3D NAVIGATION SYSTEM FOR VIRTUAL REALITY BASED ON 3D GAME ENGINE

K.H. Sharkawi, M.U. Ujang and A. Abdul-Rahman

Department of Geoinformatics,
Faculty of Geoinformation Science and Engineering,
Universiti Teknologi Malaysia,
81310 UTM Skudai, Johor,
Malaysia.

{hafiz | uznir | alias}@fksg.utm.my

KEY WORDS: Spatial Information Sciences, GIS, Navigation, Visualization, Virtual Reality

ABSTRACT:

Intense competitions in computer related industries have lead to the development of new amazing technologies. This drastically improved 3D game engine, which was initially developed for computer games, to become more adaptable and versatile thus enables it to be incorporated with GIS applications for them to support 3D capabilities. 3D navigation is one of the 3D GIS components that benefited from this technology. Visualizing geospatial data in 3D environment is more relevant for navigation systems rather than using symbolic 2D maps because 3D visualization can provides users with the real spatial information (colors and shapes) and the 3D models resembles the real world objects. Navigational landmarks and other features that can be very helpful when navigating in the real world are easily spotted in the 3D environment but hardly available in 2D maps. By incorporating a database, users can even carry out some spatial analyses that greatly improve the system's reliability. This paper describes the development of 3D navigation system in virtual 3D (indoor and outdoor) environment by utilizing a freely available 3D game engine couple with GIS elements.

1. INTRODUCTION

The state-of-the-art technologies on computer hardware and software makes 3D capabilities the must-have feature and highly demanded in most recent visualization-related software e.g. GIS software. The entertainment and computer/video game industries have invested a lot of time, money and efforts to develop high performance and powerful 3D engines that are capable to visualize 3D models look like the real objects. Rapid development and improvement on the modern engines makes it more versatile, thus, increases the possibility of using the same mechanism (3D engine) for other applications.

The incorporation of 3D visualization and analysis with GIS can greatly improve the users' comprehension on the spatial problems that occur and assist them in decisions making process. The visualization also improves the reliability of the GIS application as an effective decision making tool. Fritsch (2003) and Fritsch and Kada (2004) discussed on visualization using game engine for indoor and outdoor environments and listed several engines such as *Quake III* for indoor and *Unreal engine 2* for both indoor and outdoor visualization.

3D navigation is one of the components in 3D GIS. Many navigation systems use 2D maps to represent the real world objects and unable to provide users with enough information on real world objects such as its colors, shapes and materials. Features and navigational landmarks were not easily available even though they were very useful and helpful for navigating in the real world. According to Coors et. al (2004), people recognize landmarks and find routes in the cities easier by using 3D models than using a symbolic 2D map which makes 3D visualization more relevant for (3D) navigation and due to the high visual correspondence between map objects and real world objects, it increases the navigational value of the 3D map (Coors et. al., 2006). Some research works related to navigation have been addressed by Coors et. al. (2004) where they presented routes on mobile devices using 2D and 3D maps, supported by text and images instructions while Meijers et. al. (2005) and Pu and Zlatanova (2005) focused on indoor navigation routing.

The remainder of this paper reviews the general aspect of 3D game engine, introduces the engine used in this research and describes the experiments on developing 3D navigation system for indoor and outdoor environment with some analyses by utilizing a freely available 3D game engine as a navigation tool coupled with programming efforts.

2. 3D GAME ENGINE

3D engine or game engine is the core software component for any applications that uses real-time graphics displays such as electronic video games (computers and consoles) and other interactive applications that contain 2D or 3D graphics rendering.

3D engine often comes in packages called System Development Kit (SDK). The SDK usually contains several core functionalities for the developers to develop their applications such as rendering (so-called "renderer") engine, physics engine, media engine, scripting, Artificial Intelligence (AI) and networking capability. The renderer is used to display 2D and 3D scene to the users so that they can make their decisions based on the displays. This function is the most important part in the engine because this is where the engine will be judged by users. For most users (especially gamers), this make-or-break function depicts the quality of the engine and ultimately, the 3D games or applications that are built upon it. Most of the current game engines are optimized for both indoor and outdoor visualizations (e.g. *Unreal engine 2* and *Unreal engine 3* developed by *Epic Games*), however some of the engines are dedicated only for indoor visualization (e.g. *Doom* and *Quake* engine developed by *id Software*) in order to keep the engine's performance at a higher rate. The physics engine is used to simulate most of the physics applied in the application such as the gravity and the ballistics law meanwhile the media engine handle the sounds, videos and animations that are use in the application.

