Virtual Sardinia: a Hypermedia Fly-through with Real Data

Enrico Gobbetti[1 2], Andrea O. Leone[3], and Alberto Marini[4]

[1] UMBC, CSEE Department, Baltimore MD 21228-5398, USA
[2] CESDIS, NASA Goddard Space Flight Center, Greenbelt MD, USA
[3] CRS4, Via Nazario Sauro 10, 09123 Cagliari, Italy
[4] Dip. di Scienze della Terra, Università di Cagliari, 09100 Cagliari, Italy

Abstract. The Virtual Sardinia project aims at collecting a large amount of heterogeneous data concerning the island of Sardinia and representing them in such a way that a casual user can easily navigate through them in a virtual trip. All these data are interconnected in an hypermedia way, browsable in the World-wide Web, ranging from geographic to archaeological data, from historical to touristical info, both in 2D and 3D. The central component of Virtual Sardinia is i3D, a high-speed 3D scene viewer for the World-wide Web. Using a Spaceball, the user can intuitively navigate with continuous viewpoint control inside three-dimensional data, while selecting 3D objects with the mouse triggers requests for access to remote media documents that can be distributed over the Internet. For the Virtual Sardinia project, the main 3D model that is explored by the users is a three-dimensional reconstruction of the island of Sardinia built from a digital terrain model texture-mapped with satellite images.

1 Introduction

The World-wide Web (WWW) has rapidly become one of the fundamental structures of the Internet. By imposing a universal organization on the variety of formats in which data resides around the world, and allowing each piece to be viewed as a uniquely addressable data source, it has allowed the entire Internet to be treated as a single structured document [2]. This enormous distributed database, which can be universally accessed using a single software application known as a web browser, takes the form of a hyper-media document, combining text, images, sound and video into a seamless, hyperlinked user-interface. Recently, with the advent of graphics workstations able to display scenes composed of hundreds of thousands of polygons at interactive speeds, it has become possible to consider including 3D scenes as one of the types of documents that can be hyperlinked in this information universe. VRML browsers, that offer interactive viewing of 3D scenes and hyperlink selection capabilities are starting to be developed for integrating 3D graphics with the WWW [6]. The relatively low-cost of modern 3D graphics workstations makes it possible for a large community to benefit from this evolution.

The Virtual Sardinia project builds on these recent advances. Its goal is to collect a large amount of heterogeneous data concerning the island of Sardinia and representing them in such a way that a casual user can easily navigate through these data in a virtual trip. The front end for accessing these data is a detailed 3D reconstruction of the island of Sardinia that can be explored in real-time. All the related data are interconnected in an hypermedia way, browsable in the WWW, ranging from geographic to archaeological data, from historical to touristical info, both in 2D and 3D. Thanks to this project, Sardinia becomes a wonderful place to visit also in the intangible Web information universe.

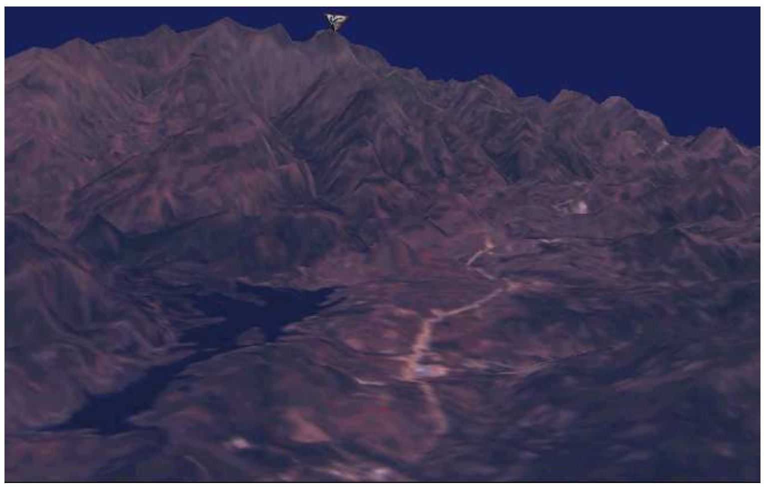

Fig. 1. Sight-seeing in Virtual Sardinia

In this paper we provide an overview of the Virtual Sardinia project. In particular, we describe the process involved in generating the 3D model from height and elevation data, give some detail on the interactive 3D viewer, discuss application areas, and present a view of our present and future work. Additional information, including videos of interactive sessions, is available on the WWW at the address http://www.crs4.it/PRJ/VIRTSARD/.

