

3D Terrain visualization for Web GIS

Chao Zhu, Eng Chong Tan, Tony Kai Yun Chan

Center for Advanced Media Technology
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798

czhu@camtech.ntu.edu.sg

Key words: Terrain, visualization, Web GIS, VRML

Abstract:

The Web has become a new medium which can display geographical information in rich forms and offer user-friendly interfaces. One of the promising trends in current Geographical Information System (GIS) is the use of Web 3D technology, especially VRML (Virtual Reality Modeling language). As one of the most fundamental geographical information, terrain information is often presented as the base for the display of other information, such as urban buildings, streets, population density, distributions of specific plants and so on. However, the 3D visualization of terrain on Web is a disaster for low bandwidth network users because of the large amount of data it involves. In this paper, we proposed and implemented a tile-based selective visualization method to facilitate 3D Web surfing. This tile concept also helps to develop a geo-information query system based on 3D terrain visualization. A Web GIS prototype has been designed and implemented featuring a hybrid 2D-3D interface.

1. Introduction

The combination of Virtual Reality (VR) and GIS is not really a new topic. From the early 1990s, scientists began to explore the possibility of using 3D visualization technologies to interactively represent urban buildings [Day 1994]. One of the major tools they used is VRML, a high level object-oriented language for the description of scenes and behaviors of 3D objects. VRML has been widely accepted as an international standard of interactive visualization since 1997, when ISO VRML97, or so called VRML2.0, is released [Web 1].

Kate Moore [Moore *et al.* 1997] started the project of The Virtual Field Course which used a VRML interface to explore a geographic database and multimedia resources fieldwork areas. The problems of geo-referenced VRML, accessing position by VRML sensors and changing the theme of information had been addressed. Year 1998 saw more expansion of VRML interfaced GIS. Shan [1998] integrated CAD/CAM, DPS and GIS data in a desktop environment based on the 3D Web. Terrain, buildings and their appearance have been successfully modeled and rendered. Coors and Jung [1998] created GOOVI-3D, a prototype system that provides access and interaction with a 3D spatial database over the WWW. GOOVI-3D was implemented in Java and VRML using Common Object Request Broker Architecture (CORBA) for accessing the data warehouse. In their work, the flexibility of VRML was explored by proposing two lightweight extensions of VRML: an integrated name dictionary management and a SQL node. At the same year, several Korean researchers [Lee *et al.* 1998] proposed a Web 3D GIS with spatial analysis functionality, featuring design of the module of Spatial Operation Manager with operations such as Near Analysis, 3D Buffering, Distance Measurement and Lantern Selection. In the last three years, researchers tend to develop more specific geo-related Web 3D applications in the area of architectural and archaeological restoration [Caiani *et al.* 2001], meteorological service [Chan *et al.* 2001] and ocean science [Campbell *et al.* 2002, McCann 2002]. VRML has been the favorite of most Web 3D GIS researchers for over ten years in that it is cheap, platform-dependent, can provide middle-quality interactive visualization and has high compatibility with other tools like Java.

However in most work mentioned above, VRML is mainly used as a display tool, especially in the early server-side Web GIS system. In a server-side Web GIS system, such as a CGI-based one, after

queries from users reach the Web server, the server will analyze the queries, retrieve relevant data from the data server and generate a new VRML file for displaying on the client side, i.e. the Web browser. Users mainly have to rely on a 2D web interface to submit their queries. The potential of VRML as interactive visualization tool has not been fully employed yet. The later research work has shown a tendency of choosing a client-side structure which passes the task of generating VRML file to the Web browser itself. For example, a Java applet or a ActiveX plug-in. However, for the visualization of a large-scale area, generating VRML nodes on the air will be too expensive in terms of interaction speed. More work is needed to explore the possibility and capability of VRML to visualize geo-information covering a large-scale area at the client-side. The best case would be 3D terrain visualization on the Web. Terrains often work as backgrounds for a lot of other information such as urban buildings, road topology, attributes distribution, and so on. A seamless combination of terrain visualization with other information layers is crucial to a good Web 3D GIS.