In 3D games, the reactions of the Non-Player Characters (NPC) or computer-controlled characters are determined by the engine's AI. Artificial Intelligence (AI) is another important component (after the renderer) in 3D game engine in order for the games to response to any interactivity made by players. AI can be described as the ability of a machine, software or virtual characters to think and react like human beings. Games' artificial intelligence also refers to techniques used by the virtual characters to react and interact with the human characters. One of the elements in AI is the path finding algorithm.

Next section describes the 3D engine used in this research which is the *Truevision3D* engine.

2.1 The Truevision3D engine

The Truevision3D (TV3D) engine is a highly optimized engine that is developed upon the DirectX platform and fully support the DirectX8.1 and DirectX 9.0 features. The engine uses hundreds of built in shaders that adapts to computer products in order to take the advantage of current and future hardware. Apart from being optimized for indoor and outdoor environments rendering, it also supports multiple programming languages such as Visual Basic 6, Visual Basic.Net, C++, C# and Delphi, developers (programmers) can develop 3D games and applications using the programming language that suited their programming skills.

The engine comes with integrated Newtonian physics engine, based upon the *Newton Game Dynamics* physics engine with features such as direct integration for engine objects, including meshes, actors, terrain and accurate collision detection and friction solver with configurable accuracy.

The engine's SDK is available for free for non-commercial use but for developers that wish to commercialize their products (applications), the license can be obtained at a very affordable cost.

3. THE DEVELOPMENT OF THE 3D NAVIGATION SYSTEM

3.1 Indoor Navigation

Network Analysis for Indoor Navigation: Implementation of network analysis (i.e. shortest path analysis) mostly based on application objectives and situation. For instance, disaster based or emergency applications in urban areas need shortest path routine to navigate in a complex surrounding architecture. System that could provide a route network within a building helps to locate safety routes for example during fire and rescue missions and other emergency operations. Shortest path routines are useful in assisting human navigation. There are several types of navigations, for instance navigation for pedestrian and car (on roads or in racing circuit). Balstrom (2001) listed some of the applications that need those routine e.g. searching the nearest restaurant, a petrol station, jungle trekking on mountains/hills and locating the emergency exit doors during fire in a building. Shortest path network for multilevel buildings thus the connectivity between levels or floors are certainly be very useful and interesting to be investigated. Dijsktra algorithm could be used for this sort of applications that is by extending to three-dimensional (3D) environment. Some research works on Dijsktra algorithm for 3D environment have been investigated e.g. by Karas et al (2006). Other related works such as Meijers et al (2005) and Pu and Zlatanova (2005) focused on indoor navigation routing. This research describes the experiment on Dijkstra algorithm based on 3D CAD network data (i.e. inside of a building).

The basic structure for the 3D network is a group of connecting nodes that consists of x, y and z (height) values. The system will extract the values for the network analysis calculation.

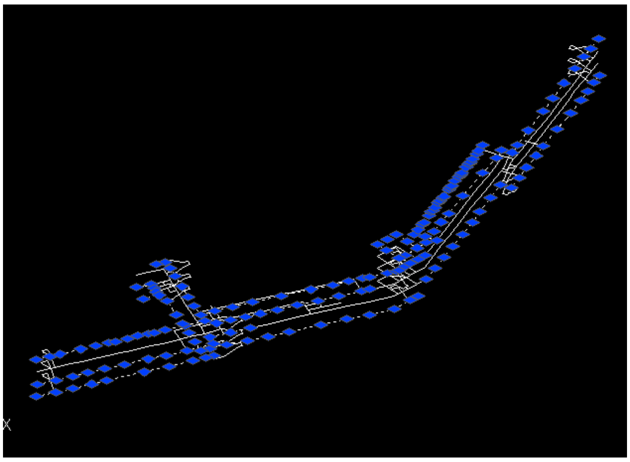

Figure 1. Generated 3D CAD network (Left) and some selected nodes in the network (Right)

Dijkstra's Algorithm for 3D Network: After we have established a 3D network for the indoor area, important information like IDs, distances, and potential candidates of nodes need to be retrieved, they are very important for the algorithm and can be extracted and assigned with proper IDs. By putting ID to each node, then the network is now applicable for Dijkstra's algorithm.

The next step is to calculate the distance between the connected nodes (i.e. successor and predecessor nodes). It is not necessary to calculate the distance between all the nodes in the network as this will consume a lot of time. If node #1 is connected to node #2 and node #3, then the calculation will be based on those three nodes. To accomplish that, information on predecessor and successor nodes is important. To make computers understand which nodes are connected or not, therefore the network data file must be examined thoroughly. From here the data file will be examined in order to find the useful data and to determine the connected/reachable node.

The distance between nodes for 3D network can easily be calculated by using simple mathematical formula, see Figure 2.