2 Generating a 3D Model

In the past, in traditional cartography, representation of geographic informations were limited to only two dimensions. The same satellite images, layed in a three dimensional model, can now point out shapes and brakes never observed in the past. The same habitat and landscape boundaries have a simpler connotation, identifiable and able to be studied without limitations of traditional representations. The first step in our project has been to create from elevation data and satellite images of Sardinia a 3D model appropriate for real-time exploration on a graphics workstation.

2.1 Original Data

The original data comes from the Landsat satellite. The satellite continuously get images of the globe from a distance of 920 km, following a course of 9 degrees toward east, allowing a vision of a strip 180 Km wide. It transmits reflectivity values of the solar energy for each square area of 30 meters per side. Each value is composed by seven data corresponding to different portions of the electromagnetic spectrum, ranging from visible to infrared. For the same area the satellite thus transmits seven different images, each corresponding to a different wave length. Each image can be thought of as a black and white photo, with 256 different gray levels per pixel.

The elevation data consists of a 359x681 regular grid matrix of equally spaced point each 400 meters, with elevation values.

2.2 Data Registration

Several steps have to be performed to obtain the final model of the Sardinia island. A first step involves the registration of the data with respect to a geographical system, taking into account the satellite viewing direction and the earth sphericity. A second step has to be performed to normalize values coming from different areas in order to have the same value distribution each corresponding bands. This problem arises because of different meteorological conditions during satellite image captures. Another step of the process involves the merging of different areas to obtain a single image. In our case we merged four different satellite images to cover the whole Sardinia island. The final step is the composition of images corresponding to different wave lengths to obtain the desired image. In our case we selected the blue, green and red wave length to have a realistic representation of the Sardinia, like a color picture taken from the space. The last operation is the color enhancement of the resulting image to correct unwanted deviation from visually realistic colors.

2.3 Data Semplification

The complexity of the terrain model represented as a regular grid is far too high to allow an interactive exploration even on high-end workstations. To be able to obtain interactive speeds during the terrain visualization, we have to optimize the model so as to allow the renderer to trade rendering quality with speed. All the optimizations are performed automatically by a software tool that we created that takes as input elevation data, satellite image, and desired quality parameters for the output.

Geometry Optimization The original terrain data (a single 359x681 regular grid containing the elevation data each 400 m) is subdivided in 16x10 quadrants (that can then be culled independently). Each of the 160 quadrants is then represented at 3 levels of detail by producing, from the original regular sub-grid, a series of irregular triangular meshes with a decimation algorithm that iteratively removes vertices from planar regions of the mesh [8]. Different planarity tolerances, adaptively adjusted during decimation, are used to obtain the desired 3 levels of details. Cracks between adjacent quadrants are avoided by using fixed low tolerances at the borders. The appropriate level of detail for each of the quadrants can then be chosen by the renderer at run time. Currently, the maximum complexity of the model after optimization is of 108617 triangles, while the minimum model complexity is of 41987 triangles.

Texture optimization Current graphics machines have limited memory for storing textures. The machine we use, a Silicon Graphics Onyx RE2 can only render textures that occupy less than 4MB. The satellite images have thus to be resized to fit in that size.

High-resolution version To allow users to view high-resolution versions of a region, we add hyperlinks from each of the quadrants to a high-resolution version of the region surrounding it, so that the interactive selection of a quadrant will trigger the loading and display of its high-resolution version. Since the high-resolution regions are smaller that the entire model and have access to the same hardware resources of the entire model when renderer (rendering speed, texture memory), more details can be included. These high-resolution versions are constructed automatically by our software tool by recursively applying to regions surrounding each quadrant the same algorithm we apply to the entire model.