In this paper, we propose a selective visualization method for 3D terrain on the Web. The prototype implementation is integrated with a hybrid 2D-3D user interface design. The 3D terrain model we used is constructed from a 2D contour data file of the Singapore terrain.

2. Terrain Modeling

Terrain models can be classified into two types: Digital Elevation Model (DEM) and Triangulated Irregular Network (TIN) [Abdelguerfi *et al.* 1998]. VRML has build-in visualization mechanisms for both types: *ElevationGrid* node for DEM and *IndexedFaceSet* node for TIN. Generally speaking, a DEM is easier to be constructed than a TIN, but for a multi-resolution model, DEM structure may have the problem of terrain tearing between tiles with different resolutions. Although a TIN works well in the multi-resolution case, but its complex algorithms of the transformation between different resolution models impede its use in real-time applications. A DEM is adopted in our work not only because of its easy construction but also its compatibility with a TIN model. A DEM can be regarded as a very special case in TINs.

The contour file we have is an ASCII text file containing more than 4,0000 pairs of longitude/latitude coordinate values with elevation values. Pre-processing work is performed to connect broken contour lines and get rid of iterative points. A small 2D visualization application is implemented to give a clear view of contour layers (Figure 1). The lines with warmer colours, i.e. yellow, red and brown, indicates higher elevation contours.

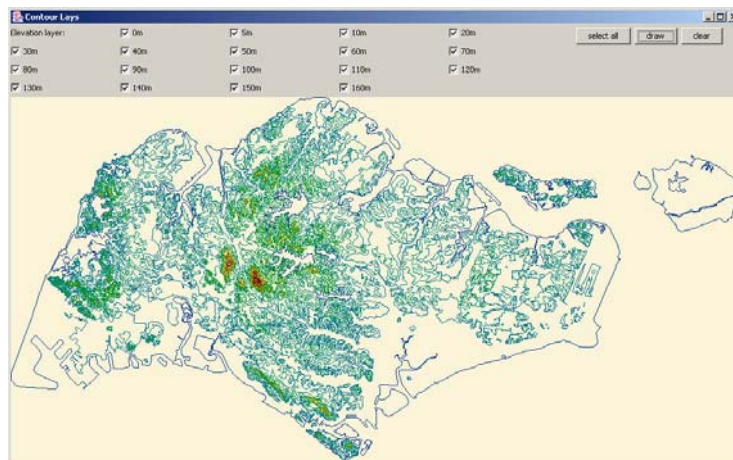


Figure 1: 2D visualization of the contour file

To construct the DEM from this contour file, we categorize the points into five cases according to their positions and treat them diversely. The five cases are mountain slopes, mountain tops, mountain ridges/valleys, sea and on-contour. Linear interpolation and cubic spline interpolation methods are employed to calculate elevation values. Since this only involves the given grid point and the contours in its neighboring area, this location-dependent method is much less time-consuming than the existing

one-point-to-all-contours methods. It has a good tradeoff between efficiency and accuracy. With this method, theoretically we can generate terrain models with any level of detail. Figure 2 illustrates the terrain model we constructed for later 3D display on the Web with a resolution of 512x256 grids.