Example: distance between node A (x_1, y_1, z_1) to node B (x_2, y_2, z_2),

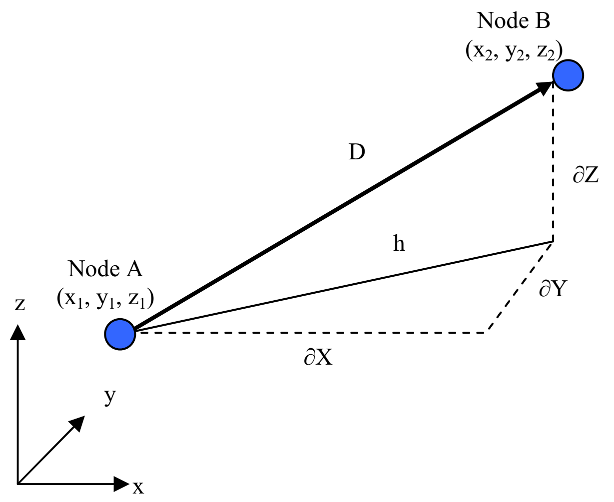

Figure 2. 3D point differential for node A (x_1, y_1, z_1) and node B (x_2, y_2, z_2) in 3D axis.

Delta (∂) for x, y and z can be calculate by finding the differential between nodes,

$$\partial X = (x_2 - x_1) \; ; \; \partial Y = (y_2 - y_1) \; ; \; \partial Z = (z_2 - z_1)$$

Then the distance (D) could be derived by two equations as below:

$$D = \sqrt{(h^2 + \partial Z^2)} \text{ ………………….. Equation 1}$$

h can be derived by:

$$h = \sqrt{(\partial X^2 + \partial Y^2)} \text{ ………………..Equation 2}$$

By combining both equation 1 and 2, the final equation to find the distance (D) between the nodes is:

$$D = \sqrt{[(\partial X^2 + \partial Y^2) + \partial Z^2]} \qquad \text{or}$$

$$D = \sqrt{[(x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2]}$$

Then from here, the Dijkstra's algorithm can be executed successfully as in the case of 2D network.

Indoor Shortest Path Result: Figure 3 shows the result of the Dijkstra's calculation statistics. The shortest path calculation windows displays the information about the *Status*, *Date*, *Calculation* time, *Nodes Calculated*, *Nodes Used*, *Shortest Distance*, and both information about current and destination location (including the x, y and z coordinates). The *Status* of the calculation is the information whether the executable calculation is successful or the destination location is not reachable for the user to navigate to. *Shortest Distance* shows the distance (shortest) in meters from current location to destination location. The system also able to show the shortest route graphically in the 3D network as illustrated (see Figure 3). The shortest route of the 3D navigation is highlighted.

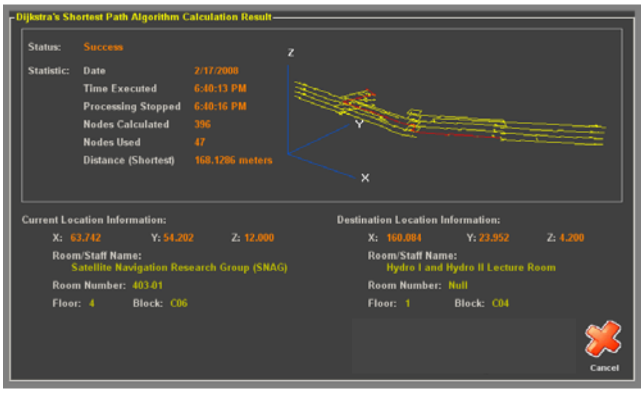

Figure 3. The Shortest Route Calculation Windows

3.2 Outdoor Navigation

The second component of the system is the outdoor navigation. The 3D outdoor navigation is developed to navigate users from location A to location B and at the same time provides them with the real spatial information of the area e.g. shapes of buildings, structures, and other objects. As mentioned in the previous sections, 2D navigation systems are proven to be useful in locating the desired locations, but with 3D navigation system, users can see how the real environment looks like, marks out the possible navigational landmarks (i.e. distinct-permanent objects or buildings) and familiarized themselves with the routes. Figure 4 shows the outdoor environment in the navigation system.

This navigation system, like in most 3D games, allows users to walk around and explore the outdoor environment. In detail, the system provides users with the following functionality e.g. users can set a desired destination and navigate to it, and retrieves information of the surrounding buildings.

Figure 4. 3D buildings with real textures in the outdoor environment

Spatial analysis is an extremely important aspect in GIS as it allows users to conduct analyses on available data and assists them in making better decisions based on the findings. This functionality is a huge advantage as it makes the GIS applications more valuable and useful. The engine offers some great functionality that commonly used in computer games and can be utilized as part of the 3D spatial analysis tools in this navigation system.