2.4 Annotating the Model

Visual annotations on the 3D model are placed at interesting sites under the form of 3D objects with an associated URL. The interactive selection of one of these markers during navigation will be translated by the browser in a request to fetch and view the associated descriptive document, that can be in the form of text, still images, animations or even other 3D models.

3 Exploring the Model

A number of VRML-1.0 browsers that offer interactive viewing and hyperlink selection capabilities have been developed so far. These include Webspace [10], WebOOGL [11], and WebView [12]. Most of these systems are limited to mouse-based interaction and are not able to ensure constant high frame rates when dealing with large datasets, thus limiting their appropriateness for large scale projects. For the needs of the Virtual Sardinia projects we developed i3D

Fig. 2. Markers

[1], a high-speed 3D Web browser that incorporates the 3D input and high-performance rendering capabilities of high end VR system with the data fetching abilities of standard network browsers.

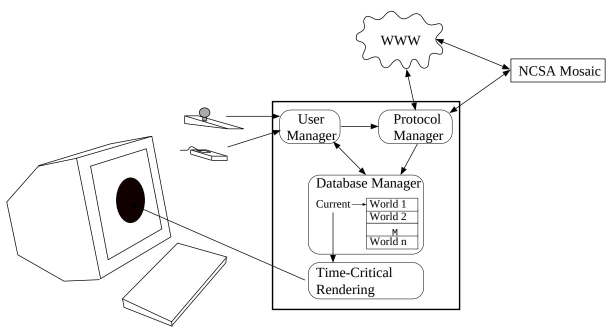

Fig. 3. Application overview

As shown in figure 3, i3D is composed of a user manager, a protocol manager, a database manager, and a rendering manager.

The user manager is responsible for sensing and analyzing the user's movements and actions in order to recompute the new viewpoint position and orientation, to trigger retrieval of media documents by the protocol manager, and to navigate among the stack of worlds that is maintained by the database manager. i3D's device configuration uses a Spaceball and a mouse as input devices.

The Spaceball is used for the continuous specification of the camera's position and orientation using an eye-in-hand metaphor[9], while the mouse is used to select objects and access media documents. Both user's hands can therefore be employed simultaneously to input information. Keyboard commands are used to control various visibility flags and rendering modes. The ability to continuously specify complex camera motions in an intuitive way together with high visual feedback rates provides an accurate simulation of motion parallax, one of the most important depth cues when dealing with large environments[3, 4].

The protocol manager is responsible for the retrieval of media documents from the World Wide Web. Three-dimensional scenes are loaded locally and transmitted to the database manager, while requests for other types of media documents are delegated to a WWW browser (NCSA Mosaic or Netscape) for retrieval and display by the most adequate viewer application.

The database manager maintains the state of the 3D scenes in order to provide the necessary geometrical information and visual attributes for the user and rendering managers to perform their tasks. It also maintains a stack of the scenes that have been visited to reach the current world and provides fast world switching upon user manager's requests.

The rendering manager is responsible for the generation of the visual representation of the current scene at a high and constant frame rate. During navigation, the rendering manager is activated at regular intervals by the main i3D event loop and is requested to refresh the screen while adhering to the user-specified timing constraint. At each activation, the rendering manager renders a single frame by executing a well defined sequence of operations.

First, the database is traversed and the objects visible from the observer's viewpoint are identified by hierarchically determining the visibility of portions of the scene through a traversal of the octree spatial subdivision maintained by the database manager. Each of the objects identified is then compiled into a graphical description by stripping off its appearance attributes and compiling them into a device-dependent sequence of drawing commands. During this conversion, geometries are optimized to reduce their rendering time; in particular, structured triangular meshes are generated from the triangle lists stored in the database. To avoid recreating compiled versions at each frame, as it is done in systems like Performer[7], the graphical descriptions generated for each database object are cached and reused while still valid.