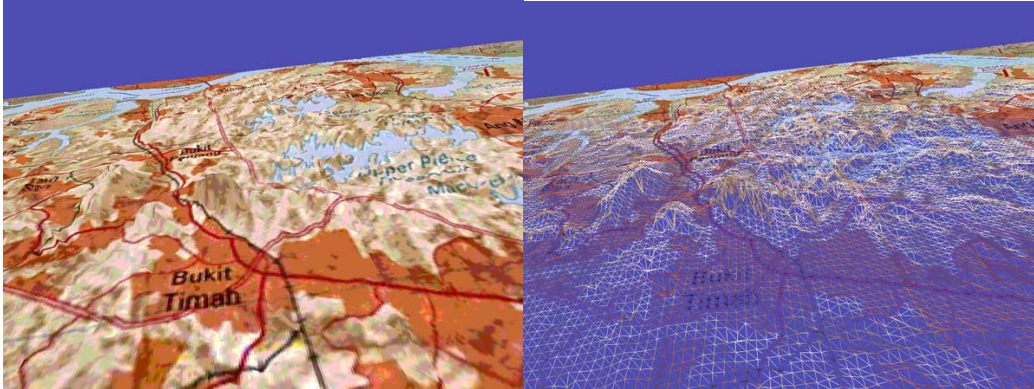


Figure 2: 512x 256 terrains shaded and wireframe

3. Terrain information Visualization on Web

VRML gives poor performances on visualization of large-scale areas because that it is an interpreted language, and not a compiled one. Therefore, assistant technology is necessary for better performance. One of the most common methods is to use Level of Detail (LOD) [Coors and Flick 1998, Kofler *et al.* 1998, Pajarola 1998]. VRML has its embedded LOD node; however, its fast speed of transformation between different levels is achieved at the price of long initializing time. When a VRML file is launched, terrain data of all levels need to be downloaded together, which normally will take longer time than what users can tolerate. The better way would be to change LOD while interaction or navigation. Some work has been carried out on progressive rendering of VRML. The Web3D Consortium [Web 2] set up a specific working group for the streaming VRML issue in 1998. However, due to the limitation of VRML specification itself, implementations in this area, especially its efficiency for large data sets is still under development with no breakthrough. Other researchers [Koh & Chen 1999] proposed an alternative way by implementing a new VRML browser that downloads 3D models in a progressive format, based on the MPEG-4 verification model source code. However, so far no further work for its application has been published. In terms of graphics, the efforts on progressive VRML are exciting. But for a Web GIS, the purpose of visualization is not only a vivid representation of reality but also a proper interpretation of information. The progressive rendering of terrain is accompanied with the problem of the presentation of other terrain-related information, for example, roads. When the surface of terrain is changed to another LOD, roads on it are supposed to follow the change too; otherwise, tearing occurs between the terrain surface and the roads. In one word, unless the progressive rendering of other information can also be achieved, the solo progressive rendering of terrain does not make much sense to a Web GIS.

Our solution is a selective visualization method to partially visualize the terrain surface near the viewpoint. Compared with the LOD technology displaying faraway area in low resolution, this method ignores the barely visible faraway area and only visualizes the area within current view culling. Users can also update the viewing area by querying the database through the user interface. The view culling is managed by switching and displaying terrain data in tiles. To do this, a new STILE node with the PROTO mechanism of VRML is designed. The new node contains the information about tileID, resolution, spacing, DEM elevation values, texture, visibility and other geo-information. When the visibility is triggered as true, the node will send out inquiry event, download geometry, texture and other information data from the database and visualize them in the VRML browser. This function is supported by the Java mechanism embedded in VRML. Figure 3 describes the basic structure of a STILE node. The information of a tile is updated in a dynamic way which means that data transmission and visualization occur when users are navigating in the virtual scene. It is implemented by using VRML Script node and EAI (External Authoring Interface). Both of them reference objects and behaviours in VRML and offer the interface to communicate between VRML scene and Java programs containing a connection to database via JDBC (Java Database Connection).

```

PROTO STILE
  field SFInt32 tileID -1
  field SFInt32 resolution 32
  field SFFloat spacing 1
  exposedField SFBool          extendOnXPos TRUE
  exposedField SFBool          extendOnXNeg TRUE
  exposedField SFBool          extendOnZPos TRUE
  exposedField SFBool          extendOnZNeg TRUE
  eventIn  SFBool set_visible
  eventOut SFBool visible_changed
  eventOut SFInt32 getTileID
]
{
  DEF TileSwitch Switch {
    choice [
      DEF TILE Transform {
        children [
          Shape {
            geometry DEF DEM ElevationGrid {
              xDimension IS resolution
              xSpacing IS spacing
              zDimension IS resolution
              zSpacing IS spacing
            }
            appearance Appearance {
              material Material {}
              texture
                DEF TEXTURE ImageTexture {}
            }
          }
        ]
        translation 0 0 0
      }
    ]
    whichChoice -1
  }
  DEF TileScript Script {
    field SFNode tileSwitch USE TileSwitch
    field SFNode tile USE TILE
    field SFNode dem USE DEM
    field SFNode texture USE TEXTURE
    field SFInt32 tileID IS tileID
    eventIn  SFBool set_visible IS set_visible
    eventOut SFBool visible_changed IS visible_changed
    eventOut SFInt32 getTileID IS getTileID
    url "http://.../Tile.class"
  }
}

