3):180.

M. Reddy, Y. G. Leclerc, L. Iverson and N. Bletter (1999). "TerraVision II: Visualizing Massive Terrain Databases in VRML". *IEEE Computer Graphics and Applications*, 19(2): 30-38.

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

APPENDIX: GEOWEB RDF SCHEMA

The following Resource Description Framework (RDF) Schema file describes the metadata that are indexed in the GeoWeb, and which can be searched by GeoWeb clients. We chose to represent these GeoWeb metadata using RDF Schema so that it could support the inclusion of additional ontologies over time, and also so that it would match closely to the Semantic Web work being performed in the W3C.

```
<?xml version="1.0" encoding="utf-8"?>

<rdf:RDF xml:lang="en"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#">

<!-- Note: this RDF schema would typically be used in RDF instance data
     by referencing it with an XML namespace declaration, for example
     xmlns:xyz="http://www.w3.org/2000/03/example/vehicles#".
     This allows us to use abbreviations such as xyz:MotorVehicle to
     refer unambiguously to the RDF class 'MotorVehicle'. -->

  <!-- Identification -->

  <rdf:Description ID="identification">
    <rdfs:comment>
      The identification bucket.
      These properties will consist of unique identifiers, such as VIN
      numbers for automobiles, RF tag IDs, etc.
    </rdfs:comment>
    <rdf:type resource="rdfs:Property"/>
    <rdf:range resource="#identifier"/>
  </rdf:Description>

  <rdf:Description ID="Identifier">
    <rdfs:comment>
      The possible contents of the Identification bucket.
    </rdfs:comment>
    <rdf:type resource="rdfs:Class"/>
  </rdf:Description>
  <rdf:Description ID="identity">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#Identifier"/>
  </rdf:Description>

  <rdf:Description ID="location">
    <rdfs:comment>
      The geographical extent bucket.
    </rdfs:comment>
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="#location"/>
  </rdf:Description>

  <!-- Location -->

  <rdf:Description ID="Location">
    <rdfs:comment>
      The possible contents of the location bucket.
    </rdfs:comment>
    <rdf:type resource="rdfs:Class"/>
  </rdf:Description>

  <rdf:Description ID="GeoLocation">
    <rdfs:comment>
      The geolocation of an object, expressed in latitude and longitude.
    </rdfs:comment>
    <rdf:type resource="rdfs:Class"/>
    <rdf:subClassOf resource="#Location"/>
  </rdf:Description>
  <rdf:Description ID="north">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:double"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>
  <rdf:Description ID="south">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:double"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>
  <rdf:Description ID="east">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:double"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>
  <rdf:Description ID="west">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:double"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>
  <rdf:Description ID="minAltitude">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:double"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>
  <rdf:Description ID="maxAltitude">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:double"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>
  <rdf:Description ID="planet">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#GeoLocation"/>
  </rdf:Description>

  <rdf:Description ID="Dynamics">
```

```
    <rdfs:comment>
      The dynamic attributes of an object.
    </rdfs:comment>
    <rdf:type resource="rdfs:Class"/>
    <rdf:subClassOf resource="#Location"/>
  </rdf:Description>
  <rdf:Description ID="updateTime">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:dateTime"/>
    <rdfs:domain resource="#Dynamics"/>
  </rdf:Description>
  <rdf:Description ID="velocity">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range> <xsd:list itemType="xsd:double" length="3"/>
</rdfs:range>
    <rdfs:domain resource="#Dynamics"/>
  </rdf:Description>
  <rdf:Description ID="acceleration">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range> <xsd:list itemType="xsd:double" length="3"/>
</rdfs:range>
    <rdfs:domain resource="#Dynamics"/>
  </rdf:Description>

  <rdf:Description ID="Address">
    <rdfs:comment>
      The geolocation of an object, expressed as a mailing address.
    </rdfs:comment>
    <rdf:type resource="rdfs:Class"/>
    <rdf:subClassOf resource="#Location">
  </rdf:Description>
  <rdf:Description ID="street">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#Address"/>
  </rdf:Description>
  <rdf:Description ID="city">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#Address"/>
  </rdf:Description>
  <rdf:Description ID="province">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#Address"/>
  </rdf:Description>
  <rdf:Description ID="country">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#Address"/>
  </rdf:Description>
  <rdf:Description ID="postalCode">
    <rdf:type resource="rdfs:Property"/>
    <rdfs:range resource="xsd:string"/>
    <rdfs:domain resource="#Address"/>
  </rdf:Description>
```

```
<!-- Time -->

<rdf:Description ID="time">
  <rdfs:comment>
    The time bucket.
  </rdfs:comment>
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="#Time"/>
</rdf:Description>

<rdf:Description ID="Time">
  <rdfs:comment>
    The possible contents of the time bucket.
  </rdfs:comment>
  <rdf:type resource="rdfs:Class"/>
</rdf:Description>

<rdf:Description ID="Lifetime">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Time"/>
</rdf:Description>
<rdf:Description ID="begin">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:dateTime"/>
  <rdfs:domain resource="#Lifetime"/>
</rdf:Description>
<rdf:Description ID="end">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:dateTime"/>