3D Network Analysis: As mentioned in previous sections, the network analysis in this research is used to determine the shortest path from point A to point B based on the available paths. Users can select their destination and the system will determine the shortest path from the network. The path network is a combination of static and dynamic nodes. Static nodes represent the permanent paths existed in the real world and unlikely to change in the future unless there are no other ways meanwhile dynamic nodes marks the temporary paths that only available for a short period of time. With the dynamic nodes, users can add or delete nodes to create or update new paths in case there are changes occur in the real world. If the determined shortest path is somehow blocked or inaccessible, the algorithm will find another route to reach the destination. The shortest path will be marked with navigational markers (arrows, lines) to guide users to their selected destination as shown in the Figure 5.

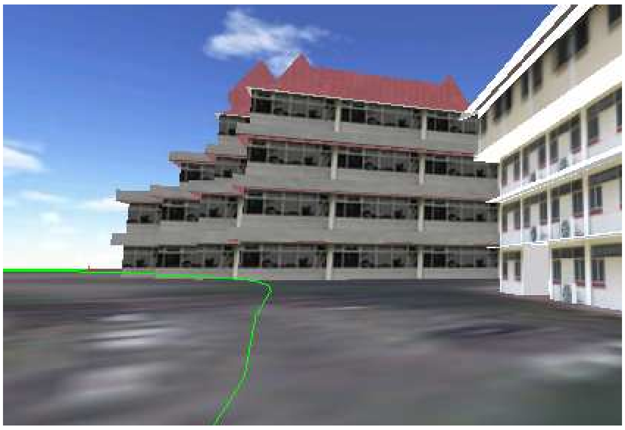

Figure 5. The system navigates users along the shortest path

Database Integration: Since data is a very important element in GIS-related application, database plays a crucial role in storing and managing those data. The attributes database is then linked with its spatial objects (3D buildings). All stored data can be edited, added or deleted during the runtime. The database is developed using proper DBMS software so that users can manage (add, edit or delete) the stored data externally (using the DBMS itself) and easier to convert the database from or to other database format. This database plays an important role for 3D spatial query that will be discussed in the next section.

3D Spatial Query: Spatial query on 2D environment is a common function in most GIS software. Since database can be incorporated with this program, it is possible for users to generate spatial query on a 3D objects in the 3D world. The engine's ability to select 3D object via Point-and-shoot (clicking on the object) technique allow users to retrieve information related to the 3D buildings by clicking on the buildings. The information displayed is retrieved directly from the attributes database as mentioned in the previous section. Users can add, edit or delete the information during runtime or using the DBMS itself.
Figure 6 shows selected building (i.e. the forefront building) and its information is shown on the screen.

Figure 6. A simple 3D spatial query on the forefront building

4. CONCLUSIONS

3D engines are originally developed for 3D games. However, as the technology grows, we have seen their applications being implemented for 3D geoinformation. The previous discussions show that the 3D engine works for virtual navigation purposes (with some GIS functionalities e.g. spatial analyses). Again, it clearly shows that the 3D engine with GIS elements significantly enhanced the navigation system.
For future work, the approach could be extended or incorporated with mobile devices and real-time positioning.

ACKNOWLEDGEMENTS

We would like to thank our research grant sponsor, Malaysian Ministry of Science, Technology and Innovation (MOSTI) and Universiti Teknologi Malaysia Research Management Centre (RMC) for their support (research grant number: Vote 79231).

REFERENCES

Balstrom, T., 2001. Identifying Least Cost Routes in Mountainous Terrain. http://gis.esri.com/library/userconf/proc00/professional/papers

Coors, V., C. Kray, K. Laakso, and C. Elting, 2004. Presenting route instructions on mobile devices. Book Chapter in "Geo-Visualization", 2004. Book Sections, 2004.

Coors, V., J. Haist, and T. Reitz, 2006. A web 3D services for Navigation Applications. 1st International Workshop on 3D Geoinformation 2006. Kuala Lumpur, Malaysia.

Fritsch, D., 2003. 3D Building Visualization – Outdoor and Indoor application. Photogrammetric week. 2003

Fritsch, D. and M. Kada, 2004. Visualisation Using Game Engines. ,XX ISPRS Congress 2004, Commission IV. Istanbul, Turkey.

Karas, I.R., F. Batuk, A. E., Akay, and I. Baz, 2006. Automatically extracting 3D models and network analysis for indoors. In. Innovations in 3D Geo Information Systems, Springer-Verlag, Heidelberg, pp. 395-404

Meijers, M., S. Zlatanova and N. Pfeifer, 2005. 3D Geoinformation Indoors: Structuring for evacuation. Proceedings of Next Generation 3D City Models, Bonn, Germany, pp. 11-16.

Shi, P., and S. Zlatanova, 2005. Evacuation route calculation of inner buildings. Geoinformation for disaster management. Springer-Verlag, Heidelberg, pp. 1143-1161.