```

Figure 3: A STILE node interface

For the terrain model generated in Section 3, an 8x4 tiles division is made. The size of a tile is determined by the performance of visualization and interaction after several trials. VRML *ProximitySensor* is used to trace the position and orientation of current viewpoint. With these values, it is easy to locate the current view tile from which a set of tile IDs under the view culling can be obtained. Figure 4 illustrates an example of updating the viewpoint.

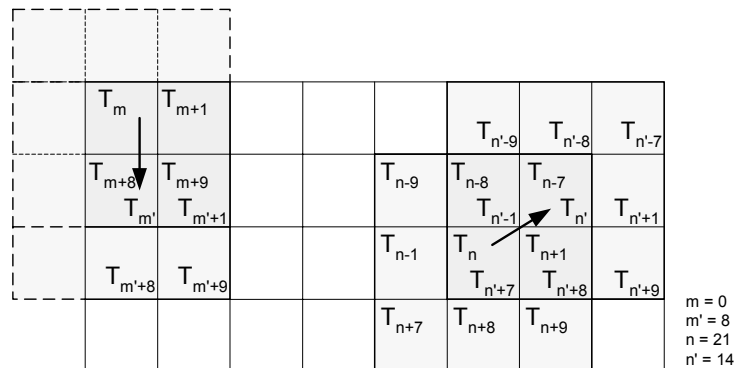


Figure 4: An example of updating the viewpoint

The entire selective visualization algorithm is specified as follows:

- Step 1:* Initialize 32 STILE nodes and obtain Java objects that reference the VRML nodes and useful fields.
- Step 2:* Get the current position and orientation value by referencing *ProximitySensor* node. The index number of current tile, denoted as *currTileID*, can be obtained as well with the help of *resolution* and *spacing* fields of STILE node.
- Step 3:* Collect the *tileID* values of the tiles that are set to be visible from 32 tiles by checking the *visible_changed* eventOuts and put the results to a set named *tilesBeforeVis*. With *currTileID* and the corresponding attributes of the extensibility of this tile (Table 1), a set named *tilesAfterVis* is established to index the tiles that are theoretically within view.
- Step 4:* Compare the elements in the set *tilesBeforeVis* with those in the set *tilesAfterVis*. If there are any iterative elements, delete them in both sets.
- Step 5:* Set the values of *set_visible* eventIns of the tiles indexed in *tilesAfterVis* as true and set the values of *set_visible* eventIns of the tiles indexed in *tilesBeforeVis* as false.

4. Prototype design and implementation

A prototype application, Singapore Terrain Explorer (STE), is designed to serve as a testbed for the abovementioned visualization method. Six main components are involved: 3D scene, local map viewer, world map viewer, coordinates display, information board and query pad.