  <rdfs:domain resource="#Lifetime"/>
</rdf:Description>

<rdf:Description ID="timestamp">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:dateTime"/>
  <rdf:subPropertyOf resource="#Time"/>
</rdf:Description>

<rdf:Description ID="lifetime">
  <rdf:type resource="rdfs:Property"/>
  <rdf:subPropertyOf resource="#time"/>
  <rdfs:range resource="#Lifetime"/>
</rdf:Description>
<rdf:Description ID="metadataLifetime">
  <rdf:type resource="rdfs:Property"/>
  <rdf:subPropertyOf resource="#lifetime"/>
</rdf:Description>
<rdf:Description ID="metadataModified">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subPropertyOf resource="#timestamp"/>
</rdf:Description>

<rdf:Description ID="UpdateInterval">
  <rdfs:comment>
    Property of objects which are updated at some regular interval.
    Represents the
  </rdfs:comment>
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Time"/>
</rdf:Description>
<rdf:Description ID="interval">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:duration"/>
  <rdfs:domain resource="#UpdateInterval"/>
</rdf:Description>

<!-- Description -->

<rdf:Description ID="description">
  <rdfs:comment>
    A textual description of the resource.
  </rdfs:comment>
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="#Description"/>
</rdf:Description>

<rdf:Description ID="Description">
  <rdfs:comment>
    General description of a resource.
  </rdfs:comment>
  <rdf:type resource="rdfs:Class"/>
</rdf:Description>
<rdf:Description ID="text">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:string"/>
  <rdfs:domain resource="#description"/>
</rdf:Description>
<rdf:Description ID="translatedFrom">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:string"/>
  <rdfs:domain resource="#description"/>
</rdf:Description>

<!-- Originators -->

<rdf:Description ID="originator">
  <rdfs:comment>
    An owner or manager of the object or this metadata.
  </rdfs:comment>
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="#Originator"/>
</rdf:Description>

<rdf:Description ID="Originator">
  <rdf:type resource="rdfs:Class"/>
</rdf:Description>
<rdf:Description ID="origin">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:string"/>
  <rdfs:domain resource="#Originator"/>
</rdf:Description>
<rdf:Description ID="metadata">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:boolean"/>
  <rdfs:domain resource="#Originator"/>
</rdf:Description>
<rdf:Description ID="onlineContact">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:anyURI"/>
  <rdfs:domain resource="#Originator"/>
</rdf:Description>

<rdf:Description ID="Owner">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Originator"/>
</rdf:Description>
<rdf:Description ID="Author">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Originator"/>
</rdf:Description>
<rdf:Description ID="Manager">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Originator"/>
</rdf:Description>
<rdf:Description ID="Distributor">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Originator"/>
</rdf:Description>
<rdf:Description ID="Signer">
  <rdf:type resource="rdfs:Class"/>
  <rdf:subClassOf resource="#Originator"/>
</rdf:Description>

<!-- Links -->

<rdf:Description ID="link">
  <rdfs:comment>
    A connection to an online representation or manifestation of the
    resource being described.
  </rdfs:comment>
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="#OnlineResource"/>
  <rdfs:domain resource="rdf:Description"/>
</rdf:Description>

<rdf:Description ID="OnlineResource">
  <rdf:type resource="rdfs:Class"/>
</rdf:Description>

<rdf:Description ID="format">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:domain resource="#OnlineResource"/>
</rdf:Description>
<rdf:Description ID="mimeType">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:subPropertyOf resource="#format"/>
  <rdfs:range resource="xsd:string"/>
</rdf:Description>
<rdf:Description ID="dtd">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:subPropertyOf resource="#format"/>
  <rdfs:range resource="xsd:anyURI"/>
</rdf:Description>
<rdf:Description ID="xmlSchema">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:subPropertyOf resource="#format"/>
  <rdfs:range resource="xsd:anyURI"/>
</rdf:Description>

<rdf:Description ID="copyright">
  <rdfs:comment>
    A copyright restriction.
  </rdfs:comment>
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:string"/>
  <rdfs:domain resource="#OnlineResource"/>
</rdf:Description>

<!-- Link annotations -->

<rdf:Description ID="annotation">
  <rdfs:comment>
    An annotation on this particular link.
  </rdfs:comment>
  <rdf:type resource="rdfs:Property"/>
  <rdfs:subPropertyOf resource="#description"/>
  <rdfs:domain resource="#OnlineResource"/>
</rdf:Description>

<rdf:Description ID="Annotation">
  <rdf:type resource="rdfs:Class"/>
  <rdfs:subClassOf resource="#Description"/>
</rdf:Description>
<rdf:Description ID="annotator">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:string"/>
  <rdfs:domain resource="#Annotation"/>
</rdf:Description>
<rdf:Description ID="annotationLifetime">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:subPropertyOf resource="#lifetime"/>
  <rdfs:domain resource="#Annotation"/>
</rdf:Description>
<rdf:Description ID="md5">
  <rdf:type resource="rdfs:Property"/>
  <rdfs:range resource="xsd:string"/>
  <rdfs:domain resource="#Annotation"/>
</rdf:Description>

<rdf:Description ID="Review">
  <rdfs:comment>
    An external review of this resource and link.
  </rdfs:comment>
  <rdf:type resource="rdfs:Class"/>
  <rdfs:subClassOf resource="#Annotation"/>
</rdf:Description>

<rdf:Description ID="Accreditation">
```

```
    <rdfs:comment>
      A note describing some accreditation of the resource,
      e.g. accurate for navigation.
    </rdfs:comment>
    <rdf:type resource="rdfs:Class"/>
    <rdfs:subClassOf resource="#Annotation"/>
  </rdf:Description>

</rdf:RDF>
```