To control the number of polygons rendered at each frame, so as to be able to meet the timing requirements, the rendering manager traverses the generated display list and selects the level of detail at which each of the objects will be represented. Level of detail selection is based on the importance of each object for the current frame, which is determined by computing an approximation of its size projected on the screen, and on feedback regarding the time required to render previous frames. The feedback algorithm is similar to the one used by Performer[7]. Update rates are associated to the different objects in the database to avoid recomputing their projection on the screen and their compiled versions at each frame.

Once the levels of details are selected, the display list is sorted to maximize rendering state coherence and rendered by executing each of the compiled command sequences for the selected level of detail of each of the objects. Rendering statistics for the current frame are updated and stored so as to be used when selecting the level of detail selection for the next frame.

Thanks to these rendering optimizations and to the preprocessing done on the data, the textured 3D model of the Sardinia island can be interactively explored at more than 10 frames/second on a Silicon Graphics Onyx RE2.

4 Conclusions and Future Work

The "Virtual Sardinia" project is well suited for the remote exploration of Sardinia in different interest area, from tourism to archaeology and geography. Heterogeneous information links can be easily added to the model in an incremental way, giving the ability to the visitors to retrieve different informations about Sardinia stored with respect to their location.

From a geographic point of view, scientist can use the model to examine different maps of Sardinia in a 3D environment, retrieving cartographic information directly tight to the elevation of the terrain. With our model it is now possible to analyze three-dimensional variations, both for territorial planning and structural geology analysis.

Future directions of the project will be toward: a finer resolution of the model, both for images and for elevation data; the addition of new hyperlinks to the model; the creation of "specialized" thematic models, e.g. a model only with touristical informations; different combinations of the satellite images to explore different aspect of the island, e.g. vegetation and water distribution.

The combined vision of the satellite images and the spatial model leeds to a new frontier in territorial variation interpretation.

5 Acknowledgements

i3D was designed and developed by Enrico Gobbetti at the Center for Advanced Studies, Research, and Development in Sardinia (CRS4), Cagliari, Italy, with help from Jean-Francis Balaguer. The work is now continued by Jean-Francis Balaguer at CERN, Geneva, Switzerland. The Virtual Sardinia processing tools have been developed by Enrico Gobbetti and Andrea Leone at CRS4. The satellite images and elevation data used in this work were provided by the University of Cagliari, "Scienze della Terra" department.

Research and development described in this paper have been carried out with the financial contribution of Sardinia Regional Authorities.

References

1. J.-F. Balaguer and E. Gobbetti: i3D: A High-Speed 3D Web Browser. Proc. ACM Symposium on VRML (1995).

2. T. J. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann: World-Wide Web: The Information Universe. Electronic Networking: Research, Applications, and Policy **2(1) (1992): 52–58.**
3. **F. P. Brooks Jr and P. Frederick: Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings, Proc. ACM SIGGRAPH Workshop on Interactive 3D Graphics (1986): 9–22.**
4. **T. A. Funkhouser and C. H. Séquin, Adaptive Display Algorithms for Interactive Frame Rates During Visualization of Complex Virtual Environments. Proc. ACM SIGGRAPH (1994): 247–254.**
5. **S. Ghee and J. Naughton-Green, Programming Virtual Worlds. ACM SIGGRAPH Tutorial Notes on Programming Virtual Worlds (1994): 6.1–6.58.**
6. **A. Parisi, M. Pesce: Virtual Reality Markup Language (VRML), URL: http://www.wired.com/vrml/ (1994).**
7. **J. Rohlf and J. Helman: Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. Proc. SIGGRAPH (1994): 381–395.**
8. **Schroeder WJ, Zarge JA, Lorensen WE: Decimation of Triangle Meshes. Proc. SIGGRAPH (1992).**
9. **C. Ware, and B. Osborne, Exploration and Virtual Camera Control in Virtual Three Dimensional Environments, Proc. ACM SIGGRAPH Workshop on Interactive 3D Graphics (1990): 175-183.**
10. **Webspace... because the World is not Flat! http://www.webspace.com/**
11. **WebOOGL: Integrating 3D graphics and the Web. http://www.geom.umn.edu/software/weboogl/**
12. **SDSC WebView - A VRML Internet Browser. http://www.sdsc.edu/EnablingTech/Visualization/vrml/webview.html**