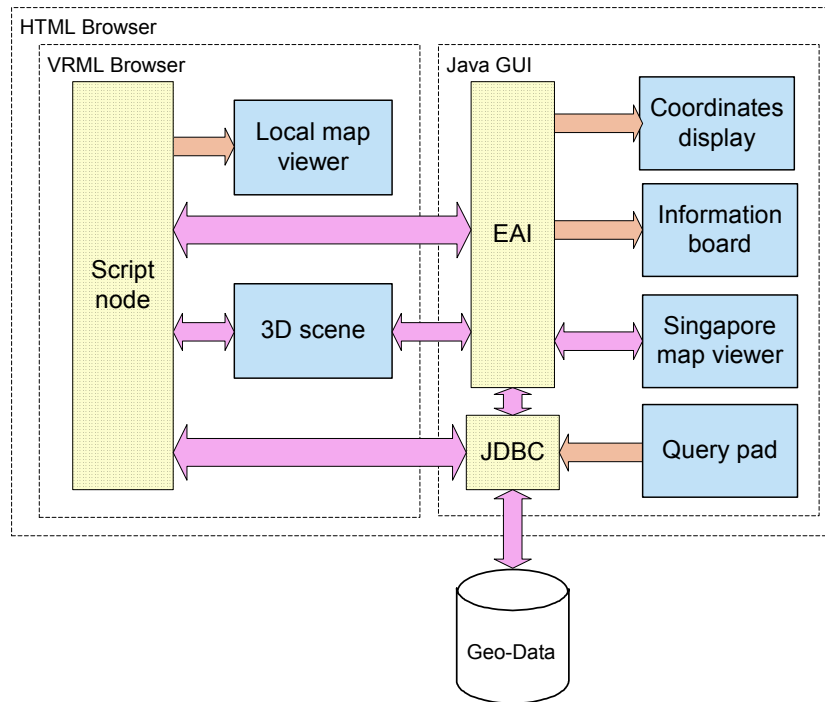


Figure 5: Prototype design of STE

The prototype implementation is on the basis of the integration of Java applet and VRML browser. Apache Web server is adopted. Java Database Connection (JDBC) acts as the interface to retrieve all geo-related information from the database. The communications between different VRML components such as the 3D scene and the local map view is done by embedded *Script* node in VRML. Other 2D components, i.e. the coordinate display, the information board, the Singapore map view and the query pad, are connected with the VRML components by EAI. If using Java scripts in *Script* node, information can be exchanged between scripts and EAI programs. Both of them have JDBC connection with geo-database. As a prototype, we simplify other geo-information as a table of names of the place of interest, important architectures, and subway stations. Figure 5 illustrates the prototype design of STE, especially the relationships among the six components in terms of implementation.

In the 3D scene, users can interactively explore the geographical information by flying through/walking on the terrain surface and switch on/off different layers of theme data. A further click on a specific object may link to a new HTML page describing the geographical details at this location. The local map viewer is a floating 2D panel over the 3D scene, which contains a map of current terrain tile with a marker to point out the position and orientation of the viewpoint in the virtual world. The world map viewer tracks the current viewpoint too, but the main use is to present the global view of all terrain tiles. It also offers the short cut to quickly move to another location by simply clicking on the desired place on the map. The coordinates display shows the longitude/latitude coordinates values of current viewpoint position to help users to have a position reference about the environment. The information board is used to present local geographical information related to the current tile. With the view area changing, the contents on board are updated as well. The query pad gives users the opportunity to textually enquire about specific information from the database, such as asking for information about a place of interest. The layout of interface design is displayed in Figure 6.

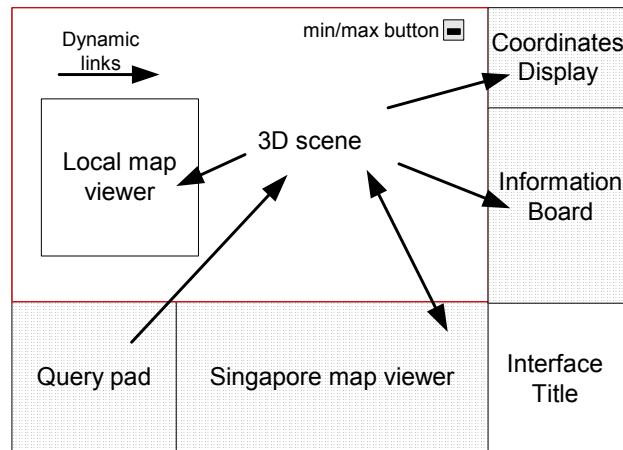


Figure 6: Layout of prototype interface design

The test of the above Web 3D GIS prototype has been done with a set of PCs distributed within the same LAN. The information database is stored in one of them as central data server. A LAN environment has similar distributed feature as the Web but can save the work of security checking. So in the prototype stage, we prefer to use the LAN than the Web. Any machine in the LAN can open the Web GIS site, retrieve and handle the 3D data by only installing the free VRML browser. The navigation and interaction show the design is efficient and effective. Figure 7 is the home page of STE. Figure 8 is a screenshot of navigation with STE.

5. Conclusion and Future Work

In this paper, we propose a prototype of Web 3D GIS on the basis of 3D terrain visualization. As the implementation of this prototype, Singapore Terrain Explorer provides an efficient and interactive environment of exploring geographical information on the Web. The tile-based selective visualization methods are employed to improve the real-time visualization for large-scale surface with VRML. Moreover, STE successfully integrates the advantages of both 2D and 3D interface and achieves dynamic links between various interface components. Future work may include the improvement on the speed of retrieving geo-data from database for the visualization of terrain tiles by the means of concurrent processing. Multi-resolution visualization can also be integrated into the selective visualization. More work can be done to design a better data structure for the storage of a large amount of other information on database and their visualization by VRML. As a result, an intelligent enquiry system can be accomplished in a complete Web 3D GIS.

Reference:

- [1] Abdelguerfi, M., Wynne, C., Cooper, E. Roy, L. and Shaw, K., Representation of 3-D elevation in terrain database using hierarchical triangulated irregular networks: a comparative analysis, *INT. J. Geographical Information Science*, 1998, Vol. 12, No. 8, pp. 853-873.

- [2] Gaiani, M., Gamberini E. and Tonelli, G., VR as work tool for architectural & archaeological restoration: the ancient Appian way 3D web virtual GIS, *Virtual Systems and Multimedia, 2001, Proceedings*. Seventh International Conference, 25-27 Oct. 2001, pp. 86 -95
- [3] Campbell, B., Collins, P., Hadaway, H., Hedley, N. and Stoermer. M., Web3D in Ocean Science Learning Environments: Virtual Big Beef Creek, *Web3D '02*, February 24-28, 2002, Tempe, Arizona, USA.
- [4] Chan, Celyn S. L., Chan, Tony K. Y. and Prakash, E. C., *Pan-Sydney workshop on Visualisation* , Sydney, Australia, December 2000, Volume 2, pp. 23 - 29
- [5] Coors, V., Flick, S., Integrating Levels of Detail in a Web-based 3D GIS, 7th, *ACM Workshop on Geographic Information Systems*, Washington, USA, Nov. 6-7, 1998.
- [6] Coors, V., Jung, V., Using VRML as an Interface to the 3D Data Warehouse, *Proceedings of the VRML '98 Symposium* (Monterey CA, Feb. 16-19), ACM Press, New York, 1998.
- [7] Day, A., From map to model: the development of an urban information system, *Design Studies*, 15(3), July 1994.
- [8] Kofler, M., Gervautz, M. and Gruber, M., The Styria Flyover-LOD management for huge textured terrain models, *Computer Graphics International, 1998 Proceedings*, 1998, pp. 444 -454.
- [9] Koh, B. and Chen, T., Progressive browsing of 3D models, *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on*, pp. 71 –76.
- [10] Lee, H.G., Kim, K. H.; Lee, K., Development of 3-dimensional GIS running on Internet, *Geoscience and Remote Sensing Symposium Proceedings 1998, IGARSS '98. 1998 IEEE International Volume 2*, 1998, pp. 1046 –1049.
- [11] McCann, M. P., Creating 3D oceanographic data visualizations for the web, *Proceeding of the seventh international conference on 3D Web technology*, Tempe, Arizona, USA, pp. 179 - 184
- [12] Moore, K., Dykes J., & Wood, J., Using Java to interact with geo-referenced VRML within a Virtual Field Course, *ICA Visualization Commission meeting*, Stockholm, June 1997.
- [13] Pajarola, R., Large Scale Terrain Visualization Using The Restricted Quadtree Triangulation, *Visualization '98 Proceedings*, 1998, pp. 19 -26, 515.
- [14] Shan, J., Visualizing 3-D geographical data with VRML, *Computer Graphics International, 1998, Proceedings*, pp. 108 –110.

Web Reference:

- [Web 1] ISO/IEC DIS 14772, 1997, The Virtual Reality Modeling Language: (VRML 97), <http://www.vrml.org/technicalinfo/specifications/vrml97/index.htm>
- [Web 2] VRML-Streaming Working group, <http://www.web3d.org/WorkingGroups/vrml-streams/>

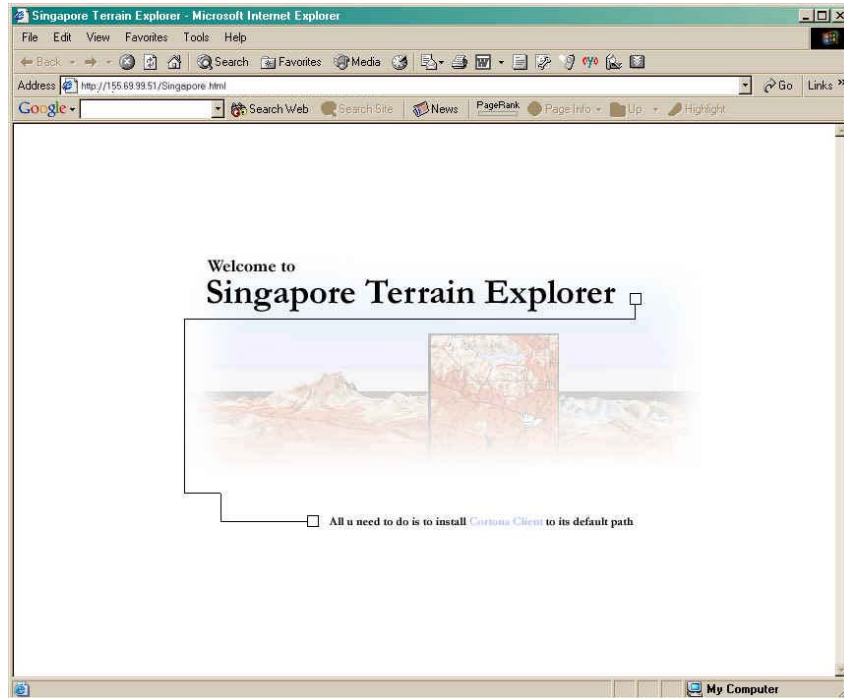


Figure 7: STE homepage

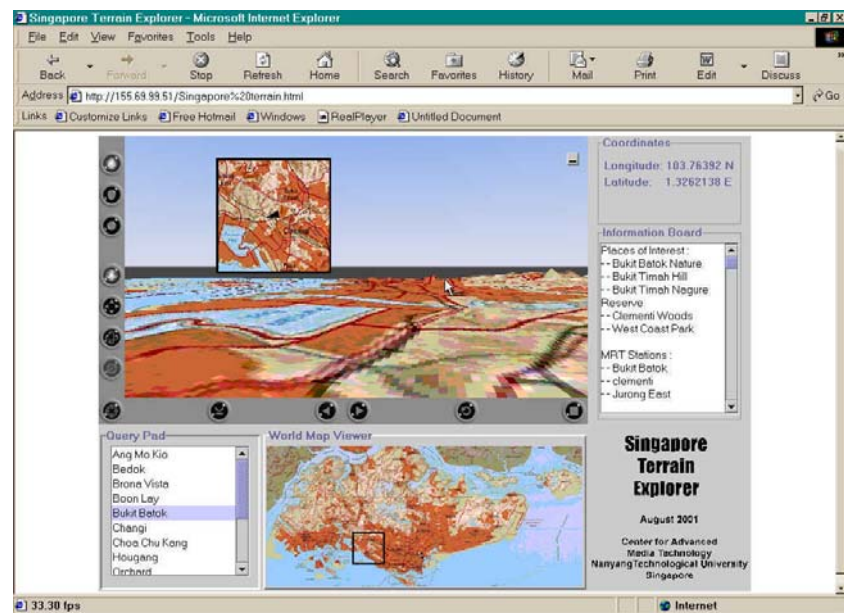


Figure 8: Navigation with STE